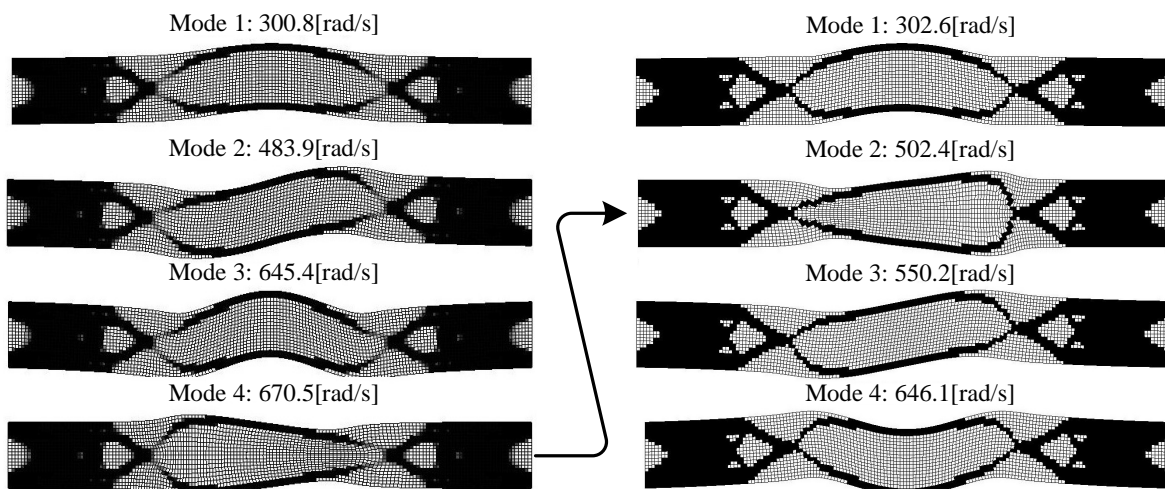


Reproducing the response of a structure by optimizing the design of a test rig

Master's thesis
by
Søren Randrup Henriksen
and
Peter Hansen



Spring 2012

Title Reproducing the response of a structure
by optimizing the design of a test rig

Project period Spring 2012
February 1st - June 1st 2012

Project group DMS4 80a

Group members

Hansen, Peter

Henrichsen, Søren Randrup

Supervisor Erik Lund

Copies 4

Number of pages 95

Preface

This thesis is written by group DMS4 80a on the fourth semester of the Master's programme Design of Mechanical Systems at Aalborg University, spring 2012. The title of the project is:

Reproducing the response of a structure by optimizing the design of a test rig

The project is conducted in collaboration with FS Dynamics AB, who provided a model of the wind turbine used in this project.

The project group would like to thank Professor Erik Lund, who supervised the project, for competent guidance and always questioning the findings of the group. His guidance has given birth to many discussions, thus increasing the groups knowledge and helped keeping the project on track.

Furthermore Hans Buus from FS Dynamics is thanked for help with the project proposal, and assistance with specific questions concerning the provided wind turbine model.

The notation used for vectors, matrices and units are as follows:

Vectors: $\{\phi\}$

Matrices: $[\mathbf{K}]$

Units: [kg]

A general list of nomenclature is found on page IX at the back. Further notation and nomenclature is presented where relevant.

In the back of the report a CD is located including:

- A PDF version of the report
- The provided model from FS Dynamics AB
- The 2D version of the code
- The 3D version of the code

Abstrakt

Denne master afhandling omhandler udviklingen af en kode, som på baggrund af et ønsket respons kan optimere en struktur således det ønskede respons er reproduceret.

Projektforslaget er fremsat i samarbejde med FS Dynamics. Ønsket er at udvikle en teststand, som kan reproducere responset fra et vindmølle tårn. Teststanden skal benyttes til fuldskala test af vindmølle naceller, således arbejdsbelastningerne kan genskabes af en meget lavere struktur. Arbejdsbelastningen er defineret af et sæt af egenfrekvenser, egensvingningsformer samt en tøjningstilstand i nacellen.

Fokus i rapporten er på udvikling af en metode, som ved hjælp af topologioptimering kan matche flere egenfrekvenser og tilhørende svingningsformer til et givent sæt af ønskede frekvenser og svingningsformer. Tøjningstilstanden er ikke inkluderet i projektet.

Før koden præsenteres er den grundlæggende teori beskrives. De benyttede stivheds- og masseinterpoleringsmetoder er først præsenteret sammen med en beskrivelse af hvorledes fænomenet "lokale svingningsformer" fjernes. Derefter er de styrende ligninger for frie svingninger præsenteret og på baggrund af disse er følsomhederne for simple egenfrekvenser udledt.

For at kunne genfinde egensvingningsformerne under optimering af topologien benyttes "Modal Assurance Criterion". Dette kriterie sikrer, at de korrekte følsomheder benyttes til at matche reference frekvenserne.

To optimeringsformuleringer er implementeret i koden. Den ene er maksimering af den fundamentale egenfrekvens, og den anden er optimering efter referencefrekvenser. Under optimering for reference frekvenser kan den endelige topologi indeholde elementer hvis densitet ligger mellem nul og et (grå elementer), og er uønskede i den endelige topologi. Derfor indføres et mål for løsningens diskretion, det vil sige hvor mange grå elementer der er til stede i modellen, og ved at inkludere dette som en restriktion kan et design med flere diskrete elementer opnås.

Efter den grundlæggende teori er etableret kan koden udvikles. Dette er foretaget i MATLAB, hvor ANSYS kan benyttes som løser til egenværdiproblemet. Både en 2D og 3D version af koden er implementeret. Da 2D problemener er hurtigere at løse end 3D problemener valideres implementeringen af koden i 2D, og efterfølgende løses et simpelt designproblem ved benyttelse af 3D koden.

For at validere koden benyttes bjælkeproblemener af "akademisk natur". Gennem disse er indflydelsen af det valgte startgæt, maksimal ændring af design variable, materiale restriktion, materialeinterpolering samt element valg undersøgt. Slutteligt er det valideret, at

topologier som rammer de ønskede frekvenser og svingningsformer med et lavt antal grå elementer kan opnås.

Slutteligt er teststanden forsøgt udviklet. Dette er gjort på baggrund af en analyse af den udleverede vindmølle model, hvor de ønskede egenfrekvenser samt svingningsformer er fundet. Frekvenserne er fundet til 0.174 samt 0.263 [Hz].

Tårnet på den udleveret model udskiftes med teststandens design domæne. Gennem analyser ses det, at med en elementstørrelse på 100 [mm] er det ikke muligt at komme under 5.5 og 6.5 [Hz] i referencefrekvenser.

Selvom det ikke er muligt at opnå de lave frekvenser er det stadig vist at det udviklede program kan frembringe designs der har et ønsket respons, hvor både egenfrekvenser og svingningsformer er specificeret fra start.

Contents

1	Introduction	1
1.1	Project Framework	2
1.2	Organization of Thesis	5
2	Definition of the Mathematical Problem	7
2.1	Topology Optimization	7
2.2	Eigenvalue Problem	11
2.3	Mode Tracking	15
2.4	The Optimization Problem Formulation	16
2.5	Filtering	19
2.6	Measure of Discreteness	20
3	Presentation of the Code	23
3.1	Overall Flow of the Code	23
3.2	Input to the Code	25
3.3	Preprogrammed Boundary Conditions	29
4	Code Development and Verification	31
4.1	Standard Configurations for the Analyzes	31
4.2	Verification of Sensitivities	32
4.3	Number of Eigenpairs Included	36
4.4	Effect of Penalization Constants	38
4.5	Move Limit	39
4.6	Start Guess and Resource Constraint	41
4.7	Mass and Stiffness Interpolation	45
4.8	Mass Matrix	54
4.9	Choice of Element	56
4.10	Filtering	58
4.11	Optimizing for Specific Frequencies	61
4.12	Eliminating Gray Elements	64
4.13	Mode Tracking	69
4.14	3D Implementation of the Code	72
5	Design of Test Rig for Wind Turbine Nacelle	77
5.1	Presentation and Analysis of the Provided Model	77
5.2	Design of Test Rig	83

6 Conclusion	89
6.1 The Developed Code	89
6.2 Design of Test Rig	92
6.3 Further Work	92
Bibliography	95
A The Method of Moving Asymptotes	I
B Adavnced Start Guesses	IV
C Artificial Material	VI
Nomenclature	IX

CHAPTER 1

Introduction

The process of designing structures has been developed over centuries. Inspired by nature (and e.g. economical/resource aspects) a continuous search for improved (optimized) structures has always been carried out. Based on engineering experience and “trial and error” structural optimization started out as an “art”, from the analytical work by Michell, Prager and Rozvany structural optimization has developed into continuum material density methods by e.g. Bendsøe and Kikuchi, where the SIMP type schemes are dominating the field today, see references in Bendsøe and Sigmund [2003]. Present the finite element method is used in conjunction with optimization providing new possibilities for the use of optimization.

Structural optimization has developed into a wide field of methods, techniques and commercial optimization programs such as TOSCA and OptiStruct are available. Now optimization is widely used as a design tool in the industry. Due to a strong competition the need for faster development and optimized (improved) products is greater than ever. Optimization tools are being developed for a vast type of problems in all engineering fields, and the ability to produce products with specific properties are increasing. Now demands for tools which can produce designs with specific eigenfrequencies, mode shapes and strain states are requested.

The proposal for the thesis is made in collaboration with the Danish division of the Swedish consultancy company FS Dynamics AB. On basis of this collaboration one application of the previously described tool is found. It is desired to conduct full scale testing of a wind turbine nacelle without mounting it on the tower. Advantages are, that it is easier to mount and test the nacelle in e.g. four meters height compared to the 90 [m] of the tower. Furthermore this increases the accessibility when testing the nacelle. From this the objective is to design a test rig which provide the possibility for full scale fatigue testing of a wind turbine nacelle. Mounted on the test rig, the nacelle should be able to experience real-life in service loads.

From FS Dynamics the in service loads are defined by a set of eigenfrequencies, mode shapes and a strain state in the nacelle. The frequencies are given as a set of distinct (simple) eigenfrequencies.

The test rig is limited to a much smaller volume than the tower, see Figure 1.1 and 1.2. FS Dynamics has specified the maximum dimensions for the test rig, which are (height \times width \times depth): 4000 \times 5000 \times 6500 [mm]. The purpose of the test rig is to provide the possibility of testing the nacelle without mounting it on a tower.

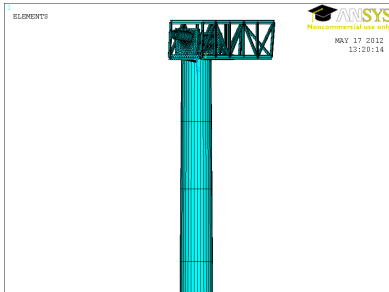


Figure 1.1: Top section of the provided structure from FS Dynamics.

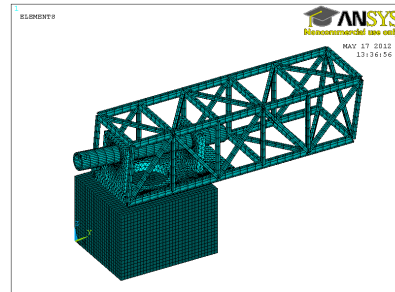


Figure 1.2: The nacelle on top of the maximum permitted design space.

To the authors knowledge it is possible to specify a upper and lower boundary for the frequencies and track a mode, in commercial optimization software such as TOSCA. However it is desired to obtain frequencies as close to the desired frequencies as possible while tracking the mode shapes and being able to associate a specific mode to a specific frequency. Since it is unclear whether this is possible or convenient in the commercial software it is decided to develop a specialized program which is capable of handling the problem at hand.

During the development of the code it is seen, that the task of matching eigenfrequencies and mode shapes is not trivial and it is chosen to focus on this area, thus neglecting the strain state in the nacelle due to lack of time in the project period. On basis of this the following project framework is established.

1.1 Project Framework

On basis on the above problem the framework for the project is established. The framework will be used to outline and restrict the approach taken to develop the optimization tool in general terms. The first step is to present the problem statement for the project, which is given as:

Design a code which can match a specified number of eigenvalues and -modes to a set of predetermined eigenvalues and -modes.

This problem statement gives a broad window for approaches which can be used to solve this problem. The problem statement set up three main areas of interest:

1. Design an eigenfrequency optimization code.
2. Track the eigenmodes as the eigenfrequencies change throughout the optimization.
3. Implement an algorithm which makes the code able to match a given number of eigenvalues to a desired value with the desired eigenmodes.

The problem statement gives a broad definition of the area of interest. It is important to narrow this area in order to gain a more precise description of the objective for the problem, thus clarifying the overall approach taken to solve the problem at hand.

For structural optimization three different categories exist i.e. sizing, shape and topology optimization [Bendsøe and Sigmund, 2003]. These are depicted in Figure 1.3. From the figure it is seen, that a change in topology is only possible for topology and sizing optimization (if a sizing value of zero is permitted). For shape optimization a change in topology is not possible, thus one must have decided on the topology for the structure prior to the optimization, and on basis of a given topology find the shape which gives the optimum.

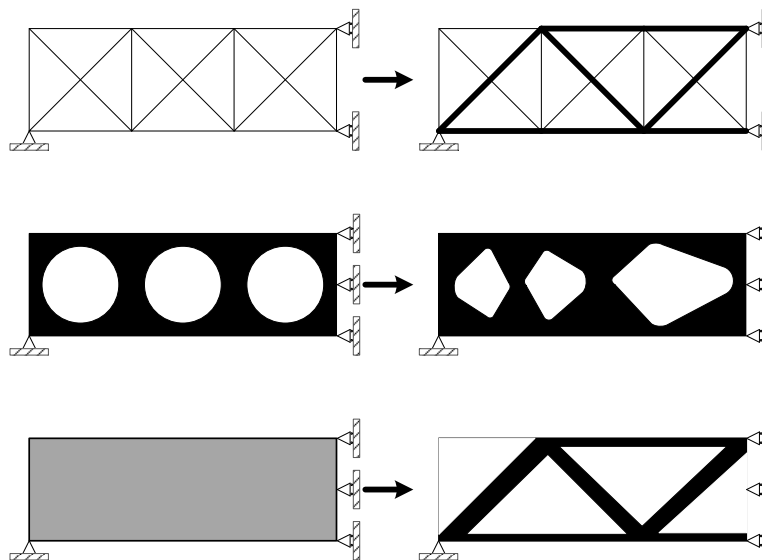


Figure 1.3: Illustration of the different categories of structural optimization. Top sizing optimization. Middle shape optimization. Bottom topology optimization

In sizing optimization all the possible connections for the domain must be predetermined and the set of sizing (e.g. cross sectional area) constants which optimizes the structure is found. Topology optimization is a material distribution method, where the objective is to find the optimum lay-out of material in the design domain. This includes information re-

garding the topology, shape and sizing of the structure making topology optimization the most general of the three since it is only bonded by the admissible design domain.

On basis of this topology optimization is chosen as the structural optimization category. For topology optimization the only necessary quantities are the applied loads, possible supports conditions and prespecified void or material domains [Bendsøe and Sigmund, 2003]. This gives the opportunity to have designs where only the most necessary parts are specified, thus keeping the largest design space.

From the above discussion a further reduction of the problem is conducted, by limiting to topology optimization, on basis of this the research objective is formulated:

Match a specified number of distinct (simple) eigenfrequencies and -modes to a desired set of frequencies and modes by making a topology optimization code.

The focus of this project is to develop a program, which is able to fulfill the research objective. Whereas the problem proposed by FS Dynamics act as a problem from the industry which the developed program can be used to solve.

A code is developed in MATLAB, which will have the ability to interact with the commercial finite element program ANSYS in order to have a more versatile solver, and have the ability to model more complex structures. The optimization is conducted in MATLAB with initial analyzes in 2D before 3D analyzes are initiated. For simple 2D problems MATLAB is used as a solver to validate the code.

As implied the combination of MATLAB and ANSYS will use a finite element (FE) discretization. This approach of discretizing the design domain into finite elements is typically used in conjunction with topology optimization, since a continuum approach will be difficult to formulate in a general manner. When using a discrete representation of the design domain, a black-white raster representation of the design domain defining the optimal topology by the finite element representation is sought [Bendsøe and Sigmund, 2003]. This means, that the optimal solution for a given problem is limited to the quality of the discretization, and a finer discretization gives the possibility for a better representation of the design domain, thus a solution closer to the continuum solution is found.

1.2 Organization of Thesis

The thesis is organized as follows: Chapter 2 gives a summary of the theory used to design the test rig. In chapter 3 a presentation of the developed code is given. This chapter gives a qualitative description of the codes to provide an overview of the code in order to ease the understanding of the choices made during the implementation.

Chapter 4 provides a deeper description of the codes where an investigation of the different parts and a verification of these are conducted. This will mainly be conducted in the 2D code, since 3D is a simple extension of this, and by limiting to a 2D verification the models are kept much smaller, thus they are faster to calculate.

In chapter 5 an example of application of the code is given. Here the topology for a test rig for a wind turbine nacelle is found. The volume of the test rig is limited to be much smaller than a wind turbine tower and a design based on the developed code is found. This design will be conducted in 3D.

Lastly in chapter 6 the findings are summed up.

CHAPTER 2

Definition of the Mathematical Problem

In this chapter the theory needed to solve the optimization problem is established. The mathematics described in this chapter provides the body of the developed code. It is chosen to describe the steps taken in stead of mathematical derivations, since it provides a good overview of the theory.

2.1 Topology Optimization

The problem is solved using topology optimization. This is a material distribution method where the optimum topology is sought. The "lay-out" of the structure includes information on the topology, shape and sizing of the structure. Topology optimization allows for addressing all three problems simultaneously, as described in Bendsøe and Sigmund [2003].

Most problems in topology design are not convex. Furthermore multiple optima exist for many problems, meaning there exist no unique solution to the problem. The final topology may change depending on the starting guess and the configuration of the optimization algorithm [Bendsøe and Sigmund, 2003].

One simple method of demonstrating this is by designing a rod in tension. It is known that the maximum load capability of the rod is defined by the cross sectional area. This means, that one can use a solid rod, a tube, an I-profile etc. and as long as the cross sectional area is the same then it is the optimum solution to the problem, thus providing in principle an infinite number of optima. This property of the optimization will be shown for the current problem in chapter 4. When targeting a set of specified eigenvalues local minima can be found as the solution. However, this is not a problem, since a design which fulfills the requirements is desired, and not necessarily the exact global optimum.

The final topology should contain a 0(void)-1(material) design which is a discrete valued (or "integer") problem. The discrete valued problem is not differentiable, therefore the optimization formulation is relaxed by creating continuous variables often referred to as densities, x_i . Since a finite element representation of the geometry these densities are the element densities. In order to obtain a discrete solution the most commonly used approach is to penalize the mass and stiffness such that the element densities steer towards either 0 or 1, thus avoiding intermediate densities.

In order to make it advantageous for the element to obtain either a zero or one density an

interpolation scheme favoring 0 and 1 densities must be chosen. For the stiffness matrix this is done by penalizing the Young's modulus such the stiffness is low compared to the density of the element. One approach is the SIMP (Solid Isotropic Material with Penalization) stiffness interpolation described in [Bendsøe and Sigmund, 2003].

$$E(x) = x^p E_0, \quad p > 1, \quad 0 < x_{min} \leq x \leq 1 \quad (2.1)$$

Here x is the density of the element, p is the penalization factor and E_0 is the Young's modulus for the solid isotropic material. Even though a 0-1 design is desired a zero density in equation (2.1) will result in computational difficulties due to the stiffness matrix becomes singular, thus a minimum density, x_{min} , is specified to overcome this problem. The SIMP interpolation is illustrated on Figure 2.1 for various penalties. From Bendsøe and Sigmund [2003] it is seen, that in order to have a good 0-1 design a sufficiently high penalty factor must be chosen. Andreassen et al. [2011] states, that a penalty factor of 3 typically is selected.

Another stiffness interpolation is the RAMP (Rational Approximation of Material Properties) given by [Rietz, 2001] and [Stolpe and Svanberg, 2001]:

$$E(x) = \frac{x}{1 + q(1 - x)} E_0 \quad (2.2)$$

It is seen that the stiffness is described as a term varying with the density x , multiplied with

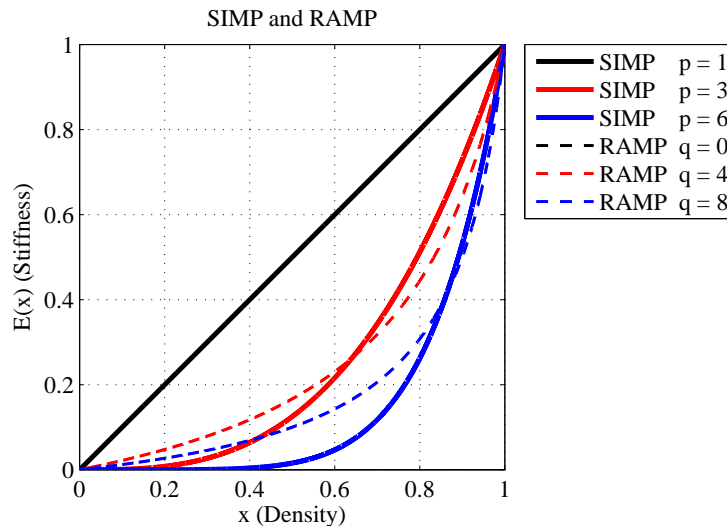


Figure 2.1: SIMP and RAMP stiffness interpolations for various penalties.

the Young's modulus for full material. The RAMP interpolation is seen to have a nonzero gradient for x approaching zero as illustrated on Figure 2.1. The advantage of this will be shown in the following for eigenvalue problems.

Since a FE based approach is used the interpolation scheme must be incorporated into the stiffness and mass matrix. As seen in equation (2.1) and (2.2) it is possible to extract the penalization function from the element matrices. The element stiffness matrix can be calculated as:

$$[\mathbf{k}_e] = E(x) [\bar{\mathbf{k}}_e^0] \quad (2.3)$$

$[\bar{\mathbf{k}}_e^0]$ is the element stiffness matrix with a unit Youngs modulus. A similar approach is taken for the mass matrix:

$$[\mathbf{m}_e] = W(x) [\bar{\mathbf{m}}_e^0] \quad (2.4)$$

Here $W(x)$ is the penalization function for the mass and $[\bar{\mathbf{m}}_e^0]$ is the mass matrix obtained using a unit value for mass density ρ . Thus it is also possible to extract the penalty function for the mass matrix. Choosing the simplest interpolation function i.e. $W(x) = x\rho$ results in problems when using the SIMP material function. With inspiration of Rayleigh's coefficient a measure of the local (element) frequency is estimated. Hansen [2005] defines a measure of the inverse Rayleigh coefficient as:

$$\Omega = \frac{W(x)}{E(x)}$$

Inserting into the equation gives:

$$\Omega(x) = \frac{W(x)}{E(x)} = \frac{x\rho}{x^p E_0} = \frac{1}{x^{p-1}} \frac{\rho}{E_0}, \quad p > 1 \quad (2.5)$$

which clearly goes towards infinity for x approaching zero as illustrated on Figure 2.2. This introduce a phenomena in vibration problems known as local modes or "artificial modes" in low density regions. If the RAMP interpolation is used it is seen, that the value is limited to $q+1$ times the material properties, where q is the penalty, see equation (2.6). Thus the local modes are avoided.

$$\Omega = \frac{W(x)}{E(x)} = \frac{x\rho}{\frac{x}{1+q(1-x)} E_0} = (q+1 - qx) \frac{\rho}{E_0}, \quad q > 1 \quad (2.6)$$

In order to avoid the localized modes, Ω must be bounded (or zero) when x approaches zero. This gives the low density elements a finite (infinite when $\Omega = 0$) frequency which is higher than the global, thus avoiding localized modes. For the SIMP interpolation this can be accomplished using the mass interpolation schemes from [Du and Olhoff, 2007]. In the Olhoff-Du interpolation $x > 0.1$ is interpolated linearly and the following interpolations are

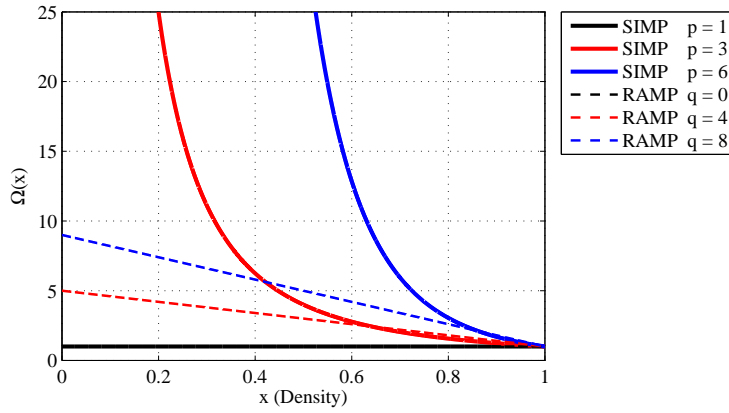


Figure 2.2: Inverse Rayleigh’s coefficient with linear mass interpolation and SIMP and RAMP stiffness interpolations for various penalties and $\frac{\rho}{E_0} = 1$.

used for $x \leq 0.1$.

$$W(x) = x^6 \rho \quad \text{Olhoff-Du(a)} \quad (2.7)$$

$$W(x) = 10^5 x^6 \rho \quad \text{Olhoff-Du(b)} \quad (2.8)$$

$$W(x) = (6 \cdot 10^5 x^6 - 5 \cdot 10^6 x^7) \rho \quad \text{Olhoff-Du(c)} \quad (2.9)$$

It is seen on Figure 2.3, that Olhoff-Du(a) is discontinuous at $x = 0.1$, whereas Olhoff-Du(b) and Olhoff-Du(c) are C^0 and C^1 continuous, respectively. The effects of this are investigated in section 4.7. Comparing to equation (2.5) it is seen, that when using these mass interpolations $\Omega \rightarrow 0$ as $x \rightarrow 0$, thereby avoiding the localized modes.

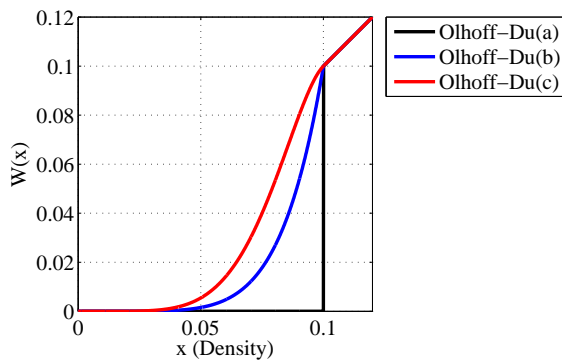


Figure 2.3: The Olhoff-Du mass interpolations from equation (2.7) to (2.9).

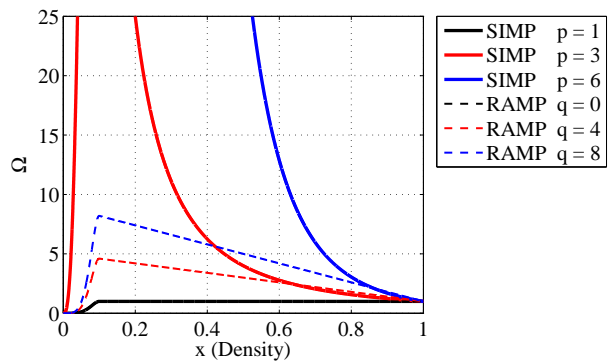


Figure 2.4: Inverse Rayleigh’s coefficient with the Olhoff-Du(c) mass interpolation, using SIMP and RAMP stiffness interpolation for various penalties and $\frac{\rho}{E_0} = 1$.

In Figure 2.4 a plot of the inverse Rayleigh coefficient when using Olhoff-Du(c) as mass interpolation and either SIMP or RAMP as stiffness interpolation for varying penalization

factors. Raising the penalty factor yields a steeper gradient in both high and low density areas. Furthermore a kink is seen at $x = 0.1$ which is due to the mass formulation changes at that density.

It is also seen, that Ω for the SIMP interpolation generally are higher than for RAMP, and attains a high value around 0.1 density. A too high penalty factor for SIMP stiffness interpolation local modes will occur, which is as expected, since both the stiffness and mass interpolations are power functions, thus the function with the highest power defines whether or not local modes occur, and it is seen, that the lowest power in the mass interpolation must be higher than the power for the stiffness interpolation.

Localized modes can also be present for densities around $x = 0.1$ when using the Olhoff-Du interpolations even if the penalty for SIMP stiffness interpolation is lower than 6. This can be seen in Figure 2.4 where Ω attains a high in this area, e.g. if $p = 3$ then $\Omega = 100 \frac{\rho}{E_0}$, which may cause a localized mode. During the development and testing of the code it is observed, that the phenomena of localized modes is more complex than described in this section. This is further discussed in section 4.7.

