Computational Optimization of Structural and Thermal Compliance Using Gradient-Based Methods

Computational Optimization of Structural and Thermal Compliance Using Gradient-Based Methods

By Mark Baczkowski, H.B.Sc.

A Thesis Submitted to the School of Graduate Studies in the Partial Fulfillment of the Requirements for the Degree of Master of Science

McMaster University

Master of Science (2019)

Hamilton, Ontario (Department of Mathematics and Statistics)

TITLE: Computational Optimization of Structural and Thermal Compliance Using Gradient-Based Methods

AUTHOR: Mark Baczkowski (McMaster University)

SUPERVISOR: Dr. Bartosz Protas and Dr. Il Yong Kim

NUMBER OF PAGES: viii, 68

Abstract

We consider the problem of structural optimization which has many important applications in the engineering sciences. The goal is to find an optimal distribution of the material within a certain volume that will minimize the mechanical and/or thermal compliance of the structure. The physical system is governed by the standard models of elasticity and heat transfer expressed in terms of boundary-value problems for elliptic systems of partial differential equations (PDEs). The structural optimization problem is then posed as a suitably constrained PDE optimization problem, which can be solved numerically using a gradient approach. As a main contribution to the thesis, we derive expressions for gradients (sensitivities) of different objective functionals. This is done in both the continuous and discrete setting using the Riesz representation theorem and adjoint analysis. The sensitivities derived in this way are then tested computationally using simple minimization algorithms and some standard two-dimensional test problems.

Acknowledgements

First and foremost, I would like to thank Dr. Bartosz Protas for being my supervisor over the last two years. Over this time he has helped me grow both as person and researcher, and thanks to his patience and guidance with me through this process, has allowed this to be possible. I would like to thank him and Dr. Il Yong Kim for offering me the opportunity to work under them for this project and for all of their help along the way. I would also like to thank Stephen Roper and the rest of the SMSD group for their help in explaining engineering concepts to me over the course of my research. I would also like to thank the members of my examination committee, Dr. Gail Wolkowicz and Dr. Traian Pirvu, for all of their input and recommendations about my thesis. In addition I would like to thank my friends, my Mom, my Dad, and my sister Ola for their continuous support and encouragement. Last, but definitely not least, I would like to thank Nicole Palumbo for always being there for me and for reminding me how to dream big.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Whenever a force is applied to an object, it causes the particles inside of the object to move closer together, but due to the Pauli exclusion principle, the particles cannot occupy the same place at the same time and therefore will repel each other. On the other hand, if the force is in the opposite direction, the particles will be pulled apart, but due to the lower energy state that they are in, this will cause an attraction force between the particles in order to keep the object together (Griffiths 2005). These internal forces in the object are called stresses and the stress tensor is typically denoted with the symbol $\boldsymbol{\sigma}$. If the external forces acting on the object are greater than the internal forces, then the object will start to deform. This deformation is called strain, and the strain tensor is often denoted with the symbol $\boldsymbol{\epsilon}$ or $\boldsymbol{\grave{o}}$ (Gere 2004). When one states that an object is stiff, they are referring to the low amount of strain in the object when acted upon by a mechanical load. In terms of stress and strain, it is most common to calculate the opposite of stiffness, namely compliance or flexibility. Since there is always a slight deformation of the object whenever a force is applied to it, compliance has become an important quantity used by engineers in the design of different structures or vehicles.

Due to the rising concerns from costumers of the environmental impact and of increasing fuel costs, vehicle manufactures have placed an increasing amount of resources into

developing fuel efficient cars in recent years. This has pushed engineers to optimize the performance of the vehicle while also minimizing its weight. This could be a reason why there has been an increasing interest in topology optimization in recent years, as it has allowed engineers to minimize the compliance of the part they are working on, while also keeping the part under some maximum weight requirement. Topology optimization is the optimization of an objective functional, specifically in this thesis the objective functional will be compliance, where the design variable is the density of some material. The design variable is typically denoted by ρ and is constrained to be between 0 and 1 in order to make physical sense. Therefore, we give the physical meaning to the values of our design variable by stating that $\rho(\boldsymbol{x}) = 0$ represents a hole in the object while, $\rho(\boldsymbol{x}) = 1$ is meant to represent the placement of the material at location $\boldsymbol{x}$. This is why topology optimization gets its name from the study of topology, as the number of holes in an object are typically used to denote the class of the object, and in topology optimization we are looking for the optimal amount and placement of holes in an object. Topology optimization differs from shape optimization in that there does not have to be a starting shape configuration that will be changed, but can be a uniform distribution of material over the entire space, which leads to more freedom for the final configuration than in shape optimization. In this thesis, we will be using topology optimization to adjust the density distribution in order to minimize compliance. The mathematical description of compliance means that a system of partial differential equations, (PDEs), must be solved in order to determine its value. This leads us to consider two types of approaches in order to optimize systems modelled by PDEs, namely, we have a choice whether we first discretize the partial differential equation and then optimize, or optimize and then discretize.

Both of the optimization approaches have advantages and disadvantages associated with them. The advantage with the "optimize-then-discretize" approach is that gradient is independent of the mesh one is using, and the numerical solution of the PDE. Since

the sensitivity information is obtained by solving symbolically-defined adjoint PDEs, valuable insights can be deduced by analysing their structure. On the other hand, the "discretize-then-optimize" approach is often simpler to solve and allows for more accurate solutions for a particular discretization (Ferguson and Peters 2005). In this thesis we derive both the continuous and discretized gradients for the elastic deformation and heat transfer problems, as the gradients derived in the continuous setting are uncommon in the literature, while the ones in the discretized setting are common but have not been given a formal mathematical treatment.

The standard way of solving simple topology optimization problems for structural compliance has been known since the creation of Sigmund's 99-line code for topology optimization came out in 2001 (Sigmund 2001). Changes to the study of topology optimization came over the next few years in the form of extensions to the code, such as an extension to solve the heat transfer problem and a 3D version of the code (Bendsøe and Sigmund 2004) (Liu and Tovar 2014). In more recent years, a greater focus has been placed on multi-material topology optimization and in multi-physics problems (Chungang et al. 2010). A current research goal in the field is the topology optimization of heat transfer problems that involve radiation. This is due to the difficulty of discretizing the partial differential equation system, as in the continuous setting, radiation is simply treated as a boundary condition. One issue that arises when the system has been discretized is that the non-linearity of the radiation boundary condition is severe, so an iterative process may not converge, and therefore an incremental method is encouraged to guarantee convergence (Bialecki and Nowak 1981). Other concerns are found in developing a method to determine the value of the view factor of each element, as the shape of the design space changes at each iteration. The view factor is a value that represents the amount that one element, or part of the object, can see any other element, or part of the object. The view factor affects the amount of heat transfer that can be achieved through radiation from one element to another. Issues also arise as to the view factor

when dealing with elements whose density value is neither 0 or 1, as these densities do not make sense in real life, and therefore it is hard to decide whether radiation should be modelled in these areas.

In this thesis, we make a contribution to the continued study of structural mechanics by mathematically formalizing the sensitivities of the discretized elasticity and heat transfer problems, and compare them to the sensitivities derived in the continuous setting. This comparison is made by running standardized topology optimization algorithms with both the discrete sensitivity and with the continuous sensitivity. We have also developed a topology optimization algorithm where the 0-1 distribution is forced by adding a penalty term to the objective function, and the volume constraint is satisfied using subspace projection. Lastly, we also derived the sensitivities of coupled elastic - heat transfer problems which include a radiation term in them. This is done in both the continuous and discrete setting, with the hope that in future years, someone will be able to accurately solve the coupled systems of partial differential equations in order to extend the optimization algorithms.

The structure of the thesis is as follows: we started by outlining structural mechanics and on its effectiveness in the current day and age. This is followed by Chapter 2, where we define the mathematical models, variables, and governing systems needed to describe elasticity and heat transfer. Then, in Chapter 3, we describe the optimization problem and the gradient-based algorithm used to solve it, while also deriving the adjoint sensitivities needed for the algorithm. Chapter 4 will go over the various computational results of the algorithm and comment on the effectiveness of it. While Chapter 5 will end the thesis with a summary of the work and any conclusions that can be made.

Chapter 2

Mathematical Models in Elasticity and Heat Transfer

In this thesis we deal with three main physical phenomena. These are elastic deformations under mechanical loads, heat transfer through conduction and heat transfer through radiation. The phenomena are coupled due to the change in temperature in the domain from both conduction and radiation, as the change in temperature causes a thermal strain on the object. The effect of the thermal strain can be found in the equations in the form of the temperature dependence of the strain tensor. In this thesis, we will assume that the small displacement hypothesis is true and it states that any deformations caused by the mechanical loads are treated to be infinitesimally small. Therefore, it is assumed that the mechanical loads do not change the domain enough to affect the temperature equations. Assuming the small displacement hypothesis also allows us to assume that the stress tensors stay the same throughout the optimization process as the ratio of plane and shear stress does not change. If the hypothesis was not assumed, then some areas of the domain would deform more than others, which would cause a bending force in the object. This would change the stress equations as a bending force is typically larger in magnitude than those placed on the object by the mechanical loads.

Since the deformations are so small, the overall shape of the object does not change, and therefore the interaction between the object and the heat source stay the same, which leads to the conduction part of the heat transfer equations staying the same. Secondly, since the shape is the same, the view factors for the radiation components are identical, and therefore the radiation boundary terms remain unchanged.

2.1 Continuous Setting

2.1.1 Notation

In this section we define the different notation used throughout the thesis.

Body force vector: $\boldsymbol{b} : \Omega \to \mathbb{R}^d$

Cauchy stress tensor: $\boldsymbol{\sigma} : \Omega \to \mathbb{R}^{d\times d}$

Convection coefficient: h $\in \mathbb{R}$

Design Domain: $\Omega \subset \mathbb{R}^d$, where d $\in \{2, 3\}$ is the dimension and $\Gamma \subset \mathbb{R}^d$ is the boundary of Ω. The Boundary is defined such that $\Gamma = \bigcup_{i=1}^{4} \Gamma_i$ and $\Gamma_i \cap \Gamma_j = \emptyset \ \ \forall i, j$.

Design variable: $\rho : \Omega \to \mathbb{R}$

Deviatoric stress tensor: $\overline{\boldsymbol{\sigma}} : \Omega \to \mathbb{R}^{d\times d}$

Displacement vector: $\boldsymbol{u} : \Omega \to \mathbb{R}^d$

Elasticity tensor: $\boldsymbol{C}_0 : \Omega \to \mathbb{R}^{d\times d\times d\times d}$

Emissivity coefficient: $\epsilon \in \mathbb{R}$

Environment Temperature: $T_e : \Gamma \to \mathbb{R}$

Euler-Almansi strain tensor: $\boldsymbol{\grave{o}} : \Omega \to \mathbb{R}^{d\times d}$

Heat Capacity: c $\in \mathbb{R}$

Heat conduction coefficient: $k_0 \in \mathbb{R}$

Hydrostatic stress tensor: $\boldsymbol{\sigma_h} : \Omega \to \mathbb{R}^{d\times d}$

Outward normal vector: $\boldsymbol{n} : \Gamma \to \mathbb{R}^d$

Perturbed displacement vector: $\delta \boldsymbol{u} : \Omega \to \mathbb{R}^d$

Perturbed Temperature: $\delta T = T - T_o : \Omega \to \mathbb{R}$, where T_o is the temperature when the volumetric strain vanishes.

Radiation Source Temperature: $T_0 : \Gamma \to \mathbb{R}$

Stefan-Boltzman Constant: $\gamma = 5.670367 \times 10^{-3} kgK^{-4}s^{-3}$

Surface Temperature: $T_s : \Gamma \to \mathbb{R}$

Temperature: $T : \Omega \to \mathbb{R}$

Time: t $\in \mathbb{R}$

Traction vector: $\boldsymbol{t} : \Gamma \to \mathbb{R}^d$

Unit tensor: $\boldsymbol{I} \in \mathbb{R}^{d \times d}$

Volumetric heat source: $p : \Omega \to \mathbb{R}$

von Mises stress: $\sigma_{vm} : \Omega \to \mathbb{R}$

The ":" operator is defined as $\boldsymbol{A} : \boldsymbol{B} = tr(\boldsymbol{B}^T \boldsymbol{A}) = \boldsymbol{B} : \boldsymbol{A}$. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are rank 2 tensors, then $\boldsymbol{A} : \boldsymbol{B} = A_{ij}B_{ij}$, where $A_{ij}B_{ij}$ is in the Einstein summation convention. It is the tensor contraction of the product of two tensors, and reduces the rank of the product by 2.

The L^2 inner product, $\forall$ functionals f, g is: $\langle f, g \rangle = \int_\Omega f g \, d\Omega$.

The H^1 inner product, $\forall$ functionals f, g is: $\langle f, g \rangle = \int_\Omega f g + \boldsymbol{\nabla} f \cdot \boldsymbol{\nabla} g \, d\Omega$.

2.1.2 Definitions of Material Properties and State Variables

In order to later use the governing systems to perform topology optimization, we will replace the elasticity tensor, $\boldsymbol{C}_0$, with

$$\boldsymbol{C} = \rho^P \boldsymbol{C_0}, \tag{2.1}$$

and the heat conduction coefficient, k_0, with

$$k = \rho^P k_0. \tag{2.2}$$

In (2.1) and (2.2), ρ is the density distribution and $P \in \mathbb{N}$ is a penalization factor. P is typically used to encourage our design variable ρ into attaining the values of 0 or 1 as it causes an increase in the objective functionals to occur whenever ρ is close to one. This means that regions that have material placed in them are better prioritized if they lower the objective function. Through heuristic methods and experience, it has been determined that 3 is the optimal value for P.

The Euler-Almansi strain tensor, which describes the relationship between the displacement and strain caused by mechanical loads on the domain, is defined as

$$\grave{\boldsymbol{o}} = \frac{1}{2}\left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right), \tag{2.3}$$

while the thermal strain, caused by the expansion resulting from the heating or cooling of an object, is defined as

$$\grave{\boldsymbol{o}}_{\boldsymbol{T}} = \kappa \cdot (\delta T)\, \boldsymbol{I}. \tag{2.4}$$

The elastic strain (2.3) and the thermal strain (2.4) are then combined to give the total strain in the system as

$$\grave{\boldsymbol{o}}_{\boldsymbol{s}} = \grave{\boldsymbol{o}} - \grave{\boldsymbol{o}}_{\boldsymbol{T}}. \tag{2.5}$$

2.1.3 Constitutive and Governing Equations

We can now use the elasticity tensor and equation (2.5) to postulate the constitutive relation between stress and strain. This particular constitutive relation is called the

Cauchy stress tensor and is defined as

$$\boldsymbol{\sigma} = \boldsymbol{C} : \dot{\boldsymbol{\delta}}_s(\boldsymbol{u}). \tag{2.6}$$

Using (2.6), we are able to define the von-Mises stresses as

$$\boldsymbol{\sigma}_{vm} = \sqrt{\frac{3}{2}\overline{\boldsymbol{\sigma}} : \overline{\boldsymbol{\sigma}}}, \tag{2.7}$$

where the deviatoric stress tensor is defined as

$$\overline{\boldsymbol{\sigma}} = \boldsymbol{\sigma} - \boldsymbol{\sigma_h}, \tag{2.8}$$

and the hydrostatic stress tensor is defined as

$$\boldsymbol{\sigma_h} = \boldsymbol{I}\left[\frac{1}{3}tr(\boldsymbol{\sigma})\right]. \tag{2.9}$$

We do note that in order to calculate the strain, and therefore the stress, present in the system, both the displacement vector $\boldsymbol{u}$ and temperature field T must be found by the solution of the governing equations which are introduced below.

