A TWO-MATERIAL TOPOLOGY OPTIMIZATION METHOD FOR THE
DESIGN OF A PASSIVE THERMAL CONTROL INTERFACE

A Thesis in
Aerospace Engineering
by
Pierre F. Thurier

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2014
The thesis of Pierre F. Thurier was reviewed and approved* by the following:

George A. Lesieutre
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering
Thesis Co-Adviser

Mary I. Frecker
Professor of Mechanical Engineering
Thesis Co-Adviser

James H. Adair
Professor of Materials Science and Engineering

*Signatures are on file in the Graduate School.
Abstract

The design of thermal subsystems is a central concern for a spacecraft that carries a significant amount of electronics. These electronic components are subject to large deviations in temperature once in space. To maintain these components within an allowable operational range, this thesis describes ways to optimize the design of a thermo-mechanical interface that passively regulates the heat transfer between the electronics and the thermal bus of the spacecraft. The interface constitutes a sandwich panel composed of cellular contact-aided compliant mechanisms arranged into an array structure. Under changes in temperature, the compliant mechanisms create or break heat conduction paths with the thermal bus, effecting a “thermal valve”. Using multiple materials, this concept stands as a new solution to achieve passive thermal control.

The contact-aided compliant mechanisms are made of two isotropic materials. One has a high coefficient of thermal expansion and high conductivity (“metal”), the other has a low coefficient of thermal expansion and low conductivity (“ceramic”). Under various thermal conditions, the materials contract and expand at different rates, modifying the contact pressure and thereby affecting the net heat flux through the interface. Based on a 2D finite element formulation and a Solid Isotropic Material with Penalization (SIMP) interpolation for material properties, topology optimization is used to determine the distribution of materials in these cells that optimizes the thermal performance of the interface. The Method of Moving Asymptotes (MMA) was used as the optimization scheme for the design variables.

Building on an existing one-material mechanical model, an algorithm was developed to consider the thermal environment of the structure. Thermo-mechanical couplings have been introduced in the model to account for the thermal expansion of materials. Simulations showed the ability of the
algorithm to reproduce results available in the literature, and served as a base model for further development.

Important aspects of this research comprise the introduction of a highly conductive layer simulating a face-sheet, which enhances the heat propagation into the structure and solves convergence problems. The effects of material thermal expansion is also analyzed for a two-material structure.

The technique was extended to a 2-material approach, and a thermal contact resistance model was implemented. 3-phase topology optimization using the SIMP method usually consists of two distinct materials and a phase representing void, where the volume fraction of each phase is constrained over the design domain. However, novel topologies can be obtained when the volume fraction of each material is not constrained, leaving the optimization process the choice of which material to be placed in a non-void element without restrictions.

Finally, a model for thermal contact is proposed in the topology optimization. This model is based on a unilateral contact approach and uses a Newton’s method to solve the governing equations. This thermal contact model constitutes an innovative concept in topology optimization.

Results for the cellular compliant mechanisms showed the ability of the algorithm to generate patterns and features that could be used in a future final design. A variety of optimal designs can be generated depending on the definition of the objective function and the boundary conditions.

Further developments could focus on the algorithms and the design of the compliant mechanisms. The thermal contact model presented here is an early stage concept and would require continued research to reach its full potential. Other future work could include post-processing of the results, tests of other objective functions and verification of the algorithm ability to generate realistic optimal designs for the cellular compliant mechanisms.
Contents

List of Figures ix
List of Tables xvi
List of Symbols xvii
Acknowledgments xix

Chapter 1: Introduction 1

1.1 Motivations & Objectives ................................................. 1
  1.1.1 Thermal Management in Space .................................. 1
  1.1.2 A Contact-aided Compliant Mechanism Concept for Thermal Control 2
  1.1.3 Modeling the Contact Interface ................................. 5
  1.1.4 Introduction to Topology Optimization ........................ 7
  1.1.5 Objectives .............................................................. 8

1.2 Overview of Topology Optimization Techniques & Applications .... 9
  1.2.1 Topology Optimization Approaches ............................. 9
  1.2.2 Compliant Mechanisms Design and Other Applications using Topology Optimization ............................................. 12
  1.2.3 Approach of the Present Work .................................. 14

Chapter 2: Single Physics Topology Optimization Problems 16

2.1 Topology Optimization for the Design of Structures ............... 16
  2.1.1 The SIMP Interpolation ........................................... 16
  2.1.2 Mechanical Governing Equations ............................... 18
  2.1.3 Sensitivity Analysis ................................................ 20
5.2 Contact for a Single Mechanical System ........................................ 73
  5.2.1 System of Equations ......................................................... 73
  5.2.2 Numerical Approach ....................................................... 75
  5.2.3 Tests and Validation ....................................................... 81

5.3 Implementation of a Thermal Contact Model ................................. 90
  5.3.1 Modeling the Contact Pressure .......................................... 90
  5.3.2 Thermal Contact Resistance (TCR) Model ............................. 91
  5.3.3 Thermal Contact Implementation for a Thermo-mechanical System ... 93

5.4 Tests and Validation ........................................................... 94

Chapter 6: Design Optimization of Cellular Contact-Aided Compliant Mechanisms 99

  6.1 Design Considerations ....................................................... 99
  6.2 How to Choose a “Good” Objective Function? .............................. 100
  6.3 Loading & Boundary Conditions ........................................... 101
  6.4 Hot Case ........................................................................ 103
    6.4.1 Results using a Two-Material Thermal Model ....................... 103
    6.4.2 Results using a Two-material Thermo-mechanical Model with Contact ... 105
  6.5 Cold Case .................................................................. 107

Chapter 7: Conclusions & Future Work ........................................ 109

  7.1 Summary of Key Results .................................................... 109
    7.1.1 Development of a Thermo-mechanical Topology Optimization Algorithm ... 109
    7.1.2 Analysis of the Behavior of a Two-Material Model .................. 110
    7.1.3 Development and Implementation of a Contact Model ............. 110
    7.1.4 On the Design of the Cellular Contact-Aided Compliant Mechanisms for Thermal Control ............. 111
  7.2 Limitations ..................................................................... 112
    7.2.1 Assumptions and Approximations .................................... 112
    7.2.2 Dimensionality ............................................................. 113
    7.2.3 Computational Efficiency ............................................... 113
    7.2.4 Optimization Method .................................................... 114
List of Figures

1.1 Typical heat conduction path from the electronics module to the thermal bus until dissipation. [3] .................................................. 3

1.2 Representation of cellular compliant mechanisms arranged into honeycombs. (a) Non-contact aided cellular compliant mechanism. (b) Contact-aided cellular compliant mechanism. A unit cell is circled in red. .................................................. 4

1.3 Thermal switch concept using a C3M for thermal control. Hot and cold cases. [3] . . 4

1.4 Conductive heat path through two conforming surfaces in contact [5]. ............... 5

1.5 Temperature behavior across the contact interface [3]. ................................. 6

1.6 Optimization of the global stiffness over the domain and resulting optimal design. For given boundary conditions (top), the expected final result (bottom) consists of an optimal distribution of material (black) and void (white). .................... 8

1.7 Hand-held devices are now able to perform 2D one-material topology optimization using the SIMP method [10]. .................................................. 10

1.8 Multi-material optimization using a phase field approach at different time steps. Material 1 in red, material 2 in blue, void in green [15]. ......................... 12

1.9 Example of a compliance minimization problem with passive areas using the 88 line code [16]. .................................................. 13

1.10 Thermal compliance optimization of a uniformly heated plate for the boundary conditions given in (a). Material is in black and void in white. [24] ..................... 14

2.1 Discretization of the domain. Square elements are considered. ...................... 17

2.2 Rectangular linear element and dimensions ........................................... 19

2.3 Example of a checkerboard problem when the sensitivities of the objective function are not filtered (same case problem as Figure 2.1). ......................... 23
2.4 Compliance-optimal non-deformed design obtained under the given boundary conditions using non-designed elements. The "disc" made of void elements is not affected by the update of the design variables. ......................................................... 24
2.5 Flowchart of the algorithm. .......................................................................................................................... 24
2.6 Undeformed compliance-optimal designs for the boundary conditions given in (a). .................................. 26
2.7 Effects of the mesh resolution on the undeformed optimal designs for the boundary conditions given in Figure 2.6(a). (a) 80x40 mesh; (b) 120x60 mesh; (c) 100x30 mesh. 26
2.8 Convergence behavior of the algorithm. (a) Convergence of the objective function. (b) Zoom on the convergence of the objective function. (c) Maximum change in density within an element over the entire domain. ......................................................... 28
2.9 Sample results for a thermal compliance optimization problem. (d), (e) and (f) are material distributions for a uniformly heated domain and for the boundary conditions (a),(b) and (c) respectively. (g), (h) and (i): Temperature profile (no-dimensions) at steady state for the distribution (d), (e) and (f) respectively. ............................................ 31
2.10 Representation of a heat flux boundary along an element. ................................................................. 32
2.11 Example of thermal compliance topology optimization using heat flux boundary conditions \( q = 1 \). (a) Boundary conditions used in the Example. (b) Optimal Topology (black=material). (c) Temperature contours over the domain. ......................................................... 33
3.1 Mechanical Boundary conditions used for comparison with the work of Rodrigues and Fernandes [28] .......................................................... 39
3.2 Effect of increasing the temperature \( T \) over the domain on the optimal topology and the objective function value \( c_0 \). In case (a), \( T = T_{ref} = 0 \) is equivalent to dealing with a mechanical system alone, with on thermal strain contribution. ................. 39
3.3 Study case used in Section 3.4.1. (a) Mechanical boundary conditions. (b) Thermal boundary conditions. ................................................................................................................. 41
3.4 Effects of heat flux boundary conditions on the special case of Figure 3.3. (a) Undeformed optimal design. (b) Temperature distribution in the undeformed design domain. (c) Deformed optimal design. (d) Temperature distribution in the optimal structure ................................................................................................................. 42
3.5 Details on the convergence of the objective function (compliance). (a) Monotonic convergence of the objective function. (b) Maximum density change in the domain.

3.6 (a) New sets of mechanical boundary conditions tested, with three nodal forces pulling downwards. (b) Optimal design at reference temperature.

3.7 Evolution of the topology at various iteration steps.

3.8 Behavior of the objective function and density changes in the case of three forces pulling down. (a) Compliance value at each iterations. (b) Maximum density change in the domain between two iterations.

3.9 Schematic of the boundary conditions with the introduction of a high conductive layer.

3.10 Mechanical compliance-optimal undeformed designs [(a) & (b)] for various values of the heat flux and the corresponding temperature contours of the domain [respectively (c) & (d)] and in the deformed structure [respectively (e) & (f)]. The high conductive layer is non-designed.

3.11 Introduction of a solid conductive layer on the mechanical compliance minimization problem and for the boundary conditions in Figure 3.6. (a) Undeformed optimal topology. (b) Temperature distribution in the domain. (c) Temperature distribution within the deformed structure.

3.12 Compliance-optimum designs [(a) & (b)] for different values of the heat flux, their corresponding (non-dimensional) temperature contours of the domain [respectively (c) & (d)] and in the deformed structure [respectively (e) & (f)].

3.13 Evolution of the objective function (a) and the density changes between each iterations (b) for $q = 1, T_{sink} = 0$.

3.14 Evolution of the objective function (a) and the density changes between each iterations (b) for $q = 2, T_{sink} = 0$.

3.15 Undeformed design domain after: (a) 15 iterations, (b) 35 iterations, (c) 225 iterations ($q = 1, T_{sink} = 0$).
4.1 Topology optimization of a 2-material mechanical model. (a) Boundary conditions. (b) Undeformed optimal topology for $E_1 = 2$ and $E_2 = 1$. (c) Undeformed optimal topology for $E_1 = 4$ and $E_2 = 1$. (d) Undeformed optimal topology for $E_1 = 2$ and $E_2 = 1$, with the volume fraction of material 1 and 2 unconstrained. Colors: white is associated with void elements, red with material 1 and blue with material 2.

4.2 Algorithm behavior for the example given in Figure 4.1(b), where $E_1 = 2$ and $E_2 = 1$. (a) Objective function values at each iterations (b) Zoom around the final value of the objective function.

4.3 Density changes at each iteration. Blue: changes in $\rho_0$. Red: changes in $\rho_0 \times \rho_1$.

4.4 Sample results for a thermal compliance optimization problem using 2 materials. (d), (e) and (f) are material distributions for a uniformly heated domain and for the boundary conditions (a), (b) and (c) respectively. (g), (h) and (i): Temperature profile at steady state for the distribution (d), (e) and (f) respectively (temperatures are non-dimensional).

4.5 Boundary conditions used in the tests for the thermo-mechanical two-material problem. (a) Mechanical boundary conditions. (b) Thermal boundary conditions.

4.6 Undeformed optimal designs for various heat flux input at the top boundary. Colors: white areas are associated with void elements, red with material 1 and blue with material 2.

4.7 Undeformed optimal topology for the minimization of the mechanical compliance. The modification of the CTEs can affect the placement of material 1 and 2. Colors: white areas are associated with void elements, red with material 1 and blue with material 2.

4.8 Undeformed optimal topology for the minimization of the mechanical compliance. The modification of the conductivities can affect the placement of material 1 and 2. Colors: white areas are associated with void elements, red with material 1 and blue with material 2.
4.9 Comparison of undeformed topologies of volume-constrained optimization and unconstrained optimization. In (a) & (c), the Volume of material 1 and 2 constrained to 20% each. In (b) & (d), the amount of material 1 and 2 in the domain not prescribed. Colors: white areas are associated with void elements, red with material 1 and blue with material 2 ................................. 69

4.10 Unconstrained optimization with $k_1 = 1$, $k_2 = 5$ and $\alpha_1 = 10 \times 10^{-4}$, $\alpha_2 = 2 \times 10^{-4}$. Undeformed optimal designed where materials’ volume fraction is unconstrained. (a) $q = 1$ (b) $q = 1.5$ Colors: white areas are associated with void elements, red with material 1 and blue with material 2 ................................. 69

5.1 Implications of the assumptions on the contact model. (a) Actual boundary of the body and the support. (b) Approximation used in the implemented model. .... 72

5.2 Undeformed linear elastic body facing a rigid support. The initial distance from point $A$ to the rigid support is denoted as $g_A$. ......................... 74

5.3 Flowchart of the algorithm ................................................ 79

5.4 The boundary conditions and loadings on the design domain can provide an indication for a good starting guess for the field variables. (a) Case 1: No contact is expected in this situation. (b) Case 2: Contact is expected ...................... 80

5.5 Topology optimization of a two-material mechanical model with contact at the bottom right corner. The parameters and results are the same as Figure 4.1. (a) Boundary conditions. (b) Undeformed optimal topology for $E_1 = 2$ and $E_2 = 1$. Colors: white is associated with void elements, red with material 1 and blue with material 2 ................................. 82

5.6 Behavior of the convergence in the minimization of compliance problem of Figure 5.5. (a) Objective function value at each iteration. (b) Density changes in the domain. Red: Contact model. Blue: Non contact model ...................... 85

5.7 Influence of the distance between the supports and the initial "free" domain. For two different boundary conditions (a) and (b), the undeformed optimal topologies are identical (c), though the deformed shapes and displacement values are different (d) and (c). Displacement contours are given for non-dimensional displacement values. 86
5.8 Influence of the distance between the supports. For 2 different boundary conditions (a) and (b), the undeformed optimal topologies (respectively (c) and (d)) and deformed optimal topologies (respectively (e) and (f)) present clear dissimilarities. Their vertical displacement contours (non-dimensional) over the domain (respectively (g) and (h)) affect the design optima. 87

5.9 Values of the normal contact forces at each node at the contact boundary. (a) Normal contact force distribution at steady state for the optimal design in Figure 5.8(c). (b) Normal contact force distribution at steady state for the optimal design in Figure 5.8(d) 88

5.10 Optimum design when no contact is made for the boundary conditions in (a). (b) Undeformed optimum. (c) Deformed optimum. (d) Profile of the vertical displacement in the structure (non-dimensional). 89

5.11 Piecewise uniform pressure distribution at the interface in the FEM. 90

5.12 Piecewise uniform distribution of the heat flux at the interface in the FEM. 91

5.13 Boundary conditions used for the test. (a) Mechanical boundary conditions. (b) Thermal boundary conditions the middle area at the bottom corresponds to an insulated face. The high conductive layer introduced in Section 3.4.2 is solid non-designed. 95

5.14 Undeformed optima for a mechanical compliance minimization problem using thermal contact. (a) Optimal design for \( h_{c0} = 1 \). (b) Optimal design for \( h_{c0} = 1 \times 10^{-2} \). 96

5.15 Forces and temperature at the nodes at the bottom boundary for the design in Figure 5.14(a). (a) Normal contact forces. (b) Temperatures 97

5.16 Forces and temperature at the nodes at the bottom boundary for the design in Figure 5.14(b). (a) Normal contact forces. (b) Temperatures 98

6.1 Boundary conditions considered for the C3M design. (a) Mechanical boundary conditions. (b) Thermal boundary conditions. 102
6.2 Minimum thermal compliance topologies using a two-material thermal model depending on the change in the thermal contact resistance (TCR) at the bottom boundary and the corresponding temperature distribution within the solid structure. Colors: white is associated with void elements, red with material 1 (high conductivity) and blue with material 2 (low conductivity).