2.2 Eigenvalue Problem

In chapter 1 the problem is defined as a vibrational problem. The governing equation for the problem is given as:

$$[\mathbf{M}] \{\ddot{D}\} + [\mathbf{K}] \{D\} = \{0\} \quad (2.10)$$

Here it is assumed, that the structure is undamped and under free vibrations. The solution to equation (2.10) has the form $\{D\} = \{\phi\} \sin(\omega t)$. Inserting this into equation (2.10) and rearranging the state equation for the problem is found.

$$[\mathbf{K}] \{\phi\} = \lambda [\mathbf{M}] \{\phi\} \quad (2.11)$$

Here $\lambda = \omega^2$ is the eigenvalue. Together with the corresponding eigenvector $\{\phi\}$ an eigenpair is formed. $[\mathbf{K}]$ and $[\mathbf{M}]$ are the global stiffness and mass matrix respectively, which are symmetric and positive definite [Cook et al., 2002]. This ensures that the eigenvalues will be positive thus no imaginary terms are present for the frequencies [Zienkiewicz and Taylor, 2005]. Symmetry will also be used in section 2.2.1 Equation (2.11) can be reshaped into the eigenvalue problem given as [Rao, 2004]:

$$\left([\mathbf{K}] - \lambda_j [\mathbf{M}] \right) \{\phi\}_j = \{0\}, \quad j = 1, \dots, N \quad (2.12)$$

Here N is the number of eigenpairs extracted. Since a non trivial solution is desired the determinant of the coefficient matrix, $([\mathbf{K}] - \lambda[\mathbf{M}])$, must be zero. This gives a polynomial in λ where the eigenvalues can be obtained. For continuum structures an infinite number of eigenpairs exist, however for FE problems one eigenpair exist for each degree of freedom, thus the number of eigenpairs is large. Since only the N lowest eigenpairs are needed it is computationally inefficient to calculate all, thus solution algorithms which only extract the needed eigenvalues are used to make the algorithm computational efficient. Furthermore the eigenvalues are ordered by magnitude as:

$$0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_j \leq \dots \leq \lambda_N$$

In the following it is assumed, that the eigenvectors $\{\phi\}$ are $[\mathbf{M}]$ orthonormalized, which means:

$$\left\{ \phi \right\}_k^T [\mathbf{M}] \left\{ \phi \right\}_j = \delta_{jk}, \quad j, k = 1, \dots, N \quad (2.13)$$

Where δ is the Kronecker's delta.

From equation (2.11) it is seen by premultiplying with $\{\phi\}_k^T$ and using equation (2.13), that:

$$\left\{ \phi \right\}_k^T [\mathbf{K}] \left\{ \phi \right\}_j = \lambda_j \delta_{jk}, \quad j, k = 1, \dots, N \quad (2.14)$$

Thus the eigenvectors $\{\phi\}$ are also $[\mathbf{K}]$ orthogonal.

2.2.1 Design Sensitivity Analysis of Simple Eigenvalues

In the following it is assumed, that only simple eigenvalues are present. This is done since the problem proposed by FS Dynamics provides a set of distinct eigenvalues, see chapter 5, thus the necessity for taking multiple eigenvalues into account is not present making the analysis for multiple eigenvalues is not needed during the analyzes and will be omitted.

When multiple eigenvalues are present, the problem in equation (2.12) can be solved using any linear combination of the eigenvectors, the problem becomes nondifferentiable in the common sense, and measures described in e.g. [Seyranian et al., 1994] and [Lund, 1994] must be taken fore the sensitivity analysis.

Since the element densities are used as design variables the derivatives with respect to these are needed. In order to obtain the design sensitivities equation (2.12) is differentiated

with respect to the design variables x_i , thus obtaining, [Lund, 1994]:

$$\frac{\partial [\mathbf{K}]}{\partial x_i} \{\phi\}_j + \left([\mathbf{K}] - \lambda_j [\mathbf{M}] \right) \frac{\partial \{\phi\}_j}{\partial x_i} - \left(\frac{\partial \lambda_j}{\partial x_i} [\mathbf{M}] + \lambda_j \frac{\partial [\mathbf{M}]}{\partial x_i} \right) \{\phi\}_j = \{0\} \quad (2.15)$$

Here $j = 1, \dots, N$ and $i = 1, \dots, NE$. NE is the number of elements (design variables) in the model. Premultiplying equation (2.15) by $\{\phi\}_j^T$ and rearranging yields:

$$\begin{aligned} \frac{\partial \lambda_j}{\partial x_i} \{\phi\}_j^T [\mathbf{M}] \{\phi\}_j = \{\phi\}_j^T \left([\mathbf{K}] - \lambda_j [\mathbf{M}] \right) \frac{\partial \{\phi\}_j}{\partial x_i} \\ + \{\phi\}_j^T \left(\frac{\partial [\mathbf{K}]}{\partial x_i} - \lambda_j \frac{\partial [\mathbf{M}]}{\partial x_i} \right) \{\phi\}_j \end{aligned} \quad (2.16)$$

From equation (2.13) $\{\phi\}_j^T [\mathbf{M}] \{\phi\}_j = 1$. Since $[\mathbf{M}]$ and $[\mathbf{K}]$ are symmetric it is seen from equation (2.12) that the first part of the right hand side equals zero leaving.

$$\frac{\partial \lambda_j}{\partial x_i} = \{\phi\}_j^T \left(\frac{\partial [\mathbf{K}]}{\partial x_i} - \lambda_j \frac{\partial [\mathbf{M}]}{\partial x_i} \right) \{\phi\}_j \quad (2.17)$$

From equation (2.17) the sensitivities are found. These equations can be further reduced using equation (2.3) and (2.4). Based on these equations it is seen, that no coupling between the different densities is present, thus equation (2.17) can be rewritten to:

$$\frac{\partial \lambda_j}{\partial x_i} = \{\bar{\phi}\}_{ji}^T \left(\frac{\partial [\mathbf{k}_e]_i}{\partial x_i} - \lambda_j \frac{\partial [\mathbf{m}_e]_i}{\partial x_i} \right) \{\bar{\phi}\}_{ji} \quad (2.18)$$

Here $\{\bar{\phi}\}_{ji}$ is the part of eigenvector j whose entries match element i . This reduces the system of equations substantially since all the sensitivities can be calculated on element basis.

2.2.2 Design Sensitivity Analysis Using ANSYS

When using a commercial finite element package like ANSYS it is not possible to extract the sensitivities of the stiffness and mass matrices directly. One method is to use the overall finite difference (OFD) where the design is perturbed in order to find the sensitivities. This

method is here shown as a first order forward difference:

$$\frac{\partial f}{\partial x_i} \approx \frac{f(x_1, \dots, x_i + \Delta x, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{\Delta x} \quad (2.19)$$

As it can be seen one function evaluation is needed for each design variable, thus this method is highly inefficient. A more efficient method can be found by examining equations (2.3), (2.4) and (2.18). By first looking at the stiffness term, it is seen, that the sensitivity is given as:

$$\frac{\partial [\mathbf{k}_e]}{\partial x} = E'(x) [\bar{\mathbf{k}}_e^0] \quad (2.20)$$

Here $E'(x) = \frac{dE(x)}{dx}$. Dividing and multiplying equation (2.20) by $E(x)$ the following equation for the sensitivity is obtained:

$$\frac{\partial [\mathbf{k}_e]}{\partial x} = \frac{E'(x)}{E(x)} E(x) [\bar{\mathbf{k}}_e^0] = \frac{E'(x)}{E(x)} [\mathbf{k}_e] \quad (2.21)$$

From equation (2.21) it is seen, that if the element stiffness matrices are extracted, then the sensitivities can be calculated. However the sensitivities can be found based on the energies, since $\{\bar{\phi}\}_{ji}^T [\mathbf{k}_e]_i \{\bar{\phi}\}_{ji} = 2U_{e,ij}$ where $U_{e,ij}$ is the element strain energy in element i due to a displacement field $\{\bar{\phi}\}_j$, it is seen, that the sensitivities of the stiffness matrix can be found from the strain energies of the model as:

$$\{\bar{\phi}\}_{ji}^T \frac{\partial [\mathbf{k}_e]_i}{\partial x_i} \{\bar{\phi}\}_{ji} = 2 \frac{E'(x_i)}{E(x_i)} U_{e,ij} \quad (2.22)$$

Similarly for the mass term it is seen, that it can be expressed based on the element kinetic energy, $T_{e,ij}$:

$$\frac{1}{2} \lambda_j \{\bar{\phi}\}_{ji}^T \frac{\partial [\mathbf{m}_e]_i}{\partial x_i} \{\bar{\phi}\}_{ji} = \frac{W'(x_i)}{W(x_i)} T_{e,ij} \quad (2.23)$$

Here $W'(x_i) = \frac{dW(x_i)}{dx_i}$. This result is obtained, since the element kinetic energy for mode $\{\bar{\phi}\}_{ji}$ is given as:

$$T_{e,ij} = \frac{1}{2} \{\dot{\bar{\phi}}\}_{ji}^T [\mathbf{m}_e]_i \{\dot{\bar{\phi}}\}_{ji}$$

$\{\dot{\bar{\phi}}\}_{ji}$ can be related to $\{\bar{\phi}\}_{ji}$ by the following equation:

$$\{\dot{\bar{\phi}}\}_{ji} = \omega_j \{\bar{\phi}\}_{ji}$$

Thus it can be seen, that:

$$T_{e,ij} = \frac{1}{2} \left\{ \dot{\bar{\phi}} \right\}_{ji}^T \left[\mathbf{m}_e \right]_i \left\{ \dot{\bar{\phi}} \right\}_{ji} = \frac{1}{2} \omega_j^2 \left\{ \bar{\phi} \right\}_{ji}^T \left[\mathbf{m}_e \right]_i \left\{ \bar{\phi} \right\}_{ji} = \frac{1}{2} \lambda_j \left\{ \bar{\phi} \right\}_{ji}^T \left[\mathbf{m}_e \right]_i \left\{ \bar{\phi} \right\}_{ji}$$

This means, that the sensitivities in equation (2.18) can be expressed using equations (2.22) and (2.23) as [Kim and Kim, 2000]:

$$\frac{\partial \lambda_j}{\partial x_i} = 2 \left[\frac{E'(x_i)}{E(x_i)} U_{e,ij} - \frac{W'(x_i)}{W(x_i)} T_{e,ij} \right] \quad (2.24)$$

This formulation for the sensitivity has the advantage, that the energies can be extracted from the finite element code. Thus the sensitivities can be found from two different equations i.e. (2.18) when the sensitivities of the stiffness and mass matrix are known, and (2.24) in terms of kinetic and strain energies. When validating the code it is possible to check whether or not the sensitivities are correct since the results from the OFD and the design sensitivity analysis (DSA), equations (2.18) and (2.24), must be the same.

2.3 Mode Tracking

During the optimization the topology changes from the initial to the final design. In addition the sequence of the eigenmodes can change e.g. the first eigenmode may switch from a bending to a breathing mode. Since specific eigenmodes can be desired for a given eigenfrequency a method to track the different eigenmodes is needed. [Kim and Kim, 2000] recommends using the Modal Assurance Criterion (MAC) to track the mode shapes as the topology of the structure changes. The MAC was originally developed to measure the difference between modes from measurements and numerical calculations. It can also be used to track the modes when the topology changes. The MAC value is calculated as in equation (2.25).

$$MAC \left(\left\{ \phi \right\}_j, \left\{ \phi_R \right\}_k \right) = \frac{\left| \left\{ \phi \right\}_j^T \left\{ \phi_R \right\}_k \right|^2}{\left(\left\{ \phi \right\}_j^T \left\{ \phi \right\}_j \right) \left(\left\{ \phi_R \right\}_k^T \left\{ \phi_R \right\}_k \right)} 100\% \quad \begin{array}{l} j = 1, \dots, N \\ k = 1, \dots, N_T \end{array} \quad (2.25)$$

Here N_T is the number of reference modes. $\left\{ \phi_R \right\}_k$ is the k'th reference eigenmode. It is seen that the MAC value gives a measure of the least squares deviation and is still a scalar quantity if a complex mode shape is given [Ewins, 2000].

It is seen, that the MAC gives a value between 0 (an entirely different mode) and 100 (the

same mode). Intermediate values are present as the topology changes since the overall mode is present, but the individual elements might not have the same relative displacement.

MAC has some nice properties: (1) The criterion is calculated using only vector operations, meaning that the criterion is fast to calculate, thus it is good to use in topology optimization. (2) If the two mode shapes differ by a scalar, i.e. $\{\phi\}_j = \gamma\{\phi_R\}_k$, it is seen, that $MAC(\{\phi\}_j, \{\phi_R\}_k) = 100$ as it should be since the mode shapes are perfectly correlated [Ewins, 2000].

When optimizing the structure to match modes to specific eigenvalues the Modal Assurance Criterion is used to adjust the constraint functions so the eigenvalues with the highest MAC values are used to target the desired eigenvalues.

2.4 The Optimization Problem Formulation

Based on chapter 1 it is seen, that a problem formulation which can handle several objective functions is needed, since several eigenpairs are extracted during the analyzes, and all of these are needed to be taken into account. In general two different methods exist to treat multiobjective optimization. One is scalar optimization, where the individual objective functions are weighted and summed up to a single scalar value. Weighted sum and weighted global criterion methods are examples of scalar optimization [Arora, 1999].

The other method is vector optimization where each objective function is treated individually [Arora, 1999]. One of these methods is a weighted min-max formulation (or equivalently a max-min formulation). The min-max formulation will be used to formulate the optimization problem. The advantage of this formulation is the focus on the "worst" of the objective functions, e.g. when maximizing the lowest eigenfrequency it is the lowest frequency that is of primary interest and when targeting e.g. two eigenfrequencies the largest deviation is of most interest.

The min-max (max-min) problem formulation focuses on the "worst" case, if e.g. the vector of deviation is used, the optimum could include one error which is large while the rest is small. When targeting eigenfrequencies it is still possible to add a weighting such e.g. the lowest frequency is of highest interest.

Here the min-max problem is formulated as:

$$\min_{x_i} (\max (w_j f_j(x_i))), \quad j = 1, \dots, N \quad (2.26a)$$

Subject to:

$$g_k(x_i) - \bar{g}_k \leq 0 \quad k = 1, \dots, K \quad (2.26b)$$

$$C(x_i) - \bar{R} \leq 0 \quad (2.26c)$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, NE \quad (2.26d)$$

$$\text{State equations} \quad (2.26e)$$

The different parts of equation (2.26) are defined as follows. Equation (2.26a) are the N objective functions f_j and the corresponding weightings w_j , which are to be minimized over a set of design variables x_i . The g_k terms in equation (2.26b) denotes global behavior constraints such as compliance or eigenvalue constraints. In equation (2.26c) a resource constraint is shown. This could be, for a material distribution problem, the amount of material present. The upper and lower bounds on the design variables are given in equation (2.26d) and lastly the state equations describe the governing equations of the structural problem.

Taking the basis on the problems at hand, i.e. topology optimization for eigenfrequency design, some parts can be identified. Equation (2.26a)

can be seen to be the equations containing the eigenvalues to be optimized e.g. maximizing the minimum eigenfrequency. Equation (2.26c) is the amount of material available for the structure and equation (2.26d) is the upper and lower bounds (1 and 0) for the design variables. Equation (2.26e) is the eigenvalue problem from equation (2.12). When targeting eigenvalues equation (2.26b) gives the lower bound for the eigenvalues not used to target specific values.

The convention is to solve optimization problem as minimization problems, thus the objective functions when e.g. maximizing the lowest eigenvalue has to be rewritten into a min-max formulation. Defining an objective function which has to be maximized (Λ_j) it is seen, that it can be transformed into a minimization formulation having the same optimum meaning:

$$\max(\min(\Lambda_j)) = \min(\max(f_j)) \quad (2.27)$$

Here the equality sign means the optimization parameters has the same value at optimum. This is achieved if for example $f_j = -\Lambda_j$ or $f_j = \Lambda_j^{-1}$. This gives two simple methods to convert a max-min problem into a min-max problem as equation (2.26).

2.4.1 The Bound Formulation

It is seen, that equation (2.26a) is not differentiable at all points, this is the case when the objective function switches between two different functions e.g. f_k and f_j , $j \neq k$. One method to solve this is by using the bound formulation [Olhoff, 1989]. The bound formulation replaces the non differentiable objective function by an artificial objective function in the additional variable β .

The min-max problem is interpreted as a simple min problem for the minimum of a bound value, where (the measures of) the criteria are related to the bound through constraints [Bendsøe et al., 1983].

The above quotation states that the bound formulation replaces the multiobjective function by a single objective function and the objective functions in equation (2.26a) are treated as constraints to the bound value.

Since linear programming will be used to solve the problem there is no reason not choosing the simplest convex descent function $\Psi(x_i, \beta) = \beta$, thus making the cost function differentiable. The statement in equation (2.26a) is transformed into constraints as:

$$\Psi(x_i, \beta) = \beta \geq w_j f_j, \quad j = 1, \dots, J \quad (2.28)$$

This means, that equations (2.26) now consist of additional constraints, thus making the min-max problem on the form:

$$\min_{x_i, \beta} \Psi(x_i, \beta) = \beta \quad (2.29a)$$

Subject to:

$$w_j f_j(x_i) - \beta \leq 0 \quad j = 1, \dots, J \quad (2.29b)$$

$$g_k(x_i) - \bar{g}_k \leq 0 \quad k = 1, \dots, K \quad (2.29c)$$

$$C(x_i) - \bar{R} \leq 0 \quad (2.29d)$$

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, NE \quad (2.29e)$$

$$\text{State equations} \quad (2.29f)$$

It is seen, that the formulation in equation (2.29) reflects a simple minimization problem, which reflects minimization of an upper bound β on all separate criteria f_j [Bendsøe et al., 1983]. From equations (2.29) it is seen, that in order to minimize β , then one must minimize f_j . This is due to no coupling exist between the design variables in both Ψ and f_j , so the x_i values minimizing f_j will also minimize Ψ .

In order to solve different optimization problems equation (2.29b) and (2.29c) are changed in accordance to the problem. For the two kind of problems solved the formulations are given in the following. Maximizing the lowest eigenvalue:

$$f_j(x_i) = C - \lambda_j \quad (2.30)$$

Here C is a positive constant large enough to make f_j positive. One example could be $C = 2\lambda_{max}$. When targeting N_T specific eigenvalue the eigenvalues are sorted according to the MAC. The N_T eigenvalues with the highest values are used to target the specified value, and is used in equation (2.31). The remaining eigenvalues are subjected to a constraint, thus moving them away from the highest value to be targeted, the lowest value which can be attained for these is denoted λ_{safety} .

$$f_j(x_i) = \left(\lambda_j - \lambda_j^{(t)}\right)^2 \quad j = 1, \dots, N_T \quad (2.31)$$

$$g_k(x_i) - \bar{g}_k = \lambda_{safety} - \lambda_k \quad k = N_T, \dots, N \quad (2.32)$$

It can be seen, that both of these formulations uses a min-max problem formulation.

2.5 Filtering

When using the standard density approach to topology optimization three insufficiencies are present. These insufficiencies are formation of checker-boards, one node hinges between elements and mesh dependency of the results. In order to avoid these insufficiencies mesh-independent filtering is used [Sigmund, 2007].

Two filters are implemented, these are Sigmunds sensitivity filter and a cone shaped density filter. These filters are implemented on basis of Sigmund [2001] and Andreassen et al. [2011].

In sensitivity filtering the main idea is to base the design update on filtered sensitivities in stead of the true sensitivities. The filtered sensitivities are given as:

$$\widehat{\frac{\partial \lambda_j}{\partial x_i}} = \frac{1}{x_i \sum_{k \in N_e} H_{ik}} \sum_{k \in N_e} H_{ik} x_k \frac{\partial \lambda_j}{\partial x_k} \quad (2.33)$$

Here N_e are the elements where the center to center distance between the elements i and k , $\Delta(i, k)$, is smaller than the filter radius. H_{ik} is a weight factor given as:

$$H_{ik} = \max(0, r - \Delta(i, k))$$

r is the filter radius. Sigmund [2007] states, that the filtered sensitivities correspond to the sensitivities of a smoothed version of the original objective function, even though the exact work-wise of the filter is not known.

The density filter works by modifying the element densities, thus the stiffness and mass of an element, to be a function of the densities in a neighborhood of an element. The filtered densities are given in equation (2.34).

$$\hat{x}_i = \frac{1}{\sum_{k \in N_e} H_{ik}} \sum_{k \in N_e} H_{ik} x_k \quad (2.34)$$

The filtered densities \hat{x}_i are in Andreassen et al. [2011] defined as the physical densities whereas x_i are the mathematical densities used in the optimization. This means, that the densities $\{x\}$ loses the physical meaning, and a distinction between the design variables $\{x\}$ and the physical densities $\{\hat{x}\}$ is made.

Since the stiffness and mass matrices are functions of the physical densities the eigenvalues from equation (2.12) are also a function of $\{\hat{x}\}$. Since a separation between the design variables $\{x\}$ and physical densities are made the sensitivities are not found directly. Using the chain rule the sensitivities of a function ψ with respect to the design variables are found [Andreassen et al., 2011]:

$$\frac{\partial \psi}{\partial x_i} = \sum_{k \in N_j} \frac{\partial \psi}{\partial \hat{x}_k} \frac{\partial \hat{x}_k}{\partial x_i} = \frac{1}{\sum_{k \in N_e} H_{ik}} \sum_{k \in N_e} H_{ik} \frac{\partial \psi}{\partial \hat{x}_k} \quad (2.35)$$

Here ψ can represent both the objective functions (λ_j) or the material volume, since the filtered densities must fulfill the volume constraint. It is important to notice, that the H -terms in equation (2.35) do not represent a filtering of the sensitivities, these terms are obtained by differentiation of equation (2.34).

2.6 Measure of Discreteness

In order to tell whether or not a found solution is a discrete design a measure of the discreteness is required. When maximizing the lowest frequency a measure of discreteness can also be used in the comparison. For targeting a specific eigenvalue some of the solutions might be a nondiscrete solution, thus the measure of discreteness can be used to assess the amount of discreteness.

Sigmund [2007] suggests a method to measure the discreteness. This measure is modified in order to have the ability to get a discreteness value of zero since the densities used in this

work is not allowed to attain a value of zero.

$$MoD = \sum_{i=1}^{NE} \frac{4(x_i - x_{min})(1 - x_i)}{NE} 100\% \quad (2.36)$$

Here MoD is the measure of discreteness. From equation (2.36) it is seen when all elements has a density 0.5 $MoD \approx 100\%$, thus a fully gray design is present. If all elements are either 1 or x_{min} then $MoD = 0\%$ and a fully discrete design is obtained.

Naturally the measure of discreteness needs to be as low as possible when the algorithm has converged, since it can be used as a measure of the “quality” of the solution.

CHAPTER 3

Presentation of the Code

After the theory required to solve the problem is established the next step is to implement this into a code which can conduct the eigenfrequency optimization, both for targeting specific eigenvalues, but also maximizing eigenvalues.

This chapter forms together with with chapter 4 a description of the properties of the developed code. In chapter 3 the code is presented in order to provide an overview of the code, the flow of the code and how to control the code, whereas chapter 4 validates the implementation of the different parts of the code.

As stated in chapter 1 it is chosen only to show the 2D implementation of the code. 3D is simply an extension of 2D. The 2D code has the advantage, that iterations are faster to complete, and results obtained from a 2D simulation are easier to interpret and visualize than the results from a 3D. Since the 3D code uses ANSYS as solver the flow of the code is similar to the 2D code with ANSYS as solver.

The code is based on Sigmund [2001] and Andreassen et al. [2011]. These contain an implementation of topology optimization for minimum compliance design, and provides the basis on which the eigenfrequency code is implemented.

For the code two different methods of extracting the eigenpairs have been implemented, namely the eigs-function in MATLAB and modal analysis in ANSYS. These two versions contain minor differences, and will be referred to MATLAB and ANSYS code respectively.

After the flow of the code is described the input options for the code are presented, and lastly the preprogrammed boundary conditions for the 2D codes are shown.

3.1 Overall Flow of the Code

The code is described in several steps, this is done in order to first provide an overall description of the code. Afterwards the two different solvers are described more detailed. In Figure 3.1 the steps in the code is seen together with the steps where the different optimization parameters are updated.

The first part of the code defines the configuration of the problem, including the user defined input described in section 3.2 with boundary conditions and design domain for the

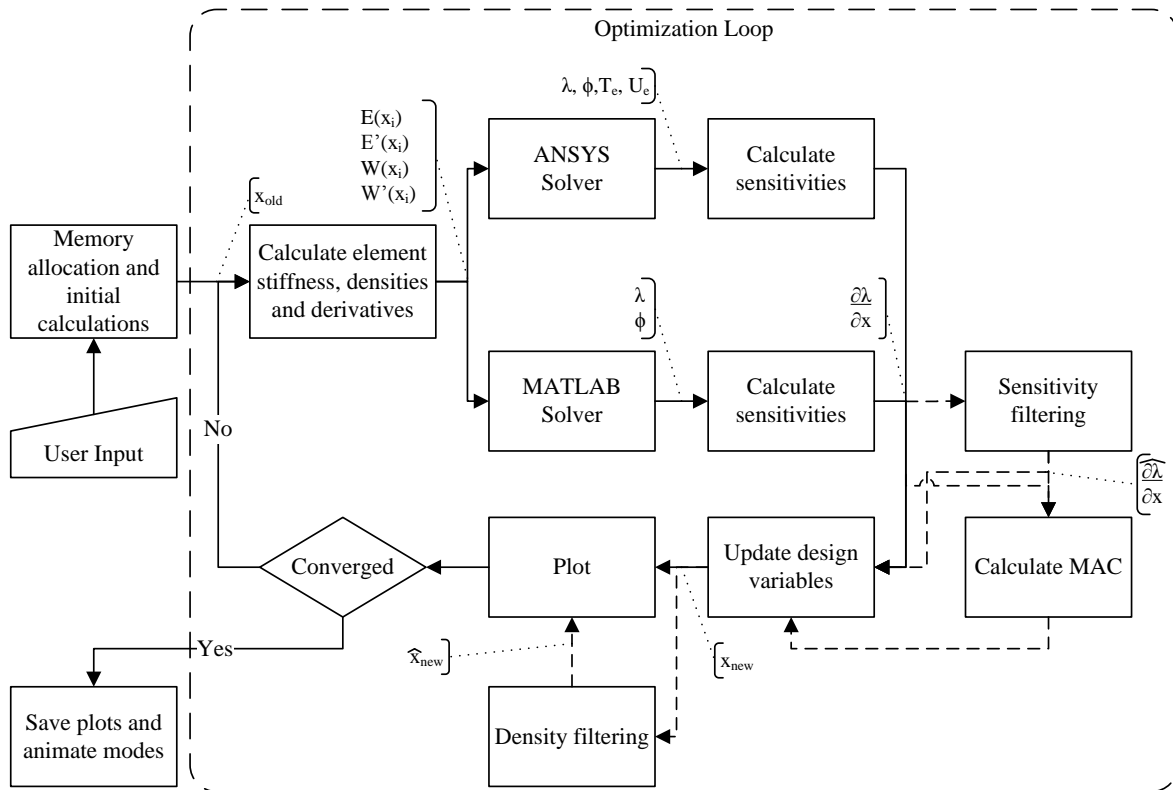


Figure 3.1: Overall flow of the code. The most important output are shown in brackets.

problem. Furthermore memory is allocated to speed up the optimization. In this part of the code one minor difference between the two solvers is present. In the ANSYS version a boundary condition file is executed in order to get a file containing the geometry described by the elements, thus making it possible to control the stiffness and density on element level in stead of structural level, since material properties must be defined for each element.

In the optimization loop the difference between the codes is the choice of solver, besides this the codes are similar. Using the equations in section 2.1 the element stiffnesses and densities are calculated, these are then used to form the global stiffness and mass matrices in either MATLAB or ANSYS to solve equation (2.12). After calculating and perhaps filtering of the sensitivities and mode tracking the optimization problem is solved. If mode tracking is used the MAC values are calculated and the eigenpairs with the highest MAC values are used in equation (2.31) whereas the remaining eigenfrequencies are used in equation (2.32).

In the design update step the Method of Moving Asymptotes (MMA) algorithm by Krister Svanberg [Svanberg, 1987]. An implementation made by Krister Svanberg can be obtained for academic purposes. This implementation of the algorithm is directly used.

MMA has a slightly different method for setting up the bound formulation than presented in section 2.4.1, this is shown in appendix A and Svanberg [2007].

If a density filter is used the design variables are filtered before the intermediate results are plotted since the design variables are not the physical densities as described in section 2.5, and as the last step in the optimization loop a convergence check is conducted.

The last step in the code after convergence is to save the plots and if desired animate the mode shapes.

Referring to Figure 3.1 the two boxes ANSYS and MATLAB solver are here described in more details. The first step in the MATLAB solver is to calculate the global stiffness and mass matrices, which are afterwards used in the eigs-solver, which can extract a desired number of eigenpairs to be used in the sensitivity analysis.

The ANSYS solver is shown in Figure 3.2. Here the ANSYS part of the code reads the stiffness and density files created in MATLAB and conduct a modal analysis, where the results are read in MATLAB in order to calculate the sensitivities.

After the code is presented, the user input to the code is given in the following section.

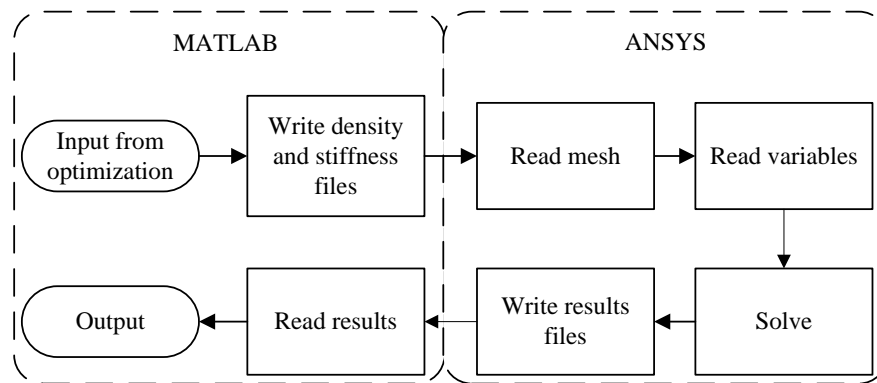


Figure 3.2: Steps in the ANSYS solver

3.2 Input to the Code

The code is controlled through an input file. This file is split up into two parts. The first part of the input file defines the general set up of an analysis, whereas the second part contain input related to different implementations of e.g. sensitivity calculations.