The governing system for the mechanical part of the model is

$$\begin{cases} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\boldsymbol{u}, T) = -\boldsymbol{b} \text{ in } \Omega & \text{(2.10a)} \\ \boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{u}, T) = \boldsymbol{t} \quad \text{on } \Gamma, & \text{(2.10b)} \end{cases}$$

where $\boldsymbol{b}$ represents the body forces in the domain, and $\boldsymbol{t}$ is the traction vector, which is meant to represent the mechanical loads placed on the object. System (2.10) is an expression of Newton's first law of motion and represents the balance between the internal forces of stress and the external forces of applied loads. The weak form of (2.10) is given

by

$$\int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\delta}(\delta \boldsymbol{u})\, d\Omega = \int_{\Omega} \boldsymbol{b} \cdot \delta \boldsymbol{u}\, d\Omega + \int_{\Gamma} \boldsymbol{t} \cdot \delta \boldsymbol{u}\, d\Gamma. \tag{2.11}$$

The weak form is obtained by taking the L^2 inner product between (2.10a) and a test function. This inner product then has integration by parts applied to it and simplified to get an equation resembling (2.11). In this particular case, the test function used was $\delta \boldsymbol{u}$. We will later use the weak form of our adjoint systems in the derivations of our gradients. The governing system for the heat transfer part of the model is

$$\begin{cases} \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) = -p & \text{in } \Omega & \text{(2.12a)} \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T) = 0 & \text{on } \Gamma_1 & \text{(2.12b)} \\ T = T_{\Gamma} & \text{on } \Gamma_2 & \text{(2.12c)} \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T) = \gamma \epsilon \left(T_0^4 - T^4 \right) & \text{on } \Gamma_3 & \text{(2.12d)} \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T) = h \left(T_s - T_e \right) & \text{on } \Gamma_4, & \text{(2.12e)} \end{cases}$$

where p is a heat source found in the domain, (2.12b) represents a part of the boundary which is insulated so that no heat flux is flowing in or out of the system in that area, (2.12c) represents a part of the boundary where the temperature is being kept constant, (2.12d) represents a region of the boundary that is feeling the effects of radiation, and (2.12e) represents the region of the boundary where there is convection through a difference in temperature with respect to the environment. The weak form of (2.12) is given by

$$\int_{\Omega} \left[\boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) \right] (\delta T)\, d\Omega + \int_{\Omega} p\, (\delta T)\ d\Omega = \int_{\Omega} \left(\rho c \frac{\partial T}{\partial t} \right) (\delta T)\ d\Omega. \tag{2.13}$$

Once again, the weak form is obtained by taking the L^2 inner product between (2.12a) and a test function, in this case δT, and then integrating by parts. Note that in transient problems, the time derivative of the temperature function must be defined in order for the form to be correct. In this thesis, we only deal with static problems, and therefore

the right hand side of (2.13) is simply zero. One should note that systems (2.10) and (2.12) are coupled through the inclusion of T in the Cauchy stress tensors in (2.10). Mathematically, this means that to determine the values of $\boldsymbol{u}$ and T, one must solve (2.10) and (2.12) simultaneously. The physical interpretation of the coupled system is that the changing temperature of the object adds thermal strain to the object and therefore will change the amount of displacement the object under-goes when a mechanical load is placed on it. In the uncoupled case, which is used in later computations, the value of T is set to equal T_0 in (2.10), and systems (2.12) and (2.10) are solved separately. This is because when $T = T_0$, then $\delta T = 0$, and therefore the thermal strain (2.4) is also equal to zero, which means that the total strain (2.5) simplifies to the structural strain (2.3). Therefore in the uncoupled case, the governing system for the mechanical part of the system (2.10) simplifies to:

$$\begin{cases} \boldsymbol{\nabla} \cdot (\boldsymbol{C} : \grave{\boldsymbol{o}}_s(\boldsymbol{u})) = -\boldsymbol{b} \text{ in } \Omega & \text{(2.14a)} \\ \quad \boldsymbol{n} \cdot (\boldsymbol{C}\grave{\boldsymbol{o}}_s(\boldsymbol{u})) = \boldsymbol{t} \quad \text{on } \Gamma, & \text{(2.14b)} \end{cases}$$

while the governing system for the thermal part (2.12) stays the same but is no longer required to satisfy the governing system (2.10). The physical interpretation of the uncoupled cases is that they are treated as two different situations, one in which the temperature is kept constant, and the other where there are no mechanical loads placed on the object.

In this thesis, we will only be using isotropic materials, which makes the Cauchy stress tensor $\boldsymbol{\sigma}$, (2.6), symmetric. Now due to the symmetry of $\boldsymbol{\sigma}$, we are able to show

$$\boldsymbol{\sigma} : \boldsymbol{\nabla} \boldsymbol{z} = \boldsymbol{\sigma} : \boldsymbol{\epsilon}(\boldsymbol{z}), \tag{2.15}$$

after some algebraic manipulation. Formula (2.15) is true $\forall \boldsymbol{z} \in \mathbb{R}^d$.

2.1.4 Objective Functionals

Structural compliance will be defined as

$$C_s(\rho) = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{u}\, d\Omega + \int_\Gamma \boldsymbol{t} \cdot \boldsymbol{u}\, d\Gamma, \tag{2.16}$$

and thermal compliance as

$$C_T(\rho) = \int_\Omega pT\, d\Omega. \tag{2.17}$$

In an elastic system, an important measurement is the amount of stress in the system to determine whether or not the material will fracture. Therefore, it is at times beneficial to minimize the von Mises stress to make sure that the material is able to handle the stress placed on it. When the von Mises stresses are the objective functional, the equation is:

$$J(\rho) = \int_\Omega (\sigma_{vm})^2\, d\Omega = \int_\Omega \frac{3}{2}\overline{\boldsymbol{\sigma}} : \overline{\boldsymbol{\sigma}}\, d\Omega, \tag{2.18}$$

Assuming that d = 3, a simplification of the von Mises stress term, using the Einstein summation convention, is:

$$(\boldsymbol{\sigma}_{vm})^2 = \frac{3}{2}\left(\overline{\boldsymbol{\sigma}} : \overline{\boldsymbol{\sigma}}\right), \tag{2.19}$$

$$= \frac{3}{2}\left((\sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk})(\sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk})\right), \tag{2.20}$$

$$= \frac{3}{2}\left(\sigma_{ij}\sigma_{ij} - \frac{2}{3}\delta_{ij}\sigma_{kk}\sigma_{ij} + \frac{1}{9}\delta_{ij}\sigma_{kk}\delta_{ij}\sigma_{kk}\right), \tag{2.21}$$

$$= \frac{3}{2}\left(\sigma_{ij}\sigma_{ij} - \frac{1}{3}\sigma_{kk}^2\right)\ . \tag{2.22}$$

2.2 Discrete Setting

When the continuous formulations, introduced above, are discretized, the structure of the resulting discrete variables is dependant on the particular discretization used. The discretization used below is based on a standard finite element method approach implemented in the 99-line topology optimization code made by Sigmund (Sigmund 2001).

2.2.1 Notation

Design Domain: $\Omega_h \subset \mathbb{R}^d$, where d $\in \{2, 3\}$ is the dimension, h is the mesh parameter, and $\Gamma_h \subset \mathbb{R}^d$ is the boundary of Ω_h. N is the number of elements in mesh Ω_h.

Design variable: $\boldsymbol{\rho} \in \mathbb{R}^N$

Displacement vector: $\boldsymbol{u} \in \mathbb{R}^{dN}$

Global Stiffness tensor: $\boldsymbol{C} \in \mathbb{R}^{dN \times dN}$

Global Thermal Stiffness matrix: $\boldsymbol{K} \in \mathbb{R}^{N \times N}$

Temperature: $\boldsymbol{T} \in \mathbb{R}^{dN}$

Volumetric heat source: $\boldsymbol{p} \in \mathbb{R}^N$

Vectors with exponents are understood to be done component wise, i.e $\boldsymbol{T}^3 = [T_1^3, ..., T_N^3]$.

The $diag()$ function is defined as:

$$diag(y_1, y_2, ..., y_N) \rightarrow \begin{bmatrix} y_1 & 0 & 0 & ... & 0 \\ 0 & y_2 & 0 & ... & 0 \\ ... & ... & ... & ... & 0 \\ 0 & 0 & 0 & ... & y_N \end{bmatrix}$$

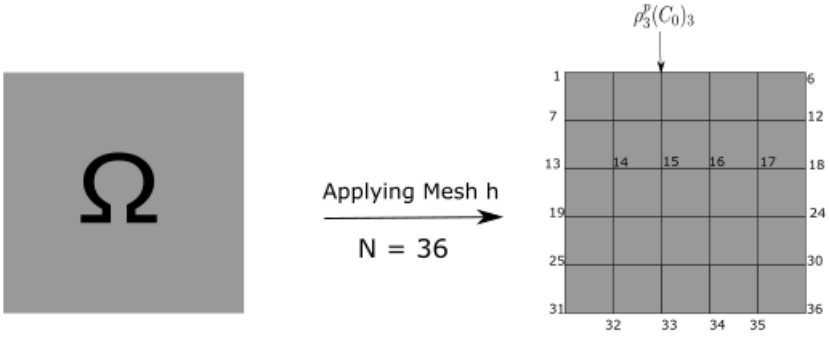


Figure 2.1: Schematic Representation of the Discretization Used

2.2.2 Definitions of Material Properties and State Variables

As in the continuous setting, we must convert the stiffness and thermal stiffness matrices, so that they are suitable for topology optimization. Therefore, the global stiffness matrix is represented as

$$\boldsymbol{C} = diag\left(\rho_i^p(\boldsymbol{C}_0)_i\right), \tag{2.23}$$

and the global thermal stiffness matrix is converted to

$$\boldsymbol{K} = diag\left(\rho_i^z(\boldsymbol{K}_0)_i\right), \tag{2.24}$$

where $i = 1, 2, ..., N-1, N$.

2.2.3 Constitutive and Governing Equations

For the discretized governing systems, the exact form of the system is dependant on whether or not the system is coupled. For both the coupled and uncoupled systems, $\boldsymbol{C}$ is a matrix that contains information about the solution to the structural part of the elliptic equation (2.10a), with Neumann boundary condition (2.10b). $\boldsymbol{K}$ is a matrix that contains information about the solution to the elliptic equation (2.12a), with Neumann

boundary conditions (2.12b) and (2.12e), and Dirichlet boundary condition (2.12c). $\boldsymbol{B}$ is a matrix that contains information about the solution to the thermal part of the elliptic equation (2.10a), with Neumann boundary condition (2.10b). $\boldsymbol{f}$ is a vector containing the values of the forces in equations (2.10a) and (2.10b). $\boldsymbol{P}$ is a vector containing the values of the thermal excitation in equations (2.12a),(2.12b) and (2.12e). $\boldsymbol{T_0}$ is a vector of the value T_0, which is the temperature when the volumetric strain is 0. $\boldsymbol{u}$ is a vector containing the values of the displacement vector $\boldsymbol{u}$ on the nodes of the mesh. $\boldsymbol{T}$ is a vector containing the values of the temperature T on the nodes of the mesh. With these vectors and matrices properly defined, we are able to define the uncoupled governing system for the elasticity problem as

$$\boldsymbol{C}\boldsymbol{u} = \boldsymbol{f}, \tag{2.25}$$

and for the heat transfer problem it is

$$\boldsymbol{K}\boldsymbol{T} = \boldsymbol{P}. \tag{2.26}$$

On the other hand, if the system is strongly coupled, ignoring the non-linear boundary condition on Γ_3, we are left with the discretized version of the governing system as

$$\begin{bmatrix} \boldsymbol{C} & -\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} - \boldsymbol{B}\boldsymbol{T_0} \\ \boldsymbol{P} \end{bmatrix}. \tag{2.27}$$

2.2.4 Objective Functionals

The discretized definition of structural compliance is

$$C_s = \boldsymbol{u}^T \boldsymbol{f}, \tag{2.28}$$

and the discretized definition for thermal compliance is

$$C_s = \boldsymbol{T}^T \boldsymbol{p}. \tag{2.29}$$

The discretized definition of the von Mises Stress is

$$\sigma_{\mathbf{vm}} = \boldsymbol{u}^T \boldsymbol{B}^T \boldsymbol{C}^T \boldsymbol{V} \boldsymbol{C} \boldsymbol{B} \boldsymbol{u}. \tag{2.30}$$

Note that because the left-hand sides of (2.28) and (2.29) are scalars, the right-hand sides are equal to their transpose, and that the T in the superscript is meant to represent the matrix transpose and not the temperature field.

Chapter 3

Structural Optimization Problem

3.1 Problem Set-up

In topology optimization, the goal is to put holes in the object, by making regions in the design space where the design variable is equal to zero. This will affect our objective functional by changing the stiffness matrices, (2.1), (2.2),(2.23), (2.8), to be equal to zero in those regions. These changes to the stiffness matrices will result in a change to the solutions of our governing systems, (2.10),(2.12),(2.25) and (2.26). This in turn will change the values of the objective functional $J(\rho)$. Our particular objective functional will depend on which problem we are looking at. In the uncoupled situations, in the continuous setting, the objective functions will be structural (2.16) and thermal compliance (2.17) respectively, with structural compliance (2.16) being used in the computational part of this thesis. While in the coupled case, the objective function will be total compliance, which is found by summing (2.16) and (2.17). In the discretize then optimize case, (2.28) and (2.29) are used when uncoupled, while their sum is used in the coupled situation. For the later computations, the objective functional will be the sum of structural compliance (2.16) and a penalization term, (J_p in equation (3.1)), to force

the design variable towards 0 or 1. Therefore, our objective function will be:

$$J_s(\rho) = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{u}\, d\Omega + \int_\Gamma \boldsymbol{t} \cdot \boldsymbol{u}\, d\Gamma + \underbrace{\alpha \int_\Omega \left(\rho\left[1-\rho\right]\right)^2 \, d\Omega}_{J_p} = J_0 + J_p, \tag{3.1}$$

where $\alpha \in \mathbb{R}^+$ is a constant determining how the two parts of the objective function (3.1) are balanced. It is used to change how fast the design variable is pushed toward the $0-1$ distribution. With this in mind, we are able to state the general problem mathematically as:

$$\min_{\rho \in L^2(\Omega)} J(\rho), \tag{3.2a}$$

$$\text{subject to:} \qquad 0 \leq \rho \leq 1, \tag{3.2b}$$

$$M = \int_\Omega \rho\, d\Omega = M_0. \tag{3.2c}$$

In the optimization problem (3.2), the second constraint (3.2c) can be viewed as a cap on the maximum mass of the object. While the first constraint (3.2b), is put in place to make sure that the design variable, density, makes sense in the physical world, as it is a mass fraction and therefore must be between 0 and 1. To satisfy this constraint computationally, a change of variables was applied to the design variable ρ. Namely, we change the definition of ρ to be a function of η such that $\rho : \mathbb{R} \to [0,1]$, where $\eta : \Omega \to \mathbb{R}$.

This change of variables converts the optimization problem into:

$$\min_{\eta \in L^2(\Omega)} J(\eta), \tag{3.3a}$$

$$\text{subject to:} \qquad M = \int_\Omega \rho(\eta)\, d\Omega = M_0. \tag{3.3b}$$

This change of variables affects the later derivations by requiring the use of the chain rule to transform the gradients into the correct variables and therefore:

$$\boldsymbol{\nabla}_\eta J(\eta) = \rho'(\eta) \boldsymbol{\nabla}_{\rho(\eta)} J(\rho(\eta)), \tag{3.4}$$

where $\nabla_{\rho(\eta)} J(\rho(\eta))$ will be the expression for the gradient of the objective functional (3.1) derived later on. For computational purposes, the change of variables used was:

$$\rho = \rho(\eta) = \frac{\tanh(\eta)}{2} + \frac{1}{2}. \tag{3.5}$$

The $\tanh(x)$ function was used as it gave the desired range, was not periodic, and has the domain of the entire real line. It was also chosen over $\arctan(x)$, as in *MATLAB* (MATLAB 2019), $\tanh(x)$ has a faster convergence to its extreme values.

