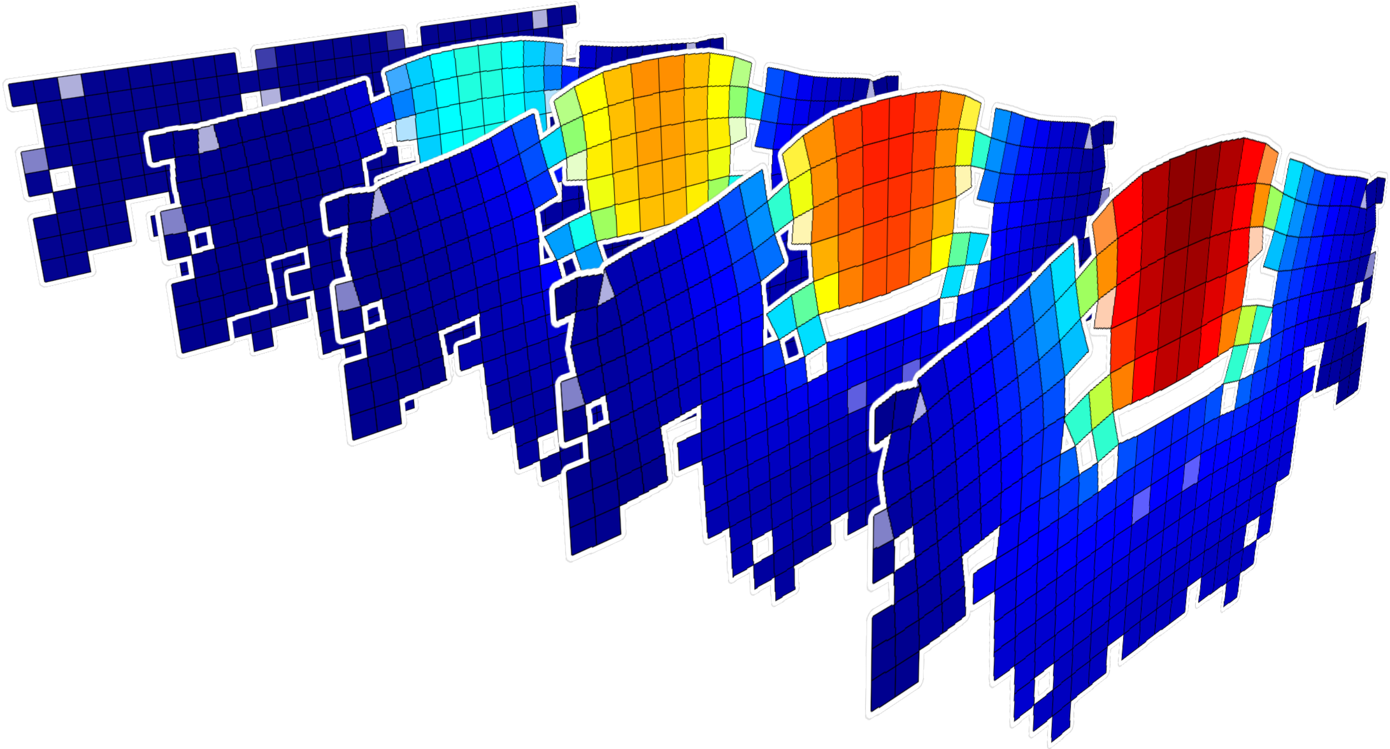
Department of Precision and Microsystems Engineering

Topology Optimization For Transient-Thermal-Mechanical Systems

Emiel van de Ven

Report no.	:	EM 2014.028
Coach	:	Ir. E.C. Hooijkamp
Professor	:	Prof.dr.ir. F. van Keulen
Specialisation	:	Engineering Mechanics
Type of report	:	Master Thesis



TU Delft Delft University of Technology

Challenge the future

Topology Optimization For Transient-Thermal-Mechanical Systems

by

Emiel van de Ven

in partial fulfillment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Delft University of Technology.

Supervisor:	Ir. E.C. Hooijkamp	TU Delft
Thesis committee:	Prof.dr.ir. F. van Keulen,	TU Delft
	Dr.ir. M. Langelaar,	TU Delft
	Dr.ir. P. Tiso,	TU Delft
	Ir. M. Koevoets	ASML

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Thermal error reduction is of major importance to the performance of any high-tech machine. Topology optimization can be a tool that is able to design geometries that reduce the thermal error, beyond human creativity. In this research, topology optimization is used to reduce the thermal error on a transient-thermal-mechanical problem. The thermal error of the specific problem was reduced by 30%. Multiphyiscal topology optimization, as applied to the problem, comes with many challenges, under which is the manufacturability of its final design. A new penalization method is proposed, that ensures that the final design contains no intermediate densities, which are common to multiphysical topology optimization, even with SIMP or RAMP penalization. The new penalization method aims at creating intermediate material that is adverse to the optimization problem, causing it to avoid intermediate material. On a test case it was shown capable of creating a well performing black and white design, where grey penalization failed to maintain the performance of the design. Furthermore, an interpretation of the adjoint in the sensitivity analysis is given.

Acknowledgements

I would like to thank several people who enabled this research. First of all Evert Hooijkamp, who I could always bother with questions and who spend a good amount of his time on colouring my draft thesis green with annotations. Also from the PME department: Fred van Keulen and Matthijs Langelaar on whom I could test case my understanding of the subject. From ASML I would like to thank Marco Koevoets and his colleges from the (Mechatronics) Research department, who always had fresh ideas when I was only looking through the PME goggles.

Furthermore, I would like to thank all the fellow students and friends at PME for the coffee brakes, lunches, and other stuff that caused study delays. And of course the friends outside PME, room mates, who I have seen a little bit less last year.

Last but not least I would like to thank Sarah and her family, my family and especially my parents for their support, and the nice time (and lovely cooking...) I had during my internship at ASML.

Thanks!

Emiel van de Ven

Contents

1
Introduction

In any machine that performs measurements or tooling on a workpiece, alignment of the tool and workpiece is of primal importance. Misalignment can have multiple sources, such as static or dynamic mechanical loads, thermo-mechanical effects or control errors. This report focuses on the reduction of the error caused by thermal-mechanical effects (i.e. thermal expansion), also called thermal reduction. Early examples of thermal reduction can be found in the 18th century, where the thermal expansion of a pendulum was compensated by using materials with different coefficients of thermal expansion (CTE) [1]. Since then many methods for thermal reduction have been applied. Three main methods of thermal reduction can be distinguished [2]:

- Material and geometry selection
- Thermal conditioning
- Compensation

The example of the pendulum would be in the first category. Thermal conditioning includes passive and active cooling or heating. Compensation is on the control side: measuring or predicting the thermal error and adjusting the position accordingly.

In the high tech industry the error budgets are extremely small. In the lithography systems produced by ASML, this is in the order of nanometers or even less. In a part of one the ASML lithography systems (further explained in chapter 5), all three above mentioned techniques are combined in order to reduce the thermal error. Although the thermal error is reduced eventually, it is not reduced in the initial transient domain. For that reason a geometry and material combination is needed that reduces the thermal error in the initial transient domain.

In the given design domain, the geometry can have any shape as long as it improves the thermal error. This geometry might be very complex and hard to think of by hand, it might even act as a mechanism. Therefore topology optimization (TO) is used to find an improved design. TO is a design tool which iteratively computes a design in a certain domain. With every iteration an objective function, e.g. the thermal error, is further reduced. TO often finds innovative designs that are sometimes easy to understand but hard to think of by hand, which is also achieved in this research in chapter 5.

The main goal of this research is to perform a transient thermal-mechanical TO on a case provided by ASML, in order to find an improved design that minimizes the thermal error. However, TO is a far from mature design tool and only few optimizations involving a transient thermal-mechanical analysis can be found in literature. Therefore the main challenge of this project is performing the transient-thermal-mechanical topology optimization itself. One of the challenges is keeping the computational time within bounds. A small research into sensitivities, a computationally costly part of the TO loop, has been done. In this research a clear distinction between direct and adjoint sensitivities has been found. Another problem of multiphysics optimizations is, that in the designs it is often not clear where there is material and where there is void. All kinds of intermediate materials are usually present. In order to improve manufacturability of the designs, a new penalization technique is presented which performs a better job at suppressing these intermediate materials. Finally, this is all applied to the case provided by ASML, and improved design has been found.

This report consists of an introductory chapter about TO, and three self-contained chapters. In the introductory chapter 2 the concept of TO is explained. The sensitivity analysis that is used in TO is analysed and interpreted in chapter 3. Since the sensitivity analysis is the most time consuming step it is important to understand it to ensure that the least time consuming method is used. In chapter 4, transient thermal-mechanical TO is explained in detail and a new penalization method is introduced that suppresses artefacts from the TO. The ASML case, already briefly mentioned, is introduced in chapter 5, and TO is applied. In the final chapter 6, conclusions and recommendations are given. Readers who are not interested in the details of TO but are curious about its possibilities might read chapter 5 only. Readers already familiar with TO might skip chapter 2.

2
Topology Optimization

TO is an optimization, which aims to find an optimal distribution of material in a predefined domain for a certain design problem. Typical for TO is that it has many design variables and relatively few constraints. That is caused by the fact that design variables describe whether material is present or absent in an element. Constraints are often imposed over the whole domain, for example on the amount of volume that is used or the maximum allowed temperature. TO can be described as an iterative loop with distinct elements. First, an initial design is proposed. After that, the performance of this design is evaluated with a model, e.g. a finite element model (FEM). The performance is evaluated in the form of an objective function which has to be as small as possible (small meaning as close to $-\infty$ as possible). The objective function is for instance the absolute displacement of a certain point. Next, the derivative of this objective function with respect to the each design variable is calculated. These derivatives, or sensitivities, are used by the optimizer to find a new design. Then the loop starts again by evaluating the new design. A schematic of the optimization loop is given in figure 2.1.

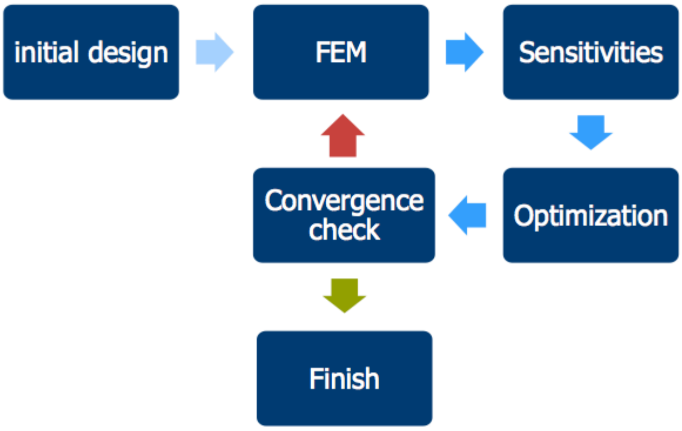


Figure 2.1: Schematic representation of the optimization loop for gradient based optimization.

In this chapter the three main blocks of the TO iteration cycle: the model, sensitivities and optimization, are discussed, each in their own section. Additionally, there is a section explaining what alterations to the presented theory have to be made to adapt it for multi-material topology optimizationm, and a short summary is given.

2.1. The model

The first step in every iteration is to build a model with the current design and test it. The model used in this report is a FEM. Inherent to creating a FEM is discretizing the design domain into elements. In TO, for each element a so called pseudo-density is defined. A pseudo density of 1 means that there is material in the element, 0 means that it is empty. So the total number of design variables is equal to the amount of elements. For the sake of simplicity and to keep the amount of degrees of freedom (DOFs) to a minimum, linear shape function were chosen. The set-up of the structural and thermal matrices involved in finite element modelling is explained in detail in Appendix A. For a transient thermal-mechanical system the state equations are:

$$\mathbf{C}(\boldsymbol{\rho})\dot{\mathbf{T}}(t) + \mathbf{K_T}(\boldsymbol{\rho})\mathbf{T}(t) = \mathbf{Q}(t, \boldsymbol{\rho}) \tag{2.1}$$

$$\mathbf{K_u}(\boldsymbol{\rho})\mathbf{u}(t) = \mathbf{A}(\boldsymbol{\rho})\mathbf{T}(t) \tag{2.2}$$

where $\mathbf{C}$ is the capacitance matrix, $\mathbf{K_T}$ is the heat conduction matrix, $\mathbf{K_U}$ the structural stiffness matrix and $\mathbf{A}$ the thermal coupling matrix, all assumed constant with temperature and time, but dependent on the design variables contained in the vector $\boldsymbol{\rho}$. $\mathbf{T}$ is a vector containing the nodal temperatures and $\mathbf{u}$ is a vector containing the nodal displacements. $\mathbf{Q}$ is a load vector dependent on time and the design variables. As stated, the performance of a design is evaluated by calculating a certain objective function f which needs to be minimized. In this report the objective will eventually be the minimization of displacement, so the objective function is only directly dependent on the displacements $\mathbf{u}$. Furthermore, because the problem is transient, the objective function is a time integral over the time frame $[0, t_f]$. The total optimization problem can be written as (with N design variables):

$$\min_{\rho_i} \left(f(\mathbf{u}(t)) = \int_0^{t_f} p(\mathbf{u}(t))\mathrm{d}t \right) \tag{2.3}$$

$$\text{subject to: } \mathbf{C}(\rho_i)\dot{\mathbf{T}} + \mathbf{K_T}(\rho_i)\mathbf{T} = \mathbf{Q}(\rho_i) \tag{2.4}$$

$$\mathbf{K_U}(\rho_i)\mathbf{u} = \mathbf{A}(\rho_i)\mathbf{T} \tag{2.5}$$

$$0 \leq \rho_i \leq 1 \qquad i = 1..N \tag{2.6}$$

2.1.1. The density approach

By assigning a design variable to every element, which should either have material or no material, a discrete optimization problem is created. However discrete problems are generally difficult to solve. Therefore, the so called density approach is used to relax the optimization problem. In this method, first proposed by Bendsøe [3], pseudo densities between 0 and 1 are also allowed. The material properties are scaled with these densities. For a purely mechanical 2D optimization, one could interpret the densities as the out of plane thickness of the 2D structure that is created. By simply using a linear scaling of the material properties, the optimum layout usually has a lot of intermediate (values between 0 and 1) densities. Because eventually a manufacturable discrete design is desired, a non-linear scaling of the material properties is used in combination with a volume constraint. The most common scaling methods are SIMP and RAMP, which are defined as follows [4]:

$$E^*_{SIMP}(\rho) = E_{min} + \rho^p \left(E_{min} + E_0\right) \tag{2.7}$$

$$E^*_{RAMP}(\rho) = E_{min} + \frac{\rho}{1 + q(1-\rho)} \left(E_{min} + E_0\right) \tag{2.8}$$

Here E_0 is the normal material parameter at full density. E_{min} is a lower bound added for numerical reasons, chosen at least a factor $1 \cdot 10^{-6}$ smaller than E_0. ρ is the pseudo density. p and q are the so called penalization parameters for SIMP and RAMP, respectively. The effect of these parameters is displayed in figure 2.2.

SIMP penalization of $p = 0$ leads to a constant material property independent of the element density. $p = 1$ is a linear scaling. For $p > 1$, the material property has become smaller at intermediate densities with respect to the density, i.e. the volume it occupies. If a high value of the material property gives better performance (for example a higher Young's modulus gives a higher stiffness to the total structure), intermediate densities are less favourable compared to full or empty densities with respect to the volume they occupy. For example with a SIMP scaling with $p = 8$, an element that has a density of 0.5 hardly adds any stiffness, but adds half of the volume of the full element to the occupied volume. Because the volume is limited, this scaling drives the optimizer to a black and white design because half densities are less "economical" to use.

For structural problems, only the Young's modulus needs to be scaled. However for transient thermal-mechanical problems there are four parameters that can be scaled: the Young's modulus

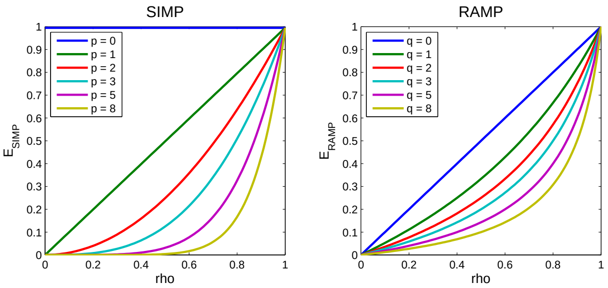


Figure 2.2: SIMP and RAMP scaling of a material property for different penalization factors.

E, the heat conduction coefficient k, the volumetric heat capacity c and the thermal stress coefficient β, defined as the product of the Young's modulus and the thermal expansion coefficient (as proposed in Gao and Zhang [5]).

Consequently, the response of the structure is a function of the (continuous) densities ρ (the penalizations are kept constant). For a given design, the derivative of the objective function with respect to the densities can be calculated, which is demonstrated in section 2.2. Any gradient based optimization algorithm can then be used to find an optimum.

2.1.2. Time integration

Because the problem is transient, the state equations have to be solved over a period of time. In this study, a simple time stepping algorithm has been used. The temperature is discretized in time using:

$$\dot{\mathbf{T}}(t) = \frac{\mathbf{T}^k - \mathbf{T}^{k-1}}{\mathrm{d}t}, \qquad \mathbf{T}(t) = (1-\alpha)\,\mathbf{T}^{k-1} + \alpha\mathbf{T}^k \tag{2.9}$$

Here $\mathbf{T}^k$ is the temperature at time step k, and $\mathrm{d}t$ is the time between each time step, taken constant. α is a parameter which defines the integration scheme. For $\alpha = 1$, explicit Euler forward time integration is used, and for $\alpha = 0$ implicit Euler backward time integration is used. The heat equation now becomes:

$$\mathbf{T}^k \underbrace{\left(\frac{1}{\mathrm{d}t}\mathbf{C}_\mathbf{T} + \alpha\mathbf{K}_\mathbf{T}\right)}_{\mathbf{E}_\mathbf{T}} + \mathbf{T}^{k-1} \underbrace{\left(-\frac{1}{\mathrm{d}t}\mathbf{C}_\mathbf{T} + (1-\alpha)\mathbf{K}_\mathbf{T}\right)}_{\mathbf{A}_\mathbf{T}} = \mathbf{Q}^k \tag{2.10}$$

Given an initial temperature, the temperature can be calculated over time:

$$\mathbf{T}^k = \mathbf{E}_\mathbf{T}^{-1}\left(\mathbf{Q}^k - \mathbf{A}_\mathbf{T}\mathbf{T}^{k-1}\right) \tag{2.11}$$

$$\mathbf{T}^0 = \mathbf{T}_{\text{ambiant}} \tag{2.12}$$

Since the mechanical behaviour is considered as being static, the deformation can be calculated at every time step when the temperature field is known. In this report the Euler backward method is used because it is unconditionally stable and shows no oscillations. The choice of the integration scheme is not extensively considered. Other schemes such as the Crank-Nicolson scheme might have provided more accurate results, but might also have required a smaller time step for stability.

2.2. Sensitivities

In order to find the optimum, the sensitivities of the objective function are needed. Straightforward differentiation of equations 2.4 gives $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i}$ and differentiation of equation 2.5 gives $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$. Substituting these derivatives in the derivative of the objective function gives:

$$\frac{\mathrm{d}f}{\mathrm{d}v_i} = \int_0^{t_f} \frac{\partial p}{\partial \mathbf{u}} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i} \mathrm{d}t \tag{2.13}$$

$$= \int_0^{t_f} \frac{\partial p}{\partial \mathbf{u}} \mathbf{K_U}^{-1} \left(-\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}\mathbf{u} + \mathbf{A}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} + \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}\mathbf{T} \right) \mathrm{d}t \tag{2.14}$$

$$= \int_0^{t_f} \frac{\partial p}{\partial \mathbf{u}} \mathbf{K_U}^{-1} \left(-\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}\mathbf{u} + \mathbf{A}\mathbf{K_T}^{-1} \left(-\mathbf{C_T}\frac{\mathrm{d}\dot{\mathbf{T}}}{\mathrm{d}v_i} - \frac{\mathrm{d}\mathbf{C_T}}{\mathrm{d}v_i}\dot{\mathbf{T}} - \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}\mathbf{T} + \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i} \right) + \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}\mathbf{T} \right) \mathrm{d}t \tag{2.15}$$

The differentiation as performed in equation 2.15 is referred to as the direct method [6]. One can reuse the factorizations of the matrices $\mathbf{K_U}$ and $\mathbf{K_T}$ that have been calculated for the model evaluation. The term between brackets by which $\mathbf{K_U}^{-1}$ is multiplied in equation 2.15, is a matrix with as many rows as there are structural or thermal degrees of freedoms, and as many columns as there are design variables. One forward and one backward substitution needs to be performed for every column (or design variable), which costs a lot of computational time for topology optimization, where there are many design variables [7]. An other option to calculate the sensitivities is called the adjoint method [6]. With the adjoint method the state equations 2.4 and 2.5 are pre-multiplied with an adjoint variable μ and λ respectively, and added to the objective function. Since the state equations are always satisfied, this will not change the objective function:

$$f^* = f = \int_0^{t_f} \left[p + \lambda^T \left(\mathbf{K_U}\mathbf{u} - \mathbf{A}\mathbf{T} \right) + \mu^T \left(\mathbf{C_T}\dot{\mathbf{T}} + \mathbf{K_T}\mathbf{T} - \mathbf{Q} \right) \right] \mathrm{d}t \tag{2.16}$$

Differentiating with respect to the design variables gives:

$$\begin{aligned} \frac{\mathrm{d}f^*}{\mathrm{d}v_i} = \int_0^{t_f} \Bigg[\frac{\partial p}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i} + \lambda^T \left(\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}\mathbf{u} + \mathbf{K_U}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i} - \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}\mathbf{T} - \mathbf{A}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} \right) + \\ \mu^T \left(\mathbf{C_T}\frac{\mathrm{d}\dot{\mathbf{T}}}{\mathrm{d}v_i} + \frac{\mathrm{d}\mathbf{C_T}}{\mathrm{d}v_i}\dot{\mathbf{T}} + \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}\mathbf{T} + \mathbf{K_T}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} - \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i} \right) \Bigg] \mathrm{d}t \end{aligned} \tag{2.17}$$

In this report the numerator layout has been used. The derivative of a scalar w.r.t. a vector is thus defined as being a row vector. Rearranging equation 2.17 gives:

$$\begin{aligned} \frac{\mathrm{d}f^*}{\mathrm{d}v_i} = \int_0^{t_f} \Bigg[\left(\frac{\partial p}{\partial \mathbf{u}} + \lambda^T\mathbf{K_U} \right) \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i} + \left(-\lambda^T\mathbf{A} + \mu^T\mathbf{K_T} \right) \frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} + \mu^T\mathbf{C_T}\frac{\mathrm{d}\dot{\mathbf{T}}}{\mathrm{d}v_i} + \\ \lambda^T \left(\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}\mathbf{u} - \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}\mathbf{T} \right) + \mu^T \left(\frac{\mathrm{d}\mathbf{C_T}}{\mathrm{d}v_i}\dot{\mathbf{T}} + \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}\mathbf{T} - \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i} \right) \Bigg] \mathrm{d}t \end{aligned} \tag{2.18}$$

In order to get rid of the term $\frac{\mathrm{d}\dot{\mathbf{T}}}{\mathrm{d}v_i}$ integration by parts can be used:

$$\begin{aligned} \frac{\mathrm{d}f^*}{\mathrm{d}v_i} = \left[\mu^T\mathbf{C_T}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} \right]_0^{t_f} + \int_0^{t_f} \Bigg[\left(\frac{\partial p}{\partial \mathbf{u}} + \lambda^T\mathbf{K_U} \right) \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i} + \left(-\lambda^T\mathbf{A} - \dot{\mu}^T\mathbf{C_T} + \mu^T\mathbf{K_T} \right) \frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} + \\ \lambda^T \left(\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}\mathbf{u} - \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}\mathbf{T} \right) + \mu^T \left(\frac{\mathrm{d}\mathbf{C_T}}{\mathrm{d}v_i}\dot{\mathbf{T}} + \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}\mathbf{T} - \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i} \right) \Bigg] \mathrm{d}t \end{aligned} \tag{2.19}$$

The evaluation of the terms $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i}$ and $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$ costs a lot of computational time, this will be explained in more detail in chapter 3. These terms can be eliminated from the equation by setting the adjoint variables such that the terms by which $\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}v_i}$ and $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$ are multiplied, equal to zero:

$$\frac{\partial p}{\partial \mathbf{u}} + \lambda^T\mathbf{K_U} = 0 \tag{2.20}$$

$$-\lambda^T\mathbf{A} - \dot{\mu}^T\mathbf{C_T} + \mu^T\mathbf{K_T} = 0 \tag{2.21}$$

$$\left[\mu^T\mathbf{C_T}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} \right]_0^{t_f} = 0 \tag{2.22}$$

Because $\mathbf{T}(t=0) = \mathbf{T}_{\text{ambiant}}$, $\frac{\mathrm{d}\mathbf{T}(0)}{\mathrm{d}v_i} = 0$. This reduces equation 2.22 to:

$$\mu^T(t = t_f) = \mathbf{0} \tag{2.23}$$

The adjoint variable μ is discretized in time in the same way as the temperature is descretized in time, as explained in section 2.1.2. In fact, when one replaces μ with $\mathbf{T}$ and $\lambda^T\mathbf{A}$ with the thermal load $\mathbf{Q}$, the reverse of the thermal state equation 2.4 is found. Because this terminal condition is at the end of the integration time (equation 2.23), the adjoint variable μ^T is integrated backwards in time. With this initial condition the sensitivity of the objective function discretized in time can be calculated:

$$\frac{\mathrm{d}f}{\mathrm{d}v_i} = \frac{\mathrm{d}f^*}{\mathrm{d}v_i} = \int_0^{t_f} \left[\lambda^T \left(\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}\mathbf{u} - \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}\mathbf{T}\right) + \mu^T \left(\frac{\mathrm{d}\mathbf{C_T}}{\mathrm{d}v_i}\dot{\mathbf{T}} + \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}\mathbf{T} - \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i}\right)\right] \mathrm{d}t \tag{2.24}$$

$$\lambda_k^T = -\mathbf{K_U}^{-1}\left(\frac{\partial p_k}{\partial \mathbf{u}}\right) \tag{2.25}$$

$$\mu_k^T = \mathbf{E}_\mathbf{T}^{-1}\left(\lambda_k^T\mathbf{A} - \mathbf{A_T}\mu_{k+1}^T\right) \tag{2.26}$$

$$\mu^T(t_f) = \mathbf{0} \tag{2.27}$$

Here the subscript k denotes time step k. The factorizations of the matrices $\mathbf{K_U}$, $\mathbf{A_T}$ and $\mathbf{E_T}$ that have been calculated for the model evaluation can be reused. The adjoint method saves computational time, because only one forward substitution and one back substitution per time step are needed since the terms between brackets in equations 2.25 and 2.26 are column vectors. That is because equations 2.26 and 2.25 are independent of the design variable to which the objective is derived.