6.3 Minimum thermal compliance topologies using a two-material thermal model depending on the change in the thermal contact resistance (TCR) at the bottom boundary. The volume fractions of the materials are unconstrained.

6.4 Undeformed optimal designs for a thermal compliance optimization problem using different values for the TCR.

7.1 Proposed design for the C3Ms. Boundary conditions on the structure (a) and heat conduction paths in both “hot” case (b) and “cold” case (c). [3]

7.2 LM-RIF iterative process. [47]
List of Tables

2.1 Standard values for the parameters in the simulations for the structural compliance minimization problem. ................................................... 25
2.2 Equivalence between thermal and mechanical finite element formulations ........ 30
2.3 Standard values of the parameters in the simulations for the structural compliance minimization problem. ................................................... 30
3.1 Standard values for the parameters in the simulations for the structural compliance minimization problem for a thermo-mechanical model. ......................... 40
4.1 Quantities appearing in the SIMP interpolation of a 2-material model ............... 56
4.2 Standard values for the parameters in the simulations for the structural compliance minimization problem for a 2-material approach. ............................ 59
4.3 Standard values for the parameters in the simulations for the thermal compliance minimization problem for a 2-material approach. ......................... 63
4.4 Standard values for the parameters in the simulations for the mechanical compliance minimization problem for a two-material thermo-mechanical approach. ........ 65
5.1 Standard values for the parameters in the simulations for the structural compliance minimization problem for a two-material approach using a mechanical contact model. 81
6.1 Standard values for the parameters in the simulations for the C3M designs. ........ 104
6.2 Standard values for the parameters in the simulations for a C3M design using a thermo-mechanical model with contact. ............................... 107
List of Symbols

$\alpha$ Coefficient of thermal expansion (m.m$^{-1}$.K$^{-1}$)

$a$ Length of a side of a square element (m)

$B$ Shape function derivatives for the mechanical finite element model

$B_{th}$ Shape function derivatives for the thermal finite element model

$c$ Constraint function

$c_0$ Objective function

$d$ Vector of nodal displacements (m)

$D$ Stress-strain matrix for plane stress formulation (N.m$^{-1}$)

$E$ Modulus of elasticity / Young’s modulus (N.m$^{-1}$)

$e$ or $e$ Element index

$f, f_0, f_1$ Volume fractions

$F_{th}$ Thermal load vector (W.m$^{-1}$.K$^{-1}$)

$F_p$ Vector of mechanical nodal forces (N)

$F_{\epsilon}$ Thermo-mechanical loadings (N)

$F_m$ Mechanical load vector (N)

$\epsilon$ Mechanical strain vector (m.m$^{-1}$)

$\epsilon_0$ Thermal strain vector (m.m$^{-1}$)

$\eta$ Contact gap (m)

$g$ Vector of contact location (m)

$\hat{H}$ Convolution operator

$H_a$ materials’ hardness (W.m$^{-2}$.K$^{-1}$)

$h_c$ Thermal contact conductance (W.m$^{-2}$.K$^{-1}$)
$h_c$  Nominal thermal contact conductance (W.m$^{-2}$.K$^{-1}$)

$J$  Jacobian

$k$  Conductivity (W.m$^{-1}$.K$^{-1}$)

$K_m$  Stiffness matrix (N.m$^{-1}$)

$K_{th}$  Conductivity matrix (W.m$^{-1}$.K$^{-1}$)

$\lambda, \lambda_m, \lambda_{th}$  Adjoint vectors

$L$  Lagrangian

$\nu$  Poisson’s ratio

$N$  Total number of element in the domain

$n_c$  Number of nodes in possible contact

$N_x$  Number of elements in the horizontal direction

$N_y$  Number of elements in the vertical direction

$\phi$  Internal strain energy (J)

$\Pi$  Potential energy (J)

$p^*$  Pressure (Pa)

$P_n$  Vector of normal contact force (N)

$q$  Heat flux (W.m$^{-2}$)

$\rho, \tilde{\rho}$  Vector of design variables

$r_{min}$  Filter’s radius

$t$  Thickness (m)

$T$  Temperature (K)

$T_{ref}$  Reference temperature (K)

$T_{sink}$  Sink temperature (K)

$TCR$  Thermal contact (m$^2$.K.W$^{-1}$)
Acknowledgments

I would like to thank Dr. George Lesieutre for his continual support and guidance throughout the project and since I arrived in the US. He was an invaluable source of inspiration to me, and his help goes beyond this work. Also, I would like to thank Dr. Mary Frecker for her guidance in the topology optimization developments. Without their insight, this work would still be in the early stages.

I thank the entire research group, and in particular Dr. James Adair and Brian Babcox, for their input on thermal conduction experiments, thermal contact considerations and the fabrication process.

I would like to acknowledge Dr. Krister Svanberg, Professor at Optimization and Systems Theory, KTH - Royal Institute of Technology, who provided the Matlab code for the Method of Moving Asymptotes used in this work.

This work is supported by the U.S. Air Force Office of Scientific Research under grant FA9550-11-1-0232, Dr. Byung-Lip Lee, Program Manager.

My friends and roommates, Kyle Williams and Rob Mukahlera, certainly deserve recognition for their support in all I do and their advice when needed. Also I want to thank my officemates for making this experience instructive and enjoyable. I do want to thank all my friends for giving me the necessary motivation to complete this work.

Finally, I would like thank my family, and in particular my parents, for encouraging me throughout, despite the distance.
Chapter 1

Introduction

1.1 Motivations & Objectives

1.1.1 Thermal Management in Space

Thermal management is a fundamental concern for spacecraft. The complexity of the missions have lead thermal control subsystems to be a central concern for engineering designs. Successful designs rely on safe systems and components, which are usually the result of extensive tests and experiments on ground environment applications. Consequently, assured reliability of these components requires maintenance of an operating range of temperatures not too far from room temperature. The temperatures of electronic “black boxes” associated with systems and payloads must be maintained within an acceptable operating range to accommodate thermal loading variations. Control of these variations depends essentially on the three major means of heat transfer [1] [2]:

1. **Radiation** is defined as “the energy transfer via electromagnetic waves” [1]. In space, the main radiative sources are the solar radiations, both direct and reflected. Also a planet may reflect or scatter the solar radiations, but it can also emit thermal radiation by itself. Passive control of those radiative inputs lead some parts of a spacecraft to be covered or painted in black to absorb this energy. Others have reflective coating, such as polished gold or silver surfaces, to help protect the spacecraft from excessive heating.

2. **Convection** corresponds to “the energy transfer between a flowing fluid and a solid interface” [1]. The main contribution of convective heat transfer happens during the launch of a satellite
through aerodynamic heating in the lower levels of the terrestrial atmosphere. Otherwise, convective heat transfer is usually not a primary concern once in space.

3. **Conduction** is “the energy transferred through matter in the absence of fluid motion”[1]. The conductive heat paths can be easily determined and examined from ground experiments. For this reason, many passive and active thermal control components usually affect the neighboring parts they are in contact with through conduction heat transfer, such as electric heaters, thermostats, thermal fillers, thermal braids and straps, etc.

Although radiative considerations are of importance when designing a spacecraft, much ongoing research on active and passive thermal control devices utilize thermal conduction. Classical solutions use strategies associated with electric heaters to adjust the temperature of these electronic “black boxes”. Besides the additional power dedicated to activate the components, these strategies usually require important control systems and sensors to monitor the temperature variations and trigger a response. In that regard, passive devices are attractive thermal solutions because of their adaptability to the environment, their reliability and their predictability. Passive devices may progressively replace other powered devices into thermal control subsystems.

### 1.1.2 A Contact-aided Compliant Mechanism Concept for Thermal Control

This research focuses on the design of an interface that controls the conductive heat flux between the electronic “black boxes” and the thermal bus or exterior of the spacecraft (see Figure 1.1). The design of a thermal baseplate at the interface is based on cellular compliant mechanisms, which deform under changing thermal conditions. Cellular compliant mechanisms are flexible structures which are able to generate a displacement under the input of a force or energy. They are arranged in arrays of individual cellular structures, such as the hexagonal cell geometry in Figure 1.2(a). The advantage of using compliant mechanisms over traditional stiff structures is their ability to repeat motion when subject to mechanical loadings, and they can also prove easy to manufacture and weigh less [4]. Cellular compliant mechanisms that present the faculty to generate internal contact, as depicted in Figure 1.2(b), are referred to as contact-aided cellular compliant mechanisms (C3Ms). Passed beyond a specific deformation threshold, the internal parts of the structure, represented as a T shape in Figure 1.2(b), come in contact with the boundary of the C3M.
In this thesis, the deformation of the C3Ms is utilized to alternatively enhance or reduce conductive heat transfer between two surfaces, based on the fact that materials can expand or retract under temperature changes. In the context of C3Ms, an increase in temperature could generate deformations within the structure and create internal contacts, thereby creating new heat conduction path. As a result, on the scale of a C3Ms array, the net heat flux through the array would be increased. Similarly, a decrease in temperature could cause the internal contact in the C3Ms to break, and the net influx through the interface to decrease.

The basic structural concept is a sandwich panel having conductive face sheets and a variable thermal-conductivity core [3]. Thermal control is achieved by using C3M that are made of two materials with different thermo-mechanical properties: one material has a high positive coefficient of thermal expansion (CTE) and a high conductivity, while the other has a low or slightly negative CTE and a low conductivity. In a case where the electronic boxes are excessively cold, the thermal switch reaches a "low" conductive mode: the material with high CTE and high conductivity contracts more than the other material, leaving a gap with the cold thermal bus. Therefore, the conductive heat losses due to the contact with the thermal bus are reduced, and the electronics heat up. On the other hand, in case of hot conditions in the boxes, the high-CTE and high-conductive
Figure 1.2: Representation of cellular compliant mechanisms arranged into honeycombs. (a) Non-contact aided cellular compliant mechanism. (b) Contact-aided cellular compliant mechanism. A unit cell is circled in red.

Figure 1.3: Thermal switch concept using a C3M for thermal control. Hot and cold cases. [3]

material establishes contact with the thermal bus, creating a new conduction path for the heat to flow out from the electronics to the spacecraft. Figure 1.3 summarizes the different configurations - high and low conductive modes - of the C3Ms.
1.1.3 Modeling the Contact Interface

Initial research by R. Stavely [3] highlighted some key points to the efficient use of the C3Ms for thermal control. The study of an idealized two-material parallel-path cell has demonstrated the basic concept on how to use the thermal expansion mismatch of two materials to realize contact. Thermal conduction occurs only in regions where there is mechanical contact. However, it appears that even at high normal pressure levels, less than 10% of the apparent area between two conforming surfaces realizes actual contact, as illustrated in Figure 1.1.

Therefore, the temperatures across the contact interface are substantially different. In Figure 1.5, this jump in temperature ($\Delta T_{\text{interface}}$) is determined by the amount of heat conducted through the contact. To characterize this phenomenon, the thermal contact resistance (TCR) is defined as:

$$ TCR = \frac{\Delta T_{\text{interface}}}{q} \quad (1.1) $$

where $q$ is the heat flux through the contact area and $\Delta T_{\text{interface}}$ is the temperature difference between the two surfaces in contact. The TCR depends on several factors. The main factor is
the contact pressure. As the contact pressure grows, the resistance to the heat flux decreases. Other factors include the finish of the surface or surface roughness, and concerns the asperities at the interface and their distribution that determine the actual contact area. Finally, the material’s hardness and the harmonic mean conductivity\(^1\) of the materials in contact can affect the heat transfer.

Stavely [3] also suggested that the displacements due to thermal expansion are expected to be small over the temperature range of interest, namely 30 degrees around room temperature (293K ± 30). The mechanisms would be closer to a thermal “valve” concept, where initial contact with zero or minimal pressure is made in the low conductivity mode. An increase in temperature would enhance the materials’ thermal expansions. Therefore, gradual increase in the contact would result in the achievement of a higher conductivity mode.

\(^1\)The harmonic mean conductivity at the contact interface usually consists of a average of the conductivities of the two material in contact such as \(k^* = k_1k_2/(k_1 + k_2)\)
The central investigation of this thesis concerns the design of the C3Ms. The structure should ideally maximize the difference between the maximum and minimum heat transfer achievable at high and low conductivity modes. The effective conductivity of the C3M could also be a quantity to optimize. Topology optimization guides the placement of material to generate topologies that maximize performance.

1.1.4 Introduction to Topology Optimization

Given a design domain, a design question is where to place the materials in order to achieve the best structural or thermal performances? Topology optimization is a mathematical tool that places material over a domain to generate an optimal design. Given a set of loads and boundary conditions on the design domain, the algorithm is expected to converge towards a design composed of distinct phases, usually void and material(s). The proportion of the material(s) to be used in the overall domain is usually limited to be less than 100% of the volume of the domain. In Figure 1.6, the boundary conditions consists of rollers on the left side of the design domain, constraining the horizontal displacement, while the vertical displacement is only constrained at the bottom left corner. A force $F$ is exerted at the top left hand corner, pushing downwards. In this example, the design objective is the maximization of the global stiffness of the structure. The optimization process converges towards a black and white topology that represents the optimal distribution of material and void (respectively).

The general optimization problem can be stated as follow [6]: given a volume constraint $c_1$ and a set of $m$ other constraints $c_i, i \in [1, ..., m]$, find the distribution of material that minimizes the objective function $c_0$ over the domain $\Omega$:

$$\min_{\rho} \quad c_0(u(\rho), \rho)$$

subject to

$$c_1(\rho) = \int\int_{\Omega} \rho(x)dV - V_0 \leq 0$$

$$c_i(\rho(\rho), \rho) \leq 0, \quad i = 1, \ldots, m$$

(1.2)
where the design variable $\rho$ takes a value 0 (void) or 1 (material) at the location $x$ in the design space. The field vector $\mathbf{u}$ must satisfy the governing equations of the physical system. This technique is commonly used for the design of compliant cell mechanisms [7].