3.2 Optimization Approach

One of the main mathematical approaches for solving an optimization problem like (3.3) is to use a gradient-based optimization algorithm. The main two methods tested in this thesis are called steepest-decent and conjugate gradients. Since the gradient is a vector that points in the direction of the steepest increase in the value of the objective functional, one can create an optimization algorithm by determining the value of the gradient at the given step and subtracting it from the design variable. Both approaches still rely on using the gradient at a particular step, which means that we require a formula to calculate the gradient of the objective functional (3.4). We know that we are able to calculate such a formula thanks to the Riesz representation theorem which states:

Theorem(Riesz Representation Theorem (Rudin 2006)) *Let $\mathcal{H}$ be a Hilbert space. Then for all bounded linear functionals $f \in \mathcal{H}^*$, there exists a $g \in \mathcal{H}$ such that for all $h \in \mathcal{H}$, $f(h) = \langle h, g \rangle$*

In our case, the Hilbert space that we will be acting in will be L^2 and the bounded linear functional will be the *Gâteaux* derivative. The *Gâteaux* derivative is the generalization

of the directional derivative and conveys information about the magnitude of the steepest ascent, as opposed to the gradient which gives the direction of the steepest ascent. The *Gâteaux* derivative is calculated using:

$$J'(\rho;\rho') = \lim_{\rho'\to 0} \frac{J(\rho+\rho') - J(\rho)}{\rho'}, \tag{3.6}$$

where ρ' is an arbitrary perturbation of the design variable. Now by using the inner product in L^2, the *Gâteaux* derivative and the Riesz Representation Theorem, we are able to create an equation which will be refereed to as the Riesz form, namely:

$$J'(\rho;\rho') = \int_\Omega \boldsymbol{\nabla} J(\rho)\rho'\, d\Omega\ . \tag{3.7}$$

where $\boldsymbol{\nabla} J$ is the gradient of the objective functional. We will derive the expression for the gradient of our objective functional by using the adjoint variable method. Our derivation will involve taking the L^2 inner product between the adjoint variables and the *Gâteaux* derivative of the governing system and transforming it into an expression which resembles the Riesz form (3.7). The adjoint variable method is a method for computing the gradient of a function by defining an adjoint variable which must solve a modification of the governing system, whose forcing terms are chosen accordingly. The adjoint variable is denoted using the $*$ symbol, for example u^*. Adjoint variables, when suitably defined, convey information about the sensitivity of the state variables to the perturbations of parameters, e.g. $\boldsymbol{u}^*$ contains information about the sensitivity of $\boldsymbol{u}$ to perturbations of ρ. The operator defining the adjoint system is determined by the governing system but we are allowed to choose the boundary conditions and the right-hand side source/forcing terms. Our choice of these terms and boundary conditions is limited by the fact that we are not allowed to pick terms that contain a perturbation variable, as in principle, they may take arbitrary values. We exploit this freedom in choosing the right-hand side and the boundary conditions, in order to use the adjoint

system to express the directional derivative in the Riesz form. This allows us to identify an expression for the gradient/sensitivity of our objective function. Typically, for a particular objective functional, our choice of the boundary values or source term in the adjoint system is unique. We have to go through these derivation to compute the expression for the gradient of the objective functional, as opposed to using a symbolic manipulation program, like *MAPLE* or the symbolic toolbox in *MATLAB*, because we are looking for the gradient with respect to ρ, which is an element of a functional space. This is an issue for a symbolic program as they are only able to integrate or find the derivatives with respect to a variable in $\mathbb{R}^n$.

One issue in using a gradient based method in topology optimization is called the checkerboard problem. The checkerboard problem occurs when there is a region in the domain in which the optimization algorithm leads to alternating values of 0 and 1, and therefore makes it look like a checker or chess board ((Bendsøe and Sigmund 2004)). This is an issue as the physical interpretation of this is regions of floating material. To combat this, many heuristic approaches have been implemented, for example, in the 99-line topology optimization code by Sigmund (Sigmund 2001), the gradient is smoothed by averaging out the value of the gradient at a particular element using the surrounding elements. The radius of the surrounding elements used is then a parameter in the code. In this thesis, we dealt with this checkerboard issue by taking our expressions for the gradients of our objective functionals derived in the L^2 space, and projecting them into the more continuous Sobolev space of H^1. We are able to apply this transformation thanks to the Riesz Representation Theorem, as it states that a bounded linear function, in this case the *Gâteaux* derivative of our objective functional, has a representation in a Hilbert space as an inner product between the gradient of the objective functional and a variable, in this case ρ'. Since the L^2 and H^1 spaces are both Hilbert spaces, they must be equal to each other, i.e:

$$\langle \boldsymbol{\nabla} J, \rho' \rangle_{L^2} = J'(\rho; \rho') = \langle \boldsymbol{\nabla} J, \rho' \rangle_{H^1}. \tag{3.8}$$

Using the definition of the inner product in these spaces, this means that:

$$\int_\Omega \left[\boldsymbol{\nabla}_{\boldsymbol{x}} J\right] \rho' \, d\Omega = \int_\Omega \left[J - \ell^2 \boldsymbol{\nabla}_{\boldsymbol{x}} J\right] \rho' \, d\Omega, \tag{3.9}$$

where the value ℓ^2 determines how much smoothing is occurring in the transition, and can be interpreted as a change of units between the two spaces. Note that this equality is only true for values of ℓ^2 away from 0 and ∞. Therefore, to project our L^2 gradient into the H^1 space, we must solve the following PDE system:

$$\begin{cases} \left[1 - \ell^2 \boldsymbol{\nabla}_{\boldsymbol{x}}\right] \boldsymbol{\nabla}_{H^1} J = \boldsymbol{\nabla}_{L^2} J \text{ on } \Omega & \text{(3.10a)} \\ \dfrac{\partial}{\partial \boldsymbol{n}} \boldsymbol{\nabla}_{H^1} J = \mathbf{0} \qquad \text{on } \Gamma. & \text{(3.10b)} \end{cases}$$

In the discrete setting, solving the PDE system (3.10) is equivalent to solving:

$$\left(\boldsymbol{M} - \ell^2 \boldsymbol{K}\right) \boldsymbol{\nabla}_{H^1} J = \boldsymbol{\nabla}_{L^2} J, \tag{3.11}$$

where $\boldsymbol{M} \in R^{d \times d}$ is the standard mass matrix for the discretization, and $\boldsymbol{K}$ is the same stiffness matrix as in equation (2.26).

One issue with using a steepest decent approach in optimization is that it can be inefficient at times, as it can get stuck in steeper local valleys as oppose to heading towards an overall minimum. To combat this, conjugate gradients can be used by taking information of the gradient at every previous step to choose a more efficient direction at the current step. The conjugate gradient algorithm is shown in Algorithm 1.

Algorithm 1 Implementation of the conjugate-gradients variant of descent approach

Input:

ρ_0 — initial guess for mass distribution

Output:

$J(\rho)$ — Structural Compliance

$\rho^{(0)} \leftarrow \rho_0$(Initial guess)
$n \leftarrow 1$
repeat
  evaluate gradient $\boldsymbol{\nabla}_\rho J$
  Compute β_n
  Update the conjugate direction: $S_n = -\boldsymbol{\nabla}_\rho J - \beta_n s_{n-1}$
  Update the design variable: $\rho_{n+1} = \rho_n + s_n$
  $n \leftarrow n + 1$
  Evaluate $J(\rho)$
until Condition Met

One of the main parameters that one can choose is the expression for β, as there are many choices for its expression, but the most popular formulas are the Fletcher-Reeves

$$\beta_n = \frac{\boldsymbol{\nabla}\eta_n^T \boldsymbol{\nabla}\eta_n}{\boldsymbol{\nabla}\eta_{n-1}^T \boldsymbol{\nabla}\eta_{n-1}} \tag{3.12}$$

and the Polak-Ribiere, ((Nocedal and Wright 1999)):

$$\beta_n = \frac{\boldsymbol{\nabla}\eta_n^T (\boldsymbol{\nabla}\eta_n - \boldsymbol{\nabla}\eta_{n-1})}{\boldsymbol{\nabla}\eta_{n-1}^T \boldsymbol{\nabla}\eta_{n-1}}. \tag{3.13}$$

3.3 Derivations in the Continuous Setting

3.3.1 Uncoupled Cases

We will now present the derivations of gradient expressions for different objective functionals, starting with structural compliance in the uncoupled system (2.16). The derivation will be subject to the structural governing system (2.10). The *Gâteaux* derivative

of the objective functional (2.16) is:

$$J'(\rho;\rho') = \int_\Omega \boldsymbol{b}\cdot\boldsymbol{u}'(\rho;\rho')\,d\Omega + \int_\Gamma \boldsymbol{t}\cdot\boldsymbol{u}'(\rho,\rho')\,d\Gamma, \tag{3.14}$$

while the *Gâteaux* derivative of the governing system (2.16) is:

$$\begin{cases} \boldsymbol{\nabla}\cdot[P\rho^{P-1}\rho'\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}) + \rho^P\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}')] = 0 \text{ in } \Omega & \text{(3.15a)} \\ \boldsymbol{n}\cdot[P\rho^{P-1}\rho'\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}) + \rho^P\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}')] = 0 \text{ on } \Gamma, & \text{(3.15b)} \end{cases}$$

where the solution of the system is given by the perturbation variable $\boldsymbol{u}'(\rho;\rho')$. We will now define the adjoint system as

$$\begin{cases} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}(\boldsymbol{u}^*) = \boldsymbol{b} \text{ on } \Omega & \text{(3.16a)} \\ \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u}^*) = \boldsymbol{t} \text{ on } \Gamma, & \text{(3.16b)} \end{cases}$$

whose weak form solution is given by:

$$\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}'):\boldsymbol{\grave{o}}(\boldsymbol{u}^*)\,d\Omega = \int_\Omega \boldsymbol{b}\cdot\boldsymbol{u}'\,d\Omega + \int_\Gamma \boldsymbol{t}\cdot\boldsymbol{u}'\,d\Gamma. \tag{3.17}$$

We start by taking the L^2 inner product between the perturbed system (3.15a) and the adjoint variable $\boldsymbol{u}^*$ to give:

$$0 = \int_\Gamma \boldsymbol{n}\cdot[P\rho^{P-1}\rho'\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}) + \rho^P\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}')]\cdot\boldsymbol{u}^*\,d\Gamma \tag{3.18}$$

$$= \int_\Gamma [\boldsymbol{n}\cdot(P\rho^{P-1}\rho'\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}))]\cdot\boldsymbol{u}^*\,d\Gamma + \int_\Gamma [\boldsymbol{n}\cdot(\rho^P\boldsymbol{C_0}:\boldsymbol{\grave{o}}(\boldsymbol{u}'))]\cdot\boldsymbol{u}^*\,d\Gamma. \tag{3.19}$$

Using integration by parts on both integrals in (3.19) gives

$$\begin{aligned} &= \int_\Omega [\boldsymbol{\nabla} \cdot (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}))] \cdot \boldsymbol{u}^* \, d\Omega + \int_\Omega [\boldsymbol{\nabla} \cdot (\rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}'))] \cdot \boldsymbol{u}^* \, d\Omega \\ &\quad - \int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega - \int_\Omega (\rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}')) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega, \end{aligned} \tag{3.20}$$

$$\begin{aligned} &= \int_\Omega \boldsymbol{\nabla} \cdot [P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) + (\rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}'))] \cdot \boldsymbol{u}^* \, d\Omega \\ &\quad - \int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega - \int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}')) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega. \end{aligned} \tag{3.21}$$

Applying identity (2.15) to the last integral and noticing the first integral contains a term from the perturbed system (3.15b) implies:

$$- \int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega - \int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}')) : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}^*) \, d\Omega = 0. \tag{3.22}$$

After rearranging and using the weak form of the adjoint system (3.17) we end up with:

$$J'(\rho; \rho') = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{u}'(\rho; \rho') \, d\Omega + \int_\Gamma \boldsymbol{t} \cdot \boldsymbol{u}' \, d\Gamma = - \int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega. \tag{3.23}$$

Finally by applying (2.15) again, and rearranging, it turns into the Riesz from (3.7):

$$J'(\rho; \rho') = - \int_\Omega \rho' ([P\rho^{P-1} \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})] : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}^*)) \, d\Omega. \tag{3.24}$$

Therefore, we are able to identify the expression for the gradient of the objective functional (2.16) as:

$$\implies \boxed{\boldsymbol{\nabla} J(\rho) = -[P\rho^{P-1} \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})] : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}^*) = - \left[\frac{d\boldsymbol{C}}{d\rho} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}) \right] : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}^*).} \tag{3.25}$$

Using the simplification (2.22), the *Gâteaux* derivative of the objective function (2.18) is:

$$J'(\rho;\rho') = \int_\Omega \frac{3}{2}\left(2\sigma_{ij}\frac{d\sigma_{ij}}{d\rho} - \frac{2}{3}\sigma_{kk}\delta_{ij}\frac{d\sigma_{ij}}{d\rho}\right) d\Omega, \tag{3.26}$$

$$= \int_\Omega 3\left(\sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}\right)\frac{d\sigma_{ij}}{d\rho}\, d\Omega, \tag{3.27}$$

$$= 3\int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : [P\rho^{P-1}\rho' \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}')]\, d\Omega\ . \tag{3.28}$$

The *Gâteaux* derivative of the governing system (2.10) is:

$$\begin{cases} \boldsymbol{\nabla}\cdot[P\rho^{P-1}\rho' \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}')] = 0 \quad \text{on } \Omega & \text{(3.29a)} \\ \boldsymbol{n}\cdot[P\rho^{P-1}\rho' \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}')] = 0 \quad \text{on } \Gamma, & \text{(3.29b)} \end{cases}$$

where the solution of the system is given by the perturbation variable $\boldsymbol{u}'(\rho;\rho')$. We will now define the adjoint system as

$$\begin{cases} \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}(\boldsymbol{u}^*) = \boldsymbol{\nabla}\cdot[3\,(\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : \boldsymbol{C}] \quad \text{on } \Omega & \text{(3.30a)} \\ \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{u}^*) = 0 \quad \text{on } \Gamma, & \text{(3.30b)} \end{cases}$$

whose weak form is given by:

$$\int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}') : \grave{\boldsymbol{o}}(\boldsymbol{u}^*)\, d\Omega = 3\int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : \boldsymbol{\sigma}(\boldsymbol{u}')\, d\Omega\ . \tag{3.31}$$

Proceeding just as before, taking the L^2 inner product between the adjoint variable $\boldsymbol{u}^*$ and the perturbed system (3.29a) gives:

$$0 = \int_\Omega [\boldsymbol{\nabla}\cdot[P\rho^{P-1}\rho' \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}')]]\cdot \boldsymbol{u}^*\, d\Omega, \tag{3.32}$$

$$= \int_\Omega [\boldsymbol{\nabla}\cdot(P\rho^{P-1}\rho' \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}))]\cdot \boldsymbol{u}^*\, d\Omega + \int_\Omega [\boldsymbol{\nabla}\cdot(\rho^P \boldsymbol{C_0} : \grave{\boldsymbol{o}}(\boldsymbol{u}'))]\cdot \boldsymbol{u}^*\, d\Omega\ . \tag{3.33}$$

Using integration by parts on both integrals in (3.33) gives:

$$\begin{aligned} &= -\int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla u}^* \, d\Omega - \int_\Omega (\rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}')) : \boldsymbol{\nabla u}^* \, d\Omega \\ &\quad + \int_\Gamma \boldsymbol{n} \cdot ([P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})] \cdot \boldsymbol{u}^*) + \boldsymbol{n} \cdot ([\rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}')] \cdot \boldsymbol{u}^*) \, d\Gamma, \end{aligned} \tag{3.34}$$

$$\begin{aligned} &= -\int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla u}^* \, d\Omega - \int_\Omega (\rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}')) : \boldsymbol{\nabla u}^* \, d\Omega \\ &\quad + \int_\Gamma [\boldsymbol{n} \cdot ([P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})] + [\rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}')])] \cdot \boldsymbol{u}^* \, d\Gamma \ . \end{aligned} \tag{3.35}$$

Applying identity (2.15) to the second integral and noticing the third integral in (3.35) contains a term from the perturbed system (3.29b) gives:

$$0 = -\int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla u}^* \, d\Omega - \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}') : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}^*) \, d\Omega \ . \tag{3.36}$$

Applying the weak form of the adjoint system (3.31) to the second integral gives:

$$0 = -\int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) : \boldsymbol{\nabla u}^* \, d\Omega - 3\int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : \boldsymbol{\sigma}(\boldsymbol{u}') \, d\Omega, \tag{3.37}$$

and after rearranging:

$$3\int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : \boldsymbol{\sigma}(\boldsymbol{u}') \, d\Omega = -\int_\Omega \left(P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})\right) : \boldsymbol{\nabla u}^* \, d\Omega \ . \tag{3.38}$$

Adding $3\int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) \, d\Omega$ to both sides gives:

$$\begin{aligned} &3\int_\Omega (\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : \left[P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}')\right] \, d\Omega \\ &= \int_\Omega 3\,(\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h) : (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})) \\ &\quad - \left(P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u})\right) : \dot{\boldsymbol{\varepsilon}}(\boldsymbol{u}^*) \, d\Omega \ . \end{aligned} \tag{3.39}$$

Noting that the left hand side of the equation is the directional (*Gâteaux*) derivative of the objective function (3.28), and the right hand side can be rearranged to be in the

Riesz form (3.7):

$$J'(\rho;\rho') = \int_\Omega \rho' \left([3\left(\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}_h\right) : (\frac{d\boldsymbol{C}}{d\rho} : \grave{\boldsymbol{o}}(\boldsymbol{u}))] - \left[\left(\frac{d\boldsymbol{C}}{d\rho} : \grave{\boldsymbol{o}}(\boldsymbol{u}) \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) \right] \right) d\Omega. \quad (3.40)$$

and therefore we can identify the expression of the gradient for the objective functional (2.7) as:

$$\implies \boxed{\boldsymbol{\nabla} J(\rho) = \left[3\overline{\boldsymbol{\sigma}}(\boldsymbol{u}) : \left(\frac{d\boldsymbol{C}}{d\rho} : \grave{\boldsymbol{o}}(\boldsymbol{u}) \right) \right] - \left(\frac{d\boldsymbol{C}}{d\rho} : \grave{\boldsymbol{o}}(\boldsymbol{u}) \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*).} \quad (3.41)$$

