Engine Pylon Topology Optimization Framework Based on Performance and Stress Criteria

Simone Coniglio*
Airbus, 31060 Toulouse, France
Christian Gogu†
University of Toulouse, 31400 Toulouse, France
Remi Amargier‡
Airbus, 31060 Toulouse, France
and
Joseph Morlier§
University of Toulouse, 31400 Toulouse, France

DOI: 10.2514/1.J058117

Reducing fuel consumption is a major driver for the design of future aircraft. The engine integration primary structure plays a significant role in the integrated engine thrust-specific fuel consumption. A topology optimization framework was developed to design the primary structure integrating the engine to the aircraft wing considering mass, stress, and engine performance criteria. The proposed approach had to address several challenges associated with the use of nonuniform meshes, the integration of the engine model as a super-element, and the presence of nonconforming mesh interfaces. Analytical adjoint evaluations for all the responses were also derived. The framework was tested on a simplified engine model providing a consistent solution.

Nomenclature

DOF	=	degrees of freedom
$\tilde{F}_c$	=	engine retained DOFs condensated load vector
$\tilde{K}_{cc}$	=	engine retained DOFs condensated stiffness matrix
$l_{(s)}$	=	sth-stage blade height
N_s	=	sth-stage number of angular position
$R_{(s)}$	=	sth-stage tip clearance root mean square
TSFC	=	thrust-specific fuel consumption
u_r^b	=	rotor blade tip radial displacement
u_r^c	=	casing radial displacement
$[K]$	=	stiffness matrix
$[P]$	=	engine recovery matrix
$\{F\}$	=	load vector
$\{r(\theta)\}$	=	radial unit vector at the angle θ
$\{U\}$	=	displacement vector
ΔTSFC%	=	TSFC approximative variation induced by mechanical load
δ_0	=	initial tip clearance
θ	=	angular position around the engine stage
$\{\delta_{(s)}\}$	=	sth-stage tip clearance variation vector

I. Introduction

A CIVIL aircraft power plant system (PPS) primary structure has as a primary function to attach the engine to the aircraft wing. Furthermore the design of all the components of the PPS, (pylon, engine mounts, and nacelle) has, however, also importance on the final integrated engine performance. For example, the variation of the radial clearances at the blade tip of each stage (called tip clearances) affects engine time on wing, compressor surge margin, and the thrust-specific fuel consumption (TSFC) [1,2]. Controlling the tip clearance variation due to engine maneuvers is for these reasons a major criterion considered during the PPS design. Including engine performance in the pylon design loop was investigated by Bettebghor et al. [3]. In that study both engine casing and pylon sizing optimization were simultaneously tackled in order to find a better mass distribution between engine and PPS structures, while achieving a feasible design. On the other hand, the engine mounts were considered in a fixed position so that only the engine casing thicknesses influenced directly tip clearances. In the present paper we consider the optimization of the PPS structure under a fixed engine architecture. The engine mounts are considered as part of the design domain. Structural optimization is the discipline that deals with finding the "best" structure of a product with respect to its weight and its structural behavior. Depending on the type of variables chosen to describe the solution, one can have size, shape, or topology optimization. In the latter, one seeks to achieve the design solution load path only knowing its design domain volume and its operating conditions. Since the pioneering work of Bendsøe and Kikuchi (1988) [4], topology optimization was developed and applied in several fields and physical problems. Nowadays there are many available topology optimization approaches, among which we can cite solid isotropic material with penalization (SIMP) [5], level set approaches [6], and evolutionary approaches [7]. In this work we will mainly focus on the SIMP approach. In [8] one can find an up-to-date review of the most promising applications of topology optimization to aerospace structures. Topology optimization has shown its advantages for the design of aircraft parts like the wing internal primary structure, wing box, and pylon [9,10]. In particular the engine pylon topology optimization was treated in [9] considering both structural compliance under several load cases and aerodynamic drag. In Xue et al. [10] an ant colony algorithm is deployed to optimize the front pylon mounts on the base of average stress under multiple load cases.

In this work we do not focus on pylon aerodynamic performance, as it was done in [9], since we fixed the design space shape. On the

*Ph.D. Candidate, Propulsion Airframe Engineering Stress Tranverse; also University of Toulouse; simone.coniglio@airbus.com.
†Associate Professor, Institut Clément Ader (ICA), CNRS, UPS, INSA, ISAE-SUPAERO, Mines-Albi, 3 rue Caroline Aigle; christian.gogu@gmail.com.
‡Stress Analysis Expert, Propulsion Airframe Engineering Stress Tranverse; remi.amargier@airbus.com.
§Professor, Institut Clément Ader (ICA), CNRS, ISAE-SUPAERO, UPS, INSA, Mines-Albi, 3 rue Caroline Aigle; joseph.morlier@isae-supaero.fr.

other hand, we include the pylon to engine interface inside the design zone. This gives the solution more freedom, which is also necessary to have an impact on both engine deformations and PPS primary structure mass. The design zone is much larger than the one considered in [10] and we did not employed ant colony algorithm to tackle the optimization problem.

In this context, the aim of this paper is to develop a topology optimization framework aimed at finding the optimal wing-engine attachment structure. Setting up such a framework is challenging for multiple reasons:

1) To consider a design zone that could both fit aerodynamic shapes and engine external shape, an irregular mesh needs to be considered for the design zone. To achieve mesh-independent solutions, filters are commonly employed. Nevertheless for such irregular meshes, a classic filter matrix can be computationally prohibitive. We propose a multigrid strategy to alleviate this computational issue.

2) The engine and design zone meshes may not be consistent at the interface. We propose to use rescaled localized radial basis function interpolation [11] approach to tie the two meshes.

3) The full engine plus design zone structural model may be very large and complex, and thus computationally expensive. Because the engine model does not change from one optimization iteration to another, we propose to statically condense it and use it as a super-element within the optimization framework.

4) To efficiently solve the optimization problem we need the gradients of the objective function and constraints of the optimization problem. We propose an adjoint-based gradient calculation for the responses specific to the considered optimization problem.

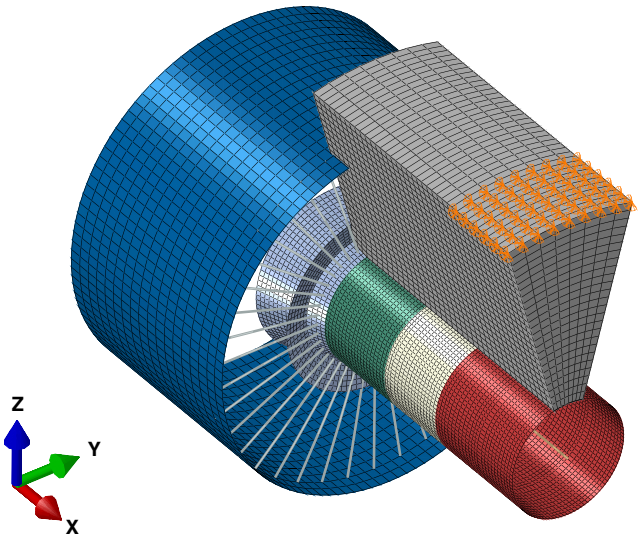


Fig. 1 Engine and design zone (in gray) finite element model. The engine model is made of 9976 finite elements, 9312 linear quadrilateral elements with reduced integration, and 664 linear beam elements. The solid design zone is clamped at the orange points.

The optimization formulation adopted for this work is a mass minimization with both stress and engine deformation constraints. The fact of including these constraints is beneficial for the total lead time in the design process. In this way, the solution provided in the topology optimization phase will need minor modifications in order to achieve a feasible design that should satisfy both stress and performance requirements. This comes, however, at increased computational cost as the gradients of these two new constraints have to be computed.

The rest of this paper is structured as follows: In Sec. II we introduce the structural finite element model (Sec. II.A) and performance model (Sec. II.B) associated with the simplified engine model used in this work. In Sec. II.C we present the formulation of the topology optimization problem. Section II.D reviews stress-constraint-related issues and describes the adopted strategy to solve them. In Sec. III we present in detail challenges 1–4 and the proposed approaches to solve them. The numerical results and discussions are presented in Sec. IV. Finally Sec. V provides concluding remarks.

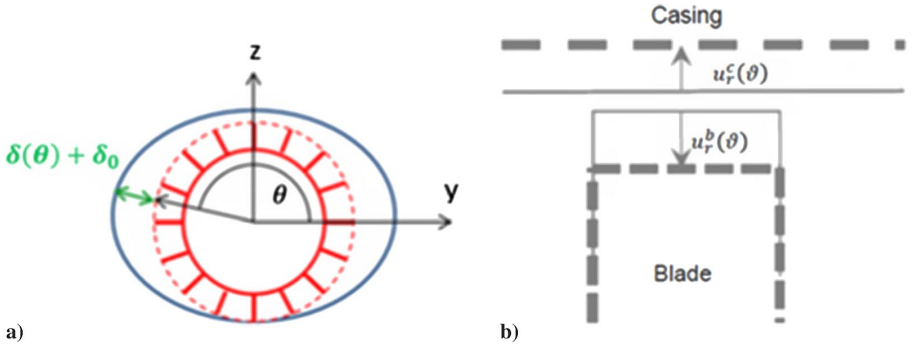
II. Formulation

A. Structural Model

The mechanical behavior of both engine and PPS structure is usually studied using a linear finite element model, integrating complex load combinations. In our study we introduce a general engine model that will validate our methodology. This model is a simplification of an industrial engine model that in general can have a fan casing structure, a core casing structure, and multiple shafts for rotors. Following assumptions are made regarding the model's mechanical behavior. The thermal growth and centrifugal growth of rotor blades are not considered in our analysis. Only aerodynamic loads linked with the aircraft maneuvers are considered. These loads are applied statically and we considered a representative axial thrust load case in this study. A general 3D solid design zone is integrated between the wing and the engine (see Fig. 1). This is the zone where the topology optimization will need to find the optimal material placement. For simplicity and because this does not have a significant effect on optimization, we did not consider the air inlet and the nacelle structure. The engine core casing and the design zone mesh are tied at some region that will enforce the solution to be linked in some specific areas to the engine. The connection between engine casing and design zone mesh will be detailed in Sec. III.B. Eight node brick elements were used to mesh the design zone, and a full eight-point integration was employed for elementary stiffness matrix and stress evaluation. The design zone model is clamped at the wing interface position. The axial load on the engine is applied with concentrated forces on the shaft nodes and with distributed loads on the engine casing. The connection between shaft node and engine casing nodes at bearing positions are modeled using rigid Kinematic couplings, that is, rigid connections. The commercial software Abaqus 13.2 was employed for mesh generation and load case application.

