



Lucas do Nascimento Sagrilo

**Structural topology optimization with many
load cases: stochastic approximation and
singular value decomposition approaches**

Dissertação de Mestrado

Thesis presented to the Programa de Pós-graduação em Engenharia Mecânica, do Departamento de Engenharia Mecânica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica.

Advisor: Prof. Anderson Pereira

Rio de Janeiro
October 2022



Lucas do Nascimento Sagrilo

**Structural topology optimization with many
load cases: stochastic approximation and
singular value decomposition approaches**

Thesis presented to the Programa de Pós-graduação em Engenharia Mecânica da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica. Approved by the Examination Committee:

Prof. Anderson Pereira

Advisor

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Daniel Milbrath de Leon

Departamento de Engenharia Mecânica – UFRGS

Prof. Ivan Fábio Mota de Menezes

Departamento de Engenharia Mecânica – PUC-Rio

Rio de Janeiro, October the 7th, 2022

All rights reserved.

Lucas do Nascimento Sagrilo

Graduado em Engenharia Mecânica pela Pontifícia Universidade Católica do Rio de Janeiro com Duplo Diploma na École CentraleSupélec, França. Suas áreas de interesse são Mecânica Computacional, Método dos Elementos Finitos, Análise Estrutural e aplicações de Otimização aos problemas de dimensionamento estrutural.

Bibliographic data

Sagrilo, Lucas N.

Structural topology optimization with many load cases: stochastic approximation and singular value decomposition approaches / Lucas do Nascimento Sagrilo; advisor: Anderson Pereira. – 2022.

80 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2022.

Inclui bibliografia

1. Mechanical Engineering – Teses. 2. Otimização Topológica. 3. Método dos Elementos Finitos. 4. Carregamentos Estáticos Equivalentes. 5. Aproximação Estocástica. 6. Decomposição em Valores Singulares. I. Pereira, Anderson. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.

CDD: 621

Acknowledgments

To my family, specially to my parents Luís Volnei Sudati Sagrilo and Joseane do Nascimento Sagrilo for the tireless support to all my decisions about study and work, and also for their dedication in giving the best for their sons in terms of education and life quality.

To my advisor, Prof. Anderson Pereira, for his patience and interest in exchanging about the results found along this work and in correcting this dissertation, and for accepting the challenge of advise me during this masters work.

To Prof. Ivan Fábio Mota de Menezes, who exchanged a lot with me and guided me for taking the best decisions for my professional carreer when I finished my bachelor course.

To my long time friends: Henrique Santiago, João Pedro Cunha and Mateus Bastos Neiva for their support in helping me with the difficulties associated with this masters and with some important personal and professional decisions.

To the institution Pontifícia Universidade Católica do Rio de Janeiro for the constant search in improving its infrastructure, academic staff and opportunities offered to students.

To the managers and colleagues that I already had in my professional life for their valuable contribution to my technical skills, in special to Jimmy Baussard, Olivier Prin, Nicolas Gallice, Bruno Holzberg, Romulo Aguiar, Eduardo D’Almeida and Anna Paula Duarte.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

Abstract

Sagrilo, Lucas N.; Pereira, Anderson (Advisor). **Structural topology optimization with many load cases: stochastic approximation and singular value decomposition approaches**. Rio de Janeiro, 2022. 80p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

It is known that most real structures are subject to different loading scenarios, related to different structural solicitations and the action of natural forces, such as winds and sea waves. In this context, it is important to consider the effect of the largest number of possible scenarios that can act on a structure when performing a topology optimization study. The traditional way of solving this type of problem involves a case-by-case analysis of the scenarios, which in the context of a structural optimization algorithm requires the solution of one finite element problem for each scenario and at each step of the algorithm, being limited by the high associated computational cost. This limitation leave room for approaches based on dimension reduction such as stochastic approximation and decomposition into singular values. This work verifies the feasibility of using these two approaches to solve structural topology optimization problems with many load cases. Two applications are presented, robust optimization and the problem of dynamic loads using the equivalent static loading method. Thus, situations involving more complex loads can be studied through efficient topology optimization algorithms. For both cases, comparisons are established between the results obtained through the methodology developed in this work and the ones from the literature.

Keywords

Topology Optimization; Finite Element Method; Equivalent Static Loads; Stochastic Approximation; Singular Value Decomposition.

Resumo

Sagrilo, Lucas N.; Pereira, Anderson. **Otimização topológica estrutural com muitos casos de carga: abordagens aproximação estocástica e decomposição de valores singulares**. Rio de Janeiro, 2022. 80p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Sabe-se que a maioria das estruturas reais estão sujeitas à diferentes casos de carregamentos, relacionadas à diferentes solicitações estruturais e à ação de forças naturais, como ventos e ondas. Neste contexto, é importante levar em consideração o efeito da maior quantidade de cenários possíveis que possam atuar em uma estrutura ao realizar um estudo de otimização topológica. A maneira tradicional de solução deste tipo de problema envolve uma análise caso a caso dos cenários, o que no contexto de um algoritmo de otimização estrutural requer a solução de um problema de elementos finitos para cada cenário em cada passo do algoritmo, ficando limitada pelo elevado custo computacional associado. Esta limitação abre espaço para abordagens baseadas em redução de dimensões como a aproximação estocástica e a decomposição em valores singulares. Este trabalho verifica a viabilidade do uso destes dois métodos na solução de problemas de otimização topológica estrutural com muitos casos de carga. Duas aplicações são apresentadas, otimização robusta e o problema de cargas dinâmicas usando o método do carregamento estático equivalente. Com isso, situações envolvendo carregamentos mais complexos podem ser estudadas através de algoritmos eficientes de otimização topológica. Para ambos os casos, são mostrados resultados comparando os resultados obtidos através da metodologia desenvolvida neste trabalho com resultados da literatura.

Palavras-chave

Otimização Topológica; Método dos Elementos Finitos; Carregamentos Estáticos Equivalentes; Aproximação Estocástica; Decomposição em Valores Singulares.

Table of contents

1	Introduction	14
1.1	Work Motivation	14
1.2	Structural Optimization	15
1.3	Bibliography Review	16
1.4	Objectives	18
1.5	Organization of the dissertation	19
2	Theoretical Background	20
2.1	Sensitivity Analysis	20
2.2	HHT- α Integration Method	23
2.3	Monte-Carlo Simulation	24
3	Multiple Load Cases Topology Optimization	27
3.1	General Topology Optimization Problem	27
3.2	PolyTop software	30
3.3	Topology Optimization involving many load cases scenarios	32
3.4	SA-approach	33
3.5	SVD-approach	37
3.6	Numerical example	39
4	Time-Dependent Topology Optimization	43
4.1	Introduction to PolyDyna software	44
4.2	PolyDyna's implementation	44
4.3	The Equivalent Static Loads (ESL) method	45
4.4	Implementation of ESL-approach into PolyDyna software	47
4.5	Case studies	48
4.5.1	Cantilever beam subjected to one half-cycle sinusoidal load	49
4.5.2	Support structure under rotating load	52
4.5.3	Bridge under moving load	55
5	Robust Topology Optimization (RTO)	59
5.1	Concepts of Robust Topology Optimization	59
5.2	Use of SVD approach in RTO context	61
5.3	Update Scheme in RTO problems	63
5.4	Numerical examples	63
5.4.1	Simply supported beam	63
5.4.2	Simply supported beam with uncertainty in load direction	66
6	Conclusions and Perspectives	70
6.1	Work conclusions	70
6.2	Suggestions for future work	71
	Bibliography	72
	Appendices	77

Appendices	77
A Published article	77
B MATLAB Code for Static Analysis in Time-Dependent Problems	78
C Modified PolyScript for ESL-based Time-Dependent Topology Optimization	79

List of figures

Figure 1.1	Examples of structures subjected to many different load conditions: long bridge in a highway (left) [1]; offshore wind turbines (right) [2].	14
Figure 1.2	Examples of structural optimization [3]: (a) parametric optimization; (b) shape optimization; (c) topology optimization.	16
Figure 2.1	Graphical illustration of forward (in the left) and central (in the right) finite difference approximations for the derivative of the scalar-valued function f at the point $x = x_0$ (Adapted from [4]).	21
Figure 2.2	Cantilever beam subjected to uncertainties in load intensity, load position and material Young's modulus	25
Figure 2.3	Example to illustrate the Monte Carlo method	26
Figure 3.1	Discretization of the domain Ω	28
Figure 3.2	Graphics of $f(\rho) = \rho^p$ for different values of ρ and p	29
Figure 3.3	Working scheme of Topology Optimization based on SIMP method	30
Figure 3.4	Regular and hexagonal meshes for one rectangular domain generated with PolyMesher [5].	31
Figure 3.5	Comparison of the optimization history of a two-parameters objective function with and without the proposed damping scheme [6]	36
Figure 3.6	Working scheme for Topology Optimization employing the SA-approach	37
Figure 3.7	Working scheme for Topology Optimization employing the SVD-approach	39
Figure 3.8	2D box domain (Adapted from [6])	39
Figure 3.9	108 independent load cases acting on the 2D box domain (Adapted from [6])	40
Figure 3.10	Final topologies obtained for the 108 load cases scenarios example and comparison with literature results	41
Figure 3.11	Values of the objective function along the optimization iterations for the three studied methods in the example of a box domain under 108 load cases	42
Figure 4.1	Illustration of the ESL method (Adapted from [7])	45
Figure 4.2	Flowchart for time-dependent topology optimization process using the ESL method (Adapted from [8]).	47
Figure 4.3	Illustration of the cantilever beam example with the three different sinusoidal loads studied (Adapted from [9])	49
Figure 4.4	Singular values of the ESL matrix at the beginning of the first cycle of topology optimization	50

Figure 4.5 Comparison between the final topologies obtained with the original and the modified PolyDyna for the three studied load durations	51
Figure 4.6 Graphics of the initial and optimized dynamic compliance for the three studied scenarios	51
Figure 4.7 Illustration of the support structure example with the the rotating load at the mid point of the upper edge (Adapted from [9])	52
Figure 4.8 Singular values of the ESL matrix at the beggining of the first cycle of topology optimization	53
Figure 4.9 Comparison between the final topologies obtained with the original and the modified PolyTop for the three studied load durations	54
Figure 4.10 Graphics of the initial and optimized dynamic compliance for the three studied scenarios	55
Figure 4.11 Illustration of the bridge example with the moving load (Adapted from [9])	56
Figure 4.12 Example of mesh used in the bridge example	56
Figure 4.13 Singular values of the ESL matrix at the begging of the first cycle of topology optimization	57
Figure 4.14 Comparison between the final topologies obtained with the original and the modified PolyDyna for the four studied velocities V_0 .	58
Figure 5.1 Illustration of the simply supported beam example with the detail of the applied loads and the mesh employed in the numerical simulations	64
Figure 5.2 Histograms of the samples of the three loads in the simply supported beam example	65
Figure 5.3 Singular values of the matrix that gathers the load cases used in the simply supported beam example	65
Figure 5.4 Final topologies obtained for the three studied combinations of values for a and b in the simply supported beam example	66
Figure 5.5 Evolution of the expected compliance and variance values along the optimization iterations for the three studied combinations of a and b .	67
Figure 5.6 Illustration of the simply supported beam example with uncertainty in the direction of the loads.	67
Figure 5.7 Histograms of the values for the angles θ_1 , θ_2 and θ_3 in the simulations involving uncertainty in the load direction.	68
Figure 5.8 Singular values of the matrix that gathers the load cases used in the example (left) with the detail of the 10 first singular values (right)	68
Figure 5.9 Final topologies obtained in the example with uncertainty in load magnitude and in load direction.	69

List of tables

Table 3.1	Physical and geometrical properties of the 2D box domain	39
Table 3.2	First 10 singular values in the 108 independent load cases problem	41
Table 3.3	Comparison between the simulations for the 108 load cases scenarios example	41
Table 3.4	Total number of Optimally Criteria updates and Finite Element solves for each of the methods	42
Table 4.1	Main Matlab files of PolyDyna software (Adapted from [9])	45
Table 4.2	Parameters in the cantilever beam example	49
Table 4.3	Numerical results for the cantilever beam with load duration of 0.05 seconds	52
Table 4.4	Numerical results for the cantilever beam with load duration of 0.03 seconds	52
Table 4.5	Numerical results for the cantilever beam with load duration of 0.01 seconds	52
Table 4.6	Parameters in the support example	53
Table 4.7	Numerical results in the support example with angular speed $\omega_1 = 50\pi$ rad/s	54
Table 4.8	Numerical results in the support example with angular speed $\omega_2 = 100\pi$ rad/s	55
Table 4.9	Numerical results in the support example with angular speed $\omega_3 = 200\pi$ rad/s	55
Table 4.10	Parameters in the bridge example	57
Table 5.1	Parameters in the simply supported beam example	64
Table 5.2	Mean and standard deviation values from the samples used in the simply supported beam example	64
Table 5.3	Comparison of the expected compliance and variance results with the ones from Dunning and Kim [10]	66
Table 5.4	Parameters in the simply supported beam example with uncertainty in load direction	69
Table 5.5	Final values of expected compliance, variance and objective function in the example with uncertainty in load direction	69

List of symbols

\mathbf{K}	Stiffness matrix
\mathbf{C}	Damping matrix
\mathbf{M}	Mass matrix
\mathbf{u}	Displacements vector
$\dot{\mathbf{u}}$	First time derivative of displacements vector \mathbf{u}
$\ddot{\mathbf{u}}$	Second time derivative of displacements vector \mathbf{u}
\mathbf{F}	Load cases matrix
\mathbf{U}	Displacements matrix associated to load cases inside \mathbf{F}
$\boldsymbol{\rho}$	Design variables vector
\mathbf{g}	Constraints vector
$\boldsymbol{\lambda}$	Adjoint vector
Δt	Time interval for time discretization
Ω	Initial domain for topology optimization
ϕ	Compliance
\mathbf{D}	Constitutive tensor
E	Young's modulus
ν	Poisson's ratio
p	Penalty factor inside SIMP method
m	Total number of elements in a finite elements mesh
v	Mesh element volume
V_{\max}	Volume constraint in topology optimization
\mathbf{f}	Load case
n	Total number of load cases in a topology optimization problem
ρ_{\min}	Minimum allowable density value inside $\boldsymbol{\rho}$
ρ_{\max}	Maximum allowable density value inside $\boldsymbol{\rho}$
\mathbb{E}	Expected value
$\boldsymbol{\xi}$	Rademacher random vector
$\boldsymbol{\xi}_k$	Sample of $\boldsymbol{\xi}$
n_s	Number of samples
$\hat{\phi}$	Compliance estimator
R	Effective step ratio
\mathbf{U}	Left singular vectors matrix
\mathbf{V}	Right singular vectors matrix
$\boldsymbol{\Sigma}$	Singular values matrix
n_{SA}	Number of Rademacher random vector samples

n_{SVD}	Number of dominant singular values
t_i	Initial time step for dynamic compliance evaluation
t_f	Final time step for dynamic compliance evaluation
\mathbf{s}	Equivalent static load
ϕ	Compliance vector
Rb	Objective function for robust topology optimization
Var	Variance

1

Introduction

1.1

Work Motivation

When analyzing real engineering structures, evidently they are not only subjected to one deterministic load case scenario. They can be affected by loads from different origins, with different natures, and all their effects should be taken into account in structural optimization problems. To better illustrate, two examples of structures that are subjected to many different load cases are shown in Figure 1.1.



Figure 1.1: Examples of structures subjected to many different load conditions: long bridge in a highway (left) [1]; offshore wind turbines (right) [2].

In the examples of Figure 1.1, the bridge is subjected to the traffic load, which has a random nature and also causes many dynamic effects on the structure. It is also important to consider the effect of the wind in the bridge design, which is also a random phenomenon. In the case of offshore wind turbines in Figure 1.1, in addition to the wind effects, sea waves are an important source of different loading scenarios that act on the structure.

The traditional approaches for solving structural optimization problems where many loading scenarios are involved imply a case by case analysis [6], which is extremely expensive from a computational point of view, as one finite element analysis is necessary for each studied scenario, resulting in very slow and inefficient algorithms.

To improve their efficiency, the literature focuses on model reduction methods that define one small set of load cases that represents equivalently

the original set of many loads. The reduction in the number of load cases implies in a small number of necessary finite element solves until reach the final topologies, increasing the performance of the topology optimization algorithms.

In this context two main works were found in the literature. The first is Zhang *et al.* [6], where the SA (Stochastic Approach)-based approach is presented, involving the sampling of one random vector to define associated loading scenarios. The other article is Tarek and Ray [11], where the SVD (Singular Value Decomposition)-based approach is discussed: equivalent loading scenarios are generated from one singular value decomposition of the matrix that gathers the loading scenarios. These methodologies are discussed in detail in Chapter 3.

The use of these two approaches is already disseminated in the literature, although only in cases where the exact set of loading scenarios that the structure is subjected to is known. Although, as showed by the applications in Figure 1.1, it is essential for engineering applications to extend the applicability of the model reduction methods for problems that involves dynamic and random loads, this last also known as robust optimization.

With this point of improvement identified, this work focuses on understanding the practices adopted for time-dependent and robust topology optimization and show how the two methods mentioned for model reduction can be applied in these scenarios to improve the efficiency of the optimization algorithms.

This work aims to contribute with methodologies and algorithms for engineers and researchers to consider efficiently the effect of time-dependent and random loads into the topology optimization formulations.

1.2

Structural Optimization

Before focusing on topology optimization, to better situate the reader, it is important to recall what structural optimization is and differentiate parametric, shape, and topology optimization approaches.

It is possible to define structural optimization as the process of finding one or more properties associated with the structure to minimize the objective function value associated with the problem in study, respecting the imposed constraints. Some examples of structural properties that can be optimized are the dimensions of the structure, the shape of one hole, and the thickness of the domain, and some examples of constraints are the maximum admissible stress in one of the structure members, the maximum displacement at one of

the extremities and the amount of material in the structure.

In Figure 1.2 it is reproduced one figure from Bendsoe and Sigmund [3] that shows well the difference between parametric, shape, and topology optimization approaches.

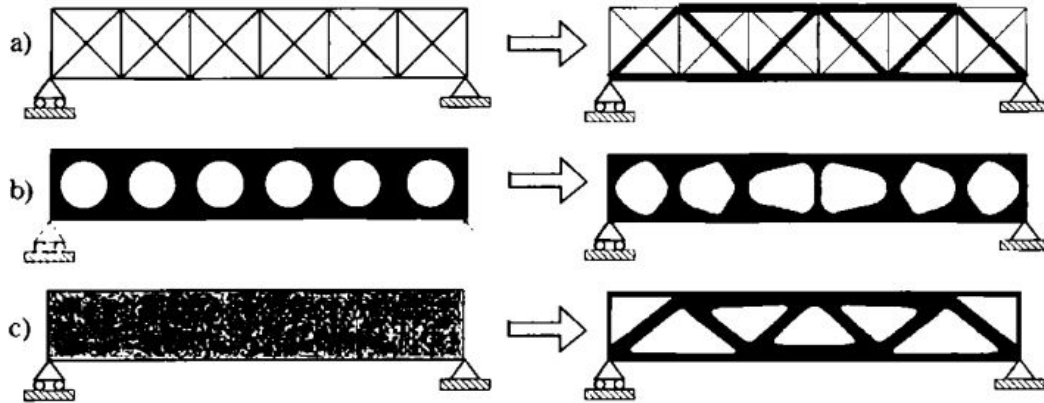


Figure 1.2: Examples of structural optimization [3]: (a) parametric optimization; (b) shape optimization; (c) topology optimization.

The objective in Figure 1.2a is to set the optimal dimensions of the bars in the truss structure, which represents one parametric optimization problem. However, in Figure 1.2b, what is optimized is the shape of the holes that exist along the length of the bar, illustrating one situation of shape optimization. The example in Figure 1.2c illustrates a topology optimization problem: from the initial domain, it is found the material distribution that minimizes the objective function value respecting the imposed volume constraint – *i.e.* the maximum volume of the final structure is a percentage of the initial domain volume. This is the application of structural optimization that is studied in this dissertation and is described in details in Section 3.1.

1.3

Bibliography Review

The first work that is known to deal with the problem of reducing the amount of material used in the design of structures is the article by Michell [12] published at the beginning of the 20th century, although it was quite limited due to the few computational tools available at the time. It is interesting to remember that the pioneering works in the Finite Element Method, such as Hrennikoff [13] and Courant [14] date from the 1940s. In the 1960s, with the advances in Mathematical Programming and the Finite Element Method, Schmit [15] was the pioneer in publishing one work combining these two tools, to solve non-linear elastic structural optimization problems.

After Schmit's work [15], many studies were performed focusing mostly on truss and beam structures. But in 1988 Bendsøe and Kikuchi [16] introduced the concept of topology optimization using one homogenization method, based on Finite Element solutions to propose the optimal distribution of material on the basis domain, respecting the imposed volume constraint as explained in Section 3.1.

With the repercussion of Bendsøe and Kikuchi's work [16], many researchers started working on extending the proposed formulation for different contexts, such as problems involving time-dependent loads [17, 18, 8, 9] and random loads [19, 10, 11], cases that are addressed in this work. But it is also possible to enumerate many other applications and new developments, such as topology optimization under stress constraints [20] and modal analysis [21].

A great advantage of studying Topology Optimization is the culture of publishing and sharing the computational codes used in the development of the published papers. For example, many open-source softwares can be found in literature, such as **PolyTop** [22, 5], 99 lines from Sigmund [23], 88 lines from Andreassen *et al.* [24] and TOBS from Picelli *et al.* [25]. The first three software focus on providing very simple programs to allow users to understand in practice how the classical process of Topology Optimization works step-by-step, but the last one has the objective of providing a very efficient program based on the last advances in Mathematical Programming.

In the beginnings of topology optimization, most effort focused on solving problems involving only static loads. But, in 1999 Min *et al.* [26] realized that many techniques developed for static problems could be used for topology optimization involving dynamic loads. Since this article was published, this subject has been receiving many attention, and in this context Giraldo-Londoño and Paulino [9] introduced the **PolyDyna** software, which is an adaptation of the **PolyTop** program to consider dynamic loads. But one of the challenges that the authors face when studying optimization applied to time-dependent problems, which includes **PolyDyna**, is the computational cost involved to solve the dynamic response and perform the sensitivity analysis at each step of the optimization algorithm. In this context, Kang *et al.* [27] and Park and Choi [7] introduced the concept of Equivalent Static Loads (ESL) into the optimization of beam and truss structures. This approach simplifies the dynamic problem into a static one subjected to multiple loading scenarios, significantly reducing the number of dynamic response computations until the final result is achieved. More recently, it was shown by the works of Jang *et al.* [28] and Lavôr and Pereira [8, 29] that the ESL formulation can also be used in the context of continuum topology optimization, and implemented inside

the PolyTop software. This topic is discussed in details in Chapter 3.