The first part of the input file contain the input where the general set up of the analyzes are defined, see Table 3.1. These setting controls the maximum number of iterations, number of elements and eigenpairs extracted. The solver and boundary conditions are also defined in this part of the input file. It is possible to select whether a maximization of the fundamental frequency or target specific frequencies is desired. Furthermore the stiffness and mass interpolation formulations are defined.

The plotting routine and if the modes are to be animated is also defined here.

Table 3.1: Input to the optimization code which are often used.

Name in input file	Description
VolumeFraction	Admissible amount of material, value between 0 and 1
meshsize	Size of elements
Neig	Number of eigenpairs extracted
maxIter	Maximum number of iterations
change_stop	Stop criteria
move_limit	Maximum change for the design variables per iteration
Plot_optimization_vector	Desired plots per iteration
Plot_after_optimization_vector	Plots to be saved
Animate_modes	Animate modes after optimization
Solver_type	MATLAB or ANSYS solver used
ANSYS and ANSYS_path	ANSYS home folder
BC_N	Boundary conditions, see section 3.3
Opt_type	Optimization algorithm used
TargetEigenvalue	Eigenfrequencies to target
Safetyzone	Size of safety zone
WeightTarget	Constraints weight
Discreteness_Target	Target measure of discreteness in percent
MAC	Enable or disable mode tracking
Filter_type	Sensitivity or density filter
filter_r	Filter radius in element lengths
Stiffness_Interpol	Definition of stiffness interpolation function
pK	Penalty for SIMP or RAMP stiffness interpolation
minDensity_e	Minimum element density
Mass_Interpol	Mass interpolation function
pM	Penalty for mass interpolation

The second part of the input to the code are input which in general does not affect the final topology of the optimization, see Table 3.2. The input are primarily used when validating

a faster implementation. However two entries in these input can have an interest, these are the MMA_c_const which controls the c constants in the MMA algorithm, see appendix A, and the Element_type which controls the element used in the ANSYS models, a further description of these are given in section 3.2.1.

Table 3.2: Input to the optimization code which are not used.

Name in input file	Description
conv_check	Convergence check function
tolerance	Tolerance in the optimality criteria
Reduction	Trust region reduction factor for SLP
MMA_c_const	c constant values for MMA
Element_type	Element used in ANSYS
Mass_function	Consistent or lumped mass matrix
Global_Mass_Stiffness_function	Algorithm used to assemble mass and stiffness matrix
Sensitivity_function	Sensitivity calculation algorithm

Furthermore the mass formulation (consistent or lumped) is defined in this part. This is due to all elements are axis parallel, thus the lumped mass formulation provides a good representation of the design domain.

3.2.1 Available Elements

The elements available in the code can be grouped into three different categories: MATLAB, ANSYS 2D and ANSYS 3D elements. A short introduction to the elements is given in the following. For the ANSYS elements a more elaborate description is available in the ANSYS Element Reference Manual [ANSYS Inc., 2010].

If the MATLAB code is used only one element is available. This is a 2D solid four node bilinear element. The bilinear formulation of the element is sufficient since the elements used in the models are rectangular, and the edges are axis parallel thus fulfilling the compatibility conditions [Cook et al., 2002]. This element is directly implemented on basis of Sigmund [2001]. In the following this element is referred to as MATLAB Q4.

In Table 3.3 the elements available in the 2D ANSYS code are shown. Names from ANSYS are used to clarify which element is used in the optimization. The elements are included to give the ability of investigating the effect different element formulations.

Table 3.3: Elements implemented in the 2D ANSYS code.

Element name	Number of nodes	Element technology
PLANE42	4	
PLANE182	4	Full integration with B-bar method
PLANE42QM6	4 with extra displacement shapes	
PLANE182QM6	4 with 5 internal degrees of freedom	Enhanced strain formulation
PLANE82	8	
PLANE183	8	

Eight elements are available in the 3D code. These are seen in Table 3.4. As for the 2D code the names from ANSYS are used.

Table 3.4: Elements implemented in the 3D code.

Element name	Number of nodes	Element technology
SOLID45	8	Full integration
SOLID45RI	8	Uniform reduced integration with hourglass control
SOLID185	8	Full integration with B-bar method
SOLID185ES	8	Enhanced strain formulation
SOLID95	20	Full integration
SOLID95RI	20	Reduced integration
SOLID186	20	Full integration
SOLID186RI	20	Uniform reduced integration

It is recommended to use the elements with the simplest formulation, this is due to the difference when using a more advanced element formulation is negligible as seen in section 4.9, thus the cheapest elements with respect to computation time are recommended. Furthermore since the model will require a fine mesh to provide a good topology, thus using expensive elements will increase the computation time unnecessary. Thus the elements recommended are the MATLAB Q4, PLANE42 and SOLID45 elements.

3.3 Preprogrammed Boundary Conditions

For the 2D code five different boundary conditions (BC's) are preprogrammed. In this section these boundary conditions are presented.

Since the codes are based on Sigmund [2001] and Andreassen et al. [2011] the method of creating the boundary conditions is similar to the one presented in these.

3.3.1 Bendsøe BCs

Inspired by Fig. 2.1 in [Bendsøe and Sigmund, 2003, p.73] two beam-like structures illustrated on Figure 3.3 are included in the MATLAB program. These two problems are named Bendsøe (a) and (b). Both beams have size $a = 5$ and $b = 1$. Since Bendsøe (a) has a solid frame of thickness t the design domain is narrowed to a smaller size, i.e. length $a - t$ and height $b - 2t$. The second Bendsøe problem has the full beam size as design domain.

Bendsøe (a) is a reinforcement problem where the outer frame structure is fixed as solid. Bendsøe (b) is basically the same problem but without the frame structure fixed as solids. Therefore the second problem has a trivial solution where an infinite eigenvalue is obtained by removing the entire structure. This situation is avoided when formulating the problem as a reinforcement problem or by adding non-structural masses to parts of the admissible design domain. It is chosen to include three discrete masses which will be used in the subsequent chapters.

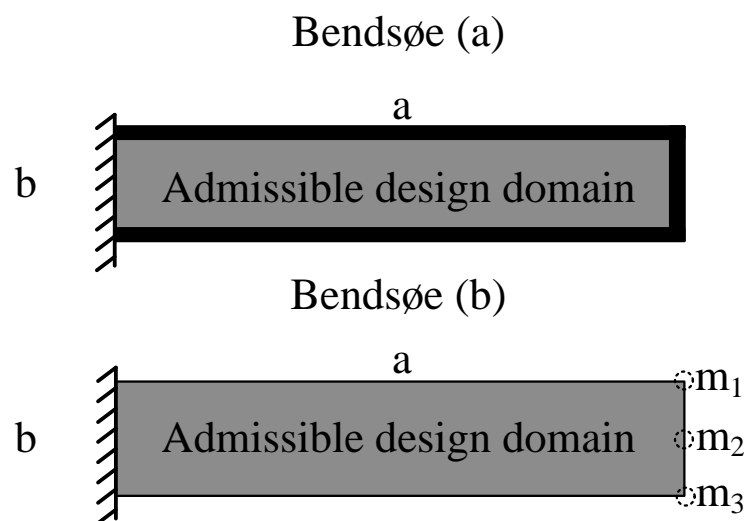


Figure 3.3: Bendsøe (a) and (b) boundary conditions.

3.3.2 Olhoff BCs

From Du and Olhoff [2007] three beam-like structures are included, these are illustrated on Figure 3.4. The admissible design domains ($a = 8, b = 1$) are either clamped or simply supported at the ends giving three different combinations for supports. These three problems are named Olhoff (a), (b) and (c). The material is isotropic with Young's modulus $E = 10^7$, mass density $\rho = 1$ and Poisson's ratio $\nu = 0.3$. These results are shown in $[rad/s]$ for easy comparison to the results from Du and Olhoff [2007].

The fundamental eigenfrequencies of the initial designs are all unimodal. With a density of 50% the fundamental eigenfrequencies are for the initial design $\omega_{1a} = 68.4$, $\omega_{1b} = 103.7$ and $\omega_{1c} = 145.7 [rad/s]$. As for many eigenfrequency problems the max-min optimum solution for these problems all contains bimodal modes. The optimum solutions found in Du and Olhoff [2007] are shown in Figure 3.5. To obtain the presented results 3200 trilinear 3D solid elements with incompatible modes are used in a plane stress configuration using a sensitivity filter.

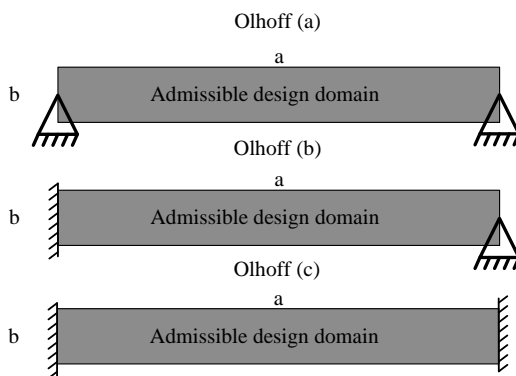


Figure 3.4: Olhoff (a),(b) and (c) boundary conditions [Du and Olhoff, 2007].

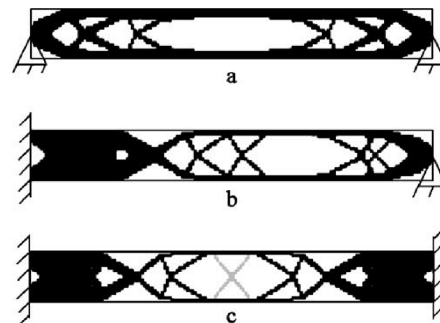


Figure 3.5: Optimum solution to Olhoff (a),(b) and (c) boundary conditions. Fundamental frequencies: (a) 174.7 $[rad/s]$, (b) 288.7 $[rad/s]$ and (c) 456.4 $[rad/s]$ [Du and Olhoff, 2007].

All of the boundary conditions will be used when validating the developed code in the following chapter.

CHAPTER 4

Code Development and Verification

Now that the code is described in general terms this chapter goes a step deeper in order to get a detailed description of the code, investigate the performance, and how different settings influences the performance of the code.

This chapter is build up by 13 sections, where one problem is analyzed at a time. Starting by verifying the sensitivities and ending with a simple 3D example targeting specified eigenvalues, switching two modes and obtaining a topology with a low measure of discreteness all in the same optimization.

4.1 Standard Configurations for the Analyzes

There are many options available in the program and in order to have a common basis for the analyzes a set of standard configurations are used. These are listed in Table 4.1. If nothing else is written the configurations from this table is used to produce the results.

Table 4.1: Standard configurations for the analyzes.

BC	Olhoff(c)
NE	3200
Resource Constraint	50 %
Solver	MATLAB
Element	MATLAB Q4
Algorithm	MaxMin MMA Bound Negative cost function
c constants	15 for ω_i 1.1 for Resource
N	4
Stiffness Interpolation	RAMP, q = 8
Mass Interpolation	Linear
Filter	Sensitivity r=1.5
Max Iterations	1000
Move limit	0.08
Convergence	$\max(\Delta x) \leq 0.02$
Mass Matrix	Lumped

These settings are not randomly generated and in the following a short reasoning for the

values in Table 4.1 are given. 3200 elements are used (20 elements in the height) as a compromise between detail in the topology and computational time. A similar discretization is used in Du and Olhoff [2007] for the same boundary conditions also with a resource constraint of 50%.

The eigs solver from MATLAB is used since the computational time in each iteration is less for the eigs solver when the number of elements is below 30,000, this is described in section 4.9.

Furthermore RAMP stiffness interpolation is used over SIMP interpolation since studies in section 4.7 shows that RAMP is a more robust stiffness interpolation scheme.

As optimization algorithm the MMA algorithm from Svanberg [2007] is used since optimality criteria can not be formulated as a bound formulation and it converges slightly faster than the Sequential Linear Programming algorithm. MMA has a nice property: it is possible to violate the constraints without operating in the infeasible region. This penalize the objective function and can be seen in appendix A. The MMA algorithm is used with the negative cost function since it showed slightly better performance for this problem compared to the inverse formulation also available in the program.

It is shown in section 4.3, that the number of eigenpairs included for this problem, must be at least three. Since multi-core processes are used a number of eigenpairs dividable with the number of cores are preferable because of the cheap computational cost of additional eigenpairs.

The sensitivity filter is used since it provides a better performance than the density filter. The filter help obtaining mesh independent topologies as shown in section 4.10. The radius of the filter is chosen to 1.5 times the element size. This radius is the smallest which provides satisfying results as seen in section 4.10.

Also a maximum number of iterations are necessary if for some reason the problem cannot converge. Choosing a too small number there is a risk of obtaining a bad or wrong topology due to it is not converged, whereas a too large value the calculation takes too much time without obtaining a useful topology. The maximum number of iterations allowed is closely related to the converges criteria, it is found that the values in Table 4.1 provide a good topology within a reasonable time span.

4.2 Verification of Sensitivities

In gradient based optimization naturally the sensitivities are of great importance. Here the aim is to verify that the sensitivities calculated during the analyzes are correct. The two

analytic sensitivities from equation (2.18) and (2.24) (in the following called Direct equation (2.18), since it is obtained by direct differentiation of equation (2.12), and Energy based (2.24)) are verified for distinct eigenvalues. In the case of repeated eigenvalues the sensitivity analysis is more complicated as described in section 2.2.1.

In order to verify the sensitivities the standard settings are used with the exception, that SIMP stiffness interpolation with a penalty of 3 and a consistent mass matrix are used. The sensitivities from Table 4.2 are produced during the first iteration, where all elements have a uniform density of 50%.

In order to verify the analytic sensitivities they are compared to a Forward finite difference, Backward finite difference and Central finite difference and a Semi-Analytic design sensitivity analysis (Semi-Analytical method). The Semi-Analytical method is similar to equation (2.17) but $\frac{\partial[K]}{\partial x_i}$ and $\frac{\partial[M]}{\partial x_i}$ are calculated using a forward difference scheme as in equation (2.19). The step size used is $\Delta x = 10^{-4}$.

The Central finite difference is used as a reference method, calculated sensitivities are compared with this to estimate the accuracy. The Semi-Analytical method is included to have an extra verification of the implemented sensitivity calculations.

From Table 4.2 it is seen, that all sensitivities are close, and the difference between these are at the fifth decimal. The largest difference is between the Forward finite difference and either the Backward finite difference or the Semi-Analytical method, with a relative difference below 1‰. The two analytic methods yields the same sensitivity, which is as expected since they are derived on basis of the same equation, thus the same result should be obtained. Meaning, that the implementation of the two are equivalent.

Table 4.2: The sensitivities using Forward finite difference (2.19), Backward finite difference, Central finite difference, Semi-Analytic design sensitivity analysis, Direct (2.18), Energy and PLANE42 sensitivities (2.24).

Element No.	Forward	Backward	Central	Semi-Analytic	Direct	Energy	PLANE42
181 (10,1)	0.18921	0.18929	0.18925	0.18929	0.18925	0.18925	0.18907
852 (80,20)	0.10423	0.10429	0.10426	0.10429	0.10426	0.10426	0.10414

Comparing the results from the analytical to the Central finite difference it is seen, that these results are the same. A small difference on the fifth decimal exist in the Forward and Backward finite difference, this is as expected, since the perturbation is 10^{-4} and a Forward/Backward finite difference has first order precision, thus the order of the error is proportional to the perturbation magnitude. Since the precision of the central difference method is second order the analytical sensitivities are expected being closer to the central different method this

verifies, that the implementation is correct.

The Direct and Energy based formulations are implemented in the MATLAB program. The OFD's and the Semi-Analytic is too slow to be used with many variables as used in Topology Optimization. Only the energy based formulation is implemented when using ANSYS, since the derivatives of the global stiffness and mass matrix is not directly available. It is seen, that a small difference of 1% exist between the sensitivities from ANSYS and MATLAB. This difference might be due to rounding errors or the element formulation is slightly different between the codes. In the following this is seen not to affect the final topology.

4.2.1 Asymmetric Designs

During the iterations when maximizing the fundamental eigenfrequency it is observed, that the design tends to become asymmetric. Since the boundary condition is doubly symmetric, and the first mode is a symmetric mode, thus the topology is expected to be doubly symmetric, thus a measure of the largest asymmetry, x_{unsym} , between the actual density and the doubly symmetric density is calculated. At a value of 0.001 for x_{unsym} it is seen, that the value rises significantly in the following iterations.

The first assumption was, that it was due to multiple eigenvalues are not accounted for. Since at first it was experienced when the two lowest eigenvalues had similar values.

However by changing the step size it is seen, that multiple eigenvalues do not initiate asymmetry, see Table 4.3. The setting to obtain the results is the default, with a convergence criteria of $\max(\Delta x) \leq 0.0005$. A tight convergence criteria is chosen to monitor whether or not the design returns to a symmetric design.

From Table 4.3 it is seen, that multiple eigenvalues are not present at the onset of asymmetric designs, however it is seen, that some peaks are present before the asymmetry is initiated, which might influence the asymmetry, see Figure 4.1 and 4.2. Furthermore it is observed, that the iteration where asymmetry occur is larger for RAMP than for SIMP. This might be because the gradient for RAMP does not vary as much as for SIMP.

Table 4.3: Number of iterations for asymmetry to occur.

Move Limit	Iteration where $x_{unsym} > 0.001$	ω_1	ω_2
1.00	45	418.8	559.8
0.75	47	420.7	455.8
0.60	50	422.1	485.3
0.50	35	395.7	420.0
0.40	34	397.4	495.9
0.30	16	353.3	372.0
0.20	27	376.0	414.7
0.10	22	371.1	400.6
0.05	31	369.7	410.4
0.01	122	377.3	408.4

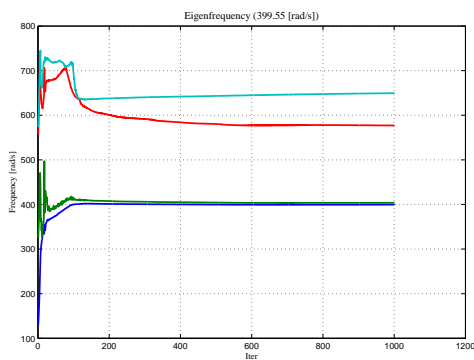


Figure 4.1: Obtained eigenfrequencies by the optimization of the Olhoff(c) problem.

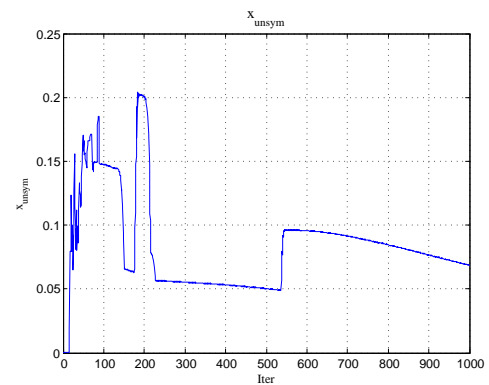


Figure 4.2: The corresponding maximum asymmetry of the densities.

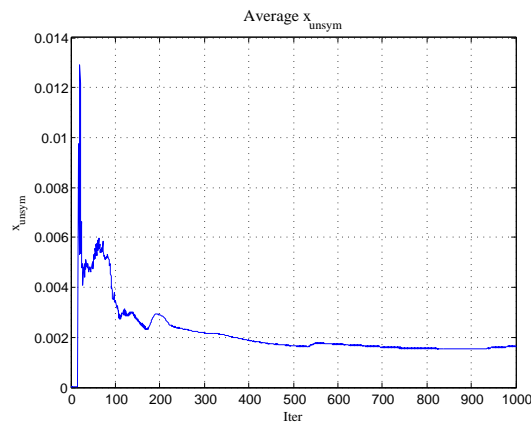


Figure 4.3: Average asymmetry of the elements.

The average asymmetry is seen in Figure 4.3. It is seen, that the average value of asymmetry is more than a factor ten lower than the absolute value, thus the asymmetry is likely to be related to a minority of the elements. This can also be assessed by visual inspection. Looking at an asymmetric design it is seen, that the visible asymmetry is present near the supports, see Figure 4.4 where the area of the most occurring asymmetry is marked. Here the beam is much stiffer than the middle of the beam, thus the effect of asymmetry is smaller than if it is present at the center of the beam.

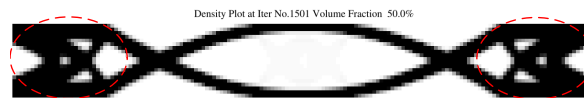


Figure 4.4: Areas where asymmetry often is present for the Olhoff (c) boundary conditions.

4.3 Number of Eigenpairs Included

The number of eigenvalues used in the optimization can have an effect on the result obtained. This is primarily when too few eigenvalues are taken into account, then the higher eigenfrequencies might interfere with the fundamental frequency. In Table 4.4 a study of the effect of the number of eigenvalues included in the optimization is seen.

Table 4.4: Results when varying the number of eigenpairs extracted.

N	1	2	3	4	6
Iter	4	135	125	124	130
MoD [%]	85.5	10.0	12.1	12.1	12.4
Volume Fraction [%]	50	58.5	50	50	50
ω_1 [rad/s]	187.17	401.08	407.77	407.57	408.74
ω_2 [rad/s]	None	406.55	412.82	412.26	414.51
ω_3 [rad/s]	None	None	607.89	619.45	616.12
ω_4 [rad/s]	None	None	None	644.98	642.75
ω_5 [rad/s]	None	None	None	None	734.37
ω_6 [rad/s]	None	None	None	None	773.49

From the table it is seen, that when more than two eigenpairs are calculated, then the topology obtained is independent of the number of eigenpairs extracted, see Figures 4.5 to 4.7. The reason is due to the third eigenfrequency does not approach the two lowest. This is seen on Figure 4.8.

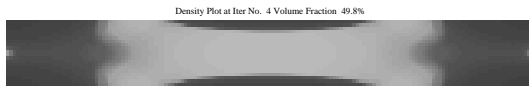


Figure 4.5: Topology obtained for Olhoff (c) boundary condition when including one eigenfrequency.

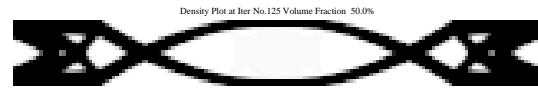


Figure 4.6: Topology obtained for Olhoff (c) boundary condition when including three eigenfrequencies.

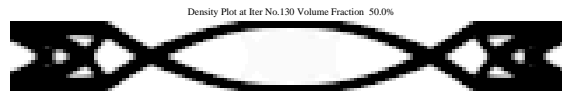


Figure 4.7: Topology obtained for Olhoff (c) boundary condition when including six eigenfrequencies.

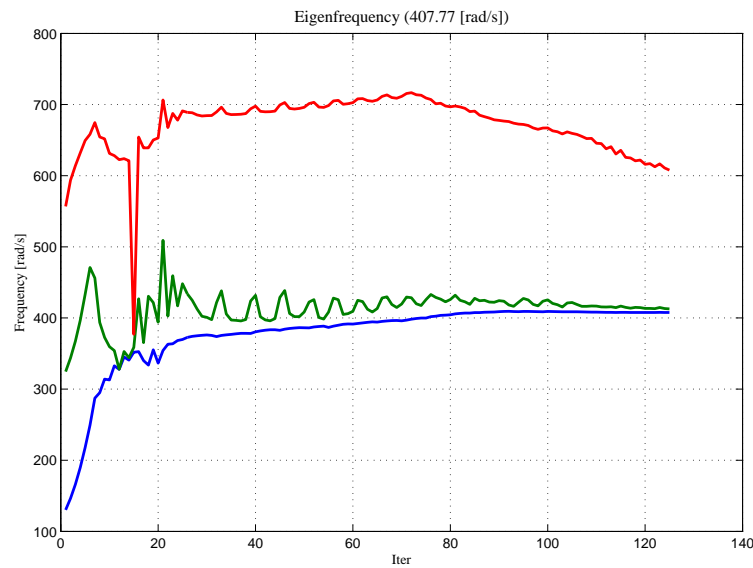


Figure 4.8: The eigenfrequency plot for Figure 4.6.

On basis of this it is recommended for maximizing the lowest eigenfrequency to choose the number of eigenpairs in accordance with the number of threads in the CPU, since the developed code is programmed to run on multi-core processors. Furthermore when tracking eigenmodes as in section 4.13 it is important to take enough eigenpairs into account in order to secure the modes to be tracked are included in the set. This is further described in section 4.13.

4.4 Effect of Penalization Constants

As described in appendix A only the c constants can be varied in the MMA algorithm. In this section the effect of changing these is examined. The standard settings, except for a consistent mass matrix, is used. In Svanberg [2007] it is seen, that the c constants from appendix A must fulfill $c_i a_i > a_0$ for all $a_i > 0$, and it is recommended, that the c constants are chosen reasonable large, which Krister Svanberg defines as $c_i = 1000$. From this different c constants are tested in order to see how these guidelines affect the result obtained.

In Table 4.5 it is seen, that $a_i c_i = a_0$ produces no change in the final results even though

Table 4.5: Results when modifying the c constants in MMA.

Test case	1	2	3	4	5	6	7	8
c_1	1 [†]	15	1.1	1000	0.1 [†]	1.1	10^5	$10^{10\ddagger}$
c_2	1 [†]	15	1.1	1000	0.1 [†]	0.1 [†]	10^5	$10^{10\ddagger}$
c_3	1 [†]	15	1.1	1000	0.1 [†]	0.1 [†]	10^5	$10^{10\ddagger}$
c_4	1 [†]	15	1.1	1000	0.1 [†]	0.1 [†]	10^5	$10^{10\ddagger}$
c_5	1	1.1	15	1000	0.1	1	1	1
Iter	136	140	142	138	318	151	136	149
ω_1 [rad/s]	401.97	401.88	401.95	401.88	295.13	400.86	401.91	402.09
ω_2 [rad/s]	409.96	409.60	408.96	409.22	783.59	407.95	409.45	409.03
ω_3 [rad/s]	619.08	617.44	614.14	616.63	1054.60	622.53	618.93	611.88
ω_4 [rad/s]	637.29	636.92	636.42	635.79	1065.47	634.05	636.52	637.30
MoD	11.7	11.8	11.7	11.7	12.1	11.9	11.7	11.7

[†] Violates the assumptions in Svanberg [2007]

[‡] Violates the recommendations in Svanberg [2007]

$a_i c_i > a_0$, see Figure 4.9 and 4.10, whereas a serious violation produces completely different results, see Figure 4.11. Test case number 7 produced no difference from the ones which fulfilled all the guidelines, whereas the huge number in 8 made it difficult for a subproblem to converge, however the same topology is obtained as for the other, as seen in Figure 4.12.

An interesting result is seen in test case number 6, where only the first and volume fraction c constant fulfill the assumptions, but the performance is seen being similar to the test cases where all guidelines are fulfilled. This might be because no multiple eigenvalues are present in the final topology.



Figure 4.9: The topology from test number 1 in Table 4.5.

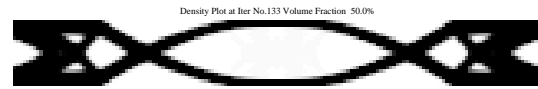


Figure 4.10: The topology from test number 2 in Table 4.5.



Figure 4.11: The topology from test number 5 in Table 4.5.

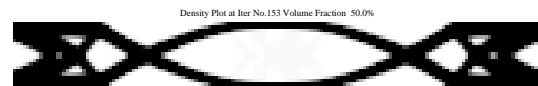


Figure 4.12: The topology from test number 8 in Table 4.5.

From this it can be concluded, that following the guidelines in [Svanberg, 2007] produces consistent results.

4.5 Move Limit

The MMA solver has a build in (default) move limit strategy, which is based on changing the asymptotes in accordance to the change in densities where fluctuations narrows the trust region, see Svanberg [2007].

It is observed, that the build in move limits has a tendency of causing fluctuations in the frequencies, see Figure 4.13, thus additional move limit strategies are considered, where a maximum move limit is added on top of the default move limit strategy.

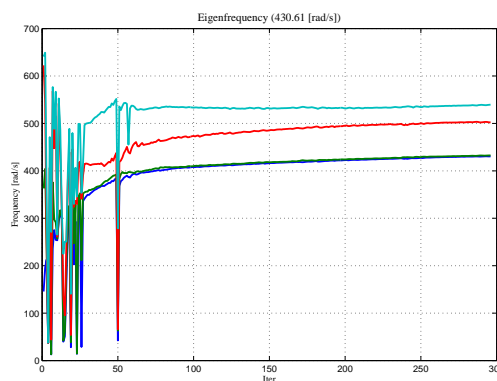


Figure 4.13: Fluctuations of eigenfrequencies due to use of a large move limit.

The alternative strategies are based on the default increase and reduction in trust region is

kept, but the maximum change in one element is limited to e.g. 10% of the total change possible. Throughout this section the effects of choosing different move limit strategies are tested, using the standard settings with the exception of SIMP stiffness interpolation with $p = 3$ and Olhoff-Du(c) mass interpolation.