The last of the objective functionals in the uncoupled continuous setting will be thermal compliance (2.17), and its governing system will be (2.12), with the simplification that we will only look at the first boundary condition for the time being (2.12). With this assumption, we get that the *Gâteaux* derivative of the objective functional (2.17) is:

$$J'(\rho;\rho') = \int_\Omega pT' \, d\Omega, \quad (3.42)$$

and the *Gâteaux* derivative of the governing system (2.12) is:

$$\begin{cases} \boldsymbol{\nabla} \cdot \left[\left(P\rho^{P-1}\rho' k_0 \right) \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] = 0 \text{ in } \Omega & (3.43a) \\ \boldsymbol{n} \cdot \left[\left(P\rho^{P-1}\rho' k_0 \right) \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] = 0 \text{ on } \Gamma. & (3.43b) \end{cases}$$

We will define the adjoint system as:

$$\begin{cases} \boldsymbol{\nabla} \cdot [k \boldsymbol{\nabla} T^*] = -p \quad \text{on } \Omega & (3.44a) \\ \boldsymbol{n} \cdot [k \boldsymbol{\nabla} T^*] = 0 \quad \text{on } \Gamma. & (3.44b) \end{cases}$$

As before, we start by taking the L^2 inner product between the adjoint variable T^* and the perturbed system (3.43a), which gives:

$$0 = \int_\Omega T^* [\boldsymbol{\nabla} \cdot \left[\left(P\rho^{P-1}\rho' k_0 \right) \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right]] \, d\Omega \; . \tag{3.45}$$

Applying integration by parts to (3.45) gives:

$$\begin{aligned} 0 = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \, d\Omega \\ &+ \int_\Gamma T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \right] \, d\Gamma \; . \end{aligned} \tag{3.46}$$

Splitting the first integral into two integrals and noticing the second integral contains a part of the perturbed system (3.43b) gives:

$$0 = - \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega - \int_\Omega \boldsymbol{\nabla} T^* \cdot [k \boldsymbol{\nabla} T'] \, d\Omega \; . \tag{3.47}$$

Integrating the second integral by parts gives:

$$\begin{aligned} = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega + \int_\Omega \boldsymbol{\nabla} \cdot ([k \boldsymbol{\nabla} T^*] T') \, d\Omega \\ &- \int_\Gamma T' [k \boldsymbol{\nabla} T^*] \cdot \boldsymbol{n} \, d\Gamma \; . \end{aligned} \tag{3.48}$$

Noting that in the second and third integrals in (3.48), there are terms defined in the adjoint system (3.44) gives:

$$\implies 0 = - \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega - \int_\Omega p T' \, d\Omega, \tag{3.49}$$

and after rearranging the terms:

$$\implies J(\rho; \rho') = \int_\Omega p T' \, d\Omega = - \int_\Omega \left[\boldsymbol{\nabla} T^* \cdot \left(P\rho^{P-1} k_0 \boldsymbol{\nabla} T \right) \right] \rho' \, d\Omega. \tag{3.50}$$

Since the above equation is in the Riesz form (3.7), we are able to identify the expression for the gradient of the objective functional (2.17) in the simplified case as:

$$\implies \boxed{\boldsymbol{\nabla} J(k(\rho)) = -\left[\boldsymbol{\nabla} T^* \cdot \left(P\rho^{P-1} k_0 \boldsymbol{\nabla} T\right)\right] .} \tag{3.51}$$

For the following derivation we will no longer use the previous simplification of not including the radiation boundary condition (2.12d) in the governing system (2.12). Since the objective functional is once again (2.17), the *Gâteaux* derivative of the objective function is once again (3.42), and the perturbed system is now:

$$\begin{cases} \boldsymbol{\nabla} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = 0 & \text{in } \Omega & (3.52a) \\ \boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = 0 & \text{on } \Gamma_1 & (3.52b) \\ T' = 0 & \text{on } \Gamma_2 & (3.52c) \\ \boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = -4\gamma\epsilon T^3 T' & \text{on } \Gamma_3 & (3.52d) \\ \boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = 0 & \text{on } \Gamma_4. & (3.52e) \end{cases}$$

We will now define the adjoint system as:

$$\begin{cases} \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T^*) = -p & \text{in } \Omega & (3.53a) \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T^*) = 0 & \text{on } \Gamma_1 & (3.53b) \\ T^* = 0 & \text{on } \Gamma_2 & (3.53c) \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T^*) = -4\gamma\epsilon T^3 T^* & \text{on } \Gamma_3 & (3.53d) \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T^*) = 0 & \text{on } \Gamma_4 & (3.53e) \end{cases}$$

We start the derivation by taking the L^2 inner product between the perturbed system (3.52a) and the adjoint variable T^*, which gives:

$$0 = \int_\Omega T^* [\boldsymbol{\nabla} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right]] \, d\Omega \ , \tag{3.54}$$

and after integration by parts we end up with:

$$\begin{aligned} 0 = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \, d\Omega \\ &+ \int_\Gamma T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \right] \, d\Gamma, \end{aligned} \tag{3.55}$$

and splitting the first integral into two integrals and splitting the second integral into 4 parts yields:

$$\begin{aligned} 0 = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega - \int_\Omega \boldsymbol{\nabla} T^* \cdot [k \boldsymbol{\nabla} T'] \, d\Omega \\ &+ \int_{\Gamma_1} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \right] \, d\Gamma_1 \\ &+ \int_{\Gamma_2} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \right] \, d\Gamma_2 \\ &+ \int_{\Gamma_3} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \right] \, d\Gamma_3 \\ &+ \int_{\Gamma_4} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \right] \, d\Gamma_4 \ . \end{aligned} \tag{3.56}$$

Noting that the boundary integrals all contain terms found in the perturbed system (3.52b)–(3.52e) gives:

$$\begin{aligned} 0 = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega - \int_\Omega \boldsymbol{\nabla} T^* \cdot [k \boldsymbol{\nabla} T'] \, d\Omega \\ &+ \int_{\Gamma_2} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \right] \, d\Gamma_2 + \int_{\Gamma_3} T^* \left(-4\gamma\epsilon T^3 T' \right) \, d\Gamma_3 \ . \end{aligned} \tag{3.57}$$

Integrating the second integral by parts gives:

$$\begin{aligned} 0 = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega + \int_\Omega \boldsymbol{\nabla} \cdot \left([k\boldsymbol{\nabla} T^*] T' \right) d\Omega \\ &- \int_\Gamma T'[k\boldsymbol{\nabla} T^*] \cdot \boldsymbol{n} \, d\Gamma \\ &+ \int_{\Gamma_2} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \right] \, d\Gamma_2 + \int_{\Gamma_3} T^* \left(-4\gamma\epsilon T^3 T' \right) \, d\Gamma_3, \end{aligned} \tag{3.58}$$

and splitting the third integral gives:

$$\begin{aligned} 0 = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega + \int_\Omega \boldsymbol{\nabla} \cdot \left([k\boldsymbol{\nabla} T^*] T' \right) d\Omega \\ &- \int_{\Gamma_1} T'[k\boldsymbol{\nabla} T^*] \cdot \boldsymbol{n} \, d\Gamma_1 - \int_{\Gamma_2} T'[k\boldsymbol{\nabla} T^*] \cdot \boldsymbol{n} \, d\Gamma_2 \\ &- \int_{\Gamma_3} T'[k\boldsymbol{\nabla} T^*] \cdot \boldsymbol{n} \, d\Gamma_3 - \int_{\Gamma_4} T'[k\boldsymbol{\nabla} T^*] \cdot \boldsymbol{n} \, d\Gamma_4 \\ &+ \int_{\Gamma_2} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \right] \, d\Gamma_2 + \int_{\Gamma_3} T^* \left(-4\gamma\epsilon T^3 T' \right) \, d\Gamma_3 \, . \end{aligned} \tag{3.59}$$

Noting that the second to the sixth integrals contain a part of the adjoint system (3.53) allows us to transform (3.59) into:

$$\begin{aligned} 0 = &- \int_\Omega \boldsymbol{\nabla} T^* \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \, d\Omega - \int_\Omega pT' \, d\Omega \\ &+ \int_{\Gamma_3} T^* \left(4\gamma\epsilon T^3 T' \right) \, d\Gamma_3 - \int_{\Gamma_3} T^* \left(4\gamma\epsilon T^3 T' \right) \, d\Gamma_3, \end{aligned} \tag{3.60}$$

and after rearranging, we are able to transform it into the Riesz form (3.7):

$$J'(\rho;\rho') = \int_\Omega pT' \, d\Omega = - \int_\Omega \left[\boldsymbol{\nabla} T^* \cdot \left(P\rho^{P-1} k_0 \boldsymbol{\nabla} T \right) \right] \rho' \, d\Omega. \tag{3.61}$$

Since the equation (3.61) is now in the Riesz form (3.7), we are able to identify the expression for the gradient of our objective functional (2.17) as:

$$\implies \boxed{\boldsymbol{\nabla} J(k(\rho)) = - \left[\boldsymbol{\nabla} T^* \cdot \left(P\rho^{P-1} k_0 \boldsymbol{\nabla} T \right) \right] \, .} \tag{3.62}$$

We note that because of the small displacement hypothesis, the mechanical loads and the displacement from them, does not affect the amount of thermal compliance in the system. This leads us to having an expression for the gradient that does not contain a term with $\boldsymbol{u}$ in it. Note that the gradient expressions for the simplified thermal case (3.51) and the full thermal case with the radiation boundary condition (3.62) are exactly the same. The difference in the two expressions is that T and T^* are solutions of different governing and adjoint systems. This is because in the simplified case there is no radiation boundary condition (3.44) as there is in the more general case (3.53).

3.3.2 Coupled Case

For the last derivation in the continuous setting, we will be looking at the most general case as we no longer assume that $T = T_0$ and therefore it will no longer be true that $\delta T = 0$. This means that there is now thermal strain in the system and the system is now coupled. Our objective functional for this case will be structural compliance (2.16). The derivation will be subject to the the coupled system of (2.12) and (2.10). The *Gâteaux* derivative of the objective functional (2.16) is then:

$$J_s'(\rho;\rho') = \int_\Omega \boldsymbol{b} \cdot \left[\boldsymbol{u}'(\rho;\rho')\right] \, d\Omega + \int_\Gamma \boldsymbol{t} \cdot \left[\boldsymbol{u}'(\rho;\rho')\right] \, d\Gamma \ , \tag{3.63}$$

and the *Gâteaux* derivative of the coupled governing system is then:

$$\begin{cases} \boldsymbol{\nabla} \cdot [P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{o}}_s(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{o}}(\boldsymbol{u}') - \dot{\boldsymbol{o}}_t(T')]] = 0 & \text{in } \Omega & \text{(3.64a)} \\ \boldsymbol{n} \cdot [P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{o}}_s(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{o}}(\boldsymbol{u}') - \dot{\boldsymbol{o}}_t(T')]] = 0 & \text{on } \Gamma & \text{(3.64b)} \\ \boldsymbol{\nabla} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = 0 & \text{in } \Omega & \text{(3.64c)} \\ \boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = 0 & \text{on } \Gamma_1 & \text{(3.64d)} \\ T' = 0 & \text{on } \Gamma_2 & \text{(3.64e)} \\ \boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = -4\gamma\epsilon T^3 T' & \text{on } \Gamma_3 & \text{(3.64f)} \\ \boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T'\right] = 0 & \text{on } \Gamma_4, & \text{(3.64g)} \end{cases}$$

whose solution is given by the perturbation variables $\boldsymbol{u}'(\rho;\rho')$ and $T'(\rho;\rho')$. We will now define an adjoint system in the form:

$$\begin{cases} \boldsymbol{\nabla} \cdot [\boldsymbol{C} : \dot{\boldsymbol{o}}(\boldsymbol{u}^*)] = -\boldsymbol{b} & \text{in } \Omega & \text{(3.65a)} \\ \boldsymbol{n} \cdot [\boldsymbol{C} : \dot{\boldsymbol{o}}(\boldsymbol{u}^*)] = \boldsymbol{t} & \text{on } \Gamma & \text{(3.65b)} \\ \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T^*) = \left[(\tau \cdot \boldsymbol{I}) : \rho^P \boldsymbol{C}_0\right] : \dot{\boldsymbol{o}}(\boldsymbol{u}^*) & \text{in } \Omega & \text{(3.65c)} \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T^*) = 0 & \text{on } \Gamma_1 & \text{(3.65d)} \\ T^* = 0 & \text{on } \Gamma_2 & \text{(3.65e)} \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T^*) = -4\gamma\epsilon T^3 T^* & \text{on } \Gamma_3 & \text{(3.65f)} \\ \boldsymbol{n} \cdot (k \boldsymbol{\nabla} T^*) = 0 & \text{on } \Gamma_4, & \text{(3.65g)} \end{cases}$$

whose weak form is given by:

$$\int_\Omega (\boldsymbol{C} : \dot{\boldsymbol{o}}(\boldsymbol{u}')) : \dot{\boldsymbol{o}}(\boldsymbol{u}^*)\, d\Omega = \int_\Omega \boldsymbol{b} \cdot \boldsymbol{u}'\, d\Omega + \int_\Gamma \boldsymbol{t} \cdot \boldsymbol{u}'\, d\Gamma\ . \tag{3.66}$$

We start the derivation by summing the L^2 inner products between the adjoint variable $\boldsymbol{u}^*$ and the perturbed system (3.64a) and between the adjoint variable T^* and the

perturbed system (3.64d) which gives:

$$\begin{aligned} 0 = &\int_\Omega [\boldsymbol{\nabla} \cdot \left[P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\o}}_s(\boldsymbol{u}) + \rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{\o}}(\boldsymbol{u}') - \dot{\boldsymbol{\o}}_T(T')] \right]] \cdot \boldsymbol{u}^* \\ &+ \boldsymbol{\nabla} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] T^* \, d\Omega, \end{aligned} \tag{3.67}$$

$$\begin{aligned} 0 = &\int_\Omega [\boldsymbol{\nabla} \cdot \left[P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\o}}_s(\boldsymbol{u}) \right]] \cdot \boldsymbol{u}^* \, d\Omega \\ &+ \int_\Omega \boldsymbol{\nabla} \cdot \left[\rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{\o}}(\boldsymbol{u}') - \dot{\boldsymbol{\o}}_T(T')] \right] \cdot \boldsymbol{u}^* \, d\Omega \\ &+ \int_\Omega [\boldsymbol{\nabla} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right]] T^* \, d\Omega + \int_\Omega \boldsymbol{\nabla} \cdot [k \boldsymbol{\nabla} T'] \, T^* \, d\Omega. \end{aligned} \tag{3.68}$$

Then applying integration by parts on (3.68) yields:

$$\begin{aligned} = &- \int_\Omega (P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\o}}_s(\boldsymbol{u})) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega \\ &- \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{\o}}(\boldsymbol{u}') - \dot{\boldsymbol{\o}}_T(T')] \right) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega \\ &+ \int_\Gamma \boldsymbol{n} \cdot ([P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\o}}_s(\boldsymbol{u})] \cdot \boldsymbol{u}^*) \\ &+ \boldsymbol{n} \cdot ([\rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{\o}}(\boldsymbol{u}') - \dot{\boldsymbol{\o}}_T(T')]] \cdot \boldsymbol{u}^*) \, d\Gamma - \int_\Omega (P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T) \cdot \boldsymbol{\nabla} T^* \, d\Omega \\ &- \int_\Omega (k \boldsymbol{\nabla} T') \cdot \boldsymbol{\nabla} T^* \, d\Omega \\ &+ \int_\Gamma \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T \right] \right] T^* \, d\Omega + \int_\Gamma \boldsymbol{n} \cdot [k \boldsymbol{\nabla} T'] \, T^* \, d\Omega, \end{aligned} \tag{3.69}$$

$$\begin{aligned} = &- \int_\Omega \left[(P\rho^{P-1} \boldsymbol{C_0} : \dot{\boldsymbol{\o}}_s(\boldsymbol{u})) : \dot{\boldsymbol{\o}}(\boldsymbol{u}^*) + (P\rho^{P-1} k_0 \boldsymbol{\nabla} T) \cdot \boldsymbol{\nabla} T^* \right] \rho' \, d\Omega \\ &- \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{\o}}(\boldsymbol{u}')] \right) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega + \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{\o}}_T(T')] \right) : \boldsymbol{\nabla} \boldsymbol{u}^* \, d\Omega \\ &- \int_\Omega (k \boldsymbol{\nabla} T') \cdot \boldsymbol{\nabla} T^* \, d\Omega \\ &+ \int_\Gamma [\boldsymbol{n} \cdot ([P\rho^{P-1}\rho' \boldsymbol{C_0} : \dot{\boldsymbol{\o}}_s(\boldsymbol{u})] + [\rho^P \boldsymbol{C_0} : [\dot{\boldsymbol{\o}}(\boldsymbol{u}') - \dot{\boldsymbol{\o}}_T(T')]])] \cdot \boldsymbol{u}^* \, d\Gamma \\ &+ \int_\Gamma \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla} T + k \boldsymbol{\nabla} T' \right] \right] T^* \, d\Gamma. \end{aligned} \tag{3.70}$$