1.1.5 Objectives

The goal of this thesis is to develop, implement and validate a topology optimization method that generates C3M topologies, which maximize the thermal control performances at the interface between the electronic black “boxes” and the spacecraft surface. Based on a simple one-material
mechanical model, a summary of the challenges tackled in this thesis are:

- Implement and validate a thermal topology optimization model.
- Develop, implement and validate a thermo-mechanical topology optimization model including a formulation for the thermal expansion of materials. The temperature distribution in the domain is not necessarily uniform. Heat flux boundary conditions can be considered.
- Modify the existing programs to study the behavior of a 2-material approach, test and analyzed the implications on adjustments of the parameters.
- Develop, implement and validate a thermal contact model within the topology optimization process.
- Generate possible optimal designs for the cellular contact-aided compliant mechanisms (C3Ms) for the thermal control.

If further tests or improvements would be needed to complete some of these tasks, important work on the topology optimization method has been performed.

1.2 Overview of Topology Optimization Techniques & Applications

1.2.1 Topology Optimization Approaches

Early developments in topology optimization trace back to the 1980’s, which makes this approach a fairly new design tool. Originally used for truss design optimization, the concept of topology optimization has extended into different procedures, all based on the general problem of Equation (1.2). Discretized topology optimization methods with continuous design variables updated iteratively have become quite popular. The fundamentals of these techniques are discussed by Sigmund and Maute [6] and briefly summarized here:

- **Density approaches** consist of discretizing the domain into finite elements, and the material properties within each element are individually defined by the design variables. A popular method used in the design of cellular compliant mechanisms is the SIMP (Solid Isotropic
Material with Penalization) [8] [9]. It consists of interpolating the material properties given a power-law approach such that, for a one-material design [7]:

$$E(\rho) = \rho^p E_0$$

(1.3)

where $E_0$ denotes the modulus of elasticity of the material and $p$ designates a penalization factor. When the density variable $\rho = 0$, the phase is assimilated as void, while $\rho = 1$ indicates the presence of material. Meanwhile, during the optimization process, $\rho$ can take intermediate continuous values before convergence, in particular at the beginning of the optimization process. Therefore, intermediate densities are penalized with $p \geq 1$ to ensure physical reliability. The value $p = 3$ is usually considered as a good value for convergence of 0-1 solutions. Variations of the SIMP method have been formulated, but conserve the general concept of material property interpolation. This interpolating scheme is relatively simple to implement and quite stable [10], and can now be implemented on hand-held devices for basic mechanical problems (Figure 1.7).

- **Topological derivatives**, also named the “bubble method” by Eschenauer et al. [11], con-
sists of introducing holes at a location \( x \) in the design domain \( \Omega \). These holes are taken as a starting point for the generation of new holes. Topological derivatives determine where to place new holes and either modify the shape of the boundaries of existing holes [11], or update the presence of holes element-wise [12].

- In most **Level set approaches**, the shape of the structure can evolve only at its boundaries. Other methods allow new holes to merge together. A level set function \( \phi(x) \) is defined as [6]:

\[
\rho = \begin{cases} 
0 : \forall x \in \Omega : \phi < 0 \\
1 : \forall x \in \Omega : \phi \geq 0 
\end{cases}
\]  

(1.4)

Therefore, the level set function is updated rather than the design variable \( \rho \). These methods are used in combination with topological derivatives [13].

- **Phase field approaches** are closer to the SIMP methods in the sense that they directly act on the density variables \( \rho_i \). This method penalizes intermediate densities by minimizing a functional. Figure 1.8 illustrates a multi-material design obtained at different stage of the optimization process.

Although the mathematical updating schemes are quite different, similar results are found using all these methods on most problems.

Topology optimization problems encountered often deal with two phases, namely void and material. Extension to more than two phases requires more complicated computation, and in general, no interpolation rules are available. Based on conventional one-material optimization methods, Tavakoli and Mohseni [14] developed a 2D active-phase algorithm for multi-material structural and thermal problems. Using the SIMP method, the optimization of the distribution for one pair of “active” phases is performed while other phases are fixed. This operation is repeated until every pairs has been considered in the algorithm. Zhou [15] proposed a multi-material method using a phase field approach, based on a generalized Cahn-Hilliard model (Figure 1.8).
1.2.2 Compliant Mechanisms Design and Other Applications using Topology Optimization

Topology optimization often generates non-intuitive structural designs. Minimizing the compliance in structural problems has been largely explored, in particular for one-material problems. The 99 line Matlab code developed by Sigmund [9], and improved later into an 88 line code [16], has become a classical problem for optimizing the global stiffness of a one material 2D structure. These codes use a SIMP interpolation scheme and filtering techniques to reduce mesh-dependency. Small modification of the code introduce passive zones, where the amount of material is prescribed all through the optimization process as shown in Figure 1.9, and multi-load case problems. This code is popular for its simplicity and efficiency in solving compliance minimization problems, where the objective is to generate the structure with the maximum global stiffness in response to mechanical loadings. Nonetheless, the study of other objective functions might be relevant, and single material topology optimization has been successfully implemented in various physical disciplines such as aerodynamics [17], electrostatics [18] or fluid dynamics [19].

Topology optimization stands as a useful tool to generate optimum designs of compliant mechanisms. Metha et al. [8] proposed a topology optimization method for the design of compliant mechanisms in high-strain configurations. Some of the cell structures obtained were extended to realize cellular contact-aided mechanisms. Linear response of the structures are usually assumed, though Bruns and Tortorelli [20] explored the behavior of non-linear elastic responses of compliant mechanisms that exhibit large displacements. Stanford and Beran [21] studied the topology optimization design of compliant mechanisms for flapping wings applications, optimizing the thrust-generating mechanism, specifically in forward flight configurations. The effect of thermal
expansion of the materials was explored by Montealegre *et al.* [22], who developed a method to optimize an output displacement while minimizing thermal effects. The thermal compensations presented in this paper showed that the resulting designs preserved the functionality of the original non-thermally compensated mechanisms, while ensuring a higher reliability when the structure is subject to thermal gradients.

Topology optimization research of thermal conduction problems has explored situations where the domain is heated uniformly, with isolated or fixed temperature boundary conditions. Most approaches used a Finite Element Method (FEM) to solve the governing equations, giving rise to mesh-dependency if efficient filtering techniques are not applied [7]. A few other approaches such as FVM [23] [24] and mesh-less techniques [25] have been investigated. In some cases, these can provide different results than FEM implementations as suggested in Figure 1.10.

Also, conduction is the only thermal aspect considered in these two-phase models, where the void phase is modeled as a quasi-null conductivity phase. Iga *et. al.* [26] investigated the influence of design-dependent effects on heat convection and internal heat generation in topology optimization. Most of these models attempt to optimize the thermal analog of the compliance for structural problems, meaning that the optimum design aims to find the best conductivity distribution that minimizes the average temperature over the domain. Conventionally, the amount of conductive material is limited and the domain is heated uniformly, by assigning to each element a heat generation source. Other studies investigated the effect of thermal expansion on a single-material design [27] [28]. They found that the geometry of the optimal design could be affected by a uniform

Figure 1.9: Example of a compliance minimization problem with passive areas using the 88 line code [16].
increase in temperature over the domain and that, in some cases, thermal expansion effects could improve the compliance. However, these papers did not solve the thermal equations to study a non-uniform temperature distribution.

Three-phase models require more sophisticated optimization techniques with a stronger mathematical basis than the heuristic method often used in two-phase designs. Thermo-mechanical couplings using extreme coefficients of thermal expansion (CTE) in a three-phase model have been explored by Sigmund and Torquato [29]. They demonstrated that it was possible to obtain a structure with an effective negative CTE by mixing materials with positive CTEs. This multi-physics topology optimization approach used sequential linear programming (SLP) as a marching scheme to update the density variables. Sigmund proposed to use another mathematical scheme to solve multi-physics and multi-material problems [30] [31], and used a Method of Moving Asymptotes (MMA) to drive the update of the design variables [32]. According to Sigmund [30], this method proved to be more stable and computationally efficient in generating design shapes.

1.2.3 Approach of the Present Work

The topology optimization algorithms developed in this thesis are based on an existing 1-material mechanical model by Sigmund [9]. The material densities are interpolated through a SIMP approach, and a finite element formulation is used to solve the governing equations. The design variables are updated following the Method of Moving Asymptotes (MMA) by Svanberg [32]. The
design domain consists of a rectangular domain with boundary conditions and loadings defined at the contour. A 1-material thermal model has been developed similarly. The algorithms are tested on compliance optimization problems and the results compared with results from the literature.

The development of a thermo-mechanical approach include heat flux boundary conditions, which can generate temperature gradients in a structure. The introduction of a high conductive layer allows the heat to propagate efficiently to the structure and solves convergence problems that can arise in the presence of high heat flux conditions. Temperature gradients effect on compliance optimization problems, that have not been clearly demonstrated in the literature, are analyzed. Also, the influence of the coefficient of thermal expansion of the materials are of great interest for the design of passive thermal control devices. Thermal expansion properties could potentially affect the optimal topology under high temperature conditions.

The previous algorithms are then improved into a 2-material model. The SIMP method and the optimization scheme are modified to incorporate more design variables. When volume of each material is not prescribed, novel topologies can be obtained.

The inclusion of a unilateral contact model is performed. Based on the Signorini’s contact formulation, the governing equations are solved using a Newton’s method and an Armijo line search. This method is extended to account for a thermal contact model, which depends on the pressure at the contact interface. This model is tested, and possible improvements are proposed.

Finally, topology optimization results suggests patterns and features that could be used for the final C3Ms design. A complete design should require additional work and post-treatment of the results, though important axis of development emanates from this work, concerning both the topology optimization method and the C3Ms designs.
Chapter 2

Single Physics Topology Optimization Problems

This chapter presents previous work on an existing topology optimization algorithm, and an extension to a thermal model.

2.1 Topology Optimization for the Design of Structures

The 99 line Matlab Code by Sigmund [9] is used as the base for the implementation of the more complex algorithms developed further in this thesis. Going through this algorithm, we walk through the different steps of the optimization strategy adopted. The Finite element method and the Solid Isotropic Material with Penalization (SIMP) interpolation scheme for a single-material structure are explored. Details on the optimization procedure and examples on a compliance minimization problem are given.

2.1.1 The SIMP Interpolation

A rectangular 2D domain is discretized into square elements and the material properties have to be defined within each element. The SIMP method defines the material properties to be used. It consists of a power-law approach that allows the material properties to be interpolated by a smooth and continuous function of the density variables, as presented in Section 1.2.1. For $n$ elements in
Figure 2.1: Discretization of the design domain. Square elements are considered.

The domain, the vector of density variables \( \mathbf{\rho} = \{\rho_1, \rho_2, ..., \rho_n\} \) defines the density of material in each element, and thereby the value for modulus of elasticity \( E \) to be used in an element following Equation (1.3). The density values take continuous values such as, for an element \( i \), \( \rho_i \in [0, 1] \).

However, a zero-material density such that \( \rho_i = 0 \) would cause singularity problems when solving the mechanical governing equations. Therefore, the presence of void is approximated as a material with a very low modulus of elasticity. This can be done by constraining the density variables such that \( 0 < \rho_{\min} \leq \rho_i \leq 1 \), where the lower bound \( \rho_{\min} \) is a small number (say \( \rho_{\min} = 10^{-3} \)).

Figure 2.1 illustrates the discretization of the design domain on the example given in Figure 1.6. The optimal topology presented corresponds to a compliance minimization problem. In summary, three scenarios can exist in an element \( e \):

1. \( \rho_e = 1 \) : The corresponding element is composed of material, and according to Equation (1.3), \( E(\rho_e) = E_0 \).

2. \( \rho_{\min} < \rho_e < 1 \) : The element is partially filled with material, and the resulting material property is an interpolation of the properties of void and material. The optimization penalizes
those configurations, which don’t have real physical meaning.

3. $\rho_e = 0$ : The corresponding element is void and according to Equation (1.3), $E(\rho_e) = \rho_{min}^p E_0$.

An initial distribution of the material over the domain has to be defined for the governing equations to be solved. Conventionally, each element is assumed to have an initial fraction of material equal to the final amount of material to be used in the domain. For example, if the total proportion of material to be used represents 60% of the design domain, the other 40% being void, each element initially contains a mixture of the same proportion. This initial material repartition is uniform and respects the constraint on the amount of material to be used. The formulation of this constraint is discussed further in Section 2.1.3.

### 2.1.2 Mechanical Governing Equations

Based on the discretization of the domain presented in the previous section, a two-dimensional four-noded finite element formulation is used to solve the mechanical governing equations. A typical rectangular element used in the finite element analysis is presented in Figure 2.2, though square elements ($a = b$)\(^1\) have been considered in this thesis. At each node is defined a displacement $d = [d_x, d_y]^T$. The materials used in the domain are assumed to be isotropic and the structure is subjected to mechanical loadings on the boundary of the design domain. Thermal expansion of the materials is neglected in this section. The system is considered to be stationary (no time dependence). The cell strain energy within an element $e$ of thickness $t$ is defined as:

$$
\phi(\rho_e) = \frac{1}{2} d_e^T \int \int_{\Omega_e} B^T D(\rho_e) B t \, dx \, dy \, d_e
$$

where $D$ is the $3 \times 3$ matrix derived from the stress-strain relation in plane stress:

$$
D = \frac{E}{1 - \nu} \begin{bmatrix}
1 & \nu & 0 \\
\nu & 1 & 0 \\
0 & 0 & \frac{1 - \nu}{2}
\end{bmatrix}
$$

\(^1\)\(a = 1\) is taken in the simulations considered in this thesis
with \( \nu \) is the Poisson’s ratio and

\[
B = \begin{bmatrix}
    \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\
    0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\
    \frac{\partial N_1}{\partial x} & \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial x} & \frac{\partial N_4}{\partial y}
\end{bmatrix}
\] (2.3)

where \( N \) is the vector of linear shape functions such as:

\[
N(x, y) = \begin{pmatrix}
    N_1(x, y) \\
    N_2(x, y) \\
    N_3(x, y) \\
    N_4(x, y)
\end{pmatrix} = \begin{pmatrix}
    \frac{(a-x)(b-y)}{4ab} \\
    \frac{(a+x)(b-y)}{4ab} \\
    \frac{(a+x)(b+y)}{4ab} \\
    \frac{(a-x)(b+y)}{4ab}
\end{pmatrix}
\] (2.4)

where \( a \) and \( b \) are the semi-length and semi-width of an element. Hence, for square elements, the expression of \( N \) can be simplified for \( a = b \).

The potential energy within an element can be derived as

\[
\Pi_p(\rho_e) = \frac{1}{2} \int \int_{\Omega_e} \mathbf{d}_e^T \mathbf{B}^T D(\rho_e) \mathbf{B} \mathbf{d}_e \ dx\,dy - \mathbf{f}_p^T \mathbf{d}_e
\] (2.5)

with \( f_p \) the mechanical forcing terms \((N)\) at each node of the element. By derivation with respect to the field variables, the element stiffness matrix \( K_{m,e} \) is

\[
K_{m,e}(\rho_e) = \int \int_{\Omega_e} \mathbf{B}^T D(\rho_e) \mathbf{B} \ t \ dx\,dy
\] (2.6)
The quasi-static mechanical equations are given by:

$$ K_m(\rho) d = F_p $$

(2.7)

where the global stiffness matrix $K_m$ and the global forcing vector $F_p$ consists of the usual sum over the element matrices (for $N = N_x \times N_y$ elements in the domain):

$$ K_m = \sum_{e=1}^{N} K_{m,e}(\rho_e) $$

$$ F_p = \sum_{e=1}^{N} f_p(\rho_e) $$

(2.8)

Once the governing equations are solved and the displacements over the domain determined, the optimization procedure can start.