B. Performance Model

The performance function that we considered in this work is a simplification of the real TSFC variation induced by the considered



a) b)

Fig. 2 Rotor and stator displacements under maneuvers loads: a) z–y engine section diagram for a given rotor stage. The red structure stands for the rotor; the blue line for the deformed stator. b) Diagram illustrating radial displacement effect on the total tip clearance. The resulting tip clearance is the sum of the initial clearance δ_0 and its variation induced by the engine deformation calculated as $\delta(\theta) = u_r^c(\theta) - u_r^b(\theta)$.

load case. For our purpose, the TSFC is mainly dependent on the engine tip clearance variations. The tip clearance variation is described in Fig. 2, and defined in Eq. (1).

$$\delta(\theta) = u_r^c(\theta) - u_r^b(\theta) = \{r(\theta)\}^T \cdot (\{u^c(\theta)\} - \{u^b(\theta)\}) \quad (1)$$

where $u_r^c(\theta)$ and $u_r^b(\theta)$ are the radial displacement of casing and rotor blade tip, respectively, at the angular position θ, and $\{r(\theta)\}$ is the radial unit vector at the angle θ. Considering an angular discretization of the circumference given by the stator mesh in the sth engine stage, $\theta_{(s)}^i$, the tip clearance variation at the angular section i is then:

$$\{\delta_{(s)}\}_i = \{r(\theta_{(s)}^i\}^T \cdot (\{u^c(\theta_{(s)}^i\} - \{u^b(\theta_{(s)}^i\}) \quad (2)$$

That is actually a system of linear relationships that can be expressed in the matrix form:

$$\{\delta_{(s)}\} = [\gamma_{(s)}]\{U\} \quad (3)$$

where $\{U\}$ is the displacement vector. To characterize a stage's contribution to the overall engine performance, tip-clearance root-mean-squared (RMS) is evaluated at each stage:

$$R_{(s)} = \sqrt{\frac{\{\delta_{(s)}\}^T\{\delta_{(s)}\}}{N_{(s)}}} \quad (4)$$

where $N_{(s)}$ is the number of nodes used for the angular discretization of the sth stage. In this paper, we use a simple model to link the tip clearances to the TSFC performance. Note that this relationship is in general quite complex, but surrogate models can be constructed and used in such a context, similarly to the analytical model we use below. The variation in TSFC due to aircraft maneuvers is considered here as a linear combination of the RMS values per stage:

$$\Delta\text{TSFC}\% = \sum_{s=1}^{ns} \frac{R_s}{l_{(s)}} 100 \quad (5)$$

where ns is the number of stages of the engine and $l_{(s)}$ the stage blade height.

C. Topology Optimization Formulation

We use the classic SIMP material model [5] to describe the material layout in the topology optimization design zone. In this method, the solution is described by a density field between zero and one that locally gives the information on the presence or the absence of material in terms of both mass and Young's modulus. A solution with gray elements is always possible but is penalized by the power interpolation law used for the Young's modulus [5]. Let us introduce a pseudo density field $x(\{X_g\})$ in the design zone Ω. This field has a physical meaning of presence or absence of material so that one should take a value in $\{0, 1\}$, reading 0 as absence of material and 1 as presence of material. For $x(\{X_g\}) = 1$ one should use the value of local density and Young's modulus of the real material E. In the case $x(\{X_g\}) = 0$ the void can be simulated using a very soft material Young's modulus $E_{\min}$ that prevents stiffness matrix singularity and ill-conditioning. To use efficient gradient-based optimization algorithms, the problem is commonly relaxed so that $x(\{X_g\}) \in [0, 1]$. The physical interpretation of results presenting large regions characterized by intermediate density is not easy as they cannot be interpreted either as full material or void. For this reason, a simple penalization technique is commonly employed to prevent the optimization algorithm from converging on gray solutions. The Young's modulus as a function of x is written as:

$$E(x) = E_{\min} + (E - E_{\min})x^p \quad (6)$$

where the penalty value p penalizes the stiffness of intermediate densities. The value of p is usually set to 3 in order to get nearly black-and-white solutions. The finite element stiffness matrix assembly procedure will then use the local value of the Young's modulus in order to compute the stiffness matrix. In our framework we considered the Young's modulus as constant in each finite element so that the elementary stiffness matrix can be computed using the value of x in the element centroid x_{el} and the unit modulus elementary stiffness matrix $[K_{el}^{(1)}]$:

$$[K_{el}(x_{el})] = E(x_{el})[K_{el}^{(1)}] \quad (7)$$

The design zone stiffness matrix is assembled summing the contribution of each elementary matrix in a classic finite element (FE) style:

$$[K_{DZ}(\{x\})] = \bigoplus_{el=1}^{N_{el}} [K_{el}(x_{el})] \quad (8)$$

where $\oplus$ represents the assembly finite element operator and N_{el} is the number of elements in the design zone. The engine and the design zone are connected using an elimination approach that we develop in Sec. III.B. The final stiffness matrix of the entire model can then be directly assembled as:

$$[K(\{x\})] = [K_{DZ}(\{x\})] + [K_E] \quad (9)$$

From here on, we will refer to $[K(\{x\})]$ as the stiffness matrix after the application of boundary conditions and to $[K_E]$ as the contribution of the engine stiffness to $[K(\{x\})]$. In the same way one can make the assembly of the load vector:

$$\{F(\{x\})\} = \{F_{DZ}(\{x\})\} + \{F_E\} \quad (10)$$

where $\{F_{DZ}(\{x\})\}$ is the load coming from the design zone that can depend on the configuration[¶] and $\{F_E\}$ is the load vector applied on the engine model. The static balance equation will be written as:

$$[K(\{x\})]\{U(\{x\})\} = \{F(\{x\})\} \quad (11)$$

The so computed displacement vector $\{U(\{x\})\}$ is employed for the evaluation of design zone Von Mises stress (cf. Sec. II.D) and the TSFC variation (cf. Sec. II.B). The mass of the PPS structure is also an important parameter that we consider through the volume fraction defined as:

$$V(\{x\}) = \frac{\sum_{i=1}^{N_{el}} x_i|\Omega_i|}{\sum_{i=1}^{N_{el}} |\Omega_i|} = \frac{\{|\Omega|\}^T\{x\}}{\{|\Omega|\}^T\{\mathbf{1}\}} \quad (12)$$

where $|\Omega_i|$ is the volume of the ith element, $\{|\Omega|\}$ is the vector containing the volume of each element, and $\{\mathbf{1}\}$ is the vector having the same length of $\{x\}$ with 1 for each row. The value of $V(\{x\})$ is between 0 and 1 and gives the fraction of volume that is filled by active material. The final design should be as light as possible and should reduce the engine consumption variation induced by aircraft maneuvers. To get reasonable results the Von Mises stress in the design zone should also be lower than an allowable value. The full problem can thus be written as a nonlinear constrained optimization problem [cf. Eq. (13)], seeking to minimize the mass of the PPS structure, while imposing constraints on TSFC variations and maximum Von Mises stress.

$$\begin{cases} \min\limits_{\{x\}} V(\{x\}) \\ \text{s.t.} \\ 0 \le x_i \le 1 \qquad \forall\, i = 1,2,\ldots,N_{el} \\ G_T(\{U(\{x\})\}) = \frac{\Delta\text{TSFC}\%(\{U(\{x\})\}) - T_0}{T_0} \times 100 \le 0 \\ [K(\{x\})]\{U(\{x\})\} = \{F(\{x\})\} \\ (\sigma_{VM})_j(\{x\},\{U(\{x\})\}) \le \sigma_{\lim} \qquad \forall\, j | x_{i(j)} > 0 \end{cases} \quad (13)$$

where the $(\sigma_{VM})_j$ is the Von Mises stress computed in the jth Gauss point, $x_{i(j)}$ is the pseudo density of the ith element that contains the jth

[¶]This is the case of acceleration-induced load that depends on the mass of the solution and so on the configuration given by the design vector $\{x\}$.

quadrature point, σ_{lim} represents the allowable Von Mises stress in the design zone, and T_0 is an allowable overconsumption due to the maneuver load case. Classical compliance minimization formulation is cheaper than our proposed formulation, and can be adopted to get the inspiration for novel designs. Nevertheless the stiffest design will not always be able to respect both stress and engine consumption specifications. For this reason the formulation of Eq. (13), which also includes engine performance and stress constraints, will be considered for the rest of this study.

D. Stress Constraints in Topology Optimization

Directly including stress-based constraints in topology optimization has been an active field of research seeking to guarantee the design feasibility. Two major challenges arise in the implementation of stress constraints: the fact that the optimization problem presents singular optima [12], and the large number of stress constraints to be considered. The first issue consists in the fact that local optima belong to degenerate subspaces of the feasible domain that are not reachable using standard gradient-based optimizations. One way of dealing with this issue consists in relaxing the constraints (for instance, using ϵ-relaxation [13], or the q-p approach [14]) and then aggregating them using a regular approximation of the maximum function (for instance, the Kreisselmeier–Steinhauser function [15,16] or the p-norm [17]). Aggregation techniques have the main drawbacks of not having a precise control on the final design maximum stress. To control the exact value of maximum stress in the final solution Le et al. [18] proposed an adaptive approach that helps obtaining designs with the desired maximum stress. The stress constraint formulation presented in [19] is adopted in this work. We will first describe the formulation, and then we will develop adjoint gradient evaluation. The first step is to evaluate the microscopic stress tensor in each Gauss point of the design zone. To do that in the stiffness assembly phase the product of stress deformation matrix $[D]|_{x=1}$ and of the displacement-deformation matrix $[B]$ has to be saved in $[DB]$. This large sparse matrix reads in terms of input a displacement vector and provides as an output a stress vector containing six stress tensor components for each Gauss point of the design zone:

$$\{\bar{\sigma}(\{U(\{x\})\})\} = [DB]\{U(\{x\}\} \tag{14}$$

The Von Mises stress in 3D is then:

$$\sigma_{VM} = \sqrt{\sigma_1(\sigma_1-\sigma_2)+\sigma_2(\sigma_2-\sigma_3)+\sigma_3(\sigma_3-\sigma_1)+3(\sigma_{12}^2+\sigma_{23}^2+\sigma_{31}^2)} \tag{15}$$

The local relaxed stress constraint violation is:

$$\bar{g}_i = x_i\left(\frac{(\sigma_{VM})_i}{\sigma_{\text{lim}}} - 1\right) \tag{16}$$

where σ_{lim} is the allowable stress that depends on the material properties and on the desired safety factor, and x_i indicates the density relative to the element to which the ith Gauss point belongs. The aggregated stress is finally evaluated by the use of the lower bound Kreisselmeier–Steinhauser function [15] that approximates the local relaxed stress constraint violation maximum:

$$G_{\text{KS}}^l = \frac{1}{P}\ell_n\left(\frac{1}{N_G}\sum_{i=1}^{N_G} e^{P\bar{g}_i}\right) \tag{17}$$

To avoid numerical issues, the formulation of Eq. (17) is rewritten as:

$$G_{\text{KS}}^l = \bar{g}_{\max} + \frac{1}{P}\ell_n\left(\sum_{i=1}^{N_G} e^{P(\bar{g}_i-\bar{g}_{\max})}\right) - \frac{\ell_n(N_G)}{P} \tag{18}$$

where $g_{\max} = \max_i g_i$. We provide in Appendix A some useful properties of the G_{KS}^l function.