2.3. The optimizer

In this study, linear programming with move limits is used. At every iteration the design variables are moved in the direction in which the derivative of the objective function is negative. The step size equals the move limit of the specific design variable, which are also updated every iteration. If two consecutive steps are in the same direction, the move limit is increased, while if the direction are opposite (indicating oscillation) the move limit is decreased. This is a computationally cheap optimization algorithm, since only the first derivatives of the objective w.r.t. the design variables are needed and no equations need to be solved.

In TO problems where the density approach is used, often an implementation of Svanberg's "method of moving asymptotes" (MMA) is used (see Svanberg [8] for a detailed description). This optimizer seems to perform well on topology optimization problems (Sigmund and Maute [9], section 6.5). MMA approximates the performance measures by reciprocal functions of the form $1/(U-x)$ or $1/(x-L)$, depending on the sign of the derivative. Here U and L are the positions the asymptotes. By moving the asymptotes the curvature of the functions can be controlled. In the left graph of figure 2.3 and example can be seen of a function (blue) approximated by MMA with the asymptote close by (red) and far away (green). There are two reasons why MMA is not used in this study.

First of all, in unconstrained optimization as with the ASML case presented in chapter 5, the minimum of these function is at either plus or minus infinity. This means that at every step the design variables bounce into the move limits that are given. Thus the reciprocal approximation has no added value and therefore simple linear programming is used.

The second reason is that MMA does not allow for a decent altering of the optimization when a bad approximation has been made. After the optimizer has made a step, a check of the difference between the predicted and calculated performance can be easily calculated. The following measure was chosen to check the performance:

$$f_{\text{quality}} = \left| 1 - \frac{f_{\text{old}} + (\boldsymbol{\rho}_{\text{new}} - \boldsymbol{\rho}_{\text{old}})^T \frac{\mathrm{d}f_{\text{old}}}{\mathrm{d}\boldsymbol{\rho}}}{f_{\text{new}}} \right| \tag{2.28}$$

If f_{quality} is bigger than 0.01 (the difference between the approximation and the real objective is larger than 1%), the new design should be rejected and a more conservative approximation should be made. An extension on MMA, globally convergent MMA (GCMMA) [10], provides a performance check. However, GCMMA only works for approximations that are too progressive compared to the real objective. If the the approximated value is lower than the real objective, the asymptotes are moved closer to the linearisation point to make a more conservative approximation (see the left graph of figure 2.3). If the approximated value is too high, the asymptotes can be moved further away which at most makes the approximation linear. This works well for problems in which the design variables by nature have a reciprocal relation with the objective function. This is for example the case when the design variables are thicknesses of beams, the problem from which the idea of using reciprocal design variables originated. However if that is not the case, there will be no means of making a more conservative approximation than linear, when the approximated value is too high (see the right graph of figure 2.3). Therefore, also GCMMA is not used since it only can move the asymptotes closer for a more conservative approximation.

Instead of linear programming with adaptive move limits, MMA can also be used with adaptive move limits. Then the optimizer has two means (lowering the move limit and moving the asymptotes closer) to compensate for a too low approximation, and only one method (lowering the move limit) to compensate for a too high approximation. Both MMA and linear programming with adaptive move limits give approximately similar performance. Since the choice of optimizer is not the main topic of this study, linear programming has been chosen for simplicity. If MMA with adaptive move limits is tweaked (for example changing the ratio of changing the move limits versus the move limits) it might outperform linear programming.

2.4. Multi-material topology optimization

Besides choosing an optimal material layout, which is basically choosing between material or void for every element, TO can also be used to choose between material 1, material 2 or void. Adding a second material to the optimization loop changes the model and especially its sensitivities. In the end of chapter 5, this multi material optimization is used to get a better performance in the space limited design domain. In this section the implementation of multi material topology optimization is discussed.

There are several options for the implementation of this extra material. In the book of Bendsoe and Sigmund [11], an interpolation scheme based on the Hashin-Shtrikman upper and lower bounds is suggested. In Yin and Ananthasuresh [12] a method using only a single design variable is suggested by defining a Gaussian distribution with different mean and standard deviation for each material parameter and linearly combining these distribution. In this report, the interpolation is done by adding an extra design variable that chooses between the two non-void materials. For every element the optimizer can choose if:

1. there is material or no material by setting the pseudo density ρ_d between 0 and 1.
2. the element is material 1 or material 2 by setting a second design variable ρ_m between 0 and 1.

This is illustrated in figure 2.4. For SIMP penalization this gives the following interpolation scheme (compared to single material SIMP in equation 2.7):

$$E^*(\rho_\mathrm{m}, \rho_d) = \rho_\mathrm{d}^{p_\mathrm{d}} \left(\rho_\mathrm{m}^{p_\mathrm{m}} E_2 + (1-\rho_\mathrm{m})^{p_\mathrm{m}} E_1\right) + \left(1-\rho_\mathrm{d}^{p_\mathrm{d}}\right) E_{min} \tag{2.29}$$

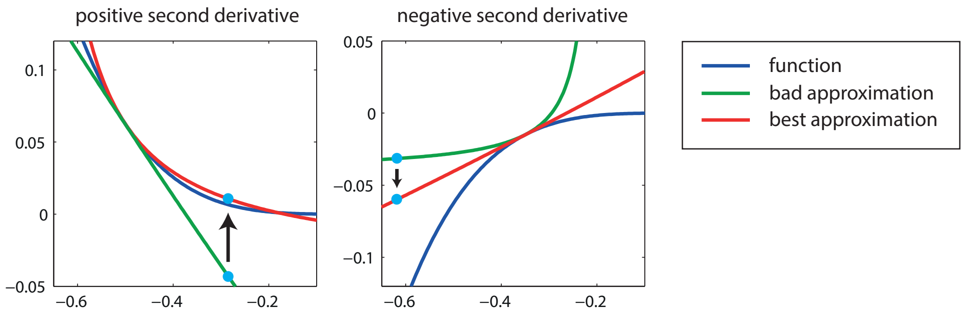


Figure 2.3: MMA can only approximate function with a positive second derivative. If the second derivative is negative, it can at best make a linear approximation.

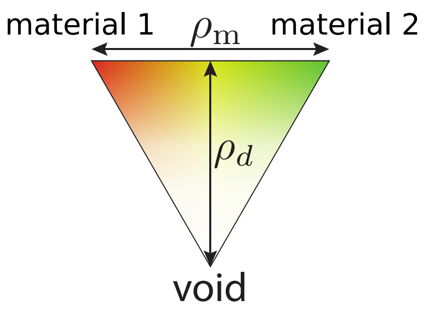


Figure 2.4: Material triangle for multiple material optimization.

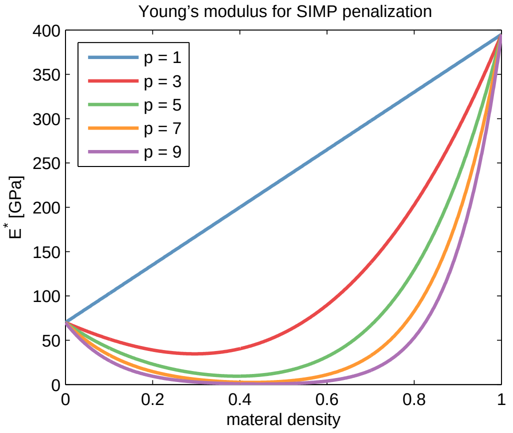


Figure 2.5: Interpolated Young's modulus for different values of the intermediate material penalization.

Here E^* is the interpolated material property, E_1 and E_2 the material properties of the two materials. p_{d} and p_{m} are the penalization parameters for the density and the material interpolation. This interpolation scheme is chosen for its simplicity and is reported to work well in for example Wang *et al.* [13], where it is used on a mechanical problem with a thermal load. The effect of the material penalization parameter for SIMP can be seen in figure 2.5. With penalization, intermediate materials have a low stiffness compared to either material 1 or 2. No penalization provides a linear transition between the two materials. During the optimizations it became apparent that no penalization gave the best performing designs, and little intermediate material was present in the multi-material optimization. Apparently either material 1 or material 2 was favourable for most cases.

With a set of material variables and pseudo densities the response of the model can be calculated by using one of the above mentioned interpolation schemes. However compared to the single material case there is one major difference: besides the Young's modulus, thermal expansion coefficient, heat conduction coefficient and heat capacity, the Poisson ratio is also dependant on the material design variables. For the Poisson ratio penalization, just as for the other material parameters, a linear dependence of the Poisson ratio on the material density was chosen:

$$\nu^*(\rho_{\mathrm{m}}) = \rho_{\mathrm{m}}\nu_1 + (1-\rho_{\mathrm{m}})\nu_2 \tag{2.30}$$

Where in the single material optimization the element matrices only need to be scaled with the interpolated material parameters, the element matrices have to be recalculated at every iteration in the multi-material optimization because of the changing Poisson ratio. This adds computational time in the model evaluation and especially when calculating the sensitivities. The derivation of the multi-material sensitivities is explained in appendix C.

Concluding, for the multi-material optimization, a design variable ρ_m is added for every element that chooses the material, besides the normal design variable ρ_d that chooses between material and void. Because of the dependence of the Poisson ratio on the new design variable, the element matrices need to be recalculated for every element which adds computational time.

2.5. Topology optimization loop

Summarizing, the topology optimization loop starts with an initial design for a design problem. For the initial design, all elements are set to half density. Based on this design, a FEM is constructed and solved. Next, the first order adjoint sensitivities are calculated. A linearisation around the current design is made and the optimizer moves the design variables such that the objective is maximally decreased. If the change in objective is below a certain value or if a maximum number of iterations is reached, the loop is ended. Otherwise, a next iteration is made starting from the new design.

3 Transient adjoint sensitivities

In a gradient optimization process it is important to know the derivatives of the performance measures, such as objective and constraint functions, with respect to the design variables. This information is used to make a step in the right direction to obtain a more optimal design. The calculation of these derivatives can be a computationally expensive process. There are three main approaches to calculate the derivatives: the global finite difference, the direct and the adjoint method for calculating derivatives. Haftka [6] was by the authors knowledge the first to apply the three methods to the case of a transient heat conduction problem. After his article there have been many articles expanding and generalizing the formulation of the adjoint sensitivity for the heat conduction problem. For example Dems [14], who added heat flux boundary conditions and expanded the performance measure to also be dependent on the gradient of the temperature, the boundary heat flux and the temperature change in time. Or Tortorelli and Haber [15], who added almost every possible boundary condition and objective function dependency.

Because of the relative simple problem treated in [6], this article is taken as a starting point for this report and is loosely summarized in Section 3.1 for sake of completeness. In Section 3.2 the adjoint sensitivity is directly derived from the direct method and an interpretation of the adjoint method is given. In Section 3.3 different methods of time integration and their (dis)advantages are discussed.

3.1. Introduction to sensitivities for the heat conduction problem

A general case of a transient heat conduction problem has the form of (discretized in space):

$$\mathbf{C}(\mathbf{v})\dot{\mathbf{T}}(t) + \mathbf{K}(\mathbf{v})\mathbf{T}(t)) = \mathbf{Q}(\mathbf{v}, t) \tag{3.1}$$

with:

$$\mathbf{T}(0) = \mathbf{T}_0 \tag{3.2}$$

where $\mathbf{C}$ is the capacitance matrix and $\mathbf{K}$ is the heat conduction matrix, both assumed constant with temperature and time, but dependent on the design variables $\mathbf{v}$. $\mathbf{T}$ is a vector containing the unknown temperatures, dependent on time. $\mathbf{Q}$ is a load vector dependent on time and the design variables. In TO the derivatives of the design variables with respect to the performance measures need to be known. In transient problems these functions acts on a certain time frame from $t = 0$ to final time $t = t_f$ and have the form:

$$g(\mathbf{v}, \mathbf{T}) = \int_0^{t_f} p(\mathbf{v}, \mathbf{T}, t)\mathrm{d}t \tag{3.3}$$

where $p(\mathbf{v}, \mathbf{T}, t)$ is a performance measure. $g(\mathbf{v}, \mathbf{T})$ is its integrated form. The derivative of g w.r.t. each design variable v_i is wanted in order to update the design variables. Thus the following vector is to be calculated:

$$\frac{\mathrm{d}g(\mathbf{v}, \mathbf{T})}{\mathrm{d}v_i} \qquad i = 1..n \tag{3.4}$$

where n is the number of design variables. In this report the numerator layout is used. Thus, the derivative of a scaler w.r.t. a vector is defined as being a row vector. In the rest of this section the addition $i = 1..n$ is omitted.

3.1.1. Global finite differences

Global finite differences is the easiest method to implement but most of the time also the computationally most expensive method. In this method the design variables are perturbed by a small amount and from the differences in solutions the derivatives can be calculated:

$$\frac{\mathrm{d}g(\mathbf{v},\mathbf{T})}{\mathrm{d}v_i} \approx \frac{g(\mathbf{v}+\Delta\mathbf{v},\mathbf{T}) - g(\mathbf{v},\mathbf{T})}{\Delta v_i} \tag{3.5}$$

where Δv_i is the perturbation of design variable v_i, and $\Delta\mathbf{v}$ is a vector containing zeros except on the ith position where it is equal to Δv_i. This method implies that for each design variable every performance measure needs to be recalculated in order to calculate $g(\mathbf{v}+\Delta\mathbf{v},\mathbf{T})$. This means the system needs to be resolved many times, which can be costly. Furthermore, the step size is not a trivial value. It needs to be small enough to be accurate, but not too small to cause numerical noise to dominate the calculation.

3.1.2. Direct method

Differentiating equation 3.3 with respect to the design variables gives:

$$\frac{\mathrm{d}g(\mathbf{v},\mathbf{T})}{\mathrm{d}v_i} = \int_0^{t_f} \left[\frac{\partial p}{\partial v_i} + \frac{\partial p}{\partial \mathbf{T}}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}\right] \mathrm{d}t \tag{3.6}$$

The direct method calculates the sensitivities by calculating $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$ and substituting it in the integral in equation 3.6. The vector $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$ is calculated by differentiating equation 3.1 with respect to the design variables v_i:

$$\mathbf{C}\frac{\mathrm{d}\dot{\mathbf{T}}}{\mathrm{d}v_i} + \mathbf{K}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} = \mathbf{R}_i \tag{3.7}$$

where $\mathbf{R}_i$ is:

$$\mathbf{R}_i(t) = \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\mathbf{T} - \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\dot{\mathbf{T}} \tag{3.8}$$

In combination with the initial condition $\frac{\mathrm{d}\mathbf{T}(0)}{\mathrm{d}v_i} = \mathbf{0}$, which follows from equation 3.2, this equation can, for instance, be solved with time stepping algorithms (see section 3.3). Equation 3.7 needs to be resolved for every design variable in order to obtain $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$. In the case of topology optimization, where there are many design variables, this is computationally expensive.

3.1.3. Adjoint method

Premultiplying equation 3.7 with the transpose of the adjoint vector λ (which is a column vector) and integrating by parts over time gives:

$$\lambda^T\mathbf{C}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}\bigg|_0^{t_f} + \int_0^{t_f} \left[\lambda^T\mathbf{K} - \dot{\lambda}^T\mathbf{C}\right]\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}\mathrm{d}t = \int_0^{t_f} \lambda^T\mathbf{R}_i\mathrm{d}t \tag{3.9}$$

Now the aim of the adjoint method is to perform the following substitution in equation 3.6:

$$\frac{\partial p}{\partial \mathbf{T}}\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i} = \lambda^T\mathbf{R}_i \tag{3.10}$$

Consequently, so we avoid calculating the term $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$ which needs to be calculated for every design variable. In order for this substitution to hold the following conditions need to be satisfied (note that

$\frac{\partial \mathbf{T}(0)}{\partial v_i} = \mathbf{0}$):

$$\lambda^T \mathbf{K} - \dot{\lambda}^T \mathbf{C} = \frac{\partial p}{\partial \mathbf{T}} \tag{3.11}$$

$$\lambda(t_f) = \mathbf{0} \tag{3.12}$$

Equation 3.11 is integrated backwards in time because the terminal condition (equation 3.12) is known. Concluding, the adjoint method transforms equation 3.6 into:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i} = \int_0^{t_f} \left[\frac{\partial p}{\partial v_i} + \lambda^T \mathbf{R}_i \right] \mathrm{d}t \tag{3.13}$$

Assuming that the derivatives $\frac{\partial \mathbf{K}}{\partial v_i}$ and $\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}$ are easy to compute, this can save a lot of time, because the adjoint vector λ is independent of the design variable to which the performance measure is derived. λ only needs to be computed once for every performance measure, so with few constraints and many design variables this method is more efficient than the direct method. Hence, the adjoint method is preferred in TO.

3.2. Interpretation of the adjoint

In order to understand the properties of the adjoint variables better, it would be nice to have a physical interpretation of the adjoint variable instead of it being merely a mathematical trick. This insight might also lead to other approaches on how to calculate the adjoint. This section focusses on the interpretation of the adjoint in transient heat conduction problems.

3.2.1. Deriving the adjoint method from the direct method

In literature, the adjoint vector is usually explained as a Lagrange multiplier for the state equation that is added to the objective function, which was identified by Belegundu and Arora [16]. Whether the adjoint is added by simply multiplying an equation with the adjoint (as done in [6], equation 3.9) or adding it as a Lagrange multiplier, at the end of the day the only difference between the direct and the adjoint method is the order in which the matrix operations are performed. This was already stated by Vanderplaats [7] as a comment on the article of Arora and Haug [17] when they first presented the adjoint method for structural problems. In this section the difference between the direct and the adjoint method is shown by only changing the order of integration, without premultiplying or adding terms. This provides a clear interpretation of the adjoint.

Explicit solution for the temperature sensitivity

In order to show this difference between the direct and the adjoint method, a explicit solution for the temperature sensitivity is needed. This is done in much the same way as done in Rixen [18] to derive a general solution for the equations for modal amplitudes in modal superposition for dynamic mechanics. Starting, a Laplace transformation can be used to find a solution for the temperature sensitivity. The Laplace transform of the temperature field and temperature sensitivities are defined as:

$$\bar{\mathbf{T}}(s) = \mathcal{L}(\mathbf{T}(t)) = \int_0^\infty e^{-st} \mathbf{T}(t) \mathrm{d}t \tag{3.14}$$

$$\frac{\mathrm{d}\bar{\mathbf{T}}(s)}{\mathrm{d}v_i} = \mathcal{L}(\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i}) = \int_0^\infty e^{-st} \frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i} \mathrm{d}t \tag{3.15}$$

The Laplace transform of equation 3.7 is then:

$$\mathbf{C}\left(s\frac{\mathrm{d}\bar{\mathbf{T}}(s)}{\mathrm{d}v_i} - \frac{\mathrm{d}\bar{\mathbf{T}}(0)}{\mathrm{d}v_i} \right) + \mathbf{K}\frac{\mathrm{d}\bar{\mathbf{T}}(s)}{\mathrm{d}v_i} = \bar{\mathbf{R}}_i(s) \tag{3.16}$$

where the Laplace transform of $\mathbf{R}_i(t)$, $\bar{\mathbf{R}}_i(s)$ is:

$$\bar{\mathbf{R}}_i(s) = \frac{\mathrm{d}\bar{\mathbf{Q}}(s)}{\mathrm{d}v_i} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\bar{\mathbf{T}}(s) - \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\left(s\bar{\mathbf{T}}(s) - \bar{\mathbf{T}}(0) \right) \tag{3.17}$$

and $\frac{\mathrm{d}\bar{\mathbf{Q}}(s)}{\mathrm{d}v_i}$ is the Laplace transform of $\frac{\mathrm{d}\mathbf{Q}(t)}{\mathrm{d}v_i}$. Solving for $\frac{\mathrm{d}\bar{\mathbf{T}}}{\mathrm{d}v_i}(s)$ gives:

$$\frac{\mathrm{d}\bar{\mathbf{T}}(s)}{\mathrm{d}v_i} = \left(s\mathbf{I} + \mathbf{C}^{-1}\mathbf{K}\right)^{-1}\left(\frac{\mathrm{d}\bar{\mathbf{T}}(0)}{\mathrm{d}v_i} + \mathbf{C}^{-1}\bar{\mathbf{R}}_i(s)\right) \tag{3.18}$$

Performing the inverse Laplace transformation gives the solution in the time domain. The multiplication of functions in the Laplace domain in equation 3.18 results in a convolution integral in the time domain. With the definition:

$$\mathbf{H}(t) = \mathcal{L}^{-1}\left(\left(s\mathbf{I} + \mathbf{C}^{-1}\mathbf{K}\right)^{-1}\right) = e^{-\mathbf{C}^{-1}\mathbf{K}t} \tag{3.19}$$

This gives:

$$\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i} = \mathbf{H}(t)\cancelto{0}{\frac{\mathrm{d}\mathbf{T}(0)}{\mathrm{d}v_i}} + \int_0^t \left[\mathbf{H}(t-\tau)\mathbf{C}^{-1}\mathbf{R}_i(\tau)\right]\mathrm{d}\tau \tag{3.20}$$

$$= \int_0^t \left[\mathbf{H}(t-\tau)\mathbf{C}^{-1}\mathbf{R}_i(\tau)\right]\mathrm{d}\tau \tag{3.21}$$

Although the initial value of the temperature sensitivity is zero in equation 3.20, the first term shows that the function $\mathbf{H}(t)$ is a time response to the initial condition. The integral can be seen as a summation of the responses to all the loads $\mathbf{Q}$ from time 0 to t. Because of the exponential nature of the response $\mathbf{H}(t)$, it can be seen that the effect of a load on a response decays when it is longer ago (as $t-\tau$ increases). For the response at a certain time t, the function $\mathbf{H}(t)$ has the exponential form depicted in figure 3.1a.

The direct method

The function for the temperature sensitivity can be directly substituted into equation 3.6 to get an explicit formulation for the sensitivity of the constraint or objective function:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i} = \int_0^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial v_i} + \frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}\int_0^t \left[\mathbf{H}(t-\tau)\mathbf{C}^{-1}\mathbf{R}_i(\tau)\right]\mathrm{d}\tau\right]\mathrm{d}t \tag{3.22}$$

This is what is normally called the direct method. On each time instant t during the interval $[0, t_f]$, the temperature sensitivity is calculated and multiplied with the derivative of the objective to the temperature, after which the sensitivity of the objective or constraint function on that time instant can be determined.