### 2.1.3 Sensitivity Analysis

Solving the governing equations determines the field variables (here the displacements) at each node. The next step consists of evaluating the objective function $c_0$ in Equation (1.2) for the current distribution of material, and calculating its derivatives. These derivatives dictate the direction of descent for the update of the density variables in each element. The calculation of these derivatives, or sensitivity analysis, is performed using the adjoint method. For a set of $l$ constraints, the optimization problem given in Equation (1.2) is equivalent to optimizing for the Lagrangian $L$:

$$ \min_{\rho} L(\rho) = c_0(\mathbf{u}(\rho), \rho) + \lambda_m^T (K_m d - F_p) $$

subject to $c_i(\mathbf{u}(\rho), \rho) \leq 0 ; i = 1, \ldots, l.$

(2.9)

where $\lambda_m$ is the vector of adjoint variables [33]. These variables will be determined soon, but are for the moment unknown. In this section, the constraints are essentially volume constraints. Those constraints ensure that the amount of one material over the whole domain is the same after each iteration of the optimization procedure, regardless of the material distribution. Each volume
constraint controls the presence of each phase as follow:

\[
\begin{align*}
\text{void: } \quad c_1 &= 1 - \frac{1}{V(1-f)} \sum_{e=1}^{N} (1 - \rho_e) V_e \leq 0 \\
\text{material: } \quad c_2 &= 1 - \frac{1}{Vf} \sum_{e=1}^{N} \rho_e V_e \leq 0
\end{align*}
\] (2.10)

where \( V_e \) denotes the element volume and \( V \) the total volume of the domain. \( f \) is defined as the volume fraction of material, and characterizes the percentage of material in the volume. Note that it is possible to define one constraint as a function of the other, since the sum of the normalized volume fractions must be equal to 1.

The derivatives of the Lagrangian for each element \( e \) are written as:

\[
\frac{\partial L}{\partial \rho_e}(\rho_e) = \frac{\partial c_0}{\partial \rho_e} + \lambda_m^T \left( \frac{\partial K_m}{\partial \rho_e} d - \frac{\partial F_p}{\partial \rho_e} \right) + \frac{\partial c_0}{\partial d} \frac{\partial d}{\partial \rho_e} + \lambda_m^T \left( K_m \frac{\partial d}{\partial \rho_e} - \frac{\partial F_p}{\partial d} \frac{\partial d}{\partial \rho_e} \right)
\] (2.11)

The adjoint method consists of eliminating the term \( \frac{\partial d}{\partial \rho_e} \) by solving the following adjoint equation, which, in the special case where \( \frac{\partial F_p}{\partial d} = 0 \), gives:

\[
\left( \frac{\partial c_0}{\partial d} + \lambda_m^T K_m \right) \frac{\partial d}{\partial \rho_e} = 0
\] (2.12)

This way, the adjoint vector \( \lambda_m \) are chosen as the solutions of the adjoint equation. Therefore, the Lagrangian derivatives are explicitly determined as:

\[
\frac{\partial L}{\partial \rho_e}(\rho_e) = \frac{\partial c_0}{\partial \rho_e} + \lambda_m^T \left( \frac{\partial K_m}{\partial \rho_e} d - \frac{\partial F_p}{\partial \rho_e} \right)
\] (2.13)

Finally, in the special case where \( \frac{\partial F_p}{\partial \rho_e} = 0 \), the expression of the Lagrangian derivatives are simplified as follow:

\[
\frac{\partial L}{\partial \rho_e}(\rho_e) = \frac{\partial c_0}{\partial \rho_e} + \lambda_m^T \left( \frac{\partial K_m}{\partial \rho_e} d \right)
\] (2.14)

The mathematical programming method could be implemented directly using the Lagrangian derivatives in (4.7). However, the material interpolation scheme and the finite element model require some special considerations before updating the design variables.
2.1.4 Filtering

The algorithm optimizes for the continuous design variables \( \rho_i \) in each element of the domain. At this point, it is possible to observe material and void elements alternatively in the same region. This problem, known as the checkerboard problem, is solved using filtering techniques. Figure 2.3 illustrates this problem, in comparison with the expected result given in Figure 2.1. The sensitivities of the objective function are modified using the same filtering technique used by Andreassen et al. [16]. The sensitivity of a particular element is replaced by a weighted average of the neighboring element sensitivities. The filtering scheme modifies the sensitivities to overcome the checkerboard problems, and also helps prevent mesh-dependent designs that could arise using a finite element formulation. For the densities \( \rho \), the filtering scheme is described as follows:

\[
\frac{\partial L}{\partial \rho_e}(\rho_e) = \frac{1}{\max(\rho_e, \rho_{\min, e})} \sum_{i=1}^{N} \hat{H}_i \rho_i \frac{\partial L}{\partial \rho_i} \tag{2.15}
\]

where \( \hat{H} \) is the mesh-independent convolution operator defined as:

\[
\hat{H}_i = \max(0, r_{\min} - \Delta(e, i)) \tag{2.16}
\]

\( \Delta(e, i) \) represents the distance between a specific element \( e \) and its neighbor \( i \), while \( r_{\min} \) is a limitation criterion on the element’s neighborhood included in the filtering scheme. When \( r_{\min} \) goes to infinity, all sensitivities are equal, while \( r_{\min} = 0 \) preserves the original sensitivity values. The adjusted sensitivities indicate the direction of descent for the optimization problem. Several updating schemes exist for the design variables as seen in Section 1.1.4. The Method of Moving Asymptotes (MMA) has been chosen because of both its performance and its suitability to the density approach, as explained in the following section.

2.1.5 Updating the Design Variables: The Method of Moving Asymptotes

Various mathematical schemes have been used to update the density variables in the optimization process. Methods such as Sequential Linear Programming (SLP) have been successfully implemented in various applications. Another popular method used in topology optimization is the Method of Moving Asymptotes (MMA) developed by Svanberg [32]. This method generates a
sequence of subproblems by linearizing the constraints and the objective function(s). The sub-
problems are solved iteratively using a dual method. The solution of these subproblems makes it
possible to solve for the new optimal density variables explicitly. According to Sigmund [30], this
method has proved to be quite stable and computationally efficient compared to traditional linear
programming methods. In this study, these arguments justified the choice made of this updating
scheme for the design variables. The Matlab code used in this thesis, developed and implemented
by Svanberg, has been slightly modified to incorporate passive elements in the design. Passive ele-
ments are elements in which the design variables are held fixed and are not affected by the updating
scheme. These elements can be non-designed elements, used to represent void areas such as holes
(Figure 2.4), or designed elements indicating a space filled with a fixed solid material.

For single physics compliance minimization problems, a simple fixed-point heuristic scheme can
be used to update the design variables instead of the MMA approach [16]. The results are essen-
tially equivalent. If simple heuristic methods present a clear advantage in terms of computational
efficiency, most problems require the use of a more complex method such as MMA.

2.1.6 Algorithm Implementation

Finally, the topology optimization algorithm flowchart follows the pattern given by Figure 2.5:

1. Solve the governing equations

2. Calculate the sensitivities and perform filtering
Figure 2.4: Compliance-optimal non-deformed design obtained under the given boundary conditions using non-designed elements. The "disc" made of void elements is not affected by the update of the design variables.

Figure 2.5: Flowchart of the algorithm.

3. Update the design variables using the Method of Moving Asymptotes

4. If some stopping criterion is not reached, go back to step 1 using the new design variables

The stopping criterion is usually related to the design variable changes in the last iteration. Here, the algorithm is said to have converged if the largest change in all design variables is less than 1% in each element over the domain.
Modulus of elasticity \( E_0 \) 1
Poisson’s ratio \( \nu \) 0.3
Volume fraction \( f \) 0.5
Filter’s radius \( r_{min} \) 2
Mesh resolution \( N_x \times N_y \) 80x40

Table 2.1: Standard values for the parameters in the simulations for the structural compliance minimization problem.

2.1.7 A Classic Example: a Compliance Optimization Problem

The objective function mentioned in the sensitivity analysis was defined as a general function of the design variables. To demonstrate the ability of the algorithm to solve thermo-mechanical problems, the objective function is chosen to be the compliance of the structure. Namely, the objective is to minimize a measure of the global stiffness over the design domain. This is a popular objective function in one-material structural optimization [7], and is therefore a good objective with which to test an algorithm performance. The objective function and its derivatives are replaced in the sensitivity analysis by the following expressions:

\[
c_0(\rho) = d^T K_m(\rho) d
\]

(2.17)

\[
\frac{\partial c_0}{\partial \rho_e}(\rho_e) = d^T \frac{\partial K_m}{\partial \rho_e} d
\]

(2.18)

The quantities in this analysis are non dimensional. Using dimensional quantities often results in ill-conditioned matrices or requires an non-reasonably high loadings to allow good convergence. In this context, topology optimization should be regarded as a mathematical tool to generate concepts for possible optimal designs rather than providing detailed designs for a specific physical problem. Following the method of typical codes used in structural optimization [16] [9], if not indicated, the standard parameters used in the following simulations are given in Table 2.1:

Optimal designs for different volume fraction of material \( f \) and different filter radius \( r_{min} \) were generated. The undeformed final structures are shown in Figure 2.6. A Matlab algorithm similar to the 88 line matlab code by Andreassen et al. [16] was used to generate these topologies.

High mesh resolutions improve not only the smoothness of the optimal design but also the
Figure 2.6: Undeformed compliance-optimal designs for the boundary conditions given in (a).

Figure 2.7: Effects of the mesh resolution on the undeformed optimal designs for the boundary conditions given in Figure 2.6(a). (a) 80x40 mesh; (b) 120x60 mesh; (c) 100x30 mesh.

accuracy of the solution to the governing equations. This can prove important when dealing with near-singular boundary conditions. Also, the ratio of length versus width of the design domain influences the final material distribution, as seen in Figure 2.7.

The adjustment of the filter radius affects the structural complexity of the results. Complex topologies obtained with a small filter radius, can present a relative improvement on the objective
function minimization but might prove harder to realize in practice. This rule is in general res-
pected for objective functions other than the compliance optimization problem. However, in the
special case of structural compliance optimization, the higher the volume fraction of material in the
design, the lower the global stiffness\(^2\) over the domain. In the simulations of Figure 2.6, the initial
guess for the material repartition is uniform and respects the volume constraints set in Equation
(4.3).

The typical behavior is that the objective function converges relatively quickly in the first few
iterations, while minor changes in the designs are observed in the later stages, as illustrated in
Figure 2.8. Figure 2.8(c) indicates the maximum change in element density over the entire domain
at each iteration. The convergence of the algorithm always meets the convergence criteria defined
in Section 2.1.6, namely, the algorithm stops when the maximum change in the element densities
is less than 1%.

However, the convergence, if achieved, highly depends on the boundary conditions and the ob-
jective function and a typical trend, especially for early stage iterations, can not be generalized
to other problems. Yet, studying the convergence behavior provides valuable information on the
optimization path leading to the final topology.

This example has served as a reference for the development of extended topology optimization
techniques. Every extension was tested with those simple structural compliance minimization
problems. Similarly, thermal compliance optimization problems are studied before implementing
fully coupled thermo-mechanical optimization approaches.

\(^2\)In compliance optimization, the objective is to minimize the global stiffness over the domain
Figure 2.8: Convergence behavior of the algorithm. (a) Convergence of the objective function. (b) Zoom on the convergence of the objective function. (c) Maximum change in density within an element over the entire domain.

2.2 Thermal Problems

2.2.1 Governing equations

Paralleling the method described in Section 2.1, topology optimization can be applied to thermal problems. Similarly to the mechanical model, the SIMP approach is adopted, such as for each element $e$:

$$k(\rho_e) = \rho_e^p k_0$$  \hspace{1cm} (2.19)
where $k_0$ is the conductivity of the material. This way, void elements are assimilated as elements with a quasi-null conductivity. The thermal governing equations are implemented with the same square linear elements used for the finite element formulation of the mechanical problem. The model assumes that the density, the specific heat and the conductivity of the material are independent of temperature. Again, the materials are considered to be isotropic. Using a Galerkin approximation applied to a variational statement for finite element formulation [34], the element thermal conductivity matrix is given by:

$$K_{th,e}(\rho_e) = \frac{1}{2} \int_{\Omega_e} T_e^T B_{th}^T k(\rho_e) B_{th} T_e \, dx \, dy$$

(2.20)

where $T_e$ is the $4 \times 1$ vector containing the temperature at each node of the element $e$, and $B_{th}$ is defined as:

$$B_{th} = \begin{bmatrix}
\frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\
\frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y}
\end{bmatrix}$$

(2.21)

and $N_i, i = 1, 2, 3, 4$ are the linear shape functions defined in Equation (2.4). Hence, the quasi-static system of thermal equations can be written as:

$$K_{th}(\rho)T = F_{th}$$

(2.22)

where the global conductivity matrix $K_{th}$ is assembled as a sum over element conductivity matrices:

$$K_{th} = \sum_{e=1}^{N} K_{th,e}(\rho_e)$$

(2.23)

$F_{th}$ is the thermal load vector ($W/m^2$), which represents both heat flux at the boundary or heat generation terms. Prescribed temperature conditions are realized by modifying the rows of $K_{th}$ and $F_{th}$ for the corresponding nodes. After the governing equations are solved, the sensitivity analysis is performed in a similar fashion as in Section 2.1 for the mechanical model by replacing the corresponding quantities as described in Table 2.2.
Mechanical quantities | Thermal quantities
---|---
Stiffness Matrix | $K_m$ | $K_{th}$
Loadings | $F_p$ | $F_{th}$
Field variables | $d$ | $T$
Material properties | $E$ | $k$

Table 2.2: Equivalence between thermal and mechanical finite element formulations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conductivity</td>
<td>$k_0$</td>
</tr>
<tr>
<td>Volume Fraction</td>
<td>$f$</td>
</tr>
<tr>
<td>Filter’s radius</td>
<td>$r_{min}$</td>
</tr>
<tr>
<td>Mesh resolution</td>
<td>$N_x \times N_y$</td>
</tr>
</tbody>
</table>

Table 2.3: Standard values of the parameters in the simulations for the structural compliance minimization problem.

### 2.2.2 Simulated Results using Prescribed Temperatures and Insulated Boundaries

Figure 2.9 shows material distributions under different thermal boundary conditions. In these examples, the domain is heated uniformly with a volumetric forcing $f_{th} = 10^{-2}$. Boundaries marked in orange indicate prescribed temperature conditions (at a temperature $T = 100K$), while black contours indicate insulated boundaries. The objective function to be minimized is the thermal compliance: the objective is to minimize the global heat over the domain. The expression for thermal compliance is given as:

$$c_0(p) = T^T K_{th}(p) T$$

(2.24)

The values of the parameters are non dimensional quantities, and standard values for the simulations are given in Table 2.3. The material tends to be placed near the heat sink where the temperature is prescribed: the heat over the domain is evacuated by conduction at these boundaries. The material is distributed such that the heat conductive paths from hot spots towards cold areas are maximized. Note that the temperature distributions in Figure 2.9 are available over the
Figure 2.9: Sample results for a thermal compliance optimization problem. (d), (e) and (f) are material distributions for a uniformly heated domain and for the boundary conditions (a), (b) and (c) respectively. (g), (h) and (i): Temperature profile (no-dimensions) at steady state for the distribution (d), (e) and (f) respectively.