The satisfaction of stress constraints is thus imposed in problem (13) as:

$$\begin{cases} \min\limits_{\{x\}} V(\{x\}) \\ \text{s.t.} \\ 0 \le x_i \le 1 \qquad \forall\, i = 1,2,\ldots,N_{el} \\ G_T(\{U(\{x\})\}) = \frac{\Delta\text{TSFC}\%(\{U(\{x\})\})-T_0}{T_0}\times 100 \le 0 \\ [K(\{x\})]\{U(\{x\})\} = \{F(\{x\})\} \\ G_{\text{KS}}^l(\{x\},\{U(\{x\})\}) \le 0 \end{cases} \tag{19}$$

III. Challenges and Methodology

This section will describe various challenges related to the resolution of the optimization formulation of Eq. (19), as well as the corresponding approaches proposed to solve these challenges. First in Sec. III.A we will describe the adopted multigrid strategy aimed at reducing the computational effort needed for the evaluation of the filter matrix in nonuniform meshes. The mesh tying between the engine casing and the solid design zone is then described in Sec. III.B. In Sec. III.C we describe the static condensation approach leading to the super-element-based framework. Such a framework has main advantages of reducing the number of DOFs to be kept in the structural model used in topology optimization. Finally details about the adjoint computation of both lower bound Kreisselmeier–Steinhauser function and consumption variation gradients are given.

A. Density Filter for Nonuniform Mesh

Some typical issues associated with topology optimization are solution mesh dependency, checkerboard patterns, and nonuniqueness of the solution. Typically mesh dependency and checkerboard-related issues are solved using mesh-independent filtering techniques. These techniques can be easily implemented for uniform structured meshes as is the case in [20] and the majority of SIMP-based topology optimization studies. The fact of having 1×1 square element can be used to have a straightforward relationship between the element indexing and their neighbors' center-to-center distances. This is not the case for nonuniform unstructured meshes, as the ones considered here. In Talischi et al. [21] the difficulties induced by those cases are treated for 2D analysis with unstructured polygonal meshes. In particular it is shown that, to be more efficient, stiffness element matrices and filter matrix can be assembled once for all before the optimization loop. In this study, eight-node 3D finite elements with tri-linear shape functions were employed and eight Gauss points per element were considered for stiffness matrix assembly. On the other hand, only one Gauss point was used in order to evaluate the filtering convolution integral. As a consequence, following the same implementation of [21] the filter matrix $[H]$ needs the evaluation of all distances d_{ij} between each couple of element's centroid:

$$H_{i,j} = \frac{\max(0,|\Omega_j|(1-(d_{ij}/r))}{\sum_{j=1}^{N_{el}}\max(0,|\Omega_j|(1-(d_{ij}/r))} \tag{20}$$

where N_{el} is the element number in design zone, r is filter radius, and $|\Omega_j|$ is the volume of jth element. The cost for this evaluation in terms of memory and CPU time grows with $N_{el}^2 - N_{el}/2$. In [21] it is argued that this operation even if expensive should be done once for all before the optimization loop and should not be a bottleneck for overall analysis. Nevertheless, when increasing the number of elements, this simple operation can encounter memory limits faster than stiffness matrix inversion. These limitations were also studied and tackled in the PDE filter proposed by Lazarov and Sigmund [22], where instead of explicitly computing the filter matrix $[H]$, the filtered field is found as the solution of a PDE problem. This solution

involves a small memory cost, even if it requires the resolution of a system of equation with the size the number of nodes twice per iteration. These resolutions are not very expensive for small problems. In fact they require a small effort compared with displacements evaluation. In this work we propose an alternative way of directly computing $[H]$ that reduces the time needed for the computation of the filtering matrix, but still requires enough memory for the storage of $[H]$. Note that for reasonably small values of the filtering radius and for unstructured refined mesh, this proposed procedure appears advantageous. On the other hand, for large values of the filtering radius and for the same kind of meshes, the approach based on the PDE filter by Lazarov and Sigmund [22] is to be preferred to reduce memory requirements. It must be noted that $H_{i,j}$ is sparse because all distances superior to r do not contribute to $H_{i,j}$. We suppose to have a first mesh like the one in Fig. 3a on which we are able to evaluate all the distances between each element d_{ij} and compare them to r. We also suppose that a refinement of this mesh can be obtained cutting each element into eight as shown in Fig. 4a.

Let us consider two coarser mesh elements and their partition as considered in Fig. 4b whose centroid distance is known $\|\boldsymbol{AC}\|$. The minimal distance between centroids of the corresponding finer mesh elements, $\|\boldsymbol{BD}\|$, can be related to $\|\boldsymbol{AC}\|$ as:

$$\|\boldsymbol{AC}\| \leq \|\boldsymbol{AB}\| + \|\boldsymbol{BD}\| + |\boldsymbol{CD}\| \tag{21}$$

So that:

$$\|\boldsymbol{BD}\| \geq \|\boldsymbol{AC}\| - \|\boldsymbol{AB}\| - |\boldsymbol{CD}\| \tag{22}$$

The distances $\|\boldsymbol{AB}\|$ and $|\boldsymbol{CD}\|$ are also bounded by the radius of the smallest sphere circumscribed around the biggest coarser mesh element r_c. Therefore Eq. (22) becomes:

$$\|\boldsymbol{BD}\| \geq \|\boldsymbol{AC}\| - 2r_c \tag{23}$$

Note that even if the scheme of Fig. 4b considers cubic elements, Eq. (23) is valid for general eight-node brick elements. If the distance between coarse mesh centroids are known, it is possible to set up a test on these distances that can help to reduce the number of centroid-to-centroid distances that have to be computed for the finer mesh. In fact for Eq. (23):

$$\|\boldsymbol{AC}\| \geq 2r_c + r \Rightarrow \|\boldsymbol{BD}\| \geq r \tag{24}$$

Then the distance that for sure needs not to be computed in the finer mesh are the one between elements obtained from coarser mesh at a distance greater then $2r_c + r$. On the other hand, we cannot conclude that each and every distance that one can evaluate in this way will be smaller than r. The final cost of $[H]$ is then equal to $28N_c + 64N_c^*$, where N_c is the number of elements in the coarse mesh and N_c^* is the number of coarse mesh element pairs at a distance less or equal to $2r_c + r$. Since $N_c = N_{el}/8$ and for reasonably small r, $N_c^* = KN_c \ll (N_c)^2$, the cost for this procedure grows up with

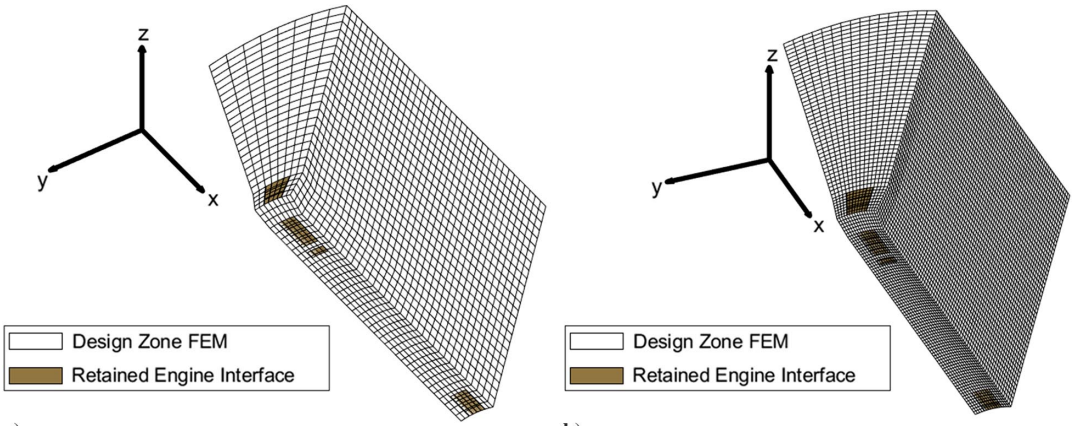


Fig. 3 Mesh refinement procedure needed for multigrid approach to the evaluation of $[H]$; the engine reduced element set is colored in yellow. Each node of these elements is kept in the engine super-element. a) Original design zone mesh, generated in Abaqus and impoted on Matlab by input file parsing. The mesh counts 7600 eight-node linear 3D solid finite elements and 9126 nodes. b) Design zone mesh after refinement. Each element of the original element is cut into 8 new elements, which give a total of $7600 \times 8 = 60{,}800$ eight-node linear 3D solid finite elements and 66,759 nodes.