Noticing the fifth integral contains a term from the perturbed system (3.64b), and the sixth integral contains terms from the perturbed system (3.64d)–(3.64g) gives:

$$\begin{aligned} 0 = &- \int_\Omega \left[(P\rho^{P-1}\boldsymbol{C_0} : \grave{\boldsymbol{o}}_s(\boldsymbol{u})) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) + (P\rho^{P-1}k_0\boldsymbol{\nabla}T)\cdot\boldsymbol{\nabla}T^* \right] \rho' \, d\Omega \\ &- \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}(\boldsymbol{u}^*)] \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}') \, d\Omega + \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}_T(T')] \right) : \boldsymbol{\nabla}\boldsymbol{u}^* \, d\Omega \\ &- \int_\Omega \boldsymbol{\nabla}T' \cdot (k\boldsymbol{\nabla}T^*) \, d\Omega \\ &+ \int_{\Gamma_2} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla}T \right] \right] d\Gamma_2 - \int_{\Gamma_3} T^* \left(4\gamma\epsilon T^3 T' \right) \, d\Gamma_3 \, . \end{aligned} \tag{3.71}$$

Performing integration by parts on the fourth integral gives:

$$\begin{aligned} 0 = &- \int_\Omega \left[(P\rho^{P-1}\boldsymbol{C_0} : \grave{\boldsymbol{o}}_s(\boldsymbol{u})) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) + (P\rho^{P-1}k_0\boldsymbol{\nabla}T)\cdot\boldsymbol{\nabla}T^* \right] \rho' \, d\Omega \\ &- \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}(\boldsymbol{u}^*)] \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}') \, d\Omega + \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}_T(T')] \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) \, d\Omega \\ &- \int_\Omega T' \left(\boldsymbol{\nabla} \cdot (k\boldsymbol{\nabla}T^*) \right) \, d\Omega - \int_\Gamma T' \left(\boldsymbol{n} \cdot (k\boldsymbol{\nabla}T^*) \right) \, d\Gamma \\ &+ \int_{\Gamma_2} T^* \left[\boldsymbol{n} \cdot \left[P\rho^{P-1}\rho' k_0 \boldsymbol{\nabla}T \right] \right] d\Gamma_2 - \int_{\Gamma_3} T^* \left(4\gamma\epsilon T^3 T' \right) \, d\Gamma_3 \, . \end{aligned} \tag{3.72}$$

Noting that the fourth, fifth and sixth integrals contain a term from the adjoint system (3.65) gives:

$$\begin{aligned} 0 = &- \int_\Omega \left[(P\rho^{P-1}\boldsymbol{C_0} : \grave{\boldsymbol{o}}_s(\boldsymbol{u})) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) + (P\rho^{P-1}k_0\boldsymbol{\nabla}T)\cdot\boldsymbol{\nabla}T^* \right] \rho' \, d\Omega \\ &- \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}(\boldsymbol{u}^*)] \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}') \, d\Omega \\ &+ \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}_T(T')] \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) \, d\Omega - \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}_T(T')] \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) \, d\Omega \\ &+ \int_{\Gamma_3} T^* \left(4\gamma\epsilon T^3 T' \right) \, d\Gamma_3 - \int_{\Gamma_3} T^* \left(4\gamma\epsilon T^3 T' \right) \, d\Gamma_3. \end{aligned} \tag{3.73}$$

$$\begin{aligned} 0 = &- \int_\Omega \left[(P\rho^{P-1}\boldsymbol{C_0} : \grave{\boldsymbol{o}}_s(\boldsymbol{u})) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) + (P\rho^{P-1}k_0\boldsymbol{\nabla}T)\cdot\boldsymbol{\nabla}T^* \right] \rho' \, d\Omega \\ &- \int_\Omega \left(\rho^P \boldsymbol{C_0} : [\grave{\boldsymbol{o}}(\boldsymbol{u}^*)] \right) : \grave{\boldsymbol{o}}(\boldsymbol{u}') \, d\Omega. \end{aligned} \tag{3.74}$$

Using the weak form of the adjoint system (3.66), turns it into the Riesz form (3.7):

$$\begin{aligned} C_s(\rho;\rho') &= \int_\Omega \boldsymbol{b}\cdot\boldsymbol{u}'\,d\Omega + \int_\Gamma \boldsymbol{t}\cdot\boldsymbol{u}'\,d\Gamma \\ &= -\int_\Omega \left[(P\rho^{P-1}\boldsymbol{C_0} : \grave{\boldsymbol{o}}_s(\boldsymbol{u})) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) + (P\rho^{P-1}k_0\boldsymbol{\nabla}T)\cdot\boldsymbol{\nabla}T^*\right]\rho'\,d\Omega, \end{aligned} \tag{3.75}$$

and therefore, we are able to identify the expression of the gradient of our objective functional as:

$$\boxed{\boldsymbol{\nabla}C_s(\rho) = -(P\rho^{P-1}\boldsymbol{C_0} : \grave{\boldsymbol{o}}_s(\boldsymbol{u})) : \grave{\boldsymbol{o}}(\boldsymbol{u}^*) - (P\rho^{P-1}k_0\boldsymbol{\nabla}T)\cdot\boldsymbol{\nabla}T^*.} \tag{3.76}$$

3.4 Derivations in the Discrete Setting

3.4.1 Uncoupled Cases

We will now begin the derivations for the objective functionals in the discrete setting. We will start with an objective functional of the von Mises stress (2.30) and the derivation is subject to (2.23). The *Gâteaux* derivative of the objective functional (2.30) is:

$$\begin{aligned} \sigma_{\mathbf{vm}}{}'(\boldsymbol{\rho};\boldsymbol{\rho}') =& (\boldsymbol{u}'(\boldsymbol{\rho};\boldsymbol{\rho}'))^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{u} \\ &+ \boldsymbol{u}^T\boldsymbol{B}^T p\left[diag\left(\rho_i^{p-1}(\boldsymbol{C}_0)_i\boldsymbol{\rho}_i'\right)\right]^T\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{u} \\ &+ \boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}p\left[diag\left(\rho_i^{p-1}(\boldsymbol{C}_0)_i\boldsymbol{\rho}_i'\right)\right]\boldsymbol{B}\boldsymbol{u} \\ &+ \boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{u}'(\boldsymbol{\rho};\boldsymbol{\rho}')\ . \end{aligned} \tag{3.77}$$

The *Gâteaux* derivative of the governing system (2.23) is:

$$\boldsymbol{K}\boldsymbol{u}' + \boldsymbol{K}'(\boldsymbol{\rho};\boldsymbol{\rho}')\boldsymbol{u} = \boldsymbol{K}\boldsymbol{u}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}' = \boldsymbol{0}, \tag{3.78}$$

whose solution is given by the perturbation variable $\boldsymbol{u}'(\boldsymbol{\rho};\boldsymbol{\rho}')$. We will define an adjoint system in the form:

$$\boldsymbol{K}\boldsymbol{u}^* = 2\left[\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{u}\right]. \tag{3.79}$$

Taking the inner product between the perturbed system (3.77) and the adjoint variable $\boldsymbol{u}^*$, which in the discrete setting is just vector-vector multiplication, yields:

$$\boldsymbol{0} = (\boldsymbol{u}^*)^T(\boldsymbol{K}\boldsymbol{u}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}')\ . \tag{3.80}$$

Subtracting the directional derivative (3.77) from both sides and multiplying both sides by -1 gives:

$$\begin{aligned}\sigma_{\mathbf{vm}}{}'(\boldsymbol{\rho};\boldsymbol{\rho}') =&(\boldsymbol{u}')^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{u}\\ &+\boldsymbol{u}^T\boldsymbol{B}^Tp\left[diag\left(\rho_i^{p-1}(\boldsymbol{C}_0)_i\boldsymbol{u}_i\right)\right]^T\boldsymbol{\rho}'\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{\rho}'\\ &+\boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}p\left[diag\left(\rho_i^{p-1}(\boldsymbol{C}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{B}\boldsymbol{\rho}'\\ &+\boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{u}'\\ &-(\boldsymbol{u}^*)^T(\boldsymbol{K}\boldsymbol{u}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}')\ .\end{aligned} \tag{3.81}$$

Rearranging the right-hand side gives:

$$\begin{aligned}\sigma_{\mathbf{vm}}{}'(\boldsymbol{\rho};\boldsymbol{\rho}') =&[\boldsymbol{u}^T\boldsymbol{B}^Tp\left[diag\left(\rho_i^{p-1}(\boldsymbol{C}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\\ &+\boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}p\left[diag\left(\rho_i^{p-1}(\boldsymbol{C}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{B}\\ &-(\boldsymbol{u}^*)^Tz\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]]\boldsymbol{\rho}'\\ &+\boldsymbol{u}'\left[2\left[\boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\right]-(\boldsymbol{u}^*)^T\boldsymbol{K}\right]\ .\end{aligned} \tag{3.82}$$

With the judicious choice of the adjoint system (3.79), the vector multiplying $\boldsymbol{u}'$ vanishes, and then the right-hand side is in the Riesz form (3.7):

$$\begin{aligned}\sigma_{\mathbf{vm}}{}'(\boldsymbol{\rho};\boldsymbol{\rho}') =&[2\boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{C}^T\boldsymbol{V}p\left[diag\left(\rho_i^{p-1}(\boldsymbol{C}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{B}\\ &-(\boldsymbol{u}^*)^T z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]]\boldsymbol{\rho}'.\end{aligned} \tag{3.83}$$

We are therefore able to identify the expression of the gradient of our objective functional (2.30) as:

$$\boxed{\boldsymbol{\nabla}\sigma_{vm}^2 = 2\boldsymbol{B}^T p\left[diag\left(\rho_i^{p-1}(\boldsymbol{u}_i)^T(\boldsymbol{C}_0)_i\right)\right]\boldsymbol{V}\boldsymbol{C}\boldsymbol{B}\boldsymbol{u} - z\left[diag\left(\rho_i^{z-1}(\boldsymbol{u}_i)^T(\boldsymbol{K}_0)_i\right)\right]\boldsymbol{u}^*.} \tag{3.84}$$

Our second derivation will be that for static compliance (2.28) and will once again be subject to (2.23). The *Gâteaux* derivative of the objective functional (2.28) is:

$$J_d'(\boldsymbol{\rho};\boldsymbol{\rho}') = \boldsymbol{f}^T\boldsymbol{u}'(\boldsymbol{\rho};\boldsymbol{\rho}'), \tag{3.85}$$

and the *Gâteaux* derivative of the governing system (2.23) is:

$$\boldsymbol{K}\boldsymbol{u}' + \boldsymbol{K}'(\boldsymbol{\rho};\boldsymbol{\rho}')\boldsymbol{u} = \boldsymbol{K}\boldsymbol{u}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}' = \boldsymbol{0}\ , \tag{3.86}$$

whose solution is given by the perturbation variable $\boldsymbol{u}'(\boldsymbol{\rho};\boldsymbol{\rho}')$. We will define an adjoint system in the form:

$$\boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{u}^* = \boldsymbol{f}, \tag{3.87}$$

whose solution is given by the adjoint variable $\boldsymbol{u}^*$. Taking the inner product between the perturbed system (3.86) and the adjoint variable $\boldsymbol{u}^*$ gives:

$$\boldsymbol{0} = (\boldsymbol{u}^*)^T(\boldsymbol{K}\boldsymbol{u}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}')\ . \tag{3.88}$$

Subtracting $\boldsymbol{f}^T\boldsymbol{u}'$ from both sides and multiplying both sides by -1 gives:

$$\boldsymbol{f}^T\boldsymbol{u}' = -(\boldsymbol{u}^*)^T(\boldsymbol{K}\boldsymbol{u}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}') + \boldsymbol{f}^T\boldsymbol{u}' \ . \tag{3.89}$$

Note that that left hand side is the *Gâteaux* derivative of the objective function (3.85), and therefore, after rearranging the right hand side:

$$J_d'(\boldsymbol{\rho};\boldsymbol{\rho}') = -(\boldsymbol{u}^*)^T z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}' + \boldsymbol{u}'\left[\boldsymbol{f}^T - (\boldsymbol{u}^*)^T\boldsymbol{K}\right] \ . \tag{3.90}$$

With the judicious choice of the adjoint system (3.87), the vector multiplying $\boldsymbol{u}'$ vanishes and what remains is an expression in the Riesz form (3.7):

$$J_d'(\boldsymbol{\rho};\boldsymbol{\rho}') = -(\boldsymbol{u}^*)^T z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}' = (\boldsymbol{\nabla}C_s)^T\boldsymbol{\rho}'. \tag{3.91}$$

We are therefore able to identify the expression of the gradient of our objective functional (2.28) as:

$$\boxed{\boldsymbol{\nabla}J_d(\boldsymbol{\rho}) = -z\left[diag\left(\rho_i^{z-1}(\boldsymbol{u}_i)^T(\boldsymbol{K}_0)_i\right)\right]\boldsymbol{u}^*.} \tag{3.92}$$

The third derivation in the discrete setting will have the objective functional of thermal compliance (2.29), and will be subject to the governing system (2.26). The *Gâteaux* derivative of the objective functional (2.29) is:

$$J_d'(\boldsymbol{\rho};\boldsymbol{\rho}') = \boldsymbol{p}^T\boldsymbol{T}'(\boldsymbol{\rho};\boldsymbol{\rho}'), \tag{3.93}$$

and the *Gâteaux* derivative of the governing system (2.26) is:

$$\boldsymbol{K}\boldsymbol{T}' + \boldsymbol{K}'(\rho;\rho')\boldsymbol{T} = \boldsymbol{K}\boldsymbol{T}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{T}_i\right)\right]\boldsymbol{\rho}' = \boldsymbol{0}, \tag{3.94}$$

whose solution is given by the perturbation variable $\boldsymbol{T}'(\boldsymbol{\rho};\boldsymbol{\rho}')$. We will define an adjoint system in the form:

$$\boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{T}^* = \boldsymbol{p}, \tag{3.95}$$

whose solution is given by the adjoint variable $\boldsymbol{T}^*$. Taking the inner product between the perturbed system (3.94) and the adjoint variable $\boldsymbol{T}^*$ gives:

$$\mathbf{0} = (\boldsymbol{T}^*)^T(\boldsymbol{K}\boldsymbol{T}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{T}_i\right)\right]\boldsymbol{\rho}') \ . \tag{3.96}$$

Subtracting $\boldsymbol{p}^T\boldsymbol{T}'$ from both sides and multiplying both sides by -1 gives:

$$\boldsymbol{p}^T\boldsymbol{T}' = -(\boldsymbol{T}^*)^T(\boldsymbol{K}\boldsymbol{T}' + z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{T}_i\right)\right]\boldsymbol{\rho}') + \boldsymbol{p}^T\boldsymbol{T}'. \tag{3.97}$$

Note that that left hand side is the *Gâteaux* derivative of the objective functional (3.93), and therefore, after rearranging the right hand side:

$$J_d'(\boldsymbol{\rho};\boldsymbol{\rho}') = -(\boldsymbol{T}^*)^T z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{T}_i\right)\right]\boldsymbol{\rho}' + \boldsymbol{T}'\left[\boldsymbol{p}^T - (\boldsymbol{T}^*)^T\boldsymbol{K}\right] \ . \tag{3.98}$$

With the judicious choice of the adjoint system (3.95), the vector multiplying $\boldsymbol{T}'$ vanishes and what remains is an expression in the Riesz form (3.7):

$$J_d'(\boldsymbol{\rho};\boldsymbol{\rho}') = -(\boldsymbol{T}^*)^T z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{T}_i\right)\right]\boldsymbol{\rho}' = (\boldsymbol{\nabla}J_d)^T\rho'. \tag{3.99}$$