Since void elements are considered as materials with an infinitesimal conductivity, temperature values are accessible at the corresponding nodes. Although temperature values in the void elements do not have any physical meaning, it is important to have access to them for optimization purposes.
2.2.3 Designs with Heat Flux Boundary Conditions

Heat flux boundary conditions are less common in the literature. Problem can arise in a void element because of their associated low conductivity value: when subject to heating high deviations in temperature can appear at the element’s node. However, this phenomenon is actually intended when optimizing for the thermal compliance.

The finite element formulation of a heat flux boundary condition for one element in Figure 2.10 is given as:

\[
fs = - \int_{-a}^{a} N(x,-b)^T tq \, dx \\mathcal{L} = - \begin{bmatrix} a \\ a \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \frac{tq}{2} \\ a \\ a \\ 0 \end{bmatrix}
\]

where \(a\) is the semi-length of the side of a square element, \(t\) is the thickness of the 2D structure, and \(N\) is the vector of shape functions given in Equation (2.4). Figure 2.11 displays the optimal result with respect to the thermal compliance optimization problem. The boundary conditions are the same as in Figure 2.9(b), except that the domain is not heated uniformly and \(T = 10\) at the top and left boundaries. The only heat input over the domain is at the bottom and the right boundaries of the domain. This example emphasizes the tendency of the material to be placed such
Figure 2.11: Example of thermal compliance topology optimization using heat flux boundary conditions ($q = 1$). (a) Boundary conditions used in the Example. (b) Optimal Topology (black=material). (c) Temperature contours over the domain.

that conductive paths between the hot and cold areas are enhanced. This result is in accordance with the expectation that the design should minimize the global heat (high temperatures) over the domain and will be important for future design considerations.

In this section, the basic features of the topology optimization strategy used in this thesis
have been introduced. In order to reproduce the physical behavior of C3Ms presented in the introduction, several extension of these models are necessary. In particular, solutions for a coupled thermo-mechanical system can be implemented by combining in a general sense both the mechanical and thermal analysis performed in this section.
Chapter 3

Thermo-mechanical Model

Under changes from a reference temperature, the materials are expected to expand or contract, which can affect the loading on the structure. The solution of the mechanical equations are coupled to those of the thermal system through the thermal expansion effects.

3.1 Thermal Expansion of Materials

In many problems, the thermal strain is modeled for a uniform temperature change over the domain. However, gradients of temperatures introduce some variability in the expansion (or contraction) of the materials from a mechanical standpoint. In particular, hot areas accentuate the effects of thermal expansion, which may be a useful phenomenon to exploit in the design of the C3Ms to achieve thermal control. The model developed here considers that the thermal environment impacts the mechanical behavior, but that the reverse relation need not be considered. As a result, the mechanical system of equations is solved after the thermal system. Because the coefficients of thermal expansion (CTEs) of each material are independent of temperature, the additional strain due to thermal expansion varies linearly with temperature. The SIMP material interpolation approach is unchanged, and equations (1.3) and (2.19) still hold. The nominal strain due to thermal expansion can be chosen to be null at a given reference temperature $T_{ref}$. The resulting thermal strain over an element reduces to:

$$\epsilon^e_0 = \epsilon^0_0(x,y) = \alpha(N(x,y)T_e - T_{ref}) \begin{pmatrix} 1 & 1 & 0 \end{pmatrix}^T$$  \hspace{1cm} (3.1)
where \( N(x,y) \) is the vector of linear shape functions for a four-noded element given in Equation (2.4) and \( \alpha \) is the CTE of the material. The expression for the internal strain energy in the cell is given as:

\[
\phi(\rho) = \frac{1}{2} \iint_{\Omega} \left( \epsilon^T - \epsilon_0^T \right) D(\rho)(\epsilon - \epsilon_0) \, dxdy
\]  

(3.2)

and thus,

\[
\phi(\rho) = \frac{1}{2} \iint_{\Omega} d^T B^T D B d \, dxdy - \iint_{\Omega} d^T B^T D \epsilon_0 \, t \, dxdy + \frac{1}{2} \iint_{\Omega} \epsilon_0^T D \epsilon_0 \, t \, dxdy
\]  

(3.3)

where \( D \) is derived from the stress-strain relations for plane stress. The potential energy can be derived as

\[
\Pi_p(\rho) = \phi(\rho) - d^T F_m
\]  

(3.4)

The mechanical loadings include the effects of thermal strain as

\[
F_m = F_p + F_\epsilon
\]  

(3.5)

where the additional loading due to thermal expansion \( F_\epsilon \) is derived as

\[
F_\epsilon = \iint_{\Omega} B^T \epsilon_0 \, t \, dxdy
\]  

(3.6)

Therefore, the quasi-static mechanical equations have the form:

\[
K_m(\rho)d = F_m
\]  

(3.7)

Note that the mechanical forcing terms depend on the temperature distribution \( (F_m = F_m(T)) \) and include the effect of the strain due to thermal expansion. The mechanical stiffness matrix \( K_m \) consists of the usual sum over element stiffness matrices:

\[
K_m = \sum_{e=1}^{N} K_{m,e} (\rho_e)
\]  

(3.8)

where \( K_{m,e} \) is given by Equation (2.6). Note that the mechanical forcing terms depend on the
temperature distribution \((F_m = F_m(T))\) and include the effect of the strain due to thermal expansion.

The thermal governing equations developed in Section 2.2 are unchanged. The system of governing equation can be summarized as follow:

\[
\begin{align*}
K_m d &= F_m(T) \\
K_{th} T &= F_{th}
\end{align*}
\]  

\[ (3.9) \]

### 3.2 Sensitivity Analysis for Thermo-mechanical Models

The adjoint method described in Section 2.1.3 is modified to take into account both thermal and mechanical equations. Thus, for a set of \(l\) constraints, the optimization problem formulation can be stated as follow:

\[
\begin{align*}
\text{minimize} \quad L(p) &= c_0(p) + \lambda_{th}^T(K_{th}T - F_{th}) + \lambda_m^T(K_m d - F_m(T)) \\
\text{subject to} \quad c_i(p) &\leq 0, \quad i = 1, \ldots, l.
\end{align*}
\]  

\[ (3.10) \]

where \(\lambda_{th}\) and \(\lambda_m\) are the vectors of adjoint variables. The constraints \(c_i\) consist of the volume constraint defined in 4.3. The sensitivity analysis of thermo-mechanical problems is more complicated than for a single physics problem, since two adjoint equations need to be solved:

\[
\frac{\partial L}{\partial \rho(e)} = \frac{\partial c_0}{\partial \rho(e)} + \lambda_{th}^T \left( \frac{\partial K_{th}T}{\partial \rho(e)} - \frac{\partial F_{th}}{\partial \rho(e)} \right) + \lambda_m^T \left( \frac{\partial K_m d}{\partial \rho(e)} - \frac{\partial F_m(T)}{\partial \rho(e)} \right)
\]

\[
+ \frac{\partial c_0}{\partial d} + \frac{\partial c_0}{\partial T} \frac{\partial T}{\partial \rho(e)} + \lambda_{th}^T \frac{K_{th}}{\partial \rho(e)} + \lambda_m^T \left( K_m \frac{\partial d}{\partial \rho(e)} - \frac{\partial F_m}{\partial T} \frac{\partial T}{\partial \rho(e)} \right)
\]  

\[ (3.11) \]

The two adjoint equations to be solved are therefore:

\[
\left( \frac{\partial c_0}{\partial T} + \lambda_{th}^T K_{th} - \lambda_m^T \frac{\partial F_m}{\partial T} \right) \frac{\partial T}{\partial \rho(e)} = 0 \quad \text{and} \quad \left( \frac{\partial c_0}{\partial d} + \lambda_m^T K_m \right) \frac{\partial d}{\partial \rho(e)} = 0
\]  

\[ (3.12) \]
This way, the adjoint vectors \((\lambda_{th}, \lambda_m)\) are chosen as the solutions of the adjoint equations. Therefore, the Lagrangian derivatives are explicitly determined as:

\[
\frac{\partial L}{\partial \rho_e} = \frac{\partial c_0}{\partial \rho_e} + \lambda_{th}^T \left( \frac{\partial K_{th}}{\partial \rho_e} T_e - \frac{\partial F_{th}}{\partial \rho_e} \right) + \lambda_m^T \left( \frac{\partial K_m}{\partial \rho_e} d_e - \frac{\partial F_m}{\partial \rho_e} \right)
\]

Finally, in the case where \(\frac{\partial F_m}{\partial \rho_e} = \frac{\partial F_{th}}{\partial \rho_e} = 0\), the expression of the Lagrangian derivatives are simplified as follow:

\[
\frac{\partial L}{\partial \rho_e} = \frac{\partial c_0}{\partial \rho_e} + \lambda_{th}^T \frac{\partial K_{th}}{\partial \rho_e} T_e + \lambda_m^T \frac{\partial K_m}{\partial \rho_e} d_e
\]

The filtering of the densities is performed following the same scheme as in Section 2.1.4 and the mathematical programming method (MMA) is unchanged.

### 3.3 Validation using a Uniform Temperature Distribution

A series of trial problems was studied to validate the performance of the algorithm. The first problem studied is a simple case where the temperature is uniform over the domain. The mechanical compliance (presented in Section 2.1.7) is minimized and the effect of elevated temperatures over the optimum is examined. This problem has been studied by Rodrigues and Fernandes [28]. The boundary conditions of the problem studied are given in Figure 3.1. Displacements in the longitudinal and transverse directions are fixed on the right and left boundaries, while a nodal force \(F\) is pulling the structure down. The standard values taken for the simulations are given in Table 3.1.

Note that the conductivity of the material and the solution of the thermal system of equations is trivial in this case. The material expands with the same behavior over the entire domain.

The effect of thermal expansion can be seen in Figure 3.2. As the value of the temperature difference with the reference temperature \(T_{ref} = 0\) increases, the shape of the optimal topology is changed. In particular, as the temperature increase, placing material on the upper right and left part of the domain comes to a disadvantage for minimizing the global strain (the objective) in the structure. Indeed, the material is placed preferably at the bottom left and right corners in hot temperature conditions, and the structure uses the thermal expansion to counteract the forcing downwards. These topology results present close similarities with the work of Rodrigues.
Figure 3.1: Mechanical Boundary conditions used for comparison with the work of Rodrigues and Fernandes [28]

\[ d_x, d_y = 0 \]

\[ F \]

Figure 3.2: Effect of increasing the temperature $T$ over the domain on the optimal topology and the objective function value $c_0$. In case (a), $T = T_{ref} = 0$ is equivalent to dealing with a mechanical system alone, with no thermal strain contribution.

(a) $T = 0, c_0 = 12.62$

(b) $T = 50, c_0 = 15.13$

(c) $T = 100, c_0 = 20.39$
Table 3.1: Standard values for the parameters in the simulations for the structural compliance minimization problem for a thermo-mechanical model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulus of elasticity</td>
<td>$E_0$</td>
</tr>
<tr>
<td>Poisson’s ratio</td>
<td>$\nu$</td>
</tr>
<tr>
<td>Conductivity</td>
<td>$k_0$</td>
</tr>
<tr>
<td>Coefficient of thermal expansion (CTE)</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>Volume Fraction</td>
<td>$f$</td>
</tr>
<tr>
<td>Filter’s radius</td>
<td>$r_{min}$</td>
</tr>
<tr>
<td>Mesh resolution</td>
<td>$N_x \times N_y$</td>
</tr>
<tr>
<td>Reference temperature</td>
<td>$T_{ref}$</td>
</tr>
<tr>
<td>Nodal Force</td>
<td>$F$</td>
</tr>
<tr>
<td>$= 1$</td>
<td></td>
</tr>
<tr>
<td>$= 0.3$</td>
<td></td>
</tr>
<tr>
<td>$= 1$</td>
<td></td>
</tr>
<tr>
<td>$= 5 \times 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>$= 0.4$</td>
<td></td>
</tr>
<tr>
<td>$= 1.2$</td>
<td></td>
</tr>
<tr>
<td>$= 60 \times 20$</td>
<td></td>
</tr>
<tr>
<td>$= 0$</td>
<td></td>
</tr>
<tr>
<td>$= -1$</td>
<td></td>
</tr>
</tbody>
</table>

The previous section highlighted potential effects of thermal expansion on a design but no thermal gradients were considered. This section addresses problems with non-zero heat flux boundary conditions.

3.4 Effects of Thermal Gradients

The previous section highlighted potential effects of thermal expansion on a design but no thermal gradients were considered. This section addresses problems with non-zero heat flux boundary conditions.

3.4.1 Problems with Heat Flux Boundary Conditions

To evaluate the influence of heat flux boundary conditions, this study case corresponds to a problem presented by Dongmei et al. [27]. The boundary conditions of the problem are given in Figure 3.3. A uniform heat flux flows into the domain at the top boundary, while the temperature $T_{sink}$ is prescribed at the bottom boundary.

Also, convergence for this thermo-mechanical problem required relaxing the stopping criteria. In fact, the density changes from one iteration to another are relatively small after only a few iterations. Convergence is slower as the effects of the thermal expansion become more pronounced.
In these conditions, the example in Figure 3.4 shows the results for the mechanical compliance optimization problem. The values of the parameters taken for this simulation are $q = 2$ and $T_{sink} = 0$. The resulting optimal design is not well defined with intermediate densities lying around the top area where the heat flux comes in. In this particular case, the convergence is quite fast and smooth. However, special attention to the values of the boundary conditions is needed when dealing with high temperatures. Large deviations in the temperature distribution can arise, especially in void elements, because of their very low non-zero conductivity value. Figure 3.4(b) illustrates this behavior, with temperatures reaching $T = 5000$ in some areas, while the maximum temperature within the actual structure is around $T = 300$ as suggested in Figure 3.4(d) $^1$.

The shape of the optimal structure shares common features with the designs of Section 3.3. However, elements with intermediate densities tend to be placed higher up in the design to link

$^1$The elements containing more than 50% of material are considered as part of the actual final structure.
Figure 3.4: Effects of heat flux boundary conditions on the special case of Figure 3.3. (a) Undeformed optimal design. (b) Temperature distribution in the undeformed design domain. (c) Deformed optimal design. (d) Temperature distribution in the optimal structure with the top boundary where the heat flux flows in. The more likely explanation for this phenomenon is that the high temperatures in the void elements tend to increase the effect of thermal expansions of the void elements. The thermal expansion of void elements can not be neglected anymore because of the extreme temperatures, which affect the derivatives of the objective function (or sensitivities). Therefore, new thermal pathways are generated. Just enough material is placed to possibly compensate these large deviations in temperature without deteriorating much the mechanical compliance of the cell.

In this case (Figure 3.4), the convergence proved to be quite smooth and almost monotonic such as shown in Figure 3.5(a). The change in element densities between two iterations is overall less than 1%, except around 20 iterations. In this example, the stopping criteria was relaxed and set to a lower bound of 0.2% maximum change in the material’s density. Besides the intermediate densities problem, the convergence does not suffer from any other setbacks. However, a high heat input at the boundary can slow the algorithm's convergence, and sometimes cause convergence
failure.