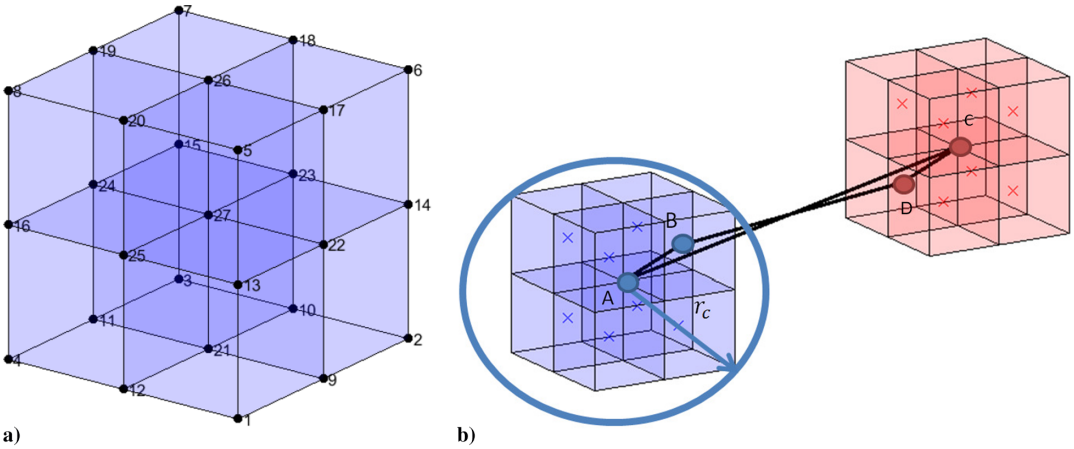


Fig. 4 Relation between coarse and fine mesh. a) 3D partition employed for mesh refinement; nodes 1–8 belong to original coarse mesh element. Partition determines node 9–27 and 8 elements of the finer mesh. b) Scheme of two elements of the original mesh after partition. The relationship between the minimal distance between two finite element centroids in the finer mesh $\|\boldsymbol{BD}\|$, the corresponding distance between the centroids of the coarse mesh $\|\boldsymbol{AC}\|$, and the radius of the sphere circumscribed around the biggest coarse mesh finite element r_c can be determined considering the vector chain $\boldsymbol{AC} = \boldsymbol{AB} + \boldsymbol{BD} + \boldsymbol{DC}$.

$(7/4 + 8K)N_{el}$, linearly and not quadratically with the problem size N_{el}. One can note that this procedure is also suitable for parallel implementations, thus further decreasing its numerical cost.

B. Mesh Projection

Another important issue that needs to be addressed is the mesh inconsistency between engine and design zone. In fact the engine finite element model in the current practice is built by the engine manufacturer and is not initially intended for use in a topology optimization framework. For this reason in order to freely mesh the design zone we would like to be able to put these two nonoverlapping domains in connection trough their interface DOFs.

The first hypothesis made is that only translational DOFs of engine shell elements have to be considered for kinematic tying. For this reason the engine condensation was performed around the retained nodes translational DOFs and not for rotational DOFs, which are therefore not constrained. The external skin of the engine's shell elements needs to have the same displacements as the external surface of the solid elements. Many techniques can be employed for dealing with this problem as described in the review [23]. The most popular approach in the literature in order to solve this problem is the Mortar method [24], which is well known to have very good performance in terms of displacement optimality convergence; however, its implementation is often quite complex. Displacement continuity has been imposed in this work considering the radial basis function interpolation approach proposed by the Deparis et al. [11] based on a B&W compactly supported radial basis function. This simple collocation approach makes the hypothesis that the displacement field of one discretized surface (commonly called master surface) can be used to evaluate, by interpolation, the displacement field of the other (so-called slave surface). Writing kinematic continuity in this way, one can have a set of linear relationships between meshes in the form of:

$$\{u_c\} = [\Pi_{cd}]\{u_d\}, \qquad \{u_d\} = [\Pi_{dc}]\{u_c\} \tag{25}$$

where $\{u_c\}$ and $\{u_d\}$ are, respectively, engine and design zone interface DOFs. These relationships are used to eliminate slave DOFs from the problem because they can be evaluated by interpolation from master DOFs. The choice of master and slave surfaces in structural finite element analysis is crucial. Commonly, in order to have better stress and displacement accuracy the surface discretized with the finest mesh should be considered as slave and the other as master. In our problem engine retained interface was considered as master and design zone interface as slave. The final set of balance equations for the assembled system is obtained imposing residual and energy balance at the interface. For instance, after eliminating engine DOFs, the final system of equation reads:

$$\begin{bmatrix} [K_{oo}] & [K_{od}] \\ [K_{do}] & [K_{dd}] + [\Pi_{cd}]^T[\tilde{K}_{cc}][\Pi_{cd}] \end{bmatrix} \begin{Bmatrix} \{u_o\} \\ \{u_d\} \end{Bmatrix} = \begin{Bmatrix} \{F_o\} \\ \{F_d\} + [\Pi_{cd}]^T\{\tilde{F}_c\} \end{Bmatrix} \tag{26}$$

where o is the index of design zone DOFs not lying on the interface. Using these techniques the engine-retained nodes stiffness matrix and load vector can be integrated to the design zone.

C. Super-Element Exploitation

The fact that an industrial engine model has to be integrated in the finite element analysis poses a real problem in terms of both implementation efficiency and development time because the combined engine and design zone structural model may be too complex and computationally expensive to be solved at each iteration of the optimization process. To circumvent these issues we will use super elements, which are a very efficient solution to deal simply with very complex models that do not change in the optimization loop [25]. Here we develop this method on a general structure shown in Fig. 5. Sorting the structure degrees of freedom into retained (c) and other degree of freedoms (E), the static balance equation can then be written in the following form:

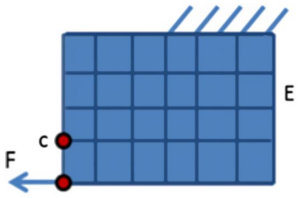


Fig. 5 Example of DOF partition: the nodes of the structure are sorted in retained nodes [whose DOFs are (c)] and other nodes [whose DOFs are (E)]. After the static condensation, a super-element containing only (c) DOFs will be generated. Nevertheless the suppressed DOFs (E) can still be computed after static analysis thanks to Eq. (28).

$$\begin{bmatrix} [K_{cc}] & [K_{cE}] \\ [K_{Ec}] & [K_{EE}] \end{bmatrix} \begin{Bmatrix} \{u_c\} \\ \{u_E\} \end{Bmatrix} = \begin{Bmatrix} \{F_c\} \\ \{F_E\} \end{Bmatrix} \tag{27}$$

Using the second-line block of equations and solving for $\{u_E\}$

$$\{u_E\} = [P]\{u_c\} + \{u_0\} \tag{28}$$

With

$$[P] = -[K_{cE}][K_{EE}]^{-1} \tag{29}$$

$$\{u_0\} = [K_{EE}]^{-1}\{F_E\} \tag{30}$$

The $[P]$ matrix is called constrained modal matrix in Nastran and recovery matrix in Abaqus. $\{u_0\}$ is called fixed interface displacement on Nastran and has to be computed in a separated analysis on Abaqus. Substituting Eq. (28) back in the first line block of Eq. (27):

$$[\tilde{K}_{cc}]\{u_c\} = \{\tilde{F}_c\} \tag{31}$$

With

$$[\tilde{K}_{cc}] = [K_{cc}] + [P][K_{cE}] \tag{32}$$

$$\{\tilde{F}_c\} = \{F_c\} + [P]\{F_E\} \tag{33}$$

In our problem we can make the evaluation of $[\tilde{K}_{cc}]$, $\{\tilde{F}_c\}$, and $[P]$ using a commercial software like Abaqus using the engine model. The $\{u_0\}$ vector can also be evaluated using a linear perturbation static analysis of the engine fixing the DOFs of the interface. If the structure has to be integrated into another model, the retained DOFs (c) can be used to describe the displacement of the whole assembly. The stiffness matrix of Eq. (27) does not need to be inverted anymore; the only knowledge of $[\tilde{K}_{cc}]$ and $\{\tilde{F}_c\}$ is sufficient. Considering the engine interface DOFs as retained DOFs (c), we can make possible the evaluation of engine and design zone assembly displacements just considering the retained DOFs stiffness matrix and load vector. This implies a significant economy in CPU time especially for big and complex engine models, because the full engine model does not have to be evaluated at each change of the design zone considered by the topology optimization. Furthermore, this also implies that the software used for evaluating the design model can be separate from the one for evaluating the design zone model and carrying out the topology optimization. This is important because in practice the engine model is usually constructed in commercial FE software. In this work we used Abaqus for the engine model and Matlab for the design zone model and topology optimization. Accordingly reading Abaqus *.dat* and *.mtx* files from Matlab environment, we can make

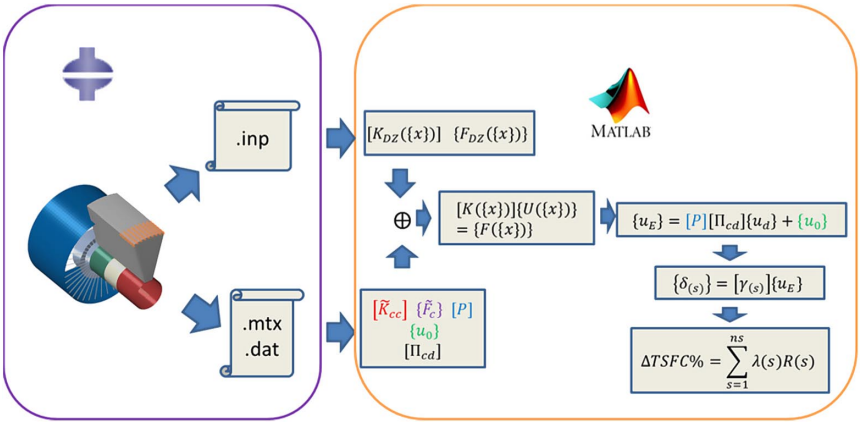


Fig. 6 Tip-clearance evaluation workflow. Abaqus is used to generate both the engine and the design zone. A parsing of the input file is made on Matlab to import the design zone mesh coordinates and finite elements. The engine model is reduced to a super-element on the interface using Abaqus, and then a parsing is made on the *.mtx* and the *.dat* files coming, respectively, from the substructuring and the linear load case. The stiffness matrix of the design zone is assembled together with the engine super-element stiffness matrix; the same is done for the load vectors. For each iteration of the topology optimization the assembled problem is solved to determine the design zone displacements and the recovered engine displacements. Finally tip clearance variations are computed in each stage and employed to compute consumption variation.

the evaluation of the whole structure just using the Matlab environment as it is summarized in Fig. 6.

This gives significant flexibility in implementing various approaches developed for topology optimization. Here we want to recall that super-element exploitation does not affect model accuracy as is the case of substructuring.