Another interesting situation widely studied is the Topology Optimization of structures subjected to random loads [19, 10], also known as Robust Topology Optimization (RTO). A very simple way to solve this problem is through a Monte-Carlo simulation, where each sample of the loads can be treated as one independent loading scenario. But, as a very large number of samples are required to have one satisfactory final result, many authors propose alternative approaches for the robust problem. In Asadpoure *et al.* [30] and Guest and Igusa [31], for example, analytical methods are proposed to efficiently solve the robust problem. Cuellar *et al.* [32] suggests one approach based on polynomial chaos expansion to solve the same question.

With the examples mentioned above, it is clear that the topology optimization of structures subjected to different loading scenarios is important, but the traditional ways to solve it involve the calculation of the displacements associated with each of the different loading conditions and at each step of the optimization algorithm, which implies in very slow algorithms. Zhang *et al.* [6] is the first work that discusses one alternative for this question, where one stochastic algorithm is proposed, with one strong reduction in the number of loading scenarios to be analyzed at each step of the optimization. More recently, another even more efficient algorithm was proposed by Zhang *et al.* [33], also based on one stochastic sampling of the load cases acting on the structure.

An alternative to the stochastic algorithm is proposed by Tarek and Ray [11], based on a Singular Value Decomposition (SVD) of the matrix that gathers the loading scenarios of the problem. This approach is much easier to implement, but can only perform great reductions in the number of loading scenarios to be analyzed by the optimization algorithm when the number of cases is small or if they are highly correlated. The stochastic and SVD-based methods to improve the computational performance of topology optimization involving many loading scenarios are the main focus of this dissertation and are also discussed in details in Chapter 3.

1.4 Objectives

The main objective of this dissertation is to show the applicability of model reduction methods (SA and SVD-based approaches) into time-dependent and robust topology optimization. To achieve this goal, the following steps are performed:

- Show the equivalence between the solution of a multiple load cases topol-

ogy optimization problem using all the involved loads and employing the SA and SVD approaches;

- Adapt the **PolyDyna** code to generate sets of load cases based on the ESL-approach instead of calculating solutions of the motion equation at all steps, and then use the mentioned approaches to define equivalent loads associated to the set of ESLs;
- Demonstrate the possibility of solving Robust Topology Optimization problems using the Monte-Carlo method in a reasonable time if the model reduction methods are used.

1.5

Organization of the dissertation

This dissertation is organized as follows:

- **Chapter 2** presents the principal theoretical subjects used in this work: Finite Element Method (FEM), sensitivity analysis, numerical solution of the motion equation (using the HHT- α method), and Monte-Carlo simulation.
- In **Chapter 3** it is presented the compliance-based topology optimization formulation and its extension to the multiple load scenario. Then, the problem studied in this work is recapitulated, and the two studied model reduction methods are described. One example that illustrates the application of the methods is discussed.
- **Chapter 4** shows how the multiple load formulation in topology optimization can be used in a problem involving dynamic loads. Then, it is presented how the model reduction approaches can be used in this scenario, and it is show how the **PolyDyna** software [9] is modified to incorporate the discussed methods. In the end, some numerical examples are discussed.
- **Chapter 5** discusses how a Monte Carlo sampling of the uncertain loads can be used in a robust topology optimization problem using the SVD-based approach. Two numerical examples are shown to illustrate this application.
- **Chapter 6** presents the conclusions and the suggestions for future works.

2

Theoretical Background

In this Chapter, it is discussed the main theoretical concepts that must be understood before proceeding to the discussion about model reduction methods in topology optimization. First, it is shown a brief discussion about sensitivity analysis, in the context of obtaining the necessary derivatives used as input in the optimization algorithms employed along with this work. Then, as time-dependent and random scenarios are the focus of this dissertation, the definitions of the HHT- α and Monte-Carlo methods are described as they are the main means to solve these scenarios.

2.1

Sensitivity Analysis

Sensitivity Analysis is a critical point for any situation involving optimization because the vast majority of the existing efficient algorithms to solve these problems need the computation of the derivatives of the objective and constraint functions with respect to each of the design variables. And the objective of the Sensitivity Analysis is to provide numerical and analytical methods to calculate the necessary derivatives, each one of the methods with its computational complexity, and then, the user can choose the best one according to the application at hand.

The simplest way to calculate the sensitivities is by the Finite Differences Method [4], which is based on the mathematical definition of derivative. For a vector-valued function $\mathbf{f}(\mathbf{x})$, this definition is:

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_j} = \lim_{\Delta x_j \rightarrow 0} \frac{\mathbf{f}(\mathbf{x} + \Delta x_j \mathbf{e}_j) - \mathbf{f}(\mathbf{x})}{\Delta x_j}, \quad (2-1)$$

where Δx_j represents one perturbation in the j -th entry of \mathbf{x} and \mathbf{e}_j is one vector that has the same size as the number of variables inside \mathbf{x} and has its j -th component equal to 1 and the others are equal to 0.

With the definition in Equation (2-1), it is possible to establish one reasonable numerical approach for the derivative of $\mathbf{f}(\mathbf{x})$ with respect to x_j , which is called the Forward Finite Difference:

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_j} \approx \frac{\mathbf{f}(\mathbf{x} + \Delta x_j \mathbf{e}_j) - \mathbf{f}(\mathbf{x})}{\Delta x_j}, \quad (2-2)$$

with Δx_j here representing the step of the finite difference associated with the j -th variable, with typical values used being around 10^{-5} . Other similar expressions can be found in the literature, also coming from the same reasoning: the Backward and Central Finite Differences shown in Equations (2-3) and (2-4), respectively:

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_j} \approx \frac{\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \Delta x_j \mathbf{e}_j)}{\Delta x_j} \quad (2-3)$$

$$\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_j} \approx \frac{\mathbf{f}(\mathbf{x} + \Delta x_j \mathbf{e}_j) - \mathbf{f}(\mathbf{x} - \Delta x_j \mathbf{e}_j)}{2\Delta x_j}. \quad (2-4)$$

Figure 2.1 illustrates the difference in calculating the derivative by the Forward or the Central Finite Difference Method.

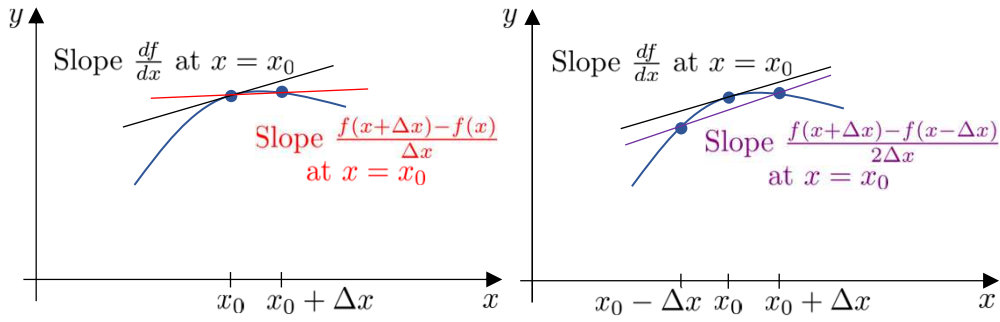


Figure 2.1: Graphical illustration of forward (in the left) and central (in the right) finite difference approximations for the derivative of the scalar-valued function f at the point $x = x_0$ (Adapted from [4]).

The great advantages of the Finite Differences Method are its simplicity of implementation and its applicability to a wide range of situations. However, it needs the evaluation of $\mathbf{f}(\mathbf{x})$ whenever called, which can result in a slow algorithm if the cost of evaluating this function is high, as happens with compliance in topology optimization.

Alternatives to the Finite Differences consist of analytical methods, such as the Direct Differentiation Method (DDM) or the Adjoint Method (AM). In the sequence, these two approaches are shown in the context of the Finite Element Analysis of a structural problem, that is described by:

$$\mathbf{K}(\boldsymbol{\rho})\mathbf{u}(\boldsymbol{\rho}) = \mathbf{F}(\boldsymbol{\rho}), \quad (2-5)$$

where \mathbf{F} is the external load vector, \mathbf{K} the stiffness matrix, \mathbf{u} is the displacements vector and let us suppose that these matrices are function of an arbitrary vector $\boldsymbol{\rho}$ of design variables. It is also possible to define in this context one constraints vector \mathbf{g} :

$$\mathbf{g}(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) = \begin{bmatrix} g_1(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \\ \vdots \\ g_m(\boldsymbol{\rho}, \mathbf{u}(\boldsymbol{\rho})) \end{bmatrix}, \quad (2-6)$$

with m representing the number of constraints in the problem.

Applying the chain rule to Equations (2-5) and (2-6) yields:

$$\mathbf{K}(\boldsymbol{\rho}) \frac{\partial \mathbf{u}}{\partial \rho_j} + \frac{\partial \mathbf{K}}{\partial \rho_j} \mathbf{u}(\boldsymbol{\rho}) = \frac{\partial \mathbf{F}}{\partial \rho_j}(\boldsymbol{\rho}) \quad (2-7)$$

$$\frac{dg_i}{d\rho_j} = \frac{\partial g_i}{\partial \rho_j} + \frac{\partial g_i}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\rho_j} \quad (2-8)$$

If one assumes that the external loads vector does not depend on $\boldsymbol{\rho}$, it is possible to write the expression for the derivatives of the displacements \mathbf{u} as function of the design variables ρ_j by the Direct Differentiation Method rearranging Equation (2-7):

$$\frac{d\mathbf{u}}{d\rho_j} = -\mathbf{K}(\boldsymbol{\rho})^{-1} \left(\frac{\partial \mathbf{K}}{\partial \rho_j} \mathbf{u}(\boldsymbol{\rho}) \right) \quad (2-9)$$

Inside Equation (2-9), it is possible to see that the term in the outer parenthesis plays the same role as the external load vector in Equation (2-5), and that is why it is commonly called in the literature as pseudo-load. It is also important to see that one linear system must be solved for each design variable in the studied model, which implies that this method is efficient only if the total number of design variables is small.

However, if the problem in study involves many design variables, but only a few constraints, it is possible to resort to the Adjoint Method. In this case, substituting Equation (2-9) into (2-8) yields:

$$\frac{dg_i}{d\rho_j} = \frac{\partial g_i}{\partial \rho_j} - \frac{\partial g_i}{\partial \mathbf{u}} \left(\mathbf{K}(\boldsymbol{\rho})^{-1} \frac{\partial \mathbf{K}(\boldsymbol{\rho})}{\partial \rho_j} \mathbf{u}(\boldsymbol{\rho}) \right) \quad (2-10)$$

Inside Equation (2-10) it is possible to define the vector $\boldsymbol{\lambda}$, where each entry λ_i is:

$$\lambda_i = \left(\frac{\partial g_i}{\partial \mathbf{u}} \mathbf{K}(\boldsymbol{\rho})^{-1} \right)^T, \quad (2-11)$$

which is known as Adjoint vector. From this result, one can see that to compute the vector $\boldsymbol{\lambda}$ one linear system must be solved for each constraint, and the result substituted into Equation 2-10 to calculate the final values of the derivative of interest:

$$\frac{dg_i}{d\rho_j} = \frac{\partial g_i}{\partial \rho_j} - \lambda_i^T \frac{\partial \mathbf{K}(\boldsymbol{\rho})}{\partial \rho_j} \mathbf{u}(\boldsymbol{\rho}). \quad (2-12)$$

From the results in Equations (2-9) and (2-11), it is evident that the Adjoint Method is more efficient when the optimization problem has more

variables than constraints and the Direct Method should be used when the opposite happens.

2.2

HHT- α Integration Method

As one of the subjects of this dissertation are structures subjected to dynamic loads, then it is necessary to discuss one method to solve the motion equation:

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{F}(t) \quad (2-13)$$

where \mathbf{M} represents the inertial mass matrix, \mathbf{K} the global stiffness matrix and \mathbf{C} the damping matrix of the structure. \mathbf{F} is the external load vector, while \mathbf{u} , $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are respectively the displacements vector and its first and second derivatives.

One possibility is to try to find an analytical solution for this equation, however this can be a difficult task. For this reason, numerical methods are commonly used to solve Equation (2-13). In the literature, it is possible to find some efficient and numerically stable algorithms, such as the Newmark- β and the HHT- α integration method [34, 35, 36].

The authors of PolyDyna [9] suggest the use of the HHT- α method (HHT are the initials of its authors, Hilber, Hughes, and Taylor) in topology optimization involving time-dependent loads. This method is a variation of the classical Newmark- β method [8], proposing a parameter α representing the time lag between damping, stiffness, and the external load vector, as:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{u}}_i + (1 - \alpha)\mathbf{C}\dot{\mathbf{u}}_i + \alpha\mathbf{C}\dot{\mathbf{u}}_{i-1} + (1 - \alpha)\mathbf{K}\mathbf{u}_i + \alpha\mathbf{K}\mathbf{u}_{i-1} = \\ (1 - \alpha)\mathbf{f}_i + \alpha\mathbf{f}_{i-1}, \quad i = 1, \dots, N_t, \end{aligned} \quad (2-14)$$

where i represents the i -th of the total of N_t time steps used in the time discretization.

According to Giraldo-Londoño and Paulino [9], to employ the relationship expressed in Equation (2-14) to solve (2-13), one should use the Newmark- β finite difference relationships:

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \Delta t \dot{\mathbf{u}}_{i-1} + \Delta t^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{u}}_{i-1} + \beta \ddot{\mathbf{u}}_i \right] \quad (2-15a)$$

$$\dot{\mathbf{u}}_i = \dot{\mathbf{u}}_{i-1} + \Delta t [(1 - \gamma) \ddot{\mathbf{u}}_{i-1} + \gamma \ddot{\mathbf{u}}_i] \quad (2-15b)$$

Replacing the set of Equations (2-15) into Equation (2-14), one obtains the motion equation in the residual form:

$$\mathbf{R}_i = \mathbf{M}_1 \ddot{\mathbf{u}}_i + \mathbf{M}_0 \ddot{\mathbf{u}}_{i-1} + \mathbf{C}_0 \dot{\mathbf{u}}_{i-1} + \mathbf{K} \mathbf{u}_{i-1} - (1 - \alpha) \mathbf{f}_i - \alpha \mathbf{f}_{i-1} = \mathbf{0}, \quad (2-16)$$

where the matrices \mathbf{M}_1 , \mathbf{M}_0 and \mathbf{C}_0 are given by the following expressions:

$$\mathbf{M}_1 = \mathbf{M} + (1 - \alpha) \gamma \Delta t \mathbf{C} + (1 - \alpha) \beta \Delta t^2 \mathbf{K} \quad (2-17a)$$

$$\mathbf{M}_0 = (1 - \alpha) (1 - \gamma) \Delta t \mathbf{C} + (1 - \alpha) \left(\frac{1}{2} - \beta \right) \Delta t^2 \mathbf{K} \quad (2-17b)$$

$$\mathbf{C}_0 = \mathbf{C} + (1 - \alpha) \Delta t \mathbf{K}. \quad (2-17c)$$

In this context, it is possible to compute the matrices \mathbf{M}_1 , \mathbf{M}_0 and \mathbf{C}_0 in Equation (2-17) and then use Equation (2-16) to calculate the i -th second derivative $\ddot{\mathbf{u}}$. Then, the set of Equations (2-15) allows the computation of the displacement \mathbf{u}_i and its derivative $\dot{\mathbf{u}}_i$ for the same time instant.

For the first time step, \mathbf{u}_0 and $\dot{\mathbf{u}}_0$ are usually the problem initial conditions, and the second derivative $\ddot{\mathbf{u}}_0$ is obtained from the motion equation: $\ddot{\mathbf{u}}_0 = \mathbf{M}^{-1} (\mathbf{f}_0 - \mathbf{C} \dot{\mathbf{u}}_0 - \mathbf{K} \mathbf{u}_0)$.

To guarantee the stability and accuracy of the HHT- α method, the parameters α , β and γ must satisfy [9]:

$$0 \leq \alpha \leq 1/3, \quad (2-18a)$$

$$\beta = (1 + \alpha)^2 / 4, \quad (2-18b)$$

$$\gamma = (1 + 2\alpha) / 2. \quad (2-18c)$$

For the reader interested in more details about finite element problems involving time-dependent loads, it is suggested to refer to the book by Clough and Penzien [34]. And for more details regarding the HHT- α method, the paper from Hilber *et al.* [36] is a good literature source.

2.3

Monte-Carlo Simulation

Another problem that is discussed in this work is the optimization of structures subjected to random loads. In this context, the Monte-Carlo

Simulation is a valuable tool to take into account the effects of load and parameter randomness in the final result of one optimization problem [37].

To illustrate the employability of this method, it is considered the cantilever beam shown in Figure 2.2. The intensity of the load represented by F acting in the upper edge follows one Normal (Gaussian) distribution, while its position along the edge follows a Uniform distribution between a and b . Moreover, Young's modulus of the material is represented by one Weibull distribution.

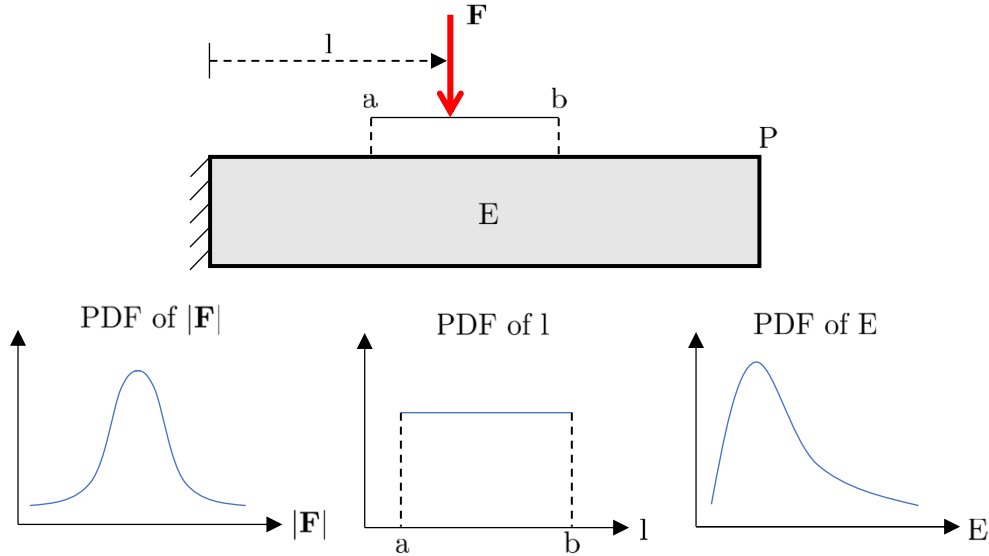


Figure 2.2: Cantilever beam subjected to uncertainties in load intensity, load position and material Young's modulus

Now imagine that a designer must determine the distribution of the displacements at the upper right vertex of this beam, represented by point P in Figure 2.2. The simplest approach, in this case, is to determine a set of admissible scenarios by generating samples of the random variables. For each scenario of the set, the displacement of interest is calculated by the beam theory [38] using samples of the random parameters. In the end, it is possible to build a histogram with the values calculated for the displacement at P, for example, which allows characterizing the distribution sought. Evidently, the quality of this characterization improves with the increase of the sample set size.

This brief example showed the main applicability of the Monte-Carlo method: characterize the distributions of the outputs from one mathematical model subjected to variations in its inputs (Figure 2.3). If a large number of samples of the input variables is used, the distributions of the outputs tend

to their real distributions, according to the Law of Large Numbers (LLN) [37, 39, 40].

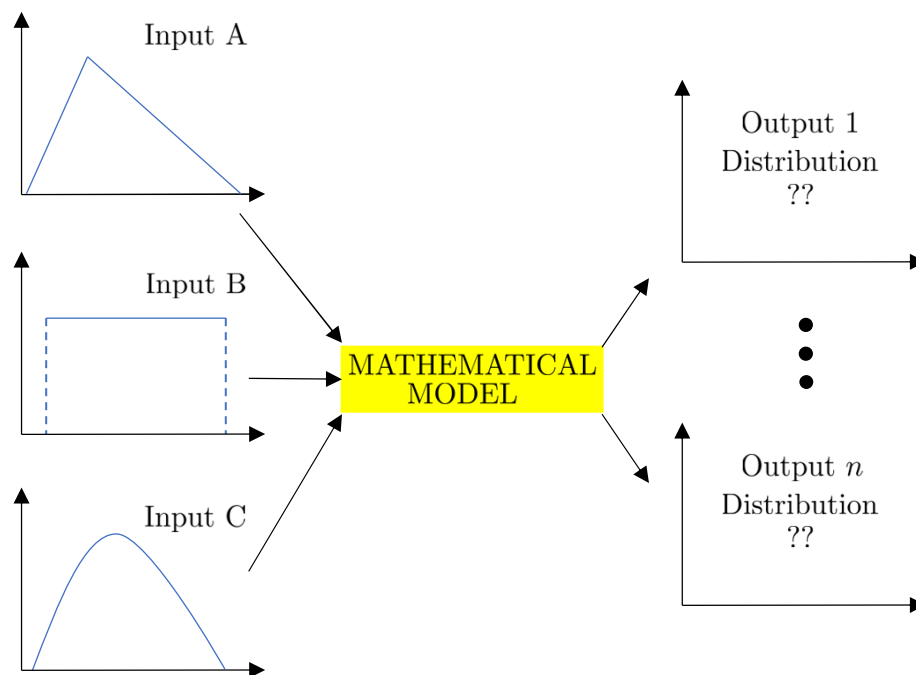


Figure 2.3: Example to illustrate the Monte Carlo method

3

Multiple Load Cases Topology Optimization

As presented in Chapter 1, this work focuses on the study of Topology Optimization involving time-dependent and random loads. However, it will be shown in what follows that these problems can be reduced to a multiple load topology optimization formulation. In this context, this chapter presents the basics of topology optimization, then extends the formulation for multiple loading scenarios.

When many loads are involved, the computational cost of solving the optimization problem becomes huge, as explained in Chapter 1. For this reason, two model reduction methods that exist in the literature are discussed, and one numerical example to show the applicability of both methods is presented. These two methods will be valuable tools when analysing time-dependent and robust problems in Chapters 4 and 5.