Swapping integrals

To move from the direct to the adjoint method, the design variable dependent functions $\mathbf{C}^{-1}$ and $\mathbf{R}_i$ have to be moved outside the inner integral to make it independent of the design variables. As a means to accomplish this, the integrals in equation 3.22 have to be reordered. The steps needed for this reordering are written down in this paragraph. For more mathematical detail on these steps the reader is referred to Priestley [19]. In this paragraph we only consider the 2nd term of the direct method (equation 3.22), which we name $\frac{\mathrm{d}g}{\mathrm{d}v_i}^*$, the other terms are not affected by the process:

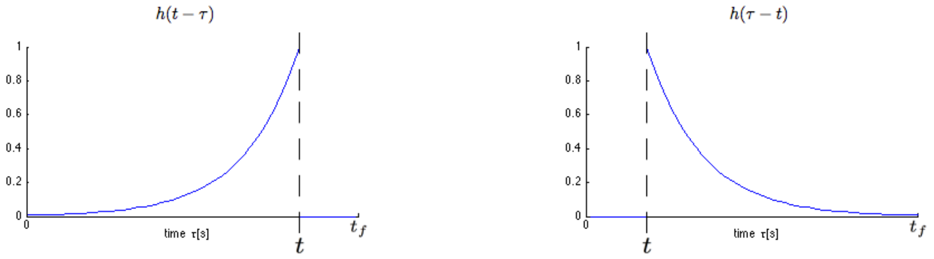
$$\frac{\mathrm{d}g}{\mathrm{d}v_i}^* = \int_0^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}\int_0^t \left[\mathbf{H}(t-\tau)\mathbf{C}^{-1}\mathbf{R}_i(\tau)\right]\mathrm{d}\tau\right]\mathrm{d}t \tag{3.23}$$

In order to make the upper limit of the nested integral independent of t, a function $k(t,\tau)$ is added that has the properties:

$$k(t,\tau) = \begin{cases} 1 & : \tau \le t \\ 0 & : \tau > t \end{cases} \tag{3.24}$$

This way the upper limit can be altered. Also the term $\frac{\partial p}{\partial \mathbf{T}}$, which is independent of τ, is put inside the nested integral:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i}^* = \int_0^{t_f}\int_0^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}\mathbf{H}(t-\tau)\mathbf{C}^{-1}\mathbf{R}_i(\tau)k(t,\tau)\right]\mathrm{d}\tau\mathrm{d}t \tag{3.25}$$



(a) The function $h(t-\tau)$ as used in the direct method grows exponentially until time instant t. It gives the contributions of the loads on the preceding time instances on the response on time t. As one can imagine, loads that are longer ago have a smaller influence.

(b) The function $h(\tau-t)$ as used in the adjoint method decays exponentially from time instant t. It gives the contribution of the load at time t on the responses of the succeeding time instances. As one can imagine, the load has a smaller influence on responses that are further away in time.

Figure 3.1: The different shapes of the impulse response function $h(t)$ in the direct and the adjoint method.

Because the function inside the integrals is piecewise continuous and the bounds are finite, the order of integration can be swapped according to Fubini's theorem [19]:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i}^* = \int_0^{t_f}\int_0^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}\mathbf{H}(t-\tau)\mathbf{C}^{-1}\mathbf{R}_i(\tau)k(t,\tau)\right]\mathrm{d}t\mathrm{d}\tau \tag{3.26}$$

We can remove the added function $k(t,\tau)$ by altering the bounds of the inner integral. Since k is only equal to 1 when $t \geq \tau$, we can remove it by setting the lower bound of the inner integral to τ. This way the condition $t \geq \tau$ is always met. We can also move the part that is independent of t to the outer integral:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i}^* = \int_0^{t_f}\int_\tau^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}\mathbf{H}(t-\tau)\right]\mathrm{d}t\mathbf{C}^{-1}\mathbf{R}_i(\tau)\mathrm{d}\tau \tag{3.27}$$

Finally we can change the variables: rename t to τ and vice versa, to complete the swapping of integrals:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i}^* = \int_0^{t_f}\int_t^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(\tau),\tau)}{\partial \mathbf{T}}\mathbf{H}(\tau-t)\right]\mathrm{d}\tau\mathbf{C}^{-1}\mathbf{R}_i(t)\mathrm{d}t \tag{3.28}$$

The adjoint method

By adding the terms that were left out during the swap, we again get an explicit formula for the sensitivity of the constraint or objective function:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i} = \int_0^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial v_i} + \underbrace{\int_t^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(\tau),\tau)}{\partial \mathbf{T}}\mathbf{H}(\tau-t)\right]\mathrm{d}\tau\mathbf{C}^{-1}}_{\lambda^T(t)}\mathbf{R}_i(t)\right]\mathrm{d}t \tag{3.29}$$

This is what is normally called the adjoint method. Comparing it with the adjoint method as derived in Haftka [6] (equation 3.13), it can be seen that exactly the same terms are present. The function that is derived for the adjoint, $\lambda(t)$, could also be derived by solving equation 3.11 for $\lambda(t)$ with the Laplace transform.

3.2.2. Interpretation of the adjoint

From equation 3.29 an interpretation of the adjoint in transient heat conduction problems can be derived. The adjoint at time t consists of a term $\mathbf{C}^{-1}$, multiplied with an integral over the interval $[t,t_f]$, the remaining time in the time frame of interest. There are two terms in the integral, the time response of the system to an impulse at time t, and the derivative of the objective function to the

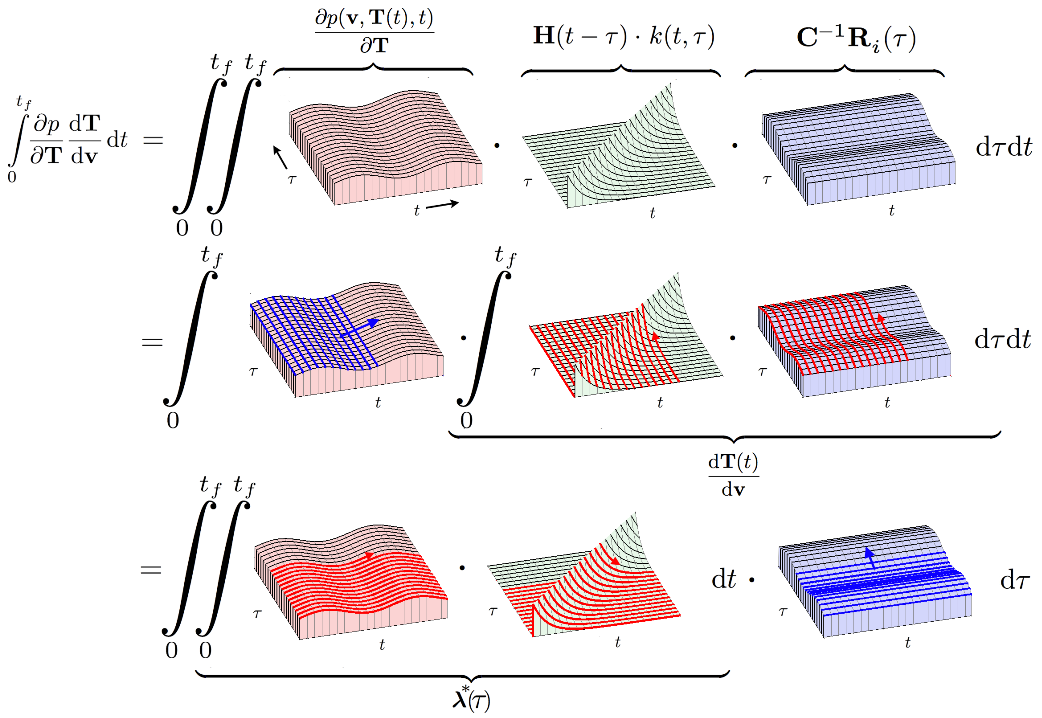

$\int_0^{t_f} \frac{\partial p}{\partial \mathbf{T}} \frac{\mathrm{d}\mathbf{T}}{\mathrm{d}\mathbf{v}} \mathrm{d}t =$
$\int_0^{t_f}\int_0^{t_f}$
$\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}$
$\mathbf{H}(t-\tau)\cdot k(t,\tau)$
$\mathbf{C}^{-1}\mathbf{R}_i(\tau)$
$\mathrm{d}\tau \mathrm{d}t$
τ
t
$= \int_0^{t_f}$
$\int_0^{t_f}$
$\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}\mathbf{v}}$
$\mathrm{d}t$
$\mathrm{d}\tau$
$\boldsymbol{\lambda}^*(\tau)$

Figure 3.2: Visualization of equation 3.25 (the top row) for a one degree of freedom system, which is then put into the direct (middle) and adjoint (bottom) form. Note the change in order of integration between the direct and the adjoint method. The red lines indicate the order the nested integral is integrating in, the blue lines the direction of the outer integral.

temperature. Because the order of t and τ is swapped in the response function $\mathbf{H}(t)$ compared to the direct method, the integral in the adjoint does not give the summation of the contributions of loads from $t = 0$ up to time t, but a summation of the contribution to the change of objective function of a single load on the remaining time from t until end time t_f. This is also depicted in figure 3.1b. The integral can therefore be seen as the total contribution of the input $\mathbf{R}$ at time t to the objective function change on the remaining time interval. The term $\mathbf{C}^{-1}$ is included in the adjoint by the definition of the substitution in [6] (equation 3.10). However, since the term is outside the integral it would make more sense not to include it in the adjoint. This would have made the adjoint purely a convolution term. Summarizing, the adjoint $\lambda(t)$ at time t, is a measure for the significance of input $\mathbf{R}(t)$ at time t, representing a pseudo load term for the change in temperature due to a change in design variable, on the objective function derivative in the remaining time interval.

A visualization of the direct and the adjoint method

During the swapping of the integrals, in equation 3.25 it looks like the derivative of the objective function is some kind of a surface integral (with dimensions t and τ) of three terms multiplied with each other. These terms are:

1. $\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}$, the objective function derived w.r.t. temperature, constant in τ but dependent on t

2. $\mathbf{C}^{-1}\mathbf{R}_i(\tau)$, the loading terms, constant in t but dependent on τ

3. $\mathbf{H}(t-\tau)k(t,\tau)$, the impulse response function, dependent on both t and τ

For a one degree of freedom system, these functions can actually be visualised as surfaces. Because the first two functions are independent of one of the variables, they can be taken outside of one of the integrals. In order to take the first term, $\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial \mathbf{T}}$, outside of one of the integrals, the nested integral needs to be over τ since term 1 is constant in τ. This is the direct method. In order to take the second term, $\mathbf{R}_i(\tau)\mathbf{C}^{-1}$, outside of one of the integrals, the nested integral needs to be over t since term 2 is constant in t. This is the adjoint method. When term 1 is put outside of the nested integral, the nested integral represents $\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i}$, while when term 2 is put outside the nested integral, it represents the adjoint $\lambda^*(t) = \lambda(t)\mathbf{C}$. This process is visualized for a one degree of freedom system in figure 3.2. In the figure it can be seen that for the direct method, the impulse response function increases with τ, while multiplying with the corresponding inputs, which gives the same shape as depicted in figure 3.1a for the response function. Because in the adjoint method the response function is integrated in the other direction, the impulse function now decreases with increasing t, as depicted in figure 3.1a. The representation as surfaces also makes the step from equation 3.25 to 3.26, where the integrals are swapped, more intuitive. After all, it does not matter which direction is integrated first in order to cover the whole surface.

3.3. Methods of integration

When the temperature field has been calculated, the derivative of the performance measures can be calculated via either the direct or the adjoint method as given in equations 3.22 and 3.29, respectively. The direct method doesn't need any information of future time instances for the derivative of the temperature at a time instance. Consequently, it can be calculated simultaneously with the temperature. However, doing this will become very costly because the nested integrals have to be re-evaluated every time step. However they can easily be put in a recursive form, which is the most obvious method for calculating the objective function derivative. By using a recursive form, the solution calculated at a previous time step can be reused and only integration over the time between the two time steps is needed.

3.3.1. Recursive calculation of the sensitivities

The direct method

The temperature sensitivity (equation 3.21) can be calculated recursively by deriving the following relation from equation 3.21:

$$\frac{\mathrm{d}\mathbf{T}(t+\Delta t)}{\mathrm{d}v_i} = \mathbf{H}(\Delta t)\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i} + \int_t^{t+\Delta t} \left[\mathbf{H}(t+\Delta t-\tau)\mathbf{C}^{-1}\mathbf{R}_i(\tau)\right] \mathrm{d}\tau \tag{3.30}$$

With the initial condition $\frac{\mathrm{d}\mathbf{T}(0)}{\mathrm{d}v_i} = \mathbf{0}$. Equation 3.30 can also be easily discretized.

The adjoint method

The adjoint function can be calculated recursively by deriving the following relation from equation 3.29:

$$\lambda^T(t+\Delta t) = \lambda^T(t)\mathbf{C}\mathbf{H}(-\Delta t)\mathbf{C}^{-1} - \int_t^{t+\Delta t} \left[\frac{\partial p(\mathbf{v},\mathbf{T}(\tau),\tau)}{\partial \mathbf{T}}\mathbf{H}(\tau-t-\Delta t)\right] \mathrm{d}\tau\mathbf{C}^{-1} \tag{3.31}$$

Now the problem is that $\lambda(0)$, unlike $\frac{\mathrm{d}\mathbf{T}(0)}{\mathrm{d}v_i}$ which is defined beforehand, is unknown, so the recursive sequence can not be initiated. However, since the adjoint at time t represents the influence the pseudo load $\mathbf{R}$ on time t has on the remaining time frame, we know that $\lambda(t_f) = 0$, because there is no time frame left so it will have no influence. We can rewrite the recursive sequence to go backwards in time:

$$\lambda^T(t-\Delta t) = \lambda^T(t)\mathbf{C}\mathbf{H}(\Delta t)\mathbf{C}^{-1} + \int_{t-\Delta t}^{t} \left[\frac{\partial p(\mathbf{v},\mathbf{T}(\tau),\tau)}{\partial \mathbf{T}}\mathbf{H}(\tau-t+\Delta t)\right] \mathrm{d}\tau\mathbf{C}^{-1} \tag{3.32}$$

Which can be solved with $\lambda(t_f) = 0$.

3.3.2. Integration methods for limited forms of the objective or constraint function

Explicit function for the adjoint

When the objective function is linear in temperature, the adjoint can be solved explicitly. This creates a second method to calculate the adjoint directly with the objective function derivative. For this example it is assumed that the objective function is independent of time (otherwise integration by parts needs to be used). The adjoint is then:

$$\lambda^T(t) = \frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}} \int_t^{t_f} \mathbf{H}(\tau-t)\mathrm{d}\tau\mathbf{C}^{-1} = \frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}}\mathbf{K}^{-1}\mathbf{C}\left(e^{-\mathbf{C}^{-1}\mathbf{K}(t_f-t)} - \mathbf{I}\right)\mathbf{C}^{-1} \tag{3.33}$$

With this explicit function for the adjoint, the sensitivity of the objective function can be calculated directly along with the temperature, forward in time. However, the matrix exponential that is part of this function is also very computationally expensive to calculate for large matrices, so this is not a good option in general.

The adjoint forward in time

This explicit formulation can also be used once to calculate an initial condition for the adjoint at $t = 0$, which enables the use of forward time integration with equation 3.31. This is only slightly slower than backwards in time, but now the sensitivities can also be calculated simultaneously with the temperature field by using the adjoint method. This eliminates the need to store all the temperatures and could possibly save memory. However, one should be careful when using this method because a small offset in the calculated initial condition can have a large influence on the accuracy. That is because when the time frame is long enough, the adjoint is almost stationary around $t = 0$, and therefore small changes in the adjoint represent large changes in time.

Explicit function for the sensitivities

However these ”explicit” function still require time stepping which can be computationally expensive for long time frames. The observant reader might already have seen the similarity between equation 3.1 and equation 3.7. The only difference is that the temperature $\mathbf{T}$ is replaced with its derivative $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$, and the heat load $\mathbf{Q}$ with $\mathbf{R}_i$, a loading term for the change in temperature due to a change in design variable. This means that the explicit equation 3.20 for the derivatives $\frac{\mathrm{d}\mathbf{T}}{\mathrm{d}v_i}$ can also be used as an explicit formula for the temperatures $\mathbf{T}$, if the load $\mathbf{R}_i$ is changed to load $\mathbf{Q}$. In this example it is assumed that the heat load $\mathbf{Q}$ is independent of time, otherwise again integration by parts needs to be used:

$$\begin{aligned}
\mathbf{T}(t) &= \mathbf{H}(t)\mathbf{T}(0) + \int_0^t \mathbf{H}(t-\tau)\mathrm{d}\tau\, \mathbf{C}^{-1}\mathbf{Q} && (3.34)\\
&= \mathbf{H}(t)\mathbf{T}(0) + \left[\left(\mathbf{C}^{-1}\mathbf{K_T}\right)^{-1}\mathbf{H}(t-\tau)\right]_0^t \mathbf{C}^{-1}\mathbf{Q} && (3.35)\\
&= \mathbf{H}(t)\mathbf{T}(0) + \mathbf{K_T}^{-1}\mathbf{C}\left(\mathbf{I}-\mathbf{H}(t)\right)\, \mathbf{C}^{-1}\mathbf{Q} && (3.36)\\
&= \mathbf{H}(t)\mathbf{T}(0) - \mathbf{K_T}^{-1}\mathbf{C}\mathbf{H}(t)\mathbf{C}^{-1}\mathbf{Q} + \mathbf{K_T}^{-1}\mathbf{Q} && (3.37)
\end{aligned}$$

This can be differentiated to time to get an explicit formula for $\dot{\mathbf{T}}$:

$$\begin{aligned}
\dot{\mathbf{T}}(t) &= \dot{\mathbf{H}}(t)\mathbf{T}(0) - \mathbf{K_T}^{-1}\mathbf{C}\dot{\mathbf{H}}(t)\mathbf{C}^{-1}\mathbf{Q} && (3.38)\\
&= -\mathbf{C}^{-1}\mathbf{K_T}\mathbf{H}(t)\mathbf{T}(0) + \mathbf{H}(t)\mathbf{C}^{-1}\mathbf{Q} && (3.39)
\end{aligned}$$

Combined this gives an explicit formula for $\mathbf{R}_i$, which only time dependent terms are $\mathbf{T}$, $\dot{\mathbf{T}}$. Substituting the explicit adjoint equation 3.33 in the adjoint formulation (3.29) gives:

$$\frac{\mathrm{d}g}{\mathrm{d}v_i} = \int_0^{t_f}\left[\frac{\partial p(\mathbf{v},\mathbf{T}(t),t)}{\partial v_i} + \frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}}\mathbf{K}^{-1}\mathbf{C}\left(\mathbf{H}(t_f-t)-\mathbf{I}\right)\mathbf{C}^{-1}\left(\frac{\mathrm{d}\mathbf{Q}(t)}{\mathrm{d}v_i} - \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\mathbf{T}(t) - \frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\dot{\mathbf{T}}(t)\right)\right]\mathrm{d}t \tag{3.40}$$

Rearranging and assuming that the objective function is not directly dependent on the design variables gives:

$$\begin{aligned}
\frac{\mathrm{d}g}{\mathrm{d}v_i} = \frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}}\Bigg[&-\mathbf{K}^{-1}\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i}t_f + \mathbf{K}^{-1}\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\int_0^{t_f}\mathbf{T}(t)\mathrm{d}t + \mathbf{K}^{-1}\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\int_0^{t_f}\dot{\mathbf{T}}(t)\mathrm{d}t\\
&+\mathbf{K}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\,\mathrm{d}t\,\mathbf{C}^{-1}\frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i}\\
&-\mathbf{K}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\mathbf{C}^{-1}\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\mathbf{T}(t)\,\mathrm{d}t\\
&-\mathbf{K}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\mathbf{C}^{-1}\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\dot{\mathbf{T}}(t)\,\mathrm{d}t\Bigg]
\end{aligned} \tag{3.41}$$

The integrals $\int_0^{t_f} \mathbf{T}(t)\mathrm{d}t$, $\int_0^{t_f} \dot{\mathbf{T}}(t)\mathrm{d}t$ and $\int_0^{t_f} \mathbf{H}(t_f - t)\mathrm{d}t$ can easily be evaluated as they only involve isolated terms of $\mathbf{H}(t)$, which is simply the integral of an exponent. The last two integrals involve the terms $\mathbf{H}(t_f - t)$ and $\mathbf{H}(t)$ in the same integral, making the difficult to solve. The last two lines of equation 3.41 are expanded by substituting 3.37 and 3.39:

$$\begin{aligned}
&\mathbf{K}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\mathbf{C}^{-1}\left(\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\mathbf{K_T}^{-1}\mathbf{C}-\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\right)\mathbf{H}(t)\,\mathrm{d}t\,\mathbf{C}^{-1}\mathbf{Q}\\
+&\mathbf{K}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\mathbf{C}^{-1}\left(\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\mathbf{C}^{-1}\mathbf{K_T}-\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\right)\mathbf{H}(t)\,\mathrm{d}t\,\mathbf{T}(0)\\
-&\mathbf{K}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\,\mathrm{d}t\,\mathbf{C}^{-1}\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}\mathbf{K_T}^{-1}\mathbf{Q}
\end{aligned} \tag{3.42}$$

The last integral is again easy to solve. However the first two integrals are difficult to solve because of the matrix in between the exponents $\mathbf{H}(t)$. There are methods proposed to solve systems of the form $\int_0^{t_f} e^{\mathbf{A}t}\mathbf{B}e^{-\mathbf{A}t}\mathrm{d}t$. The method proposed by Loan [20] can be easily altered to calculate the integral. However this involves calculating a matrix exponential of a composed matrix containing $\mathbf{A}$ and $\mathbf{B}$. Because $\mathbf{B}$ is dependent on the design variable to which is differentiated, this has to be done for every design variable which makes it inefficient. A faster method by which the matrices $\frac{\mathrm{d}\mathbf{K}}{\mathrm{d}v_i}$ and $\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}$ can be extracted out of the integrals is explained in Appendix D. However this involves diagonalization of the matrix $\mathbf{C}^{-1}\mathbf{K}$. Because all the eigenvalues and eigenvectors have to be calculated for the diagonalization, this method will not be faster than solving the system using modal superposition, though no derivatives of eigenvalues or eigenvectors are needed. For large systems the practical use will be limited due to the eigenproblem that needs to be solved. However, for small systems which require small time steps over a long time frame, this explicit formulation will be faster, dependent on the ratio of degrees of freedom versus time steps. To get a rough idea the number of time steps for which the explicit and the adjoint method costs equal amounts of time have been plotted in figure 3.3. Bare in mind that there are many other variables such as matrix sparsity and the algorithm that is used that influence the time cost, so do not use the graph quantitatively.

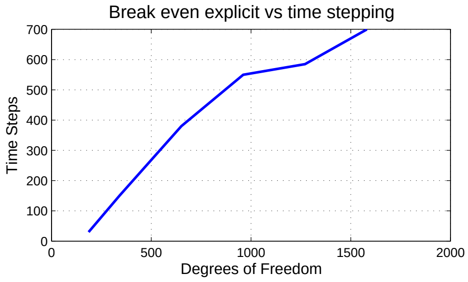
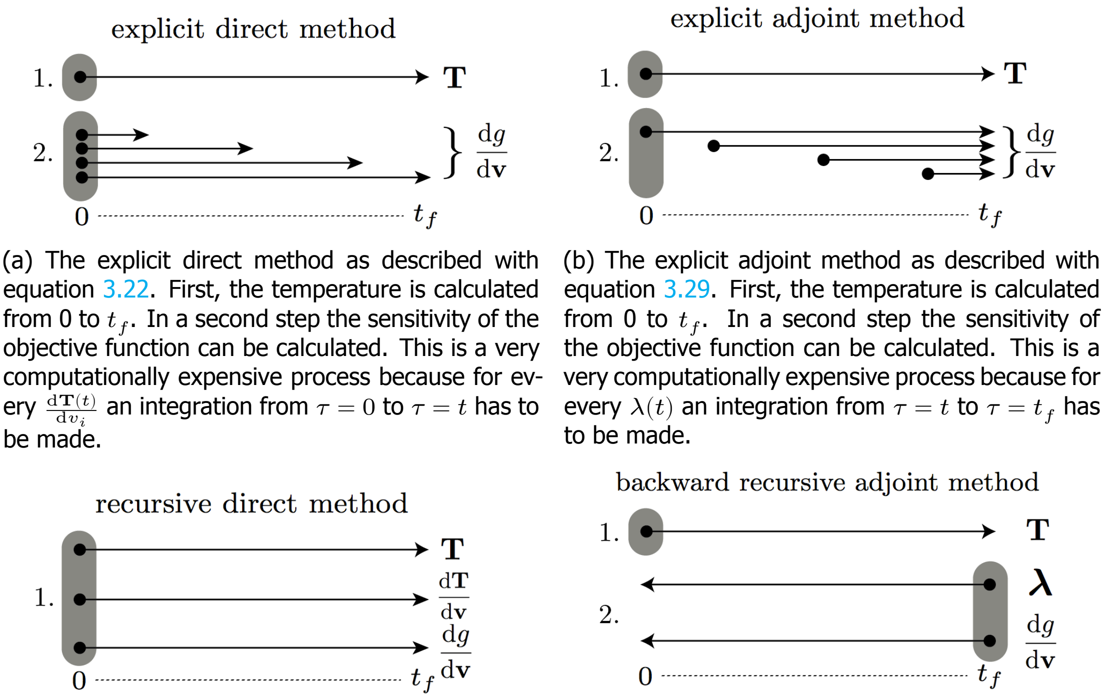


Figure 3.3: Number of time steps for which the explicit calculation costs the equal amount of time as the adjoint method. This only gives a rough idea, since it depends on many more parameters.