D. Gradient Adjoint Evaluation

To solve the nonlinear contained optimization problem we chose to use Svanberg's MMA optimization algorithm [26,27]. This optimization solver takes as input all responses values and sensitivities with respect to design variables, lower bounds, and upper bounds, and gives as output the vector of the new optimal candidate. To do so, a first-order local convex approximation of the original optimization problem is solved at each optimization iteration. To efficiently compute these gradients we propose here adjoint-based evaluations. Let us start by considering a generic response $O(\{x\},\{U(\{x\})\})$ that depends directly through the displacement vector on the design variables.[**]

$$\left\{\frac{dO}{dx}\right\} = \left\{\frac{\partial O}{\partial x}\right\} + \left[\frac{dU}{dx}\right]\left\{\frac{\partial O}{\partial U}\right\} \tag{34}$$

By taking the derivatives of equations (11) by dx_i one gets:

$$\frac{d[K(\{x\})]}{dx_i}\{U(\{x\})\} + [K(\{x\})]\frac{d\{U(\{x\})\}}{dx_i} = \frac{d\{F(\{x\})\}}{dx_i} \tag{35}$$

Writing the product $(d[K(\{x\})]/dx_i)\{U(\{x\})\}$ as columns of the matrix $[(dK/dx)U]$ and $\{F(\{x\})\}/dx_i$ as the columns of the matrix $[dF/dx]$ one can also write[††]:

$$\left[\frac{dU}{dx}\right]^T = [K]^{-1}\left(-\left[\frac{dK}{dx}U\right] + \left[\frac{dF}{dx}\right]\right) \tag{36}$$

Defining the adjoint vector as:

$$\{\beta\} = [K]^{-1}\left\{\frac{\partial O}{\partial U}\right\} \tag{37}$$

One can finally rewrite Eq. (34):

$$\left\{\frac{dO}{dx}\right\} = \left\{\frac{\partial O}{\partial x}\right\} + \left(-\left[\frac{dK}{dx}U\right]^T + \left[\frac{dF}{dx}\right]^T\right)\{\beta\} \tag{38}$$

It is straightforward to use Eqs. (37) and (38) to compute $\{dG_T/dx\}$, $\{dV/dx\}$, and $\{dG^l_{\mathrm{KS}}/dx\}$. Let us start by the evaluation of the TSFC variation constraints G_T. Since ΔTSFC% has no direct dependency on the $\{x\}$ we have:

$$\left\{\frac{\partial G_T}{\partial x}\right\} = \{0\} \tag{39}$$

By the use of Eqs. (3–5):

$$\left\{\frac{\partial G_T}{\partial U}\right\} = \frac{100}{T_0}\left\{\frac{\partial \Delta\mathrm{TSFC}\%}{\partial U}\right\} = \frac{100}{T_0}\sum_{s=1}^{ns}\lambda_{(s)}\left\{\frac{\partial R_{(s)}}{\partial U}\right\}$$
$$= \frac{100}{T_0}\sum_{s=1}^{ns}\frac{\lambda_{(s)}}{N_{(s)}R_{(s)}}[\gamma_{(s)}]^T\delta_{(s)} \tag{40}$$

Similarly, since V is only dependent on x:

$$\left\{\frac{\partial V}{\partial U}\right\} = \{0\} \tag{41}$$

By the use of Eq. (12):

$$\left\{\frac{\partial V}{\partial x}\right\} = \frac{\{|\Omega|\}}{\{|\Omega|\}^T\{\mathbf{1}\}} \tag{42}$$

For the lower bound Kreisselmeier–Steinhauser function sensitivities we have both dependency on $\{x\}$ and $\{U(\{x\})\}$:

$$\left\{\frac{\partial G^l_{\mathrm{KS}}}{\partial x}\right\} = \frac{\sum_{i=1}^{N_G}\{\partial \bar{g}_i/\partial x\}e^{P\bar{g}_i}}{\sum_{i=1}^{N_G}e^{P\bar{g}_i}} \tag{43}$$

With $\{\partial \bar{g}_i/\partial x\}$ defined as:

$$\left\{\frac{\partial \bar{g}_i}{\partial x}\right\}_j = \begin{cases}\frac{(\sigma_{VM})_i}{\sigma_{\lim}} - 1 & i \in \boldsymbol{G}_j \\ 0 & i \notin \boldsymbol{G}_j\end{cases} \tag{44}$$

$\boldsymbol{G}_j$ referred to the jth element Gauss point index. In the same way one can evaluate:

**In this section all sensitivities are computed with respect to physical densities. Filtering should be employed to get sensitivities with respect to design variables.

††Hereafter the dependency on $\{x\}$ and $\{U(\{x\})\}$ is neglected for conciseness.

Table 1 Summary table of sensitivity terms needed for Eqs. (37) and (38) to compute $\{dO/dx\}$

O	$\{\partial O/\partial U\}$	$\{\partial O/\partial x\}$
V	$\{0\}$	$\{\|\Omega\|\}/\{\|\Omega\|\}^T\{\mathbf{1}\}$
G_T	$(100/T_0)\sum_{s=1}^{ns}(\lambda_{(s)}/N_{(s)}R_{(s)})[\gamma_{(s)}]^T\delta_{(s)}$	$\{0\}$
G^l_{KS}	$(1/\sum_{i=1}^{N_G} e^{P\tilde{g}_i})[DB]^T\{\tilde{S}\}$	$\sum_{i=1}^{N_G}\{\partial\tilde{g}_i/\partial x\}e^{P\tilde{g}_i}/\sum_{i=1}^{N_G} e^{P\tilde{g}_i}$

$$\left\{\frac{\partial G^l_{\mathrm{KS}}}{\partial U}\right\} = \frac{\sum_{i=1}^{N_G}\{\partial\tilde{g}_i/\partial U\}e^{P\tilde{g}_i}}{\sum_{i=1}^{N_G} e^{P\tilde{g}_i}} = \frac{\sum_{i=1}^{N_G}\{\partial(\sigma_{VM})_i/\partial U\}(x_i/\sigma_{\mathrm{lim}})e^{P\tilde{g}_i}}{\sum_{i=1}^{N_G} e^{P\tilde{g}_i}} \tag{45}$$

From Von Mises stress definition (15) in each Gauss point i we can write[‡‡]:

$$\left\{\frac{\partial(\sigma_{VM})}{\partial U}\right\} = \frac{1}{2(\sigma_{VM})}\left((2\sigma_1-\sigma_2-\sigma_3)\left\{\frac{\partial\sigma_1}{\partial U}\right\} + (2\sigma_2-\sigma_3-\sigma_1)\left\{\frac{\partial\sigma_2}{\partial U}\right\} + (2\sigma_3-\sigma_1-\sigma_2)\left\{\frac{\partial\sigma_3}{\partial U}\right\} + 6\left(\sigma_{12}\left\{\frac{\partial\sigma_{12}}{\partial U}\right\} + \sigma_{13}\left\{\frac{\partial\sigma_{13}}{\partial U}\right\} + \sigma_{23}\left\{\frac{\partial\sigma_{23}}{\partial U}\right\}\right)\right) \tag{46}$$

In this equation, each vector, $\{\partial\sigma_1/\partial U\}$,$\{\partial\sigma_2/\partial U\}$,$\{\partial\sigma_3/\partial U\}$, $\{\partial\sigma_{12}/\partial U\}$,$\{\partial\sigma_{13}/\partial U\}$,$\{\partial\sigma_{23}/\partial U\}$ corresponds for Eq. (14) to a column of $[DB]^T$. For this reason Eq. (45) can be written as:

$$\left\{\frac{\partial G^l_{\mathrm{KS}}}{\partial U}\right\} = \frac{1}{\sum_{i=1}^{N_G} e^{P\tilde{g}_i}}[DB]^T\{\tilde{S}\} \tag{47}$$

where:

$$\{\tilde{S}\} = \begin{Bmatrix}\{\tilde{S}\}_1\\ \{\tilde{S}\}_2\\ :\\ \{\tilde{S}\}_{N_G}\end{Bmatrix},$$

$$\{\tilde{S}\}_i = \frac{x_i e^{P\tilde{g}_i}}{2\sigma_{\mathrm{lim}}(\sigma_{VM})_i}\begin{Bmatrix}2(\sigma_1)_i-(\sigma_2)_i-(\sigma_3)_i\\ 2(\sigma_2)_i-(\sigma_3)_i-(\sigma_1)_i\\ 2(\sigma_3)_i-(\sigma_1)_i-(\sigma_2)_i\\ 6(\sigma_{12})_i\\ 6(\sigma_{13})_i\\ 6(\sigma_{23})_i\end{Bmatrix} \tag{48}$$

The results of the sensitivity computations are summarized in Table 1.

To use Eq. (35) the link between the notation used for Eqs. (19) and (26) should be specified. We made a partition of vector $\{U\}$ into "o," design zone DOFs not lying on the interface, and "d," design zone DOFs introduced with Eq. (26):

$$\{U\} = \begin{Bmatrix}\{u_o\}\\ \{u_d\}\end{Bmatrix} \tag{49}$$

According to Eqs. (26) and (9–11):

$$[K] = \begin{bmatrix}[K_{oo}] & [K_{od}]\\ [K_{do}] & [K_{dd}]\end{bmatrix} + \begin{bmatrix}[0_{oo}] & [0_{od}]\\ [0_{do}] & [\Pi_{cd}]^T[\tilde{K}_{cc}][\Pi_{cd}]\end{bmatrix} = [K_{DZ}] + [K_E] \tag{50}$$

and

$$\{F\} = \begin{Bmatrix}\{F_o\}\\ \{F_d\}\end{Bmatrix} + \begin{Bmatrix}\{0_o\}\\ \{[\Pi_{cd}]^T\{\tilde{F}_c\}\}\end{Bmatrix} = \{F_{DZ}\} + \{F_E\} \tag{51}$$

By the use of Eqs. (6–8):

$$\frac{d[K]}{dx_i} = \frac{d[K_{DZ}]}{dx_i} = \bigoplus_{el=i} p(E_{\max}-E_{\min})x_i^{p-1}[K^{(1)}_{el}] \tag{52}$$

In the present work we only consider axial loading, and so $[dF/dx] = 0$. For a general uniform acceleration (e.g., inertial loads) the sensitivities of the load vector could be computed using:

$$\{F_{DZ}\}_i = \int \rho f_i \Phi_i \, \mathrm{d}V = \frac{1}{8}\sum_{j=1}^{m_i} x_j f_i \Omega_j \rho \tag{53}$$

So that:

$$\{F_{DZ}\} = \frac{1}{8}[\rho f \Omega]\{x\} \tag{54}$$

Then:

$$\left[\frac{dF}{dx}\right] = \left[\frac{dF_{DZ}}{dx}\right] = \frac{1}{8}[\rho f \Omega] \tag{55}$$

The adjoint evaluation of sensitivities presented in this subsection needs the use of Eq. (34) twice, once for the evaluation of $\{dG_T/dx\}$ and once for $\{dG^l_{\mathrm{KS}}/dx\}$. Therefore the stiffness matrix has to be inverted for three different right-hand side vectors per each optimization loop iteration. Taking the formulation described here in matrix notation also facilitates implementation in the presented Matlab framework, taking advantage from vectorization.