Typical convergence problem is illustrated in the following example: in Figure 3.6(a), the mechanical boundary conditions are modified from the original problem (Figure 3.3) by using three nodal forces pulling downward. The thermal boundary conditions are unchanged and given in Figure 3.3. The parameter values are given in Table 3.1. To facilitate the comparison with heated configurations, the undeformed optimum result for the structure at reference temperature (no thermal strain) is given in Figure 3.6(b). This optimum corresponds to the non-heated optimal structure with respect to the compliance problem.

When a heat flux input is introduced at the top boundary, the algorithm did not converge towards a final design unless the stopping criteria was much more strict. If the global features of
Figure 3.6: (a) New sets of mechanical boundary conditions tested, with three nodal forces pulling downwards. (b) Optimal design at reference temperature.

Figure 3.7: Evolution of the topology at various iteration steps.

the design are similar to the non-thermal strain case, isolated elements contain alternatively void or material, as shown in Figure 3.7. The iterative process was stopped before meeting the stopping criteria of 1% in maximum density change, but there was no indication that the algorithm would meet this requirement at a later stage.
Figure 3.8: Behavior of the objective function and density changes in the case of three forces pulling down. (a) Compliance value at each iterations. (b) Maximum density change in the domain between two iterations.

Indeed, the objective function presents irregularities, even though a low bound of $c_0 = 43$ (Figure 3.8 (a)). Also, peaks in the objective function values are observed at some iterations. The density changes criterion constantly reaches the value of 1, indicating that there exist at least one element that changes from void to material (or conversely) within the space of one iteration. This is typical of the problem with heat input in void element, in particular when the algorithm does not try to minimize the heat over the domain.

Several issues with the heat flux boundary conditions have been highlighted. Those issues are usually related, and are listed below:

- Possible numerical issues when solving the governing equations, with very high temperatures
in the void regions

- Limits using the SIMP interpolation of material properties, which cause dramatic changes in the sensitivities

- Oscillatory behavior of the densities through the iterative process, resulting in algorithmic instabilities (peaks in the objective function values)

These phenomena affect the topology, which constantly changes along with the objective function. These problems are not usually observed when minimizing the thermal compliance for thermal problems (Figure 2.11), because material with non-zero conductivity is placed near the so-called “hot” areas. Here, since the objective function is different, nothing constrains the void elements to be placed in these zones. An approach to addressing this problem consists of simulating a face sheet with high conductivity.

### 3.4.2 Simulating a High-conductivity Face Sheet for Thermal Conduction Enhancement

The introduction of a high conductive layer along the top boundary serves two main purposes. First, it alleviates the "hot spots" problem with the void elements discussed earlier. Second, it simulates a high conductivity face sheet which enhances heat propagation into the structure, and recreates the C3Ms environment. Indeed, it is expected that the sandwich panel design of the C3Ms is bounded with high conductive surfaces to enhance the thermal control capabilities.

In the finite element formulation, it is possible to create an artificial layer of highly conductive material at the top of the design domain, as shown in Figure 3.9. The thermal conductivities of the elements at the heated boundary are set to a large value in the conductivity matrix. In the following simulations, the conductivity of the corresponding elements have been set to \( k = 100 \times k_0 \) to ensure a uniform distribution regardless of the heat flux input value.

Figure 3.10 presents the results for the boundary conditions of Figure 3.1. As the heat flux increases, similarities in the optimal shapes can be seen with the designs obtained for a uniform temperature increase shown in Figure 3.2.
Figure 3.9: Schematic of the boundary conditions with the introduction of a high conductive layer.

\[
q = 1, \quad T_{\text{sink}} = 0
\]

\[
q = 2, \quad T_{\text{sink}} = 0
\]

Figure 3.10: Mechanical compliance-optimal undeformed designs [(a) & (b)] for various values of the heat flux and the corresponding temperature contours of the domain [respectively (c) & (d)] and in the deformed structure [respectively (e) & (f)]. The high conductive layer is non-designed.
In results (a) and (b) of Figure 3.10 the high conductive layer is non-designed. Intermediate densities are observed in elements around the heat flux boundary. These partially material and void elements are increasingly present as the temperatures get high. However, this "fictive" layer improves the topology in the sense that it constitutes a clear defined structural shape. Also, the heat propagates much more efficiently into the structure and hot spots in the void elements are avoided, as shown by the temperature contours in Figure 3.10(c) and (d). In hot uniform temperature conditions the algorithm tends to place voids close to the top boundaries (such as in Figure 3.2(c)). In the case of a heat flux boundary condition, the algorithm tends to place material in this region. If no link with the heated boundary is made during the optimization process (like in 3.10(a), the distribution of intermediate densities is forced to create a junction with the top boundary and conduct the heat into the structure. In the particular case of Figure 3.10(d), the highest temperature in the domain is significantly above the one into the structure given in Figure 3.10(f). Non of these added pathway is needed in Figure 3.10(d) and (e), since the placement of material, for the algorithm, comes as a natural choice from a mechanical compliance point of view. Therefore, when optimizing mechanical properties of a structure and in the presence of heat flux boundary conditions, the structure must be in contact with the high conductive layer to avoid intermediate densities. Also, lower heat input values help avoiding large temperature deviations.

In these examples, the conductive layer was declared as “fictive” or non-designed, because the moduli of elasticity of the modified elements followed the SIMP interpolation scheme. Another possibility is to declare this high conductive layer as a designed, physical material with a defined modulus of elasticity $E_0$. In the example of Figure 3.11, passive (designed) elements on the top boundary are defined as material ($\rho_e = 1$) with a modulus of elasticity $E_0 = 1$ throughout the optimization process. This case modifies slightly the optimal design when compared to Figure 3.10(a) for the same boundary conditions. The mechanical compliance optimization performance of the structure also gets worse. However, this representation might be a more realistic way to simulate a face sheet. Finally, the introduction of a single solid layer on top of the domain has a limited impact on the design when using a finer mesh for the FEM.

\[2\text{In this representation, element parts of the final solid structure contain more than 50% of material}\]
Figure 3.11: Introduction of a solid conductive layer on the mechanical compliance minimization problem and for the boundary conditions in Figure 3.6. (a) Undeformed optimal topology. (b) Temperature distribution in the domain. (c) Temperature distribution within the deformed structure.

Returning to the three-forces case of Figure 3.6, the high conductive layer significantly improves the algorithm behavior. First, the stopping criteria is met in each configuration (less than 0.2% of the maximum change in element density). On the compliance-optimal designs in Figure 3.12(a) and (b), void regions are clearly defined, and no isolated material element appears as in Figure 3.7. Note that in these configurations, the high conductive layer is non-designed. If high temperature conditions lead the topology to lie lower in the design domain, though pathways are generated to conduct the heat into the structure. This is well illustrated in Figure 3.12(f), where the heat in the structure is principally coming through the top middle elements. Comparison with the temperature distributions of the two cases of Figure 3.12(c) and (d) suggests that heat propagation in the structure is important for the temperature in void regions not to be too large.
Figure 3.12: Compliance-optimum designs [(a) & (b)] for different values of the heat flux, their corresponding (non-dimensional) temperature contours of the domain [respectively (c) & (d)] and in the deformed structure [respectively (e) & (f)].

Following Figure 3.13 and Figure 3.14, the behavior of the objective function is also much smoother for both configurations. Density changes can be decomposed into three phases in this problem:

- **Phase 1:** The density changes are relatively small and constant over the domain. With a uniform repartition of the material as a starting guess, the domain does not present any clear features. In this case this phase lasts for around 15 iterations (see Figure 3.15(a)).

- **Phase 2:** The density changes can be high and present high variations, for about 20 iterations here. The principal features of the final design appear. (see Figure 3.15(b))

- **Phase 3:** The design is barely changing. These corresponds to refinements of the final design, where densities change locally in some elements. (see Figure 3.15(c))
Figure 3.13: Evolution of the objective function (a) and the density changes between each iteration (b) for $q = 1, T_{\text{sink}} = 0$.

In general, the higher the heat flux conditions, the higher the irregularities in the objective function. In the third phase, element densities that change the most are usually those close to the face sheet, at the junction with the structure.
Figure 3.14: Evolution of the objective function (a) and the density changes between each iterations (b) for $q = 2, T_{sink} = 0.$
Figure 3.15: Undeformed design domain after: (a) 15 iterations, (b) 35 iterations, (c) 225 iterations ($q = 1, T_{sink} = 0$).
3.5 Conclusions on the Thermo-mechanical model

This chapter presented the implementation of a coupled model for the mechanical and thermal system of equations. The principal findings were:

- The mechanical behavior of the algorithm for the structure under thermal loading efficiently reproduces results of previous analysis for a uniform temperature distribution.

- Heat boundary conditions can create hot-spots in void elements, causing convergence issues for the optimization process, as well as design singularities.

- Topology optimization results when dealing with heat flux boundary conditions have to be carefully analyzed. Topologies usually include thermal pathways, if not naturally built in the optimization process. This can favor the appearance of intermediate densities in the design.

Topology optimization using heat flux boundary conditions requires special attention and algorithmic testing. A good approach is to perform initial tests using prescribed temperature conditions and intuitive objective functions like the compliance of the structure. Also, excessive heat input increases numerical instabilities and ill-conditioned matrices. Finally, adjusting the stopping criteria may be necessary for the algorithm to “converge”.
Chapter 4

Two-Material Models

4.1 SIMP Interpolation for Two-material Designs

Two-phase topology optimization problems have been explored so far, where one phase was referred to as void and the other as material. However, the concept of the cellular contact-aided compliant mechanisms (C3Ms) presented in Section 1.1.3 is based on the mismatch of the coefficients of thermal expansion (CTEs) of two different materials, which thereby would enable contact. In this section, a three-phase (two-material plus void) implementation is presented. With respect to the 1-material methods presented in the previous sections, the major changes involve the SIMP approach and the sensitivity analysis. Internal contact between materials is not considered. The other assumptions remain the same as those presented in previous sections. Therefore, the governing equations derived earlier are not modified. In the presence of a material 1 and a material 2, the SIMP approach for a 2-material model is described by Sigmund [31]:

\[
E(\rho_0^e, \rho_1^e) = (\rho_0^e)_{p0} ((\rho_1^e)^{p1} E_1 + (1 - \rho_1^e)^{p1} E_2)
\]

\[
k(\rho_0^e, \rho_1^e) = (\rho_0^e)_{p0} ((\rho_1^e)^{p1} k_1 + (1 - \rho_1^e)^{p1} k_2)
\]

\[
\alpha(\rho_0^e, \rho_1^e) = (\rho_1^e)^{p1} \alpha_1 + (1 - \rho_1^e)^{p1} \alpha_2
\]

where the quantities in Equation (4.1) are given in Table 4.1. For clarity’s sake, the element-wise quantities are represented by a superscript \( e \).

The vectors of design variables \( \rho_0, \rho_1 \) control the quantities of the material in the domain. In
Table 4.1: Quantities appearing in the SIMP interpolation of a 2-material model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>Modulus of Elasticity of material 1</td>
</tr>
<tr>
<td>$k_1$</td>
<td>Conductivity of material 1</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>CTE of material 1</td>
</tr>
<tr>
<td>$E_2$</td>
<td>Modulus of Elasticity of material 2</td>
</tr>
<tr>
<td>$k_2$</td>
<td>Conductivity of material 2</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>CTE of material 2</td>
</tr>
<tr>
<td>$\rho_0^e, \rho_1^e$</td>
<td>Element design variable</td>
</tr>
<tr>
<td>$p_0, p_1$</td>
<td>Penalization terms</td>
</tr>
</tbody>
</table>

particular, $\rho_0$ is a density variable controlling the presence of void ($\rho_0 = \rho_{\text{min}} \sim 0$) or material ($\rho_0 = 1$). $\rho_1$ is not strictly a density variable but regulates the presence of one material over the other. $\rho_1 = 1$ indicates the presence of material 1 while $\rho_1 = 0$ indicates the presence of material 2. The penalty term $p_0$ is set equal to 3, since its role is equivalent to the penalty term used in single material models. $p_1$, however, is chosen to be equal to 1: if an element contains material, the material properties of this element are determined by the weighted average of each material property, where the weight is given by the value of $\rho_1$.

### 4.2 Sensitivity Analysis

The sensitivity analysis is similar to that described in the previous sections, though the Lagrangian has to be derived considering both design variables. Detailed development of this part is given for a thermo-mechanical model, though mechanical and thermal systems follow the same scheme:

$$\begin{align*}
\text{minimize} \quad & L(\rho_0, \rho_1) = c_0(\rho_0, \rho_1) + \lambda_{\text{th}}(K_{\text{th}}T - F_{\text{th}}) + \lambda_{\text{m}}(K_{\text{m}}d - F_{\text{m}}(T)) \\
\text{subject to} \quad & c_i(\rho_0, \rho_1) \leq 0, \ i = 1, \ldots, l.
\end{align*}$$  \hspace{1cm} (4.2)

where $\lambda_{\text{th}}$ and $\lambda_{\text{m}}$ are the vectors of adjoint variables. The volume constraints $c_i$ can now be expressed as a set of three constraints. Each volume constraint controls the presence of each phase.
as follows:

\[
\text{void: } 1 - \frac{V_e}{V(1-f_0)} \sum_{e=1}^{N} (1 - \rho_0^e) \leq 0
\]

material 1: \[1 - \frac{V_e}{V f_0 f_1} \sum_{e=1}^{N} \rho_0^e \rho_1^e \leq 0 \quad (4.3)
\]

material 2: \[1 - \frac{V_e}{V f_0 (1 - f_1)} \sum_{e=1}^{N} \rho_0^e (1 - \rho_1^e) \leq 0 \]

where \(V_e\) denotes the element volume and \(V\) the total volume of the domain. \(f_0\) and \(f_1\) characterize the overall quantity of each phase in the volume. In particular, \(f_0\) sets the fraction of material (and thus void) in the domain, while \(f_1\) regulates the trade-off between the presence of material 1 and 2. Note that it is possible to define one constraint as a function of the other two, since the sum of the normalized volume fractions must be equal to 1. In some special cases, those constraints have been alternatively dropped to relax the optimization and let the algorithm choose one material over the other. Similarly to Section 3.2, the derivatives of the Lagrangian for each element are, with \(i = 0, 1\):

\[
\frac{\partial L}{\partial \rho_i^e}(\rho_0^e, \rho_1^e) = \frac{\partial c_0}{\partial \rho_i^e} + \lambda_{th}^T \left( \frac{\partial K_{th}}{\partial \rho_i^e} T - \frac{\partial F_{th}}{\partial \rho_i^e} \right) + \lambda_m^T \left( \frac{\partial K_m}{\partial \rho_i^e} d - \frac{\partial F_m}{\partial \rho_i^e} \right)
\]

\[
+ \frac{\partial c_0}{\partial d} \frac{\partial d}{\partial \rho_i^e} + \frac{\partial c_0}{\partial T} \frac{\partial T}{\partial \rho_i^e}
\]

\[
+ \lambda_{th}^T \left( K_{th} \frac{\partial T}{\partial \rho_i^e} \right) + \lambda_m^T \left( K_m \frac{\partial d}{\partial \rho_i^e} - \frac{\partial F_m}{\partial \rho_i^e} \frac{\partial T}{\partial \rho_i^e} \right)
\]