3.4. Conclusions

In this chapter it is shown that the difference between the direct and the adjoint method is that the direct method calculates for each time instance what the influence of the preceding loadings is on that time instance, while the adjoint method calculates for each time instance how much influence the load on that time instance has on all the subsequent time instances. The two methods give rise to a number of integration methods. The recursive calculation of the direct and the adjoint method are the fastest methods. For objective functions linear in temperature, an explicit formula for the adjoint, and also for all the sensitivities, can be derived. These, however, include an expensive matrix exponential, which limits its use to small systems which have to be evaluated with a relatively large number of time steps.

The explicit and recursive direct and adjoint methods, discussed in section 3.3.1 have been visualized in figure 3.4. The explicit functions for the sensitivities discussed in section 3.3.2 have been visualized in figure 3.5.



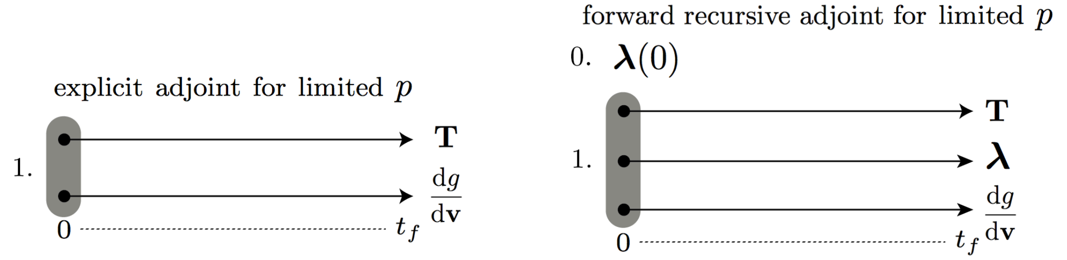
(a) The explicit direct method as described with equation 3.22. First, the temperature is calculated from 0 to t_f. In a second step the sensitivity of the objective function can be calculated. This is a very computationally expensive process because for every $\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i}$ an integration from $\tau = 0$ to $\tau = t$ has to be made.

(b) The explicit adjoint method as described with equation 3.29. First, the temperature is calculated from 0 to t_f. In a second step the sensitivity of the objective function can be calculated. This is a very computationally expensive process because for every $\lambda(t)$ an integration from $\tau = t$ to $\tau = t_f$ has to be made.

(c) The recursive direct method (equation 3.30). Becuase $\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i}$ only needs $\mathbf{T}(t)$, it can be calculated along with the temperature. In this way nothing has to be stored in memory and everything can be calculated in the same run. The method is fast but $\frac{\mathrm{d}\mathbf{T}(t)}{\mathrm{d}v_i}$ needs to be re calculated for every design variable.

(d) The backward recursive adjoint method (equation 3.32). Becuase $\lambda(t)$ is calculated backwards in time, two separate runs are needed and the temperatures need to be stored. However $\lambda(t)$ does only need to be recalculated for every objective or constraint function.

Figure 3.4: Overview of the different methods discussed in section 3.3.1 to calculate the objective function sensitivity.


explicit adjoint for limited p
1.
T
dg/dv
0
t_f
forward recursive adjoint for limited p
0. $\boldsymbol{\lambda}(0)$
1.
T
λ
dg/dv
0
t_f

(a) The explicit adjoint for limited forms of the objective function (equation 3.33). Because of the explicit function for the adjoint everything can be calculated in one run saving memory, however the needed matrix exponential is computationally expensive to calculate.

(b) The forward recursive adjoint method for limited forms of the objective function (equation 3.31). This method needs only one matrix exponential, making it only slightly slower than the adjoint backward method. Because the adjoint can now be calculated forward in time, everything can be done in one run and no temperatures need to be stored. However care should be taken with accuracy.

Figure 3.5: Overview of the different methods discussed in section 3.3.2 to calculate the objective function sensitivity for special forms of the objective function.

4

New penalization method: material penalization

The density approach used in this study for TO (as explained in chapter 2), allows intermediate densities in the design (densities between 0 and). It is important that the final solution of a TO does not contain any intermediate densities to ensure manufacturability and interpretation of the design. In 2D problems, the thickness can sometimes be used to mimic intermediate densities, but often in 2D, and generally in 3D problems intermediate densities are not manufacturable.

For stiffness optimizations, the combination of a volume constraint and material interpolation mostly leads to black and white designs, without intermediate densities. However, for multiphysics problems, this is often not the case. The focus of this study lies on transient-thermal-mechanical (TTM) problems. Only very few TTM topology optimizations have been found in literature using the density approach, most notably the ones of Li *et al.* [22] and more recently Mello *et al.* [23]. Both have final designs that have intermediate densities. Quite a few (non transient) thermal-mechanical optimization using the density approach have been done. Nevertheless, also here intermediate densities are present. For example in the paper of Cho and Choi [21] and Yin and Ananthasuresh [12].

There are methods to suppress intermediate densities. Grey penalization is a widely applied method. It adds a term to the objective function, which increases the objective function for intermediate densities. Depending on the scaling with the original objective, this can suppress the intermediate densities, but in most cases it pushes the densities that are greater than 0.5 to 1, and those smaller than 0.5 to 0. This is likely to destroy the performance of a design because it might rely on elements with an intermediate density. Grey penalization has been used in Yin and Ananthasuresh [12] for example. Nonetheless, it was not able to produce a black and white design for every design case. Other methods available are projection methods, which are post processing steps. This has been used in Mello *et al.* [23], where a threshold was set at 0.5, above which the densities were set to 1. The thresholded design has a much lower performance, because there is no guarantee that a post processed design is an optimum in any sense.

In this chapter a new penalization method, material penalization, is proposed that produces black and white designs for TTM problems. First, the new penalization method, material penalization, is formulated in section 4.1. Then, it is tested on two simple test cases in section 4.2 and its limitation are discussed in section 4.3. Material penalization is tested on a larger test case in section 4.5. Finally, conclusions and recommendation are given in section 4.6.

4.1. The material penalization method

In stiffness optimizations, penalization is added to obtain a black and white design. For high penalization factors, the interpolated Young's modulus has a low stiffness to volume ratio for intermediate densities. This means that the element costs a lot of volume compared to the stiffness it adds, and thus a high penalization factor makes intermediate densities less economical to use, if there is a volume constraint.

As stated in section 2.1.1, in TTM problems, there are four material properties that can be interpolated. Little research has been conducted on how to penalize the material properties, often they are all penalized with the same penalization factor. Sigmund [24] is, by the authors knowledge, the first to mention the possibility of penalizing the material properties by different factors for thermal-mechanical problems. However, the choice of different penalization factors is explicitly omitted, as he stated: *"In principle, the power p* [the penalization parameter] *could take different values for each physical property. However, for simplicity and to avoid having to choose multiple parameters, the same power is selected for all material interpolations."*. This is what is mostly done in (transient-)thermal-mechanical optimizations. Gao and Zhang [5] are the first to explore the effect of different penalizations on a mechanical optimization with a given temperature rise and finds that the thermal stress coefficient β should be penalized higher than the Young's modulus, thus $q_\beta > q_E$.

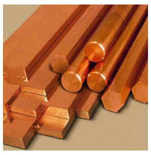
Outside of the thermal-mechanical domain, some research has been done on the effect of different penalization parameters. Kim *et al.* [25] claim to be the first to pose a set of rules for different penalization parameters for piezoelectric systems. In the paper, different penalization rules are found for different types of piezoelectric systems. Noh and Yoon [26] also investigate the effect of different penalization parameters on a piezoelectric energy harvesting device under static and dynamic loading. One of the remarks about the penalization factors in the paper is: *"Depending on the penalization values of p1, p2 and p3* [penalization parameters for different material properties] *very different layouts are obtained"*. Because these papers employ the electro-mechanical domain it is hard to draw quantitative conclusions for the thermal-mechanical domain. However, as in the electro-mechanical domain, different penalization parameters have influence on the final designs.

4.1.1. Additional materials at intermediate densities

Different combinations of penalization parameters give different designs, because the material properties of intermediate densities change with the penalization parameters. If all the material properties are interpolated by the same factor, the material properties of intermediate densities will be similar to the original material, except it will behave as if there is less material (e.g. a less thick plate in 2D). When the penalization parameters are changed relative to each other, the properties will change. Let q_X bet the RAMP penalization parameter for material property X (section 2.1.1). An element with $q_c = 0$, $q_k = 5$, $q_E = 0$ and $q_\beta = 0$ will have a lower conduction compared to the other material properties because the conduction is penalized more. This will give some kind of concrete like material which is stiff but has a low conduction. On the opposite side an element with $q_c = 5$, $q_k = 0$, $q_E = 5$ and $q_\beta = 0$ has a low stiffness but a high conduction. This could represent a type of copper wire connection. Table 4.1 gives an interpretation of intermediate materials at different penalization combinations, for the static case. That is, without considering heat capacity, in order not to overcomplicate the purely illustrative example.

For some design problems, the optimizer might find it optimal to conduct heat without adding stiffness. If some kind of copper wire material is available at half density, a lot of intermediate densities will be present in the final design due to their advantageous properties. However, if the intermediate material has properties that are adverse to the optimization problem, the final design will not contain

Table 4.1: The materials listed in this table are (clockwise): copper, SiSiC, copper wire, SiSiC foam, concreate, zerodur, oak and wool. These materials represent the special material properties of penalized materials at intermediate densities for different penalization factors.

	$q_E = 0$ $q_\beta = 0$	$q_E = 0$ $q_\beta = 5$	$q_E = 5$ $q_\beta = 0$	$q_E = 5$ $q_\beta = 5$
$q_k = 0$				
$q_k = 5$				

any intermediate densities.

4.1.2. Material penalization

The basic idea behind material penalization is: find a set of penalization parameters, that gives the material at intermediate densities adverse properties for the optimization problem. This will prevent the optimizer from using half densities as design elements.

As was shown in Kim *et al.* [25], different objectives can benefit from a different set of penalization parameters. It is therefore not possible to find a single set of penalization parameters that gives a black and white design for all TTM topology optimizations. Even within a single optimization, intermediate material that is adverse in one location can have favourable properties in other locations. In order to ensure a black and white design, each element should have its own set of penalization parameters that creates intermediate material that is adverse to the design problem for that specific element. Because the working principle of a design might change over iterations, these penalization parameters should be updated at every iteration.

Hence, material penalization adds for every element the penalization parameters as design variables. The penalization parameters are optimized simultaneous with the density variables, but the penalization parameters are optimized to maximize the objective function, whereas the density variables minimize the objective function. Maximizing will provide the set of penalization parameters for each element that gives adverse intermediate density material. These parameters will not change the material properties of the final design as it should consist of only black or white elements, which are not affected by the material penalization. The new optimization problem can be stated as a continuous min-max problem:

$$\min_{\rho_i} \max_{q_{s_j}} \left(f(\mathbf{u}(t)) = \int_0^{t_f} p(\mathbf{u}(t)) \mathrm{d}t \right) \tag{4.1}$$

$$\text{subject to: } \mathbf{C}(\rho_i, q_{s_j})\dot{\mathbf{T}} + \mathbf{K_T}(\rho_i, q_{s_j})\mathbf{T} = \mathbf{Q} \tag{4.2}$$

$$\mathbf{K_U}(\rho_i, q_{s_j})\mathbf{u} = \mathbf{A}(\rho_i, q_{s_j})\mathbf{T} \tag{4.3}$$

$$0 \leq \rho_i \leq 1 \qquad i \in [1, N] \tag{4.4}$$

$$-1 \leq q_{s_j} \leq 1 \qquad j \in [1, 4N] \tag{4.5}$$

where q_{s_j} is a vector of substitute penalization parameters. Since there are four penalization parameters per element, this vector has four times the length of the density vector. The real penalization parameters q_j are correlated to these substitutes via the relations:

$$q^*_{s_j} = c_1 q^3_{s_j} + c_2 q_{s_j} \tag{4.6}$$

$$q_j = \begin{cases} q^*_{s_j} & q_{s_j} \geq 0 \\ \frac{q^*_{s_j}}{-q^*_{s_j}+1} & q_{s_j} < 0 \end{cases} \tag{4.7}$$

The larger the penalization parameter in the RAMP function gets, the smaller the effect it has on changing the RAMP curve. Relation 4.6 was added to alleviate this effect by adding a cubic relation between the design variable and the true penalization parameter. This way the change of the RAMP curve is somewhat linear over the domain of the design variable. The constants are chosen as $c_1 = 13$ and $c_2 = 2$. Note that the sum of the two constants determines the maximum penalization. Furthermore, in the positive domain the penalization parameter can increase to infinity to create a more extreme interpolation, but in the negative domain the maximum penalization is at $q = -1$. This is solved with relation 4.7, by which the design variable can go to plus and minus infinity, while the penalization parameter will go to -1 and infinity, respectively. The RAMP curves for the different values of design variable q_s are plotted in figure 4.1.

4.1.3. Penalization parameter sensitivities

In order to solve the new optimization problem with a gradient based optimizer, the sensitivities of the objective function towards the penalization parameters also need to be calculated. Again an adjoint objective can be defined and derived with respect to the penalization parameters as done in equation

2.16 and 2.24 for the element densities. Because both the element densities and element penalization parameters only appear explicitly in the FE matrices, the result up to the derivation of the FE matrices w.r.t. the density or the penalization parameters is exactly the same. Since no derivatives w.r.t. the densities appear in the definition for the adjoint, the adjoint as caclulated for the density variables is also the adjoint for the penalization parameters. Only the matrix derivatives need to be recalculated to obtain the penalization parameter derivatives:

$$\frac{\mathrm{d}f}{\mathrm{d}q_i} = \frac{\mathrm{d}f^*}{\mathrm{d}q_i} = \int_0^{t_f} \left[\lambda^T \left(\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}q_i}\mathbf{u} - \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}q_i}\mathbf{T} \right) + \mu^T \left(\frac{\mathrm{d}\mathbf{C_T}}{\mathrm{d}q_i}\dot{\mathbf{T}} + \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}q_i}\mathbf{T} - \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}q_i} \right) \right] \mathrm{d}t \tag{4.8}$$

This means that the penalization sensitivities come almost for free once the adjoint has been calculated, as the remaining FE matrix derivatives are known analytically. In the code used for this study, the calculation of the penalization parameter sensitivities costs about 1% of the time of the total sensitivity calculation.

4.1.4. Optimization algorithm

The penalization parameters are optimized using linear programming in combination with move limits. Because the penalization parameters are only bounded between -1 and 1, and not constrained, this is also a computationally cheap process. The optimizer basically moves the penalization variables in the direction of their derivatives with a step the size of their move limit. If two succeeding steps of a certain variable are in the same direction, the move limit of that variable is increased, while if the directions are opposite, indicating oscillations, the move limit is decreased. See also section 2.3.

4.2. Effect on the solution space

Intuitively it makes sense that by choosing disadvantageous intermediate materials, the optimizer will find a black and white solution. By studying the effect of changing penalization parameters on the solution space, the actual working principle of material penalization is found.

4.2.1. Creating concave functions

At the basis of the material penalization method lies the fact that a concave solution space will give a black and white design. That is because the minima of a concave function always lie at its boundaries, where the densities are either 0 or 1. If the penalization parameters can make the solution space for the density variables concave, the solution will be black and white. One of the properties of RAMP penalization is that it can make every strictly monotonically increasing or decreasing function either convex or concave, by tuning the penalization parameter, which is explained in appendix E. With RAMP penalization, the more concave a functions is, the higher the objective it has at intermediate densities (see figure 4.2). Maximizing the objective function by using the penalization design variables thus makes the solution space concave.

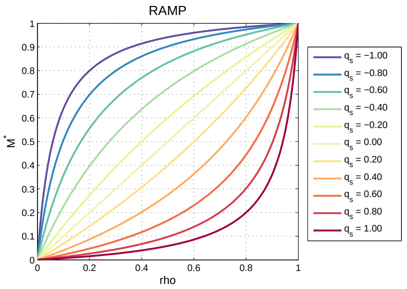


Figure 4.1: RAMP interpolation of a material property M for different penalization factors of the design variable q_s.

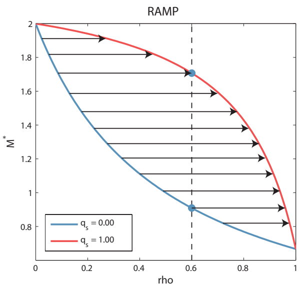


Figure 4.2: Ramp penalization applied to a convex function ($q_s = 0$) in order to make it more concave ($q_s = 1$). RAMP penalization translates the values of the original function, illustrated with the arrows. By making it more concave, the function value at a certain density will increase (displayed for $\rho = 0.6$).

4.2.2. Stiffness optimization

The effect of material penalization is first shown on a 2 element stiffness case. The stiffness case is depicted in figure 4.3. 2 elements lie on top of each other, with all their overlapping nodes rigidly connected to each other. A vertical force is applied and the compliance should be minimized. Out of plane rotations are not considered. Furthermore a volume constraint is applied with a maximum volume half of the total volume.

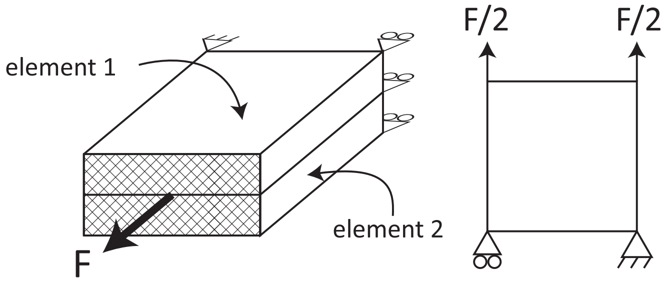


Figure 4.3: Stiffness case.

The solution space of the problem without penalization is shown in figure 4.4. It can be seen that stiffness optimization is a convex problem with an asymptote at $\rho_1 = \rho_2 = 0$, and the minimum lies at $\rho_1 = \rho_2 = 1$. This makes sense since more material adds more stiffness, and zero density means that the stiffness of both elements is at its lower limit. However the minimum lies in the infeasible domain. The edge of the feasible domain is horizontal, meaning that it does not matter for the objective in which element the material is placed, as long as both densities add up to 1. Depending on the initial design and the optimizer, the final design will end up somewhere along this line, with a high chance of containing grey densities.

Maximizing the objective by adding penalization, makes the solution space concave, as is plotted in figure 4.5. It can be seen that for every combination of intermediate densities the objective is higher than for the same combination in the unpenalized solution space. Now, there are only two minima in the feasible domain: at $\rho_1 = 0$ and $\rho_2 = 1$, or at $\rho_1 = 1$ and $\rho_2 = 0$. Depending on the initial design the optimizer will end up in either of the two. Note that because of the very low stiffness at $\rho_1 = \rho_2 = 0$, a huge penalization factor was needed to make the solution space concave. For this particular case, $q = 5 \cdot 10^9$ has been used. However, any penalization factor above 0 would have had the desired effect of creating two minima in the feasible domain, and this huge penalization parameter is only applied for illustrative purposes.

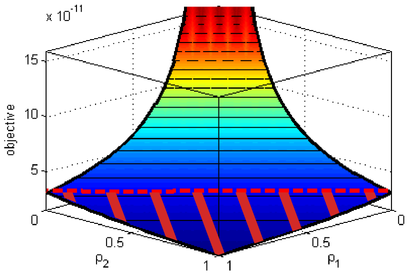


Figure 4.4: Solution space for the stiffness optimization case depicted in figure 4.3. No penalization has been applied. The red shaded area below the red dotted line is the infeasible domain, where the volume constraint is exceeded. There are infinitely many minima along the edge of the feasible domain.

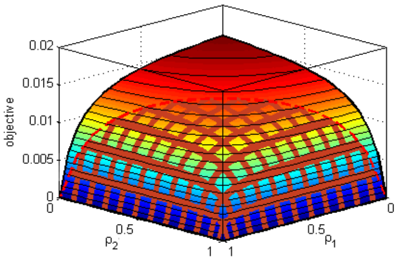


Figure 4.5: Solution space for the stiffness optimization case depicted in figure 4.3 when RAMP penalization is applied. The red shaded area below the red dotted line is the infeasible domain, where the volume constraint is exceeded. Notice that with penalization there are two feasible minima, at $\rho_1 = 0$ and $\rho_2 = 1$, and at $\rho_1 = 1$ and $\rho_2 = 0$.

4.2.3. Transient-thermal-mechanical optimization

In this section, the effect of material penalization on a 2 element TTM case is shown. But first, some function are defined. Let us define the objective response function (ORF) $f(\rho_i)$ of an element i as the objective as a function of the density of that element. For example the slice $\rho_1 = [0,1], \rho_2 = 0.5$ in figure 4.4 would give the ORF of element 1, when element 2 has a density of 0.5. For stiffness optimization, the ORF only changes due to a change in stiffness. For TTM optimization, this function will change due to a change in all the material parameters:

$$f(\rho_i) = f_{TTM}(c(\rho_i), k(\rho_i), E(\rho_i), \beta(\rho_i)) \tag{4.9}$$

One can also make an ORF for a single material parameter, around a density ρ_c, by only modifying a single material parameter and keeping the rest constant at ρ_c. This is named a single parameter ORF (SPORF). For example if this is done for the heat capacity we get:

$$f_C(\rho_i) = f_{TTM}(c(\rho_i), k(\rho_c), E(\rho_c), \beta(\rho_c)) \tag{4.10}$$

On can approximate the original ORF by adding all the SPORF's and subtracting the value around which the ORF's are made three times:

$$f(\rho_i) \approx -3f(\rho_c) + f_C(\rho_i) + f_k(\rho_i) + f_E(\rho_i) + f_\beta(\rho_i) \tag{4.11}$$

This represents an infinite Taylor expansion of the ORF, but with the interdependencies between the material parameters ignored. Although this is only locally a valid approximation, looking at the full density domain ($\rho_i \in [0,1]$) gives insight in what the effect of different penalization factors is on the shape of the ORF.

In figure 4.6 the four SPORF's as described in equation 4.10 have been plotted, with no penalization (linear interpolation) applied. Furthermore, the approximated ORF as described in equation 4.11 and the real ORF as described in equation 4.9 have been plotted. It can be seen that for this particular case the interdependencies are small over the domain as the approximated ORF resembles the real ORF quite accurately. As can be seen in figure 4.6, the SPORF's are generally not very complex functions, which are often either convex, concave, or a function with a local minimum. The penalization parameters act on these SPORF's to try to make them more concave. If this succeeds, the approximated ORF will also be concave and the minimum will lie at the boundaries. These SPORF's constantly change as densities and penalization parameters change over iterations.

Figure 4.7 shows the same situations as figure 4.6 except with penalization applied on all the material parameters such that all the SPORF's become concave. All of the SPORF's have become concave and so has the real ORF. Simply penalizing all the material parameters by the same positive value would for this example have created a minimum at an intermediate density because then some SPORF's would have been made concave but others more convex.

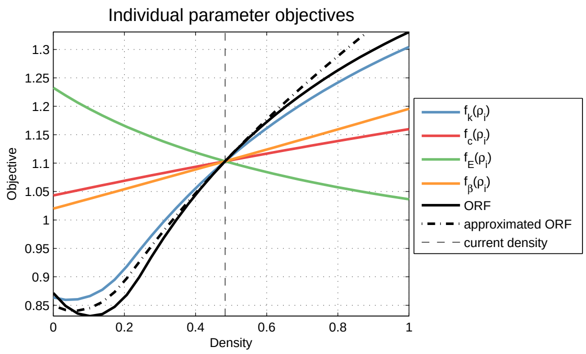


Figure 4.6: The objective response curves plotted for the individual parameters, when the other material parameters are kept constant at their value at the current density. Summing these curves approximates the real objective, which is achieved by changing all the material parameters simultaneously, i.e. representing a real density change.

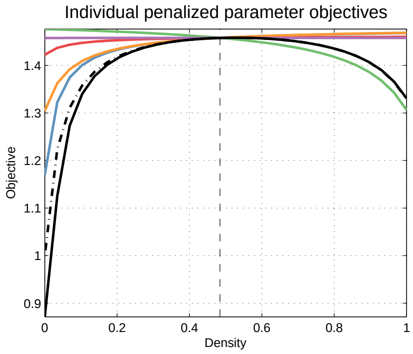


Figure 4.7: When penalization is applied to the situation in figure 4.6, the response curves become more concave. This is represented in the overall objective response function, which now has two minima almost at the boundary.