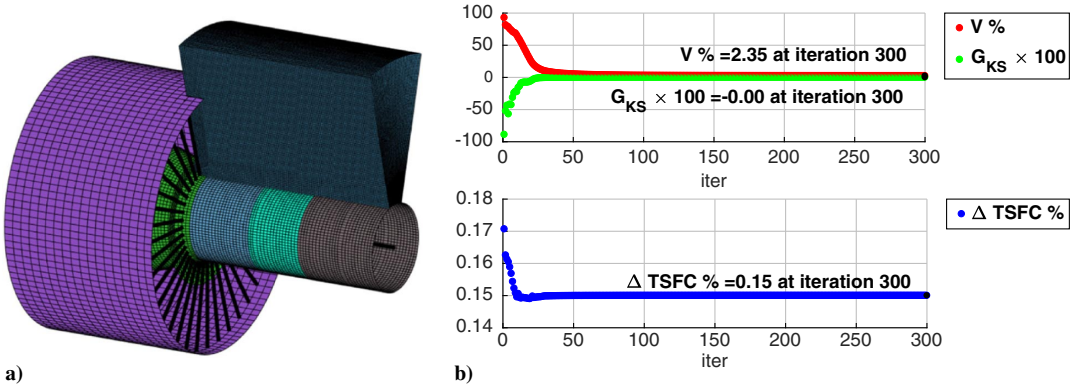
IV. Numerical Results

The optimization framework developed was applied on a problem involving the engine model described in Sec. III.A and on the design mesh represented in Fig. 7a. In Table 2 one can find the optimization setup details.[§§]

The mesh refinement procedure introduced in Sec. IV.A was applied twice to the original mesh imported from Abaqus cf. Fig. 3a. The final stiffness matrix has 1.5 million DOFs before applying boundary conditions. The filtering radius was taken as two times the average element size that is corresponding to twice the average size of the original mesh. The SIMP penalty value was set to 3 and the stress constraint aggregation constant to 4. This is a relatively small value for P that improves optimization convergence, which requires the use

[‡‡]We did not indicate the Gauss point index i for conciseness.

[§§]MMA maximum asymptote distance from the current point value has not a particular name in the mmasub Matlab function provided by Svanberg [27]. It can be found in the lines where lowmin and uppmax variables are computed as the coefficient that multiplies the variable range. When this value is reduced, the algorithm behaves more conservatively when approximating the real functions overestimating their convexity.



a) b)

Fig. 7 a) Design zone mesh used for final topology optimization problem. Each element of the original element is cut into 64 new elements, which give a total of 7600 × 64 = 486,400 eight-node linear 3D solid finite elements and 509,949 nodes. b) Convergence history of ΔTSFC%, V%, $G^l_{\mathrm{KS}} \times 100$ after 300 design iterations.

of a scaling factor on the stress allowable. In fact for Eq. (59), we can conclude that $\bar{g}_{\max} - G^l_{\mathrm{KS}} < (\ell_n(N_G)/P) = (\ell_n(3891200)/4) \approx 3.7936$, so that even if $G^l_{\mathrm{KS}} \leq 0$ this will only imply that $g_{\max} < 3.7936$. Assuming that the maximum relaxed stress constraint violation is on a material with $x_i = 1$ this means that the actual maximum Von Mises stress allowable is $\sigma_{\max} < (1 + 3.7936) \times \sigma_{\lim} = \sigma_{alw} = 47.9$ MPa, that is, the value of stress that we do not want to attain even locally. We then set $\sigma_{\lim} = (\sigma_{\mathrm{alw}}/1 + 3.7936) \approx 10$ MPa. The initial design consists of $x_i = 1$, $\forall\ i = 1,2,\ldots,N_{el}$; the allowed TSFC variation was 0.15%, which is a 12% improvement from the initial design. Stress constraint nonlinearities can be a source of MMA convergence difficulties and sometimes divergence. To tackle this problem we propose to set a smaller value of the MMA external move limit (here considered as the maximum difference between the asymptotes distance from the configuration point). This imposes MMA to produce conservative local approximations of the original optimization problem that are less prone to violate optimization constraints. The convergence history of volume fraction, ΔTSFC%, and of G^l_{KS} is presented in Fig. 7b, and the final design configuration is presented in Fig. 8a. Note that just considering Fig. 7b one could conclude that convergence was achieved approximately after 50 iterations. This is because the stress constraints are very nonlinear so that MMA needs to keep the optimization step very small in order to avoid stress constraint violation. Stopping the optimization after 50 iteration would lead to a nonconverged design full of gray elements. In the same way considering the design variable variation as stopping criterion could lead again to gray solutions. The optimization was therefor stopped after 300 design iterations. A KKT norm condition of 0.001 was also considered but was not achieved in the maximum number of iterations. The final design is well connected and respects constraints. We can find two main load paths, one at the front of the engine and a second at the rear. Moreover engine casing reinforcement structures can be found at the front of the solution to avoid tip clearance variations. To make displacement and stress plots in Figs. 8c and 8d the solution was thresholded, that is:

$$\bar{x}\mathrm{Phys}_i = \begin{cases} 1 & \text{if } x\,\mathrm{Phys}_i \geq t_{sh} \\ 0 & \text{otherwise} \end{cases} \tag{56}$$

where xPhys are the physical densities and $t_{sh} = 0.22$, selected in order to form a well-connected solution. Doing so the final solution performance is deteriorated as summarized in Table 3.

The thresholded solution still respects Von Mises stress constraint; on the other hand, the allowable fuel consumption constraint is violated by 1.47% and the final volume fraction is increased by 34%. Even after this increase, the final volume fraction still represents a large improvement; the solution was thus considered as acceptable for the sake of this study.

Von Mises stress color maps, based on the average over Gauss points, are presented in Fig. 8c. As expected the final design has a final maximum stress that is greater than 10 MPa imposed through the G^l_{KS} function, but lower than 47.9 MPa as a consequence of the choice of P. The solutions displacement field 8d is consistent with boundary conditions and to the load applied to the structure. The final Von Mises stress is obviously not homogeneous within the solution. It reaches its maximum in the regions adjacent to the wing and at the interface with the engine model. Note that the high stresses at the attachment with the wing are induced by the geometry and loading. On the other hand, the high stresses at the interface with the engine model are numerical artifacts of the kinematic tying approach of the design zone and the engine super-elements. These latter could

Table 2 Optimization problem setup

Symbol	Name	Value
E_0	Young's modulus	210 GPa
ν	Poisson ratio	0.29
N_{el}	Number of elements in the design zone	486,400
N_{DOFs}	Number of rows of the stiffness matrix	1,529,847
N_G	Number of Gauss points in the design zone	3,891,200
r	Filtering radius	2× mesh average size
p	SIMP penalty	3
P	Aggregation constant	4
$\sigma_{\lim}$	Allowable stress to be used in 16	10 MPa
σ_{alw}	Maximum local allowable stress	47.9 MPa
T_0	Allowable consumption variation	0.15%
$\{\rho f\}$	Inertial load	$\{0\}$
	Stopping condition on the Karush-Kunt Tucker residual norm	$KKTn \leq 10^{-3}$
	Stopping condition on the iteration number	iter ≥ 300
	MMA external move limit for first 2 iterations	0.4
	MMA maximum asymptote distance from the current point	0.1

The hypothesis made to get numerical results is listed here.

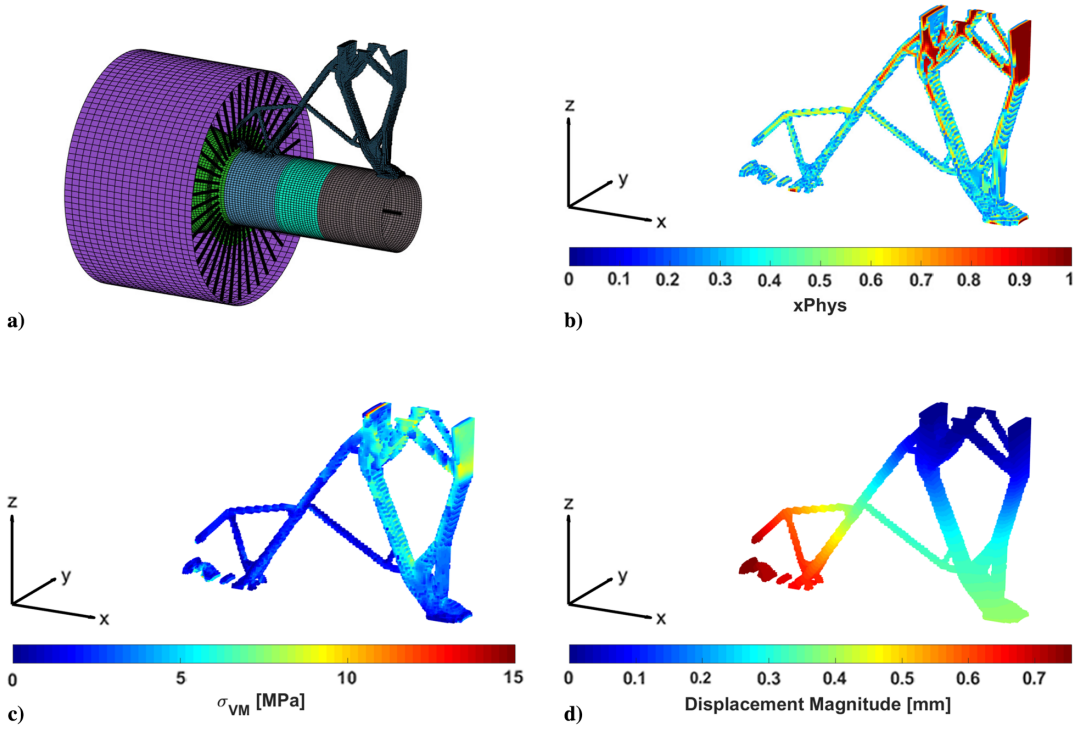


Fig. 8 Topology optimization results. a) Design configuration after 300 iterations; only densities greater than 0.22 are displayed. b) Physical density color map. c) Von Mises stress color map. d) Displacement magnitude color map.

potentially be reduced by a more complex kinematic tying approach such as the Mortar approach. The structure found by the optimization algorithm is nearly planar and contained in the x–z plane. This is due to the load considered here (axial load) and to the symmetry of the engine model. This solution can therefore be further improved considering multiple load cases, which will load the structure in different directions. Changing mesh tying approaches used to deal with the mesh inconsistency at the interface with the engine could also potentially improve the solution. The results found are not of direct practical interest due to several simplifying assumptions considered in the models being used (especially the simplified engine model). Nevertheless the solution is consistent with the model hypothesis and shows that it is possible to deal with both engine deformation and stress criteria in the same 3D topology optimization framework.