Maximum move limit strategies are tested in a range from 0.025 to 1.0, where 1.0 represent the default move limit strategy. On Figure 4.14 the fundamental frequency is shown as a function of the move limit. Using a move limit above 0.2 it is seen, that the fundamental frequencies are within 15 [rad/s].

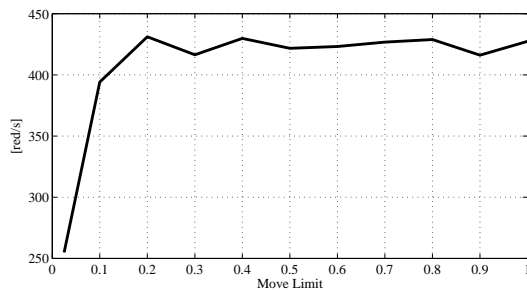


Figure 4.14: The lowest eigenfrequency obtained by the optimization of the Olhoff(c) problem various different move limits.

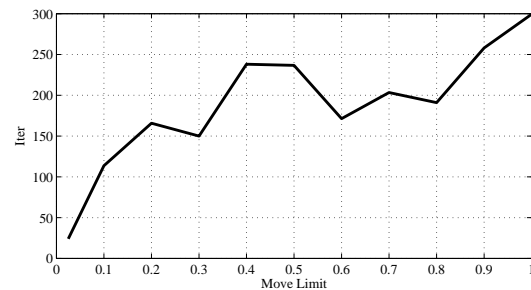


Figure 4.15: The number of iterations before convergence of the Olhoff(c) problem various different move limits.

The number of iterations is shown on Figure 4.15. It is seen, that with small move limits fewer iterations are needed to converge than for large move limits. This might be due to fewer reductions of the asymptotes are needed before convergence.

Varying the move limit strategy it is seen, that different topologies are obtained as seen in Figure 4.16 to 4.21. Comparing this to the fundamental frequencies in Figure 4.14 an other proof of the non-convexity for the problem is seen, thus the final topology is (as expected) dependent on the move limit strategy.

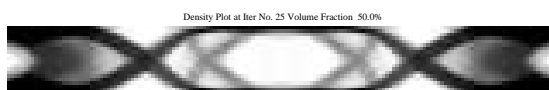


Figure 4.16: The topology with a move limit of 0.025, $\omega_1=254.8$ [rad/s].

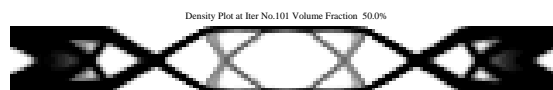


Figure 4.17: The topology with a move limit of 0.080, $\omega_1=403.4$ [rad/s].



Figure 4.18: The topology with a move limit of 0.2, $\omega_1=431.1$ [rad/s].



Figure 4.19: The topology with a move limit of 0.5, $\omega_1=421.7$ [rad/s].

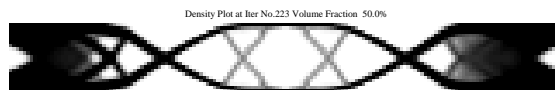


Figure 4.20: The topology with a move limit of 0.7, $\omega_1=426.8$ [rad/s].

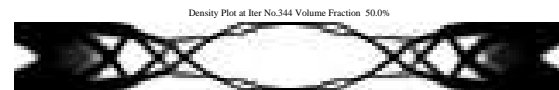


Figure 4.21: The topology with a move limit of 1.0, $\omega_1=427.4$ [rad/s].

Comparing the results from the SIMP stiffness interpolation scheme to a RAMP interpolation in Figure 4.22 to 4.25 it is seen, that using a RAMP interpolation results in similar topologies and fundamental frequencies regardless of the move limit strategy chosen.



Figure 4.22: The topology with a move limit of 0.08, $\omega_1=407.04$ [rad/s] using RAMP stiffness interpolation.

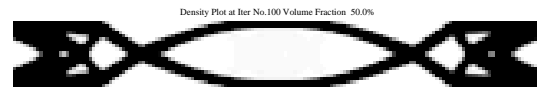


Figure 4.23: The topology with a move limit of 0.5, $\omega_1=404.15$ [rad/s] using RAMP stiffness interpolation.

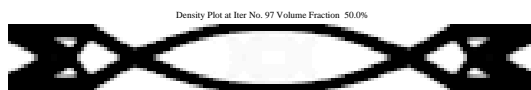


Figure 4.24: The topology with a move limit of 0.7, $\omega_1=401.46$ [rad/s] using RAMP stiffness interpolation.

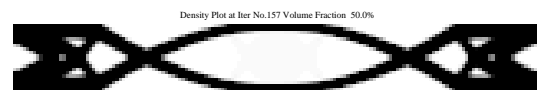


Figure 4.25: The topology with a move limit of 1.0, $\omega_1=408.07$ [rad/s] using RAMP stiffness interpolation.

4.6 Start Guess and Resource Constraint

Since the problem is inherently nonconvex, the start guess has an influence on the final topology obtained. In this section the influence of these is investigated. The standard settings are used in the analyzes.

The most common start guess is to use the material constraint and evenly distribute this over the design domain. In Figure 4.26 and 4.27 results obtained by evenly distributed material with different initial volume fractions between 30 and 110% (meaning the maximum density is violated in all elements) still keeping the resource constraint of 50%.

It was observed, that a start guess below 30% yields the trivial solution. SIMP is even more sensitive and results in the trivial solution if the start guess is $\leq 40\%$. As long as the start guess is larger than 40% the solution obtained converges towards the same optimum. The MMA algorithm has no trouble with the volume constraint being violated and it does not have any significant effect on the number of iterations needed in order to converge. The measure of discreteness is also unaffected as illustrated on Figure 4.27.

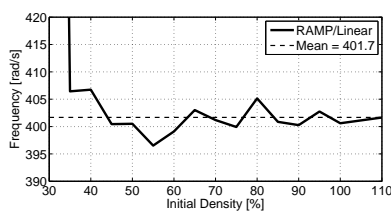


Figure 4.26: The fundamental frequency as a function of the evenly distributed start guess with different initial volume fractions. An initial volume fraction $\leq 30\%$ for RAMP stiffness interpolation and $\leq 40\%$ for SIMP stiffness interpolation result in the trivial solution.

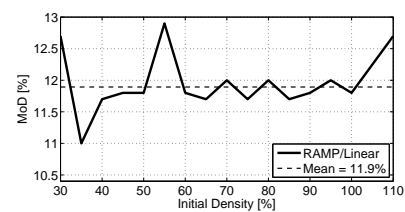


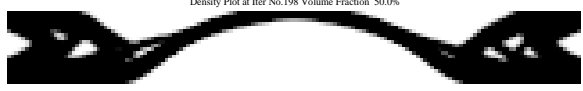
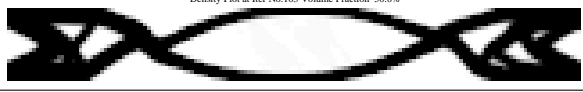
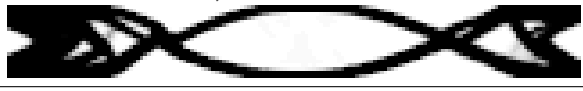


Figure 4.27: The Measure of Discreteness as a function of the evenly distributed start guess with different initial volume fractions.

Since the cost function is strictly non-convex an unevenly distributed start guess results in different topologies due to the non-convexity. This is illustrated in Table 4.6 where all elements are assigned a random value between zero and one as the start guess. It is clear that the start guess has influence on the topologies obtained, since neither of the obtained topologies correspond to the one obtained with an evenly distributed density.

In appendix B a table with different start guesses is shown, this is done to see how guesses which might reassemble the final topology will affect the optimization. One example could be, that it is desired to track a “breathing” mode, thus guessing on an uniformly distributed density then that particular mode might not be present among the lowest eigenpairs. As also seen from the random start guesses that the final topology is affected.

As shown previously and in appendix B the start guess has a rather big influence on the

Table 4.6: Final topologies for five random start guesses.

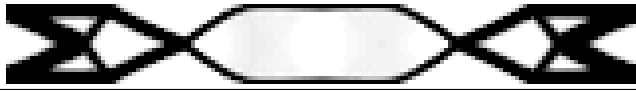

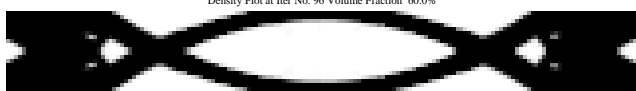

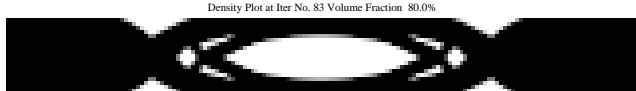
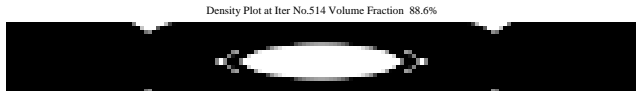
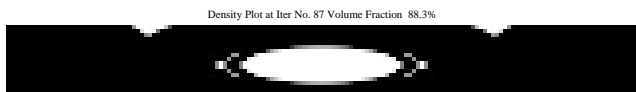
ω_1 [rad/s] Iter MoD[%]	Topology
438.73 198 7.9	Density Plot at Iter No.198 Volume Fraction 50.0% 
393.65 163 12.7	Density Plot at Iter No.163 Volume Fraction 50.0% 
391.78 115 13.3	Density Plot at Iter No.115 Volume Fraction 50.0% 
440.33 260 8.0	Density Plot at Iter No.260 Volume Fraction 50.0% 
396.17 189 13.1	Density Plot at Iter No.189 Volume Fraction 50.0% 

topology obtained, which of course also applies for the material available, if the solution is bounded by the volume fraction constraint.

The results in Table 4.7 are obtained using the program configurations from Table 4.1, but with different resource constraints, where the start guess are evenly distributed. This shows some alarming results. For a low resource constraint as 40% it is obvious that the optimum found might be lower since the objective function can be constrained from the high frequencies. For resource constraints greater than 50% the algorithm should find optimums better or as good as the one found for 50%. In Table 4.7 it is seen, that this is not the case. The same problem occur even though a larger move limit is chosen.

For higher volume fractions, it is seen, that the outer shapes reassemble the shape on Fig. 29 in Seyranian et al. [1994], which is a discretisation of the classical clamped-clamped continuum column determined by Niels Olhoff and Steen Rasmussen in 1977.

Table 4.7: Dependency of resource constraint on the final topology.

Resource constraint / Final volume fraction	ω_1 [rad/s] Iter MoD [%]	Topology
40%/40%	390.46 80 16.1	Density Plot at Iter No. 80 Volume Fraction 40.0% 
50%/50%	401.84 134 11.7	Density Plot at Iter No. 134 Volume Fraction 50.0% 
60%/60%	395.58 96 9.3	Density Plot at Iter No. 96 Volume Fraction 60.0% 
70%/70%	372.17 75 7.2	Density Plot at Iter No. 75 Volume Fraction 70.0% 
80%/80%	345.08 83 6.3	Density Plot at Iter No. 83 Volume Fraction 80.0% 
90%/88.6%	324.93 514 3.8	Density Plot at Iter No. 514 Volume Fraction 88.6% 
100%/88.3%	325.05 87 4.4	Density Plot at Iter No. 87 Volume Fraction 88.3% 

On Figure 4.28 the frequency plot for the 90% topology in Table 4.7 is shown. From the figure it is seen, that the lower frequency is not due to bimodality, however by disabling the filter the topology is seen on Figure 4.30, and the frequencies on Figure 4.29 From this it is seen, that the filter prevents the removal of material, thus obtaining a lower fundamental frequency.

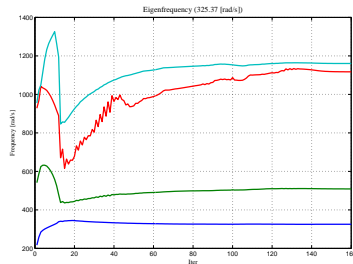


Figure 4.28: The first four frequencies for the Olhoff (c) beam with 90% volume fraction in Table 4.7.

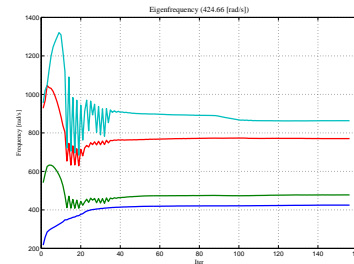


Figure 4.29: The same set up as in Figure 4.28 with disabled filter.

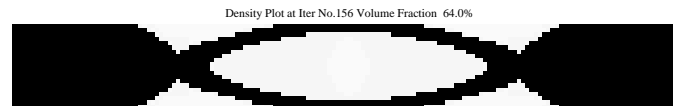


Figure 4.30: The topology obtained for the Olhoff (c) beam with 90% volume fraction in Table 4.7 and disabled filter.

4.7 Mass and Stiffness Interpolation

In this section the phenomenon of local modes are discussed and some solutions to the problem are presented and investigated. The classical solutions to the problem are described in section 2.1 and are further investigated in this section.

First to illustrate the problem on Figure 4.31 the lowest eigenfrequency is maximized using the program configuration from Table 4.1 with linear mass interpolation and a minimum density of 10^{-12} . As explained in section 2.1 the local modes occur when the stiffness approaches zero faster than the mass, i.e. the frequency of a single element becomes lower than the mode which are maximized. Furthermore as will be shown in the following, an other aspect must also be considered when having a patch of elements with low stiffness.

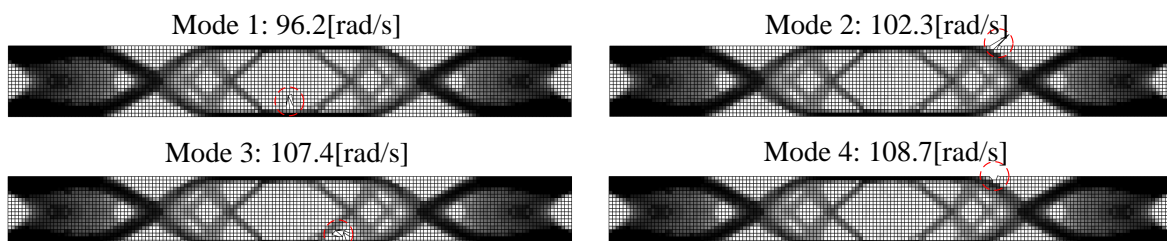


Figure 4.31: SIMP stiffness interpolation with penalty $p=3$ and linear mass interpolation. Local modes are shown in the red circle.

4.7.1 SIMP with Olhoff-Du Mass Interpolations

As described in section 2.1 one of the methods which can be used to avoid the local modes is to use SIMP with either one of the Olhoff-Du mass interpolations. Varying the penalty of SIMP different results are obtained. On Figure 4.32 the fundamental frequency obtained in the optimization is illustrated as a function of the stiffness penalty p .

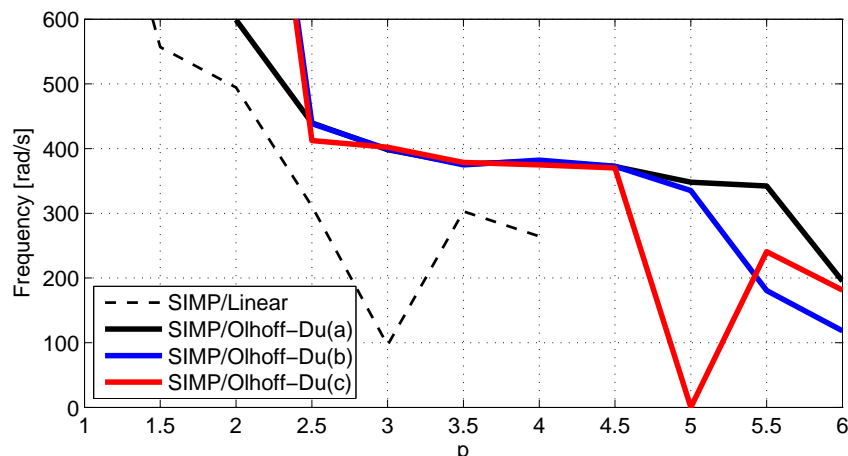


Figure 4.32: Fundamental frequency when varying the penalty for SIMP stiffness interpolation for the implemented mass interpolations.

When a linear mass interpolation is used the local modes are present when the penalty is ≥ 2.5 as illustrated on Figure 4.31. On the other hand if $p \leq 2.5$ the solution goes toward the trivial solution with zero material. On Figure 4.32 this is seen as very high frequencies, and when the penalty is too high the local modes result in low frequencies. With the Olhoff-Du mass interpolations the local modes are occasionally still present in some iterations when the penalty is ≥ 3 . This is due to the "peak" in the Ω function, see Figure 2.4 on page 10, combined with a patch of low stiffness elements. The algorithm is able to pass this point when using the Olhoff-Du mass interpolations. If the linear mass interpolation is used " Ω " approaches infinity as the density goes to zero.

Topologies when varying the SIMP penalty are shown in Figures 4.33 to 4.36. On the figures different mass interpolations are shown. As stated in Du and Olhoff [2007] the difference between the three Olhoff-Du mass interpolations are negligible, and the figures display the effect of altering the penalty of the stiffness interpolation. In Figure 4.33 a linear mass interpolation is used. Here it is seen, that the topology in Figure 4.35 can be identified, but due to local modes it is not possible to obtain that topology. This shows, that the Olhoff-Du mass interpolations eliminate local modes in the final topology. Figure 4.34 shows, that a too high penalty results in local modes due to the mass and stiffness penalty are too close.

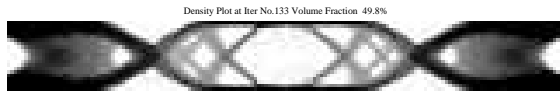


Figure 4.33: The final topology for SIMP stiffness interpolation with penalty $p=3$ and linear mass interpolation. The lowest frequency is a local mode at $96.2[\text{rad/s}]$. The mode shape is shown on Figure 4.31.

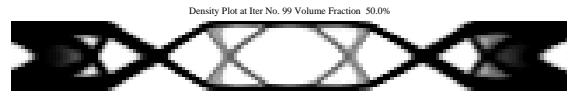


Figure 4.35: SIMP stiffness interpolation with penalty $p=3$ and Olhoff-Du(c) mass interpolation. The lowest frequency is $402[\text{rad/s}]$ without any local modes.

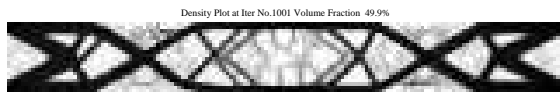


Figure 4.34: The topology at iteration 1001 using SIMP stiffness interpolation with penalty $p=5.5$ and Olhoff-Du(b) mass interpolation. The lowest frequency is a local mode at $180.5[\text{rad/s}]$. The optimization has not converged as a result partly of the local modes and partly of the close to singular matrices.



Figure 4.36: The final topology using SIMP stiffness interpolation with penalty $p=4$ and Olhoff-Du(a) mass interpolation. The lowest frequency is $380.3[\text{rad/s}]$. The solution is obtained in 60 iterations.

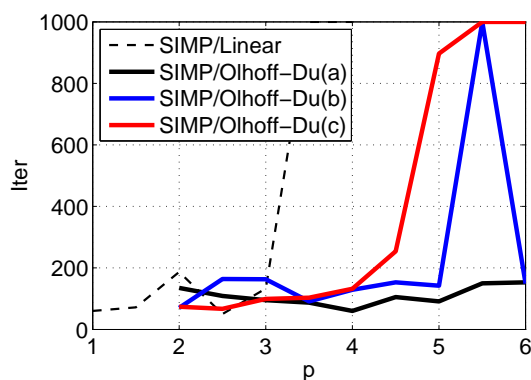


Figure 4.37: The number of iterations necessary for convergence as a function of the stiffness penalty.

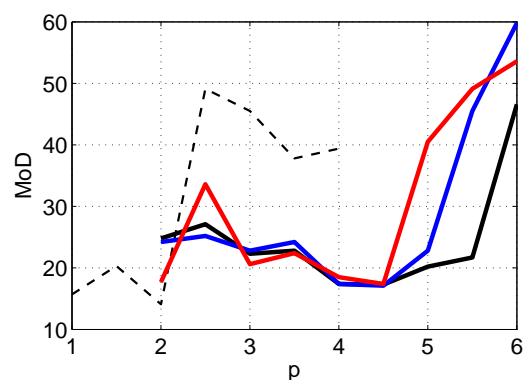


Figure 4.38: The Measure of Discreteness as a function of the stiffness penalty.

On Figure 4.37 and 4.38 the number of iterations and the measure of discreteness are illustrated as a function of the stiffness penalty p . It is seen that a penalty > 4.5 in general results in slow convergence and a poor measure of discreteness. However as mentioned in Du and Olhoff [2007] the difference between the three Olhoff-Du mass interpolations are small and all three solves the problem associated with local modes compared to the linear mass interpolation. In this specific problem a slightly lower number of iterations are needed for

the Olhoff-Du(a) mass interpolation when $p > 3.5$ as seen on Figure 4.37 the lowest eigenfrequency and the measure of discreteness is nearly the same for all three mass interpolations. A penalty between 2.5 and 4 results in roughly the same topology and frequency when using SIMP with either one of the Olhoff-Du mass interpolations.

4.7.2 RAMP Stiffness Interpolation

Another solution to the problems associated with the local modes is to use the RAMP stiffness interpolation. The RAMP stiffness interpolation can also be used together with the Olhoff-Du mass interpolations as shown on Figure 4.39. The RAMP stiffness interpola-

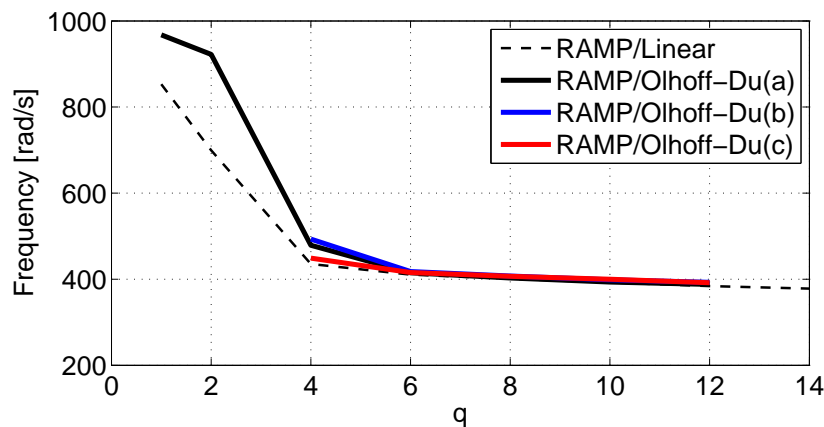


Figure 4.39: The results of the lowest eigenfrequency optimized using RAMP stiffness interpolation and the implemented mass interpolations.

tion with linear mass interpolation has been tested using penalties as high as 10^6 without any problem associated with localized modes with a minimum density of 10^{-3} . However using such a large stiffness it is seen, that a measure of the condition number ($q = 16$) is $\frac{\max(K(i,i))}{\min(K(i,i))} = 3.83 \cdot 10^4$, $i = 1, \dots, NE$ meaning, that for penalties ≥ 16 the penalty is too large. This also results in low frequencies because the intermediate stiffnesses are close to zero. Despite the large penalty the topology obtained is much alike (nearly the same) as illustrated in Figure 4.41 and 4.43.

However if the penalty is too small the topology will converge toward the trivial solution as with the SIMP this is illustrated on Figure 4.40. With a penalty of 3 to 5 the topology obtained is similar to the SIMP solutions as illustrated on Figure 4.42 but the void areas contributes with too much stiffness resulting in too high frequencies. The RAMP stiffness interpolation is much more robust than the SIMP stiffness interpolation since no problems with local modes or singular stiffness matrices which occur if the penalty is too large.

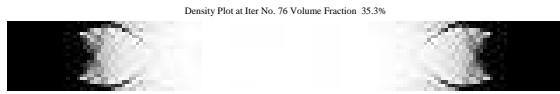


Figure 4.40: The topology with RAMP interpolation with penalty $q=2$ and linear mass interpolation. As for SIMP interpolation if the penalty is too low the topology found during the optimization converges toward the trivial solution.

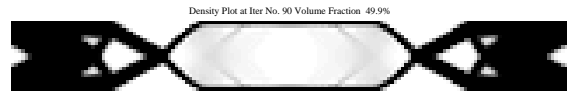


Figure 4.42: RAMP stiffness interpolation with a penalty of $q=4$ and linear mass interpolation result in a topology close to the one obtained using SIMP stiffness interpolation and a Olhoff-Du mass interpolation.

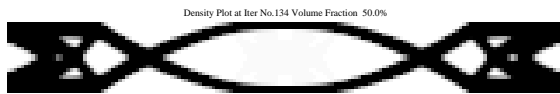


Figure 4.41: RAMP interpolation with a penalty of $q = 8$ results in a different topology than SIMP stiffness interpolation. Here $\omega_1 = 400.8[\text{rad/s}]$. Nearly the same topology is found with $6 \leq q \leq 50$.



Figure 4.43: The topology using RAMP stiffness interpolation with a penalty of $q=100$. The lowest frequency is $252.4[\text{rad/s}]$. The same topology is obtained for $100 \leq q \leq 10^6$. These penalties are too high to be used.

RAMP Stiffness Interpolation with Olhoff-Du Mass Interpolation

The topologies obtained using RAMP stiffness interpolation with any of the Olhoff-Du mass interpolations schemes are similar to the ones obtained using RAMP with linear mass interpolation. On Figure 4.45 it is shown that the measure of discreteness is nearly the same regardless of the interpolation chosen, and on Figure 4.44 it is seen that the number of iterations needed for convergence is reduced when using RAMP together with the Olhoff-Du mass interpolations for larger values of q . It is difficult to conclude anything general from this, but using RAMP together with Olhoff-Du(b) and (c) seems to reduce the number of iterations needed for convergence. During testing it is experienced, that RAMP stiffness interpolation with either Olhoff-Du(c) or a fully linear mass interpolation provides a better convergence than the other mass interpolations.

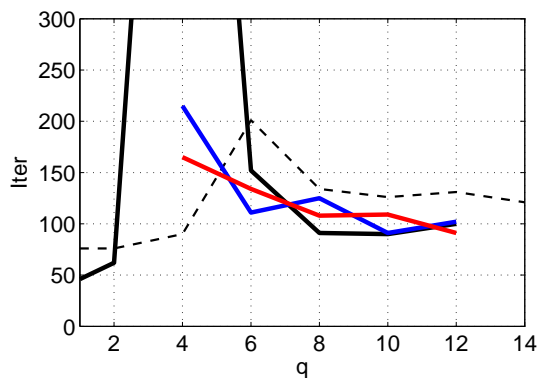


Figure 4.44: The number of iterations necessary for convergence as a function of the RAMP stiffness penalty. With a penalty of 4 convergence was never achieved within 1000 iterations using Olhoff-Du(a) mass interpolation.

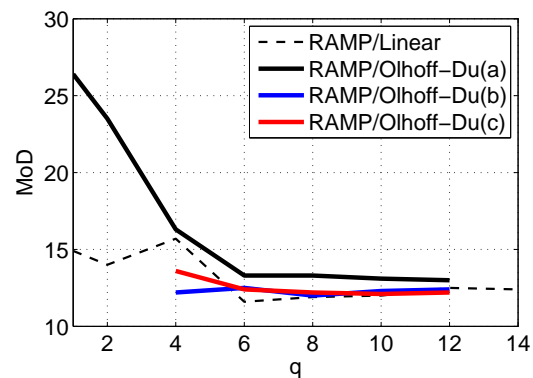


Figure 4.45: The Measure of Discreteness as a function of the RAMP stiffness penalty.

RAMP and Local Modes

Through the development of the code it is seen, three conditions has to be fulfilled in order for local modes to occur:

1. A high inverse Rayleigh's coefficient (Ω)
2. Low element stiffness
3. The above two conditions must be fulfilled for all elements surrounding a single node

On Figure 2.1 it is seen that RAMP stiffness interpolation has a lower stiffness for densities ≥ 0.5 and the opposite is the case for densities < 0.5 . When using Olhoff-Du mass interpolations with either SIMP or RAMP stiffness interpolation Ω goes toward 0 for low densities. When using SIMP stiffness interpolation the minimum density allowed in the optimization is 10^{-3} due to the stiffness matrix becomes badly scaled, in many cases this lower bound works fine with RAMP stiffness interpolation, but because RAMP has a higher stiffness than SIMP for the low density regions, RAMP becomes stiffer in the void elements. When the resource constraint is low this becomes a problem as seen in chapter 5.

Since a low stiffness is desired in low density areas the density for RAMP interpolation ($q = 8$) providing the same stiffness as SIMP ($p = 3$) for $x = 10^{-3}$ is 10^{-8} . If a minimum element density of 10^{-8} is used RAMP stiffness interpolation will produce local modes, unless one of the Olhoff-Du mass interpolations are used. But if a Olhoff-Du mass interpolation is used with RAMP stiffness interpolation the local modes are completely avoided since RAMP does not have a peak in the inverse Rayleigh coefficient as illustrated on Figure 4.46. If the

lower bound for the design variables are lower than 10^{-3} for SIMP and 10^{-8} for RAMP the system of equations becomes close to singular.