We are therefore able to identify the expression for the gradient of our objective functional (2.29) as:

$$\boxed{\boldsymbol{\nabla}J_d(\boldsymbol{\rho}) = -z\left[diag\left(\rho_i^{z-1}(\boldsymbol{T}_i)^T(\boldsymbol{K}_0)_i\right)\right]\boldsymbol{T}^*.} \tag{3.100}$$

3.4.2 Coupled Case

The last derivation will be the coupled system in the discrete setting, where the objective functional will be the combined structural and thermal compliance, c.f (2.28,(2.29)). The derivation will be subject to (2.27). The *Gâteaux* derivative of the objective functional (2.29) and (2.28) is:

$$J'(\boldsymbol{\rho};\boldsymbol{\rho}') = \boldsymbol{f}^T\boldsymbol{u}'(\boldsymbol{\rho};\boldsymbol{\rho}') + \boldsymbol{P}^T\boldsymbol{T}'(\boldsymbol{\rho};\boldsymbol{\rho}'), \tag{3.101}$$

and the *Gâteaux* derivative of the governing system (2.27) is:

$$\begin{bmatrix} \boldsymbol{C} & -\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}' \\ \boldsymbol{T}' \end{bmatrix} = \begin{bmatrix} -z\left[diag\left(\rho_i^{z-1}\left(\boldsymbol{C}_0\right)_i\boldsymbol{u}_i - \left(\boldsymbol{B}_0\right)_i(\delta\boldsymbol{T})_i\right)\right] \\ -z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right] \end{bmatrix} \boldsymbol{\rho}', \tag{3.102}$$

whose solution is given by the perturbation vectors $\boldsymbol{u}'(\boldsymbol{\rho};\boldsymbol{\rho}')$ and $\boldsymbol{T}'(\boldsymbol{\rho};\boldsymbol{\rho}')$. We will also define an adjoint system in the form:

$$\begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \\ -\boldsymbol{B} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^* \\ \boldsymbol{T}^* \end{bmatrix} = \begin{bmatrix} -\boldsymbol{f} \\ -\boldsymbol{P} \end{bmatrix}. \tag{3.103}$$

Taking the inner product between the perturbed system (3.102) and the adjoint vector gives:

$$\begin{aligned} \boldsymbol{0} = &\begin{bmatrix} (\boldsymbol{u}^*)^T & (\boldsymbol{T}^*)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{C} & -\boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{K} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}' \\ \boldsymbol{T}' \end{bmatrix} \\ &+ \begin{bmatrix} z\left[diag\left(\rho_i^{z-1}\left(\boldsymbol{C}_0\right)_i\boldsymbol{u}_i - \left(\boldsymbol{B}_0\right)_i(\delta\boldsymbol{T})_i\right)\right] \\ z\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right] \end{bmatrix} \boldsymbol{\rho}', \end{aligned} \tag{3.104}$$

$$\begin{aligned} 0 = &(\boldsymbol{u}^*)^T\boldsymbol{C}\boldsymbol{u}' - (\boldsymbol{u}^*)^T\boldsymbol{B}\boldsymbol{T}' + (\boldsymbol{T}^*)^T\boldsymbol{K}\boldsymbol{T}' \\ &+ z(\boldsymbol{u}^*)^T\left[diag\left(\rho_i^{z-1}\left(\boldsymbol{C}_0\right)_i\boldsymbol{u}_i - \left(\boldsymbol{B}_0\right)_i(\delta\boldsymbol{T})_i\right)\right] \\ &+ z(\boldsymbol{T}^*)^T\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}'. \end{aligned} \tag{3.105}$$

Subtracting $\boldsymbol{f}^T\boldsymbol{u}' + \boldsymbol{P}^T\boldsymbol{T}'$ from both sides and multiplying both sides by -1 gives:

$$\begin{aligned}\boldsymbol{f}^T\boldsymbol{u}' + \boldsymbol{P}^T\boldsymbol{T}' =& -(\boldsymbol{u}^*)^T\boldsymbol{C}\boldsymbol{u}' + (\boldsymbol{u}^*)^T\boldsymbol{B}\boldsymbol{T}' - (\boldsymbol{T}^*)^T\boldsymbol{K}\boldsymbol{T}' \\ &- z(\boldsymbol{u}^*)^T\left[diag\left(\rho_i^{z-1}\left(\boldsymbol{C}_0\right)_i\boldsymbol{u}_i - (\boldsymbol{B}_0)_i(\delta\boldsymbol{T})_i\right)\right] \\ &- z(\boldsymbol{T}^*)^T\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\boldsymbol{\rho}' \\ &+ \boldsymbol{f}^T\boldsymbol{u}' + \boldsymbol{P}^T\boldsymbol{T}'.\end{aligned} \tag{3.106}$$

Note that that left hand side is the *Gâteaux* derivative of the objective functional (3.102), and therefore, after rearranging the right hand side, by collecting the terms proportional to $\boldsymbol{u}'$ and $\boldsymbol{T}'$ gives:

$$\begin{aligned}J'(\boldsymbol{\rho};\boldsymbol{\rho}') =& \left[-(\boldsymbol{u}^*)^T\boldsymbol{C} + \boldsymbol{f}^T\right]\boldsymbol{u}' + \left[(\boldsymbol{u}^*)^T\boldsymbol{B} - (\boldsymbol{T}^*)^T\boldsymbol{K} + \boldsymbol{P}^T\right]\boldsymbol{T}' \\ &- \left[z(\boldsymbol{u}^*)^T\left[diag\left(\rho_i^{z-1}\left(\boldsymbol{C}_0\right)_i\boldsymbol{u}_i - (\boldsymbol{B}_0)_i(\delta\boldsymbol{T})_i\right)\right)\right] \\ &- z(\boldsymbol{T}^*)^T\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]]\boldsymbol{\rho}'.\end{aligned} \tag{3.107}$$

With the judicious choice of the adjoint system (3.103), the vector multiplying $\boldsymbol{u}'$ and $\boldsymbol{T}'$ vanish and what remains is an expression in the Riesz form (3.7):

$$\begin{aligned}J'(\boldsymbol{\rho};\boldsymbol{\rho}') =& [-z(\boldsymbol{u}^*)^T\left[diag\left(\rho_i^{z-1}\left(\boldsymbol{C}_0\right)_i\boldsymbol{u}_i - (\boldsymbol{B}_0)_i(\delta\boldsymbol{T})_i\right)\right] \\ &- z(\boldsymbol{T}^*)^T\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]]\boldsymbol{\rho}',\end{aligned} \tag{3.108}$$

and we are therefore able to identify the expression for the gradient of the objective functional (2.29,2.28) as:

$$\begin{aligned}\boldsymbol{\nabla}J(\boldsymbol{\rho}) =& \left(-z(\boldsymbol{u}^*)^T\left[diag\left(\rho_i^{z-1}\left(\boldsymbol{C}_0\right)_i\boldsymbol{u}_i - (\boldsymbol{B}_0)_i(\delta\boldsymbol{T})_i\right)\right]\right)^T \\ &\left(-z(\boldsymbol{T}^*)^T\left[diag\left(\rho_i^{z-1}(\boldsymbol{K}_0)_i\boldsymbol{u}_i\right)\right]\right)^T.\end{aligned} \tag{3.109}$$

3.5 Computational Algorithm

One of the main components in the algorithm is dealing with the mass constraint (3.3b) in the optimization problem. We will do this by first defining a set that contains all distributions that meet our mass constraint called the "target manifold", and it will be defined as:

$$\mathcal{M} = \left\{ \eta \in L^2(\Omega) : \int_\Omega \rho(\eta)\, d\Omega = M \right\}. \tag{3.110}$$

Now if our initial distribution is an element of our target manifold (3.110) then we will be able to keep our mass distributions close to the manifold by only travelling in the tangent subspace to the target manifold, which we will define as:

$$T_\eta \mathcal{M} = \left\{ \eta' \in L^2(\Omega) : \int_\Omega \frac{d\rho}{d\eta} \eta'\, d\Omega = 0 \right\}. \tag{3.111}$$

Now since we are using a gradient-based optimization method, the direction we will be travelling in will be in the direction of the gradient of the objective functional (3.4) at each step. The issue with this is that we cannot guarantee that the gradient of the objective functional is a direction that is in the tangent subspace (3.111). To make sure that we end up in the tangent subspace, we will project the gradient onto the tangent subspace (3.111). We will use the direction $\frac{d\rho(\eta)}{d\eta}$ to orthogonally project onto the tangent subspace as (3.111) implies that $\frac{d\rho(\eta)}{d\eta}$ spans the space orthogonal to $T_\eta \mathcal{M}$, whose dimension is 1. We are then able to define the projection of the gradient of the objective function (3.4) onto the tangent subspace (3.111) as:

$$P \boldsymbol{\nabla} J = \boldsymbol{\nabla} J - c \frac{d\rho(\eta)}{d\eta}, \tag{3.112}$$

where $P : L^2 \to T_\eta \mathcal{M}$ is the projection operator and c is a suitably chosen constant. To guarantee that this projection lands us in the tangent subspace (3.111), we must make

sure the projection meets the following condition:

$$\int_{\Omega} \frac{d\rho(\eta_0)}{d\eta} \left(\boldsymbol{\nabla} J - c \frac{d\rho(\eta_0)}{d\eta} \right) d\Omega = 0, \tag{3.113}$$

and after solving for c we end up with:

$$c = \frac{\int_{\Omega} \frac{d\rho(\eta_0)}{d\eta} \boldsymbol{\nabla} J \, d\Omega}{\int_{\Omega} \left(\frac{d\rho(\eta_0)}{d\eta} \right)^2 d\Omega}. \tag{3.114}$$

So after the projection onto the tangent subspace (3.111), our new gradient is:

$$\boldsymbol{\nabla}_{T_\eta \mathcal{M}} J_s = \boldsymbol{\nabla}_{\rho(\eta)} J_s - \left(\frac{\int_{\Omega} \frac{d\rho(\eta_0)}{d\eta} \boldsymbol{\nabla}_{\rho(\eta)} J \, d\Omega}{\int_{\Omega} \left(\frac{d\rho(\eta_0)}{d\eta} \right)^2 d\Omega} \right) \frac{d\rho(\eta_0)}{d\eta}, \tag{3.115}$$

where $\boldsymbol{\nabla}_{\rho(\eta)} J_s$ is (3.4).

Since the change of variables is non-linear, we cannot ensure that any arbitrarily sized step in the tangent subspace (3.111) will remain in the target manifold (3.110), therefore adding the gradient of the objective functional (3.4) to the current mass distribution may cause us to "drift" from the target manifold. Now since staying on the target manifold (3.110) is a condition in the optimization problem (3.2) we must address this issue. We will do this by defining a retraction function, $R : T_\eta \mathcal{M} \to \mathcal{M}$, in two alternative ways. The first way is using an "additive" or shifting method defined $\forall \eta \in T_\eta \mathcal{M}$ as:

$$R_1 \eta = \eta + \beta, \tag{3.116}$$

where β is chosen to satisfy the condition:

$$\int_{\Omega} \rho(\eta + \beta) \, d\Omega = M. \tag{3.117}$$

The second way we can define the retraction function is using a "multiplicative" or stretching method so that $\forall \eta \in T_\eta \mathcal{M}$:

$$R_2 \eta = \eta \gamma, \tag{3.118}$$

where γ is chosen to satisfy the condition:

$$\int_\Omega \rho(\eta\gamma)\, d\Omega = M. \tag{3.119}$$

Now since the conditions placed on the retraction function (3.117, 3.119) are continuous, we are able to use *MATLAB*'s `fsolve` function (MATLAB 2019), or any root finding algorithm, to determine the value of β or γ that will define our retraction functions (3.116,3.118) and therefore eliminate the "drift" found in the subspace projection. In our computational experiment, the retraction function defined as (3.119) reduces the magnitude of the gradient so that change in the objective functional is smaller than any reasonable stop condition and therefore we have used (3.117).

The last main component of the algorithm is called line minimization. We have previously discussed how the gradient gives the direction of the steepest decent and so line minimization is the process of finding how far one should go in this direction. This is done mathematically by solving the minimization problem:

$$\tau_k = \arg\min_{\tau > 0} h(\tau), \tag{3.120}$$

where in our case, we define the function $h : \mathbb{R} \to \mathbb{R}$ as:

$$h(\tau) = J_s(\rho(\eta_k + \tau \boldsymbol{\nabla}_{T_\eta M} J_s)), \tag{3.121}$$

where k is the current step of the algorithm and $\tau \in \mathbb{R}$ is the step size. We then solve

(3.120) by looking for a sufficient decrease in h's value (3.121). This is done by testing values of τ in a given range and minimizing to within a certain tolerance. In *MATLAB* (MATLAB 2019), this can be done using the `fminbnd` function, which will find the minimum value of $h(\tau)$ for $\tau_1 < \tau < \tau_2$. To make this process as efficient as possible a suitably small range of (τ_1, τ_2) is chosen using a heuristic method of shrinking or expanding the brackets by testing values of the objective functionals at τ_2, and comparing them to a suitably chosen value. After all of these steps have been implemented, the algorithm is shown in Algorithm 2.

Algorithm 2 Implementation of the Sobolev Gradient Algorithm

Input:
η_0 — Initial guess for mass distribution
$\boldsymbol{C}$ — Stiffness Tensor
$\boldsymbol{F}$ — Force Vector

Output:
$J(\rho(\eta))$ — Structural Compliance

$\rho^{(0)} \leftarrow \rho(\eta_0)$(Initial guess)
$n \leftarrow 1$
repeat
Solve $\boldsymbol{C}(\rho^{(n)})\boldsymbol{u} = \boldsymbol{F}$ for $\boldsymbol{u}$
Evaluate Gradient $\boldsymbol{\nabla}_\rho J_s(\boldsymbol{u})$
Convert to Sobolev Gradient $\boldsymbol{\nabla}_{H_1} J_s \leftarrow \boldsymbol{\nabla}_\rho J_s$
Compute the Subspace Projection $\boldsymbol{\nabla}_{T_\eta M} J_s$
Perform Line Minimization $\tau^{(n)} \leftarrow min_{\tau>0}\, h(\tau) = J_s(\rho(\eta_k^{(n-1)} + \tau \boldsymbol{\nabla}_{T_\eta M} J_s))$
Perform Retraction: $\beta^{(n)} \leftarrow$ given by solution of $R(\eta^{(n-1)}, \beta) = 0$
Update: $\rho^{(n)} \leftarrow \rho(\eta^{(n-1)} + \tau^{(n)} \boldsymbol{\nabla}_{T_\eta M} J_s)$
Evaluate $J(\rho^{(n)})$
$n \leftarrow n + 1$
until $\frac{J(\rho^{(n-1)}) - J(\rho^{(n)})}{J(\rho^{(n-1)})} < 10^{-3}$

Chapter 4

Computational Results

4.1 Validation of Derivation

We will now verify the validity of the derived expressions for the gradients, in both the discrete and continuous settings, by performing a Kappa test on them. The Kappa test is a method for determining how well two different gradient expressions agree with each other. In our case we will be comparing how well the gradient expression that we derived using the Riesz representation theorem matches up with the *Gâteaux* derivative calculated by definition. i.e, by comparing:

$$\lim_{\epsilon \to 0} \frac{J(\rho + \epsilon) - J(\rho)}{\epsilon} \quad \text{vs.} \quad \int_{\Omega} \boldsymbol{\nabla} J(\rho) \rho' \, d\Omega. \tag{4.1}$$

This comparison is done by simply taking the ratio of the two terms. We also test it over a non-uniform perturbation to make sure the values match up for different values of ρ. Lastly, because we numerically cannot take the limit as ϵ goes to zero, we must approximate it by taking distinct values for epsilon. The test is then performed by calculating $\kappa(\epsilon)$ at different values of ϵ and noticing the patterns. So in our case the

equation will be:

$$\kappa(\epsilon) = \frac{J(\rho + \epsilon\rho'(s)) - J(\rho)}{\epsilon \int_\Omega \boldsymbol{\nabla} J(\rho)\rho'(s)\, d\Omega}, \tag{4.2}$$

where ϵ will range from 10^{-1} to 10^{-14}. For the test to be successful, we should notice that the value of $\kappa(\epsilon)$ should be close to one for small values of ϵ. Note that because we are solving this value numerically there will be two types of errors that will affect the value of κ in three different ways. The first type of error is called round-off errors, which occur due to the fact that the computer only uses a finite amount of digits for its calculations. This should cause the value of $\kappa(\epsilon)$ to increase for smaller values of ϵ. The second type of error is called truncation errors, and they occur for two different reasons. The first reason is that ϵ is not close enough to 0, and therefore does not approximate the error very well. The second reason truncation errors occur is that the mesh is too coarse to solve the PDE correctly.