V. Conclusions

This work addressed the problem of optimization of the topology of the engine-wing attachment structure of a typical medium-range commercial passenger aircraft. The proposed framework for solving this problem had to overcome several challenges—notably, irregularity of the design zone mesh, nonconsistent interfaces between the design zone and the engine model, numerical costs related to the engine model developed in commercial finite element software, engine performance, and stress criteria to be included in the topology optimization. Accordingly, in the proposed framework stress-based and engine relative displacement constraints were successfully implemented in a mass-driven 3D topology optimization of pylon and engine mounts for a representative engine model. The engine model stiffness was integrated in the topology optimization using a super-element-based framework to reduce overall optimization time, whereas objective and stress constraint gradients were efficiently evaluated by the adjoint method. Considering mesh nonuniformity, super-element integration and mesh tying techniques make our framework general enough to consider various related structural and topology optimization problems, notably including design zone constraints due to external aerodynamic shapes or the presence of other components. The generalization of our study to multiple load cases and several other performance and manufacturing criteria represents a future line of work. A future line of work is also related to the hyperstatic interface between engine and design zone that has important consequences on the feasibility of the solution with respect to both engine installation and thermal expansion. It will also be interesting to investigate reducing the optimization overall time needed to pass from a topology optimization solution to a final design and increase the robustness of our design with respect to fail safe scenarios.

Table 3 Solution responses before and after thresholding

Response	Original solution	After thresholding ($t_{sh} = 0.22$)
ΔTSFC%	0.15	0.1522
V%	2.35	3.15
G^l_{KS}	-4.3×10^{-6}	-9×10^{-3}

Appendix A: KS Function Properties

In the following subsection we review essential properties of the Kreisselmeier–Steinhauser (KS) functions used for the stress constraints. Given that $P(g_i - g_{\max}) \le 0, \forall\, i$, $e^{P(g_i - g_{\max})} \le 1, \forall\, i$, and $(1/P)\,\ell_n\left(\sum_{i=1}^{N_G} e^{P(\bar{g}_i - \bar{g}_{\max})}\right) - (\ell_n(N_G)/P) \le 0$ so that one can conclude $G^l_{\mathrm{KS}} \le g_{\max}$ and knowing that $\sum_{i=1}^{N_G} e^{P(g_i - g_{\max})} > 1$ implies $(1/P)\,\ell_n\left(\sum_{i=1}^{N_G} e^{P(\bar{g}_i - \bar{g}_{\max})}\right) > 0$ so that:

$$\bar{g}_{\max} - G^l_{\mathrm{KS}} = \frac{\ell_n(N_G)}{P} - \frac{1}{P}\,\ell_n\left(\sum_{i=1}^{N_G} e^{P(\bar{g}_i - \bar{g}_{\max})}\right) < \frac{\ell_n(N_G)}{P} \tag{A1}$$

As a remarkable result we have:

$$G^l_{\mathrm{KS}} \le \bar{g}_{\max} < G^l_{\mathrm{KS}} + \frac{\ell_n(N_G)}{P} = G_{\mathrm{KS}} \tag{A2}$$

where G_{KS} is the Kreisselmeier–Steinhauser function. Equivalently:

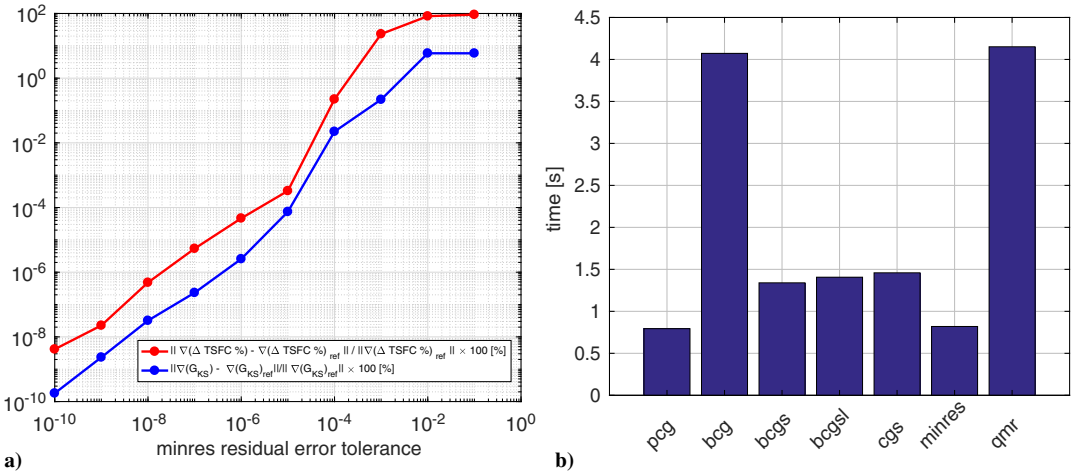


Fig. A1 a) *minres* residual error impact on stress and ΔTSFC constraints sensitivity. b) Displacement evaluation elapsed time [s] using several Matlab native functions.

$$0 \leq g_{\max} - G^l_{\mathrm{KS}} < \frac{\ell_n(N_G)}{P} \tag{A3}$$

Finally $\lim_{P\to\infty}(\ell_n(N_G)/P) = 0$, which for Eq. (A3) implies:

$$\lim_{P\to\infty} G^l_{\mathrm{KS}} = g_{\max} \tag{A4}$$

One can also limit the range of variation of $g_{\max}$ using an allowable tolerance:

$$|G^l_{\mathrm{KS}} - g_{\max}| < \epsilon \tag{A5}$$

According to Eq. (A1) one can chose a value of P as

$$P > \frac{\ell_n(N_G)}{\epsilon} \tag{A6}$$

One must also keep in mind that for greater value of P, the G^l_{KS} can have very nonlinear behavior. This has some negative consequences on gradient-based optimization solver. An opposite strategy to deal with stress nonlinearity was explored in Lian et al. [28]. In this paper they proposed to increase the number of Gauss points N_G. According to Eq. (A1) this strategy increases the distance from the true maximum function, hence reducing the stress aggregation function nonlinearities. When using lower bound KS function and p-mean, this difference is inconvenient because the final maximum stress will be greater than the one corresponding to KS aggregation, needing compensation on the stress limit (as suggested in this paper). On the other hand, using KS function or p-norm the design is always slightly oversized so that increasing the aggregation constant could still be beneficial to the final mass of the solution. A less elegant but still effective way of accounting for this gap is to change allowable stress during iterations in the way proposed in Le et al. [18].

Appendix B: Solver Benchmark

In Sec. III.D we showed that in order to compute objective and constraints sensitivity three inversions of stiffness matrices are needed per iteration (one for displacement evaluation and the other two for adjoint vector evaluation). When small problems have to be tackled, direct approaches are usually preferred to iterative approaches because of their robustness and accuracy. These methods work in three phases: 1) the stiffness matrix is reordered to reduce the decomposition computational burden, 2) the Cholesky decomposition is computed, and 3) the decomposition is used to compute linear system solution, making the inversion of an upper and a lower triangular matrix, once per each right-hand side. The first and the last steps are quite straightforward and less expensive than the decomposition computation. For this reason the cost of direct approaches is less dependent on the number of right-hand side vectors to be considered. Nevertheless the computational burden of the direct Cholesky decomposition method is of $O(N^3_{\mathrm{DOFs}})$ for full matrices and of $O(N^{3/2}_{\mathrm{DOFs}})$ for sparse matrices using, for example, nested dissection [29]. Moreover they can be problematic due to memory limits for very large problems as stiffness matrix factorizations need to be computed and stored. For these reasons, for large sparse matrices iterative methods can be particularly inexpensive especially when accepting an error on the system solution [30]. In topology optimization, these methods can take advantage from the solutions evaluated at the previous optimization iteration, especially close to convergence. On the other hand, a major drawback of these approaches is that the computational effort is much more linked with the number of right-hand sides. In fact a first solution needs to be computed for displacements evaluation, then the right-hand side of the adjoint problems can be built as it depends on displacements, and finally the adjoint vectors can be computed through another iterative process. Even a parallel version with respect to the number of right-hand side vectors could not enable the evaluation of adjoint in parallel with the displacements for iterative approaches. Compliance-based problems being self-adjoint do not suffer of these drawbacks, because displacement vector is also the adjoint vector for the compliance response. In Matlab several iterative approaches are implemented in native functions like *pcg, cgs,* and *minres*. Moreover, incomplete factorizations (*ichol* and *ilu* functions) are also implemented in order to provide preconditioners. Several trade-offs have been assessed for 27,000-DOF problem in Fig. 3a. In Fig. A1a *minres* has been used to find the largest residual error tolerance needed to achieve a correct evaluation of both stress and ΔTSFC constraint gradients. The reference displacement and gradient were evaluated with *mldivide* Matlab native function (i.e., with a direct approach) still achievable on small problems. On the basis of this study a value of 10^{-5} for the residual error tolerance has been considered as sufficiently small. Finally in order to select the algorithm best-suited for our problem, several Matlab native functions (*pcg, bcg, bcgs, bcgsl, cgs, minres, qmr*) have been compared with the same residual tolerance and with maximum number of iterations big enough to ensure that every method reached residual error smaller than tolerance value. Incomplete Choleski factorization (*ichol* Matlab function) was used to provide a preconditioner. *Ichol* drop tolerance and diagonal compensation were kept constant to 10^{-3} and 10^{-4}, respectively. The elapsed time for each tested function is compared in Fig. A1b for displacement