MMA tend to stay close to the upper and lower bound of the design variables while never

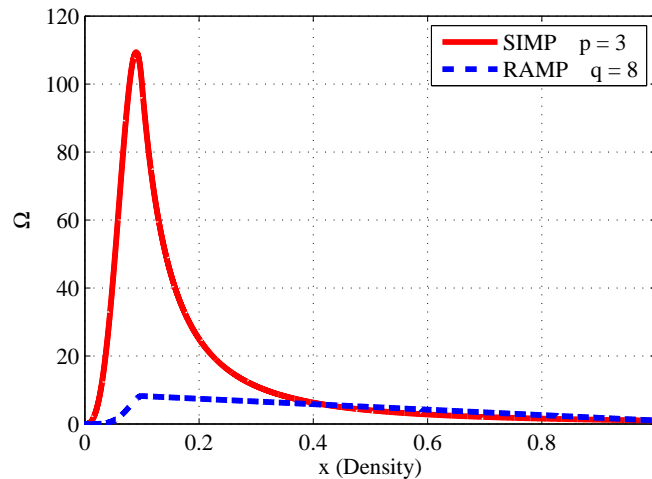


Figure 4.46: The inverse Rayleigh’s coefficient for SIMP and RAMP stiffness interpolations both using Olhoff-Du(c) mass interpolation. Note that RAMP stiffness interpolation avoids the same peak in Ω as SIMP stiffness interpolation.

reaching exact x_{min} and 1. The effect of forcing low density elements to x_{min} , and high density elements to 1 has been investigated, and the difference is negligible, for the tested examples. If values larger than 10^{-4} are forced to the lower bound of 10^{-8} it will interfere with the solution and more iterations are needed before convergence.

4.7.3 Comparison of RAMP and SIMP Stiffness Interpolation

Comparing the two stiffness interpolations with each other it is seen, that the penalty q has to be a larger value than the penalty p used in SIMP. The frequencies obtained with SIMP and RAMP stiffness interpolations are similar although for fair comparison of the results, the intermediate values have to be penalized consistently, e.g. p = 3 and q = 8. SIMP converges using slightly fewer iterations as illustrated on Figure 4.47, whereas RAMP obtains topologies with a lower measure of discreteness as illustrated on Figure 4.48.

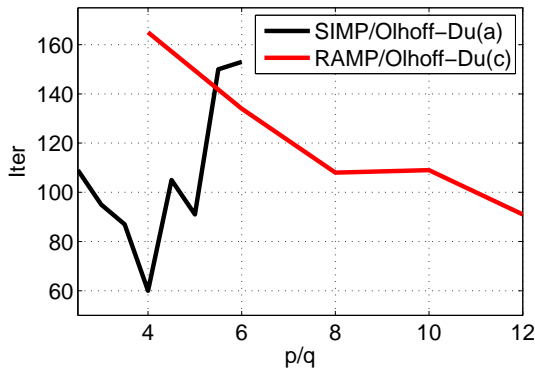


Figure 4.47: The number of iterations necessary for convergence as a function of the RAMP and SIMP stiffness penalties.

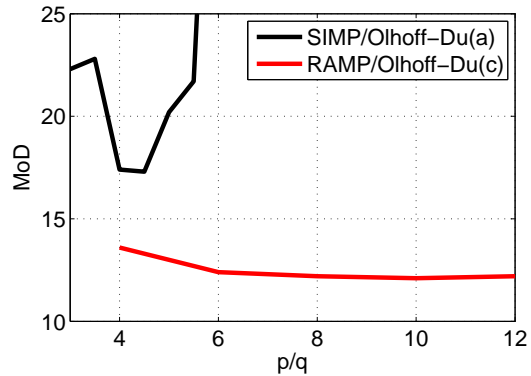


Figure 4.48: The Measure of Discreteness as a function of the RAMP and SIMP stiffness penalties.

Since discrete designs are desired the results obtained using a RAMP stiffness interpolation are considered being better than the results using a SIMP stiffness interpolation even though the frequencies are lower, as seen on Figure 4.49.

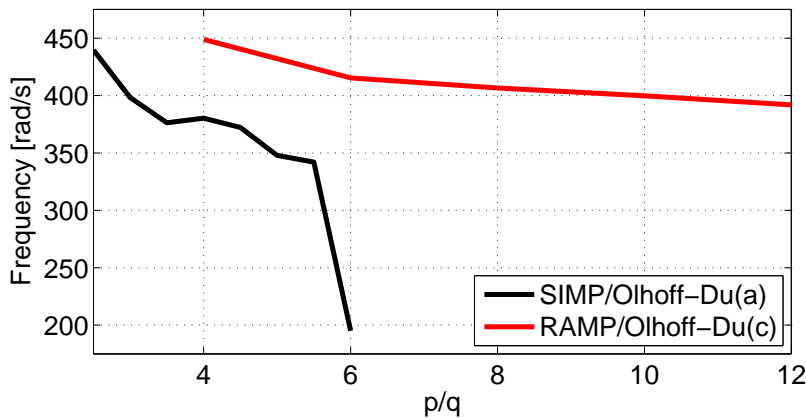


Figure 4.49: Comparison of frequencies using SIMP and RAMP stiffness interpolation.

In Figures 4.50 to 4.57 a comparison between SIMP ($p = 3$) with Olhoff-Du(a) and RAMP ($q = 8$) with Olhoff-Du(c) mass interpolation are tested with the preprogrammed BC's in Figure 3.3 and 3.4 except for Bendsøe(b) since it will give the trivial solution.

For all boundary conditions the lowest frequency is within 5% and the number of iterations are nearly the same for the two test set up. However the set up using RAMP stiffness interpolation consistently obtains results with a lower measure of discreteness. For some unknown reason RAMP with the Olhoff-Du(c) mass interpolation has difficulties with the

Bendsøe(a) problem. The solution obtained using RAMP stiffness interpolation can be improved by using a linear mass interpolation.

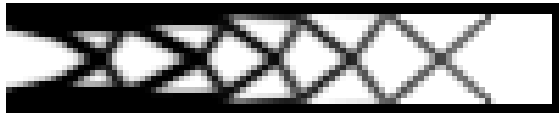


Figure 4.50: Bendsøe (a) boundary condition for SIMP stiffness interpolation with a penalty of $p=3$ and Olhoff-Du(a) mass interpolation. Convergence achieved in 74 iterations, $\omega_1=43.64$ [Hz], MoD=22.9 %.



Figure 4.54: Bendsøe (a) boundary condition for RAMP stiffness interpolation with a penalty of $q=8$ and Olhoff-Du(c) mass interpolation. Convergence achieved in 143 iterations, $\omega_1=41.92$ [Hz], MoD=21.3 %.

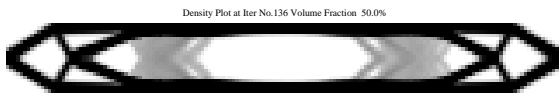


Figure 4.51: Olhoff (a) boundary condition for SIMP stiffness interpolation with a penalty of $p=3$ and Olhoff-Du(a) mass interpolation. Convergence achieved in 136 iterations, $\omega_1=159.82$ [rad/s], MoD=27.2 %.

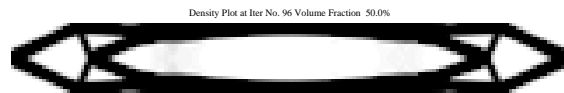


Figure 4.55: Olhoff (a) boundary condition for RAMP stiffness interpolation with a penalty of $q=8$ and Olhoff-Du(c) mass interpolation. Convergence achieved in 96 iterations, $\omega_1=159.81$ [rad/s], MoD=14.7 %.



Figure 4.52: Olhoff (b) boundary condition for SIMP stiffness interpolation with a penalty of $p=3$ and Olhoff-Du(a) mass interpolation. Convergence achieved in 177 iterations, $\omega_1=251.65$ [rad/s], MoD=26.6 %.

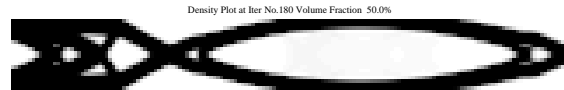


Figure 4.56: Olhoff (b) boundary condition for RAMP stiffness interpolation with a penalty of $q=8$ and Olhoff-Du(c) mass interpolation. Convergence achieved in 180 iterations, $\omega_1=261.39$ [rad/s], MoD=13.8 %.

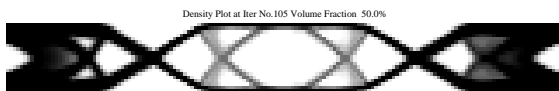


Figure 4.53: Olhoff (c) boundary condition for SIMP stiffness interpolation with a penalty of $p=3$ and Olhoff-Du(a) mass interpolation. Convergence achieved in 105 iterations, $\omega_1=397.29$ [rad/s], MoD=22.9 %.

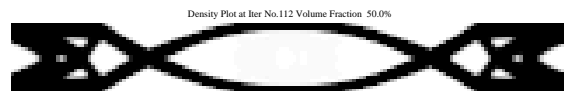


Figure 4.57: Olhoff (a) boundary condition for RAMP stiffness interpolation with a penalty of $q=8$ and Olhoff-Du(c) mass interpolation. Convergence achieved in 112 iterations, $\omega_1=406.47$ [rad/s], MoD=12.1 %.

It is impossible to declare one single interpolation method being better at all times. There are some differences, and what is best for one case is not necessarily the best for all cases. At least the optimum compromise between the mass and stiffness interpolation is not found

here, but RAMP with $q = 8$ and Olhoff-Du(c) mass interpolation is in general resulting in well defined topologies with a low measure of discreteness, although the void areas contributes with more stiffness than the SIMP interpolation. In cases where the void areas contributes with too much stiffness the lower limit for the densities can be reduced to 10^{-8} and if necessary it is possible to force MMA to use the upper and lower bound for the design variables.

In Du and Olhoff [2007] a penalty of about 3 is used for the SIMP stiffness interpolation. This can be due to the local modes which occur when a patch of elements reach a density close to 0.1. The problem is increasing as number of elements is increased, but it can be reduced by tuning the move limit and the penalty for the problem. E.g. for the Olhoff(c) problem with 3200 elements, a SIMP penalty of 3.2 and a move limit of 0.08 result in a good solution without encountering local modes. It is difficult to know what penalty exactly is used in Du and Olhoff [2007], and the move limit also have an influence if the peak in the inverse Rayleigh must be avoided, and if the number of elements are increased the problem enhances. In Figure 4.58 and 4.59 the Olhoff(c) problem with 12800 elements are solved with SIMP and RAMP stiffness interpolation. From this it is seen that RAMP stiffens interpolation avoids local modes as opposed to the SIMP stiffness interpolation, which is the main reason for using the RAMP interpolation rather than SIMP.

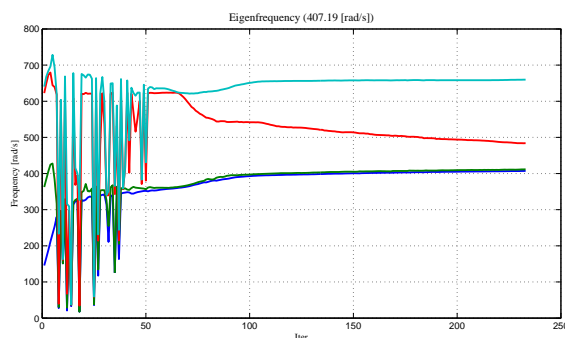


Figure 4.58: The frequency plot from solving Olhoff(c) boundary conditions using 12800 elements, SIMP stiffness interpolation with a penalty of 3, sensitivity filter with $r=3$ and Olhoff-Du(c) mass interpolation.

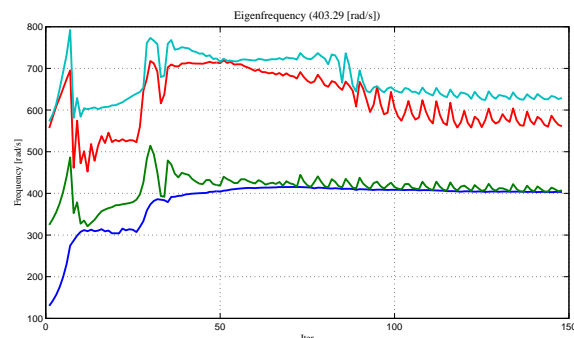


Figure 4.59: The frequency plot from solving Olhoff(c) boundary conditions 12800 elements, RAMP stiffness interpolation with a penalty of 8, sensitivity filter with $r=3$, Olhoff-Du(c) mass interpolation and a lower bound of 10^{-8} .

4.8 Mass Matrix

Two different mass matrix representations are available in the code, a consistent and a lumped. In the MATLAB code both matrices are implemented on basis of Cook et al. [2002]. In ANSYS the implemented matrices are used.

The consistent element mass matrix $[m_e]$ is symmetric, full and positive definite, and the global mass matrix $[M]$ has the same sparse topology as the global stiffness matrix $[K]$. The lumped mass matrix has only entries on the diagonal.

To show the difference between the two matrices a mesh refinement study is conducted. The boundary condition is Olhoff(c), and all elements have a density of 50%.

The relative error ϵ is calculated as in equation (4.1), the ten lowest eigenfrequencies are used and ω_i^{ref} are the converged eigenfrequencies with a precision of two decimals, which are found during mesh refinement.

$$\epsilon = \sum_{i=1}^{10} \frac{|\omega_i - \omega_i^{ref}|}{\omega_i^{ref}} 100 \tag{4.1}$$

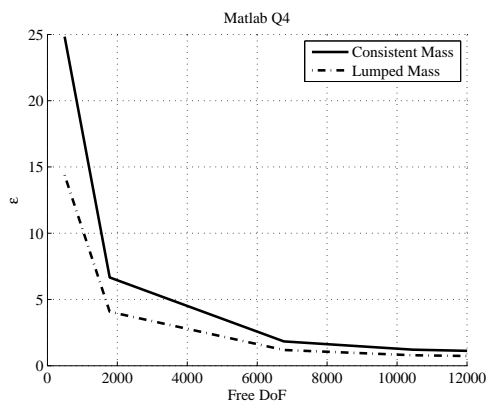


Figure 4.60: The relative error in frequencies for the MATLAB Q4 element using consistent and lumped mass matrix.

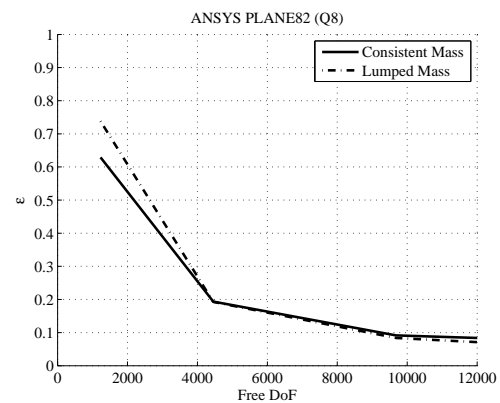


Figure 4.61: The relative error in frequencies for the PLANE82 element using consistent and lumped mass matrix.

As illustrated on Figure 4.60 and 4.61 the difference between a consistent and lumped mass matrix is minor for this case.

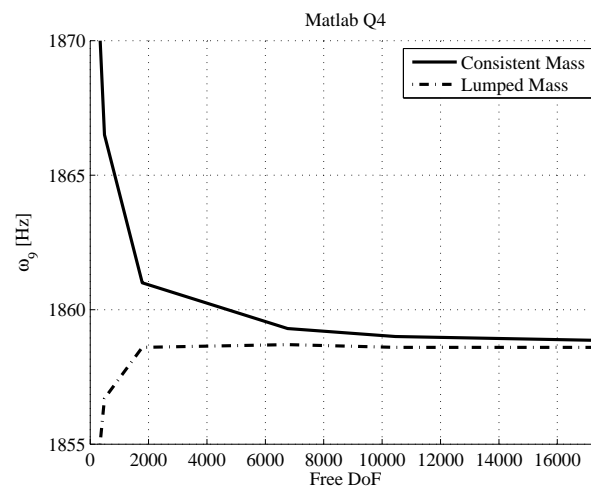


Figure 4.62: Convergence of the ninth eigenfrequency. Calculated using MATLAB Q4 element with mesh refinement.

If the element stiffness is not softened by low-order integration rules, and the mass matrix is consistent, then computed natural frequencies are upper bounds to the exact frequencies of the mathematical model. [Cook et al., 2002]. On Figure 4.62 it is seen that using the consistent mass matrix the computed eigenfrequencies are upper bounds of the exact frequencies, this is not the case when using lumped masses as also illustrated on Figure 4.62.

In this specific case the lumped mass matrix often results in improved results using the same mesh. This is partly due to the simple geometry and the square elements. If a complex shape and distorted elements were used the error using lumped mass matrix might have been larger than using a consistent mass matrix.

Using lumped mass the reduction in computational is time negligible for small problems. But when using ANSYS as solver, ANSYS will write relatively large amounts of data to the hard drive if the computer is lacking memory, and since only the diagonal is saved when using lumped masses the mass matrix uses a lot less memory. Furthermore the slow process of writing to the hard drive can be avoided. Therefore the lumped mass matrix is recommended especially for large problems.

4.9 Choice of Element

As described in section 3.2 different elements are implemented. In the MATLAB code a 2D bilinear Q4 plane stress element is implemented in the MATLAB program. The element is implemented in the same manner as in Sigmund [2001]. This simple element requires a filter in order to avoid the checkerboard pattern in the topology as described in section 4.10. Thus a sensitivity filter is used to obtain mesh independent solutions. When ANSYS is used as the

solver it is easy to switch between different types of elements as described in section 3.2. The difference between the elements is investigated in this section. The standard configuration is used. The result for each of the different element types is listed in Table 4.8.

Table 4.8: Dependency of the element type on the final topology.





Element Type	ω_1 [rad/s] Iter MoD [%]	Topology
MATLAB Q4	402.16 138 11.7	Density Plot at Iter No.138 Volume Fraction 50.0% 
PLANE42	401.62 132 11.8	Density Plot at Iter No.132 Volume Fraction 50.0% 
PLANE42QM6	400.60 139 11.7	Density Plot at Iter No.139 Volume Fraction 50.0% 
PLANE82	396.31 148 11.8	Density Plot at Iter No.148 Volume Fraction 50.0% 

Table 4.8 shows the difference between the different element types is negligible, as long as the sensitivity filter is used. The MATLAB element is the fastest for small problems, and the PLANE42 for large problems. Especially the PLANE82 elements is significantly slower and the results are remarkable similar for constant volume fractions.

Three more elements are available in the ANSYS code which are newer versions of the shown elements, but there are no advantages by using them, thus these are not discussed.

As Figure 4.63 shows the MATLAB solver is faster when using less than 30,000 elements. This is due to the time spend on opening and closing ANSYS. It is clear, that the eight node PLANE82 element is relatively slow compared to the ANSYS PLANE42 Q4. When using the different elements with a mesh independent filter no benefits are obtained using the PLANE82 element it is recommended using Q4 elements. One advantage using a higher order element is that a filter is not necessary to avoid checker boards, this is shown in section 4.10.

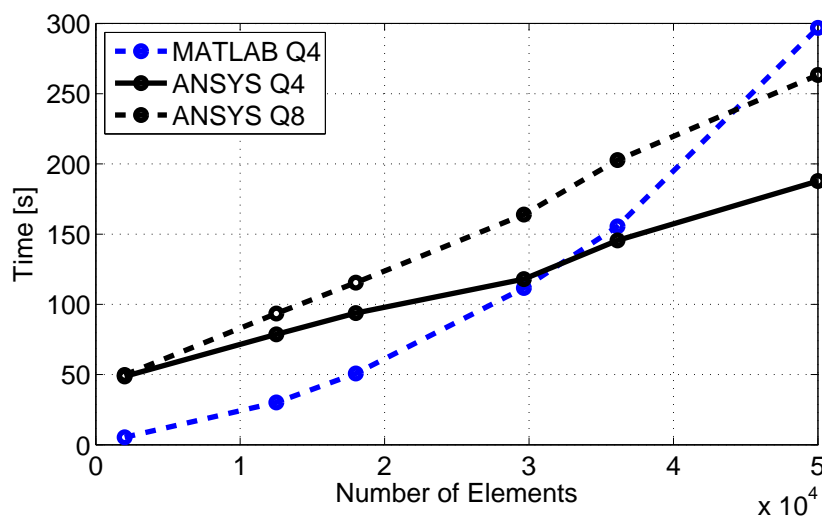


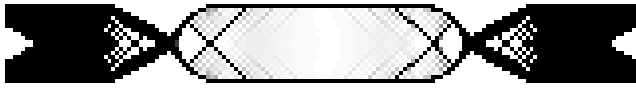



Figure 4.63: Computation time for ten iterations as a function of number of elements

Exactly the same is the case when using SIMP, the same topology is obtained using different elements as long as the filter settings are the same.

4.10 Filtering




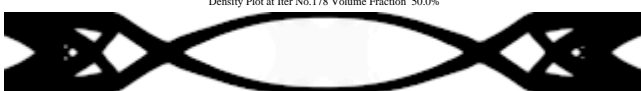


As stated in section 4.9 Q4 elements has a problem of creating the so-called checkerboard, see Table 4.9. As seen in the table it is not a problem for Q8 elements. Using filtering it is possible to remove these checkerboards which is also shown in Table 4.9. Using a too small filter, e.g. filter radius of 1.1, it is seen that the one node hinges are not avoided, thus a filter which both prevents checkerboards and one node hinges are desired.

Table 4.9: Formation of checkerboard. Results using a sensitivity filter with radius of 1.5 are seen in Table 4.10.

Filter radius and Element Type	ω_1 [rad/s] Iter MoD [%]	Topology
0 MATLAB Q4	437.22 147 6.6	Density Plot at Iter No.147 Volume Fraction 49.8% 
0 ANSYS Q8	418.09 92 12.3	Density Plot at Iter No. 92 Volume Fraction 49.8% 
1.1 MATLAB Q4	426.74 270 10.9	Density Plot at Iter No.270 Volume Fraction 49.9% 
1.1 ANSYS Q8	421.66 238 9.9	Density Plot at Iter No.238 Volume Fraction 49.9% 



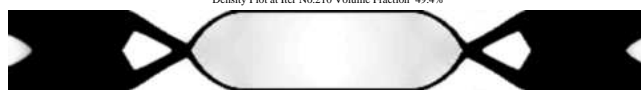
Furthermore if mesh refinement is used while keeping the filter radius constant, then the solution becomes mesh dependent, as seen in the latter three topologies in Table 4.10. However by keeping the filter radius constant with respect to the geometry, it is seen, that mesh independent designs can be obtained as seen in the first three topologies in Table 4.10. As seen in the table a mesh dependent solution allows for the construction of small members, thus it is possible to obtain higher fundamental frequencies compared to the mesh independent filtering.

Table 4.10: Mesh independent filtering for Olhoff(c) boundary conditions

Filter radius and Mesh	ω_1 [rad/s] Iter MoD [%]	Topology
1.5 3200	401.98 141 11.8	Density Plot at Iter No.141 Volume Fraction 50.0% 
3.0 12800	398.18 154 11.8	Density Plot at Iter No.154 Volume Fraction 50.0% 
4.5 28800	404.93 163 12.3	Density Plot at Iter No.163 Volume Fraction 50.0% 
1.5 12800	437.22 178 7.1	Density Plot at Iter No.178 Volume Fraction 50.0% 
1.5 28800	454.51 363 6.0	Density Plot at Iter No.363 Volume Fraction 50.0% 
1.5 51200	462.36 531 5.7	Density Plot at Iter No.531 Volume Fraction 50.0% 

The results in Table 4.10 are obtained using RAMP with sensitivity filtering. A density filter is also implemented, which is done on basis of Sigmund [2007] and Andreassen et al. [2011]. During testing of the density filter it is observed, that the performance is much worse than using the sensitivity filter this is shown in Table 4.11.

Table 4.11: Density filtering for Olhoff(c) boundary conditions.

Filter radius and Mesh	ω_1 [rad/s] Iter MoD[%]	Topology
1.5 3200	518.00 90 12.8	Density Plot at Iter No. 90 Volume Fraction 44.4% 
1.5 12800	413.85 209 16.6	Density Plot at Iter No. 209 Volume Fraction 49.9% 
1.5 28800	442.03 210 13.8	Density Plot at Iter No. 210 Volume Fraction 49.4% 

4.11 Optimizing for Specific Frequencies

Up until now the focus has been on maximizing the lowest frequency and investigate the different behavior of the code when changing the settings. Now the focus will change on how to obtain a topology that fulfills the request of matching a set of specific eigenfrequency. Optimization in this section is conducted using the standard configuration.

In order to target specific frequencies the f and g constraints must be changed. As described in section 2.4.1 the new constraint functions are split up into two types, i.e. target frequencies and remaining frequencies. The N_T target frequencies are formulated so the deviation must be minimized as stated in equation (2.31) and repeated here.

$$f_j(x_i) = \left(\lambda_j - \lambda_j^{(t)} \right)^2, \quad j = 1, \dots, N_T \quad (2.31)$$







From this it is seen, that in order to minimize f_j , $\lambda_j = \lambda_j^{(t)}$. The sensitivities are given as:

$$\frac{\partial f_j}{\partial x_i} = 2 \left(\lambda_j - \lambda_j^{(t)} \right) \frac{\partial \lambda_j}{\partial x_i} \quad (4.2)$$

The remaining eigenvalues are subjected to a “safety-zone” constraint which defines the minimum value for the remaining eigenfrequencies. Later in this section the effect of the safety-zone is shown. The purpose of the safety-zone is to force non specified modes away from the target eigenpairs, this will ensure that the response is governed by the specified eigenfrequencies.

In Table 4.12 a list of results obtained targeting various (lowest) frequencies are shown. It is easily seen that the algorithm is able to obtain topologies with specific frequencies. If the target frequency is too low it is not possible to obtain a topology with a frequency as low as the target frequency. On the other hand if the target frequency is too high, the algorithm will violate the volume constraint in an attempt to obtain a topology with the target frequency.

Table 4.12: Results when targeting various specified fundamental frequencies using Olhoff(c) boundary conditions. No safety-zone is applied.

Target ω_1 [rad/s]	ω_1 [rad/s] Iter MoD [%]	Topology
MAX-MIN	399.17 360 13.2	Density Plot at Iter No.360 Volume Fraction 50.0% 
400	397.90 229 12.4	Density Plot at Iter No.229 Volume Fraction 50.0% 
350	349.84 60 32.7	Density Plot at Iter No. 60 Volume Fraction 49.9% 
150	150.03 8 98.4	Density Plot at Iter No. 8 Volume Fraction 49.8% 
10	15.56 41 32.6	Density Plot at Iter No. 41 Volume Fraction 40.4% 
500	329.75 105 3.8	Density Plot at Iter No.105 Volume Fraction 86.9% 

The penalty on the stiffness alone is not sufficient in order to obtain 0-1 designs, since the algorithm has no disadvantage by choosing a non discrete density. It might be advantageous to choose intermediate densities if the frequency is higher than the target frequency (ω'), whereas it is beneficial from a max-min point of view to choose 0 or 1. One solution to this problem is suggested in section 4.12.




It is possible to target as many eigenfrequencies as the number of eigenpairs extracted. How-

ever in order to have information concerning the higher frequencies it is recommended to extract at least twice the amount of eigenpairs.

In Table 4.13 three topologies are shown here the fundamental frequency is specified to target 300 [rad/s]. The difference exist for the second target eigenfrequency ($\omega_2^{(t)}$). For the first topology $\omega_2^{(t)}$ is not specified, whereas for the latter two it is given as 400 and 500 [rad/s] respectively.

From the table it is seen, that it is possible to target a higher and lower frequency for the second eigenfrequency than the one where it is not specified.

Table 4.13: Topologies obtained for various specified frequencies illustrating the capability of obtaining different frequencies for ω_2 .

Target $\omega_1^{(t)}$ [rad/s] $\omega_2^{(t)}$ [rad/s]	ω_1 [rad/s] ω_2 [rad/s] ω_3 [rad/s] ω_4 [rad/s]	Iter MoD [%]	Topology
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = \text{None}$	$\omega_1 = 298.0$ $\omega_2 = 461.1$ $\omega_3 = 662.5$ $\omega_4 = 763.2$	81 50	Density Plot at Iter No. 81 Volume Fraction 50.0% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 400$	$\omega_1 = 299.3$ $\omega_2 = 402.4$ $\omega_3 = 637.7$ $\omega_4 = 652.0$	50 52.8	Density Plot at Iter No. 50 Volume Fraction 50.0% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 500$	$\omega_1 = 300.1$ $\omega_2 = 500.0$ $\omega_3 = 694.3$ $\omega_4 = 811.4$	55 45.4	Density Plot at Iter No. 55 Volume Fraction 49.9% 

When targeting eigenvalues it might be, that the higher eigenfrequencies are too close to the target frequencies, thus these must be removed.


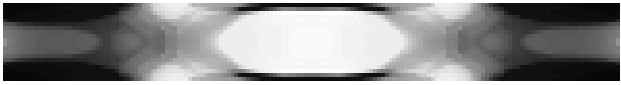
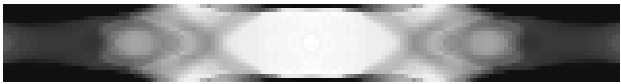

The safety-zone is given as a relative number i.e.:

$$\omega_{safety} = \omega_{sz} + \max \left(\omega^{(t)} \right)$$

ω_{safety} can be rewritten to eigenvalues (λ_{safety}), thus giving the g constraints from equation (2.32) as:

$$g_k(x_i) - \bar{g}_k = \lambda_{safety} - \lambda_k \leq 0, \quad k = N_T + 1, \dots, N \quad (4.3)$$

Table 4.14: Effect of safety-zone on final topology.