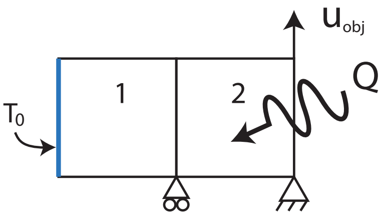


Figure 4.8: Transient-thermal-mechanical test case. The objective is to minimize the thermal error in the first 600 seconds. The thermal error is defined as the displacement u_{obj} minus a certain target displacement, squared.

TTM optimization test case

In order to see the effect of material penalization on the solution space, again a two element case is optimized. The test case is drawn in figure 4.8. A heat load is applied on element 2, while the left nodes of element 1 are kept at ambient temperature. The other edges are thermally isolated. The objective is to minimize the thermal error in the first 600 seconds. The thermal error is defined as the square of the difference between the displacement u_{obj} and a certain target displacement. In the black and white solution element 2 should always be filled, otherwise there will be a very large expansion due to the low stiffness (although the thermal stress coefficient β will also be low, the displacement is linearly related to β but reciprocal to E). Element 1 can be filled or empty depending on how much expansion is required.

The solution space without penalization is plotted in figure 4.9. Without penalization there is a clear minimum around $\rho_1 = 1$ and $\rho_2 = 0.23$. When material penalization is applied, the final solution space is largely concave as shown in figure 4.10, and all the minima are at black and white designs. The minimum at $\rho_1 = 0$ and $\rho_2 = 1$ is the global minimum, however there is a small local minimum at $\rho_1 = 1$ and $\rho_2 = 1$. Depending on the initial conditions and the path the optimizer follows, it can get stuck in this local minimum. In order to make the design space truly concave, very large penalization is needed because of the asymptote at $\rho_2 = 0$. However this is not required for material penalization to work, as small penalization already gets rid of the interior minima.

This case also shows that conventional penalization, where all the material properties are penalized by the same amount, does not work. The solution space would then look as depicted in figure 4.11. The minimum will still give a design with intermediate densities.

4.3. Comparison to grey penalization

On the two 2 element cases presented, material penalization does a good job in making the design space concave. There are other methods that also make the design space concave, one which is widely

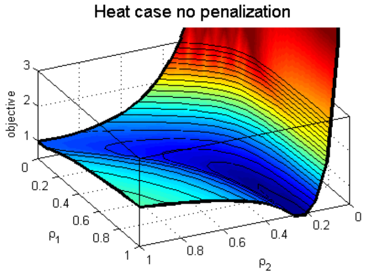


Figure 4.9: Solution space for the heat optimization case. No penalization has been applied. The minimum does not give a black and white design.

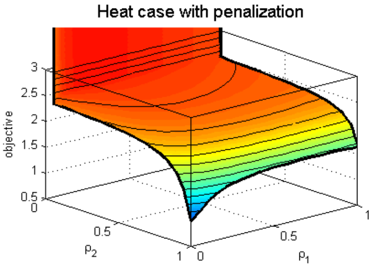


Figure 4.10: Solution space for the heat optimization case when material penalization is applied. Notice that with penalization there are only two minima, at $\rho_1 = 0$ and $\rho_2 = 1$, and at $\rho_1 = 1$ and $\rho_2 = 1$.

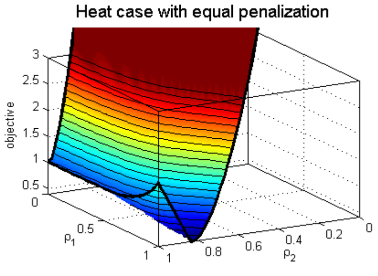


Figure 4.11: Solution space for the heat optimization case when equal penalization is applied, as is normally done in literature. It is clear that this does not work as it still contains a minimum at intermediate densities.

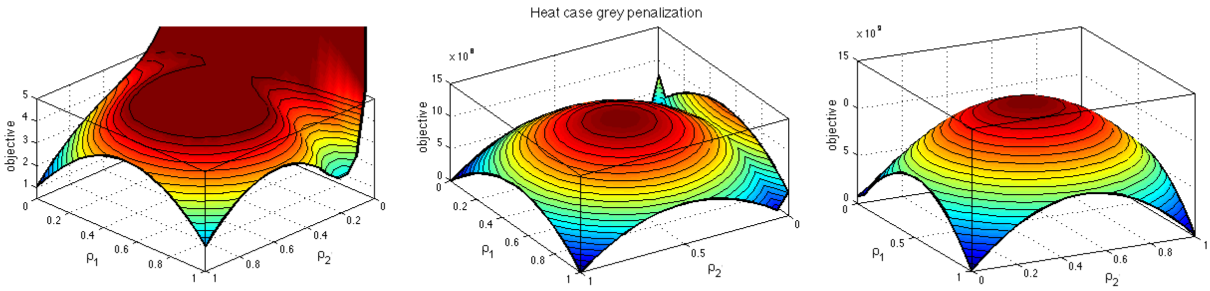


Figure 4.12: The solution space for the heat case when grey penalization is applied with different scaling factors (no grey penalization gives figure 4.9). From left to right the scaling factors are; 20, $5 \cdot 10^9$ and $5 \cdot 10^{10}$. It is clear that grey penalization only adds a parabola, which will eventually create a concave domain by pure brute force, but the solution space has nothing to do with the original optimization problem.

used is grey penalization. Grey penalization adds the following term to the objective function:

$$f_{gp} = \frac{\alpha}{NE} \sum_{i=1}^{NE} \rho_i (1 - \rho_i) \tag{4.12}$$

here α is the grey penalization scaling factor, and NE is the total number of elements. Although this will create a concave design space when α is made large enough, it does a very poor job at guiding the optimizer to the right edge of the solution space. The design space for the heat case when grey penalization is added with different scaling factors, is plotted in figure 4.12. When grey penalization is added in the order of the unpenalized objective, the interior minimum is still present. When ramping up the grey penalization scaling factor, another interior minimum is created on the other side of the solution space. Only for ridiculously high scaling factor the interior minima disappear, however the complete shape of the solution space is gone and only the grey penalization parabola is left (with four slightly differing minima). Clearly material penalization does a much better job in removing the interior minimum without distorting the original .

4.4. Limitations

Material penalization will, compared to grey penalization, not be able to make a concave design space for every optimization problem, for example, if the individual objective response curves are more complex than the ones plotted in figure 4.6, as plotted in figure 4.13. The type of function plotted in figure 4.13 frequently appears in more complex optimization problems if for instance a specific stiffness value is desired for an element. Although the interior minimum can be made very small, reducing the chance of ending up in one, it can not be eliminated. If other factors like heat conduction are also important for that element, the other material penalization response functions can usually prevent that the overall objective response function also has a interior minimum. However, if only one material constant is important, there is a good chance of ending up in the interior minimum. This can happen for example if at a certain point a hinge is created and heat conduction does not play an important role. In that case a certain stiffness value is required and becomes dominant for that element.

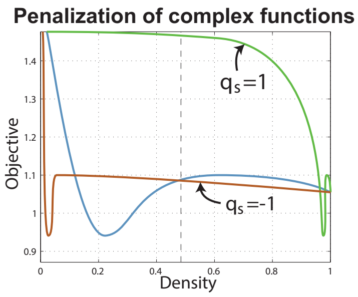


Figure 4.13: A complex function (blue) can not be made concave by applying RAMP penalization, as this can only translate the function values over the x-axis.

When the final design ends up being a black and white design, there still is a chance that the final design is a local optimum. As seen in figure 4.10, there are two optima in the final solution space. When in the transition from convex to concave functions, all the SPORF's are penalized such that they are all linear, the resulting ORF will also be a linear function, decreasing towards the lowest minimum. Whether this happens, depends on the shape of the SPORF's and the penalizations applied. It seems that if the SPORF's are all convex there is often a point in the transition that the ORF is monotonically decreasing towards the lowest minimum.

The solution space being temporarily monotonically decreasing towards the global minimum does not imply that the final design ends up in this minimum. That depends on how fast the penalization parameters are moved compared to the density variables. If the densities are optimized much faster than the penalization parameters, there is a good chance of ending in the global minimum. However this will cause a lot of iterations since the penalization parameters move very slowly. The user has to tweak both optimization speeds to get a good optimum in a reasonable amount of time.

One other important drawback of the material penalization method is, that the number of iterations increases. What often happens is that the density optimization shows very little improvement, upto the point where one normally quits the optimization loop, but the penalization parameters are then still changing and make all the intermediate material disadvantageous. Then the density optimization reacts to that and starts to change again. This can take a few cycles before a steady state is reached and therefore increases the number of iterations.

4.5. Results

In this section the material penalization method presented in the previous sections will be applied on a larger TTM case. The optimization problem is depicted in figure 4.14. On the left and right side the domain is clamped, and ambient temperature is prescribed, which is also the initial temperature of the complete domain. The top and bottom side are thermally and mechanically isolated. On the red elements a heat load is applied from $t = 0$. The objective is to minimize the thermal error from $t = 30$ to $t = 60$. The thermal error is defined as $(u_{obj} - 1.5 \cdot 10^{-3})^2$, which means that a displacement of 1.5mm is desired over the time frame of interest. Furthermore, because no volume constraint is applied, large structures appear in the design that only have a marginal influence on the objective. In order to suppress this and keep the design 'clean', a small value $(1 \cdot 10^{-5})$ is added to the density sensitivities. Elements that give an objective improvement smaller than this value (i.e. elements that have a sensitivity value lower than $-1 \cdot 10^{-5}$), will stay empty.

The solution without penalization is displayed in figure 4.16. The mechanism that is created pushes the area to which the heat load is applied upward, to reach the target displacement. Around the fixed domain there is a ring that provides heat capacity and slightly decreases the objective displacement when it expands. Although this design gives a clue for the engineer in what direction the final design can be made, it leaves a lot undecided. No design for the suspension of the domain to which the heat

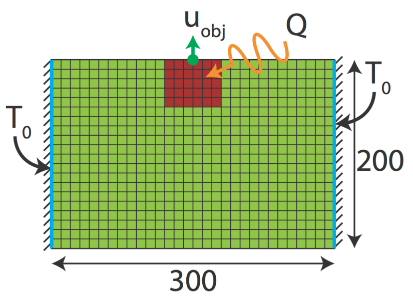


Figure 4.14: Transient-thermal-mechanical case. The red elements on which the heat load is applied are fixed, the optimization domain consists of the green elements.

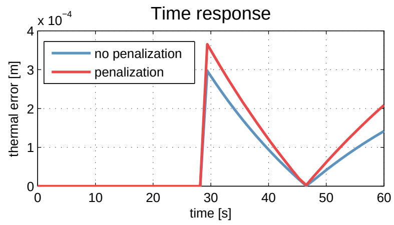


Figure 4.15: Time response of the non penalized and material penalized designs. Both cross the desired displacement in the middle of the time frame which gives the lowest objective when the designs are roughly linearly expanding.

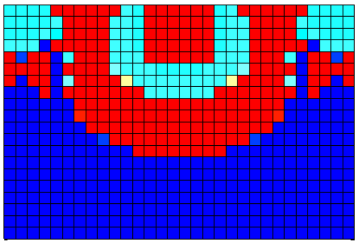


Figure 4.16: Optimal design when no penalization is applied. Red elements are filled, blue elements are void. Intermediate colours represent intermediate densities.

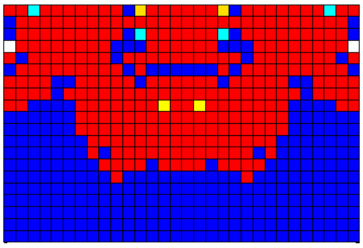


Figure 4.17: Optimal design with material penalization applied. Its transient behaviour is displayed on the cover of this report.

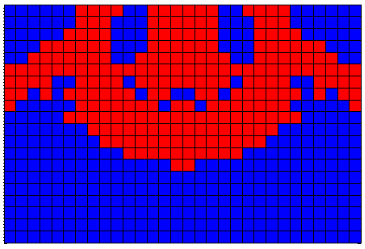


Figure 4.18: Optimal design when grey penalization is applied after the design has reached is optimum without penalization.

load is applied can be deduced because it is suspended in intermediate densities.

When material penalization is applied, it becomes much clearer what the final design should look like. This is displayed in figure 4.16. Now there are hardly any intermediate densities left. The intermediate densities that remain are on places where for example a certain stiffness is required: around hinge points. These are points to which the engineer should pay special attention to e.g. calculate the exact required thickness. The penalization parameters for the final design and the evolution of the penalization parameters over the iteration process are discussed in appendix F.

When grey penalization is applied to the unpenalized case, a complete black and white design can be achieved which, to the eye, looks to have a similar behaviour as the other two (figure 4.18). However the objective of the grey penalized design is 40 times higher than the non penalized objective, whereas the material penalized design has only a 1.3 times higher objective. The time responses of the unpenalized and the material penalized designs are plotted in figure 4.15. It can be seen that both designs are able to cross the desired displacement in about the middle of the time frame of interest. The grey penalized design was already beyond the displacement of interest at the start of the time frame

4.6. Conclusions and recommendations

Material penalization is a promising penalization method. The method is able to do for multiphysics optimization what normal penalization does in stiffness optimizations: creating a concave design space. Certainly material penalization will often not create a complete black and white design. Especially in areas where only one material parameter is dominant. However, these intermediate densities are usually easy to interpret and point the engineer towards possible critical spots in the design. That is a much better option than grey penalization, which will always provide a black and white design, but in that design the performance might be completely lost.

Recommendations

Material penalization as presented in this chapter is a first step in creating a concave design for multiphysics optimization. So far this only works for convex SPORF's, for more difficult functions the optimizer can still get stuck in local minima. Different techniques might be applied to overcome this. Making a combination with (local) grey penalization can be one.

Furthermore, it would be interesting to do more research in when the transition between a convex and a concave design is monotonically decreasing towards the global minimum. If this point can be recognized the speed of the penalization parameters can be halted at such moments to allow the densities to move towards the global optimum. This might also help to decrease the number of required iterations.

5
ASML case study

NOTE: DUE TO CONFIDENTIALITY PARTS OF THIS CHAPTER HAVE NOT BEEN DISPLAYED WHICH MIGHT AFFECT READABILITY.

This chapter will present the results of a transient-thermal-mechanical topology optimization done on a case provided by ASML. First, a brief overview of the ASML case is given. In section 5.1, the models used for the different topology optimizations that have been done are described. In section 5.2, the results of the different topology optimizations are presented and discussed. In the final section 5.3, the results of the optimizations of section 5.2, which used an axisymmetric model, are compared with a topology optimization on a semi-3D model.

Problem background

Currently ASML is working on the new extreme ultraviolet lithography machines. However, the TWINSCAN machines are also still in development. The TWINSCAN uses immersion lithography. In immersion lithography an image is projected on the wafer by illuminating an image with 193nm laser light and focussing it with multiple lenses on a wafer. In order to improve the minimum feature size, the k1-factor is reduced by placing water between the lens and the wafer [27]. Because not every wafer has the same diameter, there is a small gap between the wafer edge and the cover ring of the wafer table. When the edge dies are illuminated water crosses the wafer edge introducing a load, deforming the wafer. As objective for TTM TO we want to minimize the deformation in the first 3 seconds. This will be further referred to as the edge-load case. A sketch is given in figure 5.1.
Five different topology optimizations have been done on the edge-load case:

1. Steady state thermal-mechanical TO, using an axisymmetric model.
2. Transient thermal-mechanical TO, using an axisymmetric model.
3. Transient thermal-mechanical TO, using an axisymmetric model and a extended optimization domain.
4. Multi-material transient thermal-mechanical TO, using an axisymmetric model and a extended optimization domain.
5. Multi-material transient thermal-mechanical TO, using an non-axisymmetric model and a extended optimization domain.

The steady state optimization has mainly been done in order to familiarize with the topology optimization on the edge-load case without having to deal with complex sensitivities. The transient optimization represents the original edge-load case, while the succeeding optimizations were done in order to get an increasingly better performance. The final optimization with non-axisymmetrical loading was done to check the axisymmetric designs. The following sections 5.1 and 5.2 focus on the first four axisymmetric optimizations. The non axisymmetrical optimization is discussed in section 5.3.

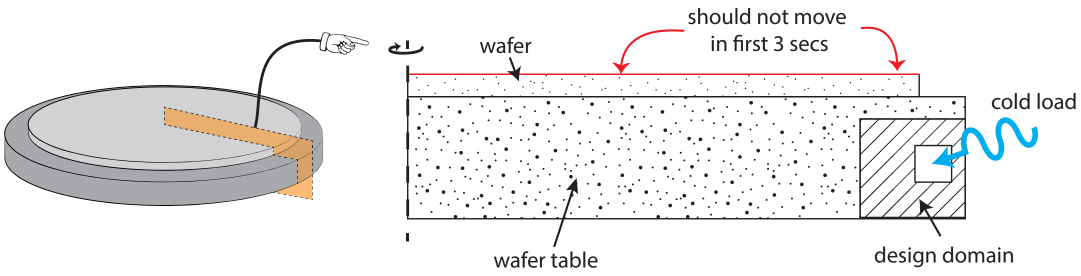


Figure 5.1: Sketch of the edge-load case.

5.1. Axisymmetric model

The model is an important part of the topology optimization loop. The model is evaluated every iteration to calculate the response of the updated design. With this response the objective function can be evaluated and it can be used to calculate the design variable sensitivities. The correctness of the model influences the practicability of the final design, and the size of the model can have a large impact on the speed of the optimization.

5.1.1. Model assumptions and parameters

Assumptions

For the topology optimization model, some simplification are made with respect to reality. Most of the simplifications result in reduction of the model, and therefore in less computational time. For a final design the model might be extended by not making any of the following assumptions. An overview of the main assumptions is given.

1. The problem is considered to be axisymmetric.
   The wafer and the wafer table are geometrically axisymmetric (besides from some small details), but the load is local. By making the model axisymmetric the load is changed considerably. The expectation was that the performance of the designs only slightly reduces with a local load. In the final section this is tested by altering the model to incorporate the non symmetric load.

2. Only the wafer and the wafer table are modelled.
   The chuck to which the wafer table is connected, is considered to be much stiffer and therefore not affected by the deformation of the wafer table. Also the convection to the chuck is small so its thermal deformation is neglected. The mechanical connection with the chuck is modelled as a nominal and a translational stiffness. The thermal connection is modelled by heat convection.

3. The wafer and the wafer table are thermally and mechanically isolated except for the interaction with the chuck and the cold load.
   Heat transfer by convection trough the air and radiation are neglected. Also interaction with the immersion water and the radiation from the source are not modelled.

4. The structural dynamic behaviour is neglected.
   The dynamic behaviour is considered to be much faster than the transient thermal behaviour. Therefore the assumption is made that static equilibrium is reached at every time step.

5. The material is linear elastic, homogeneous, isotropic and temperature independent.
   These assumptions can be made because the deformations are very small and the temperature fluctuations in the order of tenths of Kelvins.

Parameters

With the assumptions given above a 2D axisymmetric model is made. The model represents the wafer and wafer table. For the optimizations with the extended domain, the width and height of the optimization domain is increased. The wafer table height is also increased. On the area where the cold load is applied, material has to be present, and this area does not belong to the optimization domain. This area is connected with a ring over the complete circumference to the top surface of the wafer table. This connection can also not be altered.

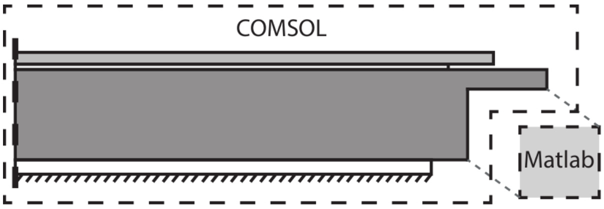


Figure 5.2: Division of the model into two substructures.

5.1.2. Discretization

The model is divided into two substructures: the optimization domain, and the rest of the model (see figure 5.2). The FE model of the optimization domain will change every iteration, since all the matrices ($\mathbf{C_T}$, $\mathbf{K_T}$, $\mathbf{K_u}$ and $\mathbf{A}$) are dependent on the design variables. The rest of the model does not change so it only needs to be constructed once. The two substructures are combined at every iteration after the matrices for the optimization domain have been constructed. The matrices for the optimization domain are calculated within the optimization code in Matlab. However, the rest of the model is made with the COMSOL Multiphysics software. This is done for a number of reasons. First of all, the structure is very thin and long. At the tip a fine mesh is required for the topology optimization domain, but at the other end a course mesh is sufficient because not much will happen there. This means that an advanced meshing tool is required, which is built in in COMSOL. Secondly, using COMSOL greatly reduces the risk of error in the FE matrices since no custom programming is done. Finally, COMSOL has a smooth connection with Matlab which makes extracting the matrices quite easy. A disadvantage of this method is that after the matrices have been imported into Matlab, the degrees of freedom (DOFs) at the interface between the two substructures have to be located and connected to the right DOFs of the optimization domain. Since COMSOL can also export selections of boundaries, this was solved quite easily.

The two substructures and their assembly process are described in more detail in appendix **??**. Static condensation is applied to the mechanical part of the COMSOL substructure to reduce the matrix size. This is described in more detail in appendix G.

5.1.3. Objective

The objective of all the optimization problems is to minimize the absolute displacement of the top wafer surface in the first three seconds after the cold load is applied. For the static problem the displacement at steady state is to be minimized. Furthermore, the radial displacement is roughly ten times more important than the axial displacement. The transient optimization problem is defined as:

$$\min_{\rho} \left(f(\mathbf{u}(t)) = \int_0^3 (\mathbf{s}_R^{\mathbf{T}} + 0.01\mathbf{s}_A^{\mathbf{T}})(\mathbf{u}(t) \circ \mathbf{u}(t))\mathrm{d}t \right) \tag{5.1}$$

$$\text{subject to: } \mathbf{C}(\rho)\dot{\mathbf{T}} + \mathbf{K_T}(\rho)\mathbf{T} = \mathbf{Q} \tag{5.2}$$

$$\mathbf{K_U}(\rho)\mathbf{u} = \mathbf{A}(\rho)\mathbf{T} \tag{5.3}$$

$$0 \leq \rho \leq 1 \tag{5.4}$$

where s_R and s_A are boolean vectors that select the radial and axial displacement DOFs of the top wafer surface respectively, and $\circ$ is the Hadamard (element wise) product. Because the squared displacements are used, the axial displacements are multiplied by a factor 0.01.

5.2. Axisymmetric results

The results of the topology optimizations of the axisymmetric models are given in table 5.1. The performance of the designs is given in table 5.2. The performance of the static optimization is obviously bad, it hardly performs better than the completely filled (conventional) design. That is because no transient effects are used in the optimization. The design twists the end of the wafer by pulling the gutter down left as it cools down, slightly increasing the performance.

The transient optimization has a different operating principle, which is also utilized in the other two extended domain optimizations. An isolated ring of material is suspended on the edge-load area.

Table 5.1: Design for the optimization domain of the axisymmetric cases. Green elements are filled with the original material, while red elements are filled with a higher CTE material. The white elements are empty. All the transient designs are thus only connected to the wafer table via the mandatory connection area at the top of the edge-load area.

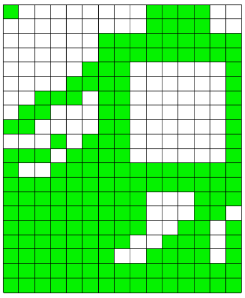
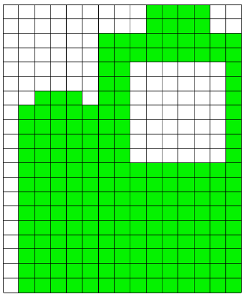
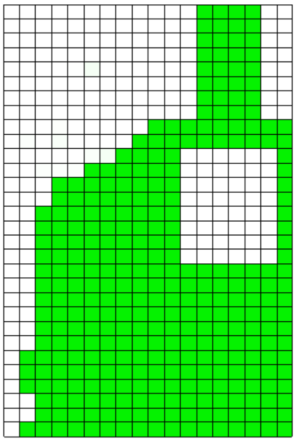
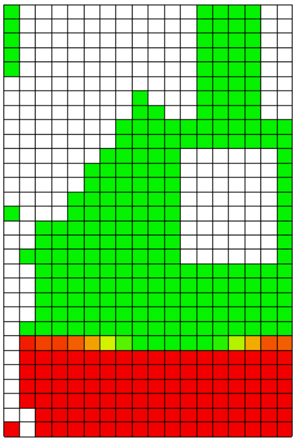
Normal domain		Extended domain	
Static optimization	Transient optimization		
Single-material			Multi-material
			

Table 5.2: Relative performance of the four optimization on the axisymmetric model compared to the filled (original) design.