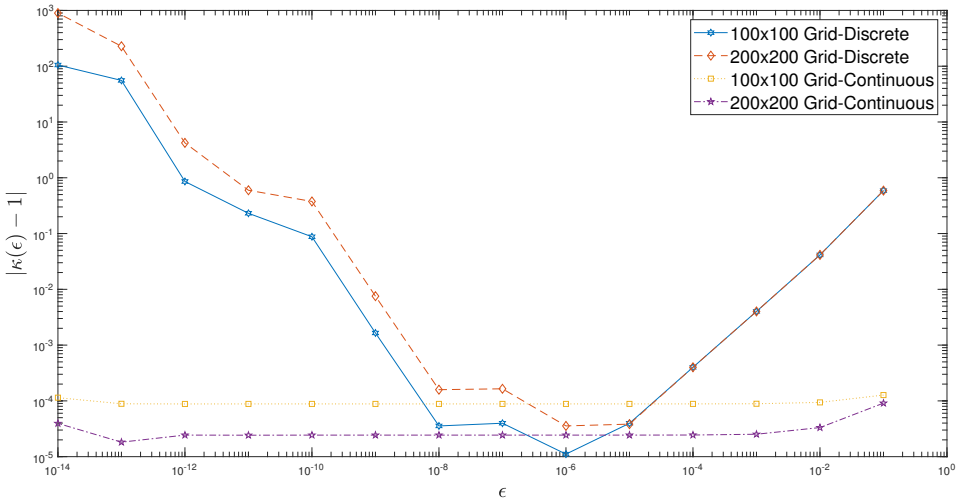


Figure 4.1: Graph of $|\kappa(\epsilon) - 1|$ vs. ϵ for the two Gradients at different grid sizes

The expression for the gradient of the objective function in the discrete setting (3.92) and in the continuous setting (3.4) have been used in the kappa test and the results were

then plotted in figure (4.1). As we can see from figure (4.1), all three of the errors are present in the calculation for the gradient of the objective functional in the continuous setting (3.4). Round-off errors occur for $\epsilon \leq 10^{-13}$ and for values smaller than that, the value of the error increases. While the presence of truncation errors can be seen by the error for $\epsilon > 10^{-3}$ and by the fact that when the grid size was doubled, the difference between the two trials in the continuous setting was cut in half, as seen by the smaller values of κ for the 200×200 grid. For the gradient of the objective functional in the discrete setting (3.92), figure (4.1) shows that two types of errors were present in the calculations. The main difference between the gradient derived in the continuous setting and the one derived in the discrete setting is that the one in the discrete setting seems to be much more sensitive to round-off errors, as when $\epsilon \leq 10^{-7}$, the error starts to increase, as opposed to 10^{-13} in the continuous setting. The second main difference between the results for the two gradients is the shape of their plots, as for the discrete setting it is a very clear v shape, while for the continuous setting it is mostly a flat line that increases at the boundaries. This is due to truncation errors from having too coarse a mesh which "cuts-off" the v-shape of the continuous gradient's results as it is the dominating error at this truncation level. This error is not present for the gradient in the discrete setting as it is solved using the weak form of the PDE whose solution does not depend on the size of the mesh. Lastly, based off of this graph. we can conclude that we have successfully derived the gradient, in the continuous setting using the Riesz representation theorem, as the difference between the *Gâteaux* derivative and the expression of the gradient is in the realm of 10^{-5}.

4.2 Optimization Results

We used four different algorithms in order to compare the implementation of the different ideas mentioned in Chapter 3. The first algorithm that was tested was the standard

99-line code by Sigmund (Sigmund 2001). This algorithm was used as it is a standard algorithm in topology optimization and therefore gives a good benchmark for comparison. The algorithm will be referred to by `Top99` discrete and the amount of smoothing in the algorithm is given by the Rmin value. The smoothing of the gradient of the objective function in `Top99` discrete uses a heuristic method of averaging out the values of the elements within a certain radius. The next set of trials that were run we called `Top99` continuous, and it is where we changed `Top99` discrete to use the derived gradient of the objective functional in the continuous setting (3.4). We then ran trials in an algorithm we called continuous ℓ^2 gradient which had the same form of Algorithm 2, except the step that converts the gradient into the Sobolev space was skipped. When the conversion into the Sobolev space is not skipped, and therefore Algorithm 2 is in full effect, we called those trials "continuous Sobolev gradient". In these trials, the amount of smoothing of the gradient of the objective functional is given by the ℓ^2 value. Table (4.1) contains information about the different trials that were run, and the value of the different parts of the objective functional at the conclusion of the algorithms. Found after table (4.1) are figures containing the final topologies for each of the trials, as well as the objective function at each iteration of the algorithm.

As we can see from (4.2a), the final mass distribution suffers from the checkerboard issue mentioned in Chapter 3. This is due to having the Rmin value less than one, as this implies that there is no smoothing on the gradient of the objective functional (3.92). As we can see from figure (4.2b), simply changing the gradient of the objective functional to be the one derived in the continuous setting (3.4) does not fix the issue of checker-boarding. The main difference between the two distributions can be found in the middle of the structures where figure (4.2a) is all checker-boarded, while figure (4.2b) has connected beams. As we can see from figures (4.2c) and (4.2d), increasing the value of Rmin, and therefore applying some smoothing to the gradients, has fixed the checker-boarding issue from figures (4.2a) and (4.2b). The main difference between

Table 4.1: Values of Key Parameters and Results. Note J_p is calculated using $\alpha = 1$ in order to be able to compare values

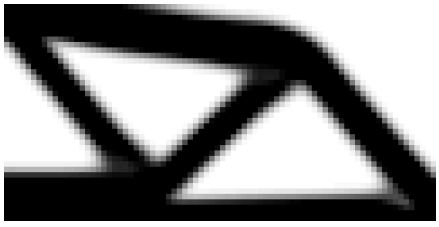
Gradient Type	α	ℓ^2/Rmin	J_s	J_0	J_p	Iterations
Top99 Discrete	N/A	0.5	84.3908	83.8749	0.5159	43
	N/A	1.5	103.0256	80.0115	23.0141	430
	N/A	2.5	108.3188	83.4568	24.8621	80
Top99 Continuous	N/A	0.5	84.2736	84.0004	0.2732	46
	N/A	1.5	102.7021	80.1697	22.5324	225
	N/A	2.5	108.3751	83.4777	24.8975	80
Continuous ℓ^2 Gradient	0	N/A	84.4289	83.4986	0.9303	19
	0.5	N/A	86.1054	85.9875	0.2357	11
	1	N/A	93.9628	93.8512	0.1116	10
Continuous Sobolev Gradient	0	0.5	92.1622	91.4057	0.7565	22
	0	1.5	90.7574	89.5230	1.2344	27
	0	2.5	89.9616	88.6364	1.3252	28
	0	4.5	90.7798	90.1155	0.6644	40
	$\frac{1}{16}$	0.5	91.1860	91.008	0.6808	16
	$\frac{1}{16}$	1.5	90.0616	89.8733	1.5063	22
	$\frac{1}{16}$	2.5	94.2743	94.1421	1.0576	21
	$\frac{1}{16}$	4.5	94.9084	94.7202	1.5056	24
	1	0.5	93.4479	93.1210	0.3269	14
	1	1.5	94.2602	93.6265	0.6337	12
	1	2.5	96.3020	95.8281	0.4739	14
	1	4.5	103.0734	102.2983	0.7751	14

(a) Discrete Setting, Rmin = 0.5

(b) Continuous Setting, Rmin = 0.5

(c) Discrete Setting, Rmin = 1.5

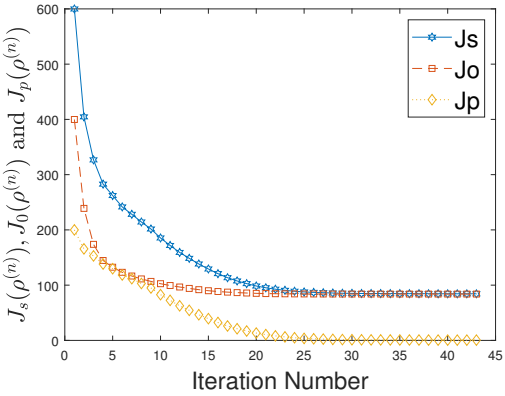
(d) Continuous Setting, Rmin = 1.5



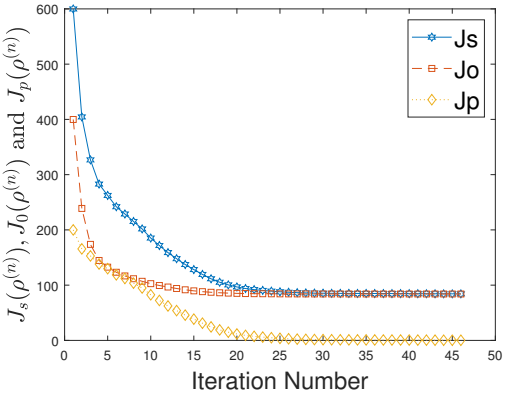
(e) Discrete Setting, Rmin = 2.5

(f) Continuous Setting, Rmin = 2.5

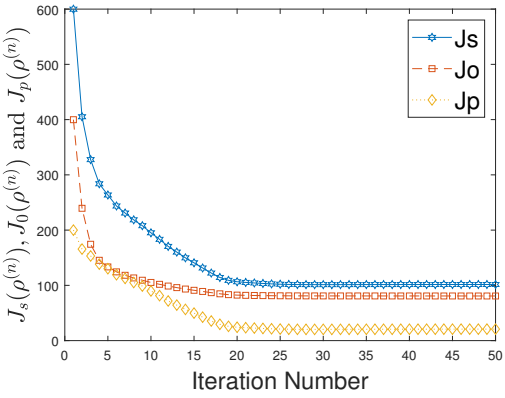
Figure 4.2: Final Mass Distributions of Top99 Algorithms



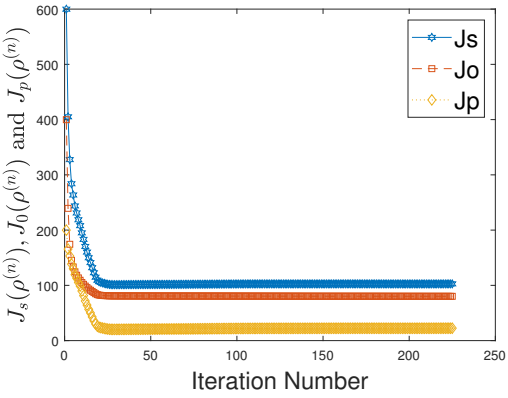
(A) Discrete Setting, Rmin = 0.5



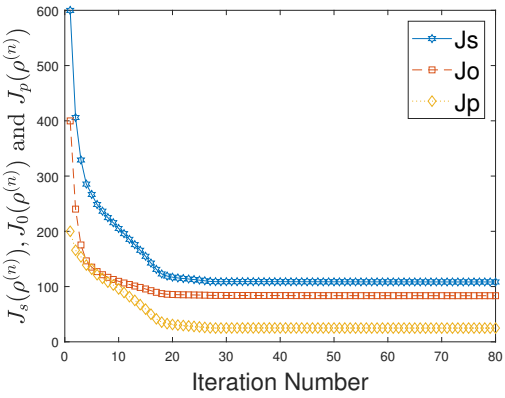
(B) Continuous Setting, Rmin = 0.5



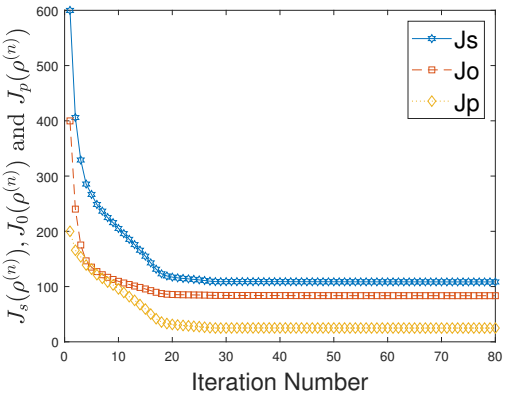
(C) Discrete Setting, Rmin = 1.5



(D) Continuous Setting, Rmin = 1.5



(E) Discrete Setting, Rmin = 2.5



(F) Continuous Setting, Rmin = 2.5

Figure 4.3: Objective Functional Vs. Iteration of Top99 Algorithms

figure (4.2d) and figure (4.2c) is that figure (4.2d) does not have the hole in the very centre of the domain as those two supports have been merged into one. As we can see from figures (4.2e) and (4.2f), increasing the Rmin value to 2.5 has had the effect of smoothing the inner supports together from figures (4.2c) and (4.2d) into just having two support beams instead of five. At this amount of smoothing, figures (4.2e) and (4.2f) have become indistinguishable from each other.

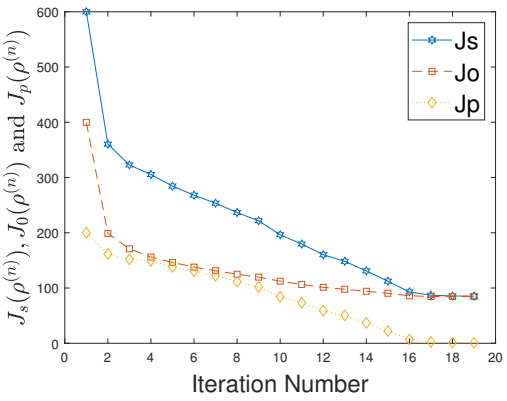
In the next set of trials, we used the continuous ℓ^2 gradient algorithm and tested different values for α, which is the constant in front of our 0-1 penalization term in the objective functional (3.1). Note for these three trials, we do not use the Sobolev gradients and therefore there is no smoothing on the gradient. The mass distributions can then be found in figure (4.4), while the value of the objective function at each step can be found in figure (4.5). From figure (4.4a), we can see that it resembles the distributions in figures (4.2c) and (4.2d), with the main difference being the checker-boarding distribution in the top-right corner of the domain. The mass distribution in figure (4.4b) resembles the distributions found in figures (4.2e) and (4.2f), with the main difference being the extra support beam in the bottom-right of the domain. The main issue with this distribution is that there are regions in the domain where there is mass that is not connected to anything, or just sticking out of the support beams and therefore should not affect the structural compliance values (2.16). When α is increased to 1, this issue is more prominent as can be seen in figure (4.4c). The main idea to solve this issue is by smoothing the gradient by projecting it into the Sobolev space, which is the only difference in the continuous Sobolev gradient algorithm (2).