Target $\omega_1^{(t)}$ [rad/s] $\omega_2^{(t)}$ [rad/s] ω_{sz} [rad/s]	ω_1 [rad/s] ω_2 [rad/s] ω_3 [rad/s] ω_4 [rad/s]	Iter MoD[%]	Topology
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 400$ $\omega_{sz} = 410$	$\omega_1 = 299.5$ $\omega_2 = 400.2$ $\omega_3 = 637.0$ $\omega_4 = 648.4$	50 52.8	Density Plot at Iter No. 50 Volume Fraction 50.0% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 400$ $\omega_{sz} = 625$	$\omega_1 = 299.9$ $\omega_2 = 400.6$ $\omega_3 = 658.1$ $\omega_4 = 688.4$	44 59.7	Density Plot at Iter No. 44 Volume Fraction 49.9% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 400$ $\omega_{sz} = 700$	$\omega_1 = 300.0$ $\omega_2 = 399.8$ $\omega_3 = 721.4$ $\omega_4 = 792.7$	91 60.4	Density Plot at Iter No. 91 Volume Fraction 49.9% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 400$ $\omega_{sz} = 1200$	$\omega_1 = 299.8$ $\omega_2 = 399.8$ $\omega_3 = 1194.5$ $\omega_4 = 1195.4$	1001 20.6	Density Plot at Iter No.1001 Volume Fraction 53.7% 

From Table 4.14 it is seen, that when the safety-zone is small, then the final topology is not affected, which is as expected since the constraint is inactive. Raising the safety-zone it is seen, that the topology changes, and if a too large safety-zone is chosen it is not possible to fulfill the constraint.

But it is also seen, that even though the safety-zone is not fulfilled the lowest frequencies are close to the target frequency. Still many gray elements are present as seen in the table. One method to eliminate these is presented in the following section.

4.12 Eliminating Gray Elements

Since RAMP and SIMP stiffness interpolations are used no information concerning the physical properties of gray elements are present. When topology optimizing a 3D structure the density represent the material properties, and if the optimum topology contains a large amount of intermediate densities it is difficult to interpret the design. Therefore the results shown in Tables 4.12 to 4.14 are of little interest. In this section one method to obtain a discrete design is presented, first the method is tested on the max-min Olhoff(c) problem, and it is then implemented in order to reduce the intermediate densities when targeting specific

frequencies.

The method used here is based on the measure of discreteness from equation (2.36). In appendix A it is seen, that the “y” variables are present both in the descent function, but also in the constraints. This is used to obtain a low measure of discreteness.

Using the same approach as for the “safety zone” the constraint based on the measure of discreteness is defined as:

$$g = MoD - MoD^{(t)}$$

Here $MoD^{(t)}$ is the target measure of discreteness. Inserting this into the MMA formulation in equation (A.1) the constraint will have the form:

$$MoD - MoD^{(t)} - y \leq 0 \quad (4.4)$$

It is seen, that the y variable will attain a value higher than zero when the measure of discreteness is above the target measure of discreteness. The value of the y variable is included in the descent function, thus raising the value of this. When the target eigenfrequencies are reached the measure of discreteness must be lowered in order to attain a lower descent function value, thus making it uneconomical to have intermediate densities.

In order to have a small as possible influence on the solution, this constraint is weighted very low at the beginning of the optimization and then gradually raised to penalize gray elements. This is done by altering the c constant of the MMA algorithm.

Table 4.15: Effect of discreteness constraint on the final topology when varying the element size using a mesh independent sensitivity filter.

Discreteness Target and Mesh	ω_1 [rad/s] Iter MoD [%]	Topology
$MoD^{(t)} = \text{None}$ 3200	400.74 275 12.0	Density Plot at Iter No.275 Volume Fraction 50.0%
$MoD^{(t)} = 0\%$ 3200	406.71 185 7.4	Density Plot at Iter No.185 Volume Fraction 50.0%
$MoD^{(t)} = 0\%$ 12800	403.52 432 3.6	Density Plot at Iter No.432 Volume Fraction 51.4%

In Table 4.15 the effect of the discreteness constraint is shown. It is clear from the two top figures that the same topology and a more discrete design is obtained. The topology has generally not changed by adding the discreteness constraint, even though the discreteness target is as low as 0%.

Note that the discreteness constraint is not fulfilled, this is due to the low c constant for the measure of discreteness used in the MMA formulation. The goal here is to obtain a solution with a clear topology not to obtain a solution where the measure of discreteness is 0%. In the lower illustration of Table 4.15 the topology when using a finer discretization is shown. It shows how the measure of discreteness can be improved when using a finer mesh. This is an effect of the better representation of the topology. Note that the radius of the sensitivity filter is set, so mesh independent solutions as described in section 4.10 are obtained, and the convergence criteria is lowered to 0.005 to avoid convergence before a discrete design is obtained.

On Figure 4.64 the c constant for the measure of discreteness of the middle solution in Table 4.15 is shown as a function of the iterations. The constraint is active in the first iteration since the measure of discreteness starts at 50%, but because the discreteness constant in the MMA formulation starts at $5 \cdot 10^{-5}$ it has nearly no influence on the descent function. The c constant is gradually increased by 0.004 if the change in MoD is less than 3% over the last four iterations. This ensures a steady decrease in MoD.

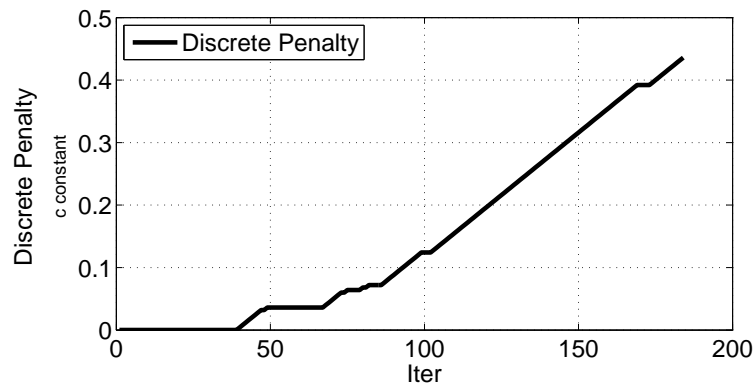


Figure 4.64: The discreteness penalty (c constant) as a function of iterations. The Measure of Discreteness target is 0%. This target is never reached as seen on Figure 4.66.

The difference in MoD when using the discreteness constraint and not using it is illustrated in Figure 4.65 and 4.66, where it is seen, that there is no difference in the first 50 iterations until the c constant from Figure 4.64 gradually is increased. The discreteness c constant is intentionally kept low at the beginning, and raised in small steps. This is done in order to give the targeting of the frequencies more importance, and then remove the gray elements after the frequencies are targeted.

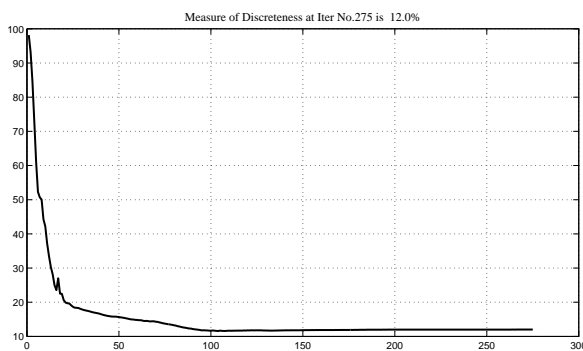


Figure 4.65: The Measure of Discreteness as a function of the iterations without a discreteness target, corresponding to the top picture in Table 4.15.

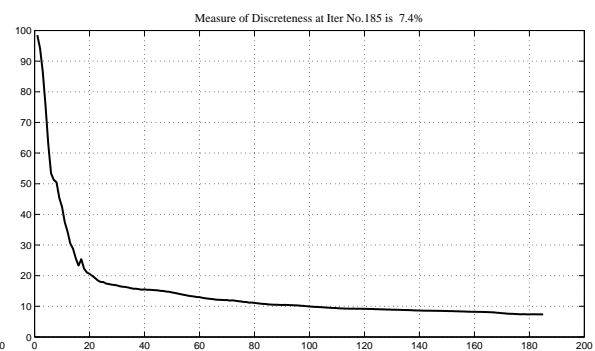


Figure 4.66: The Measure of Discreteness as a function of the iterations with a discreteness target of 0%, corresponding to the middle picture in Table 4.15.






4.12.1 Targeting frequencies

As described in section 4.11 the problem is more difficult when targeting a specific frequency, especially if the frequency is relatively low. In Table 4.16 5 results are listed. Here three of them are constrained with a measure of discreteness of 0%. It is not recommended to specify a measure of discreteness of 0% and it is highly unlikely that any algorithm will

achieve a measure of discreteness that low. This is only conducted here to show the extreme case and investigate how low a measure of discreteness it is possible to obtain.

In the first two illustrations in Table 4.16 only one target frequency is desired. The difference between using a discreteness constraint and not is seen. Using the measure of discreteness as a constraint effects the topology, since it nearly only consisting of 0 and 1 densities.

Table 4.16: Effect of Measure of Discreteness on the final topology when targeting frequencies.

Target $\omega_1^{(t)}$ [rad/s] $\omega_2^{(t)}$ [rad/s] $MoD^{(t)}$ [%]	ω_1 [rad/s] ω_2 [rad/s] ω_3 [rad/s] ω_4 [rad/s]	Iter MoD [%]	Topology
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = \text{None}$ $MoD^{(t)} = \text{None}$	$\omega_1 = 298.0$ $\omega_2 = 461.1$ $\omega_3 = 662.5$ $\omega_4 = 763.2$	81 50	Density Plot at Iter No. 81 Volume Fraction 50.0% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = \text{None}$ $MoD^{(t)} = 0$	$\omega_1 = 298.9$ $\omega_2 = 303.5$ $\omega_3 = 405.1$ $\omega_4 = 428.5$	174 1.8	Density Plot at Iter No. 174 Volume Fraction 49.8% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 500$ $MoD^{(t)} = \text{None}$	$\omega_1 = 300.1$ $\omega_2 = 500.0$ $\omega_3 = 694.3$ $\omega_4 = 811.4$	55 45.4	Density Plot at Iter No. 55 Volume Fraction 49.9% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 500$ $MoD^{(t)} = 0$	$\omega_1 = 297.2$ $\omega_2 = 494.5$ $\omega_3 = 499.5$ $\omega_4 = 618.3$	206 2.9	Density Plot at Iter No. 206 Volume Fraction 50.0% 
$\omega_1^{(t)} = 300$ $\omega_2^{(t)} = 500$ $MoD^{(t)} = 0$ $\omega_{sz} = 50$	$\omega_1 = 299.4$ $\omega_2 = 498.9$ $\omega_3 = 549.6$ $\omega_4 = 622.3$	266 3.2	Density Plot at Iter No. 266 Volume Fraction 50.0% 

In illustration three and four the two lowest frequencies are targeted, note that ω_2 is moved from 303.5[rad/s] to 494.5[rad/s] and how the topology change. The measure of discreteness is slightly higher compared to targeting only one frequency and ω_2 and ω_3 are relative close. This could be a problem, therefore a safety zone of 50[rad/s] is added to the problem formulation. The results are shown as the last illustration in Table 4.16. The two targeted frequencies are within 0.3% of the desired values, which is better than expected and the measure of discreteness are still sufficiently low.

It seems that the method of constraining the design to a low measure of discreteness results in discrete topologies. One problem is, that the connectivity of the last topology is lost, and many one-node hinges are present. These problems are due to the target for measure of discreteness has been 0% , this is the extrema and depending on the mesh size, geometry and material at disposal. A measure of discreteness of less than 10% should be enough in practical applications, since it provides a clear topology without the existence of one-node hinges.

4.13 Mode Tracking

Mode tracking, as described in section 2.3, is a method for measuring the difference between two eigenvectors. This method can be used to track or identify the modes as the topology changes. First a reference vector is created which represent the mode shape to be tracked. This reference vector is then compared to the eigenvectors obtained in an iteration. If the MAC value is close to one the modes are considered identical and the mode shape is identified. Since the eigenmodes are linear independent one mode should have a MAC value much higher than the others, meaning it is possible to track the modes through the iterations.

The important part is how to create a reference vector which represent the desired mode, since it is difficult to construct a reference vector which describes the global shape of a mode before the optimization is initiated. Therefore the method presented here will use an eigenvector calculated during the optimization as the reference vector.

It is convenient to pick the reference mode as the eigenvectors obtained in the first iteration, this might create some problems. Since the topology changes during the optimization the desired mode is not necessarily one of the lowest eigenmodes, thus cannot be selected in the first iteration. If one of these modes are to be tracked the user is able to associate the modes during the optimization using the "Associate Modes" button. Only the modes associated with a reference frequency are tracked using MAC. This allows higher modes to switch and ensuring that the problem is not unnecessary constrained.

The button enables the user to select the reference modes during the optimization and makes it possible to reselect a new set of reference modes as the topology change during the optimization. In some cases the best strategy is to associate the modes in the first few iterations while in other cases it may be advantageous to wait until a desired mode is present.

The MAC values for global modes appears to retain a value close to one although the topology might have changed considerably. Therefore there is no reason to redefine the reference mode in each iteration.

The implemented method has some drawbacks. It is difficult to produce the same results several times, since it requires user input to associate the modes. But if the desired modes are present after the first iteration it is relatively quick to associated the modes.

4.13.1 Mode Switching

An example where the fourth mode shape is moved down as the second mode is illustrated on Figure 4.67 and 4.68. This example is based on the last solution from Table 4.16. The target frequencies are still $\omega_1^{(t)}=300$, $\omega_2^{(t)}=500$ and $\omega_{sz}=50$, but $MoD^{(t)}$ is increased from 0% to 5%. The modes are associated in iteration 37, because the fourth mode in Figure 4.67 needs to be present before it can be associated as a reference mode shape. The MoD is approximately 16% when the modes are associated in iteration 37.

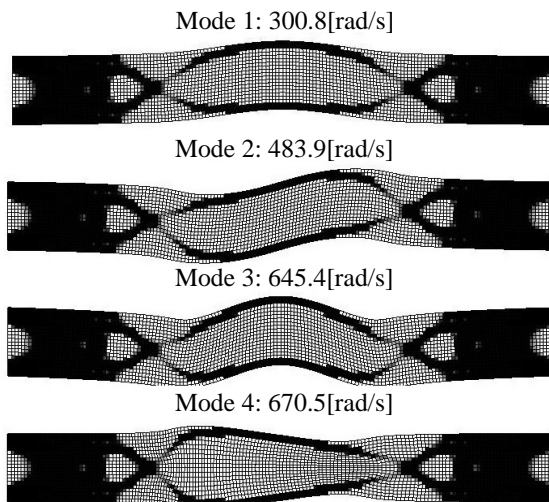


Figure 4.67: The four mode shapes in iteration 37 where the first mode is associated with a target frequency of 300[rad/s] and the fourth mode is associated with a target frequency of 500[rad/s].

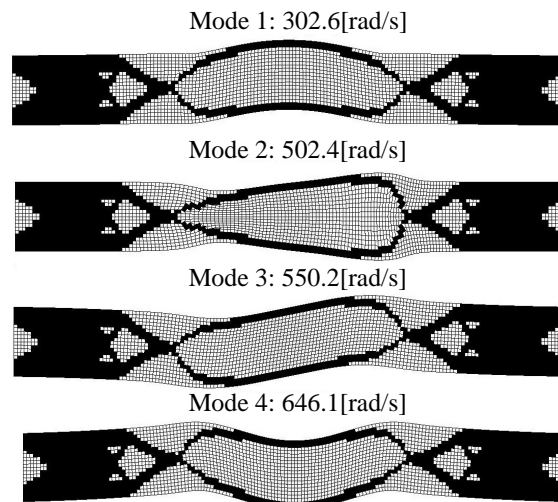


Figure 4.68: The four mode shapes after convergence in iteration 120. Here the breathing mode is moved from the fourth to the second mode.

This results in a clear 0-1 topology where the frequencies and modes are close to the desired values. The change in frequencies through the iterations is shown on Figure 4.69. Since the topology changes the most in the first 20 iterations the modes also changes a lot while they are switching. This appears as a "zig zag" behavior in the frequency plot. After approximately 20 iterations the four lowest frequencies stop switching until the fourth mode is associated with the second eigenfrequency at iteration 37. Here it is seen, that the third eigenvalue drops, thus switches with the second while the fourth mode is moved down as the second mode shape.

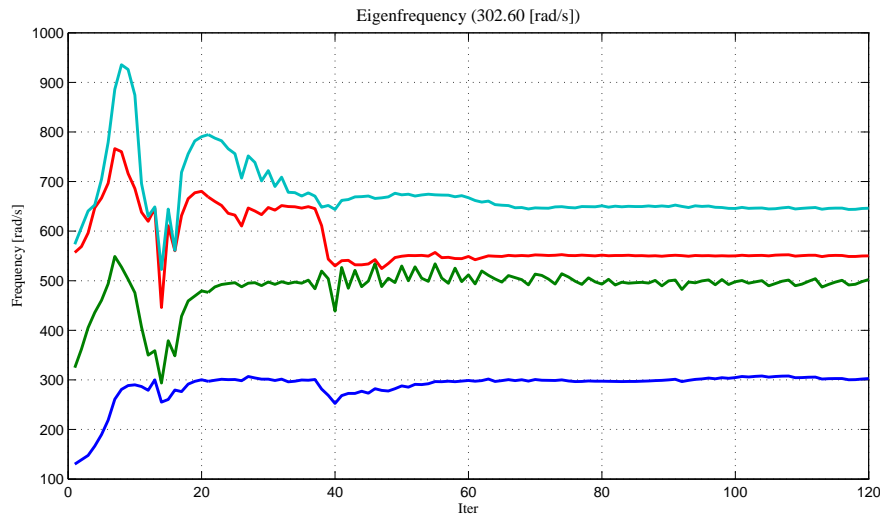


Figure 4.69: The change in frequencies through the iterations. The fourth mode is associated with the second frequency target of 500[rad/s] in iteration 37, after which the three highest modes are switching.

In Table 4.17 the results with and without the mode switching is listed. The two topologies are alike but the small difference is enough to change the respond of the structure significantly. As it is clear form Figure 4.67 and 4.68 the breathing mode is moved from the fourth mode all the way down to the second mode, and this is while the frequency of the modes are changed to the target value. The difference between the desired frequencies and the actual frequencies are less than 1%. The safety zone is nearly fulfilled and the design is a clear 0-1 design. This proves that it is possible to use optimization methods to produce designs with a specific response.

Table 4.17: Targeting frequencies with measure of discreteness, safety zone and associating modes. Top result: without mode switching. Bottom result: A breathing mode is associated with the second eigenfrequency.

Target	ω_1 [rad/s]	Iter	Topology
$\omega_1^{(t)}$ [rad/s]	ω_2 [rad/s]		
$\omega_2^{(t)}$ [rad/s]	ω_3 [rad/s]	MoD[%]	
$MoD^{(t)}$ [%]	ω_4 [rad/s]		
$\omega_1^{(t)} = 300$	$\omega_1 = 299.4$	266 3.2	Density Plot at Iter No.266 Volume Fraction 50.0%
$\omega_2^{(t)} = 500$	$\omega_2 = 498.9$		
$MoD^{(t)} = 0$	$\omega_3 = 549.6$		
$\omega_{sz} = 50$	$\omega_4 = 622.3$		
$\omega_1^{(t)} = 300$	$\omega_1 = 302.6$	120 4.8	Density Plot at Iter No.120 Volume Fraction 49.8%
$\omega_2^{(t)} = 500$	$\omega_2 = 502.4$		
$MoD^{(t)} = 5$	$\omega_3 = 550.2$		
$\omega_{sz} = 50$	$\omega_4 = 646.1$		

Although this example seems to be relatively easily solved one could be misled to believe, that this is always the case. In this example the breathing mode which is relatively decoupled from the three bending modes, making it easier to switch with the bending modes. It is much tougher to switch two bending modes because they are closely related.

4.14 3D Implementation of the Code

Before the test rig for the nacelle is designed the 3D implementation of the code is tested on a structure which resembles the nacelle.

Since the 3D code is an extension of the 2D code the flow of the code is the same as in Figure 3.1 with the only exception of ANSYS being the only implemented solver.

Compared to a 2D problem, a 3D problem requires more elements to provide a good description of the design area since a representation in 3D is needed. Furthermore more degrees of freedom are present in each element. This is seen by considering the simplest 3D brick element i.e. the 8 node trilinear brick element to its 2D counterpart; the 4 node plane bilinear Q4 element. It is seen, that three times as many degrees of freedom are present in the brick element compared to the plane element. Furthermore, the bandwidth of the global stiffness matrix is much larger. This results in a longer computation time, and higher memory requirement. A problem can easily be formulated larger than a normal or fast desktop PC can possibly handle, and the computational times easily multiplied with a factor of 100 compared to a 2D problem. For simple structures this results in computation time increases from a couple of minutes to several hours.

An other problem is, that a fine representation of the design space is desired, thus causing the amount of elements to increase dramatically since a halving of the element size results in eight times as many elements. An ANSYS Academic Teaching Advanced license is used and a too fine mesh will violate the license restrictions. Using an ANSYS Academic Teaching Advanced license the maximum number of nodes and elements is restricted to 256,000.

This will also pose a problem when optimizing the test rig. This is further discussed in chapter 5.

To verify it is possible to handle a 3D problem just as well as the 2D problems the structure in Figure 4.70 is optimized. The structure is a rectangular design domain fixed at the bottom with a volume of $(2200 \times 3000 \times 4000 \text{ [mm]})$, as illustrated on Figure 4.70. The settings for the optimizations are seen in Table 4.18. This example represents a version of the test rig problem in chapter 5. Therefore a plate with a thickness of 200 [mm] is predefined at the top. As seen in Table 4.18 the model is meshed with 100 [mm] SOLID45 elements giving $22 \times 30 \times 40$ elements in each direction. The elements in the plate are specified

being massive and represents the nacelle in this example.

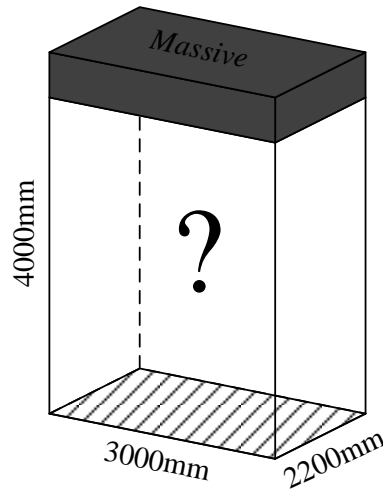


Figure 4.70: The rectangular design domain fixed at the bottom.

In the rectangular design domain the density is initially evenly distributed. The first two modes are bending modes. These modes are used when optimizing for specific frequencies.

The problem is solved in four different ways. In the first example the fundamental frequency is maximized. The second where the lowest bending mode is matched to the lowest target frequency and the other bending mode to the second target frequency. The third is similar to the second besides a safety zone of 20 [Hz] is applied, and finally in the fourth example the two modes are swapped, and is otherwise similar to the third example.

Treating the problem as a max-min the topology in Figure 4.71 is found. The four frequencies listed in Table 4.19 indicate the maximum frequency possible to specify as target frequency.

Subsequently the example is treated as a problem where ω_1 and ω_2 are specified to 30 and 80 Hz. In the second row of Table 4.19 the results from the topology optimization is seen. It is seen, that the frequencies are within 1.5 [Hz] of the target frequencies. However the Measure of Discreteness is not fulfilled. The topology on Figure 4.72 is seen to contain elements which are connected to low density elements. This does not cause local modes and when interpreting the design these elements are removed.

Since the second and third eigenfrequencies are close a safety zone of 20 [Hz] is applied. From Table 4.19 it is seen, that all constraints are fulfilled while the frequencies are within 1.5 [Hz] of the target frequencies. The topology is seen in Figure 4.71.

Finally the first two bending modes are associated with the opposite frequencies in the second iteration, this results in a completely different topology although the two first frequen-

cies are nearly the same, as illustrated on Figure 4.73 and 4.74. It is seen, that the deviation is slightly larger (2.2 [Hz]), however the constraints are fulfilled.

Table 4.18: Settings for the 3D analyzes.

NE	26,400
Element Size	100 [mm]
Element Type	SOLID45
Volume Fraction	30 %
N	12
BC	As Figure 4.70
Filter	Sensitivity r=1.5
Convergence	$\max(\Delta x) \leq 0.01$
Mass Matrix	Lumped
Stiffness Interpolation	RAMP q = 8
Mass Interpolation	Olhoff-Du(c)
Min Density	10^{-8}
Max Iterations	1000
Discreteness Target	5%
Material Stiffness	210,000 [MPa]
Material Density	7800 [kg/m ³]

Table 4.19: The results for a max-min, two target and a target optimization where mode 1 and 2 are switch during the optimization.

No.	$\omega_1^{(t)}$ [Hz] $\omega_2^{(t)}$ [Hz] ω_{sz} [Hz] $MoD^{(t)}$ [%]	ω_1 [Hz] ω_2 [Hz] ω_3 [Hz] ω_4 [Hz]	Topology Iter MoD [%] Resource [%]
1	$\omega_1^{(t)}$ = None $\omega_2^{(t)}$ = None ω_{sz} = None $MoD^{(t)}$ = 5	ω_1 = 92.1 ω_2 = 92.2 ω_3 = 141.6 ω_4 = 155.1	Figure 4.71 737 5.2 30
2	$\omega_1^{(t)}$ = 30 $\omega_2^{(t)}$ = 80 ω_{sz} = None $MoD^{(t)}$ = 5	ω_1 = 29.8 ω_2 = 81.3 ω_3 = 84.3 ω_4 = 88.7	Figure 4.72 161 5.5 29.3
3	$\omega_1^{(t)}$ = 30 $\omega_2^{(t)}$ = 80 ω_{sz} = 20 $MoD^{(t)}$ = 5	ω_1 = 29.8 ω_2 = 81.1 ω_3 = 101.2 ω_4 = 107.9	Figure 4.73 181 5.0 26.4
4	$\omega_1^{(t)}$ = 30 $\omega_2^{(t)}$ = 80 ω_{sz} = 20 $MoD^{(t)}$ = 5	ω_1 = 32.2 ω_2 = 81.7 ω_3 = 104.9 ω_4 = 112.8	Figure 4.74 135 5.0 27.6

From the different examples it is seen, that the 3D code (as the 2D code) is able to obtain discrete topologies both when maximizing the fundamental frequency, and when targeting frequencies while using mode tracking.

As seen throughout this chapter a method which has the ability to match the eigenpairs from a model to a specified number of eigenpairs successfully is implemented. Referring to section 1.1 it is seen, that the research objective, i.e. *Match a specified number of distinct (simple) eigenfrequencies and -modes to a desired set of frequencies and modes by making a topology optimization code*, is fulfilled.

Since the objective is fulfilled the problem from FS Dynamics is now undertaken. Here the code is used to design a test rig for a wind turbine nacelle.

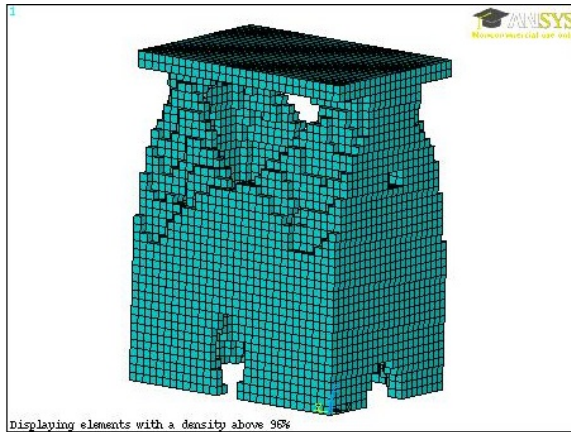


Figure 4.71: The topology obtained after 737 iterations using the configurations from Table 4.18 and 4.19 solving the maxmin problem.

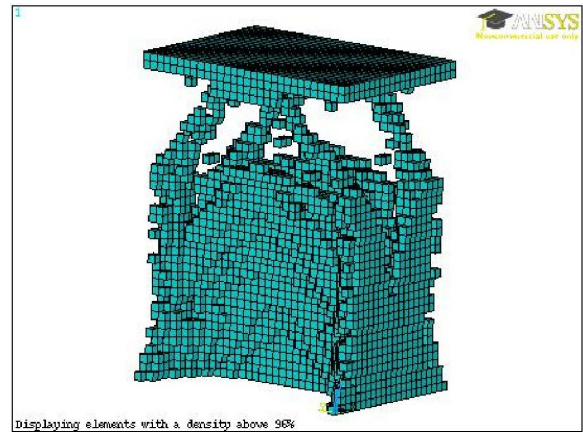


Figure 4.72: The topology obtained after 161 iterations using the configurations from Table 4.18 and 4.19 targeting two frequencies.

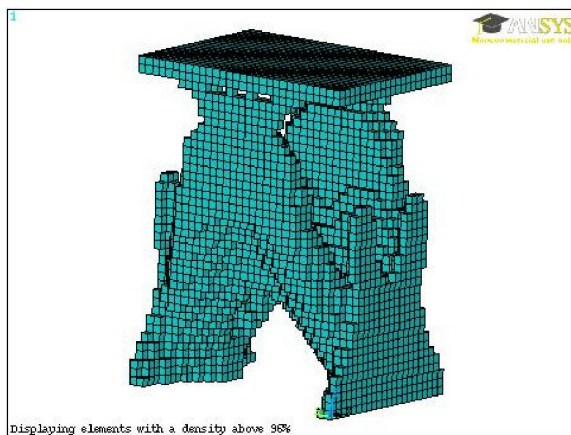


Figure 4.73: The topology obtained after 181 iterations using the configurations from Table 4.18 and 4.19 targeting two frequencies and with a safety zone of 20 Hz.