3.1

General Topology Optimization Problem

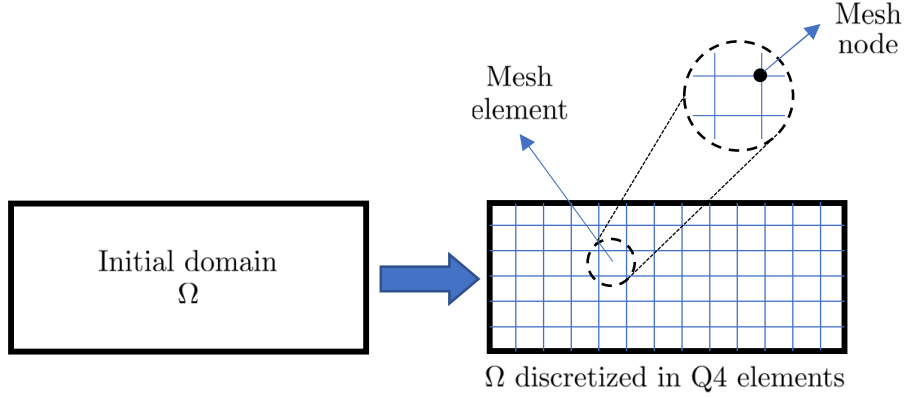
Since the basics of structural optimization were presented in Section 1.2, it is now possible to introduce the mathematical formulation of the topology optimization.

It must be remembered that topology optimization aims to propose a material distribution inside the given domain that minimizes the objective function of the problem and satisfies the imposed constraints [3]. In this context, the first step is to propose a mesh that discretizes the initial domain, as shown in Figure 3.1.

In this work, it is used the density approach, where one variable ρ_i is associated with each element of the mesh. ρ_i is considered a continuum variable, and for simplicity, its lower bound is set to $\rho_i = 0$, meaning that the element has no material, and the upper bound is $\rho_i = 1$, representing a solid element. With this concept, it is possible to define the density vector $\boldsymbol{\rho}$:

$$\boldsymbol{\rho} = [\rho_1 \ \rho_2 \ \dots \ \rho_m]^T, \quad (3-1)$$

where m is the number of elements in the mesh.

Figure 3.1: Discretization of the domain Ω

As a final result of the optimization process, it is suitable to have only 0's and 1's inside $\boldsymbol{\rho}$, *i.e.* the domain composed only of solid and void regions, without intermediate density values. To achieve this goal, the Solid Isotropic Method with Penalization (SIMP) [3] is used.

Throughout this work, the objective functions considered are related to the domain's compliance ϕ , that can be written as a function of the nodal loads acting on the structure \mathbf{f} and the associated nodal displacements \mathbf{u} as:

$$\phi(\boldsymbol{\rho}, \mathbf{u}) = \mathbf{f}^T \mathbf{u}(\boldsymbol{\rho}), \quad (3-2)$$

if the problem is subjected to one load case scenario only. Compliance minimization is the same to minimize the strain energy or find the material distribution that maximizes the structure stiffness, satisfying the imposed volume constraint.

The challenge for a robust method used for topology optimization is to propose a numerical scheme that makes the values of the density vector $\boldsymbol{\rho}$ converge to binary values. With the SIMP method, this goal is achieved by imposing one penalty parameter p to the densities inside $\boldsymbol{\rho}$, and defining the i -th element constitutive tensor \mathbf{D}_e used in the finite element computations as a function of its density ρ_i :

$$\mathbf{D}_e = \rho_i^p \mathbf{D}_e^0, \quad (3-3)$$

where \mathbf{D}_e^0 represents the constitutive tensor for the solid element, that can be written as:

$$\mathbf{D}_e^0 = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \quad (3-4)$$

if a plane stress condition is assumed, or:

$$\mathbf{D}_e^0 = \frac{E}{(1+\nu)(1-\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{bmatrix}, \quad (3-5)$$

if a plane strain condition is assumed for the problem.

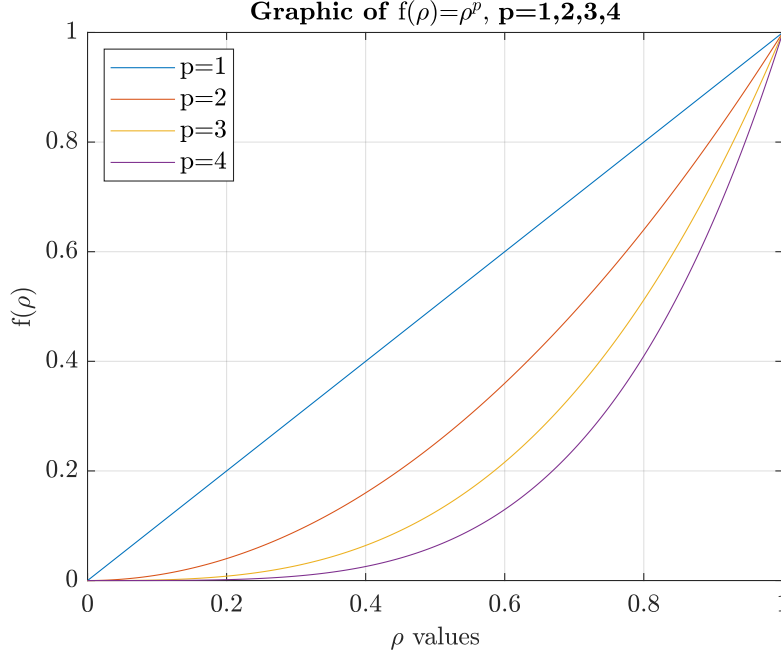


Figure 3.2: Graphics of $f(\rho) = \rho^p$ for different values of ρ and p

From the graphic shown in Figure 3.2, it can be noted that when the penalty parameter p is increased, the element stiffness is reduced for intermediate values of ρ_i . This results in higher values of the objective function in Equation (3-2), since less stiffness implies greater displacements for the same applied load \mathbf{f} . Consequently, the components of the density vector $\boldsymbol{\rho}$ converges to the binary values with the increase of p .

By the SIMP approach, it is now evident that finding the optimal topology means to determine the optimal densities vector $\boldsymbol{\rho}$ – the vector of design variables – that minimizes the objective function. The most common constraint g in these problems is a volume constraint, that can also be written as a function of the vector $\boldsymbol{\rho}$:

$$g(\boldsymbol{\rho}) = \sum_{e=1}^m \rho_e v_e \leq V_{\max}, \quad (3-6)$$

where m is the total number of elements in the mesh, v_e is the volume of the e -th element, and V_{\max} is the maximum allowable volume for the optimized structure.

Common topology optimization algorithms, such as PolyTop (presented in Section 3.2), propose one continuity scheme for the penalty factor p as illustrated in Figure 3.3. Initially, iterations of the optimization algorithm represented by the purple box are performed with the initial value $p = 1$, until achieving convergence of the values inside ρ or exhausting the maximum number of allowed iterations for this value of p , conditions that stops the optimization algorithm. Usually, the densities after this step still have a lot of intermediate densities, and for this reason, the value of p is augmented and a new cycle of iterations is performed, as shown in Figure 3.3. In 2D problems, to achieve good final results, typical values used for p are between 1 and 3, with increases of 0.5 between the cycles [6].

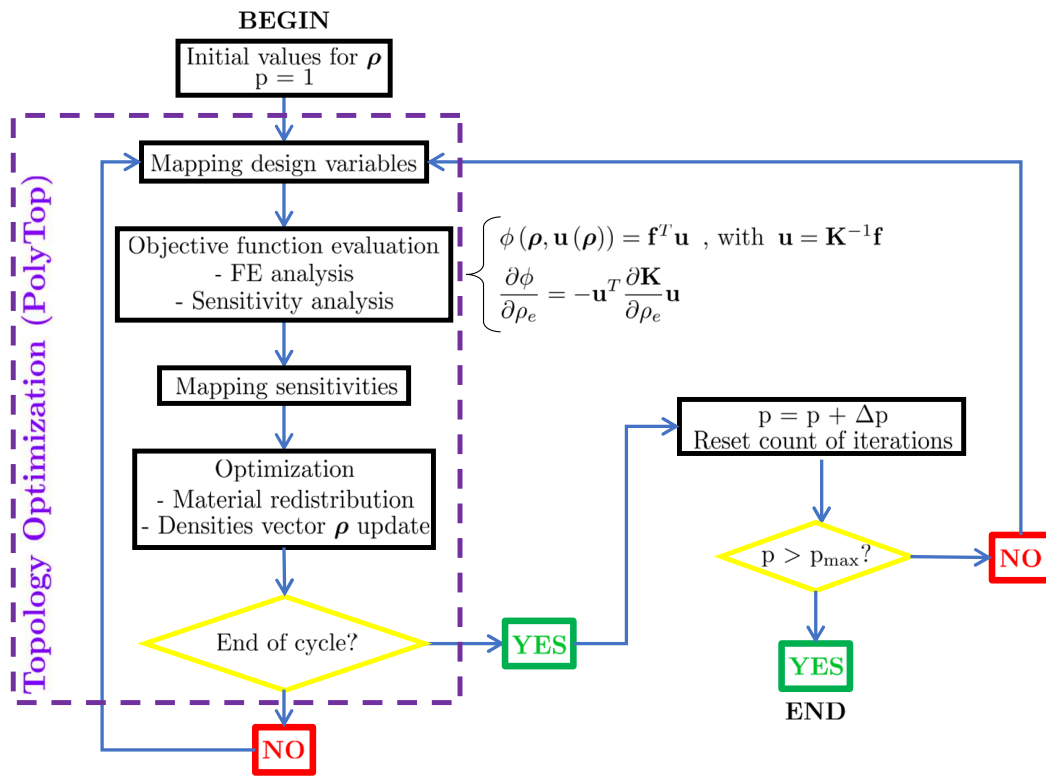


Figure 3.3: Working scheme of Topology Optimization based on SIMP method

3.2

PolyTop software

The PolyTop educational software introduced by Talischi *et al.* [22] is an efficient Matlab program used for structural compliance-based topology optimization. It is integrated with the PolyMesher polygonal mesh generator program [5] and is widely used in many published works.

To illustrate one application of the PolyMesher program, Figure 3.4 shows two meshes for a rectangular domain. The great advantages of polygonal

meshes is to avoid vertex-to-vertex intersections between the elements, which is very interesting in a topology optimization context [5].

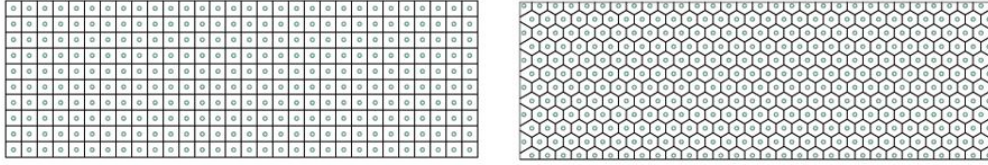


Figure 3.4: Regular and hexagonal meshes for one rectangular domain generated with PolyMesher [5].

The great advantage of **PolyTop** is its computational efficiency combined with the modular structure. It means that it is very easy for anyone to find specific sections of the code, for example, the one where the Finite Element problem is solved. Moreover, it also allows easy modifications into the code, such as the change of the update scheme, parameter values, and the objective function and its derivatives.

Due to this versatility, some evolutions of the original software were recently published, such as **PolyStress** [20] to insert stress constraints into the problem and **PolyDyna** [9] for topology optimization involving dynamic loadings.

The program also contributed to advances in different areas of engineering. For example, Martin and Deierlein [21] applied **PolyTop** to solve the topology optimization of buildings subjected to earthquake loadings, with restrictions related to the natural frequencies of the structure. Their work resulted in a new and very efficient method, which was a strong contribution to this field.

The first step to use the software is to set the domain, loading, and boundary conditions. Then, the user must enter the mesh details, and can choose between regular and polygonal meshes. To finish the setup, the parameters informations are entered. Since the program is run, it works exactly as described in Section 3.1 and illustrated by Figure 3.3.

The update scheme used in the software is the Optimally Criteria (OC) Method, ideal for the situation where the number of project variables is high, but the problem involves only one constraint and only negative derivatives. For more details about the Optimally Criteria method, it is suggested to look at Bendsøe and Sigmund [3].

3.3

Topology Optimization involving many load cases scenarios

As described in Chapter 1, real engineering structures while in service are not only subjected to one deterministic loading scenario, and it is suitable to take into account for topology optimization the largest possible quantity of loads that can act on it.

In Section 3.1 the topology optimization problem for a body subjected to one loading scenario only was presented. To extend this approach to include the effects of different loading scenarios, one first defines the i -th weighted load vector \mathbf{f}_i as:

$$\mathbf{f}_i = \alpha_i \tilde{\mathbf{f}}_i, \quad (3-7)$$

where $\tilde{\mathbf{f}}_i$ is the i -th load case acting on the structure, and α_i is its respective weighting factor. Then, it is possible to define the objective function for the multiple load cases topology optimization problem as [6]:

$$\phi(\boldsymbol{\rho}, \mathbf{u}) = \sum_{i=1}^n \mathbf{f}_i^T \mathbf{u}_i(\boldsymbol{\rho}). \quad (3-8)$$

Given the Equations (3-7) and (3-8), if $\alpha_i = 1/n$, $\phi(\boldsymbol{\rho}, \mathbf{u})$ represents the mean compliance, and if $\alpha_i = 1$, $\phi(\boldsymbol{\rho}, \mathbf{u})$ is the sum of the compliances of all loading scenarios.

Using Equation (3-8) as the objective function for the optimization problem means finding the optimal distribution of material that minimizes a function of the sum of the compliances of all loadings acting on the domain, satisfying the imposed constraints. With this concept, it is possible to write the mathematical formulation for the topology optimization problem involving many load cases scenarios as:

$$\left\{ \begin{array}{l} \min \quad \phi(\boldsymbol{\rho}, \mathbf{u}_i) = \sum_{i=1}^n \mathbf{f}_i^T \mathbf{u}_i(\boldsymbol{\rho}) \\ \text{s.t.:} \quad \sum_{e=1}^m \rho_e v_e - V_{max} \leq 0 \\ \quad \quad 0 < \rho_{min} \leq \rho_e \leq \rho_{max}, \quad e = 1, \dots, m \\ \text{with:} \quad \mathbf{K}(\boldsymbol{\rho}) \mathbf{u}_i(\boldsymbol{\rho}) = \mathbf{f}_i, \quad i = 1, \dots, n \end{array} \right., \quad (3-9)$$

where v_e and V_{max} represent the element and the maximum allowable volumes, respectively, and ρ_{min} and ρ_{max} the lower and upper bounds for each component ρ_i of the design variables vector $\boldsymbol{\rho}$, \mathbf{K} is the global stiffness matrix of the structure and m is the total number of elements in the finite element model mesh.

From Equation (3-9), it is evident that the simplest way to compute the objective function value along the steps of the optimization algorithm involves

the calculation of the displacements vector \mathbf{u}_i associated with each load \mathbf{f}_i and the values inside $\boldsymbol{\rho}$ at the respective optimization step by the Finite Element Method, and then multiply by vector \mathbf{f}_i to obtain the load cases compliances. However, if the mesh employed in the simulation is refined and the number of loading scenarios is high – *i.e.* the values of m and n in Equation (3-9) are high – the computational solution of the problem becomes extremely inefficient because one Finite Element problem must be solved for each load case and at each optimization step.

This way of solving the problem is called in this work as the standard approach, and in the following sections, two methods are presented to significantly reduce this computational cost: the SA and SVD methods.

3.4

SA-approach

The Stochastic Approach (SA) was introduced in 2017 by Zhang *et al.* [6], based on a probabilistic analysis to reduce the number of load cases to compute the objective function in topology optimization problems involving multiple load cases scenarios.

To describe the SA approach, the first step is to define the loading scenarios matrix \mathbf{F} and the associated displacements matrix \mathbf{U} . Inside \mathbf{F} , the i -th column represents the i -th load case scenario, and the nodal displacements \mathbf{u}_i associated to the i -th load \mathbf{f}_i are placed in the i -th column of \mathbf{U} :

$$\mathbf{F} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \vdots & \vdots & & \vdots \\ \mathbf{f}_1 & \mathbf{f}_2 & \dots & \mathbf{f}_n \\ \vdots & \vdots & & \vdots \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{\Delta \times n}, \quad (3-10a)$$

$$\mathbf{U} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \vdots & \vdots & & \vdots \\ \mathbf{u}_1(\mathbf{f}_1, \boldsymbol{\rho}) & \mathbf{u}_2(\mathbf{f}_2, \boldsymbol{\rho}) & \dots & \mathbf{u}_n(\mathbf{f}_n, \boldsymbol{\rho}) \\ \vdots & \vdots & & \vdots \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{\Delta \times n}. \quad (3-10b)$$

The product $\mathbf{F}^T \mathbf{U}$ can be expressed as:

$$\mathbf{F}^T \mathbf{U} = \begin{bmatrix} \mathbf{f}_1^T \mathbf{u}_1 & \dots & \mathbf{f}_1^T \mathbf{u}_n \\ \vdots & \ddots & \vdots \\ \mathbf{f}_n^T \mathbf{u}_1 & \dots & \mathbf{f}_n^T \mathbf{u}_n \end{bmatrix}_{n \times n}. \quad (3-11)$$

From the result in Equation (3-11), it is possible to see that the trace of $\mathbf{F}^T \mathbf{U}$ is exactly the value of the compliance $\phi(\boldsymbol{\rho}, \mathbf{u})$ expressed in Equation (3-8). In this fashion, it is equivalent to calculate the objective function by the sum expressed in Equation (3-8) or by the trace in Equation (3-11).

Hutchinson [41] introduces an stochastic estimator for the trace of a square matrix \mathbf{A} :

$$\mathbb{E}(\boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\xi}) = \text{tr}(\mathbf{A}), \quad (3-12)$$

where $\boldsymbol{\xi}$ is a discrete random vector of entries $+1$ or -1 , each value with the same probability of occurrence. Thus, it is possible to use the Sample Average Approximation (SAA) [42, 43] to estimate the value of the trace of \mathbf{A} from a set of realizations $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_{n_s}$ of the random vector $\boldsymbol{\xi}$:

$$\mathbb{E}_S(\boldsymbol{\xi}^T \mathbf{A} \boldsymbol{\xi}) = \frac{1}{n_s} \sum_{k=1}^{n_s} \boldsymbol{\xi}_k^T \mathbf{A} \boldsymbol{\xi}_k \xrightarrow{n_s \rightarrow \infty} \text{tr}(\mathbf{A}). \quad (3-13)$$

As $\mathbf{F}^T \mathbf{U}$ is a square matrix, it is possible to use the result in Equation (3-13) to calculate its trace:

$$\text{tr}(\mathbf{F}^T \mathbf{U}) = \mathbb{E}_S(\boldsymbol{\xi}^T \mathbf{F}^T \mathbf{U} \boldsymbol{\xi}). \quad (3-14)$$

Since $\mathbf{U} = \mathbf{K}^{-1} \mathbf{F}$, where \mathbf{K} represents the global stiffness matrix of the studied body, it is possible to rewrite the objective function in Equation (3-8) using the Hutchinson trace estimator as:

$$\mathbb{E}(\boldsymbol{\xi}^T \mathbf{F}^T \mathbf{U} \boldsymbol{\xi}) = \mathbb{E}(\boldsymbol{\xi}^T \mathbf{F}^T \mathbf{K}^{-1} \mathbf{F} \boldsymbol{\xi}) = \mathbb{E}[(\mathbf{F} \boldsymbol{\xi})^T \mathbf{K}^{-1} (\mathbf{F} \boldsymbol{\xi})] = \phi(\boldsymbol{\rho}, \mathbf{u}). \quad (3-15)$$

The same procedure can be applied to the problem sensitivities obtained by the adjoint method, which yields [3, 6]:

$$\frac{\partial \phi}{\partial \rho_e} = -\text{tr} \left(\mathbf{F}^T \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{K}^{-1} \mathbf{F} \right) = -\mathbb{E} \left[(\mathbf{F} \boldsymbol{\xi})^T \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{K}^{-1} (\mathbf{F} \boldsymbol{\xi}) \right]. \quad (3-16)$$

By this approach, the objective function and its sensitivities are estimated

by:

$$\hat{\phi}(\boldsymbol{\rho}, \mathbf{u}) = \frac{1}{n_s} \sum_{k=1}^{n_s} (\mathbf{F}\boldsymbol{\xi}_k)^T \mathbf{K}(\boldsymbol{\rho})^{-1} (\mathbf{F}\boldsymbol{\xi}_k) \quad (3-17a)$$

$$\frac{\partial \hat{\phi}}{\partial \rho_e}(\boldsymbol{\rho}, \mathbf{u}) = -\frac{1}{n_s} \sum_{k=1}^{n_s} (\mathbf{F}\boldsymbol{\xi}_k)^T \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{K}^{-1} (\mathbf{F}\boldsymbol{\xi}_k) \quad (3-17b)$$

The great advantage of the SA-approach according to its authors is that a number of samples $n_s = 6$ of the random vector $\boldsymbol{\xi}$ is enough to obtain good estimates for the objective function and its sensitivities in Equations (3-17). Consequently, the number of Finite Element problems to be solved at each step of the optimization algorithm is reduced from the total number of load cases in the problem to only 6 ($n_s = 6$), which apports an enormous efficiency to the algorithms [6].

But according to the authors of this approach, the common convergence criterium used in optimization algorithms – *i.e.* the modulus of the difference of the values between two consecutive steps smaller than a tolerance value – is not sufficient. From the studies performed during this work, it was possible to notice that the Optimally Criteria updates work well for the first iterations for each value of the penalty parameter p , but then they start to become ineffective. To solve this situation, it is proposed by Zhang *et al.* [6] the use of an empirical damping scheme, that compares the actual step size with the design variable values n_{steps} before. This leads to the definition of the effective step ratio R:

$$R = \frac{\frac{1}{n_{\text{steps}}} \left\| \left(\boldsymbol{\rho}_k - \boldsymbol{\rho}_{k-n_{\text{steps}}+1} \right) \right\|}{\left\| \boldsymbol{\rho}_k - \boldsymbol{\rho}_{k-1} \right\|}, \quad (3-18)$$

where $\boldsymbol{\rho}_k$ represents the densities vector values at the actual optimization step. Once the R-value is below a prescribed tolerance τ_{step} , the move limit of the OC method should be reduced by a scale factor γ .

Figure 3.5 from Zhang *et al.* [6] illustrates the importance of the damping scheme in the SA approach. This situation involves the optimization of an objective function composed of two variables, which allows the analysis of their level curve plots. In plot (a), it is compared the optimization history by the SA approach without using the damping scheme (in red) with the analytical solution (in green). It is evident that when the variables approach the optimal value, the method starts to become ineffective. Although, when the damping scheme is used, this problem is reduced, as shown in the graphic of Figure 3.5b.