In the continuous Sobolev gradient trials we have two parameters in α and ℓ^2 to experiment with. The mass distributions in these trials can be seen in figures (4.6), (4.8) and (4.10), and the objective functional values at each step can be found in figures (4.7),(4.9) and (4.11). As can be seen in figures (4.6), (4.8) and (4.10), Algorithm 2 ends

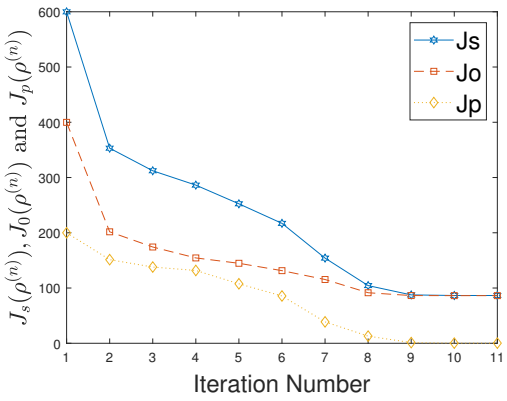
(A) $\alpha = 0$ (B) $\alpha = 0.5$

(C) $\alpha = 1$

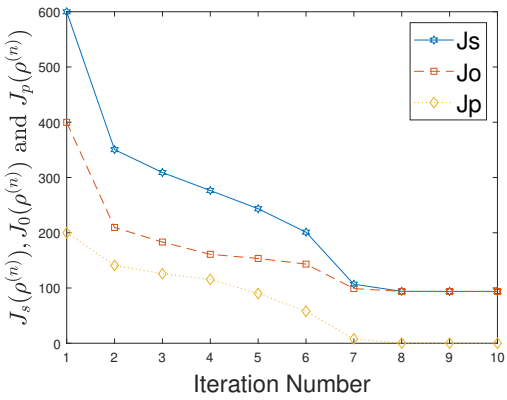
FIGURE 4.4: Final Mass Distributions of Continuous ℓ^2 Gradient Algorithms



(A) $\alpha = 0$



(B) $\alpha = 0.5$



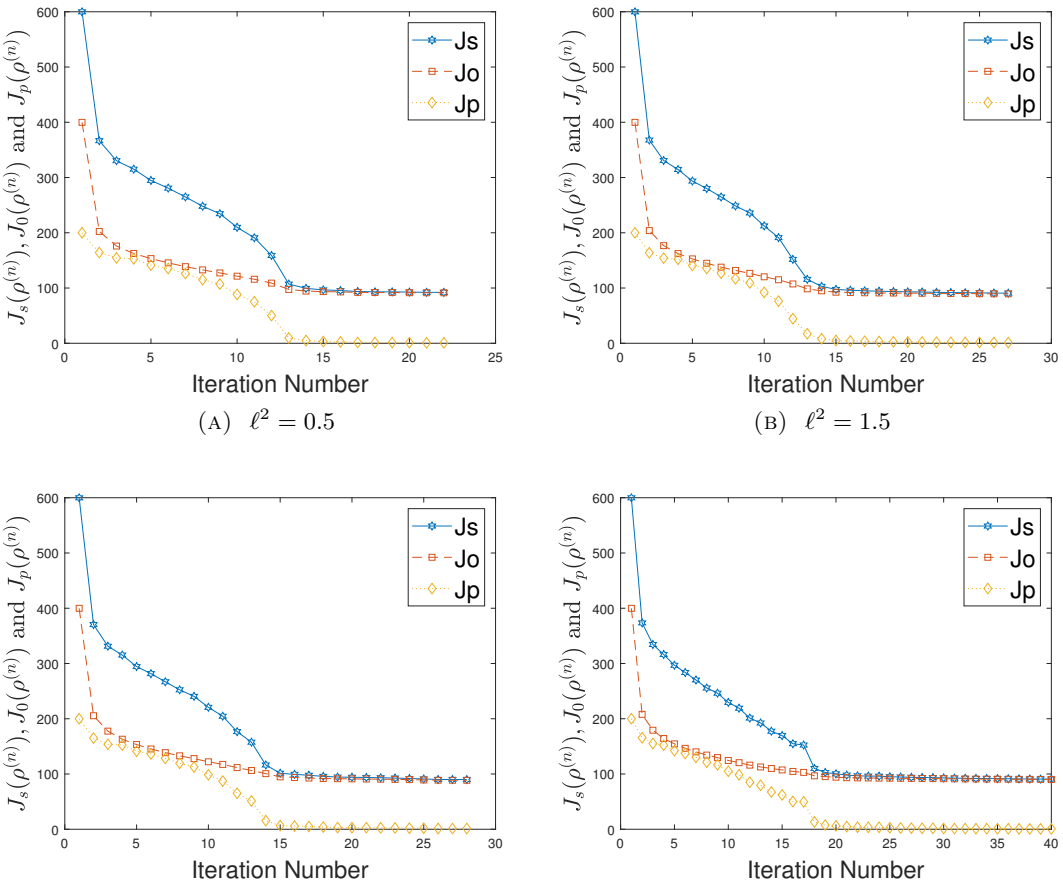
(C) $\alpha = 1$

FIGURE 4.5: Objective Functional Vs. Iteration of Continuous ℓ^2 Gradient Algorithms

(A) $\ell^2 = 0.5$ (B) $\ell^2 = 1.5$

(C) $\ell^2 = 2.5$ (D) $\ell^2 = 4.5$

Figure 4.6: Final Mass Distributions of Continuous Sobolev Gradients Algorithms at $\alpha = 0$



(A) $\ell^2 = 0.5$ (B) $\ell^2 = 1.5$

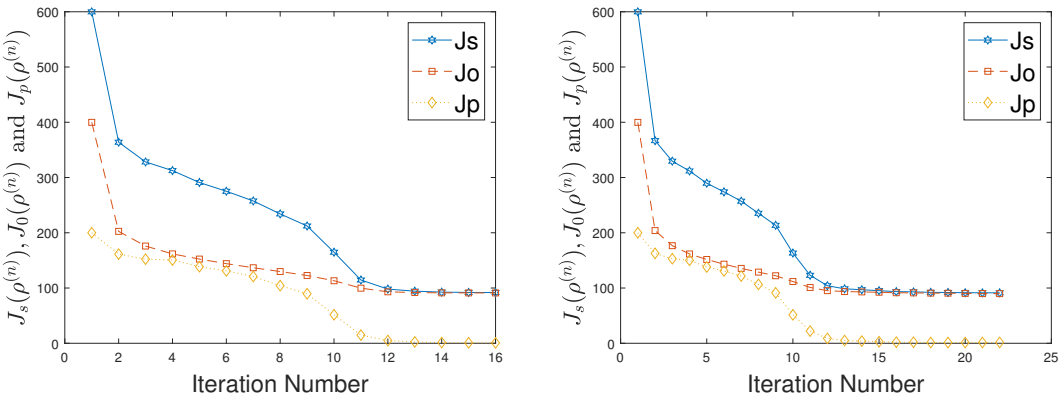
(C) $\ell^2 = 2.5$ (D) $\ell^2 = 4.5$

Figure 4.7: Objective Functional Vs. Iteration of Continuous Sobolev Gradients Algorithms at $\alpha = 0$

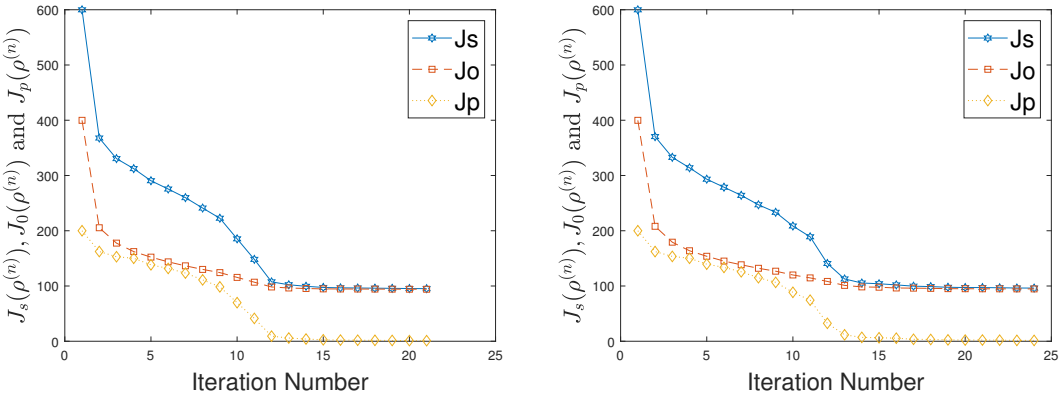
(A) $\ell^2 = 0.5$ (B) $\ell^2 = 1.5$

(C) $\ell^2 = 2.5$ (D) $\ell^2 = 4.5$

Figure 4.8: Final Mass Distributions of Continuous Sobolev Gradients Algorithms at $\alpha = \frac{1}{16}$



(A) $\ell^2 = 0.5$ (B) $\ell^2 = 1.5$



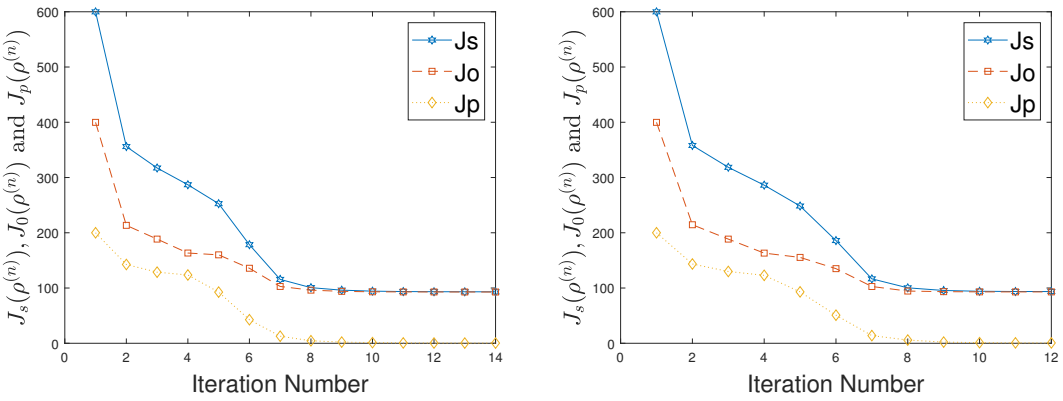
(C) $\ell^2 = 2.5$ (D) $\ell^2 = 4.5$

Figure 4.9: Objective Functional Vs. Iteration of Continuous Sobolev Gradients Algorithms at $\alpha = \frac{1}{16}$

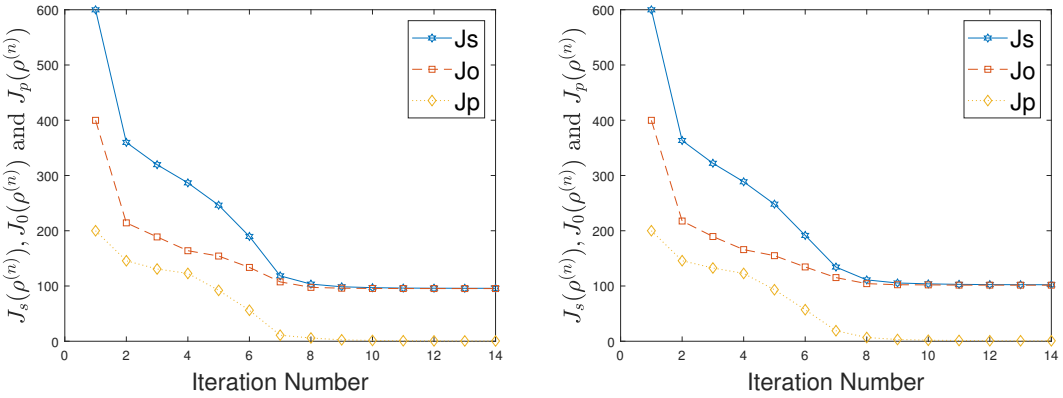
(a) $\ell^2 = 0.5$ (b) $\ell^2 = 1.5$

(c) $\ell^2 = 2.5$ (d) $\ell^2 = 4.5$

Figure 4.10: Final Mass Distributions of Continuous Sobolev Gradients Algorithms at $\alpha = 1$



(A) $\ell^2 = 0.5$ (B) $\ell^2 = 1.5$



(C) $\ell^2 = 2.5$ (D) $\ell^2 = 4.5$

Figure 4.11: Objective Functional Vs. Iteration of Continuous Sobolev Gradients Algorithms at $\alpha = 1$

with distributions that most resemble figure (4.4a), with the exception that there is no longer any major case of a checker-board distribution. From these figures, we are able to make two key observations. The first is that as the value of α increases, the size of the hole in the centre left of the structure decreases, and no longer exists at $\alpha = 1$. This is an interesting pattern as it is the same hole that disappeared when the gradient of the objective functional was changed in `Top99` from one derived in the discrete setting (3.92) to one that was derived in the continuous setting (3.4), as was pointed out when comparing figures (4.2c) and (4.2d). The second key observation we can make is that as the smoothing increases, the distributions change by also minimizing the previously mentioned hole, and the support beam in the bottom-right corner starts to merge with the bottom-right edge of the structure due to the increased smoothing.

Chapter 5

Conclusions

In this thesis we presented a detailed investigation into the derivation of expressions for gradients of different objective functionals, using the Riesz representation theorem, for the purpose of topology optimization. Our specific focus was on structural and thermal compliance in both the continuous and discrete settings, as well as a look at the differences between the uncoupled and coupled systems. We verified the gradient expressions in both the discrete and continuous setting by performing a Kappa test on them. We then used the gradient expressions obtained for the uncoupled system in the continuous setting and created an algorithm to minimize structural compliance (2), with constraints on the total mass and a limited range for our design variable (3.3). We concluded the analysis by comparing the results of the new algorithm with the standard algorithm for topology optimization, namely the 99-line code created by Sigmund (Sigmund 2001). The final results showed that we successfully met the optimization problem constraints using a change of variables and using a subspace projection, while we were also able to deal with the common topology optimization issue of a checker-board mass distribution (Bendsøe and Sigmund 2004) with the use of Sobolev gradients.

From figure 4.1 we can see that the gradient expression for the objective function derived in the continuous setting has a slower rate for the propagation of errors than the gradient

expression derived in the discrete setting. From the discussion in Chapter 4, it can be concluded that Algorithm 2 was successful in optimizing the objective function with a mostly similar structure to the ones obtained by the 99-line code. The main advantage in Algorithm 2 is that it is able to deal with both the checker-boarding issue and the ideal 0-1 distinction of the objective function better than the standard 99-line code. This can be seen in the figures in chapter 4 and the information found in table (4.1). This is because, when the standard 99-line code has a low value of J_p, for example when Rmin $= 0.5$ and $J_p = 0.5159$, then the corresponding distribution had a severe checker-board distribution issue (4.2a). When the smoothing was increased for the 99-line code, the checker-boarding issue was resolved (4.2c - 4.2e) but it resulted in J_p values in the range of 22-24, which suggests that there are many elements where the design variable was not close to either 0 or 1. This lack of 0-1 distinction can be seen in the final mass distributions as the figures have a "fussiness" to them that stems from some elements being neither fully black nor white. In contrast, Algorithm 2 never had a J_p value above 1.5063, which resulted in figures that looked very sharp in comparison to the ones from the 99-line code. Also, unlike the 99-line code, the algorithm was able to achieve these lower J_p values without succumbing to any major checker-boarding distributions as can be seen in figures (4.6a)–(4.10d).

There are two main disadvantages in Algorithm 2. The first is that it generally takes longer due to the line minimization step using the `fminbnd` function which is not the most efficient. While the second draw back of the algorithm is that it performs worse in minimizing compliance, as the J_0 values are in the range of $7\% - 20\%$ higher than the standard 99-line code. We believe that some of this worse performance is caused by the sharpness of the 0-1 distinction, as when we look at the combined values of J_p and J_0, namely J_s, Algorithm 2 tends to perform 10% better than the standard 99-line code, as the increased value of J_0 is compensated for by the much lower values of J_p. The different outcomes of Algorithm 2 for different values of ℓ^2 may suggest that the

optimization problem, 3.2, is not well posed in a mathematical sense. The argument for this is that the Riesz Representation Theorem is used to state the equivalence between the gradient in the L^2 space and in the Sobolev space, and since they are equivalent, for the problem to be well posed, they should both lead to the same unique final topology. One may argue against this by noting that the Riesz Representation Theorem implies that the integrals of the two gradients are equivalent. This means that the gradient are equivalent in an integral or in an overall sense and not in a exact or at every point sense. Now since they are not equivalent at every part of the domain, an optimization algorithm could lead to different local minimums using the different gradients, and both may miss the unique distribution that gives the global minimum.

We hope that this thesis, in its classification and in depth description in its use of the adjoint method will help in the deviation of gradient expressions for more complicated, or more non-standard topology optimization problems for future engineers. Specifically, we hope that the gradient used for the coupled elastic-thermal system will help in the solution of the open problem in topology optimization involving thermal radiation. Lastly, we hope that the ideas used in the proposed algorithm, such as the penalization term for the 0-1 distribution, the subspace projection to deal with the mass constraint, or the use of transforming the gradient into the Sobolev space to smooth the gradients and deal with the checker-board distribution, will help with or inspire ideas for future research in the field of topology optimization.

Bibliography

Bendsøe, M. and Sigmund, O. (2004). *Topology Optimization Theory, Methods and Applications*. Springer.

Bialecki, R. and Nowak, A. J. (1981). Boundary Value Problems in Heat Conduction with Nonlinear Material and Nonlinear Boundary Conditions. *Applied Mathematical Modelling* 5 (6), 417–421.

Chungang, Z., Zhenhua, X., and Han, D. (2010). Topology optimization of multi-material for the heat conduction problem based on the level set method. *Engineering Optimization*, 811–831.

Ferguson, D. and Peters, T. (2005). *Discretize then optimize, in Mathematics for Industry: Challenges and Frontiers*. SIAM.

Gere, J. M. (2004). *Mechanics of Materials*. Vol. 6. Thomson Learning Inc., 3–30.

Griffiths, D. J. (2005). *Introduction to Quantum Mechanics*. Vol. 2. Pearson Education Inc., 145–155, 204–205.

Liu, K. and Tovar, A. (2014). An efficient 3D topology optimization code written in Matlab. *Struct Multidisc Optim*, 1175–1196.

MATLAB (2019). *version 9.6.0 (R2019b)*. Natick, Massachusetts: The MathWorks Inc.

Nocedal, J. and Wright, S. J. (1999). *Numerical Optimization*. Springer-Verlag.

Rudin, W. (2006). *Real and Complex Analysis*. Vol. 3. McGraw Hill Education Inc., 40–45.

Sigmund, O. (2001). A 99 line topology optimization code written in Matlab. *Struct Multidisc Optim*, 120–217.