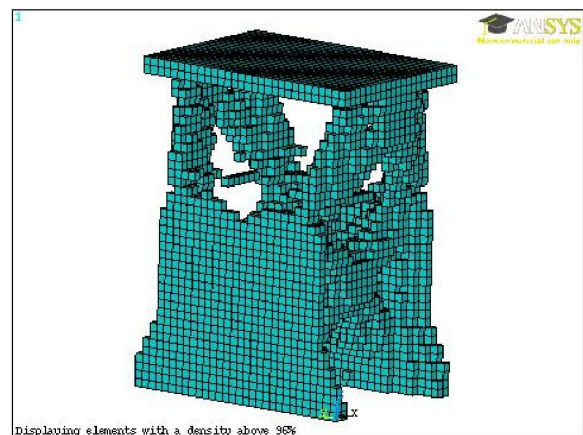


Figure 4.74: The topology obtained after 135 iterations using same configurations as Figure 4.73. Here ω_1 and ω_2 are associated with mode 2 and 1 respectively in the second iteration.

CHAPTER 5

Design of Test Rig for Wind Turbine Nacelle

In this chapter the code presented in chapter 3 and 4 is used to design a test rig for a wind turbine nacelle. As stated in chapter 1 this application of the code comes from the collaboration with FS Dynamics AB. The wind turbine model provided by FS Dynamics AB represents a simple version of a wind turbine. This poses no problem, since the final topology is a proof of concept showing the capabilities of the developed code. If the code has the ability of finding a suitable topology for the simple wind turbine nacelle it is also able to find a topology for a more complex structure.

The test rig is limited to a much smaller height than the wind turbine tower and it has to have the same response. The requests for the test rig are:

- Maximum dimensions $4000 \times 5000 \times 6500$ [mm] (Height \times width \times depth)
- Match the two lowest eigenfrequencies from the wind turbine model
- Match the two lowest mode shapes from the wind turbine model

This chapter contains a presentation of the provided wind turbine model, an analysis of the model, preparations of the model in order to reduce the iteration time and lastly the code is applied to design the topology of the test rig for the wind turbine nacelle.

5.1 Presentation and Analysis of the Provided Model

In this section the wind turbine model provided by FS Dynamics is analyzed. The structure is a simple version of a wind turbine model, and is created in ANSYS ADPL programming language.

The top part of the model is depicted on Figure 5.1. The model consists of a tower and a nacelle structure. The tower is 90 [m] high, and conically shaped with an outer diameter of 4200 [mm] at the bottom and 3000 [mm] at the top. The wall thickness of the tower is 22 and 16 [mm] at the bottom and top respectively.

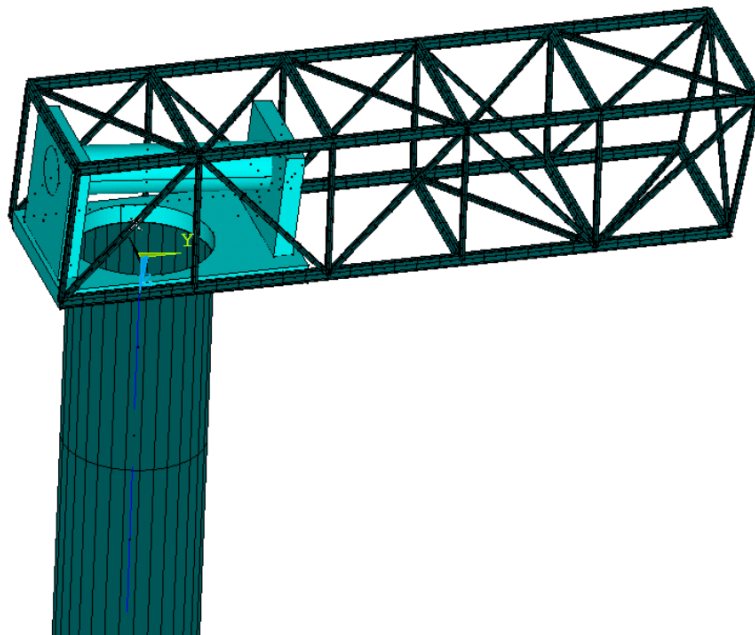


Figure 5.1: The top part of the wind turbine model provided by FS Dynamics AB [FS Dynamics, 2012].

An illustration of the different parts of the nacelle is seen on Figure 5.2 (bed frame, bearings, frame structure, shaft, hub + blades, gearbox and generator) and a plot of the beam sections and solid elements are seen on Figure 5.3. In Table 5.1 the masses for the parts are listed. As seen in Table 5.1 the model consists of different element types. To couple these, constraint equations are defined. In the model all couplings are rigid and are illustrated on Figure 5.4. The following couplings exist:

- Bed frame - Bearings
- Bed frame - Gearbox
- Bed frame - Frame structure
- Frame structure - Generator
- Bearings - Shaft
- Shaft - Hub + blades
- Shaft - Gearbox

Furthermore a coupling exists between the bed frame and an artificial dummy mass in the origin (on Figure 5.1 to 5.4). The artificial mass is used to model the interface between the nacelle and tower/test rig. Furthermore when analyzing the nacelle without the tower the support is located in the artificial mass.

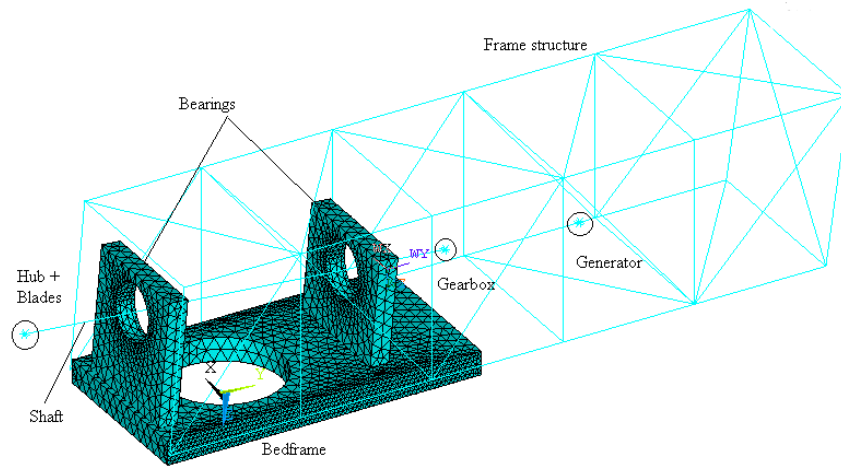


Figure 5.2: Plot of the provided nacelle. * displays discrete masses.

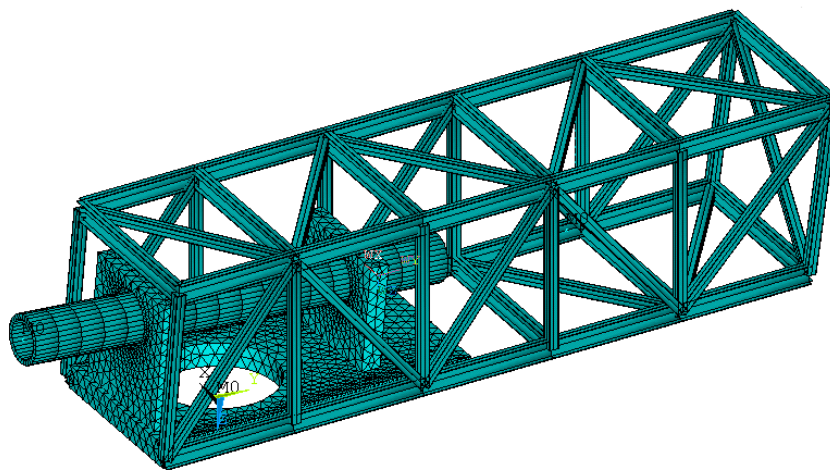


Figure 5.3: Plot of the solid elements and beam sections for the nacelle.

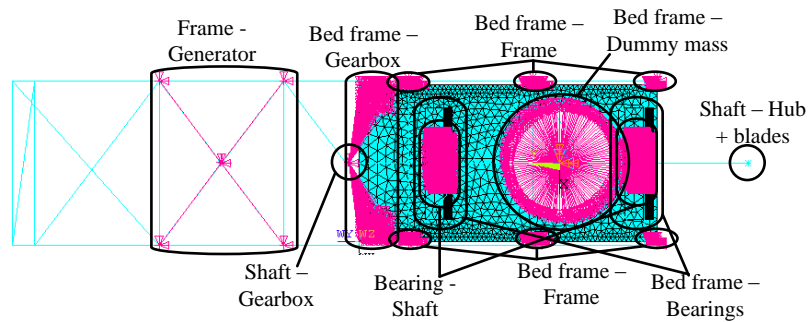


Figure 5.4: Plot of the couplings in the nacelle.

In Table 5.2 the elements used to model the nacelle are listed. Together with 10 BEAM189 elements in the tower, a total of 28,072 elements are used.

Table 5.1: Parts and corresponding mass for the nacelle.

Part	Mass [tonne]	Element type
Bed frame	72	Solid
Bearings	30	Solid
Shaft	6	Beam
Frame	11	Beam
Hub + blades	60	Mass
Gearbox	55	Mass
Generator	52	Mass
Total	286	

Table 5.2: Elements in the nacelle. Names from ANSYS are used.

Element type	Number
SOLID187	24,837
BEAM189	77
MASS21	4
TARGE170	3,052
CONTA174	92

A simulation where the first ten eigenpairs are calculated is preprogrammed for the provided model. The calculated eigenpairs are seen in Table 5.3.

Table 5.3: Eigenfrequencies and associated eigenmodes for the wind turbine model.

	Frequency [Hz]	Mode shape
ω_1	0.174	First tower bending around x-axis (Figure 5.5)
ω_2	0.263	First tower bending around y-axis (Figure 5.6)
ω_3	1.769	Second tower bending around x-axis
ω_4	1.897	Nacelle frame wriggle
ω_5	2.630	Second tower bending around y-axis + nacelle frame wriggle
ω_6	3.696	Second tower bending around x-axis + first nacelle frame bending
ω_7	5.895	Second tower and nacelle bending around x-axis
ω_8	6.743	Second tower bending around y-axis
ω_9	7.612	Second tower bending around x-axis + shaft and nacelle frame bending
ω_{10}	9.605	Third tower bending around x-axis + shaft and nacelle frame bending

From Table 5.3 it is seen that the two bending eigenpairs which are desired for the test rig have corresponding eigenfrequencies of $\omega_1 = 0.174[Hz]$ and $\omega_2 = 0.263[Hz]$. In Figure 5.5 and 5.6 the mode shapes corresponding to the frequencies are shown. These are the eigenpairs which

represent the response of the structure, thus are the eigenpairs desired to replicate in the test rig.

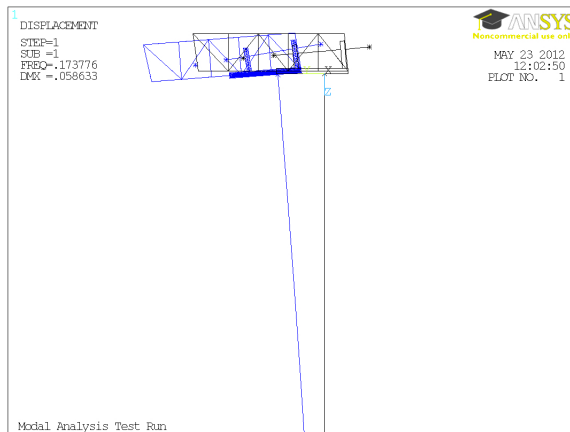


Figure 5.5: First bending mode, $\omega_1 = 0.174$ [Hz].

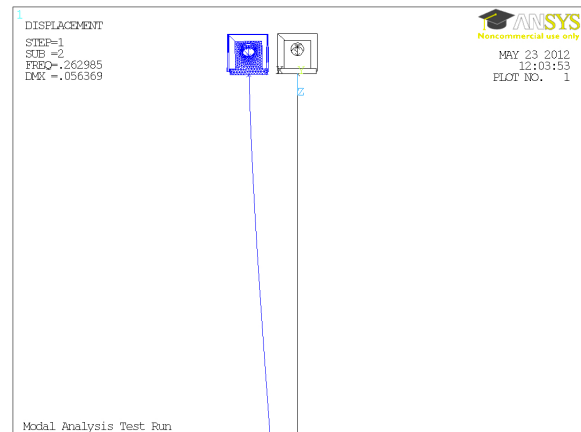


Figure 5.6: Second bending mode, $\omega_2 = 0.263$ [Hz].

5.1.1 Model Preparation

When designing the test rig, the total number of elements has a big influence on the computational time. Many elements in the nacelle will limit the number of elements which can be "afforded" in the design domain. Therefore it is beneficial only to use the necessary elements in the nacelle structure. Thereby the design domain can be constructed with as many elements as desired in a compromise between computational time and the representation of the discrete topology.

One approach is to create a super element to replace the nacelle. This approach resulted in the first two bending modes becoming a multiple eigenvalue with the frequency of ω_2 . Although the approach reduced the computational time from 37 to 3 [s] this method is discarded due to lack of accuracy.

Another approach is to use a coarser discretization in the nacelle. Therefore an analysis of the mesh size is conducted. In order to only analyze the nacelle the tower is not included in the analyzes. The reason for doing this is to monitor the change in the response from the nacelle, and since the tower structure is replaced by the design space for the test rig it is the change in nacelle response which is of interest.

To keep the test simple two factors $\alpha_1, \alpha_2 \geq 1$ are introduced. The factors are used to scale the element size for the solid (α_1) and beam (α_2) elements. In Table 5.4 the results while varying α_1 and α_2 are seen.

Table 5.4: Eigenfrequencies in the nacelle when altering α_1 and α_2 .

α_1	1	1	1	1.5	2	2.5	3
α_2	1	1.5	3	1.5	1.5	1.5	1.5
Time [s]	37	37	38	15	8	3	5
Solid	24,837	24,837	24,837	8257	2587	1806	1240
Beam	77	72	67	72	72	72	72
Contact	3052	3052	3052	1524	764	568	360
Target	92	92	92	42	28	24	24
ω_1 [Hz]	1.751	1.751	1.751	1.751	1.751	1.751	1.751
ω_2 [Hz]	5.835	5.835	5.835	5.841	5.853	5.863	5.884
ω_3 [Hz]	10.053	10.053	10.053	10.097	10.097	10.100	10.145
ω_4 [Hz]	10.533	10.533	10.533	10.552	10.552	10.572	10.579
ω_5 [Hz]	11.186	11.186	11.186	11.184	11.184	11.188	11.188
ω_6 [Hz]	11.889	11.889	11.890	11.909	11.909	12.080	12.057
ω_7 [Hz]	17.867	17.867	17.867	17.867	17.867	17.867	17.867
ω_8 [Hz]	17.867	17.867	17.867	17.867	17.867	17.867	17.867
ω_9 [Hz]	17.868	17.868	17.868	17.868	17.868	17.868	17.868
ω_{10} [Hz]	17.869	17.870	17.870	17.870	17.870	17.870	17.870

From Table 5.4 it is seen that it is possible to remove many elements from the model while keeping a good precision of the frequencies. Furthermore this approach reduces the computational time by a factor of 12. It is seen, that the computational time rises for the coarsest models. This might be due to the time is calculated per second, thus small differences in computational time can be shown as a “large” difference.

A coarser mesh is tested with the tower to verify that the change in response is negligible. Defining $\alpha_1 = 2.5$ and $\alpha_2 = 1.5$ the frequencies of the model together with the relative difference is given in Table 5.5. These values are chosen since a short computational time is obtained, which is important for an iterative process. It is seen, that the difference between the frequencies in the model are negligible, thus the coarse model can be used in the design of the test rig.

Table 5.5: Eigenfrequencies for original and new ($\alpha_1 = 2.5$ and $\alpha_2 = 1.5$) wind turbine model and the differences between the two.

	Original Frequency [Hz]	New frequency [Hz]	Difference ‰
ω_1	0.174	0.175	5.74
ω_2	0.263	0.264	3.79
ω_3	1.769	1.774	2.74
ω_4	1.897	1.899	2.83
ω_5	2.630	2.639	3.42
ω_6	3.696	3.704	2.16
ω_7	5.895	5.899	0.68
ω_8	6.743	6.769	3.86
ω_9	7.612	7.615	0.39
ω_{10}	9.605	9.653	5.00

5.2 Design of Test Rig

After the wind turbine model is analyzed the tower is replaced by the design domain for the test rig, as depicted on Figure 5.7. As for the tower model the artificial mass is used to model the interface between design domain and nacelle. Since the beam elements from the tower are replaced by solid elements a new rigid connection is established. The nodes included in the rigid connection are located within one element side length of the 2500 [mm] bed frame hole. An illustration of this is presented on Figure 5.8. In order to ensure a good connection, the top plan of elements is prescribed with a density of one.

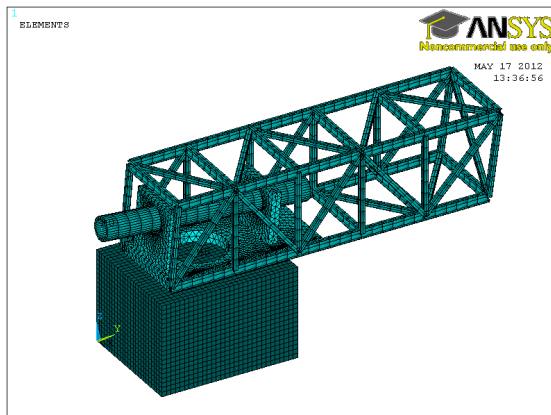


Figure 5.7: The nacelle on top of the maximum allowable design domain. The size of the design domain is $4000 \times 5000 \times 6500$ [mm]. It contain 130,000 elements with a size of 100 [mm] ($40 \times 50 \times 65$ elements).

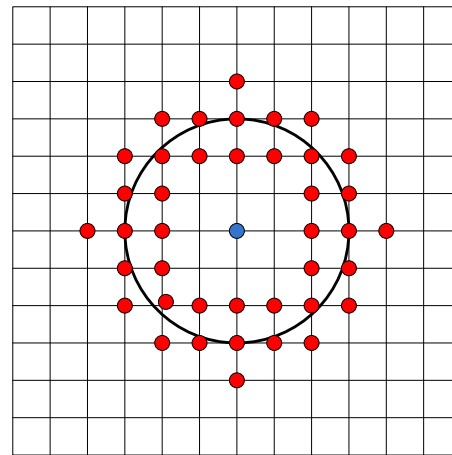


Figure 5.8: Illustration of the coupling between the nacelle and design domain. Red dots are the nodes on the design domain which are coupled to the artificial mass (blue dot).

Some limitations to the model need to be taken into account. Since the top wall thickness of the tower is 16 [mm] it is advantageous to have an element size smaller than this. This provides the ability of modeling the smallest geometrical size in the tower. This however will introduce too many elements in the model e.g. choosing an element side length of 16 [mm] result in 31.7 million elements for the full design domain ($4000 \times 5000 \times 6500$ [mm]), thus violating the license restrictions. Furthermore since a desktop computer is used for the optimization including 31.7 million elements render a too long iteration time, therefore some compromises have to be made.

Increasing the element size to 100 [mm] reduces the number of elements to 130,000, which is illustrated on Figure 5.7. Together with the elements from the nacelle, this yield an iteration time around 5 minutes. From the 3D example in section 4.14 it can be expected that a few hundred iterations are needed to obtain a clear 0-1 design, resulting in a solution time of about 15 hours. This iteration time is considered being too long. On basis of the top diameter of the tower (3000 [mm]) the design is reduced to $4000 \times 3000 \times 3000$ [mm] with an element side length of 100 [mm] this result in 36,000 elements. The smaller design domain should not reduce the possibility of obtaining a test rig design with low eigenfrequencies, since the large design domain mainly provides the opportunity of obtaining a large bending stiffness, which lead to high eigenfrequencies. Furthermore the iteration time is reduced to approximately one minute.

5.2.1 Targeting the Specific Frequencies

The specified eigenfrequencies desired are low compared to the ones from the design domain of $4000 \times 3000 \times 3000$ [mm]. E.g. using RAMP stiffness ($q=8$) and Olhoff-Du(c) mass interpolations with an evenly distributed element density of 30% as start guess. The eigenfrequency for the first tower mode is 7 [Hz], which is 40 times the lowest specified eigenfrequency (0.174 [Hz]). Combined with a large element size (a coarse topology) and a filter radius of 1.5 element length, the possibility of obtaining a coherent topology which fulfill the requirements is very poor.

Since the first desired tower mode has a frequency of 0.174 [Hz] the volume fraction required for a solution can be expected being very low. This causes the low density elements to have a significant influence on the result since a stiffness of zero is not possible. In this case the SIMP stiffness interpolation should have an advantage over the RAMP interpolation, due to the higher penalty for low density elements, as described in section 4.7.

When the design domain of $4000 \times 3000 \times 3000$ [mm] is prescribed with an element density of 0.001, using the RAMP stiffness interpolation ($q=8$) and the Olhoff-Du(c) mass interpolation, the lowest eigenfrequency obtained is 1 [Hz]. The measure of discreteness is close to 0%, thus well below the normally used discreteness target of 5%. This reintroduces the problem of gray elements in the final topology described in section 4.11. In section 4.12 the problem is solved by introducing the measure of discreteness constraint, but when the constraint is satisfied the algorithm is still able to manipulate with the almost void elements in order to satisfy the specified frequencies. Therefore the topology obtained only contains low density elements, naturally this solution is futile.

For the SIMP stiffness interpolation the low density elements have a much lower stiffness, compared to the RAMP stiffness interpolation, which in theory should benefit the SIMP interpolation. With SIMP stiffness interpolation ($p=3$), Olhoff-Du(a) mass interpolation and an element density of 0.001 the measure of discreteness is 0%, and the lowest eigenfrequency is 0.003 [Hz]. However although SIMP stiffness interpolation is able to obtain the low frequencies it also produces a topology where all elements are close to void, since an element density of 0.09 is enough to obtain a fundamental eigenfrequency of 0.174 [Hz].

As discussed above, the void elements contribute with too much stiffness to the structure, mostly because of the low volume fraction which results in these elements being in absolute majority. This problem can be reduced by confining the design domain to a smaller volume.

One option is to remove the elements at the center of the design domain. This however is not an option for the program developed in this project due to the filters used. The filters

implemented can only handle rectangular continuous design domains. Therefore the only option is to further reduce the rectangular volume which has to be at least 2500 [mm] plus the element size (100 [mm]), due to the connection with the nacelle. The model obtained with this additional reduction in the design domain is illustrated on Figure 5.9. It is not possible to reduce the design domain further in the program, and therefore it is not expected that it is possible to obtain a coherent topology with a eigenfrequency as low as 0.174 [Hz].

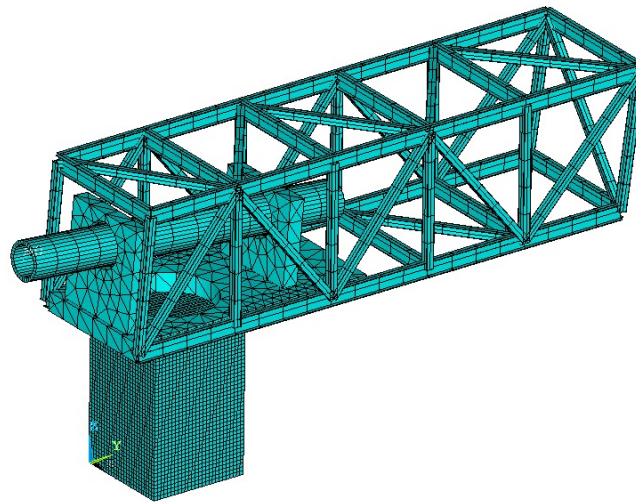


Figure 5.9: The nacelle on top of the design domain. The size of the design domain is $4000 \times 2600 \times 2600$ [mm] and it contains 27,040 elements with a size of 100 [mm] ($40 \times 26 \times 26$ elements).

5.2.2 Treating Nacelle Modes

As seen in section 5.1 some eigenpairs are independent or nearly independent of the structure below (tower or test rig). These are modes governed by the nacelle. Still some displacements in the design domain are present, thus the sensitivities for these modes do exist although the difference in frequency is nearly unchanged by the topology of the design domain.

To overcome these modes in the nacelle three different approaches are proposed. (1) The nacelle mode is completely ignored during the optimization, because the reference eigenpairs are unaffected by these frequencies. If a nacelle mode has a frequency below the safety zone the descent function for MMA will have a higher value due to the violation of the safety zone constraint otherwise no effect of the nacelle mode is present. (2) The nacelle mode can also be targeted, as one of the specified frequencies, and since the nacelle frequency is nearly constant through the optimization this term is almost zero. (3) It is also possible to use a negative safety zone, but this can introduce problems where higher order modes obtain a frequency below, between or close to the specified frequencies, which might affect the response of the structure. This method is not tested.

5.2.3 Standard Configurations

The program configurations used in the attempt to solve this problem are listed in Table 5.6. The number of modes extracted are increased from 4 to the 12 lowest eigenpairs, since a 3D structure exhibit more mode shapes than a 2D structure. The SIMP interpolation is used due to the higher penalty for low element densities and the start guess is evenly distributed material of 30% in all non prescribed elements in the design domain. Keep in mind the prescribed elements at the top of the design domain, which ensures a connection between the design domain and the nacelle.

Table 5.6: Standard configurations for the design of the test rig.

Resource Constraint	30 %
Elements	27040 SOLID45
Element size	100 [mm]
Algorithm	MMA Bound Target Discrete
c constants	15 for frequencies 1.1 for Resource
N	12
Stiffness Interpolation	SIMP, $p = 3$
Mass Interpolation	Olhoff-Du(a)
Filter	Sensitivity $r=1.5$
Max Iterations	1000
Move limit	0.08
Discreteness Target	5%
Convergence	$\max(\Delta x) \leq 0.01$
Mass Matrix	Lumped
MAC	ON
E_0	210,000 [MPa]
ρ	7800 [kg/m ³]

5.2.4 Lowest Possible Target Frequencies

In this section the lowest possible target frequencies using steel as material for the test rig is presented.

When using the reduced design domain ($4000 \times 2600 \times 2600$ [mm]) and the SIMP stiffness interpolation the lowest target frequencies obtained, while the topology is coherent, are 5.5 and 6.5 [Hz] with a safety zone of 0.5 [Hz]. In this case a local nacelle mode is present at approximately 2.93 [Hz]. This frequency is targeted together with the two desired frequencies, and does not affect the solution. The modes shapes are associated in the third iteration where all three mode shapes are selected.

The final topology is illustrated on Figure 5.10 and the corresponding results are given in Table 5.7. It is probably possible to improve this result by applying an advanced start guess, but it is unlikely that a perfect start guess will be able to achieve a frequency of 0.174 [Hz].

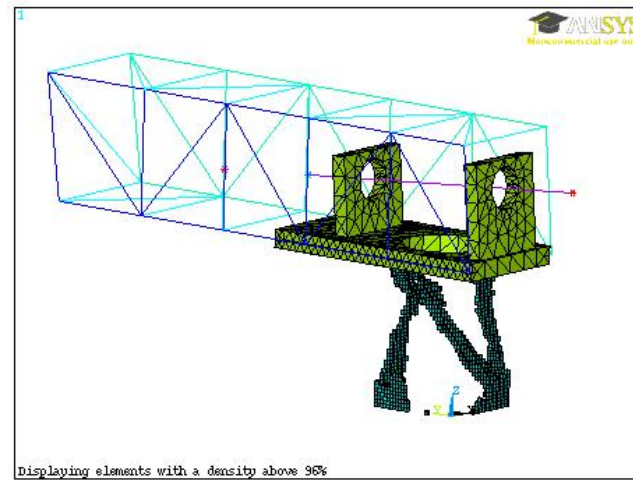


Figure 5.10: The topology obtained with $\omega_1=2.93$, $\omega_2=5.50$ and $\omega_3 = 6.48$ [Hz]. The volume fraction is 7.9% in the final topology.

Table 5.7: The result for the lowest possible target frequencies for the test rig design. The results are obtained using the configurations in Table 5.6.

$\omega_1^{(t)}$ [Hz]	ω_1 [Hz]	Iter	Topology
$\omega_2^{(t)}$ [Hz]	ω_2 [Hz]		
$\omega_3^{(t)}$ [Hz]	ω_3 [Hz]	MoD [%]	
$MoD^{(t)}$ [%]	ω_4 [Hz]		
$\omega_1^{(t)} = 2.93$	$\omega_1=2.93$	244	Figure 5.10
$\omega_2^{(t)} = 5.5$	$\omega_2=5.50$		
$\omega_3^{(t)} = 6.5$	$\omega_3=6.48$		
$MoD^{(t)} = 5$	$\omega_4=7.43$		

From this example it is seen, that the code is able to optimize for specific eigenfrequencies with mode tracking when the nacelle is attached to the design domain. Using steel as material with the current discretization it is seen, that achieving the reference frequencies of 0.174 [Hz] is not possible. However a finer discretization and a smaller design domain might solve the problem since it provides the possibility of representing a more detailed design. To show that it is possible to obtain the low eigenfrequencies with the program, the material model is altered in appendix C and topologies with a $\omega_1=0.174$ [Hz] is obtained. Furthermore a discussion and recommendations for further work is presented in section 6.3.