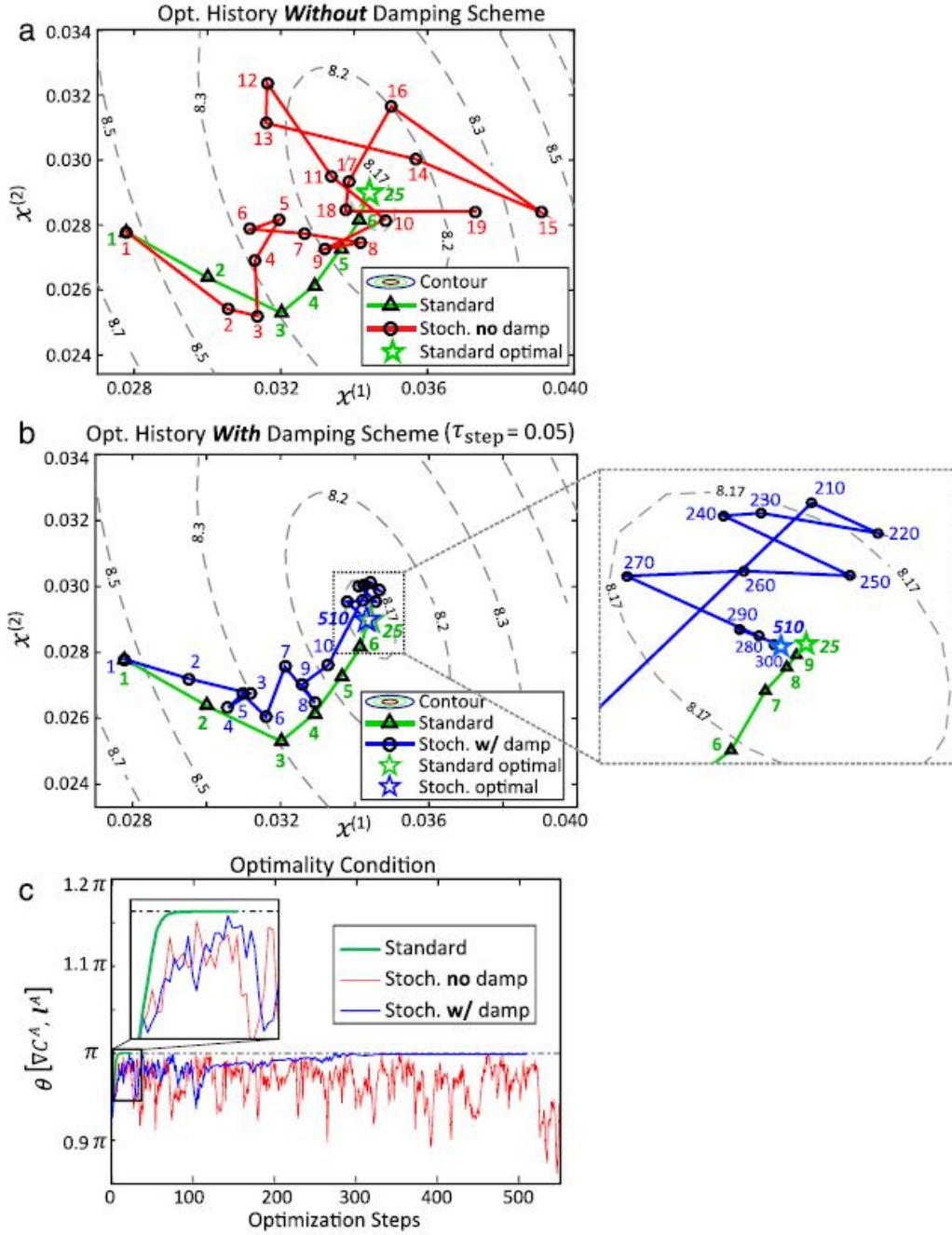


Figure 3.5: Comparison of the optimization history of a two-parameters objective function with and without the proposed damping scheme [6]

The suggested values in Zhang *et al.* [6] for the parameters inside Equation (3-18) are $R = 0.1$ and $n_{\text{steps}} = 10$, and the parameters to adjust the move limit are $\tau_{\text{step}} = 0.1$ and $\gamma = 2$. It is also suggested to not reduce the value of the move limit for the first 100 iterations of the update scheme.

In Figure 3.6 it is illustrated the working scheme of the SA-based topology optimization. At this point, the reader is invited to compare this scheme with the one for the standard approach, shown in Figure 3.3, and verify that one sampling of the random vector ξ is made at the beginning of each optimization

iteration, and the same topology optimization algorithm is used, but employing the associated loads $\mathbf{F}\xi$ instead of all the load cases inside \mathbf{F} when the SA-based approach is used.

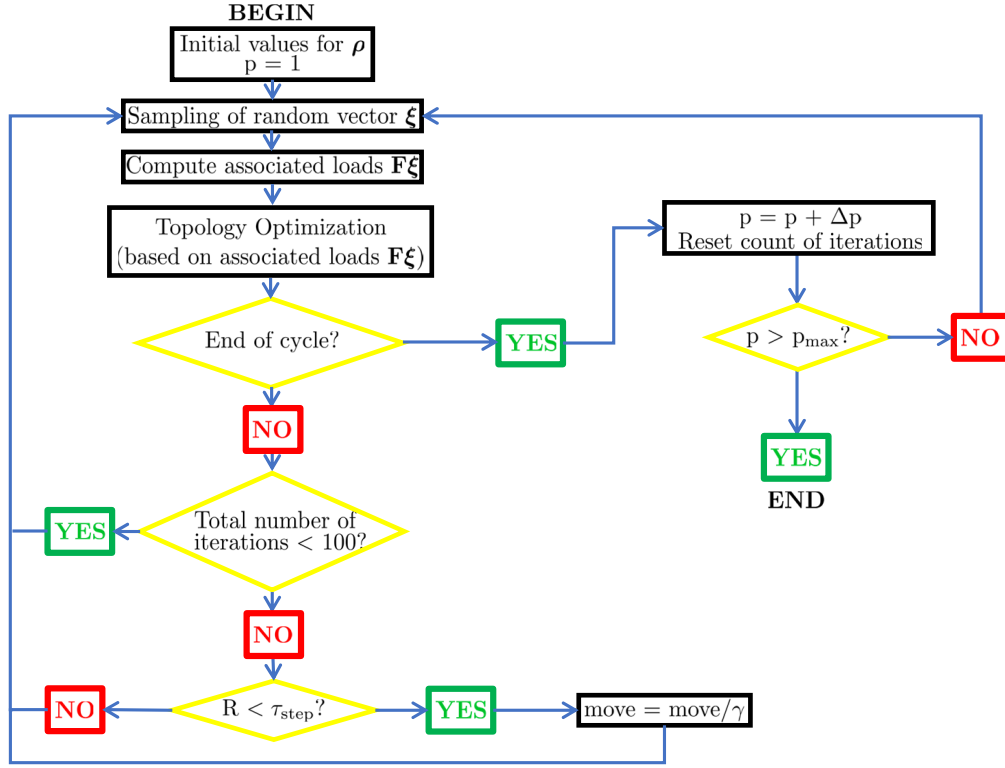


Figure 3.6: Working scheme for Topology Optimization employing the SA-approach

3.5 SVD-approach

Another alternative for the standard approach is based on the Singular Values Decomposition (SVD) of the matrix that gathers the loading scenarios defined in Equation (3-10a), which was introduced by Tarek and Ray in 2021 [11].

To introduce this approach, it is remembered that the SVD decomposition for any real-valued rectangular matrix \mathbf{A} of dimensions $p \times q$ can be written as [44, 11]:

$$\mathbf{A}_{p \times q} = \mathbf{U}_{p \times q} \mathbf{\Sigma}_{q \times q} \mathbf{V}_{q \times q}^T, \quad (3-19)$$

which is known as the simplified form of the SVD decomposition. In this formulation, \mathbf{U} represents the left and \mathbf{V} the right singular vectors matrix, and $\mathbf{\Sigma}$ is a diagonal matrix that gathers the q first singular vectors of \mathbf{A} . It is important to emphasize that in this work the notation \mathbf{U} is adopted for the left

singular vectors matrix instead of \mathbf{U} as found in literature, to avoid confusion with the matrix \mathbf{U} defined in Equation (3-10b).

The columns inside \mathbf{U} are the q -first eigenvectors of $\mathbf{A}^T \mathbf{A}$, while the singular vectors of \mathbf{A} that are gathered in $\mathbf{\Sigma}$ are the square root of the first q eigenvalues of $\mathbf{A}^T \mathbf{A}$.

As it was shown that the objective function of the compliance-based topology optimization can be expressed by the trace of a square matrix in Equation (3-11), it is now possible to re-write it using the definition given by Equation (3-19):

$$tr(\mathbf{F}^T \mathbf{U}) = tr(\mathbf{F}^T \mathbf{K}^{-1} \mathbf{F}) = tr(\mathbf{V} \mathbf{\Sigma} \mathbf{U}^T \mathbf{K}^{-1} \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T) = tr(\mathbf{\Sigma} \mathbf{U}^T \mathbf{K}^{-1} \mathbf{U} \mathbf{\Sigma}), \quad (3-20)$$

using the cyclic property of the trace and the fact that $\mathbf{\Sigma}^T = \mathbf{\Sigma}$, as $\mathbf{\Sigma}$ is a diagonal matrix.

It is known that if the loading scenarios have only a few degrees of freedom loaded, or if the scenarios are highly correlated, the matrix \mathbf{F} has only a few dominant singular vectors [11]. This affirmation means that only a few of all singular vectors have their numerical value much greater than the others.

If the result from Equation (3-20) is rewritten in the form:

$$tr(\mathbf{\Sigma} \mathbf{U}^T \mathbf{K}^{-1} \mathbf{U} \mathbf{\Sigma}) = \sum_{i=1}^n \sigma_{ii}^2 \mathbf{u}_i^T \mathbf{K}^{-1} \mathbf{u}_i, \quad (3-21)$$

where \mathbf{u}_i represents the i -th column of $\mathbf{U} \mathbf{\Sigma}$ and σ_{ii} the i -th element on the diagonal of $\mathbf{\Sigma}$, it is clear that it is possible to neglect the terms in the sum associated to the non-dominant singular values. Consequently, the number of Finite Element problems to be solved to evaluate the objective function in a step of the optimization algorithm is reduced from the number of load cases n to the number of dominant singular values of \mathbf{F} .

Compared to the SA approach, the method employing the SVD decomposition is much easier to implement, but the great reduction in the number of Finite Elements solves to evaluate the objective function only happens if the number of dominant singular values is small, which is not guaranteed for a general problem. Another important point is that the SVD decomposition of the matrix \mathbf{F} , which is an expensive operation from the computational point of view, must be calculated only once as the load scenarios do not change along with the steps of the optimization.

Figure 3.7 illustrates the working scheme for SVD-based topology optimization, and it is suggested for the reader to compare it with Figures 3.6 and 3.3 for the SA-based and the standard approaches, respectively.

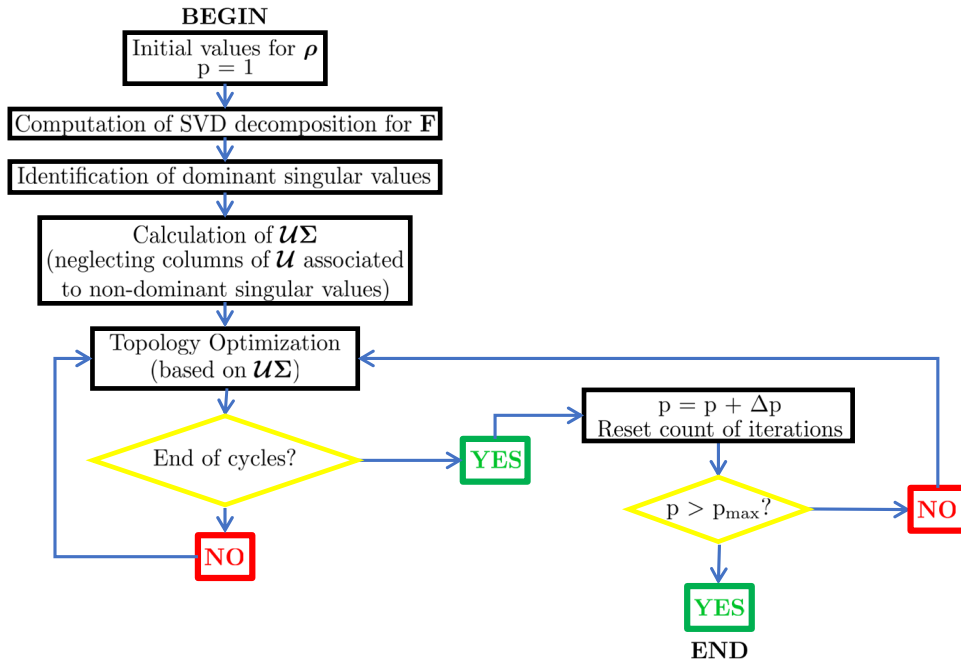


Figure 3.7: Working scheme for Topology Optimization employing the SVD-approach

3.6

Numerical example

This first example aims to compare the final topologies obtained by employing the three discussed approaches along this chapter. For that, it is considered the two dimensional box domain proposed in Zhang *et al.* [6] and schematized in Figure 3.8.

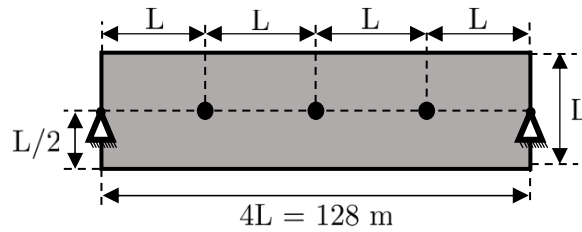


Figure 3.8: 2D box domain (Adapted from [6])

Table 3.1 gathers the physical and geometrical properties of the domain presented in Figure 3.8.

Table 3.1: Physical and geometrical properties of the 2D box domain

Parameters	Symbol	Value	Unit
Young's modulus	E	2×10^{11}	Pa
Poisson's ratio	ν	0.33	—
Domain thickness	t	0.01	m

For this analysis, it is supposed that this domain is subjected to 108 independent loads of magnitude $|\mathbf{f}_i| = 500$ N acting on the three highlighted nodes in Figure 3.8 and distributed as shown in Figure 3.9.

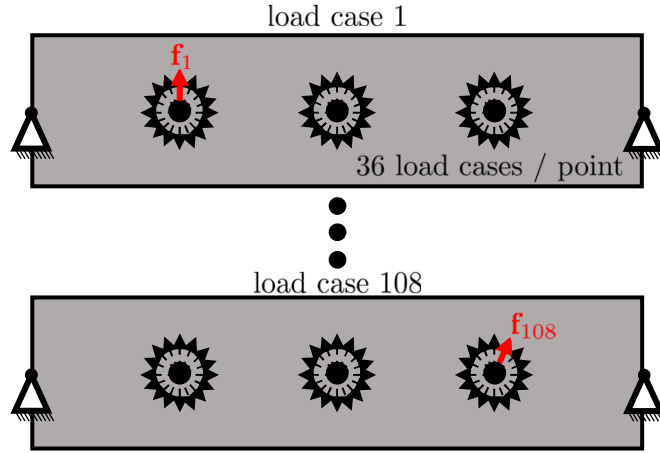


Figure 3.9: 108 independent load cases acting on the 2D box domain (Adapted from [6])

In this case, where all loads acting on the structure are known, 108 Finite Element solves are necessary to evaluate the value of the objective function if the standard approach is employed. But using the SA approach with the suggested number of samples for the Rademacher random vector discussed in Section 3.4, this number is reduced to 6.

In this case, the mesh used is composed of 320×80 equally distributed Q4 elements, the filter radius inside the PolyTop implementation is set as $R = 0.90$ m and the volume constraint is 30 % of the domain volume. The continuation of the penalty parameter is run from $p = 1$ to 3 in steps of 0.5, with a maximum number of 500 iterations for each value. The optimization tolerance is $\tau_{opt} = 10^{-2}$ and the parameters values for the damping scheme in the SA-approach are the same suggested in Zhang *et al.* [6] and recapitulated in Section 3.4.

When calculating the singular vectors of matrix \mathbf{F} , one finds the singular values gathered in Table 3.2. This result shows that only 6 singular values have their numerical values much greater than the others singular values, consequently only 6 columns of the left singular values matrix \mathbf{U} should be considered for the computations, leading to 6 necessary solves to evaluate the objective function value.

Figure 3.10 shows the final topologies obtained by employing the three discussed forms to solve the topology optimization problem, and also compares with the topologies obtained in Zhang *et al.* [6] for the standard and SA-based results, and in Tables 3.3 and 3.4 it is established a comparison between the

Table 3.2: First 10 singular values in the 108 independent load cases problem

#	Value	#	Value
1	204.1241	6	204.1241
2	204.1241	7	1.0469×10^{-14}
3	204.1241	8	1.0469×10^{-14}
4	204.1241	9	5.4117×10^{-15}
5	204.1241	10	5.4117×10^{-15}

compliance optimized value obtained by the three methods. Unfortunately, Zhang *et al.* [6] does not mention some parameter values that are necessary for the numerical simulations, and for this reason, the results in Tables 3.3 and 3.4 are not compared with the literature, and some differences are expected in the final topologies.

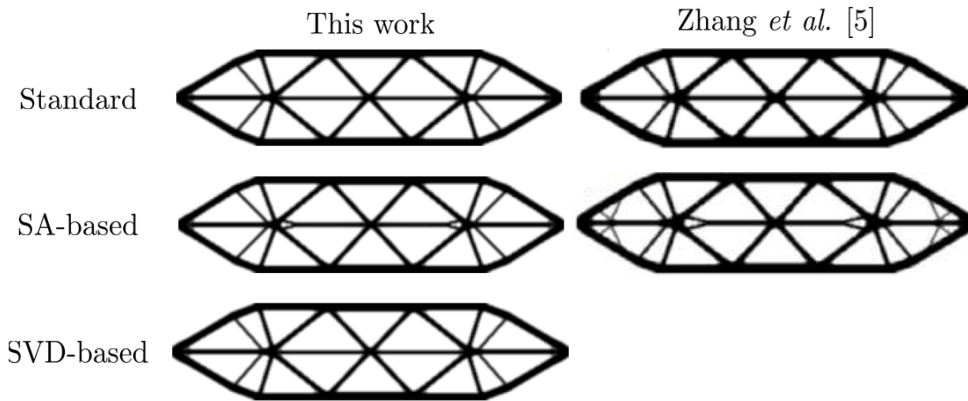


Figure 3.10: Final topologies obtained for the 108 load cases scenarios example and comparison with literature results

Table 3.3: Comparison between the simulations for the 108 load cases scenarios example

Method	n	$C(\mathbf{u}(\boldsymbol{\rho}))$	Difference
Standard	108	4.74×10^{-3}	—
SA-approach*	6	4.84×10^{-3}	2.11%
SVD-approach	6	4.74×10^{-3}	0.00%

* Results are averaged over 5 trials

Another interesting result is shown in Figure 3.11, where it is shown the behavior of the objective function values along with the iterations of the optimization algorithm. It is possible to notice that the values of the standard and SVD approaches are very close, while for the SA-based one can observe some fluctuations in the compliance values, associated with the random variable involved in the method.

Table 3.4: Total number of Optimally Criteria updates and Finite Element solves for each of the methods

Method	OC iterations	Total solves	Reduction
Standard	1192	128736	–
SA-approach*	1875	11250	91.3%
SVD-approach	1192	7152	94.4%

* Results are averaged over 5 trials

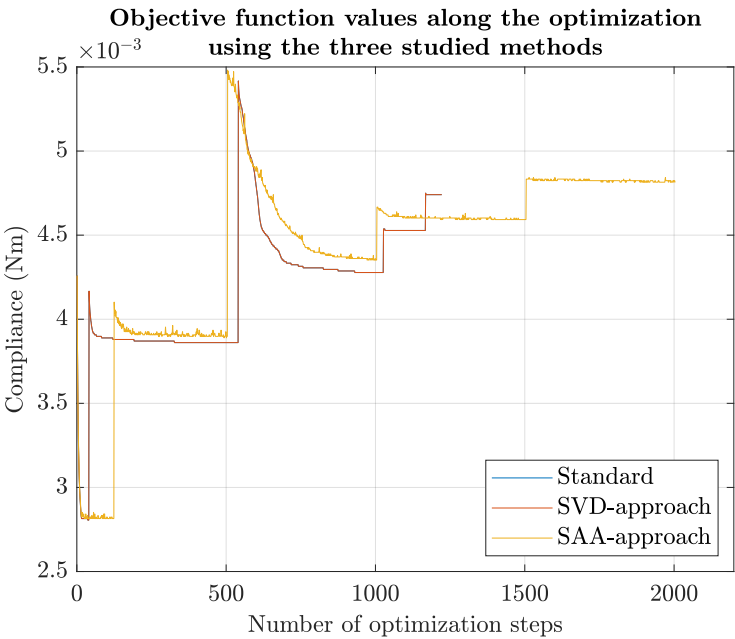


Figure 3.11: Values of the objective function along the optimization iterations for the three studied methods in the example of a box domain under 108 load cases

4

Time-Dependent Topology Optimization

In this chapter, it is discussed how to perform topology optimization when the loads are time-dependent and the dynamic effects should be taken into account in the final design. In this work, the dynamic compliance-based approach [9, 8] is employed, where the objective function for the optimization is given by:

$$\phi(t, \boldsymbol{\rho}, \mathbf{u}(t, \boldsymbol{\rho})) = \int_{t_i}^{t_f} \mathbf{f}(t)^T \mathbf{u}(t, \boldsymbol{\rho}) dt \quad (4-1)$$

where $\mathbf{f}(t)$ is the function that represents the nodal loads along time, and $\mathbf{u}(t, \boldsymbol{\rho})$ represents the nodal displacements along time that are solutions of the motion equation:

$$\mathbf{K}(\boldsymbol{\rho})\mathbf{u}(t, \boldsymbol{\rho}) + \mathbf{C}(\boldsymbol{\rho})\dot{\mathbf{u}}(t, \boldsymbol{\rho}) + \mathbf{M}(\boldsymbol{\rho})\ddot{\mathbf{u}}(t, \boldsymbol{\rho}) = \mathbf{f}(t) \quad (4-2)$$

with $\mathbf{K}(\boldsymbol{\rho})$, $\mathbf{C}(\boldsymbol{\rho})$ and $\mathbf{M}(\boldsymbol{\rho})$ representing, respectively, the stiffness, damping and mass matrix of the studied structure. The values t_i and t_f in Equation (4-1) are the initial and final time instants for dynamic compliance analysis.

More recently, an adaptation of the educational software **PolyTop** [22] was proposed by Giraldo-Londoño and Paulino [9], allowing the users to implement time-dependent loading conditions and maintaining the modular structure of the original program, which is important for learning, teaching and performing improvements and modifications into the code.

In this context, some modifications to the **PolyDyna** software are implemented in this work, based on the Equivalent Static Loads (ESL) approach from Kang *et al.* [27] to improve its computational performance.

First, a brief presentation of the **PolyDyna** software and the ESL approach is given, then it is explained in details how to implement these methods into the original software. At the end of the chapter, three numerical examples are presented, where the results obtained with and without the ESL approach are compared.