Domain	Normal			Extended		
Optimization	-	Static	Transient	-	Transient	
	Filled	Single-material		Filled	Single-material	Multi-material
Objective decrease compared to filled	0%	5.38%	32.70%	0%	77.55%	96.46%
Radial displacement decrease at $t = 3$s	0%	-0.2%	18.8%	0%	59.5%	99.4%
Axial displacement decrease at $t = 3$s	0%	30.5%	-38.8%	0%	-182.3%	-1059.8%

Because it is not connected to the wafer table directly, this part gets the coldest and therefore shrinks the most. Because the model is axisymmetric, this shrinking ring pulls the edge-load area inward. This causes bending of the beam to which the edge-load area is connected. Due to the bending this beam is stretched on the top surface, where the wafer is connected. This mechanism will counteract the shrinking of the wafer. The principle is displayed in figure 5.3. Here the deformed model is displayed, without the optimization domain. As the top beam is bend further, the wafer is stretched in the opposite direction of the shrink. Of course this comes at the cost of large axial displacements. However, axial displacement is ten times less costly.

In the normal domain, the optimizer is not able to create enough force to effectively bend the edge-load area inward. Radial displacement of the tip displacement is only improved by 20% after three seconds. In the Extended domain a larger moment can be created on the beam, and the performance is improved with 64%. By adding a higher CTE material as an optional material, it can, with its larger coefficient of thermal expansion, exert an even higher inward force, and effectively improve the radial displacement objective by almost 100%.

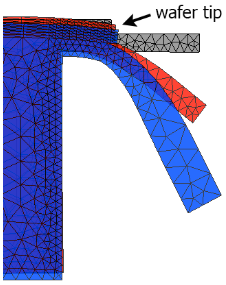


Figure 5.3: Deformation of the wafer table, plotted without the optimization domain. As the edge-load area is pushed inward harder, the beam to which it is suspended stretches more at the top, decreasing the radial displacement, but increasing the axial displacement of the wafer.

5.3. Non-axisymmetric optimization

The load is not applied to the full circumference but assumed to work only on a part of 45°. Therefore a 3D model was constructed to test the axisymmetric designs, and if necessary, perform a topology optimization on the 3D model.

5.3.1. Semi-3D model

The difficulty with a full 3D model is that the degrees of freedom and therefore the computational time increases dramatically. Instead of making a full 3D model, the load can also be expressed as a Fourier series in the tangential direction:

$$\mathbf{Q} \approx \sum_{i=1}^{n} \bar{\mathbf{Q}}_i \cos i\theta + \sum_{i=1}^{n} \bar{\bar{\mathbf{Q}}}_i \sin i\theta \tag{5.5}$$

here $\bar{\mathbf{Q}}_i$ and $\bar{\bar{\mathbf{Q}}}_i$ are the modal amplitudes. Because the load is not very local, only a few harmonics are needed to represent the load. Here, 11 harmonics have been chosen. The load was defined as a cosine function with a period of 45°, symmetrical around $\theta = 0$. This cancels all the sine terms, further reducing the problem. The load and its approximation are both displayed in figure 5.4. The temperatures and displacements can then also be described as a series of cosines or sinuses:

$$\mathbf{T} \approx \sum_{i=1}^{n} \bar{\mathbf{T}}_i \cos i\theta \tag{5.6}$$

$$\mathbf{u} \approx \sum_{i=1}^{n} \bar{\mathbf{u}}_i \cos i\theta \qquad \mathbf{v} \approx \sum_{i=1}^{n} \bar{\mathbf{v}}_i \sin i\theta \qquad \mathbf{w} \approx \sum_{i=1}^{n} \bar{\mathbf{w}}_i \cos i\theta \tag{5.7}$$

where again $\bar{\mathbf{T}}_i$, $\bar{\mathbf{u}}_i$, $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{w}}_i$ are the modal amplitudes. The cosine and sine terms can be included in the shape functions that are constructed for the discretization of the equilibrium equations. That way they will eventually enter the FE matrices. These will be independent of θ because an integration in the direction of θ between 0 and 2π is done. However they will be dependent on i, the harmonic number. This gives for each harmonic that is used, a separate system of equations that needs to be solved to obtain the modal amplitudes for that harmonic:

$$\mathbf{K}_{\mathbf{U}_i} \bar{\mathbf{u}}_i = \mathbf{A}_i \bar{\mathbf{T}}_i \tag{5.8}$$

$$\mathbf{C}_{\mathbf{T}_i} \dot{\bar{\mathbf{T}}}_i + \mathbf{K}_{\mathbf{T}_i} \bar{\mathbf{T}}_i = \bar{\mathbf{Q}}_i \tag{5.9}$$

By using 11 harmonics, the problem size increases only 11 times. For a detailed description of the method and the construction of the FE matrices, the reader is referred to Cook *et al.* [28]. Because COMSOL 4.3 does (by the authors knowledge) not support axisymmetric elements with nonaxisymmetric loading, the complete model is constructed in Matlab. Gauss quadrature has been used to numerically integrate the element matrices. This avoids the need to sample at the singularity at $r = 0$.

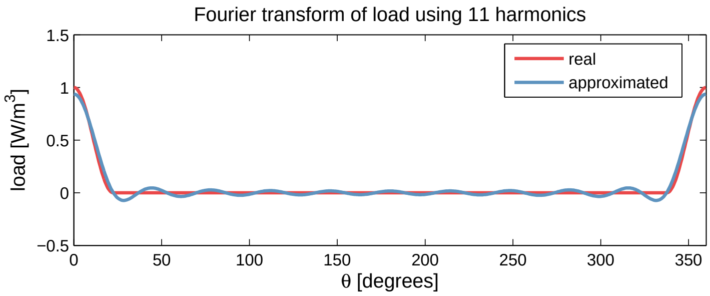


Figure 5.4: Fourier transform of the load using the first 11 harmonics.

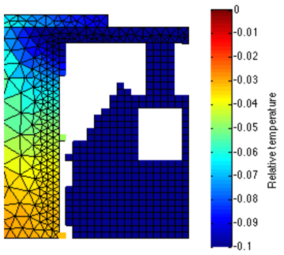


Figure 5.5: Temperatures after 3 seconds of the multi-material extended domain design. Note that the temperature scale is truncated at its lower limit.

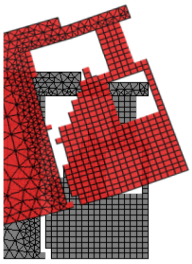


Figure 5.6: Sketch of the bending due to the non uniform vertical heat distribution in the wafer table (red) as compared to uniform (grey). The bending decreases the performance as the wafer tip is displaced inward.

5.3.2. Performance of axisymmetric optimizations

The three transient axisymmetric results of section 5.2 have been tested on the 3D model to see if the principle still works. The performance appeared to be even worse than for the original completely filled design. This a result of the combination of the transient designs and the reduced heat load:

Because of the transient designs, where the edge-load area is only connected to the wafer table via the necessary connection ring, the cold can only enter the wafer table through the thin flange at the top edge of the wafer table (see figure 5.5). This causes and uneven cold distribution which bends the wafer table upward (see figure 5.6). This increases the radial displacement of the wafer tip.

In the axisymmetric case this bending is also present, however if the edge-load are creates a large enough inward force, the top surface is bent downward, decreasing the radial displacement of the wafer. However in the 3D-model, this inward force is much smaller since the cold load works on only 1/8th of the edge-load area. Therefore the upward bending due to the difference in temperatures becomes dominant and decreases the performance even compared to the completely filled design.

5.3.3. 3D optimization

Because the axisymmetric designs do not work for a non-axisymmetric load, a topology optimization on the 3D model is performed. The optimal design for the 3D model is displayed in figure 5.7. From this design it is hard to understand its working principle. Therefore a simplified design is displayed in figure 5.8. This design has a similar working principle, but a slightly (3%) less performance.

As seen with the axisymmetric designs, the upward bending of the wafer table due to the cold load that enters the wafer table from the top (figure 5.5 and 5.6), decreases the performance. The new design prevents this bending by dissipating the cold on the top as well as on the bottom of the wafer table. This provides a uniform vertical temperature distribution in the wafer table and prevents bending (see figure 5.9). Because of the connection with the bottom of the wafer table, the top beam to which the wafer is connected can not be bent downward completely as was done in the axisymmetric case (figure 5.3). Therefore only the part to which the wafer is connected is bend downwards. This is done by connecting a higher CTE material to the beam which bends it downward due to its higher CTE. The optimal design has almost the same working principle, except that it has a cold sink in the lower right corner, slightly improving the objective. It also introduces some holes in order to reduce the axial

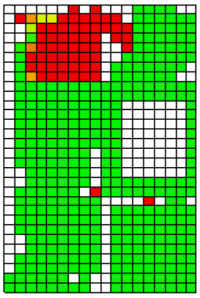


Figure 5.7: Optimal design for the 3D model.

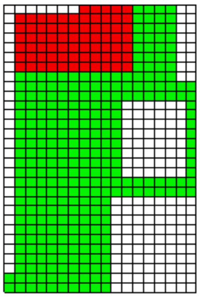


Figure 5.8: Simplified version of the optimal designs that shows the working principle.

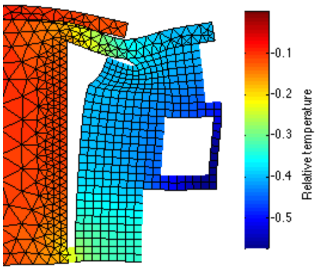


Figure 5.9: Temperature distribution and displacement of the simplified version of the optimal design (figure 5.8) after 3 seconds. The displacement is scaled with a factor $3{\cdot}10^4$.

displacement at the cost of some radial displacement. The objective function of the optimal 3D design is 30% lower compared to a fully filled design. The radial wafer tip displacement after 3 seconds is improved with 50%-compared to the filled design. The axial displacement is degraded by 750%.

5.3.4. Design variations

Now that the design principle of the optimal design is known, variations can be made. For example if less radial deflection is required, one can increase the amount of higher CTE material. This of course will increase the axial displacement, since the wafer will be pulled downwards stronger. By making the bottom support more flexible, the wafer will be pushed up a little due to bending. Different designs have been plotted in figure 5.10. The different designs are all variations on the optimal design found with the topology optimization. For the working principle of bending the wafer in order to counteract the shrink, there is a clear trade-off between radial and axial displacement. The weighting of the radial and axial displacement will determine the optimal design.

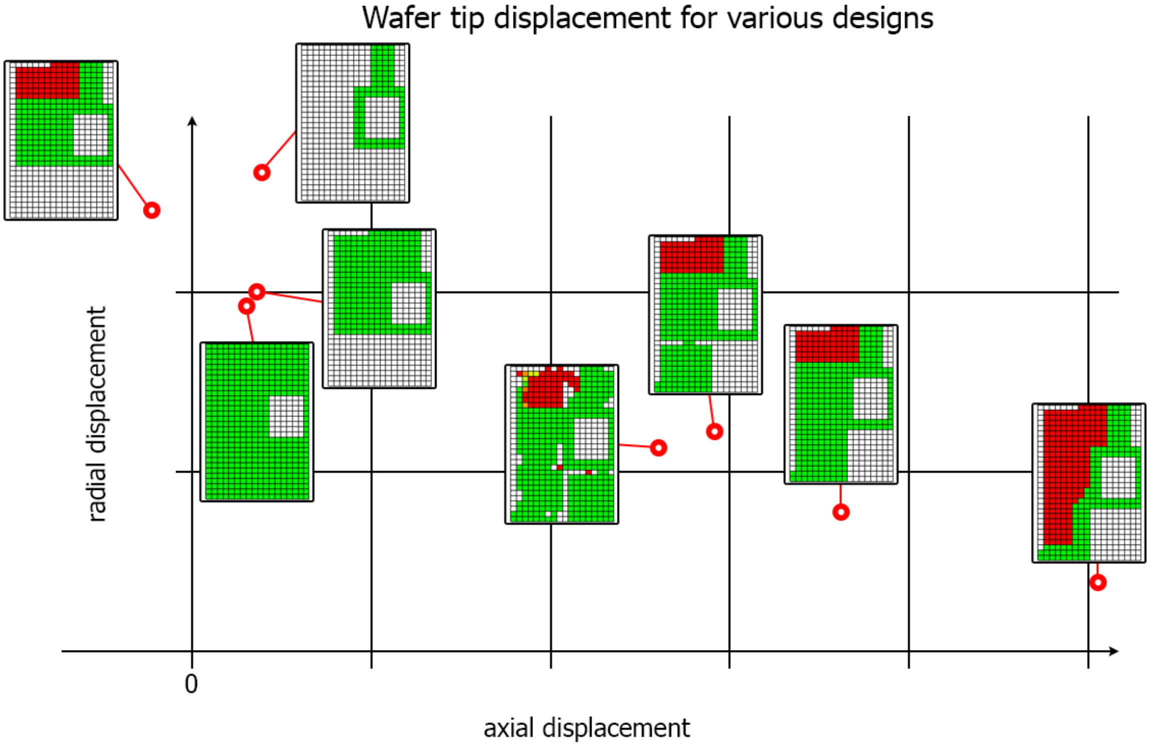


Figure 5.10: Wafer tip displacements of different designs at $t = 3$s. For the working principle of bending the top beam to counteract the shrink, a clear trade-off between axial and radial performance can be seen.

5.4. Conclusion

Topology optimization is shown to be a capable tool for the design of material lay-outs for thermal-mechanical problems. Both for the axisymmetric and the non-axisymmetric case, innovative designs have been created. The original assumption that the load could be represented as being axisymmetric turned out to be false. The real load, which is much smaller, was not able to recreate the axisymmetric results. A topology optimization incorporating the non-axisymmetric load gave a new optimal design. Final radial displacement was improved by 79.2%. This was at the cost of axial displacement, which performance was decreased with 661.8%. Since radial displacement was a factor 10 more important, the new design has an overall performance increase of 30%.

6
Conclusions and Recommendations

In this chapter the final conclusions and recommendation concerning TTM TO's are stated.

6.1. Conclusions

The goal of this project was performing a TTM topology optimization on the ASML case. This has been done with success on an axisymmetric, as well as on a semi-3D model. The semi-3D model showed that the initial assumption of an axisymmetric load was not a realistic simplification, and that this assumption was exploited by the optimizer. The TO on the semi-3D model produced an innovative solution with which, by allowing two materials, improvements in the objective of up to 30% can be reached. Furthermore once the working principle of the new design was understood, modification could be made to alter its behaviour.

In the process of performing the TTM TO, a new interpretation for the difference between direct and adjoint sensitivities has been found. It is shown that the difference between the direct and the adjoint method is that the direct method calculates for each time instance what the influence of the preceding loadings is on that time instance, while the adjoint method calculates for each time instance how much influence the load on that time instance has on all the subsequent time instances. Also an explicit formula for the adjoint, and also for all the sensitivities, can be derived. These, however, include an expensive matrix exponential, which limits its use to small systems which have to be evaluated with a relatively large number of time steps.

Furthermore, a new penalization method, material penalization, is proposed. This method aims at creating a concave solution space for multiphysics problem, and thereby creating a black and white solution. Material penalization is not always able to create a black and white design. This is due to the fact that RAMP penalization can only make monotonically increasing or decreasing functions concave. If the objective response function for individual material parameters is not of this form, local minima will persist. Nevertheless, material penalization outperforms conventional constant penalization and grey penalization in creating a black and white design, that still has a good performance. This makes it a promising new penalization method.

6.2. Recommendations

For future research in material penalization and for performing TTM TO's in general, a few recommendations emerge from this research:

In order to make TTM TO a more plug-and-play method which can be easily used by engineers, one of its capabilities should be that it is able to create a black and white design without the need tweaking several parameters endlessly. Although this is not fully accomplished by material penalization, it is a step in the right direction. During this research, briefly experiments have been done with local grey penalization, with promising results. In this method, each element is given its individual grey penalization scaling factor, which is increased if the element is grey, and decreased if its black or white. This method can be combined with material penalization to make it able to get out of interior local minima.

Furthermore, the effect of the magnitude of the move limits for penalization parameters versus density variables should be further investigated. If material penalization is capable of creating a concave solution space, there is often a point in the transition between convex and concave solution space in which it is monotonically decreasing towards the global minimum. Tuning the move limits to freeze the solution space at this point might give much better convergence towards the global minimum.

For future TO problems like the ASML case, every assumption should be reviewed for validity. Although in conventional designs a smaller load might give only less performance, TO is known for exploiting every flaw in the model, and base the design completely around some extra freedom it might have compared to reality.

In the ASML case, manufacturability of the final design was not a constraint, except for the need of a black and white design. Although some of the designs can be made with conventional techniques, this will not always be the case for TO. In future optimizations constraints should be added in order to produce designs which can be manufactured (within a reasonable budget) in order to validate the TO.

The link between Matlab and COMSOL was very easy to set up. For larger 3D optimizations it might be interesting to perform the complete FEM part in COMSOL (i.e. feeding the design back to COMSOL every iteration). This way parallelization can be easily achieved since it is build into COMSOL. (This probably also holds for other FE packages like ANSYS)

Finally, it will be nice to compare material penalization to evolutionary structural optimization (ESO). No time was left at the end of this research to implement and compare ESO to the density approach with material penalization, but it will be interesting to see how the results differ because ESO by definition creates a black and white design.

A

Building the FEM model

For the optimization domain, a finite element code was implemented in order to build the model. In this section, the creation of the FE matrices for linear rectangular thermo-mechanical elements as used in the code will be explained.

Equilibrium Equations

First, the equilibrium equations are derived. These equations are later discretized using Galerkin's method to form the FE matrices.

Thermal An infinitesimal axisymmetric element is considered to construct the equilibrium equations (see figure A.1). In this element, there can be heat conduction through the walls, heat generation and heat storage. Because the element is axisymmetric, the conduction in tangential direction will be constant. The energy balance of the element is then [29]:

$$\begin{pmatrix}\text{Rate of energy con-}\\ \text{ducted out of con-}\\ \text{trol volume}\end{pmatrix} + \begin{pmatrix}\text{Rate of energy}\\ \text{stored inside con-}\\ \text{trol volume}\end{pmatrix} = \begin{pmatrix}\text{Rate of energy con-}\\ \text{ducted into control}\\ \text{volume}\end{pmatrix} + \begin{pmatrix}\text{Rate of energy gen-}\\ \text{erated inside con-}\\ \text{trol volume}\end{pmatrix} \tag{A.1}$$

The surface areas of the element are given by:

$$A_{\text{back}} = r\mathrm{d}\theta\mathrm{d}z \tag{A.2}$$

$$A_{\text{front}} = (r + \mathrm{d}r)\mathrm{d}\theta\mathrm{d}z \tag{A.3}$$

$$A_{\text{top}} = A_{\text{bottom}} = \frac{\mathrm{d}\theta}{2}((r + \mathrm{d}r)^2 - r^2) \tag{A.4}$$

$$A_{\text{left}} = A_{\text{right}} = \mathrm{d}r\mathrm{d}z \tag{A.5}$$

With the internal energy per unit volume given by $Q_i = c_p \rho T$ equation A.1 becomes:

$$\left(\left(q_r + \frac{\partial q}{\partial r}\mathrm{d}r\right) A_{\text{front}} + \left(q_z + \frac{\partial q}{\partial z}\mathrm{d}z\right) A_{\text{top}} + q_\theta A_{\text{right}}\right) + \left(c_p \rho \frac{\partial T}{\partial t} A_{\text{top}}\mathrm{d}z\right) = \\ (q_r A_{\text{back}} + q_z A_{\text{bottom}} + q_\theta A_{\text{left}}) + (Q A_{\text{top}}\mathrm{d}z) \tag{A.6}$$

Here q is the rate of heat flow trough a surface in [W/m^2], Q the internal heat generation in [W], c_p the heat capacity at constant pressure in [J/kgK], ρ the density in [kg/m^3] and T the temperature in [K]. $\mathrm{d}r$, $\mathrm{d}z$ and $r\mathrm{d}\theta$ are the lengths of the sides of the element in [m] (see figure A.2). Substituting equations A.2-A.5 and rearranging gives:

$$c_p \rho \frac{\partial T}{\partial t}\left(\frac{1}{2}\mathrm{d}r^2 + r\mathrm{d}r\right)\mathrm{d}\theta\mathrm{d}z + q_r \mathrm{d}r\mathrm{d}\theta\mathrm{d}z + \frac{\partial q_r}{\partial r}\left(\mathrm{d}r^2 + r\mathrm{d}r\right)\mathrm{d}\theta\mathrm{d}z + \frac{\partial q_z}{\partial z}\left(\frac{1}{2}\mathrm{d}r^2 + r\mathrm{d}r\right)\mathrm{d}\theta\mathrm{d}z = \\ Q\left(\frac{1}{2}\mathrm{d}r^2 + r\mathrm{d}r\right)\mathrm{d}\theta\mathrm{d}z \tag{A.7}$$

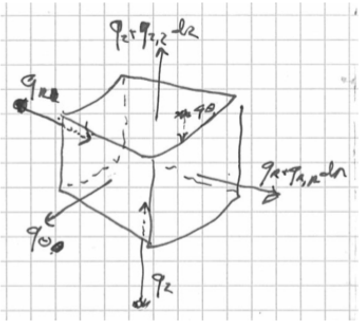
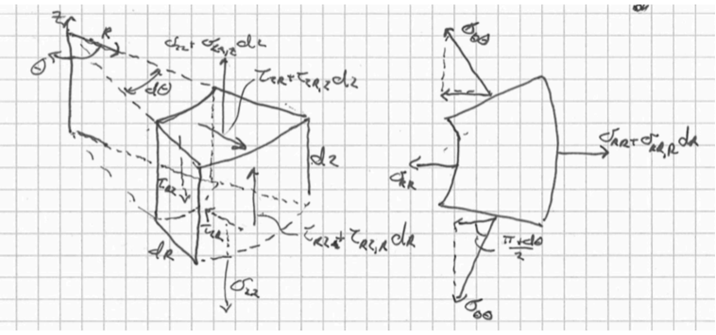


Figure A.1: A finite element in polar coordinates with heat flows.



z
R
Θ
$d\Theta$
$\sigma_{zz} + \sigma_{zz,z}\,dz$
$\tau_{zR} + \tau_{zR,z}\,dz$
dz
τ_{Rz}
τ_{zR}
$\tau_{Rz} + \tau_{Rz,R}\,dR$
dR
σ_{zz}
$\sigma_{\theta\theta}$
$\sigma_{RR} + \sigma_{RR,R}\,dR$
σ_{RR}
$\frac{\pi + d\theta}{2}$
$\sigma_{\theta\theta}$

Figure A.2: A finite element in polar coordinates with the possible stresses working on it. The left figure shows the shears stresses and the stress in z-direction, the right figure is a top view and shows the remaining stresses.

Because an infinitesimal element is considered, the terms containing $\mathrm{d}r^2$ are neglected. Applying Fourier's law of heat conduction for isotropic material [30], $q_i = -k\frac{\partial T}{\partial i}$ and dividing by the (approximated) volume $r\mathrm{d}r\mathrm{d}\theta\mathrm{d}z$ brings us to the equilibrium equation:

$$c_p\rho\frac{\partial T}{\partial t} - k\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{\partial^2 T}{\partial z^2}\right) = Q \tag{A.8}$$

Structural For the structural equilibrium equations also an infinitesimal small axisymmetric element is considered. The stresses working on such an element are shown in figure A.2. As a consequence of the axial symmetry, the shears involving the tangential direction are zero. Also the normal stress is constant in the tangential direction. By adding all the forces two equilibrium equations for static equilibrium in radial and axial direction are obtained:

$$\begin{aligned}\sigma_{rr}A_{\text{back}} + \tau_{zr}A_{\text{bottom}} + 2\sigma_{\theta\theta}A_{\text{left}}\cos\left(\frac{\pi}{2} - \frac{\mathrm{d}\theta}{2}\right) = \\ \left(\sigma_{rr} + \frac{\partial\sigma_{rr}}{\partial r}\mathrm{d}r\right)A_{\text{front}} + \left(\tau_{zr} + \frac{\partial\tau_{zr}}{\partial z}\mathrm{d}z\right)A_{\text{top}} + F_r A_{\text{top}}\mathrm{d}z\end{aligned} \tag{A.9}$$

$$\sigma_{zz}A_{\text{bottom}} + \tau_{rz}A_{\text{back}} = \left(\sigma_{zz} + \frac{\partial\sigma_{zz}}{\partial z}\mathrm{d}z\right)A_{\text{top}} + \left(\tau_{rz} + \frac{\partial\tau_{rz}}{\partial r}\mathrm{d}r\right)A_{\text{front}} + F_z A_{\text{top}}\mathrm{d}z \tag{A.10}$$

Here σ_{xy} and τ_{xy} are stresses working on a plane with its normal in the x direction and working in the y direction, given in [N/m^2]. F_r and F_z are body forces acting on the element in [N/m^3]. Since an infinitesimal small element is considered, $\cos\left(\frac{\pi}{2} - \frac{\mathrm{d}\theta}{2}\right) \approx \frac{\mathrm{d}\theta}{2}$ and again the quadratic terms $\mathrm{d}r^2$ are neglected. Substituting equations A.2-A.5 gives the equilibrium equations:

$$\frac{\partial\sigma_{rr}}{\partial r} + \frac{\partial\tau_{zr}}{\partial z} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} + F_r = 0 \tag{A.11}$$

$$\frac{\partial\sigma_{zz}}{\partial z} + \frac{\partial\tau_{rz}}{\partial r} + \frac{\tau_{rz}}{r} + F_z = 0 \tag{A.12}$$

Approximation of the dependent variables

Since there probably are no exact solutions for the equilibrium equations on the complicated wafer table and wafer structure, an approximation of the dependent variables (temperature T and radial and axial displacements u and w) is made. The approximated fields are chosen to be a linear combination of basis functions. The i-th basis function is chosen to be equal to 1 on the position of corresponding dof i, and zero on the other dofs's positions. This gives a linear approximation field between the dofs. On an element level the basis functions will be of the following form:

$$N_i(r,z) = \alpha_i + \beta_i r + \gamma_i z + \delta_i rz \tag{A.13}$$

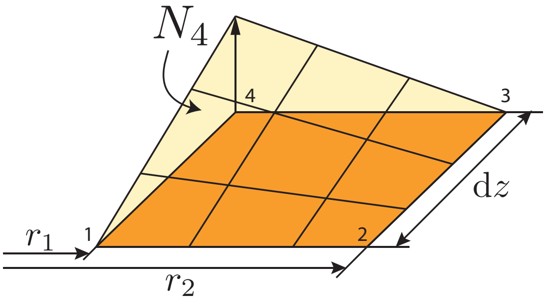


Figure A.3: A rectangular finite element (viewed in perspective) with its fourth basis function drawn on top of it. The basis function is equal to one at node 4, and zero at the other nodes.