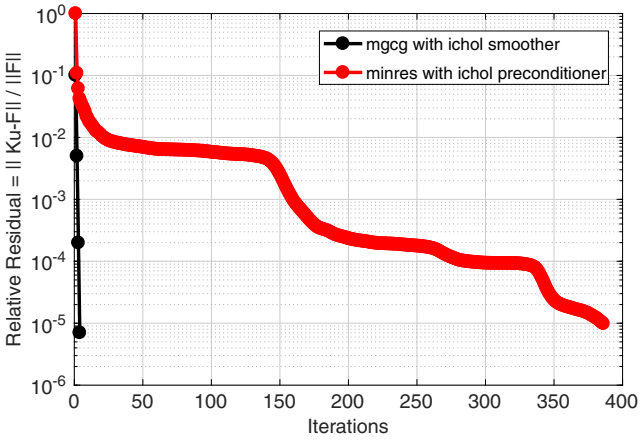


Fig. B1 Convergence history of ichol preconditioned *minres* and *mgcg* with ichol smoother.

evaluation in a configuration constituted by maximal Young's modulus everywhere. The preconditioned conjugate gradient *pcg* and *minres* were the fastest algorithms tested initially. To further reduce the computation time in the most refined mesh 7a, a geometric multigrid preconditioned conjugate gradient (mgcg) [30,31] was tested, using three refinement meshes and replacing damped Jacobi with incomplete Cholesky for the smoothening. The convergence history of ichol preconditioned *minres* and *mgcg* is compared to compute displacement vector for the mesh in Fig. B1 for the configuration with all design variables equal to 1 and with a convergence criteria of 10^{-5} over the relative residual. Multigrid preconditioners are way more effective than Matlab native function iterative solvers (4 vs 386 iterations needed to converge) and were therefore considered for the topology optimization problem analyzed in Sec. V.

References

[1] Lattime, S. B., and Steinetz, B. M., "Turbine Engine Clearance Control Systems: Current Practices and Future Directions," *38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, AIAA Paper 2002-3790, July 2002. doi:10.2514/6.2002-3790

[2] Benito, D., Dixon, J., and Metherell, P., "3D Thermomechanical Modelling Method to Predict Compressor Local Tip Running Clearances," *Proceedings of the ASME Turbo Expo*, 2008, pp. 1–10, http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid=1624570.

[3] Bettebghor, D., Blondeau, C., Toal, D., and Eres, H., "Bi-Objective Optimization of Pylon-Engine-Nacelle Assembly: Weight vs. Tip Clearance Criterion," *Structural and Multidisciplinary Optimization*, Vol. 48, No. 3, 2013, pp. 637–652. doi:10.1007/s00158-013-0908-7

[4] Bendsøe, M. P., and Kikuchi, N., "Generating Optimal Topologies in Structural Design Using a Homogenization Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, No. 2, 1988, pp. 197–224. doi:10.1016/0045-7825(88)90086-2

[5] Bendsøe, M. P., "Optimal Shape Design as a Material Distribution Problem," *Structural Optimization*, Vol. 1, No. 4, 1989, pp. 193–202. doi:10.1007/BF01650949

[6] Allaire, G., Jouve, F., and Toader, A.-M., "Structural Optimization Using Sensitivity Analysis and a Level-Set Method," *Journal of Computational Physics*, Vol. 194, No. 1, 2004, pp. 363–393. doi:10.1016/j.jcp.2003.09.032

[7] Xie, Y. M., and Steven, G. P., "A Simple Evolutionary Procedure for Structural Optimization," *Computers & Structures*, Vol. 49, No. 5, 1993, pp. 885–896. doi:10.1016/0045-7949(93)90035-C

[8] Zhu, J.-H., Zhang, W.-H., and Xia, L., "Topology Optimization in Aircraft and Aerospace Structures Design," *Archives of Computational Methods in Engineering*, Vol. 23, No. 4, 2016, pp. 595–622. doi:10.1007/s11831-015-9151-2

[9] Remouchamps, A., Bruyneel, M., Fleury, C., and Grihon, S., "Application of a Bi-Level Scheme Including Topology Optimization to the Design of an Aircraft Pylon," *Structural and Multidisciplinary Optimization*, Vol. 44, No. 6, 2011, pp. 739–750. doi:10.1007/s00158-011-0682-3

[10] Xue, C.-J., Xu, F.-J., Xu, Y., and Tan, W., "Structural Topology Optimization of a Pylon's Mount Using Ant Colony Algorithms," *Journal of Aircraft*, Vol. 49, No. 3, 2012, pp. 724–734. doi:10.2514/1.C031612

[11] Deparis, S., Forti, D., and Quarteroni, A., "A Rescaled Localized Radial Basis Function Interpolation on Non-Cartesian and Nonconforming Grids," *SIAM Journal on Scientific Computing*, Vol. 36, No. 6, 2014, pp. A2745–A2762. doi:10.1137/130947179

[12] Rozvany, G., "On Design-Dependent Constraints and Singular Topologies," *Structural and Multidisciplinary Optimization*, Vol. 21, No. 2, 2001, pp. 164–172. doi:10.1007/s001580050181

[13] Cheng, G., and Guo, X., "ϵ-Relaxed Approach in Structural Topology Optimization," *Structural Optimization*, Vol. 13, No. 4, 1997, pp. 258–266. doi:10.1007/BF01197454

[14] Bruggi, M., "On an Alternative Approach to Stress Constraints Relaxation in Topology Optimization," *Structural and Multidisciplinary Optimization*, Vol. 36, No. 2, 2008, pp. 125–141. doi:10.1007/s00158-007-0203-6

[15] Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *Computer Aided Design of Control Systems*, Elsevier, 1980, pp. 113–117. doi:10.1016/B978-0-08-024488-4.50022-X

[16] Yang, R., and Chen, C., "Stress-Based Topology Optimization," *Structural Optimization*, Vol. 12, Nos. 2–3, 1996, pp. 98–105.

[17] Duysinx, P., and Sigmund, O., "New Developments in Handling Stress Constraints in Optimal Material Distribution," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 1998-4906, 1998.

[18] Le, C., Norato, J., Bruns, T., Ha, C., and Tortorelli, D., "Stress-Based Topology Optimization for Continua," *Structural and Multidisciplinary Optimization*, Vol. 41, No. 4, 2010, pp. 605–620. doi:10.1007/s00158-009-0440-y

[19] Verbart, A., Langelaar, M., and Van Keulen, F., "A Unified Aggregation and Relaxation Approach for Stress-Constrained Topology Optimization," *Structural and Multidisciplinary Optimization*, Vol. 55, No. 2, 2017, pp. 663–679. doi:10.1007/s00158-016-1524-0

[20] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., "Efficient Topology Optimization in MATLAB Using 88 Lines of Code," *Structural and Multidisciplinary Optimization*, Vol. 43, No. 1, 2011, pp. 1–16. doi:10.1007/s00158-010-0594-7

[21] Talischi, C., Paulino, G. H., Pereira, A., and Menezes, I. F., "PolyTop: A Matlab Implementation of a General Topology Optimization Framework Using Unstructured Polygonal Finite Element Meshes," *Structural and Multidisciplinary Optimization*, Vol. 45, No. 3, 2012, pp. 329–357. doi:10.1007/s00158-011-0696-x

[22] Lazarov, B. S., and Sigmund, O., "Filters in Topology Optimization Based on Helmholtz-Type Differential Equations," *International Journal for Numerical Methods in Engineering*, Vol. 86, No. 6, 2011, pp. 765–781. doi:10.1002/nme.v86.6

[23] Coniglio, S., Gogu, C., and Morlier, J., "Weighted Average Continuity Approach and Moment Correction: New Strategies for Non-Consistent Mesh Projection in Structural Mechanics," *Archives of Computational Methods in Engineering*, 2018, pp. 1–29. doi:10.1007/s11831-018-9285-0

[24] Bernardi, C., Maday, Y., and Patera, A. T., "Domain Decomposition by the Mortar Element Method," *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, edited by H. G. Kaper, M. Garbey, and G. W. Pieper, Vol. 384, NATO ASI Series (Series C: Mathematical and Physical Sciences), Springer, Dordrecht, The Netherlands, 1993, pp. 269–286. doi:10.1007/978-94-011-1810-1

[25] Krog, L., Tucker, A., Kemp, M., and Boyd, R., "Topology Optimisation of Aircraft Wing Box Ribs," *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2004-4481, 2004.

[26] Svanberg, K., "The Method of Moving Asymptotes—A New Method for Structural Optimization," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 2, 1987, pp. 359–373. doi:10.1002/(ISSN)1097-0207

[27] Svanberg, K., "Some Modelling Aspects for the Matlab Implementation of MMA," KTH Royal Inst. of Technology Tech. Rept., Stockholm, Sept. 2004, https://people.kth.se/~krille/mmamatlab.pdf.

[28] Lian, H., Christiansen, A. N., Tortorelli, D. A., Sigmund, O., and Aage, N., "Combined Shape and Topology Optimization for Minimization of Maximal von Mises Stress," *Structural and Multidisciplinary Optimization*, Vol. 55, No. 5, 2017, pp. 1541–1557. doi:10.1007/s00158-017-1656-x

[29] Davis, T. A., "Direct Methods for Sparse Linear Systems," *Fundamentals of Algorithms*, Vol. 2, SIAM, 2006. doi:10.1137/1.9780898718881

[30] Amir, O., Aage, N., and Lazarov, B. S., "On Multigrid-CG for Efficient Topology Optimization," *Structural and Multidisciplinary Optimization*, Vol. 49, No. 5, 2014, pp. 815–829. doi:10.1007/s00158-013-1015-5

[31] Aage, N., Andreassen, E., Lazarov, B. S., and Sigmund, O., "Giga-Voxel Computational Morphogenesis for Structural Design," *Nature*, Vol. 550, No. 7674, 2017, pp. 84–86. doi:10.1038/nature23911