4.1

Introduction to PolyDyna software

The PolyDyna software [9] was proposed by Giraldo Londoño and Paulino in 2020 adding a series of new routines into the PolyTop software developed by Talischi *et al.* [22]. Its authors realized that the biggest part of real engineering structures are subjected to dynamic loads, while the original PolyTop software is limited to static loads.

In this context, due to the modular framework from PolyTop, the authors of PolyDyna could update a series of the routines from the original code without having to make major changes in the structure of the program, to make possible the treatment of time-dependent loads. For example, the finite element analysis inside PolyDyna uses the HHT- α method to compute the displacements, in the place of a more simple analysis in PolyTop for static loads.

According to the authors, PolyDyna allows users to implement dynamic topology optimization, and the loads can change their magnitude, direction, or location over a given period of time. This represents a big leap from the original PolyTop's capabilities.

4.2

PolyDyna's implementation

Similar to its parent PolyTop, the input data is filled in a file called PolyScript. Inside this file, all data associated with the parameters of the simulations must be set and the mesh, geometry, and loading characteristics are defined with the help of the domain functions as in PolyTop. The difference now is that the load is a function of the time, and along with its definition, the user is required to provide the time step that will be used inside the dynamic analysis.

Then, the problem works exactly as in the scheme presented in Figure 3.3: at each optimization step, the update scheme computes new values for the element densities, and this process is repeated until convergence or achieve the maximum number of allowed iterations.

The software is composed of eight principal Matlab files, with the names and their main functions listed in Table 4.1.

Observing the PolyDyna implementation, it is possible to notice that at each call of the update scheme, one dynamic analysis is performed, which brings an enormous computational cost to the software. In this context, the interest of this work here is to establish an alternative approach to this fact

Table 4.1: Main Matlab files of PolyDyna software (Adapted from [9])

File	Description
PolyScript	Input data required for running PolyDyna
MatIntFnc	Computes the element Young's modulus and densities
PolyFilter	Computes the filter matrix of the problem
PreComputations	Computes the element stiffness matrix and load vector
FEM_Dyna	Solves the structural dynamics problem
AdjointProblem	Solves the sensitivity analysis
Centroids	Computes the centroids of all elements
Areas	Computes the areas of all elements

to reduce the cost involved. For this reason, the ESL approach is presented in the sequence, and then its implementation in the PolyDyna software.

4.3

The Equivalent Static Loads (ESL) method

The Equivalent Static Loads (ESL) method was introduced in the context of structural optimization in the late 1990s by Kang *et al.* [27], initially for beam and trusses sizing optimization. The same research group published two years later in Choi and Park [7] a more simple explanation of the theory behind the method and more interesting ESL applications examples. In both cases, the motivation of the papers was the computational cost involved when it is necessary to solve the dynamics at each iteration of the optimization algorithm, and the authors realized that with their approach, it can be significantly reduced without compromising the final result.

The scheme in Figure 4.1 describes the procedure for obtaining the Equivalent Static Loads.

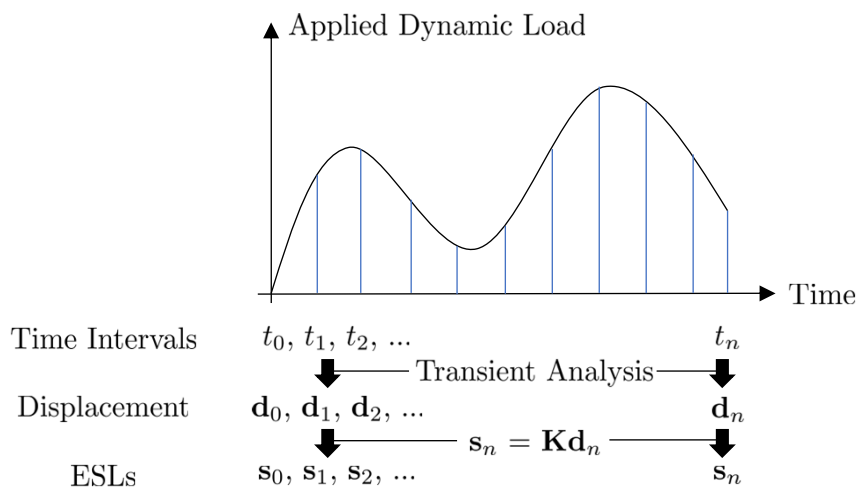


Figure 4.1: Illustration of the ESL method (Adapted from [7])

In text form, the procedure in Figure 4.1 can be expressed as follows:

- Propose one time discretization $t_0, t_1, t_2, \dots, t_n$ for the time in n intervals;
- For each time step, compute the displacements vector \mathbf{d}_i associated to each of the n time intervals. This calculation uses the HHT- α method shown in Section 2.2;
- For each \mathbf{d}_i , compute the static load \mathbf{s}_i that provoke the same displacement calculated by the dynamic analysis, multiplying \mathbf{d}_i by the global stiffness matrix $\mathbf{K}(\boldsymbol{\rho})$;
- The set of n loads \mathbf{s}_i are the ESLs for the problem.

This procedure allows the definition of a set of static loads that produces the same displacements at all times as the dynamic response analysis. In this context, this set can be used as multiple loading scenarios for topology optimization if one re-writes Equation (4-1) in the form:

$$\int_{t_i}^{t_f} \mathbf{f}(t)^T \mathbf{u}(t, \boldsymbol{\rho}) dt \approx \sum_{i=1}^n \omega_i \mathbf{s}_i^T \mathbf{K}^{-1}(\boldsymbol{\rho}) \mathbf{s}_i, \quad (4-3)$$

where \mathbf{s}_i are the ESLs associated to the nodal loads $\mathbf{f}(t)$ and ω_i are numerical integration weights.

The reasoning represented by Equation (4-3) was already used in the literature by Jang *et. al* [28] and Lavôr and Pereira [8], with the difference that the first article updates the ESLs at each step of the optimization, while the second proposes the execution of ESL cycles as shown in the flowchart of Figure 4.2, reducing even more the cost associated to the dynamic analysis.

There is a discussion in the literature if the solution from the ESL method satisfies the KKT optimal conditions. In 2003, Park and Kang [45] published one brief note where, according to the authors, it is possible to demonstrate that the KKT conditions are satisfied by the solution using the ESL method. However, in 2014 Stolpe [46] shows that the proof from Park and Kang was incomplete, and in 2018 Stolpe *et al.* [47] published another paper with a concrete example showing that the KKT conditions of the ESL solution violate the optimum conditions.

In 2019, Park and Lee [48] recognized the errors pointed out by Stolpe [46] but realized that the correction proposed in the article brings an enormous computational cost when calculating the sensitivity analysis, which compromises the main objective of ESL that is accelerating the structural optimization algorithms.

In practice, during the simulations performed for this work, it was possible to notice that the optimal solutions diverge when pulse-shaped loads

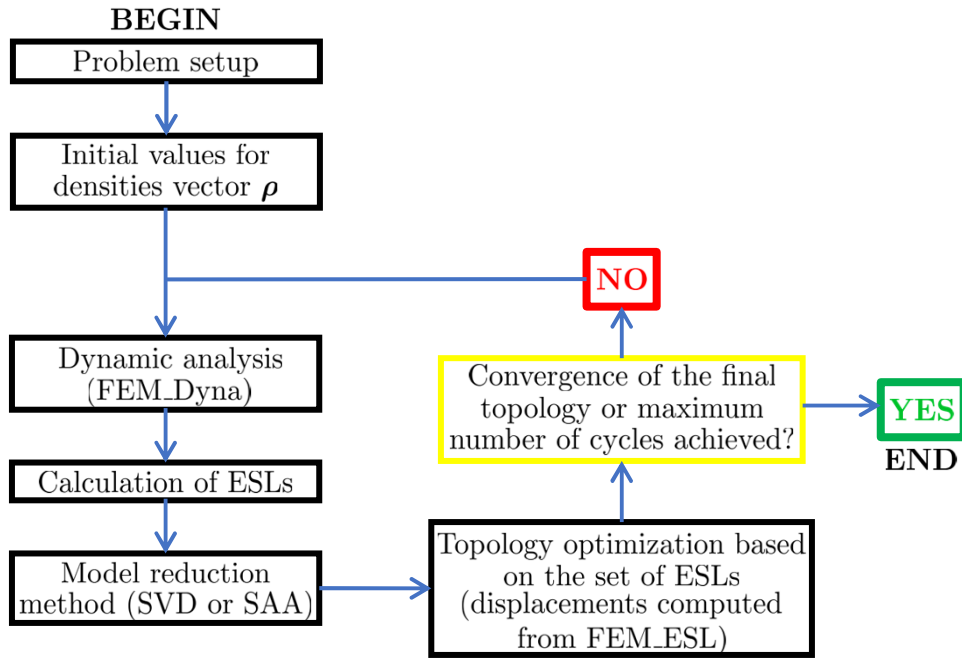


Figure 4.2: Flowchart for time-dependent topology optimization process using the ESL method (Adapted from [8]).

are involved. This will be better explained at the end of the chapter by numerical examples.

4.4

Implementation of ESL-approach into PolyDyna software

Now, it is explained how to implement the ESL approach into the PolyDyna software based on the cyclic procedure from Lavor and Pereira [8] represented in Figure 4.2. As this new approach involves multiple static loading scenarios, the model reduction methods presented in Sections 3.4 and 3.5 are also implemented into the original program to save computational time.

The first important step is to observe Table 4.1. As the interpolation and filtering schemes and the optimization setup are not changed, the files `MatIntFnc`, `PolyFilter`, `PreComputatons`, `Centroids`, and `Areas` do not need any modifications.

As the studied ESL approach still needs to solve the structure dynamics once for each cycle, the `FEM_Dyna` script must be maintained. But as the loads are now static for the topology optimization, one needs to provide a script for static finite element computation. This script called `FEM_ESL` is exactly the finite element routine from the original paper `PolyTop` with the use of Cholesky decomposition as in `FEM_Dyna`, and is reproduced in Appendix B for the interested readers.

Once the finite element routine was changed for the new implementation,

inside `PolyDyna` the function `FEM_ESL` must be called in the place of `FEM_Dyna`.

But the most important modification is in the `PolyScript` file, where `PolyDyna` is called. After the problem setup, it is necessary to implement the cyclic setup represented in Figure 4.2. For that, once a modified `PolyDyna` cycle is initiated, it is necessary to compute the nodal displacements along time using `FEM_Dyna`, and then calculate the equivalent static loads by multiplying these displacements by the global stiffness matrix of the structure. In the sequence, the dominant singular values and vectors of the ESLs should be computed, to apply the model reduction methods to the problem if the SVD approach is used. After that, cycles of the modified `PolyDyna` code are executed following the interpolation scheme defined by the user, and at the end of this step, a new set of equivalent static loads are defined, based on the final topology of the previous cycle. This process is repeated until convergence of the topology or achieving the maximum number of cycles allowed.

During the numerical simulations performed in this work, it was possible to see the need to implement one low-densities filter at the end of each optimization cycle. This filter sets the density value equal to zero for all elements that have at the end of the cycle the density value below a prescribed value. Without this filter, numerical instabilities are observed as loads are applied to elements with low densities, which results in high compliance values and final topologies with a lot of intermediate densities.

For the interested reader, in Appendix C it is reproduced the lines of the `PolyScript` file that represents flowchart in Figure 4.2.

4.5

Case studies

In this section, the three numerical examples from the `PolyDyna` article [9] that involve the minimization of the dynamic compliance are reproduced. For each case, the original and the modified codes are run, and the final topologies are compared.

The two first examples involve simple geometries and regular Q4 meshes and have the objective to show the limitations associated with the ESL method when pulse-shaped loads are applied. The third example is the topology optimization of a bridge structure, that involves passive regions in the optimization domain, one moving load and a polygonal mesh is employed. This last example has the objective to show the applicability of the ESL approach to complex scenarios and non-regular meshes.

4.5.1

Cantilever beam subjected to one half-cycle sinusoidal load

In this example, it is studied the effect of the duration of the 1 kN half-cycle sinusoidal load in the cantilever beam represented in Figure 4.3. The geometric and optimization parameters used in the simulation are summarized in Table 4.2.

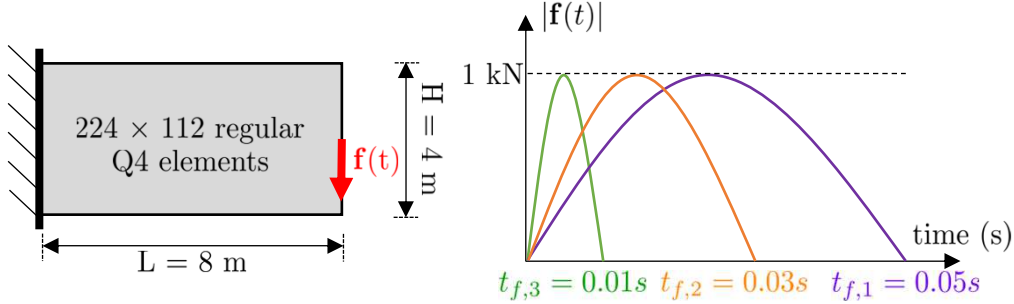


Figure 4.3: Illustration of the cantilever beam example with the three different sinusoidal loads studied (Adapted from [9])

Table 4.2: Parameters in the cantilever beam example

Parameter	Value
Domain thickness	0.01 m
Young's modulus (E_0)	200 GPa
Poisson's ratio (ν_0)	0.3
Rayleigh damping parameters	$\alpha_r = 10, \beta_r = 1 \times 10^{-5}$
Filter parameters	$R = 0.2$ m, $q = 1$
Number of time steps	100
Mass density (solid material)	7800 kg/m ³
Volume fraction	0.50

The objective of this study is to understand the behavior of the final topology when the duration of the load decreases. For that, as illustrated in Figure 4.3, A regular mesh is employed with 25088 Q4 elements. For the optimization, it is used the RAMP-H1 model as suggested in *PolyDyna*, with the penalty parameter p_0 starting at 0 and increasing by 1.5 every 25 iterations until a maximum value of 9.

As in this example only one node is loaded in all possible loading scenarios, according to Tarek and Ray [11] the best model reduction method to be used in this situation is the SVD-based. To illustrate, in the graphic of Figure 4.4 it is plotted all the singular values of the problem at the beginning of the first topology optimization cycle. For this reason, only the equivalent loading scenarios associated with the 4-first singular values will be considered

and the others can be neglected. Consequently, the ESL problem is reduced from 101 load cases to only 4, which saves an enormous computational time.

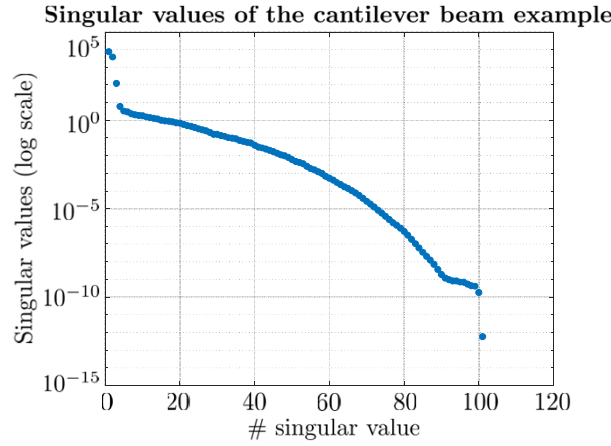


Figure 4.4: Singular values of the ESL matrix at the beginning of the first cycle of topology optimization

The final topologies obtained by the original PolyDyna and with the modifications proposed in this work are presented in Figure 4.5, for each of the durations indicated in Figure 4.3. Analyzing the results, it is possible to notice that the two first results are very similar, but the last one is completely different. According to PolyDyna's authors, this is because the effects of the load are extremely fast, and the two parallel beam-like members in the result help in reducing the vertical deflection of the beam.

Another interesting comparison can be established from the graphic of the dynamic compliance for the three studied load durations. In Figure 4.6 it is plotted one graphic with the initial and the optimized dynamic compliances for each of the durations. Again, one notices that in the last situation, the final dynamic compliance obtained with the proposed approach diverges considerably from what is expected.

From the results showed in Figure 4.6 it is possible to evaluate the value of the objective function represented in Equation (4-1), which is shown in Tables 4.3, 4.4 and 4.5. Along with the objective function values, it is shown the computational time of simulation to achieve the final topologies. All these times were measured in the same PC with an Intel(R) Core (TM) i7-8550U CPU, 1.80 GHz processor and 16 GB of RAM memory, running MATLAB R2019a.

In this example, the divergences on the last case of load duration can be associated with the limitation of the ESL method stated by Stolpe [46] due to the pulse shape of the load, which results in the ESLs not representing well the loads acting on the structure.

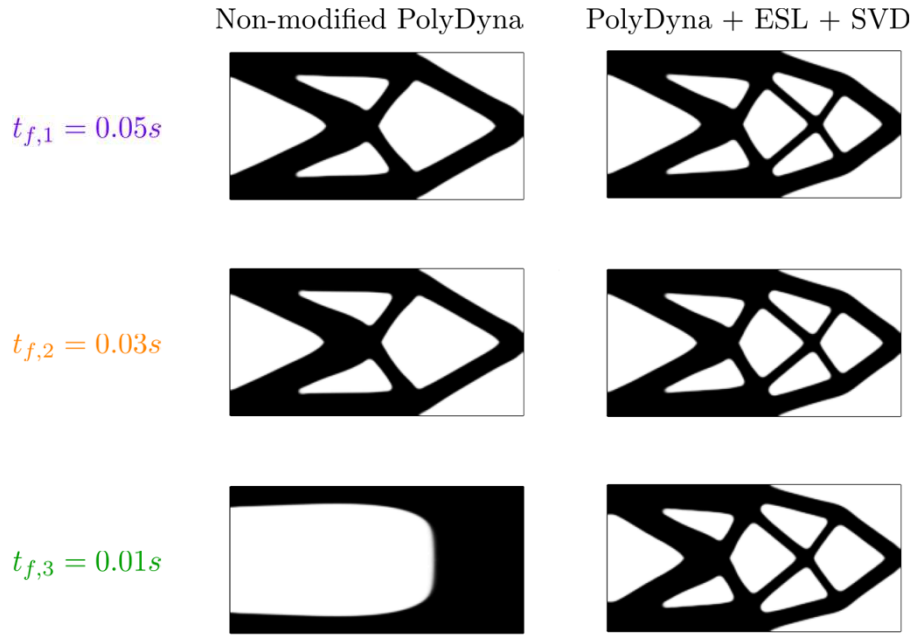


Figure 4.5: Comparison between the final topologies obtained with the original and the modified PolyDyna for the three studied load durations

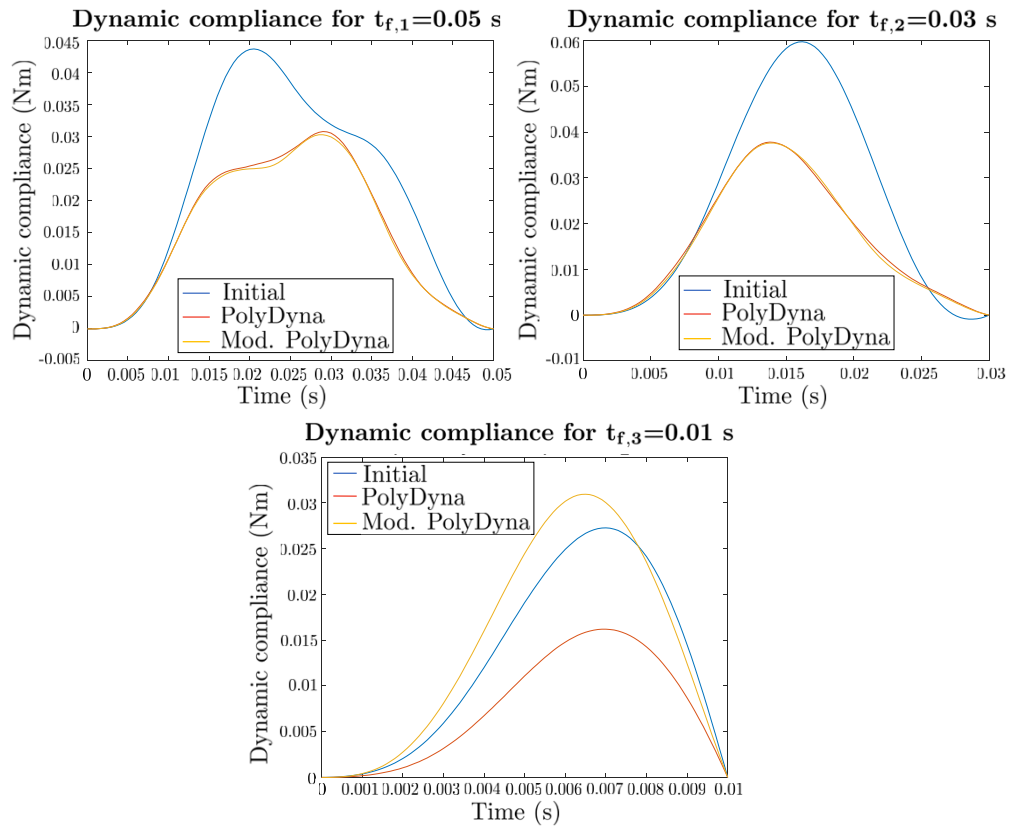


Figure 4.6: Graphics of the initial and optimized dynamic compliance for the three studied scenarios

Table 4.3: Numerical results for the cantilever beam with load duration of 0.05 seconds

Case	Dynamic compl. (Nm)	Time of simulation (s)
Initial	1.064×10^{-3}	–
PolyDyna (original)	7.061×10^{-4}	2472
PolyDyna (modified)	7.476×10^{-4}	354

Table 4.4: Numerical results for the cantilever beam with load duration of 0.03 seconds

Case	Dynamic compl. (Nm)	Time of simulation (s)
Initial	7.160×10^{-3}	–
PolyDyna (original)	4.727×10^{-4}	2532
PolyDyna (modified)	4.655×10^{-4}	355

Table 4.5: Numerical results for the cantilever beam with load duration of 0.01 seconds

Case	Dynamic compl. (Nm)	Time of simulation (s)
Initial	1.316×10^{-4}	–
PolyDyna (original)	7.703×10^{-5}	774
PolyDyna (modified)	1.489×10^{-4}	853

4.5.2

Support structure under rotating load

In this other example from PolyDyna, the squared domain in Figure 4.7 is subjected to one rotating load in the middle point of its upper edge. In this case, the magnitude of the load is fixed, but it is studied the effect of its angular speed on the final topology using the original PolyDyna and the modified PolyDyna to improve the computational efficiency.