For a rectangular element as shown in figure A.3 the four basis functions will satisfy the equations:

$$\begin{bmatrix} 1 & r_1 & 0 & 0 \\ 1 & r_2 & 0 & 0 \\ 1 & r_2 & \mathrm{d}z & r_2\mathrm{d}z \\ 1 & r_1 & \mathrm{d}z & r_1\mathrm{d}z \end{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \delta_1 & \delta_2 & \delta_3 & \delta_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.14}$$

The four basis function of element i are then:

$$\mathbf{N}_i(r,z) = [N_1(r,z) \quad N_2(r,z) \quad N_3(r,z) \quad N_4(r,z)] = [1 \quad r \quad z \quad rz] \left(\begin{bmatrix} 1 & r_1 & 0 & 0 \\ 1 & r_2 & 0 & 0 \\ 1 & r_2 & \mathrm{d}z & r_2\mathrm{d}z \\ 1 & r_1 & \mathrm{d}z & r_1\mathrm{d}z \end{bmatrix} \right)^{-1} \tag{A.15}$$

Now the dependent variables can be approximated by:

$$\tilde{\mathbf{T}}(\mathbf{x},t) = \sum_{i=1}^{n_e} \mathbf{N}_i(\mathbf{x})\mathbf{T}_i(t) \tag{A.16}$$

$$\tilde{\mathbf{u}}(\mathbf{x}) = \sum_{i=1}^{n_e} \begin{bmatrix} \mathbf{N}_i(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_i(\mathbf{x}) \end{bmatrix} \mathbf{u}_i \tag{A.17}$$

Where n_e is the number of elements. $\mathbf{T}_i(t) = [T_1 \quad T_2 \quad T_3 \quad T_4]^T$ contains the temperatures on the four nodes of element i and $\mathbf{u}_i = [u_1 \quad u_2 \quad u_3 \quad u_4 \quad w_1 \quad w_2 \quad w_3 \quad w_4]^T$ contains the radial and axial displacement on the four nodes of element i.

Galerkin method

The equilibrium equations can now also be expressed in terms of the approximated fields. In order to accomplish this, a relation between stress and displacement is needed. For this, the strain-displacement and the stress-strain relations are used [28]:

$$\underbrace{\begin{bmatrix} \varepsilon_r \\ \varepsilon_z \\ \varepsilon_\theta \\ \gamma_{rz} \end{bmatrix}}_{\varepsilon} = \underbrace{\begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{1}{r} & 0 \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix}}_{\partial} \underbrace{\begin{bmatrix} u \\ w \end{bmatrix}}_{\mathbf{u}}, \quad \varepsilon = \partial \mathbf{u} \tag{A.18}$$

$$\underbrace{\begin{bmatrix} \sigma_{rr} \\ \sigma_{zz} \\ \sigma_{\theta\theta} \\ \tau_{rz} \end{bmatrix}}_{\sigma} = \underbrace{\frac{E(1-v)}{(v+1)(1-2v)} \begin{bmatrix} 1 & \frac{v}{1-v} & \frac{v}{1-v} & 0 \\ \frac{v}{1-v} & 1 & \frac{v}{1-v} & 0 \\ \frac{v}{1-v} & \frac{v}{1-v} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2v}{2(1-v)} \end{bmatrix}}_{\mathbf{E}} \left(\underbrace{\begin{bmatrix} \varepsilon_r \\ \varepsilon_z \\ \varepsilon_\theta \\ \gamma_{rz} \end{bmatrix}}_{\varepsilon} - \underbrace{\begin{bmatrix} \alpha T \\ \alpha T \\ \alpha T \\ 0 \end{bmatrix}}_{\varepsilon_0} \right), \quad \sigma = \mathbf{E}\left(\varepsilon - \varepsilon_0\right) \tag{A.19}$$

The approximated stresses are:

$$\tilde{\sigma} = \mathbf{E}\left(\partial\tilde{\mathbf{u}} - \tilde{\varepsilon}_0\right) \tag{A.20}$$

With:

$$\tilde{\varepsilon}_0 = \alpha \underbrace{\begin{bmatrix}\mathbf{N}_i\\ \mathbf{N}_i\\ \mathbf{N}_i\\ \mathbf{0}\end{bmatrix}}_{\mathbf{N}_{\alpha,i}} \mathbf{T}_i = \alpha\mathbf{N}_{\alpha,i}\mathbf{T}_i \tag{A.21}$$

Because the temperatures and stresses are approximated, the equilibrium equations will most likely not be satisfied. A residual will be present:

$$c_p\rho\frac{\partial\tilde{T}}{\partial t} - k\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\tilde{T}}{\partial r}\right) + \frac{\partial^2\tilde{T}}{\partial z^2}\right) - Q = R_T \tag{A.22}$$

$$\frac{\partial\tilde{\sigma}_{rr}}{\partial r} + \frac{\partial\tilde{\tau}_{zr}}{\partial z} + \frac{\tilde{\sigma}_{rr} - \tilde{\sigma}_{\theta\theta}}{r} + F_r = R_u \tag{A.23}$$

$$\frac{\partial\tilde{\sigma}_{zz}}{\partial z} + \frac{\partial\tilde{\tau}_{rz}}{\partial r} + \frac{\tilde{\tau}_{rz}}{r} + F_z = R_w \tag{A.24}$$

The Galerkin method is one of several so called weighted residual methods [31], where the aim is to eliminate the average residuals over the total volume weighted by some function of position. By doing this for as many weighting functions as there are dofs, there are as many equations as there are unknowns and the dofs can be calculated:

$$\int_\Omega W_i R_x \mathrm{d}V = 0, \quad \text{for} \quad i = 1, 2, ..., n_{dof} \tag{A.25}$$

The Galerkin method choses these weighting functions the same as the basis functions. On an element level the equation for element i are:

$$\int_{e_i} \mathbf{N}_i^T R_T \mathrm{d}V_{e_i} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \quad \int_{e_i} \mathbf{N}_i^T R_u \mathrm{d}V_{e_i} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \quad \int_{e_i} \mathbf{N}_i^T R_w \mathrm{d}V_{e_i} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix} \tag{A.26}$$

The order of these functions can be decreased by use of the divergence theorem and the relation $\mathrm{d}V_{e_i} = 2\pi r\mathrm{d}r\mathrm{d}z$. Care should be taken with the extra r that comes from the volume integral. By doing so the following set of equations is obtained:

$$c_p\rho\int_{e_i}\mathbf{N}_i^T\mathbf{N}_i r\mathrm{d}r\mathrm{d}z\dot{\mathbf{T}}_i + k\int_{e_i}\mathbf{J}_{\mathbf{N}_i}\mathbf{J}_{\mathbf{N}_i}^{\mathbf{T}} r\mathrm{d}r\mathrm{d}z\mathbf{T}_i = \int_{e_i}\mathbf{N}_i^T Q r\mathrm{d}r\mathrm{d}z + k\int_{\delta e_i}\mathbf{N}_i^T\left(\mathbf{T}_i\frac{\partial\mathbf{N}_i}{\partial r}n_r + \mathbf{T}_i\frac{\partial\mathbf{N}_i}{\partial z}n_z\right)r\mathrm{d}S \tag{A.27}$$

$$\int_{e_i}\begin{bmatrix}\mathbf{N}_i & \mathbf{0}\\ \mathbf{0} & \mathbf{N}_i\end{bmatrix}^T\partial^T\sigma r\mathrm{d}r\mathrm{d}z = \int_{e_i}\begin{bmatrix}\mathbf{N}_i & \mathbf{0}\\ \mathbf{0} & \mathbf{N}_i\end{bmatrix}^T\begin{bmatrix}F_r\\ F_z\end{bmatrix}r\mathrm{d}r\mathrm{d}z + \int_{\delta e_i}\begin{bmatrix}\mathbf{N}_i & \mathbf{0}\\ \mathbf{0} & \mathbf{N}_i\end{bmatrix}^T\begin{bmatrix}\sigma_{rr}n_r + \tau_{zr}n_z\\ \sigma_{zz}n_z + \tau_{rz}n_r\end{bmatrix}r\mathrm{d}S \tag{A.28}$$

with:

$$\mathbf{J}_{\mathbf{N}_i} = \begin{bmatrix}\frac{\partial\mathbf{N}_i^{\mathbf{T}}}{\partial r} & \frac{\partial\mathbf{N}_i^{\mathbf{T}}}{\partial z}\end{bmatrix} \tag{A.29}$$

n_r and n_z are the r and z components of the surface normal of the integration surface. A boundary flow f_b and traction vector Φ can be defined as [32]:

$$f_b = k\frac{\partial T}{\partial r}n_r + k\frac{\partial T}{\partial z}n_z \tag{A.30}$$

$$\Phi = \begin{bmatrix}\Phi_r\\ \Phi_z\end{bmatrix} = \begin{bmatrix}\sigma_{rr}n_r + \tau_{zr}n_z\\ \sigma_{zz}n_z + \tau_{rz}n_r\end{bmatrix} \tag{A.31}$$

By using these definitions, the stress-displacement relationships and by defining:

$$\mathbf{N}_{disp,i} = \begin{bmatrix} \mathbf{N}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{N}_i \end{bmatrix} \tag{A.32}$$

$$\mathbf{B}_i = \partial \mathbf{N}_{disp,i} \tag{A.33}$$

the equations A.27 and A.28 can be rewritten and structural matrices can be defined:

$$\underbrace{c_p\rho \int_{e_i} \mathbf{N}_i^T \mathbf{N}_i r\mathrm{d}r\mathrm{d}z}_{\mathbf{C_{T_e}}} \dot{\mathbf{T}}_i + \underbrace{k \int_{e_i} \mathbf{J}_{\mathbf{N}_i} \mathbf{J}_{\mathbf{N}_i}^{\mathbf{T}} r\mathrm{d}r\mathrm{d}z}_{\mathbf{K_{T_e}}} \mathbf{T}_i = \underbrace{\int_{e_i} \mathbf{N}_i^T Q r\mathrm{d}r\mathrm{d}z}_{\mathbf{Q_e}} + \int_{\delta e_i} \mathbf{N}_i^T f_b r\mathrm{d}S \tag{A.34}$$

$$\underbrace{\int_{e_i} \mathbf{B}_i^T \mathbf{E} \mathbf{B} r\mathrm{d}r\mathrm{d}z}_{\mathbf{K_{U_e}}} \mathbf{u}_i = \underbrace{\alpha \int_{e_i} \mathbf{B}_i^T \mathbf{E} \mathbf{N}_{\alpha,i} r\mathrm{d}r\mathrm{d}z}_{\mathbf{A_e}} \mathbf{T}_i + \int_{e_i} \mathbf{N}_{disp,i}^T \begin{bmatrix} F_r \\ F_z \end{bmatrix} r\mathrm{d}r\mathrm{d}z + \int_{\delta e_i} \mathbf{N}_{disp,i}^T \Phi r\mathrm{d}S \tag{A.35}$$

Since no body forces and boundary flows and forces are present in the optimization domain, equations A.26 become:

$$\mathbf{C_{T_e}} \dot{\mathbf{T}}_i(\mathrm{t}) + \mathbf{K_{T_e}} \mathbf{T}_i(\mathrm{t}) = \mathbf{Q}_e(\mathrm{t}), \quad \text{for} \quad i = 1, 2, ..., n_e \tag{A.36}$$

$$\mathbf{K_{U_e}} \mathbf{u}_i(\mathrm{t}) = \mathbf{A_e} \mathbf{T}_i(\mathrm{t}), \quad \text{for} \quad i = 1, 2, ..., n_e \tag{A.37}$$

These element matrices are compared with the element matrices by COMSOL. The results are given in appendix B. The local element dofs can be written as a function of a vector containing all the dofs and a selection matrix:

$$\mathbf{T}_i = \mathbf{L_T}_i \mathbf{T} \tag{A.38}$$

$$\mathbf{u}_i = \mathbf{L_U}_i \mathbf{u} \tag{A.39}$$

Here the vectors $\mathbf{T}$ and $\mathbf{u}$ contain all the global dofs and the matrices $\mathbf{L_T}_i$ and $\mathbf{L_U}_i$ are selection matrices containing only ones and zeros. By doing so the equations A.36 and A.37 can be added for all the elements to create the global matrices. In order to keep the equations separate, they are also premultiplied with the transpose of the selection vector to arrive at the transient thermo-mechanical state equations:

$$\underbrace{\sum_{i=1}^{n_e} \left[\mathbf{L_T}_i^T \mathbf{C_{T_e}} \mathbf{L_T}_i \right]}_{\mathbf{C_T}} \dot{\mathbf{T}}(\mathrm{t}) + \underbrace{\sum_{i=1}^{n_e} \left[\mathbf{L_T}_i^T \mathbf{K_{T_e}} \mathbf{L_T}_i \right]}_{\mathbf{K_T}} \mathbf{T}(\mathrm{t}) = \underbrace{\sum_{i=1}^{n_e} \left[\mathbf{L_T}_i^T \mathbf{Q}_e(\mathrm{t}) \right]}_{\mathbf{Q}} \tag{A.40}$$

$$\underbrace{\sum_{i=1}^{n_e} \left[\mathbf{L_U}_i^T \mathbf{K_{U_e}} \mathbf{L_U}_i \right]}_{\mathbf{K_U}} \mathbf{u}_i(\mathrm{t}) = \underbrace{\sum_{i=1}^{n_e} \left[\mathbf{L_U}_i^T \mathbf{A_e} \mathbf{L_T}_i \right]}_{\mathbf{A}} \mathbf{T}(\mathrm{t}) \tag{A.41}$$

B

Comparison of the element matrices

The element matrices as derived in appendix A are implemented in a Matlab function that returns all the relevant element matrices for a rectangular element at a certain radius and with a certain height and radial length. In this appendix the matrices outputted by this file are compared with the matrices outputted by COMSOL for a similar element. The element is chosen as an element at a radius of 1m, and with length and height of 0.1m. The results of the element by element division of the Matlab matrix by the COMSOL matrix is given in the following matrices:

$$\frac{\mathbf{K_{UU}}_{m,ij}}{\mathbf{K_{UU}}_{C,ij}} - 1 = 1 \cdot 10^{-5} \begin{bmatrix} 0.0369 & -0.0000 & 0.0569 & -0.0000 & 0.0329 & -0.0000 & 0.2283 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0569 & 0.0000 & 0.0292 & 0.0000 & 0.1849 & 0.0000 & 0.0329 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 \\ 0.0329 & -0.0000 & 0.1849 & -0.0000 & 0.0292 & -0.0000 & 0.0569 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.2283 & 0.0000 & 0.0329 & -0.0000 & 0.0569 & 0.0000 & 0.0369 & -0.0000 \\ -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 \end{bmatrix} \tag{B.1}$$

$$\frac{\mathbf{A}_{m,ij}}{\mathbf{A}_{C,ij}} - 1 = 1 \cdot 10^{-15} \begin{bmatrix} -0.1110 & 0.2220 & 0.0000 & 0.0000 \\ -0.6661 & -0.3331 & 0.0000 & -0.3331 \\ -0.1110 & 0.2220 & 0.0000 & 0.2220 \\ 0.2220 & 0.2220 & 0.4441 & 0.2220 \\ -0.1110 & 0.0000 & -0.1110 & -0.1110 \\ -0.3331 & 0.2220 & 0.2220 & -0.3331 \\ -0.1110 & 0.0000 & 0.0000 & -0.1110 \\ -0.1110 & 0.0000 & 0.2220 & 0.2220 \end{bmatrix} \tag{B.2}$$

$$\frac{\mathbf{C_{TT}}_{m,ij}}{\mathbf{C_{TT}}_{C,ij}} - 1 = 1 \cdot 10^{-15} \begin{bmatrix} -0.1110 & -0.2220 & 0.0000 & 0.0000 \\ -0.2220 & 0.2220 & 0.4441 & 0.0000 \\ 0.0000 & 0.4441 & 0.6661 & 0.2220 \\ 0.0000 & 0.0000 & 0.2220 & -0.1110 \end{bmatrix} \tag{B.3}$$

$$\frac{\mathbf{K_{TT}}_{m,ij}}{\mathbf{K_{TT}}_{C,ij}} - 1 = 1 \cdot 10^{-13} \begin{bmatrix} 0.0377 & -0.0222 & -0.0011 & 0.2998 \\ -0.0222 & 0.0933 & 0.2776 & 0.0022 \\ -0.0011 & 0.2776 & 0.0910 & -0.0255 \\ 0.2998 & 0.0022 & -0.0255 & 0.0400 \end{bmatrix} \tag{B.4}$$

It can be seen that the relative error of the matrices $\mathbf{A}$, $\mathbf{C_{TT}}$ and $\mathbf{K_{TT}}$ is in the range of numerical noise (hence the change of sign). The relative error of the stiffness matrix is somewhat higher. This is most likely caused by the division by the radius that happens in the stiffness element (the error decreases at larger radii). The closer the element is to the symmetry axis, the closer it is to division by zero. COMSOL compensates for this effect, the Matlab code doesn't. This is not done because this effect is not dominant on the radius where the Matlab elements are placed. For a typical element with the smallest radius as used in the optimization loop, the relative error showed to be at maximum $1.3 \cdot 10^{-5}\%$.

C

Multi-material sensitivities

In this appendix the derivation of the multi-material sensitivities will be explained. The sensitivities of the multi material problem are equal to the sensitivities of the single material transient problem as described in section 2.2 up to the point where the actual matrix derivatives $\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}$, $\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}$, $\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}$ and $\frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}$ are calculated. This makes sense because until then, it is not defined what the design variables $\mathbf{v}$ actually are (they can still be both ρ_d and ρ_m). So we can start from the last equation of section 2.2:

$$\frac{\mathrm{d}f}{\mathrm{d}v_i} = \frac{\mathrm{d}f^*}{\mathrm{d}v_i} = \int_0^{t_f} \left[\lambda^T \left(\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}\mathbf{u} - \frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}\mathbf{T} \right) + \mu^T \left(\frac{\mathrm{d}\mathbf{C_T}}{\mathrm{d}v_i}\dot{\mathbf{T}} + \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}\mathbf{T} - \frac{\mathrm{d}\mathbf{Q}}{\mathrm{d}v_i} \right) \right] \mathrm{d}t \tag{C.1}$$

$$\lambda_k^T = -\mathbf{K_U}^{-1} \left(\frac{\partial p_k}{\partial \mathbf{u}} \right) \tag{C.2}$$

$$\mu_k^T = \mathbf{E}_\mathbf{T}^{-1} \left(\lambda_k^T \mathbf{A} - \mathbf{A_T} \mu_{k+1}^T \right) \tag{C.3}$$

$$\mu^T(t_f) = \mathbf{0} \tag{C.4}$$

Now the matrix derivatives $\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i}$, $\frac{\mathrm{d}\mathbf{A}}{\mathrm{d}v_i}$, $\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}$ and $\frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}$ can be worked out in more detail. This will be done for the stiffness matrix only since it involves the most terms and the rest is quite similar. The stiffness matrix is defined as (see equations A.40 and A.41):

$$\mathbf{K_U} = \sum_{i=1}^{n_e} \left[\mathbf{L_U}_i^T \mathbf{K_U}_{\mathbf{e}_i} \mathbf{L_U}_i \right] \tag{C.5}$$

Here $\mathbf{L_U}_i$ are the boolean selection matrices, $\mathbf{K_U}_{\mathbf{e}_i}$ the element stiffness matrices and n_e the number of elements. The stiffness matrix derivative with respect to the design variables is then:

$$\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i} = \sum_{i=1}^{n_e} \left[\mathbf{L_U}_i^T \frac{\mathrm{d}\mathbf{K_U}_{\mathbf{e_i}}}{\mathrm{d}v_i} \mathbf{L_U}_i \right] \tag{C.6}$$

Since the element stiffness matrix is defined as:

$$\mathbf{K_U}_{\mathbf{e_i}} = \int_{e_i} \mathbf{B}_i^T \mathbf{E} \mathbf{B}_i r \mathrm{d}r \mathrm{d}z \tag{C.7}$$

and only the matrix $\mathbf{E}$ is dependent on the material parameters, the stiffness matrix derivatives are:

$$\frac{\mathrm{d}\mathbf{K_U}}{\mathrm{d}v_i} = \sum_{i=1}^{n_e} \left[\mathbf{L_U}_i^T \int_{e_i} \mathbf{B}_i^T \frac{\mathrm{d}\mathbf{E}}{\mathrm{d}v_i} \mathbf{B}_i r \mathrm{d}r \mathrm{d}z \mathbf{L_U}_i \right] \tag{C.8}$$

The matrix $\mathbf{E}$ is defined as (equation A.19):

$$\mathbf{E} = \frac{E^*(v^*-1)}{(v^*+1)(1-2v^*)} \begin{bmatrix} 1 & \frac{v^*}{1-v^*} & \frac{v^*}{1-v^*} & 0 \\ \frac{v^*}{1-v^*} & 1 & \frac{v^*}{1-v^*} & 0 \\ \frac{v^*}{1-v^*} & \frac{v^*}{1-v^*} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2v^*}{2(1-v^*)} \end{bmatrix} \tag{C.9}$$

The design variables are now specified as:

$$\mathbf{v} = \begin{bmatrix}\rho_{\mathsf{d}} & \rho_{\mathsf{m}}\end{bmatrix}^T \tag{C.10}$$

The first half of the vector of design variables are density variables, while the second half are the material variables. It can be seen that when deriving with respect to the density variables ρ_{d}, only the term for the interpolated Young's modulus E^* has to be derived to ρ_{d}. For a single material, this term could easily be taken out of the integral in equation C.7. The element stiffness matrices could be calculated without the Young's modulus at the start of the optimization, and then only need to be scaled during the optimization. With the changing interpolated Poisson's ratio ν^*, this can no longer be done and the element stiffness matrices have to be calculated at every iteration. The derivatives of the matrix $\mathbf{E}$ with respect to the design variables are:

$$\frac{\mathrm{d}\mathbf{E}}{\mathrm{d}\rho_{\mathsf{d}}} = \frac{\partial \mathbf{E}}{\partial E^*}\frac{\mathrm{d}E^*}{\mathrm{d}\rho_{\mathsf{d}}} = \frac{(\nu^*-1)}{(\nu^*+1)(1-2\nu^*)}\begin{bmatrix} 1 & \frac{\nu^*}{1-\nu^*} & \frac{\nu^*}{1-\nu^*} & 0 \\ \frac{\nu^*}{1-\nu^*} & 1 & \frac{\nu^*}{1-\nu^*} & 0 \\ \frac{\nu^*}{1-\nu^*} & \frac{\nu^*}{1-\nu^*} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu^*}{2(1-\nu^*)} \end{bmatrix}\frac{\mathrm{d}E^*}{\mathrm{d}\rho_{\mathsf{d}}} \tag{C.11}$$

$$\begin{aligned} \frac{\mathrm{d}\mathbf{E}}{\mathrm{d}\rho_{\mathsf{m}}} &= \frac{\partial \mathbf{E}}{\partial E^*}\frac{\mathrm{d}E^*}{\mathrm{d}\rho_{\mathsf{m}}} + \frac{\partial \mathbf{E}}{\partial \nu^*}\frac{\mathrm{d}\nu^*}{\mathrm{d}\rho_{\mathsf{m}}} \\ &= \frac{(\nu^*-1)}{(\nu^*+1)(1-2\nu^*)}\begin{bmatrix} 1 & \frac{\nu^*}{1-\nu^*} & \frac{\nu^*}{1-\nu^*} & 0 \\ \frac{\nu^*}{1-\nu^*} & 1 & \frac{\nu^*}{1-\nu^*} & 0 \\ \frac{\nu^*}{1-\nu^*} & \frac{\nu^*}{1-\nu^*} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu^*}{2(1-\nu^*)} \end{bmatrix}\frac{\mathrm{d}E^*}{\mathrm{d}\rho_{\mathsf{m}}} + \\ &\qquad \frac{-2E^*\nu^*(\nu^*-2)}{(\nu^*+1)^2(2\nu^*-1)^2}\begin{bmatrix} 1 & -\frac{2\nu^{*2}+1}{2\nu^*(\nu^*-2)} & -\frac{2\nu^{*2}+1}{2\nu^*(\nu^*-2)} & 0 \\ -\frac{2\nu^{*2}+1}{2\nu^*(\nu^*-2)} & 1 & -\frac{2\nu^{*2}+1}{2\nu^*(\nu^*-2)} & 0 \\ -\frac{2\nu^{*2}+1}{2\nu^*(\nu^*-2)} & -\frac{2\nu^{*2}+1}{2\nu^*(\nu^*-2)} & 1 & 0 \\ 0 & 0 & 0 & \frac{(2\nu^*-1)^2}{4\nu^*(\nu^*-2)} \end{bmatrix}\frac{\mathrm{d}\nu^*}{\mathrm{d}\rho_{\mathsf{m}}} \end{aligned} \tag{C.12}$$

These derivatives can be substituted into equation C.8 to calculate the stiffness matrix derivatives with respect to the design variables in a multi-material optimization. The same can be done for the coupling matrix $\mathbf{A}$, where the same matrix $\mathbf{E}$ is used. In both the heat conduction and heat capacity matrix the Poisson's ratio does not occur, so those matrix derivatives do not contain the partial derivative with respect the Poisson ratio term when deriving for the material design variables.