The adjoint method consists of eliminating the terms \(\partial d/\partial \rho_i^e\) and \(\partial T/\partial \rho_i^e\). Although the Lagrangian is derived with respect to two design variables, only two adjoint equations have to be solved:

\[
\left( \frac{\partial c_0}{\partial T} + \lambda_{th}^T K_{th} - \lambda_m^T \frac{\partial F_m}{\partial T} \right) = 0 \quad \text{and} \quad \left( \frac{\partial c_0}{\partial d} + \lambda_m^T K_m \right) = 0 \quad (4.5)
\]

Therefore, the Lagrangian derivatives are determined explicitly :

\[
\frac{\partial L}{\partial \rho_i^e}(\rho_0^e, \rho_1^e) = \frac{\partial c_0}{\partial \rho_i^e} + \lambda_{th}^T \left( \frac{\partial K_{th}}{\partial \rho_i^e} T - \frac{\partial F_{th}}{\partial \rho_i^e} \right) + \lambda_m^T \left( \frac{\partial K_m}{\partial \rho_i^e} d - \frac{\partial F_m}{\partial \rho_i^e} \right)
\]
Finally, in the case where $\partial F_m / \partial \rho_i^e = \partial F_{th} / \partial \rho_i^e = 0$, the expression of the Lagrangian derivatives are simplified as follow:

$$\frac{\partial L}{\partial \rho_i^e} (\rho_0^e, \rho_1^e) = \frac{\partial c_0}{\partial \rho_i^e} + \lambda_{th} T \frac{\partial K_{th}}{\partial \rho_i^e} T + \lambda_m T \frac{\partial K_m}{\partial \rho_i^e} d$$

(4.7)

The filtering technique follows as in the previous implementations, except that the sensitivities have to be filtered for both variables. For the density variables $\rho_0$, the filtering technique remains the same as in Equation 2.15:

$$\frac{\hat{\partial} L}{\partial \rho_0^e} (\rho_0^e, \rho_1^e) = \frac{1}{\max(\rho_0^e, \rho_{min}^e)} \sum_{i=1}^{N} \hat{H}_i \rho_0^e \frac{\partial L}{\partial \rho_0^e}$$

(4.8)

According to Sigmund [31]: “experience shows that the density weighting stabilizes convergence and provides a smoothing of the sensitivities and the resulting design”. However, the design variables $\rho_1$ are not strict density variables. It only regulates the trade-off between the presence of material 1 and material 2, and does not require weighting in the filtering technique. This way, the filtering of the sensitivities with respect to $\rho_1^e$ becomes:

$$\frac{\hat{\partial} L}{\partial \rho_1^e} (\rho_0^e, \rho_1^e) = \frac{1}{\sum_{i=1}^{N} \hat{H}_i} \sum_{i=1}^{N} \hat{H}_i \rho_1^e \frac{\partial L}{\partial \rho_1^e}$$

(4.9)

Finally, the inputs for the MMA subroutine scheme for the update of the design variables developed by Svanberg [32] needs to optimize for both $\rho_0$ and $\rho_1$; namely, twice as much design variables than for a single material model.

To the author’s knowledge, few papers considered this problem for thermal and mechanical couplings for two-material structures [31].
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson’s ratio</td>
<td>$\nu$</td>
</tr>
<tr>
<td>Volume Fraction of void</td>
<td>$1 - f_0$</td>
</tr>
<tr>
<td>Volume Fraction of material 1</td>
<td>$f_0 \times f_1$</td>
</tr>
<tr>
<td>Volume Fraction of material 2</td>
<td>$f_0 \times (1 - f_1)$</td>
</tr>
<tr>
<td>Filter’s radius</td>
<td>$r_{\text{min}}$</td>
</tr>
<tr>
<td>Mesh resolution</td>
<td>$N_x \times N_y$</td>
</tr>
</tbody>
</table>

Table 4.2: Standard values for the parameters in the simulations for the structural compliance minimization problem for a 2-material approach.

## 4.3 Results on Single Physics Problems

### 4.3.1 Mechanical Systems

Optimization of the mechanical compliance of the structure was performed considering only a mechanical system of equations. Thus, the objective function is given by Equation (2.17). The system of equations is equivalent to the model introduced in Section 2.1 with the added complexity of a 2-material plus void model. The standard values used are non dimensional and are given in Table 4.2. The parameter $f_0$ constitutes the total volume fraction of materials in the domain, while $f_1$ is not exactly volume fractions, but defines together with $f_0$ the volume fraction of each individual material. In the following simulations, $f_0 = 0.5$ and $f_1 = 0.4$. In the topology representations, the material with the highest modulus of elasticity ($E_1$) is represented in red and will be referred to as material 1. Material 2, in blue, is defined as the material with the lowest modulus of elasticity ($E_2$). The void phase is still represented in white. The modulus of elasticity of each material is modified to study the effect of the moduli ratio ($E_2/E_1$) on the design.

As the modulus of elasticity ratio increases, the design does seem to be slightly modified in Figure 4.1. However, the materials seem to be placed in the same areas, regardless of the moduli ratio. The “stiffest” material in red is placed near solid boundaries and where high loadings can appear. Material 2 acts as a secondary support, like a supportive layer around material 1 in the areas that matter, or in the inner part of the domain where stresses are lower. The topology seems to be very dependent on the material with the higher modulus, and much less on the other material.
It is also possible to remove the constraint on the volume fraction of the materials to be used in the domain. Therefore, the algorithm chooses which material to be placed (1 or 2), and the only volume constraint is the amount of void over the domain. In this simple case of mechanical compliance optimization, logically, the material with the highest modulus of elasticity is chosen over the other one (Figure 4.1(d)), regardless of the starting guess for the volume fraction (or $\rho_1$ values).

Figure 4.1: Topology optimization of a 2-material mechanical model. (a) Boundary conditions. (b) Undeformed optimal topology for $E_1 = 2$ and $E_2 = 1$. (c) Undeformed optimal topology for $E_1 = 4$ and $E_2 = 1$. (d) Undeformed optimal topology for $E_1 = 2$ and $E_2 = 1$, with the volume fraction of material 1 and 2 unconstrained. Colors: white is associated with void elements, red with material 1 and blue with material 2.

The objective function behavior for the case in Figure 4.1(b) is fairly smooth (see Figure 4.2, and can be regarded as a generalization of the behavior of the other configurations tested. However, irregularities around the final compliance value are clearly visible on Figure 4.2(b). The maximum density change is characterized by the maximum amount of change in both the $\rho_0$ and the amount
Figure 4.2: Algorithm behavior for the example given in Figure 4.1(b), where $E_1 = 2$ and $E_2 = 1$.
(a) Objective function values at each iterations (b) Zoom around the final value of the objective function.
of material 1 ($\rho_0 \times \rho_1$). Also, the maximum density change at each iteration exhibits an oscillatory behavior, typical of a two-material optimization problem. More interestingly, the oscillations for each phase seem very similar (Figure 4.3): little changes in the proportion of void over the domain (blue) coincides with little changes in the proportions of material (red). Very little improvement on the objective function is made after 100 iterations, and no major changes are observed in the topology. Therefore, the stopping criterion of the algorithm can be defined with respect to the changes in the void phase without much loss of optimality of the objective function. From now on, simulations with two-material designs will consider the stopping criterion to be a small fraction of changes in the void phase only. A maximum of 1 % of change in the void element density can be considered as a good stopping criterion.

4.3.2 Thermal Systems

Optimization of the thermal compliance, given in Equation (2.24), is expected to proceed similarly to the mechanical model. The system of equations is equivalent to the model introduced in Section 2.1 with the added complexity of a 2-material plus void model. Note that, as mentioned in Section 2.2, the materials are isotropic and the material properties are independent from temperature.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conductivity of material 1 (in red)</td>
<td>$k_1$</td>
</tr>
<tr>
<td>Conductivity of material 2 (in blue)</td>
<td>$k_2$</td>
</tr>
<tr>
<td>Volume fraction of void</td>
<td>$1 - f_0$</td>
</tr>
<tr>
<td>Volume fraction of material 1</td>
<td>$f_0 \times f_1$</td>
</tr>
<tr>
<td>Volume fraction of material 2</td>
<td>$f_0 \times (1 - f_1)$</td>
</tr>
<tr>
<td>Filter’s radius</td>
<td>$r_{min}$</td>
</tr>
<tr>
<td>Mesh resolution</td>
<td>$N_x \times N_y$</td>
</tr>
</tbody>
</table>

Table 4.3: Standard values for the parameters in the simulations for the thermal compliance minimization problem for a 2-material approach.

The standard values used in the simulations are non dimensional quantities and given in Table 4.3. Similarly to the mechanical counterpart, the material with the highest conductivity ($k_1$) is represented in red and will be referred to as material 1. Material 2, in blue, is defined as the material with the lowest conductivity ($k_2$).

The configurations tested are similar to those studied in Section 2.2, where the domain is heated uniformly. In Figure 4.4 (a), (b) and (c), prescribed temperature boundary conditions are indicated by an orange outline, while insulated sides are indicated in black lines. Logically, the material with the highest conductivity (red) is essentially placed near the areas where the temperature is prescribed and enhances the heat conduction at those boundaries (Figure 4.4). The less conductive material (blue) is placed around the areas occupied by the red material.

Finally, if the relative amount of material 1 and material 2 is not constrained, the material with the highest conductivity is the unique material chosen by the algorithm for thermal compliance optimization. This result is observed regardless of the initial guess for the volume fractions of the materials. Also, the remarks on convergence are the same as for the mechanical system. The stopping criterion was set to a maximum change in material/void density of 1% over the domain, and the algorithm always converged.
Figure 4.4: Sample results for a thermal compliance optimization problem using 2 materials. (d), (e) and (f) are material distributions for a uniformly heated domain and for the boundary conditions (a), (b) and (c) respectively. (g), (h) and (i): Temperature profile at steady state for the distribution (d), (e) and (f) respectively (temperatures are non-dimensional).

4.4 Thermo-mechanical Two-material Model

Again, the system of equations is unchanged from Section 3.1, and a high conductivity face-sheet is simulated at the top boundary. The mechanical compliance is considered as the objective function to test the algorithm. The standard values for the parameters in the following simulations are given in Table 4.4. Those values are subject to change when indicated. Nonetheless, as a general rule, material 1 (red) is considered to be the material with the highest modulus of elasticity. In
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulus of elasticity of material 1</td>
<td>$E_1$</td>
</tr>
<tr>
<td>Modulus of elasticity of material 2</td>
<td>$E_2$</td>
</tr>
<tr>
<td>Conductivity of material 1</td>
<td>$k_1$</td>
</tr>
<tr>
<td>Conductivity of material 2</td>
<td>$k_2$</td>
</tr>
<tr>
<td>CTE of material 1</td>
<td>$\alpha_1$</td>
</tr>
<tr>
<td>CTE of material 2</td>
<td>$\alpha_2$</td>
</tr>
<tr>
<td>Reference temperature</td>
<td>$T_{ref}$</td>
</tr>
<tr>
<td>Sink temperature</td>
<td>$T_{sink}$</td>
</tr>
<tr>
<td>Heat flux</td>
<td>$q$</td>
</tr>
<tr>
<td>Poisson’s ratio</td>
<td>$\nu$</td>
</tr>
<tr>
<td>Volume Fraction of void</td>
<td>$1 - f_0$</td>
</tr>
<tr>
<td>Volume Fraction of material 1</td>
<td>$f_0 \times f_1$</td>
</tr>
<tr>
<td>Volume Fraction of material 2</td>
<td>$f_0 \times (1 - f_1)$</td>
</tr>
<tr>
<td>Filter’s radius</td>
<td>$r_{\text{min}}$</td>
</tr>
<tr>
<td>Mesh resolution</td>
<td>$N_x \times N_y$</td>
</tr>
</tbody>
</table>

Table 4.4: Standard values for the parameters in the simulations for the mechanical compliance minimization problem for a two-material thermo-mechanical approach.

In the following simulations, the goal is to observe possible differences in the placement of the two materials depending on the material properties and the boundary conditions.

The boundary conditions are the same as those prescribed in Section 3.4, with three nodal forces pulling downwards at the bottom of the domain and a heat flux flowing in on top (Figure 4.5). The high conductive layer introduced in Section 3.4.2 is used and is solid non-designed.
Figure 4.5: Boundary conditions used in the tests for the thermo-mechanical two-material problem. (a) Mechanical boundary conditions. (b) Thermal boundary conditions.

4.4.1 Simulations with Constrained Material Volume Fractions

Figure 4.6 presents optimal topologies for various heat flux values. The structure’s shapes are similar to the ones obtained in Section 3.4 for a one-material model. However, patterns in the placement of the two materials can now be observed. The topologies suggest the material with high elastic modulus (red) is to be placed close to the high stress regions, near the mechanical forces or fixed displacement regions. In particular, when no heat is applied to the domain (Figure 4.6(a)), this material is at the core of the structure, while material 2 (blue) is placed in so-called “annex” regions, or even on the contour of the red areas. As the heat flux is increased, material 2 is placed in more strategic locations. In Figure 4.6(b) this material is placed near the top boundary. The result is consistent with the results examined for a one-material model, where the thermal expansion does not improve the objective function. In the other designs (Figure 4.6(b) & (c)), material 2 is placed in areas where either low conduction or low coefficient of thermal expansion (CTE) is required to reduce the structure’s compliance.
Figure 4.6: Undeformed optimal designs for various heat flux input at the top boundary. Colors: white areas are associated with void elements, red with material 1 and blue with material 2

(a) \( q = 0, c_0 = 17.35 \)

(b) \( q = 1, c_0 = 22.08 \)

(c) \( q = 1.5, c_0 = 21.82 \)

(d) \( q = 2, c_0 = 22.98 \)

Figure 4.7: Undeformed optimal topology for the minimization of the mechanical compliance. The modification of the CTEs can affect the placement of material 1 and 2. Colors: white areas are associated with void elements, red with material 1 and blue with material 2

(a) \( \alpha_1 = 5 \times 10^{-4}, \alpha_2 = 2 \times 10^{-4}, c_0 = 21.82 \)

(b) \( \alpha_1 = 2 \times 10^{-4}, \alpha_2 = 5 \times 10^{-4}, c_0 = 21.89 \)

In Figure 4.7, the effect of the CTE values, and in particular the ratio of the two CTEs is examined (for \( q = 1.5 \) and the standard values of Table 4.4). Interestingly, the placement of the material is nearly opposite for the inverted CTE ratios. For the same conditions, although the shape of the structures are very similar, the material with the lowest CTE is systematically placed in the middle and top corner branches of the structure. However, Material 1, with the highest modulus, is placed at the boundaries where the stresses and loading are high.

To a lesser extent than the CTEs, the conductivity ratios affect the design. In Figure 4.8, the
Figure 4.8: Undeformed optimal topology for the minimization of the mechanical compliance. The modification of the conductivities can affect the placement of material 1 and 2. Colors: white areas are associated with void elements, red with material 1 and blue with material 2.

placement of the highly conductive material is strategic to keep the structure cool, and reduce the thermal strains. The placement of the highly conductive material become increasingly important with the heat flux at the boundary.

4.4.2 Partially Constrained Problems

All the above simulations have been performed with the volume of material 1 and 2 over the domain both prescribed (20% of each material over the domain). However it is possible to alleviate this volume constraint. The only volume constraint is thereby the presence of material/void (kept at 40%/60% here). Since no information concerning the final amount of each material in the design the available, the initial guess for the design variables $\rho_1$ is less intuitive. The starting guess for the initial volume fraction of each material is 20% (equal initial amount). For different initial guesses, the optimal topologies are globally similar, and the final volume fractions of each material are usually the same. However, extensive study might be required since the results can vary given the parameters and boundary conditions. In the following figures, the optimal topologies for the volume-constrained problems and their unconstrained counterparts are displayed next to each other.