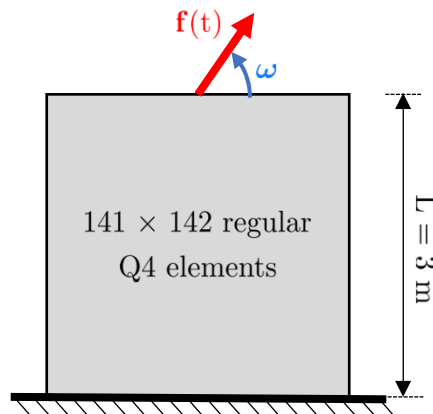


Figure 4.7: Illustration of the support structure example with the rotating load at the mid point of the upper edge (Adapted from [9])

Three angular speeds are considered in this study: $\omega_1 = 50\pi$ rad/s, $\omega_2 =$

100π rad/s and $\omega_3 = 200\pi$ rad/s. The simulation time is 5 complete turns of the load – *i.e.* $T_f = 10\pi/\omega$, and is discretized in 150 time steps as suggested in PolyDyna. The other parameters associated with the material, mesh and optimization are summarized in Table 4.6.

Table 4.6: Parameters in the support example

Parameter	Value
Domain thickness	0.05 m
Young's modulus (E_0)	70 GPa
Poisson's ratio (ν_0)	0.33
Rayleigh damping parameters	$\alpha_r = 50, \beta_r = 3 \times 10^{-5}$
Filter parameters	$R = 0.08$ m, $q = 1$
Mass density (solid material)	2700 kg/m ³
Volume fraction	0.25
Load magnitude ($ \mathbf{f}(t) $)	1×10^6 N

As in the cantilever beam example, the ESLs of this case are also highly correlated, because the load is applied to only one node of the structure. To illustrate, the value of all singular vectors of the matrix that regroups the ESLs at the beginning of the first cycle is shown in Figure 4.8, where it is possible to distinguish the 4 dominant singular values that are considered in the SVD-based model reduction applied to the ESL approach in this example.

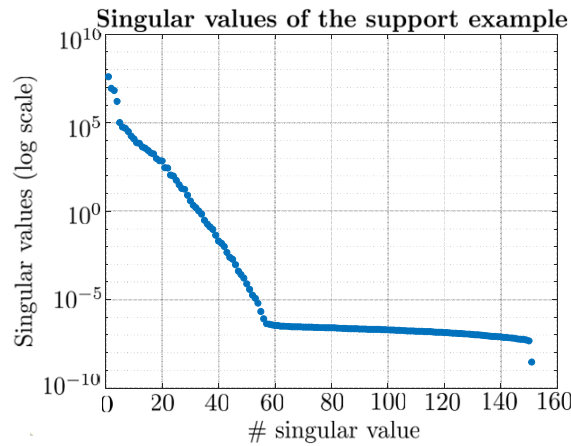


Figure 4.8: Singular values of the ESL matrix at the beginning of the first cycle of topology optimization

As happened in the last example, the final topology for $\omega = 200\pi$ rad/s diverges from the two other studied velocities, due to the high speed of the load. Figure 4.9 shows the final topologies for each of the scenarios achieved by employing the original and the modified PolyDyna software. According to PolyDyna's authors, the lateral bracing system that appears in addition to the diagonal members serves to restrict the lateral movement of the structure due

to the high speed of the load, which is not taken into account when the ESL approach is used.

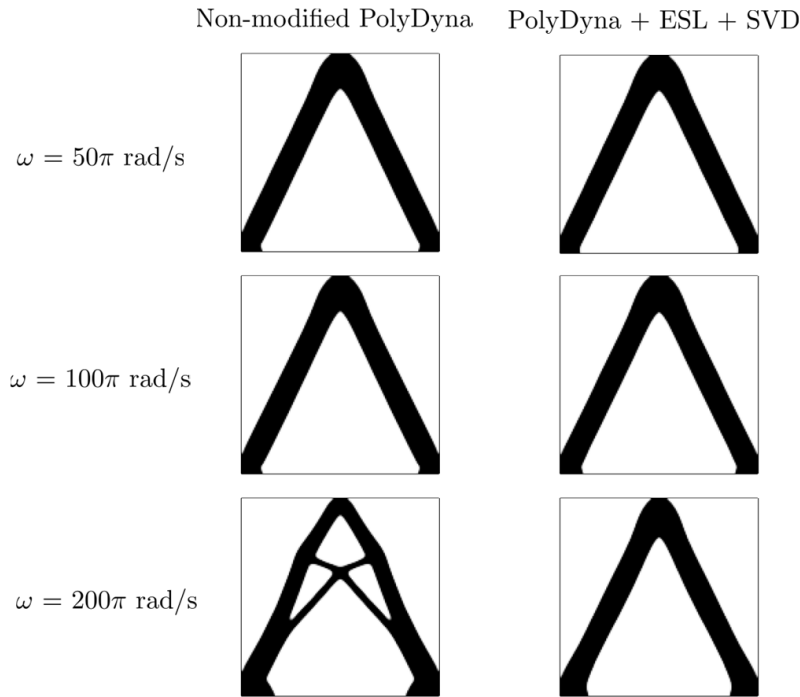


Figure 4.9: Comparison between the final topologies obtained with the original and the modified **PolyTop** for the three studied load durations

Following the analysis of the results, Figure 4.10 shows the graphics of the dynamic compliance values before and after the topology optimization. According to [9], the first time steps are influenced by the dynamic effects, and for this reason, the transient behavior is observed. After this region, it is possible to notice that the final result for the compliance using the original or the modified **PolyDyna** software is comparable, but diverges in the last case due to the limitation of the ESL method.

In Tables 4.7, 4.8 and 4.9 it is shown the initial and final objective function for each of the three scenarios and the time necessary to achieve the final topologies in the same computer with the properties listed in Section 4.5.1.

Table 4.7: Numerical results in the support example with angular speed $\omega_1 = 50\pi$ rad/s

Case	Dynamic compl. (Nm)	Time of simulation (s)
Initial	1.704×10^3	–
PolyDyna (original)	0.940×10^3	2916
PolyDyna (modified)	1.012×10^3	289

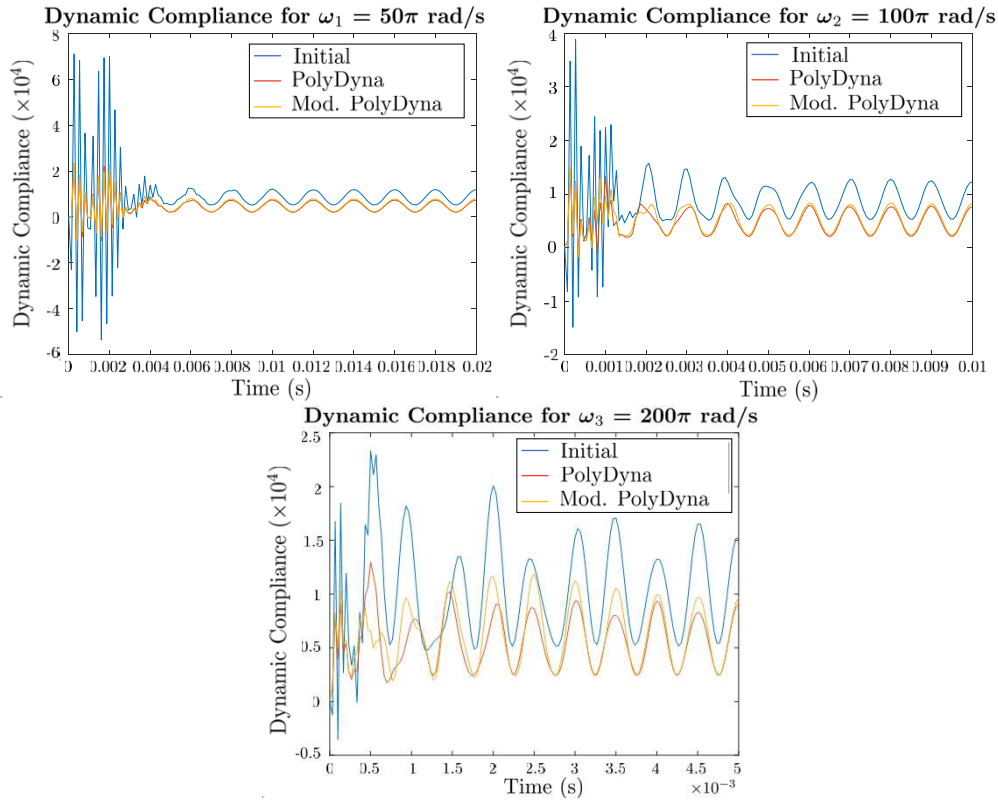


Figure 4.10: Graphs of the initial and optimized dynamic compliance for the three studied scenarios

Table 4.8: Numerical results in the support example with angular speed $\omega_2 = 100\pi$ rad/s

Case	Dynamic compl. (Nm)	Time of simulation (s)
Initial	8.886×10^2	—
PolyDyna (original)	4.872×10^2	2838
PolyDyna (modified)	5.263×10^2	357

Table 4.9: Numerical results in the support example with angular speed $\omega_3 = 200\pi$ rad/s

Case	Dynamic compl. (Nm)	Time of simulation (s)
Initial	5.139×10^2	—
PolyDyna (original)	2.795×10^2	3012
PolyDyna (modified)	1.229×10^2	691

4.5.3 Bridge under moving load

In this last example, the objective is to show the application of the modified PolyDyna software to the topology optimization of a more realistic problem. To this end, one bridge structure with the geometry given in Figure 4.11 is subjected to one moving load that can represent the traffic crossing the structure.

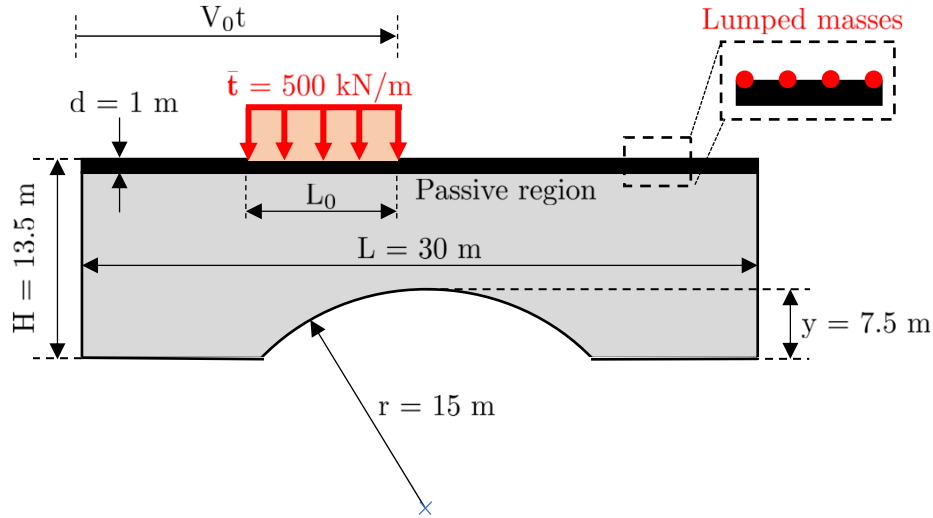


Figure 4.11: Illustration of the bridge example with the moving load (Adapted from [9])

In the above edge of the structure, it is placed a passive region, and lumped masses are uniformly distributed across the elements of this zone to add $m = 0.18 \times 10^6$ kg to the structure, which represents the deck mass.

Due to the complexity of the problem, it is suggested in **PolyDyna** the use of a graded polygonal mesh composed of 20000 elements inside the optimization domain, which is coarse towards the bottom of the bridge and becomes finer towards the top. This is important because the final topology tends to be more complex toward the bridge's deck. One example of mesh used in the simulations is given in Figure 4.12, as the polygonal mesh generator in the **PolyMesher** software is random [5].

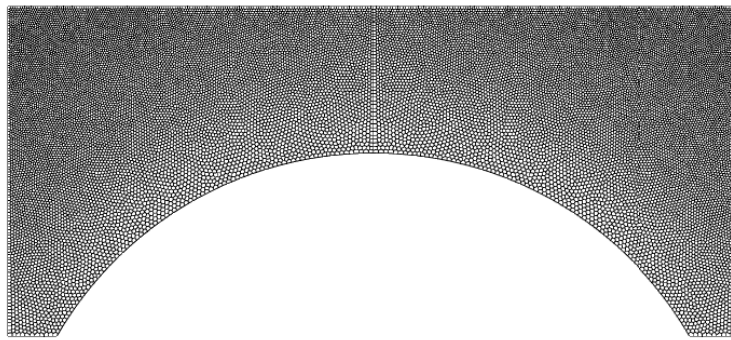


Figure 4.12: Example of mesh used in the bridge example

For this study, four different velocities V_0 are considered: 30, 60, 90, and 120 km/h. The total time of simulation is L/V_0 , which is discretized in 100-time steps. The other necessary parameters for the simulation are listed in Table 4.10.

Table 4.10: Parameters in the bridge example

Parameter	Value
Domain thickness	1 m
Young's modulus (E_0)	35 GPa
Poisson's ratio (ν_0)	0.25
Rayleigh damping parameters	$\alpha_r = 2.5, \beta_r = 4.5 \times 10^{-4}$
Filter parameters	$R = 0.3$ m, $q = 1$
Mass density (solid material)	2400 kg/m^3
Volume fraction	0.30

In the last two examples, it was possible to determine the dominant singular values associated with the ESLs, but in this case, the loads are not necessarily correlated, as the loaded nodes will not always be the same. Due to this fact, it is impossible to use only a few load cases in the ESL approach as illustrated by the graphic in Figure 4.13. Consequently, there is no advantage in employing the SVD-based model reduction method.

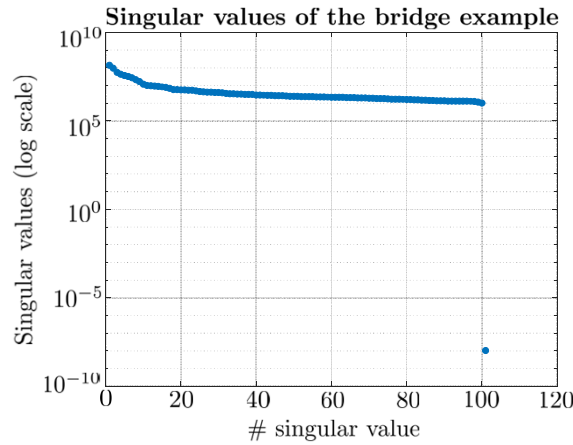


Figure 4.13: Singular values of the ESL matrix at the beginning of the first cycle of topology optimization

To deal with this situation, in this example it is used the SA-based method presented in Section 3.4. Its advantage is that the problem is always reduced to the desired number of associated loads regardless of the singular values, and 6 is a good choice according to Zhang *et al.* [6]. In this context, in Figure 4.14 the final topologies for each of the considered velocities are shown, using the original PolyDyna formulation and the modified algorithm, in this case with the SA model reduction.

After Figure 4.14, it is possible to notice that the modified PolyDyna code could achieve final topologies that are comparable to the ones from the original program. In the computer specified in Section 4.5.1, the simulations

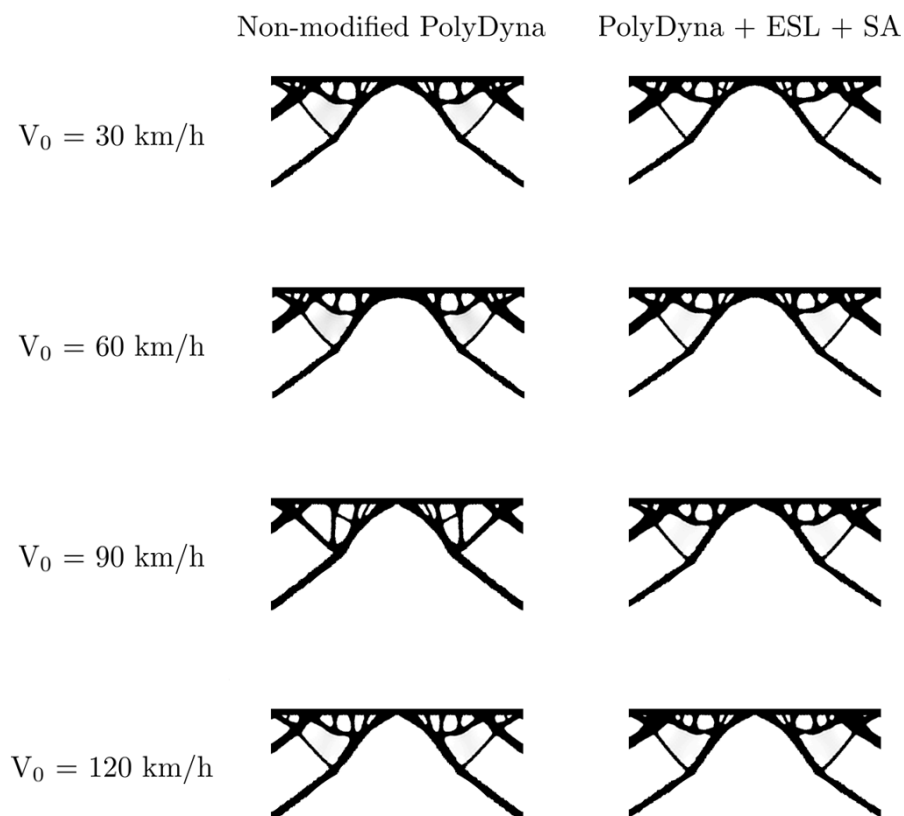


Figure 4.14: Comparison between the final topologies obtained with the original and the modified PolyDyna for the four studied velocities V_0 .

with the non-modified code take around two hours to achieve the final result, while the modified program reduces this time to less than one hour.

5

Robust Topology Optimization (RTO)

The second application that is explored in this work is the Robust Topology Optimization (RTO) when the structure is subjected to random loads. The RTO involves the expected value of the compliance and the variance in the objective function, and this chapter shows how it is possible to use the contents that were already discussed in this work to solve this problem based on a multiple loads scenario built from a Monte Carlo sampling. Especially it is shown how to apply the SVD-based model reduction method, which brings an enormous reduction in the computational effort necessary to achieve the final topologies.

5.1

Concepts of Robust Topology Optimization

The robust topology optimization, based on load uncertainty as introduced in the works from Dunning *et al.* [19] and Dunning and Kim [10], seeks to find the optimal material distribution inside a given domain, that minimizes the objective function:

$$Rb[\phi] = a \mathbb{E}[\phi] + b \text{Var}[\phi], \quad (5-1)$$

where $\mathbb{E}[\phi]$ represents the expected value of the compliance and $\text{Var}[\phi]$ its variance. The coefficients a and b are the weighting coefficients for each term of the sum and are commonly used as $b = 1 - a$, although not mandatory.

Supposing that \mathbf{f} is the random vector that represents the nodal loadings, an estimator for the expected compliance based on n_s samples of \mathbf{f} can be written by:

$$\hat{\mathbb{E}}[\phi] = \frac{1}{n_s} \sum_{i=1}^{n_s} \phi_i, \quad (5-2)$$

where ϕ_i represents the compliance associated with the i -th load sample $\bar{\mathbf{f}}_i$:

$$\phi_i = \bar{\mathbf{f}}_i^T \bar{\mathbf{u}}_i, \quad (5-3)$$

with $\bar{\mathbf{u}}_i$ representing the nodal displacement associated to $\bar{\mathbf{f}}_i$:

$$\bar{\mathbf{u}}_i = \mathbf{K}^{-1} \bar{\mathbf{f}}_i. \quad (5-4)$$

To evaluate the second term of the sum in Equation (5-1), one can first define the compliances vector $\boldsymbol{\phi} = [\phi_1, \phi_2, \dots, \phi_n]$, where the i -th component contains the compliance value associated to the i -th load sample. From this definition, it is possible to rewrite the expected value estimator in Equation (5-2) in vector form:

$$\hat{\mathbb{E}}[\boldsymbol{\phi}] = \frac{1}{n_s} \boldsymbol{\phi}^T \mathbf{1}, \quad (5-5)$$

where $\mathbf{1}$ is a ones vector with the same dimension as $\boldsymbol{\phi}$.

Regarding the variance estimator, that is typically calculated using the scalar form [37]:

$$\hat{\text{Var}}[\phi] = \frac{1}{n_s - 1} \sum_{i=1}^{n_s} (\phi_i - \hat{\mathbb{E}}[\phi])^2, \quad (5-6)$$

it is also possible to rewrite it in a vector form to facilitate computational implementation [11]:

$$\hat{\text{Var}}[\boldsymbol{\phi}] = \frac{1}{n_s - 1} (\boldsymbol{\phi} - \hat{\mathbb{E}}[\boldsymbol{\phi}] \mathbf{1})^T (\boldsymbol{\phi} - \hat{\mathbb{E}}[\boldsymbol{\phi}] \mathbf{1}), \quad (5-7)$$

where $\hat{\mathbb{E}}[\boldsymbol{\phi}]$ is the expected compliance calculated in Equation (5-2).

To compute the optimization, it is important to establish analytical expressions for the sensitivities of the robust objective function $Rb[\boldsymbol{\phi}]$ in Equation (5-1). In this context, applying the sum rule to the derivative concerning the e -th element in the densities vector $\boldsymbol{\rho}$, ρ_e :

$$\frac{\partial Rb[\boldsymbol{\phi}]}{\partial \rho_e} = a \frac{\partial \mathbb{E}[\boldsymbol{\phi}]}{\partial \rho_e} + b \frac{\partial \text{Var}[\boldsymbol{\phi}]}{\partial \rho_e}. \quad (5-8)$$

Observing the result in Equation (5-8), one notes that the derivative in the first term of the sum is exactly the derivative of the expected compliance value concerning the e -th density in $\boldsymbol{\rho}$. Based on the result from Equation 5-2, in this case, it is possible to use the multiple loading scenarios approach, and consequently this value can be calculated by [3, 11]:

$$\frac{\partial \mathbb{E}[\boldsymbol{\phi}]}{\partial \rho_e} = - \sum_{i=1}^{n_s} \bar{\mathbf{u}}_i^T \frac{\partial \mathbf{K}_e}{\partial \rho_e} \bar{\mathbf{u}}_i = - \sum_{i=1}^{n_s} (\mathbf{K}^{-1} \bar{\mathbf{f}}_i)^T \frac{\partial \mathbf{K}_e}{\partial \rho_e} (\mathbf{K}^{-1} \bar{\mathbf{f}}_i), \quad (5-9)$$

where \mathbf{K}_e is the e -th element stiffness matrix.