D

Calculating integrals involving matrix exponentials

This appendix will show a method to solve the integrals:

$$\frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}}{\mathbf{K_T}}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\mathbf{C}^{-1}\left(\frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}{\mathbf{K_T}}^{-1}\mathbf{C}-\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\right)\mathbf{H}(t)\,\mathrm{d}t\,\mathbf{C}^{-1}\mathbf{Q} \tag{D.1}$$

$$\frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}}{\mathbf{K_T}}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\mathbf{C}^{-1}\left(\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}\mathbf{C}^{-1}\mathbf{K_T}-\frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}\right)\mathbf{H}(t)\,\mathrm{d}t\,\mathbf{T}(0) \tag{D.2}$$

Only half of the first integral will be treated:

$$\frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}}{\mathbf{K_T}}^{-1}\mathbf{C}\int_0^{t_f}\mathbf{H}(t_f-t)\mathbf{C}^{-1}\frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}{\mathbf{K_T}}^{-1}\mathbf{C}\mathbf{H}(t)\,\mathrm{d}t\,\mathbf{C}^{-1}\mathbf{Q} \tag{D.3}$$

The same method can be used for the second integral. In order to efficiently solve these integrals, the matrices $\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}v_i}$ and $\frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}$ should be separated because they change with every design variable to which the objective function is differentiated. We first define the matrix $\mathbf{G} = -\mathbf{C}^{-1}\mathbf{K_T}$, and perform the diagonalization:

$$\mathbf{G} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1} \tag{D.4}$$

Here $\mathbf{P}$ is a matrix whose columns represent the eigenvectors of $\mathbf{G}$, and $\mathbf{D}$ is a diagonal matrix with eigenvalues of $\mathbf{G}$ on its diagonal. This gives:

$$\mathbf{H}(t) = e^{-\mathbf{C}^{-1}\mathbf{K_T}t} = e^{\mathbf{P}\mathbf{D}\mathbf{P}^{-1}t} = \mathbf{P}e^{\mathbf{D}t}\mathbf{P}^{-1} \tag{D.5}$$

The integral now is:

$$\underbrace{\frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}}{\mathbf{K_T}}^{-1}\mathbf{C}\mathbf{H}(t_f)\int_0^{t_f}\mathbf{P}}_{\mathbf{a^T}}\,e^{-\mathbf{D}t}\,\underbrace{\mathbf{P}^{-1}\mathbf{C}^{-1}}_{\mathbf{A}}\,\underbrace{\frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i}}_{\mathbf{K}}\,\underbrace{{\mathbf{K_T}}^{-1}\mathbf{C}\mathbf{P}}_{\mathbf{B}}\,e^{\mathbf{D}t}\,\underbrace{\mathbf{P}^{-1}\,\mathrm{d}t\,\mathbf{C}^{-1}\mathbf{Q}}_{\mathbf{b}} \tag{D.6}$$

$$= \mathbf{a^T}\int_0^{t_f} e^{-\mathbf{D}t}\mathbf{A}\mathbf{K}\mathbf{B}e^{\mathbf{D}t}\,\mathrm{d}t\,\mathbf{b} \tag{D.7}$$

Because $\mathbf{D}$ is a diagonal matrix, the exponential term can be evaluated element wise:

$$= \sum_{k,l,m,n=1}^{N} a_k \int_0^{t_f} e^{-D_{kk}t} A_{kl} K_{lm} B_{mn} e^{D_{nn}t} \,\mathrm{d}t \, b_n \tag{D.8}$$

$$= \sum_{k,l,m,n=1}^{N} a_k A_{kl} K_{lm} B_{mn} b_n \int_0^{t_f} e^{(D_{kk}-D_{nn})t} \,\mathrm{d}t \tag{D.9}$$

$$= \begin{cases} \sum_{k,l,m,n=1}^{N} a_k A_{kl} K_{lm} B_{mn} b_n \frac{1}{D_{kk}-D_{nn}} \left(e^{(D_{kk}-D_{nn})t_f} - 1 \right) & k \neq n \\ \sum_{k,l,m,n=1}^{N} a_k A_{kl} K_{lm} B_{mn} b_n t_f & k = n \end{cases} \tag{D.10}$$

A matrix $\mathbf{R}$ is defined as:

$$R_{kn} = \begin{cases} \frac{1}{D_{kk}-D_{nn}} \left(e^{(D_{kk}-D_{nn})t_f} - 1 \right) & k \neq n \\ t_f & k = n \end{cases} \tag{D.11}$$

Now we can write the integral as:

$$= \sum_{k,l,m,n=1}^{N} a_k A_{kl} K_{lm} B_{mn} b_n R_{kn} \tag{D.12}$$

$$= \sum_{l,m=1}^{N} \left[K_{lm} \underbrace{\sum_{k,n=1}^{N} a_k A_{kl} B_{mn} b_n R_{kn}}_{X_{lm}} \right] \tag{D.13}$$

$$= \mathbf{K} \circ \mathbf{X} \tag{D.14}$$

Here $\circ$ is the Hadamard (elementwise) product. $\mathbf{X}$ can be rewritten as:

$$\mathbf{X} = \sum_{k,n=1}^{N} a_k^T A_{kl} R_{kn} B_{mn} b_n \tag{D.15}$$

$$= \mathbf{aARBb} \tag{D.16}$$

Summarizing this gives:

$$\frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}} \mathbf{K_T}{}^{-1} \mathbf{C} \int_0^{t_f} \mathbf{H}(t_f - t) \mathbf{C}^{-1} \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i} \mathbf{K_T}{}^{-1} \mathbf{C} \mathbf{H}(t) \,\mathrm{d}t \, \mathbf{C}^{-1} \mathbf{Q} \tag{D.17}$$

$$= \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i} \circ \left(\frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}} \mathbf{K_T}{}^{-1} \mathbf{C} \mathbf{H}(t_f) \mathbf{P} \mathbf{P}^{-1} \mathbf{C}^{-1} \mathbf{R} \mathbf{K_T}{}^{-1} \mathbf{C} \mathbf{P} \mathbf{P}^{-1} \mathbf{C}^{-1} \mathbf{Q} \right) \tag{D.18}$$

$$= \frac{\mathrm{d}\mathbf{K_T}}{\mathrm{d}v_i} \circ \left(\frac{\partial p(\mathbf{v},\mathbf{T})}{\partial \mathbf{T}} \mathbf{K_T}{}^{-1} \mathbf{C} \mathbf{H}(t_f) \mathbf{C}^{-1} \mathbf{R} \mathbf{K_T}{}^{-1} \mathbf{Q} \right) \tag{D.19}$$

Since the right side of the Hadamard product is independent of the design variable to which the objective function is being differentiated, it only needs to be calculated once per iteration.

E

RAMP interpolation on strict monotonic functions

RAMP interpolation has the property to make every strictly monotonically increasing or decreasing function convex or concave, depending on the penalization parameter. Let us consider a function $f(x)$ which is strictly monotonically increasing on the interval $0 \leq x \leq 1$, meaning that $\frac{\mathrm{d}f}{\mathrm{d}x} > 0$. Let us also define a RAMP function $g(x)$ with penalization parameter q:

$$g(x) = \frac{x}{1+q(1-x)} \tag{E.1}$$

Applying RAMP interpolation to $f(x)$ means substituting x for $g(x)$ giving $f(g(x))$. In order for $f(g(x))$ to be convex or concave, its second derivative needs to be greater or smaller than 0 respectively on the interval $0 \leq x \leq 1$. Differentiating twice w.r.t. x gives:

$$\frac{\mathrm{d}^2 f(g(x))}{\mathrm{d}x^2} = \frac{\partial^2 f}{\partial x^2}(g(x)) \left(\frac{\mathrm{d}g(x)}{\mathrm{d}x}\right)^2 + \frac{\partial f}{\partial x}(g(x)) \frac{\mathrm{d}^2 g(x)}{\mathrm{d}x^2} \tag{E.2}$$

Writing out the derivatives of the RAMP function gives:

$$\frac{\mathrm{d}g(x)}{\mathrm{d}x} = \frac{q+1}{(q(x-1)-1)^2} \tag{E.3}$$

$$\frac{\mathrm{d}^2 g(x)}{\mathrm{d}x^2} = \frac{2(q+1)q}{(q(1-x)+1)^3} \tag{E.4}$$

Notice that the second derivative $\frac{\mathrm{d}^2 g(x)}{\mathrm{d}x^2}$ is either negative or positive on the interval $0 \leq x \leq 1$, depending on when $-1 < q < 0$ or $q > 0$, respectively, or zero for $q = 0$). The square of the first derivative is always positive due to the square. The sign of the first derivative of $f(x)$ is positive because it is strictly monotonically increasing, and its second derivative can be both positive and negative:

$$\frac{\mathrm{d}^2 f(g(x))}{\mathrm{d}x^2} = \underbrace{\frac{\partial^2 f}{\partial x^2}(g(x))}_{\text{sign unkown}} \underbrace{\left(\frac{\mathrm{d}g(x)}{\mathrm{d}x}\right)^2}_{\geq 0} + \underbrace{\frac{\partial f}{\partial x}(g(x))}_{>0} \underbrace{\frac{\mathrm{d}^2 g(x)}{\mathrm{d}x^2}}_{\text{dependent on q}} \tag{E.5}$$

Let us look at the ratio of the second derivative versus the squared first derivative:

$$\frac{\frac{\mathrm{d}^2 g(x)}{\mathrm{d}x^2}}{\left(\frac{\mathrm{d}g(x)}{\mathrm{d}x}\right)^2} = \frac{2(q^2(1-x)+q)}{q+1} \tag{E.6}$$

This ratio can approach $-\infty$ or $+\infty$ by making q approach -1 or ∞. This means that by tuning q, the last term in equation E.5 can always make equation E.5 positive or negative because the sign of this term can be chosen with q, and the magnitude can be made larger than the first term, making the complete equation positive or negative. This makes the interpolated function either convex or concave. If $f(x)$ would have been strictly monotonically decreasing, this also holds, except that q has to approach -1 for convexity and ∞ for concavity instead of the other way around.

F

Penalization parameters over iterations

In this appendix extra information about the penalization parameters from the topology optimization as performed in section 4.5 with material penalization are displayed.

In figure F.1 the configuration of penalization parameters for the final design is plotted for the different material properties. Especially in the bottom half the penalization parameters are scattered over the domain. This is mainly due to numerical noise that dominates this section. In principle one could determine what type of material is wanted in which position. For example in areas where the heat capacity is penalized positively, more heat capacity is desired, and where it's penalized negatively, less heat capacity is desired. As can be seen the penalization parameters are very local and differ from element to element. A single set of penalization parameters for the whole domain does not make sense because every element has a different function in the structure.

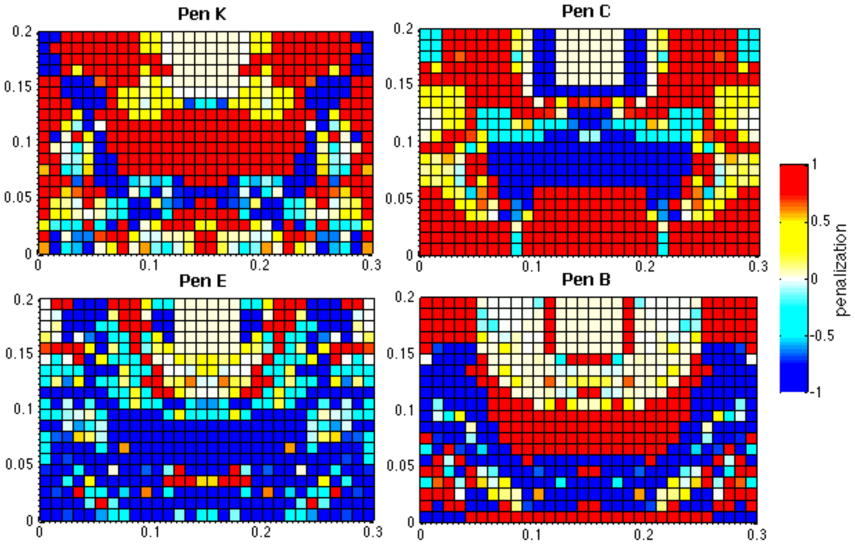


Figure F.1: The final penalization parameters for the different material properties for every element.

The speed a which the penalization parameters change also differs from element to element, and from penalization parameter to penalization parameter. In figure F.2 and figure F.3 the penalization parameters over the iterations have been plotted for two almost neighbouring elements, close to the area where the heat input is applied. It can be seen that for one element the penalization parameters change rapidly, while for the other element it is more constant. That is because in the element in figure

F.3, there are a lot of oscillations which keeps the move limits low compared to the move limits of the element in figure F.2.

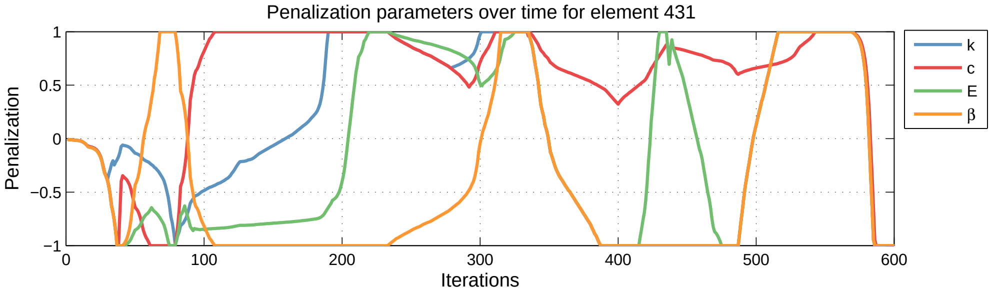


Figure F.2: The penalization parameters of element 431 over the iterations. It can be seen that the penalization parameters are constantly changing in this element.

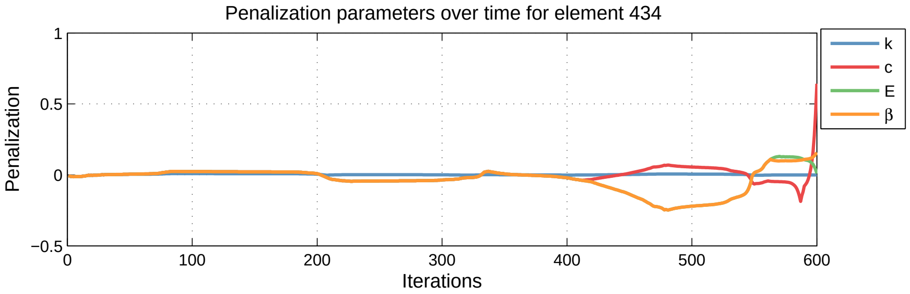


Figure F.3: The penalization parameters of element 434 over the iterations. In this element the penalization are quite constant over the iterations.

G

Static condensation

In order to speed up the model, static condensation was applied on the structural matrices of the COMSOL substructure. Since the structural response is considered as being static, this is an exact procedure. The thermal part is not static and is therefore not reduced. Reduction methods such as the Craig-Bampton method can be used to also reduce the thermal part, this is however not done in this study. Only the interface DOFs between the two substructures and the DOFs used in the objective function are retained, the rest is condensed out. This can be done by re-ordering the structural matrices by separating the set of DOFs that are to be retained as $\mathbf{u}_1$, and the DOFs that should be condensed out as $\mathbf{u}_2$:

$$\begin{bmatrix} \mathbf{K}_{\mathbf{U}_{11}} & \mathbf{K}_{\mathbf{U}_{12}} \\ \mathbf{K}_{\mathbf{U}_{21}} & \mathbf{K}_{\mathbf{U}_{22}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} [\mathbf{T}] \tag{G.1}$$

By writing down and reordering the bottom equation, $\mathbf{u}_2$ can be written as:

$$\mathbf{u}_2 = {\mathbf{K}_{\mathbf{U}_{22}}}^{-1} \left(\mathbf{A}_2 \mathbf{T} - \mathbf{K}_{\mathbf{U}_{21}} \mathbf{u}_1 \right) \tag{G.2}$$

Substituting this in the top equation gives:

$$\underbrace{\left(\mathbf{K}_{\mathbf{U}_{11}} - \mathbf{K}_{\mathbf{U}_{12}} {\mathbf{K}_{\mathbf{U}_{22}}}^{-1} \mathbf{K}_{\mathbf{U}_{21}} \right)}_{\tilde{\mathbf{K}}_{\mathbf{U}}} \mathbf{u}_1 = \underbrace{\left(\mathbf{A}_1 + \mathbf{K}_{\mathbf{U}_{12}} {\mathbf{K}_{\mathbf{U}_{22}}}^{-1} \mathbf{A}_2 \right)}_{\tilde{\mathbf{A}}} \mathbf{T}, \qquad \tilde{\mathbf{K}}_{\mathbf{U}} \mathbf{u}_1 = \tilde{\mathbf{A}} \mathbf{T} \tag{G.3}$$

For the analysis as done in order to obtain the results from figure **??**, the decomposition of the stiffness matrix took roughly 20% of the time of the unreduced decomposition. The total time savings are problem dependent and thus hard to define, because the decomposition is only a (albeit substantial) portion of the total calculation.

Bibliography

[1] J. Mayr, J. Jedrzejewski, E. Uhlmann, M. A. Donmez, W. Knapp, F. Härtig, K. Wendt, T. Moriwaki, P. Shore, R. Schmitt, C. Brecher, T. Würz, and K. Wegener, *Thermal issues in machine tools,* {CIRP} Annals - Manufacturing Technology **61**, 771 (2012).

[2] R. Lamers, *When precision cannot stand the heat,* Mikroniek (2010).

[3] M. Bendsøe, *Optimal shape design as a material distribution problem,* Structural optimization **1**, 193 (1989).

[4] J. D. Deaton and R. V. Grandhi, *A survey of structural and multidisciplinary continuum topology optimization: post 2000,* Structural and Multidisciplinary Optimization **49**, 1 (2014).

[5] T. Gao and W. Zhang, *Topology optimization involving thermo-elastic stress loads,* Structural and Multidisciplinary Optimization **42**, 725 (2010).

[6] R. T. Haftka, *Techniques for thermal sensitivity analysis,* International Journal for Numerical Methods in Engineering **17**, 71 (1981).

[7] G. N. Vanderplaats, *Comment on "methods of design sensitivity analysis in structural optimization",* AIAA Journal **18**, 1406 (1980).

[8] K. Svanberg, *The method of moving asymptotes—a new method for structural optimization,* International Journal for Numerical Methods in Engineering **24**, 359 (1987).

[9] O. Sigmund and K. Maute, *Topology optimization approaches,* Structural and Multidisciplinary Optimization **48**, 1031 (2013).

[10] K. Svanberg, *MMA and GCMMA – version September 2007*, Tech. Rep. (KTH, Stockholm, Sweden, 2007).

[11] M. Bendsoe and O. Sigmund, *Topology Optimization: Theory, Methods and Applications*, Engineering online library (Springer, 2003).

[12] L. Yin and G. Ananthasuresh, *A novel topology design scheme for the multi-physics problems of electro-thermally actuated compliant micromechanisms,* Sensors and Actuators A: Physical **97–98**, 599 (2002).

[13] B. Wang, J. Yan, and G. Cheng, *Optimal structure design with low thermal directional expansion and high stiffness,* Engineering Optimization **43**, 581 (2011).

[14] K. Dems, *Sensitivity analysis in thermal problems—i: Variation of material parameters within a fixed domain,* Journal of Thermal Stresses **9**, 303 (1986).

[15] D. A. Tortorelli and R. B. Haber, *First-order design sensitivities for transient conduction problems by an adjoint method,* International Journal for Numerical Methods in Engineering **28**, 733 (1989).

[16] A. D. Belegundu and J. S. Arora, *A sensitivity interpretation of adjoint variables in optimal design,* Computer Methods in Applied Mechanics and Engineering **48**, 81 (1985).

[17] J. S. Arora and E. J. Haug, *Methods of design sensitivity analysis in structural optimization,* AIAA Journal **17**, 970 (1979).

[18] D. Rixen, *Engineering dynamics,* (2009).

[19] H. A. Priestley, *Introduction to integration,* (Oxford University Press, USA, 1997) Chap. 26: Fubini's Theorem and Tonelli's Theorem, pp. 197–198.

[20] C. V. Loan, *Computing integrals involving the matrix exponential,* IEEE Transactions on Automatic Control **23**, 395 (1978).

[21] S. Cho and J.-Y. Choi, *Efficient topology optimization of thermo-elasticity problems using coupled field adjoint sensitivity analysis method,* Finite Elements in Analysis and Design **41**, 1481 (2005).

[22] Y. Li, K. Saitou, and N. Kikuchi, *Topology optimization of thermally actuated compliant mechanisms considering time-transient effect,* Finite Elements in Analysis and Design **40**, 1317 (2004).

[23] L. A. M. Mello, R. A. Salas, and E. C. N. Silva, *On response time reduction of electrothermomechanical mems using topology optimization,* Computer Methods in Applied Mechanics and Engineering **247–248**, 93 (2012).

[24] O. Sigmund, *Design of multiphysics actuators using topology optimization part i: One-material structures,* Computer Methods in Applied Mechanics and Engineering **190**, 6577 (2001).

[25] J. E. Kim, D. S. Kim, P. S. Ma, and Y. Y. Kim, *Multi-physics interpolation for the topology optimization of piezoelectric systems,* Computer Methods in Applied Mechanics and Engineering **199**, 3153 (2010).

[26] J. Y. Noh and G. H. Yoon, *Topology optimization of piezoelectric energy harvesting devices considering static and harmonic dynamic loads,* Advances in Engineering Software **53**, 45 (2012).

[27] R. Schmidt, G. Schitter, and J. van Eijk, *The design of high performance mechatronics: High-tech functionality by multidisciplinary system integration,* (Delft University Press, 2011) Chap. 7: Optics in mechatronic systems, p. 521.

[28] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, *Concepts and applications of finite element analysis, 4th edition,* (Wiley, 2001) Chap. 14: Solids of revolution, pp. 508–512, 4th ed.

[29] W. Janna, *Engineering heat transfer,* (Van Nostrand Reinhold International, City, 1988) Chap. 2: The general conduction equation, p. 43.

[30] A. Mills, *Basic heat and mass transfer,* (Prentice Hall, 1999) Chap. 2: Steady one-dimensional heat conduction.

[31] R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, *Concepts and applications of finite element analysis, 4th edition,* (Wiley, 2001) Chap. 5: Formulation techniques: Galerkin and other weighted residual methods, pp. 179–202, 4th ed.

[32] R. Hibbeler, *Mechanics of materials,* (Prentice Hall PTR, 2011) Chap. 9: Stress Transformation.