CHAPTER 6

Conclusion

Through this Master's thesis a method for obtaining topologies with prespecified eigenmodes and -frequencies is presented. The method is tested on a real life problem, i.e. designing a test rig for a wind turbine nacelle. The real life problem is provided by the Danish department of the consultancy company FS Dynamic AB. The goal is to reproduce the response of a wind turbine tower model in a test rig where the maximum height is severely restricted. The test rig will provide easier access to the nacelle under testing, thereby making it easier, faster and cheaper to test new nacelle designs.

Ideally it is desired to obtain a test rig design where both frequencies, mode shapes and the strain state are the same for the nacelle, as in the real wind turbine. The two lowest frequencies and the associated mode shapes are of most importance thus are used as reference frequencies and modes. Throughout the project period the focus has been on developing a code which is capable of matching a desired set of specified eigenpairs using topology optimization.

Methods to obtain a given strain state in the nacelle are not investigated in this Master's thesis. Some thoughts on the subject are presented in section 6.3.1.

6.1 The Developed Code

A MATLAB program is developed to investigate the behavior of the eigenfrequency optimization for distinct (simple) eigenvalues. The program is able to solve 2D problems as verification, including problems found in Du and Olhoff [2007]. The sensitivity analysis is only capable of addressing distinct eigenvalues, since the desired frequencies in the test rig all are distinct the necessity of including multiple eigenvalues is not present. The program is able to target prespecified frequencies and associate an user defined mode shape to the target frequencies, where examples are presented as proof of concept. It is added the capability to use ANSYS as the solver while performing the optimization inside MATLAB.

The code is extended into a 3D version, where only the ANSYS solver is available. This version of the code is used to solve the test rig design problem.

The program is developed on basis of Sigmunds 99 lines code [Sigmund, 2001] and the 88 lines code [Andreassen et al., 2011]. These codes are extended and rewritten in order to obtain the capability of conducting eigenfrequency optimization using the MMA algorithm

developed by Krister Svanberg [Svanberg, 1987].

6.1.1 Maximizing the Fundamental Eigenfrequency

As the first step, maximization of the fundamental frequency is implemented. Multiple eigenvalues often occur during topology optimization, which is also encountered for the preprogrammed boundary conditions. Since this is not taken into account the solutions obtained are only considered being close to the optimum solution. As the simplest form of eigenfrequency optimization this allows for closer investigation on the effects of move limits, start guess, resource constraint, mass and stiffness interpolation, choice of elements and filtering.

Through the analyzes it is seen, that the start guess and resource constraint have a large effect on the final topology. However since a local optimum is sufficient for the goal of the project it is important to be aware of the possibility of obtaining multiple topologies dependent upon variables such as start guess and resource constraint.

The choice of elements did not show any effect on the final topology when using mesh independent filtering, whereas the move limits had a large influence on the topology when using the SIMP stiffness interpolation, but not for the RAMP stiffness interpolation.

In order to eliminate local modes (low density areas with a local mode having a lower frequency than the general structure) different mass and stiffness interpolations are investigated. It is observed, that three conditions can provoke local modes. This differ from what is found in the literature, where the focus is on the local inverse Rayleigh's coefficient (Ω).

The following three conditions need to be fulfilled for local modes to occur:

1. A high inverse Rayleigh's coefficient (Ω)
2. Low element stiffness
3. The above two conditions must be fulfilled for all elements surrounding a single node

It is seen, that SIMP stiffness interpolation has problems with local modes if a linear mass interpolation is used. Using one of the mass interpolations in Du and Olhoff [2007] (shown in equation (2.7) to (2.9) on page 10) the problem is eliminated except for a peak in Ω (see Figure 4.46 on page 51) which cause local modes in a few iterations obstructing the optimization, while having little effects on the final topology.

If the RAMP stiffness interpolation is used local modes are in most cases eliminated, but a low density (e.g. 10^{-8}) can result in local modes when using linear mass interpolation. This problem is addressed using the mass interpolations in Du and Olhoff [2007]. Using RAMP

stiffness interpolation with the mass interpolation in equation (2.9) removes the problem of local modes completely, since Ω approaches zero as the density is lowered.

6.1.2 Targeting Specified Frequencies

In order to obtain topologies with prespecified eigenfrequencies the objective function is formulated such the deviation between a calculated and target frequency is minimized. A safety zone is introduced to ensure that eigenpairs which are not prespecified are forced above the target frequencies, thus avoiding these eigenpairs to interfere with the response of the structure.

When targeting frequencies many gray elements might be present especially when specifying low frequencies. Nothing is favoring the algorithm to obtain clear 0-1 topologies, as explained in section 4.12. It is proposed to remove these with a constraint on the measure of discreteness. The constraint is gradually enabled during the optimization such the algorithm can slowly alter the topology while still satisfying the specified frequencies. Specifying a maximum value of 10% or less for the measure of discreteness provided sufficient 0-1 designs to gain a clear topology.

6.1.3 Associating Modes

The ability to track modes using Modal Assurance Criterion (MAC) is incorporated into the program. When matching eigenmodes with the specified frequencies a set of reference modes can be chosen during the optimization, and new reference modes can be reselected at any time during the optimization.

The reference modes and the current modes are compared and the eigenmodes with the highest MAC values are matched to the corresponding target frequencies. It is shown in section 4.13 that it is possible to switch two modes while optimizing for two specified frequencies.

6.1.4 3D Code

The program is extended into a 3D version. This version is verified in section 4.14 to ensure the program is functioning properly. It is shown that the 3D version is able to obtain topologies with a predefined response, where specified modes vibrate with the target frequencies showing that the algorithm can solve these types of problems.

6.2 Design of Test Rig

To test the capabilities of the code a real life problem is provided by FS Dynamic AB. The problem is to design a test rig for a provided wind turbine model. The wind turbine model is initially analyzed, and the number of elements in the model is reduced to improve the iteration time during optimization. It is possible to reduce the number of elements by a factor of 11 while keeping the maximum change in frequency below 1%.

Replacing the tower by the design domain for the test rig it is shown that the program can handle advanced models as long as the design domain is a rectangular volume.

Local modes are present in the nacelle. These modes are independent or nearly independent of the structure below. In section 5.2 different methods are proposed to eliminate the significance of these in the optimization.

It is possible to obtain a topology with a predefined response where specified modes vibrate with the target frequencies as shown for both the 2D and the simple 3D problem. However obtaining topologies satisfying the very low frequencies from the original wind turbine model was not possible. Among other things due to the low volume fraction required to obtain the low frequencies.

Although it was not possible to obtain a topology satisfying the low frequencies desired for the test rig, the program developed still have the ability to identify a specific mode shape and obtain topologies with specific frequencies for the identified mode shapes.

6.3 Further Work

In this section some thoughts on the project in general are presented and discussed. Some proposals for further work on omitted issues in this Master's thesis are presented.

Although the program is able to obtain a topology with one or more specific eigenpairs there is no guarantee that a solution obtained actually can support the heavy nacelle or any other load case. It is therefore suggested to introduce one or more static load cases as a compliance constraint which will ensure that the topology has a minimum stiffness for the given load cases. This is a simple extension, where one or more constraints are added to the MMA formulation. This requires additional static load cases in the analyzes, but it is a "cheap" analysis compared to the eigenvalue problem and should not provide an unreasonable large increase in computational time.

Even though a compliance constraint is included in the formulation, this is no guarantee

that the structure can withstand the life cycle of the test rig. Structural failure criteria such as fatigue, buckling and yielding have to be addressed before production. This requires a interpretation of the design in order to conduct a detailed stress analysis which is necessary before the test rig is ready to contribute to the development of the new generation of nacelles.

6.3.1 Reproducing a Strain State

Ideally it is desired to obtain the same strain state in the nacelle when using the test rig instead of the wind turbine tower. From FS Dynamics no detailed definition of the strain state is presented and this subject is not investigated in this Master's thesis, but in the following methods of including this are given.

Topology optimization is inherently a large scale problem, thus it is recommended to use global measures for the objectives [Bendsøe and Sigmund, 2003]. Since the strain state in the nacelle is a local phenomena it is inefficient to include this in the formulation. One method to avoid this is to rewrite the strain state in the nacelle as a global measure and including this into the topology optimization. This is not a trivial task, and possible methods for conducting this has not been sought for.

Another method is to conduct a topology optimization followed by e.g. a shape optimization or a parameter study. Here the optimal topology is found which satisfies the global criteria (the specified eigenfrequencies and modes). On basis of the topology optimization a parametrization is conducted (e.g. the boundaries of the topology or interpretation the design as a beam structure.) allowing the finer details of the test rig to be modified by a moderate number of design variables which is better suited for studying local problems [Bendsøe and Sigmund, 2003].

Bibliography

- Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B., and Sigmund, O. (2011). Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43:1–16.
- ANSYS Inc. (2010). *ANSYS Release 13.0 Documentation*.
- Arora, J. S. (1999). *Introduction to Optimum Design*. Elsevier, second edition. ISBN: 0-12-064155-0.
- Bendsøe, M. P., Olhoff, N., and Taylor, J. E. (1983). A variational formulation for multicriteria structural optimization*. *Journal of Structural Mechanics*, 11(4):523–544.
- Bendsøe, M. P. and Sigmund, O. (2003). *Topology Optimization: Theory, Methods and Applications*. Springer, second corrected printing edition. ISBN: 3-540-42992-1.
- Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J. (2002). *Concepts and Applications of Finite Element Analysis*. Wiley, fourth edition. ISBN 978-0-471-35605-9.
- Du, J. and Olhoff, N. (2007). Topological design of freely vibrating continuum structures for maximum values of simple and multiple eigenfrequencies and frequency gaps. *Structural and Multidisciplinary Optimization*, 34:91–110.
- Ewins, D. (2000). *Modal testing: theory, practice, and application*. Mechanical engineering research studies: Engineering dynamics series. Research Studies Press, second edition. ISBN: 0-86380-218-4.
- FS Dynamics (2012). Fs dynamics presentation aau.
- Hansen, L. (2005). Topology optimization of free vibrations of fiber laser packages. *Structural and Multidisciplinary Optimization*, 29:341–348. 10.1007/s00158-004-0495-8.
- Kim, T. S. and Kim, Y. Y. (2000). Mac-based mode-tracking in structural topology optimization. *Computers & Structures*, 74(3):375 – 383.
- Lund, E. (1994). *Finite Element Based Design Sensitivity Analysis and Optimization*. PhD thesis, Institute of Mechanical Engineering, Aalborg University.
- Olhoff, N. (1989). Multicriterion structural optimization via bound formulation and mathematical programming. *Structural and Multidisciplinary Optimization*, 1:11–17. 10.1007/BF01743805.

- Rao, S. S. (2004). *Mechanical Vibrations*. Pearson, fourth edition. ISBN: 0-13-120768-7.
- Rietz, A. (2001). Sufficiency of a finite exponent in simp (power law) methods. *Structural and Multidisciplinary Optimization*, 21:159–163. 10.1007/s001580050180.
- Seyranian, A. P., Lund, E., and Olhoff, N. (1994). Multiple eigenvalues in structural optimization problems. *Structural and Multidisciplinary Optimization*, 8:207–227. 10.1007/BF01742705.
- Sigmund, O. (2001). A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21:120–127.
- Sigmund, O. (2007). Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33:401–424. 10.1007/s00158-006-0087-x.
- Stolpe, M. and Svanberg, K. (2001). An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22:116–124. 10.1007/s001580100129.
- Svanberg, K. (1987). The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373.
- Svanberg, K. (2007). Mma and gmma, versions september 2007. This note describes the algorithms used in the author's latest implementations of MMA and GCMMA.
- Zienkiewicz, O. C. and Taylor, R. L. (2005). *THE FINITE ELEMENT METHOD - ITS BASIS & FUNDAMENTALS*. Elsevier, sixth edition.

APPENDIX A

The Method of Moving Asymptotes

This appendix describes the formulations used in the Method of Moving Asymptotes (MMA) algorithm to update the design variables. The solver is used as a black box call where the files made by Krister Svanberg are directly used. The formulation of the algorithm is based on Svanberg [2007].

The MMA solver is a non-linear programming algorithm primarily used for structural optimization. It is an iterative process which solves a set of subproblems all strictly convex [Svanberg, 1987].

MMA is a trust region method where the asymptotes controls the trust region. Here the approximation is assumed accurate within the trust region, as opposed to line search algorithms (e.g. steepest descent [Arora, 1999]) where the minimum in a given search direction is found. In the following all is based on Svanberg [2007]. The MMA problem is defined as in equation (A.1).

$$\text{minimize } f_0^{(k)}(x_i) + a_0 z + \sum_j \left(c_j y_j + \frac{1}{2} d_j y_j^2 \right), \quad i = 1, \dots, NE \quad (\text{A.1a})$$

Subject to:

$$f_j(x_i) - a_j z - y_j \leq 0, \quad j = 1, \dots, J \quad (\text{A.1b})$$

$$0 < x_{min} \leq x_i \leq 1, \quad y_j \geq 0, \quad z \geq 0 \quad (\text{A.1c})$$

Here x_i are the NE design variables. f_0 and f_j are given, continuously differential functions, and J are the number of constraint functions f_j . z and y_j are artificial variables ≥ 0 which helps the convergence of a subproblem. a_0, a_j, c_j and d_j are given real numbers satisfying: $a_0 > 0, a_j \geq 0, c_j \geq 0, d_j \geq 0, c_j + d_j > 0$ and $a_j c_j > a_0$ for all j with $a_j > 0$.

MMA has an equivalent to the bound formulation in equation (2.29). By choosing $f_0 = 0$, $a_0, a_1, \dots, a_J = 1$, $a_{J+1}, \dots, a_K = 0$, $d_1, \dots, d_K = 1$ and the c constants “a large number” e.g. 1000 the MMA equivalent bound formulation is obtained.

$$\text{minimize } z + \sum_l \left(c_l y_l + \frac{1}{2} y_l^2 \right), \quad l = 1, \dots, K \quad (\text{A.2a})$$

Subject to:

$$f_j(x_i) - z - y_j \leq 0, \quad i = 1, \dots, NE \quad j = 1, \dots, J \quad (\text{A.2b})$$

$$f_k(x_i) - y_k \leq 0, \quad k = J + 1, \dots, K \quad (\text{A.2c})$$

$$0 < x_{min} \leq x_i \leq 1, \quad y_l \geq 0, \quad z \geq 0 \quad (\text{A.2d})$$

This formulation replaces the formulation in equation (2.29) as the bound formulation. K is the number of constraints which is either $K = N + 2$ or $K = N + 1$ depending on whether or not the measure of discreteness is included as a constraint and N is the number of eigenpairs extracted.

As described in chapter 4 two formulations are available, i.e. maximizing the fundamental these are presented in the following. Here the measure of discreteness is included as a constraint.

When maximizing fundamental frequency the bound formulation is given as:

$$\text{minimize } z + \sum_l \left(c_l y_l + \frac{1}{2} y_l^2 \right), \quad l = 1, \dots, K \quad (\text{A.3a})$$

Subject to:

$$f_j(x_i) - z - y_j = C - \lambda_j - z - y_j \leq 0, \quad j = 1, \dots, N \quad (\text{A.3b})$$

$$f_{N+1}(x_i) - y_{N+1} = MoD - MoD^{(t)} - y_{N+1} \leq 0 \quad (\text{A.3c})$$

$$f_K(x_i) - y_K = \sum_i \frac{x_i}{v_i} - V - y_K \leq 0, \quad i = 1, \dots, NE \quad (\text{A.3d})$$

$$0 < x_{min} \leq x_i \leq 1, \quad y_l \geq 0, \quad z \geq 0 \quad (\text{A.3e})$$

Here the first N f functions are replaced by the bound formulation for the eigenvalues (equation (2.30)). C is a constant making the f functions positive. Equation (A.2c) is replaced by the discreteness (equation (4.4)) and resource constraint.

When targeting eigenvalues the formulation is changed as described in section 4.11. Im-

plemented in the MMA algorithm it has the following form:

$$\text{minimize } z + \sum_l \left(c_l y_l + \frac{1}{2} y_l^2 \right), \quad l = 1, \dots, K \quad (\text{A.4a})$$

Subject to:

$$f_j(x_i) - z - y_j = \left(\lambda_j - \lambda_j^{(t)} \right)^2 - z - y_j \leq 0, \quad j = 1, \dots, N_T \quad (\text{A.4b})$$

$$f_k(x_i) - z - y_k = \lambda_{safety} - \lambda_k - z - y_k \leq 0, \quad k = N_T + 1, \dots, N \quad (\text{A.4c})$$

$$f_{N+1}(x_i) - y_{N+1} = MoD - MoD^{(t)} - y_{N+1} \leq 0 \quad (\text{A.4d})$$

$$f_K(x_i) - y_K = \sum_i \frac{x_i}{v_i} - V - y_K \leq 0, \quad i = 1, \dots, NE \quad (\text{A.4e})$$

$$0 < x_{min} \leq x_i \leq 1, \quad y_l \geq 0, \quad z \geq 0 \quad (\text{A.4f})$$

Here equation (A.4b) are the N_T equations for targeting the specified set of eigenvalues $\lambda^{(t)}$. Equation (A.4d) contain the “safety zone” constraint, and as in equation (A.3) the last two equations are the discreteness and resource constraints.

For both formulations the equations are normalized in order to give an initial order of magnitude for the equations in the area around 1.

APPENDIX B

Advanced Start Guesses

In this appendix “advanced” start guesses are investigated, in order to see the effect of varying start guesses. Here “advanced” refer to guesses where the available material is not evenly distributed.












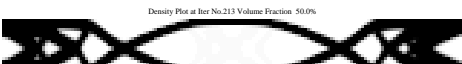
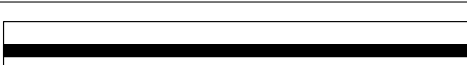
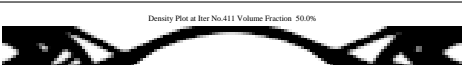
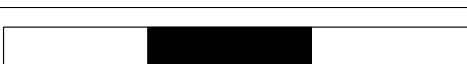
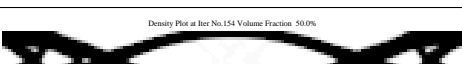
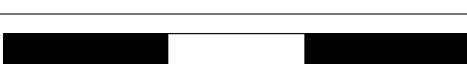
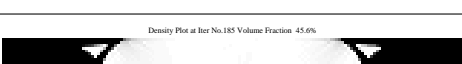
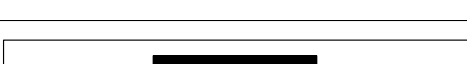


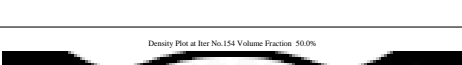
Different start guesses are shown in Table B.1 all with the standard settings from Table 4.1 on page 31. These guesses are chosen in order to gain a broad field of start guesses, thus a set of final topologies are obtained which represent different local minima.

From Table B.1 it is seen, that all topologies are single symmetric, and some are doubly symmetric. The only exception is number three, where the areas around the supports differ.

In general two topologies are present i.e. a double arc (e.g. number 1) and a single arc (e.g. number 3). Guessing a large non coherent area results in the trivial solution (number 9). Otherwise it is seen, that the double symmetric topology is present when the initial guess is double symmetric.

The double arc topologies are seen to have a lower fundamental frequency than the single arc topologies. The reason for this is unknown and is irrelevant for the project, since it is not the purpose to maximize the fundamental frequency, but the eigenfrequencies are still lower than the ones obtained in Du and Olhoff [2007]. The number of iterations before convergence are higher than the evenly distributed start guess (130 iterations) except for number 4 which is in the same area of iterations.

Table B.1: Start Guesses, all results are obtained using the standard configuration.

No	Start Guess	ω_1 [rad/s] Iter MoD [%]	Topology
1		405.00 217 6.2	Density Plot at Iter No. 99 Volume Fraction 50.0% 
2		385.52 199 12.3	Density Plot at Iter No. 199 Volume Fraction 50.0% 
3		449.13 175 8.8	Density Plot at Iter No. 175 Volume Fraction 50.0% 
4	Density Plot at Iter No. 1 Volume Fraction 49.5% 	396.58 118 12.0	Density Plot at Iter No. 118 Volume Fraction 50.0% 
5		267.06 17 60.1	Density Plot at Iter No. 17 Volume Fraction 49.9% 
6		406.10 213 12.5	Density Plot at Iter No. 213 Volume Fraction 50.0% 
7		449.21 411 8.5	Density Plot at Iter No. 411 Volume Fraction 50.0% 
8		375.87 154 12.6	Density Plot at Iter No. 154 Volume Fraction 50.0% 
9		543.13 185 6.5	Density Plot at Iter No. 185 Volume Fraction 45.0% 
10		387.76 170 13.5	Density Plot at Iter No. 170 Volume Fraction 50.0% 
11		397.82 154 11.7	Density Plot at Iter No. 154 Volume Fraction 50.0% 

APPENDIX C

Artificial Material

Through section 5.2.4 it is seen, that the stiffness of the test rig design domain and the large mass of the nacelle governs the frequencies obtainable. The mass of the nacelle is unchanging and much larger than the mass of the test rig, the nacelle weighs 286 [tonne] and the test rig on Figure 5.10 on page 88 only weighs 16.7 [tonne]. Therefore the mass of the test rig only has a small influence on the responds of the structure.

A material with a lower stiffness will result in a lower influence from the void elements. This can result in a clear 0-1 design for lower frequencies. It might be possible to obtain a topology using artificial material properties and transform this topology into a steel structure, which can be optimized using a parameter study.

This approach is attempted by using material properties one thousandth the material properties of steel (keeping the Poisson's ratio at 0.3, $E_0 = 210[MPa]$ and $\rho = 7.8[kg/m^3]$). The settings used are seen in Table 5.6 on page 87.

The start guess is found to have a large influence on the ability to obtain a coherent design and the results are highly related to the start guess. Using an evenly distributed density as start guess results in all elements in the design domain ends as nearly void elements. Therefore a sensible coherent start guess is preferable. Here two designs are shown obtained with the two design guess in Figure C.1 and C.2.

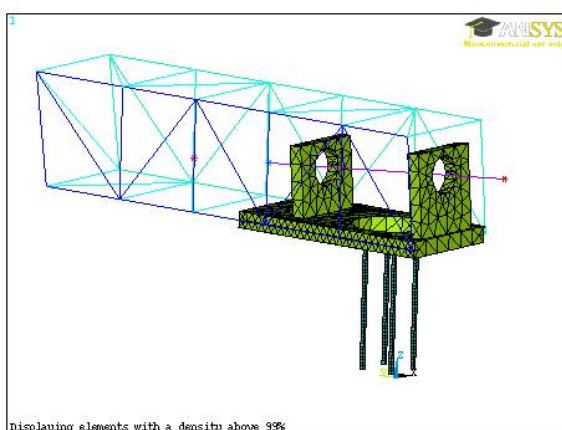


Figure C.1: A start guess consisting of four beams supporting the nacelle.

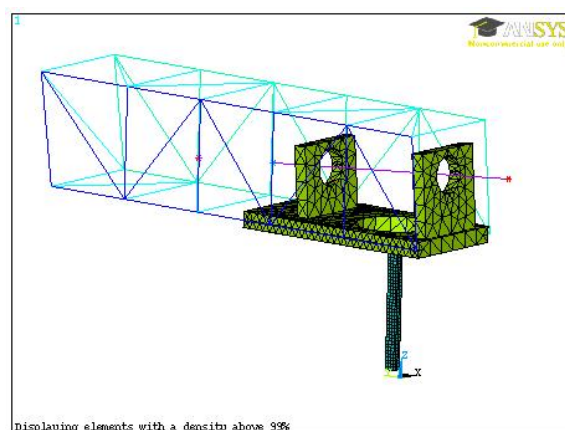


Figure C.2: A start guess consisting of a single beam supporting the nacelle.

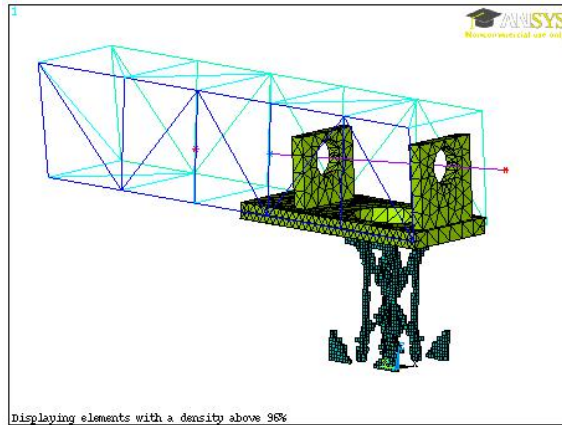


Figure C.3: The topology obtained with Figure C.1 as start guess. The desired bending modes are reached as the two lowest modes at the desired frequencies as listed in Table C.1.

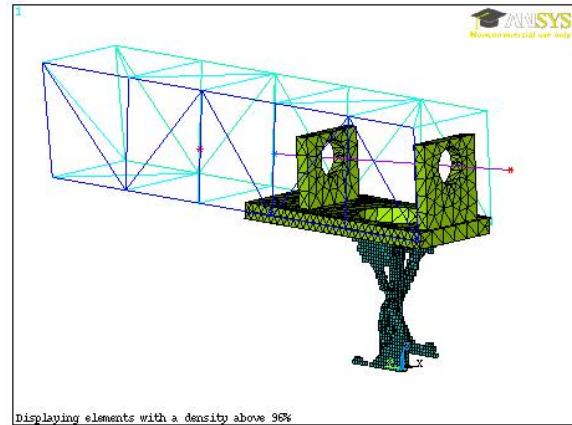


Figure C.4: The topology obtained with Figure C.2 as start guess. The desired bending modes are reached as the two lowest modes at the desired frequencies as listed in Table C.1

Table C.1: The result obtained for the test rig design. The results are obtained using the configurations in Table 5.6 on page 87 together with the artificial material properties.

$\omega_1^{(t)}$ [Hz]	ω_1 [Hz]	Iter	Topology
$\omega_2^{(t)}$ [Hz]	ω_2 [Hz]		
ω_{xz} [Hz]	ω_3 [Hz]	MoD [%]	
$MoD^{(t)}$ [%]	ω_4 [Hz]	Reccource [%]	
$\omega_1^{(t)} = 0.174$	$\omega_1=0.172$		
$\omega_2^{(t)} = 0.263$	$\omega_2=0.261$	213	Figure C.3
$\omega_{xz}=0.2$	$\omega_3=0.679$	4.6	
$MoD^{(t)} = 5$	$\omega_4=3.080$	8.4	
$\omega_1^{(t)} = 0.174$	$\omega_1=0.174$		
$\omega_2^{(t)} = 0.263$	$\omega_2=0.263$	247	Figure C.4
$\omega_{xz}=0.2$	$\omega_3=0.676$	4.6	
$MoD^{(t)} = 5$	$\omega_4=3.080$	9.5	

As illustrated on Figure C.3 and C.4 and the results in Table C.1 the artificial material properties makes it possible to obtain coherent designs where both the desired eigenmodes and frequencies are reached. Showing that the program is able to reach designs with low frequencies.

It should however be noted, that a material with a young's modulus of 210 [MPa] corresponds to low-density polyethylene (used for water bottles) and the total mass of the two topologies are approximately 20 [kg] each. Therefore this is a strictly theoretical example illustrating that the program has no problems with low frequencies. On basis of this it is expected, that with more elements and a good start guess it might be possible to obtain a topology with steel as the material.

Nomenclature

$[\mathbf{K}]$	Global stiffness matrix	N	Number of eigenpairs extracted
$[\mathbf{M}]$	Global mass matrix	N_T	Number of eigenmodes to be tracked
$[\bar{\mathbf{k}}_e^0]$	Element stiffness matrix with unit Youngs modulus	NE	Number of elements in a model
$[\bar{\mathbf{m}}_e^0]$	Element mass matrix with unit mass density	p	Penalty factor for SIMP
$[\mathbf{k}_e]$	Element stiffness matrix	q	Penalty factor for RAMP
$[\mathbf{m}_e]$	Element mass matrix	$T_{e,ij}$	Element kinetic energy for element i due to a velocity field j
$\{\bar{\phi}\}_{ji}^T$	Entries in eigenvector j matching element i	$U_{e,ij}$	Element strain energy for element i due to a displacement field j
$\{\phi_R\}_k$	Reference eigenvector k	$W'(x)$	Derivative of mass density interpolation function
$\{\phi\}_j$	Eigenmode or eigenvector for eigenvalue j	$W(x)$	Mass density interpolation function
\bar{g}_k	Maximum value for constraint function k	w_j	Weighting for objective function j
$E'(x)$	Derivative of Youngs modulus for a density x	x	Element densities (design variables)
$E(x)$	Youngs modulus for a density x	β	Bound variable
E_0	Youngs modulus for solid material	$\lambda_j^{(t)}$	Reference (target) eigenvalue number j
f_j	Objective function j	λ_j	Eigenvalue number j
g_k	Constraint function k	λ_{safety}	Minimum eigenvalue for non-target eigenvalues
MAC	Modal Assurance Criterion value	Ω	Inverse Rayleighs coefficient
MoD	Measure of discreteness	ω_j	Eigenfrequency for eigenvalue j
		ρ	Mass density of the material