To calculate the derivative of the second term of Equation (5-8), the first step involves the application of the chain rule:

$$\frac{\partial \text{Var}[\boldsymbol{\phi}]}{\partial \rho_e} = \left[\frac{\partial \text{Var}[\boldsymbol{\phi}]}{\partial \mathbb{E}[\boldsymbol{\phi}]} \right]^T \frac{\partial \mathbb{E}[\boldsymbol{\phi}]}{\partial \rho_e}. \quad (5-10)$$

The result in Equation (5-10) shows that the sensitivity computed in (5-9) appears again in the variance derivative calculation. Using the expression

obtained in Equation (5-7), it is possible to calculate the sensitivity inside the brackets in (5-10):

$$\frac{\partial \text{Var}[\phi]}{\partial \mathbb{E}[C]} = \frac{2}{n_s - 1} (\phi - \hat{\mathbb{E}}[\phi] \mathbf{1}) = \mathbf{w}. \quad (5-11)$$

In this fashion, it is possible to write the analytic expression for the derivative in Equation (5-10):

$$\frac{\partial \text{Var}[\phi]}{\partial \rho_e} = - \sum_{i=1}^{n_s} w_i (\mathbf{K}^{-1} \bar{\mathbf{f}}_i)^T \frac{\partial \mathbf{K}_e}{\partial \rho_e} (\mathbf{K}^{-1} \bar{\mathbf{f}}_i), \quad (5-12)$$

where w_i is the i -th element of the vector \mathbf{w} defined in Equation (5-11).

Despite the great simplifications performed to compute analytically the sensitivities, to compute the estimators in Equations (5-2) and (5-7), it is necessary to solve one Finite Element problem for each sample from the Monte-Carlo analysis, which is computationally very expensive, as happens with the standard approach presented in Chapter 3. For this reason, it is shown in Section 5.2 how to introduce the SVD approach shown in Section 3.5 to improve the efficiency in obtaining the estimations and its sensitivities discussed in this Section.

5.2

Use of SVD approach in RTO context

In this part, it is discussed how to implement the SVD approach discussed in Section 3.5 into the robust topology optimization problem. The point of departure for this analysis is again the application of the sum rule to the derivative of the robust objective function expressed in Equation (5-8), where one can observe one derivative of the expectance $\mathbb{E}[\phi]$ and other on the variance $\text{Var}[\phi]$. In this context, the first derivative was already discussed in Chapter 3 and can be written as follows, considering $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ the expression for SVD decomposition of the matrix \mathbf{F} that gathers the n_{MC} samples from Monte-Carlo analysis:

$$\frac{\partial \mathbb{E}[\phi]}{\partial \rho_e} \approx - \sum_{i=1}^{n_{\text{SVD}}} [\mathbf{K}^{-1} (\mathbf{U}\mathbf{\Sigma})_i]^T \frac{\partial \mathbf{K}_e}{\partial \rho_e} [\mathbf{K}^{-1} (\mathbf{U}\mathbf{\Sigma})_i], \quad (5-13)$$

where n_{SVD} is the number of dominant singular values ($n_{\text{SVD}} \leq n_{\text{MC}}$) and $(\mathbf{U}\mathbf{\Sigma})_i$ represents the i -th column of matrix $\mathbf{U}\mathbf{\Sigma}$.

The challenge is now how to compute the variance since the estimator in Equation (5-7) needs the value of the compliances for all loading scenarios. The key to solving this question is to go back into the compliance definition, and apply the SVD decomposition definition:

$$\phi_i = \bar{\mathbf{f}}_i^T \bar{\mathbf{u}}_i = \bar{\mathbf{f}}_i^T \mathbf{K}^{-1} \bar{\mathbf{f}}_i = \bar{\mathbf{f}}_i (\mathbf{K}^{-1} \mathbf{U}\mathbf{\Sigma}) \mathbf{V}_i^T, \quad (5-14)$$

with ϕ_i representing the i -th compliances vector entry, and \mathbf{V}_i the i -th column of the right singular vectors matrix \mathbf{V} . In this fashion, it is possible to save the result:

$$\mathbf{Q} = \mathbf{K}^{-1}\mathbf{U}\Sigma, \quad (5-15)$$

while calculating the expectance, which allows the calculations of the compliance values in Equation (5-14) with constant time complexity, without solving Finite Element problems.

Once shown how to compute the vector ϕ , the next step is to calculate the variance derivative involving the SVD decomposition of the loading scenarios. In Equation (5-11) it was shown that the derivative of the variance is exactly the derivative of the expected value multiplied by weighting factors expressed in vector \mathbf{w} . Due to this fact, it is possible to write:

$$\frac{\partial \text{Var}[\phi]}{\partial \rho_e} = \frac{\partial}{\partial \rho_e} (\phi^T \mathbf{w}) = \frac{\partial}{\partial \rho_e} \text{tr} (\mathbf{D}_w \mathbf{F}^T \mathbf{K}^{-1} \mathbf{F}), \quad (5-16)$$

writing the objective function as the trace of a matrix and introducing the diagonal matrix \mathbf{D}_w with the values of the weighting vector \mathbf{w} on its diagonal. Applying the SVD decomposition $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^T$ to the result in (5-16):

$$\text{tr} (\mathbf{D}_w \mathbf{F}^T \mathbf{K}^{-1} \mathbf{F}) = \text{tr} (\mathbf{V}^T \mathbf{D}_w \mathbf{V} \Sigma \mathbf{U}^T \mathbf{K}^{-1} \mathbf{U} \Sigma). \quad (5-17)$$

If one defines:

$$\mathbf{X} = \mathbf{V}^T \mathbf{D}_w \mathbf{V}, \quad (5-18)$$

it is possible to develop Equation (5-16) as:

$$\frac{\partial}{\partial \rho_e} \text{tr} (\mathbf{D}_w \mathbf{F}^T \mathbf{K}^{-1} \mathbf{F}) = \frac{\partial}{\partial \rho_e} \text{tr} (\mathbf{X} \Sigma \mathbf{U}^T \mathbf{K}^{-1} \mathbf{U} \Sigma) = -\text{tr} (\mathbf{X} \mathbf{Q}^T \mathbf{K}_e \mathbf{Q}), \quad (5-19)$$

using the definition of \mathbf{Q} in Equation (5-15) and using the notation \mathbf{K}_e for the element stiffness matrix associated with the e -th element in the mesh that discretizes the base domain.

The result in (5-19) shows that to compute the variance derivative with respect to the project variables, one needs to compute the matrix \mathbf{X} when calculating the SVD decomposition of the load cases matrix \mathbf{F} and update the values of \mathbf{Q} at each update of the densities vector $\boldsymbol{\rho}$. This means that it is possible to obtain the sensitivities for the robust topology optimization problem by a simple equation when employing the SVD method, as in the scenario of compliance minimization involving multiple loads.

5.3

Update Scheme in RTO problems

In the robust topology optimization problem, the objective function becomes considerably more complex, as it involves the expected value of the compliance and its variance. In this context, one should remember that the Optimality Criteria (OC) method is functional only if the derivative of the compliance with respect to the e -th density:

$$\frac{\partial \phi}{\partial \rho_e} = -\mathbf{u}^T \frac{\partial \mathbf{K}}{\partial \rho_e} \mathbf{u} \quad (5-20)$$

results in negative values [3]. In this new scenario where the variance is involved, there is no guarantee of negative derivatives along all the steps of the optimization, and consequently an alternative method should be employed.

In this context, the Method of Moving Asymptotes (MMA) from Svanberg [49] is used in the RTO simulations in the place of OC. MMA is also widely used in topology optimization combined with SIMP, without limitations related to the derivative sign.

5.4

Numerical examples

In this section, the focus is first in reproducing one numerical example from Dunning and Kim [10], the article that introduced the robust design concept into topology optimization. Then, one uncertainty in the direction of the load is added into the first example, in order to compare the effect of this uncertainty in the optimization results.

5.4.1

Simply supported beam

In this first example, a simply-supported beam as illustrated in Figure 5.1 is considered. This domain is subjected to three equally-distributed loads f_1 , f_2 and f_3 of same mean magnitude, $\mu = 1.0$, but their respective standard deviations are different: $\sigma_1 = 0.5$, $\sigma_2 = 0.1$ and $\sigma_3 = 0.2$.

The optimization and geometric parameters for the domain in Figure 5.1 are gathered in Table 5.1.

To define the loads, it is associated with each of the three loads one normal distribution of mean 1.0 and standard deviation equal to the respective standard deviation of each of the loads. In the sequence, 5000 samples of each load are generated to define 5000 loading scenarios to be analyzed in the optimization process.

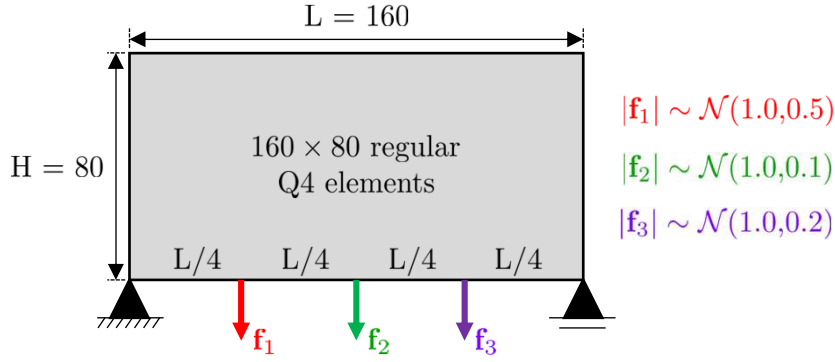


Figure 5.1: Illustration of the simply supported beam example with the detail of the applied loads and the mesh employed in the numerical simulations

Table 5.1: Parameters in the simply supported beam example

Parameter	Value (dimensionless)
Domain thickness	1.0
Young's modulus (E_0)	1.0
Poisson's ratio (ν_0)	0.3
Filter radius	$1.20\sqrt{2}$
Volume fraction	0.40

Three scenarios are studied at this point: the first one is the optimization of the mean compliance only ($a = 1$ and $b = 0$), then equal weight for the two terms in Equation (5-1) ($a = b = 0.5$) and the last is the variance only ($a = 0$ and $b = 1$). The same set of samples from the three random loads is used along the scenarios, and in Figure 5.2 is presented the histogram of the samples. Table 5.2 gathers the samples mean values and standard deviations.

Table 5.2: Mean and standard deviation values from the samples used in the simply supported beam example

Load	f_1	f_2	f_3
Mean value	1.0100	0.9983	0.9987
Standard deviation	0.4975	0.0988	0.1963

In this example, it is evident that all possible loading scenarios are highly correlated, as the loads are every time applied at the same mesh nodes. To ensure the applicability of the SVD-based model reduction method to this example, in Figure 5.3 it is shown the plot of the singular values of the matrix \mathbf{F} of the load cases defined, where it is possible to distinguish three dominant singular values.

The final topologies obtained using the MMA update scheme are presented in Figure 5.4, and in Figure 5.5 it is presented the evolution of the

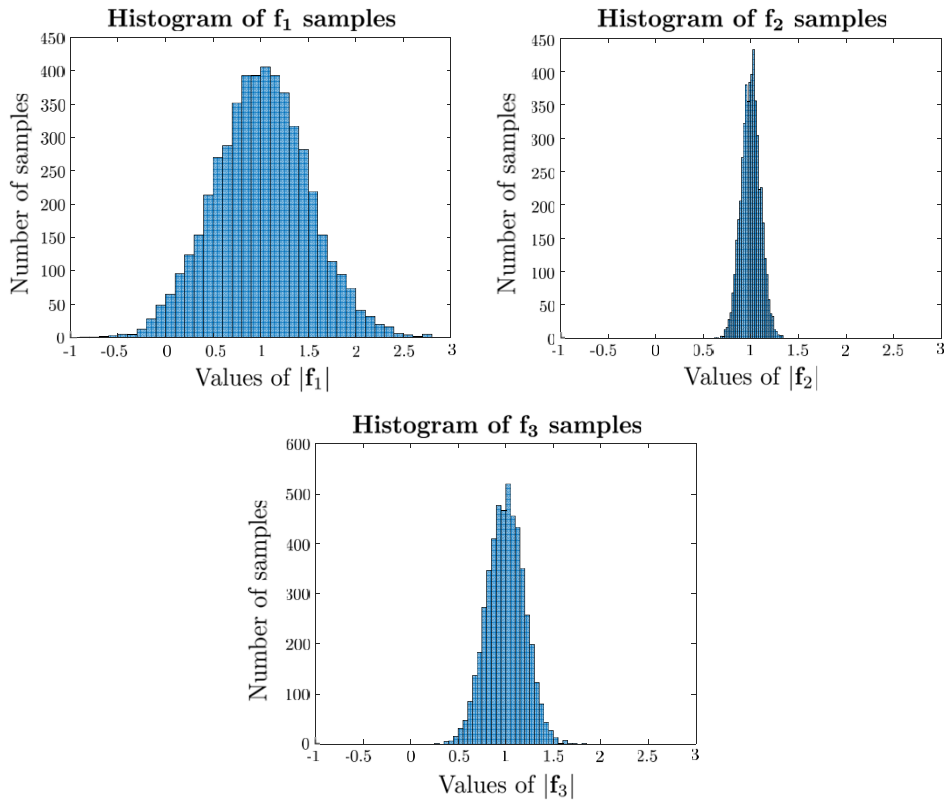


Figure 5.2: Histograms of the samples of the three loads in the simply supported beam example

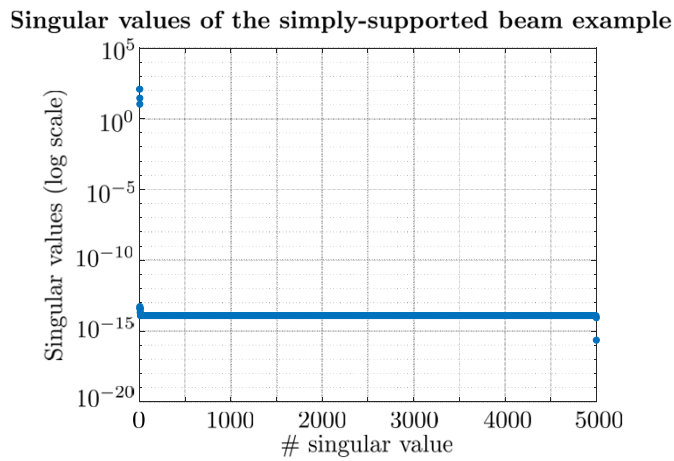


Figure 5.3: Singular values of the matrix that gathers the load cases used in the simply supported beam example

expected compliance and variance values along the iterations of the MMA scheme for the three analysed scenarios.

In Table 5.3 it is compared the expected compliance and variance final values with the results from Dunning and Kim [10].

With the SVD approach, the results in Figure 5.4 are obtained in around

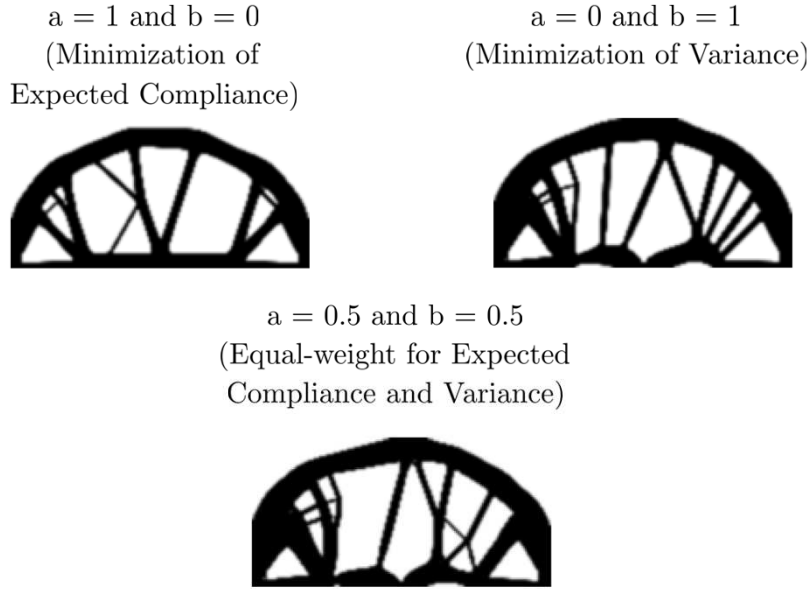


Figure 5.4: Final topologies obtained for the three studied combinations of values for a and b in the simply supported beam example

Table 5.3: Comparison of the expected compliance and variance results with the ones from Dunning and Kim [10]

Value	Expected compliance		Variance	
Scenario	This work	Ref. [10]	This work	Ref. [10]
$a = 1$ and $b = 0$	118.8	109.6	1508.8	1286.1
$a = 0$ and $b = 1$	137.5	138.9	717.8	825.7
$a = 0.5$ and $b = 0.5$	132.1	131.8	710.8	826.3

one hour in the standard PC specified in Section 4.5.1.

5.4.2

Simply supported beam with uncertainty in load direction

In this second example, a second uncertainty is added to the problem, now regarding the direction of each of the loads in the last example. It is supposed that the loads can vary by 45 degrees ($\pi/4$ radians), to the left or the right, as shown in Figure 5.6. The geometry and the mesh in this example remain the same as in section 5.4.1, as well as the uncertainty associated with the magnitude of the loads.

For simplicity, the samples of the load magnitudes used in this simulation were the same as in Section 5.4.1, and their histograms were shown in Figure 5.2. Now, in Figure 5.7 it is shown the histograms of the random variables θ_1 , θ_2 and θ_3 associated with the uncertainty in the loads direction.

Analyzing the singular values in Figure 5.8, it is possible to notice that the

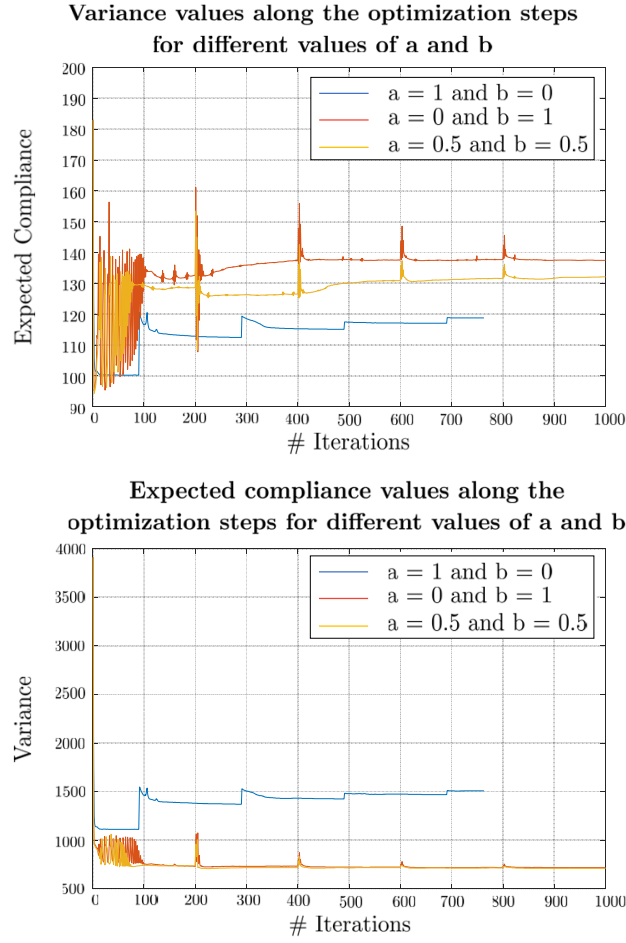


Figure 5.5: Evolution of the expected compliance and variance values along the optimization iterations for the three studied combinations of a and b .

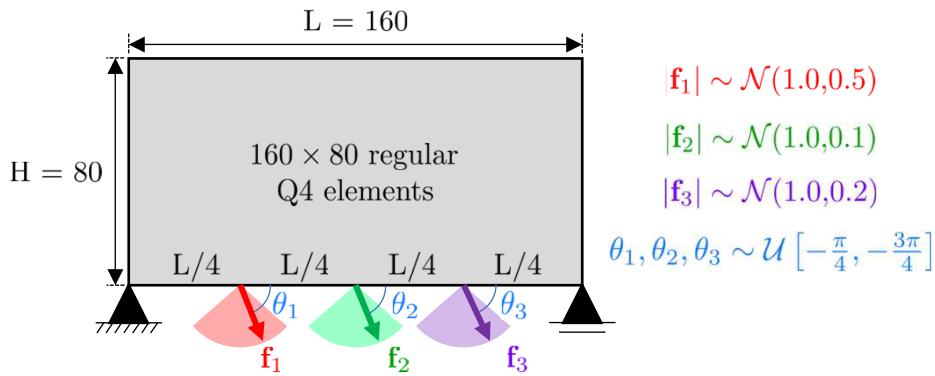


Figure 5.6: Illustration of the simply supported beam example with uncertainty in the direction of the loads.

problem now has 6 dominant singular values due to the existence of loading in the horizontal direction. Further, Table 5.1 gathers the parameter values used in the numerical simulations of this example.

In Figure 5.9 it is shown the final results of the optimization for three different scenarios of objective function: $a = 1$ and $b = 0$, $a = 0.5$ and $b = 0.5$,

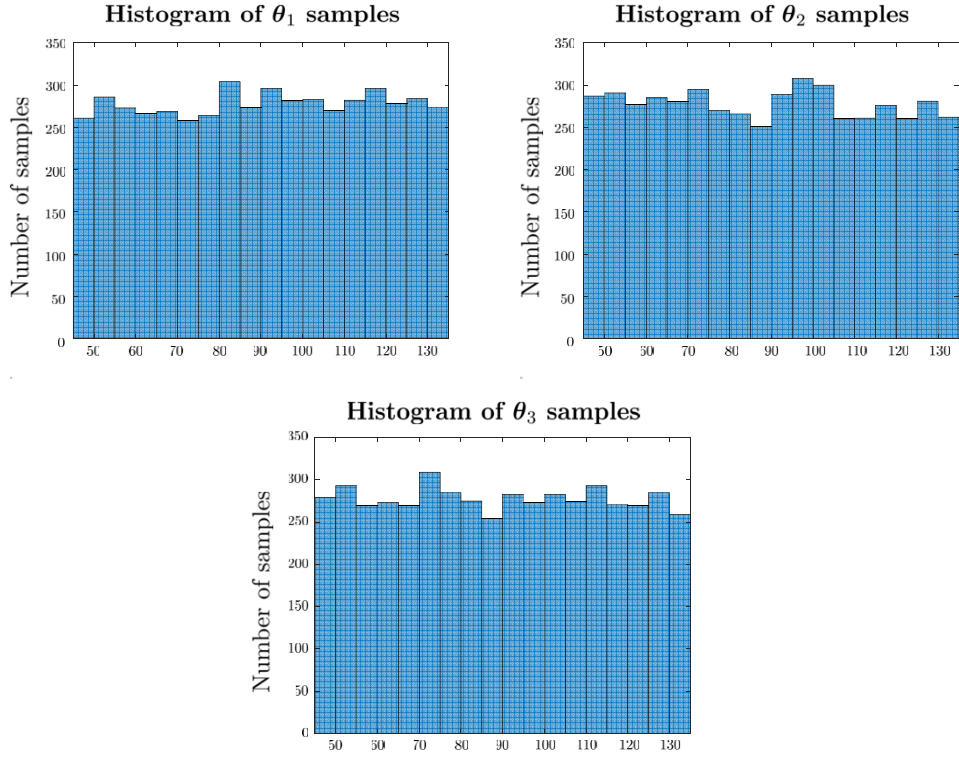


Figure 5.7: Histograms of the values for the angles θ_1 , θ_2 and θ_3 in the simulations involving uncertainty in the load direction.

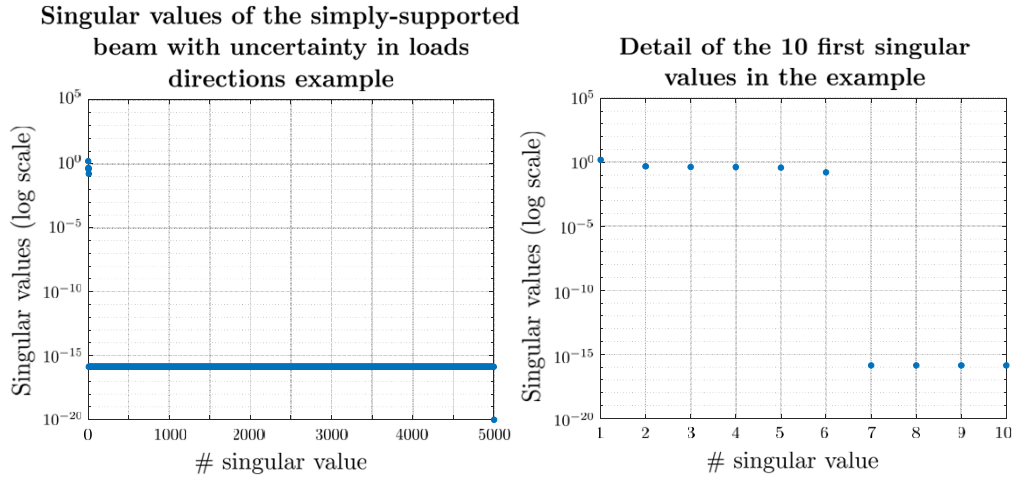


Figure 5.8: Singular values of the matrix that gathers the load cases used in the example (left) with the detail of the 10 first singular values (right)

and $a = 0$ and $b = 1$.

Compared to the result without uncertainty in the load direction, it is possible to observe that more material is placed close to the lower edge of the domain, due to the effects of the horizontal component of the loads that exist in this example. In Table 5.5 it is gathered the expected compliance, variance, and objective function final values for each of the three examples.

Table 5.4: Parameters in the simply supported beam example with uncertainty in load direction

Parameter	Value (dimensionless)
Domain thickness	1.0
Young's modulus (E_0)	1.0
Poisson's ratio (ν_0)	0.3
Filter radius	$1.20\sqrt{2}$
Volume fraction	0.50

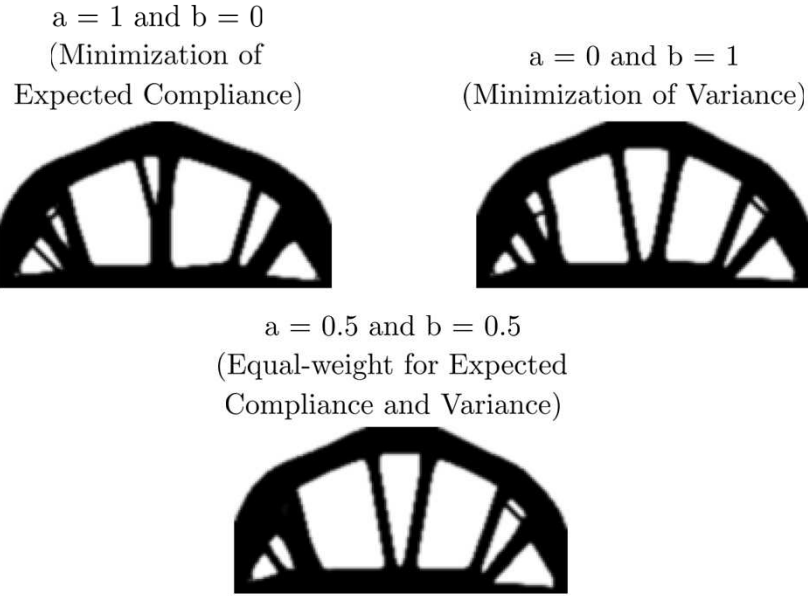


Figure 5.9: Final topologies obtained in the example with uncertainty in load magnitude and in load direction.

Table 5.5: Final values of expected compliance, variance and objective function in the example with uncertainty in load direction

Scenario	Expected Compliance	Variance	Obj. Function
$a = 1$ and $b = 0$	84.6598	1297.2748	84.6598
$a = 0.5$ and $b = 0.5$	79.3396	1094.6639	587.0017
$a = 0$ and $b = 1$	85.5705	1241.4627	1241.4627

It is important to emphasize that in this example, even with the complexity introduced by the uncertainty in the load direction, the time to perform the optimization remained around one hour, which demonstrates once again the importance of the model reduction methods in the context of topology optimization.

6

Conclusions and Perspectives

6.1

Work conclusions

At the beginning of this work it was described the problem when many load cases are involved in a topology optimization problem: a high number of finite element solutions are necessary during the optimization process, which is computationally very expensive. To improve the efficiency of these problems, it was recapitulated two existing methods in the literature, the first based on a stochastic approach (SA-based) and the other based on the singular value decomposition of the matrix that gathers the load cases (SVD-based).

After the presentation of the two approaches, one example from the literature was presented, to show in practice the effects of employing the SA or the SVD-based model reduction methods on the number of finite element problems that need to be solved until the convergence of the final structure is obtained. In this specific case, the reduction in this number was greater than 90%.

In the sequence, two applications of topology optimization involving multiple loading scenarios were identified: time-dependent topology optimization and robust topology optimization. For the first application, it is possible to apply the equivalent static loads (ESL) method to generate load cases that represent the dynamic effects into the structure.

In this work, it was shown the possibility of implementing the model reduction methods to the ESLs in dynamic applications. Three examples were shown, and the total time necessary to perform the simulations with and without the studied approaches to reduce the computational effort was compared. However, based on the numerical experimentations performed along this work, the ESL approach was not able to reproduce the same final topologies as the ones from **PolyDyna** when pulse-shaped loads are applied.

After that, the robust topology optimization was discussed. It was shown how the SVD-based approach can be employed in the calculations of the expected compliance and especially of the variance. With this method, it is possible to use Monte-Carlo sampling in a Topology Optimization problem,

an approach that is usually avoided in works of the subject in the literature.

To demonstrate this applicability, two different scenarios of random loads in a simply-supported beam were studied, with different combinations of the weighting factors for the expected compliance and the variance in the robust objective function. All the simulations took an extremely reasonable time to achieve the final topologies, which confirms that Monte-Carlo-based approaches can also be used in topology optimization contexts, and can be an efficient alternative to the other existing methods for RTO, such as Polynomial Chaos Expansion [32].

With the applications discussed in this dissertation, this work is contributing to extending the range of topology optimization applications.

6.2

Suggestions for future work

Despite the advances presented in this work, some suggestions for future works are listed in the sequence to continue expanding the applications of topology optimization:

- Proposition of an efficient method capable to generate multiple loading scenarios that represent well the effects of pulse-shaped loads, based on the works from Stolpe [46], Stolpe *et al.* [47] and Park and Lee [48];
- Apply the discussed ESL approach and the model reduction methods to non-linear dynamics, based on the works of Kim and Park [50] and Lee and Park [51];
- Investigate the possibility of extending the model reduction methods presented in this work to other topology optimization applications that are not compliance-based with volume constraints – for example, for topology optimization with stress constraints [20];
- Study the applicability of the proposed approaches to 3D time-dependent and robust problems.

Bibliography

References

- [1] DEPOSITPHOTOS. Tournage aérien d'un gros camion passant devant la caméra sur un grand pont d'une route routière et d'une route autoroutière sous le pont. Available in: <https://fr.depositphotos.com/97557042/stock-video-big-truck-passing-the-camera.html>. Access: 29/11/2020.
- [2] OFFSHORE MAGAZINE. Offshore wind energy. Available in: <http://https://www.offshore-mag.com/field-development/article/14072668/offshore-wind-energy>. Access: 29/11/2020.
- [3] BENDSOE, M. P.; SIGMUND, O.. **Topology optimization: theory, methods, and applications**. Springer Science & Business Media, 2013.
- [4] BURDEN, R. L.; FAIRES, J. D. ; BURDEN, A. M.. **Análise numérica**. Cengage Learning, 2016.
- [5] TALISCHI, C.; PAULINO, G. H.; PEREIRA, A. ; MENEZES, I. F. M.. **PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab**. Struct Multidisc Optim, 45(3):309–328, 2012.
- [6] ZHANG, X. S.; DE STURLER, E. ; PAULINO, G. H.. **Stochastic sampling for deterministic structural topology optimization with many load cases: Density-based and ground structure approaches**. Comput. Methods Appl. Mech. Engrg., 325:463–487, 2017.
- [7] CHOI, W. S.; PARK, G. J.. **Structural optimization using equivalent static loads at all time intervals**. Comput. Methods Appl. Mech. Engrg., 191:2077–2094, 2002.
- [8] LAVOR, B. V. A.; PEREIRA, A.. **Topology optimization of structures subjected to dynamic loads using equivalent static loads**. Proceedings of the 25th ABCM International Congress of Mechanical Engineering, 2019.

- [9] GIRALDO-LONDOÑO, O.; PAULINO, G. H.. **PolyDyna: a Matlab implementation for topology optimization of structures subjected to dynamic loads.** *Structural and Multidisciplinary Optimization*, 64(2):957–990, 2021.
- [10] DUNNING, P. D.; KIM, H. A.. **Robust topology optimization: minimization of expected and variance of compliance.** *AIAA journal*, 51(11):2656–2664, 2013.
- [11] TAREK, M.; RAY, T.. **Robust and stochastic compliance-based topology optimization with finitely many loading scenarios.** *Structural and Multidisciplinary Optimization*, 64(6):3439–3452, 2021.
- [12] MICHELL, A. G. M.. **The limits of economy of material in frame-structures.** *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(47):589–597, 1904.
- [13] HRENNIKOFF, A.. **Solution of problems of elasticity by the framework method.** *Journal of Applied Mechanics*, 8(4):169–175, 1941.
- [14] COURANT, R.. **Variational methods for the solution of problems of equilibrium and vibrations.** *Bulletin of the American Mathematical Society*, 49:1–23, 1943.
- [15] SCHMIT, L. A.. **Structural design by systematic synthesis.** In: *PROCEEDINGS OF THE SECOND NATIONAL CONFERENCE ON ELECTRONIC COMPUTATION*, ASCE, SEPT., 1960, 1960.
- [16] BENDSØE, M. P.; KIKUCHI, N.. **Generating optimal topologies in structural design using a homogenization method.** *Computer methods in applied mechanics and engineering*, 71(2):197–224, 1988.
- [17] BEHROU, R.; GUEST, J. K.. **Topology optimization for transient response of structures subjected to dynamic loads.** *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017.
- [18] ZHAO, J.; WANG, C.. **Dynamic response topology optimization in the time domain using model reduction method.** *Structural and Multidisciplinary Optimization*, 53(1):101–114, 2016.
- [19] DUNNING, P. D.; KIM, H. A. ; MULLINEUX, G.. **Introducing loading uncertainty in topology optimization.** *AIAA journal*, 49(4):760–768, 2011.

- [20] GIRALDO-LONDOÑO, O.; PAULINO, G. H.. **PolyStress: a Matlab implementation for local stress-constrained topology optimization using the augmented lagrangian method**. *Structural and Multidisciplinary Optimization*, 63(4):2065–2097, 2021.
- [21] MARTIN, A.; DEIERLEIN, G. G.. **Structural topology optimization of tall buildings for dynamic seismic excitation using modal decomposition**. *Engineering Structures*, 216:110717, 2020.
- [22] TALISCHI, C.; PAULINO, G. H.; PEREIRA, A. ; MENEZES, I. F. M.. **PolyTop: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes**. *Struct Multidisc Optim*, 45(3):329–357, 2012.
- [23] SIGMUND, O.. **A 99 line topology optimization code written in Matlab**. *Struct Multidisc Optim*, 21:120–127, 2001.
- [24] ANDREASSEN, E.; CLAUSEN, A.; SCHEVENELS, M.; LAZAROV, B. S. ; SIGMUND, O.. **Efficient topology optimization in matlab using 88 lines of code**. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011.
- [25] PICELLI, R.; SIVAPURAM, R. ; XIE, Y. M.. **A 101-line matlab code for topology optimization using binary variables and integer programming**. *Structural and Multidisciplinary Optimization*, 63(2):935–954, 2021.
- [26] MIN, S.; KIKUCHI, N.; PARK, Y.; KIM, S. ; CHANG, S.. **Optimal topology design of structures under dynamic loads**. *Structural optimization*, 17(2):208–218, 1999.
- [27] KANG, B. S.; CHOI, W. S. ; PARK, G. J.. **Structural optimization under equivalent static loads transformed from dynamic loads based on displacement**. *Computers and Structures*, 79:145–154, 1999.
- [28] JANG, H.-H.; LEE, H.; LEE, J. ; PARK, G.. **Dynamic response topology optimization in the time domain using equivalent static loads**. *AIAA journal*, 50(1):226–234, 2012.
- [29] LAVOR, B. V. A.; PEREIRA, A.. **Otimização topológica de estruturas planas sob a ação de cargas dinâmicas utilizando o método do carregamento estático equivalente**. Msc thesis, Mechanical Engineering Department, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2019.

- [30] ASADPOURE, A.; TOOTKABONI, M. ; GUEST, J. K.. **Robust topology optimization of structures with uncertainties in stiffness—application to truss structures.** Computers & Structures, 89(11-12):1131–1141, 2011.
- [31] GUEST, J. K.; IGUSA, T.. **Structural optimization under uncertain loads and nodal locations.** Computer Methods in Applied Mechanics and Engineering, 198(1):116–124, 2008.
- [32] CUELLAR, N.; PEREIRA, A.; MENEZES, I. F. ; CUNHA, A.. **Non-intrusive polynomial chaos expansion for topology optimization using polygonal meshes.** Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40(12):1–18, 2018.
- [33] ZHANG, X. S.; DE STURLER, E. ; SHAPIRO, A.. **Topology optimization with many right-hand sides using mirror descent stochastic approximation — reduction from many to a single sample.** Journal of Applied Mechanics, 87:051005, 2020.
- [34] CLOUGH, R.; PENZIEN, J.. **Dynamics of Structures.** McGraw-Hill, 1993.
- [35] NEWMARK, N. M.. **A method of computation for structural dynamics.** Journal of the engineering mechanics division, 85(3):67–94, 1959.
- [36] HILBER, H. M.; HUGHES, T. J. ; TAYLOR, R. L.. **Improved numerical dissipation for time integration algorithms in structural dynamics.** Earthquake Engineering & Structural Dynamics, 5(3):283–292, 1977.
- [37] GRIMMETT, G.; WELSH, D.. **Probability: an introduction.** Oxford University Press, 2014.
- [38] GERE, J.; GOODNO, B.. **Mecânica dos Materiais—7ª edição.** Cengage Learning, 2010.
- [39] HERBIN, E.. **Probabilités.** École Centrale Paris, 2016.
- [40] FAY, G.; VAYATIS, N.. **Statistique.** École Centrale Paris, 2016.
- [41] HUTCHINSON, M. F.. **A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines.** Commun. Statist, -Simula., 19(2):433–450, 1990.

- [42] VERWEIJ, B.; AHMED, S.; KLEYWEGT, A. J.; NEMHAUSER, G. ; SHAPIRO, A.. **The sample average approximation method applied to stochastic routing problems: A computational study**. Computational Optimization and Applications, 24:289–333, 2003.
- [43] KLEYWEGT, A. J.; SHAPIRO, A. ; HOMEM-DE MELLO, T.. **The sample average approximation method for stochastic discrete optimization**. SIAM Journal on Optimization, 12(2):479–502, 2002.
- [44] STRANG, G.. **Álgebra linear e suas aplicações**. Cengage Learning, 2010.
- [45] PARK, G.; KANG, B.. **Validation of a structural optimization algorithm transforming dynamic loads into equivalent static loads**. Journal of optimization theory and applications, 118(1):191–200, 2003.
- [46] STOLPE, M.. **On the equivalent static loads approach for dynamic response structural optimization**. Structural and Multidisciplinary Optimization, 50(6):921–926, 2014.
- [47] STOLPE, M.; VERBART, A. ; ROJAS-LABANDA, S.. **The equivalent static loads method for structural optimization does not in general generate optimal designs**. Structural and Multidisciplinary Optimization, 58(1):139–154, 2018.
- [48] PARK, G.-J.; LEE, Y.. **Discussion on the optimality condition of the equivalent static loads method for linear dynamic response structural optimization**. Structural and Multidisciplinary Optimization, 59(1):311–316, 2019.
- [49] SVANBERG, K.. **The method of moving asymptotes—a new method for structural optimization**. International journal for numerical methods in engineering, 24(2):359–373, 1987.
- [50] KIM, Y.-I.; PARK, G.-J.. **Nonlinear dynamic response structural optimization using equivalent static loads**. Computer Methods in Applied Mechanics and Engineering, 199(9-12):660–676, 2010.
- [51] LEE, H.-A.; PARK, G.-J.. **Nonlinear dynamic response topology optimization using the equivalent static loads method**. Computer Methods in Applied Mechanics and Engineering, 283:956–970, 2015.

A Published article

The methodology developed in this work to apply the SA and SVD-based approaches into dynamic and robust topology optimization was published in the XLII CILAMCE (Ibero-Latin-American Congress on Computational Methods in Engineering).



Efficient compliance-based topology optimization with many loading scenarios

Lucas do Nascimento Sagrilo¹, Anderson Pereira¹

¹*Dept. of Mechanical Engineering, Pontifical Catholic University of Rio de Janeiro
R. Marquês de São Vicente, 225, 22451-900, RJ/Rio de Janeiro, Brazil
lucas.sagrilo96@gmail.com, anderson@puc-rio.br*

Abstract. In topology optimization setting, we can cast a variety of problems into a weighted-sum of compliances minimization. Robust design, for example, is commonly addressed in the form of a finite sum of deterministic loading scenarios. Another example is the optimization of structures subjected to dynamic loads using the equivalent static load method, where a finite set of associated loads are defined according to the displacements over time. But when the number of loads involved is high, the solution of these problems becomes extremely expensive from a computational point of view, due to the necessity of solving one finite element problem for each of these load cases along the steps of the optimization algorithm to evaluate the objective function. In this context, two methods for dimensionality reduction of the problem are presented. First, an equivalent stochastic problem to the original one is determined, which reduces to only a few the number of necessary solves at each step. The second approach uses the singular-value decomposition of the matrix that gathers the different loading scenarios to reduce the number of linear systems solved. The applicability of both methods to different topology optimization scenarios is discussed, and numerical examples are proposed to compare the final topologies obtained and quantify the reduction in the number of necessary finite element solves.

Keywords: Topology optimization, Stochastic sampling, Singular value decomposition, Sample average approach, Equivalent static loads.

B

MATLAB Code for Static Analysis in Time-Dependent Problems

In this Appendix it is reproduced the MATLAB code used for the time dependent simulations of this work to perform static FE analysis based on a Cholesky decomposition as suggested in PolyDyna inside the ESL cycles.

```
1 %% DEFINITION OF FUNCTION FEM_ESL FOR STATIC FE ANALYSIS
2 function [un,K,fem] = FEM_ESL(fem,E,V)
3 fDOF = fem.FreeDofs;
4 un = zeros(2*fem.NNode,size(fem.F,2));
5 K = sparse(fem.i,fem.j,E(fem.e).*fem.k0); % Stiffness
    matrix
6 K = (K+K')/2;
7 Ft = fem.F;
8 for It=1:size(fem.F,2) % Loop over time steps
9     Fext = Ft(:,It);
10    if It==1
11        [un(fDOF,It),L,s] = SolveLinSys(K(fDOF,fDOF),Fext(fDOF));
12        fem.L = L; fem.s = s; % Store Cholesky decomposition
            info
13    else
14        un(fDOF(s),It) = L'\(L\Fext(fDOF(s)));
15    end
16 end
17 %% SOLVE LINEAR SYSTEM USING CHOLESKY DECOMPOSITION
18 function [U,L,s] = SolveLinSys(K,F)
19 [L,~,s] = chol(K,'lower','vector');
20 U(s,:) = L'\(L\F(s,:));
```

C

Modified PolyScript for ESL-based Time-Dependent Topology Optimization

As a complement of the text, in this appendix it is reproduced the lines of the PolyScript file that represents the flowchart in Figure 4.2, where the outer iterations represents the ESL cycles and the inner iterations the calls of the modified PolyDyna to update the topology densities.

```
1 %% MODIFIED POLYDYNAL OPTIMIZATION ROUTINE
2 for i = 1:MaxCycles
3     z = opt.zIni; [E,~,V,~] = MatIntFnc(opt.P*opt.zIni, '
4         RAMP-H1',
5         [0,1,0.5]);
6     [fem.u0,fem.v0] = deal(zeros(2*fem.NNode,1));
7     [un,~,~,fem.M,fem.C,fem.K,~] = FEM_Dyna(fem,E,V);
8     if (i>1)
9         lbDensities = 0.5; % lower bound threshold
10        nLowDensities = find(opt.mP*V<lbDensities);
11        ESLDofs = setdiff([1:2*fem.NNode],
12            [2*nLowDensities-1;2*
13            nLowDensities]);
14    else
15        nLowDensities = [];
16        ESLDofs = [1:2*fem.NNode];
17    end
18    fem.F = fem.K*un;
19    ActiveESLDofs = [2*nLowDensities-1;2*nLowDensities];
20    fem.F(ActiveESLDofs,:) = 0;
21    [fem.U,fem.S,fem.V] = svds(fem.F,4);
22    fem.F = fem.U*fem.S;
23    fields = {'K','M','C'}; fem = rmfield(fem,fields);
24    if (i==1), ini = 1; B = 1; p_i = 0:1.5:9; tic;
25    else, B = 10; p_i = 9; end
26    for ii=ini:length(p_i) %Continuation on the penalty
27        parameter
```

```

25         if ii==length(p_i); opt.MaxIter = 300; end
26         disp(['current p: ', num2str(p_i(ii)),
27             'current B: ', num2str(B)]);
28         opt.MatIntFnc = @(y)MatIntFnc(y,'RAMP-H1',[p_i(
29             ii),B,0.5]);
30         [opt.zIni,V,fem] = PolyDynaMOD(fem,opt);
31         B = min(B+2,10);
32     end
33     Change = max(abs(z-opt.zIni))/(opt.zMax-opt.zMin);
34     fprintf('Cycle: %i \t Change: %1.8f\n',i,Change);
35     if Change<opt.Tol, break; end
36 end

```