In most cases, for the current compliance optimization problem (Figure 4.5), the material with the highest modulus of elasticity (material 1, in red) is the unique choice in the domain, as illustrated in Figure 4.9(b). Even for very high temperature conditions the modulus of elasticity is the parameter that sets the material to be placed in priority. This phenomenon is not that surprising given that the objective - the mechanical compliance - is to generate the stiffest topology. However,
Figure 4.9: Comparison of undeformed topologies of volume-constrained optimization and unconstrained optimization. In (a) & (c), the Volume of material 1 and 2 constrained to 20% each. In (b) & (d), the amount of material 1 and 2 in the domain not prescribed. Colors: white areas are associated with void elements, red with material 1 and blue with material 2.

Figure 4.10: Unconstrained optimization with $k_1 = 1, k_2 = 5$ and $\alpha_1 = 10 \times 10^{-4}, \alpha_2 = 2 \times 10^{-4}$. Undeformed optimal designed where materials’ volume fraction is unconstrained. (a) $q = 1$ (b) $q = 1.5$ Colors: white areas are associated with void elements, red with material 1 and blue with material 2.

for some configurations, the use of the other material can be chosen, as in Figure 4.9(d). In this figure, the conductivity ratio is inverted compared to the preceding case (Figure 4.9(b)), and leads to significant design changes.

Following the idea that the conductivity ratio affects the material placement in the algorithm decision, Figure 4.10 shows optimal topologies for varying heat flux. The presence of the highly conductive material becomes more pronounced as the heat input increases. Finally, the behavior of the objective function is similar to the behavior observed for the mechanical problem (Section...
4.3.1). Partially-constrained volume fraction problems are only interesting if different parameters can affect the objective function and could be of interests in thermally actuated mechanical structures.

4.5 Implications of Two-Material Model

Finally, a partially constrained problem as discussed here seems a relevant approach for the design of the C3Ms for thermal control. Topology optimization can thus provide answers to questions like:

- How much of each material should be used?
- Where should these be placed?

Certainly, the thermal expansion effects are essential in the development of the optimal topology for the C3Ms, though maybe not sufficient. The contact at the bottom interface is yet to be modeled and requires significant changes to the governing equations.
Chapter 5

Implementation of Unilateral Contact

Under various loadings, and in particular when subject to thermal expansion, the C3Ms must be able to make or break contact with the different surfaces for effective thermal control. The contact is defined by the distance between the structure and a rigid support. The presence of a normal contact force must depend on this condition. Also, the estimation of the thermal contact resistance (TCR), which characterizes the thermal contact, requires the knowledge of the pressure at the interface. Therefore, the previous model is extended to account for unilateral mechanical contact at the domain boundary. Solving the governing equations involves iterative techniques such as the Newton’s method with an Armijo line search. This formulation is essentially based on the model developed by Strömberg and Klarbring [35].

5.1 Signorini’s Contact Model

The displacements within the rectangular domains studied in previous sections can generate contact with an external support. In this model, the contact is frictionless and assumed to be unilateral. The surface of a material usually presents some irregularities as depicted in Figure 5.1(a). In the FEM formulation, the surface is considered to be linear between two nodes. A rigid support is located at a prescribed distance (possibly null) from the initial position of the undeformed body, when no loading is applied. In Figure 5.1(b), the vector $\tilde{n}_A$ is no longer the real normal surface vector, but indicates the direction in which the body will exert a force when contact is made with the rigid support.
Following Strömberg and Klarbring [35], the contact is modeled using Signorini’s contact model. With $\xi$ the contact force and $\eta$ the contact gap between the body and the fixed support, Signorini’s contact conditions are given as:

$$\xi \geq 0, \quad \eta \geq 0, \quad \xi \eta = 0$$

where $\xi$ and $\eta$ are independent from one another. Note that if one of the two terms is non-zero, the other has to be null. Strömberg and Klarbring [35] extended this approach by using a smooth formulation, and a non smooth formulation related to an augmented Lagrangian approach. They found that the non-smooth formulation was a little more efficient than the smooth one. In this
study, the non-smooth formulation is chosen, though a smooth formulation could be used. For more information on the smooth formulation, the interested reader is referred to the work of Facchinei et al. [36].

The non-smooth formulation follows closely the definition of Signorini’s contact conditions. The contact function in the Lagrangian approach is defined as:

\[ \Psi(\xi, \eta) = -\xi + (\xi - \eta)_+ \]  \hspace{1cm} (5.2)

where \((\xi - \eta)_+\) is the function defined as:

\[
(\xi - \eta)_+ = \begin{cases} 
0 & \text{if } \xi - \eta \leq 0 \\
\xi - \eta & \text{if } \xi - \eta > 0 
\end{cases}
\]  \hspace{1cm} (5.3)

This function is equivalent to \(\Psi = -\min(\xi, \eta)\).

5.2 Contact for a Single Mechanical System

5.2.1 System of Equations

The forcing vector on the body’s boundary is now subject to change in case of contact. The governing equations are thereby modified in order to take into account the contact condition. Thermal strains and thermally-induced contacts are not considered in this section, which will be studied later. The governing equations are developed for the mechanical behavior only. The finite element formulation presented in Section 2.1 is unchanged, though the system includes the following contact conditions:

\[ d^A \cdot n^A - g^A \leq 0 \]  \hspace{1cm} (5.4)

where \(g^A\) is the vertical distance from node \(A\) to the rigid support and \(d^A\) is the displacement vector of node \(A\) (see Figure 5.2). Essentially in this case, the only displacement considered for the contact is the vertical displacement. For example, for \(n^A = (0, -1)\), the approximation \(d^A \cdot n^A = -d^A_y\) is obtained, where \(d^A_y\) is the vertical displacement at node \(A\).
Figure 5.2: Undeformed linear elastic body facing a rigid support. The initial distance from point A to the rigid support is denoted as $g_A$.

Letting $nc$ be the number of nodes in potential contact at the boundary and $n$ the total number of nodes in the domain, the contact conditions can be rearranged such that:

$$C_n d - g \leq 0 \quad (5.5)$$

where $C_n$ is a $nc \times 2n$ matrix of zeros and ones such that, if $C_n^A$ is a row of $C_n$, we have:

$$C_n^A d = d^A \cdot n^A = -d^A_y \quad (5.6)$$

Therefore, the governing equations for the mechanical system presented in Section 2.1.2 become:

$$K_m d + C_n^T P_n = F_p \quad (5.7)$$

with $P_n$ being the vector of normal contact forces at the contact interface. Given this notation, Signorini’s contact conditions introduced in Section 5.1 at a node $A$ can be written as:

$$P_n^A \geq 0, \quad C_n^A d - g^A \leq 0, \quad P_n^A r(C_n^A d - g^A) = 0 \quad (5.8)$$

where $P_n^A \sim \xi$ and $r(C_n^A d - g^A) \sim \eta$. The scalar $r > 0$ is introduced for numerical reasons, but it does not change the constraint equations. $r$ is set to a large value (say $10^6$) to increase the numerical precision of the condition $C_n^A d - g^A = 0$. Now, the contact forces have to be determined along with the displacements. It follows that, for a node $A$, the contact function satisfies the equation:
\[ \Psi(d, P^A) = -P^A_n + (P^A_n + r(C^A_n d - g^A))_+ = 0 \] (5.9)

Instead of dealing with \( N \) unknowns (displacements), the system of equation needs to be solved for \( nc \times n \) unknowns and as many equations, where \( nc \) designates the number of constraint equations. Therefore, the system of equations that needs to be solved can be written as:

\[
\begin{cases}
K_m d + C^T_n P_n = F_p \\
\Psi(d, P_n) = 0
\end{cases}
\] (5.10)

5.2.2 Numerical Approach

The system of equations (5.2.2) depends on both displacements \( d \) and contact normal forces \( P_n \). The equations are strongly coupled, unlike the thermo-mechanical system of Chapter 3. In particular the contact function \( \Psi(d, P_n) \) dictates the nature of the contact forces according to the displacement values of the body. In return, normal forces at the interface have the ability to constrain the displacements in the structure. Therefore, an iterative approach based on a Newton’s method is used to determine the solution of the governing equations. The interested reader is referred to the work of Strömberg [35] [37] [38] for further information on the Newton’s method. Solving the governing equations can be seen as an optimization problem where the objective consists of minimizing the potential energy of the structure. Details on the system of equations are presented below.

Define \( \Gamma \) as the set of \( (nc) \) nodes \( A \) that can potentially be in contact with the rigid support. These nodes verify the contact condition in Equation (5.9), and the set of points \( \Gamma \) that verifies this condition is described as:

\[
\Gamma = \left\{ A, \text{such that} \ P^A_n \geq 0, \ C^A_n d - g^A \leq 0, \ P^A_n r(C^A_n d - g^A) = 0 \right\} \] (5.11)

Thus, it is possible to rewrite the constraint equations of Equation (5.9) in the following matrix form:

\[
r \bar{C}_n d + \bar{1}_{nc} P_n = r \bar{g} \] (5.12)
where the matrix $\mathbf{C}_n$ is a $nc \times n$ matrix. For a node $B$, the corresponding line of $\mathbf{C}_n$ is filled as:

$$
\mathbf{C}^B_n(j) = \begin{cases} 
0 & \text{if } B \not\in \Gamma \\
1 & \text{if } d^B_n \cdot n^B \geq g^B \\
0 & \text{if } d^B_n \cdot n^B < g^B 
\end{cases}
$$

(5.13)

where $j \in [1,n]$ is the column index of $\mathbf{C}_n$. The $nc \times nc$ square matrix $\mathbf{1}_{nc}$ is filled as follow, for each element $(i,j)$:

$$
\mathbf{1}_{nc}(i,j) = \begin{cases} 
0 & \text{if } i \neq j \\
0 & \text{if } d^B_n \cdot n^B \geq g^B \\
1 & \text{if } d^B_n \cdot n^B < g^B 
\end{cases}
$$

(5.14)

Thus, the system of equations that needs to be solved can be expressed as:

$$
H = \begin{bmatrix} 
\mathbf{K}_m & \mathbf{C}_n^T \\
\mathbf{C}_n & \mathbf{1}_{nc}
\end{bmatrix} \begin{pmatrix} 
d \\
\mathbf{P}_n
\end{pmatrix} - \begin{pmatrix} 
\mathbf{F}_p \\
\mathbf{g}
\end{pmatrix} = \mathbf{0}
$$

(5.15)

In Equation (5.15), the stiffness matrix $\mathbf{K}_m = \mathbf{K}_m(\rho_0, \rho_1)$ and the nodal force vector $\mathbf{F}_p$ are identical to those presented for a two-material mechanical model in Section 4.3.1.

The topology optimization problem (for a two-material problem) can be stated as:

$$
\begin{align*}
\text{minimize}_{\rho_0, \rho_1} & \quad c_0(\rho_0, \rho_1, d) \\
\text{subject to} & \quad H(\rho_0, \rho_1, d, \mathbf{P}_n) = 0 \\
& \quad c_i(\rho_0, \rho_1) \leq 0, \; i = 1, \ldots, l.
\end{align*}
$$

(5.16)

To simplify the notation, the vector of design variables is noted $\tilde{\mathbf{\rho}} = (\rho_0, \rho_1)$ and the solutions of the governing equations: $\mathbf{x} = (d, \mathbf{P}_n)$. Solving the system of equations $H(\tilde{\mathbf{\rho}}, \mathbf{x})$ is equivalent to minimizing the potential energy of the system. The Newton’s method corresponds to an optimization problem, where the direction of descent is calculated at each iteration. It is sometimes referred to as a quasi-Newton’s method, since there is no need to calculate second order derivatives [33]. The step size is given by the Armijo-line search. The Newton-Armijo algorithm used in this
thesis is derived from the Matlab file \texttt{nsold.m} by Kelly [39]. This algorithm computes the Newton’s direction of descent by using Gaussian elimination, performed using an LU factorization of the Jacobians. The development of this algorithm is beyond the scope of this thesis, though the interested reader is referred to Kelly [39]. An important note however, is that the convergence can fail for ill-conditioned Jacobians or in the presence of singularities.

The direction of search for the Newton’s method can be written as:

$$z = (z_d, z_n)^T = -[J(x)]^{-1} H$$

Therefore, the Jacobian $J$ is defined as $H = J(x)x$. In this special case of the mechanical-only system, the Jacobian is independent of the field variables: $J(x) = J$. The expression of the Jacobian can be directly derived from Equation (5.15).

$$J = egin{bmatrix} K_m & C_n^T \\ C_n & -1_{nc} \end{bmatrix}$$

Once the search direction is determined, an Armijo line search determines the step size $\delta$. This approximate line search technique ensures that the step size significantly reduces the objective function ($H$) at each iteration $k$ such that

$$H(x_k + \delta z_k) \leq \gamma \delta J z_k$$

where $0 \leq \gamma \leq 1$. See Appendices 7.4 for more information on the Armijo line search. The iterative process is repeated until a stopping criterion is met. The stopping criterion is defined as:

$$||H|| < \text{Abstol} + \text{Reltol} ||H||$$

where the absolute tolerance Abstol and the relative tolerance Reltol are small quantities ($\sim 10^{-6}$). Once this stopping criterion is met, the solution $x$ is determined. The optimization continues as described in Section 2.1.3 with the necessary modifications for a two-material model. In particular, the adjoint method is unchanged, and the nodal forces $P_n$ do not appear in the sensitivity calcu-
lations when optimizing for the mechanical compliance. Thus, the mechanical compliance is given by:

\[ c_0(\tilde{\rho}) = x^T \begin{bmatrix} K_m & 0 \\ 0 & 0 \end{bmatrix} x \]

\[ = \{ F_p - C_n^T P_n, 0 \} x \tag{5.21} \]

which is equivalent to:

\[ c_0(\tilde{\rho}) = d^T K_m d \]

\[ = (F_p - C_n^T P_n)^T d \tag{5.22} \]

Therefore, the Lagrangian is given by for \( i = \{1, 2\} \):

\[ \frac{\partial L}{\partial \rho_i^e} (\rho_0^e, \rho_1^e) = \frac{\partial c_0}{\partial \rho_i^e} + \lambda_m^T \left( \frac{\partial K_m}{\partial \rho_i^e} d \right) \tag{5.23} \]

At this point, the design variables \( \tilde{\rho} \) are updated using the method of moving asymptotes (MMA). If the change in the design variables is too large, the governing equations are solved again, and the process starts over. Therefore, the solution of the governing equations via the Newton’s method constitutes a nested iterative scheme within the optimization loop. The flowchart of the algorithm is summarized in Figure 5.3.

Using the Newton’s method iteratively to search for a solution of the governing equations hopefully converges. Obviously, failure to converge would lead to inaccurate results or worse, crash the algorithm. If the problem is well posed, this should not happen. Of course, the boundary conditions have to be carefully stated, and an important aspect concerns the starting guess for the solutions of the governing equations. Indeed, the starting point for the Newton’s search is very important for the algorithm to converge towards the correct solution. Poor starting guesses for the solutions can result in longer computations at best, and convergence failure or non-feasible solutions at worst. Several starting guesses were tried. If no best values can be generalized for the initial guess, some empirical rules emerged from multiple tests on various configurations. In particular, if the initial guess for the solution does not respect the following inequalities, the algorithm might not converge to the correct solution, and sometimes does not converge at all:
Figure 5.3: Flowchart of the algorithm
Figure 5.4: The boundary conditions and loadings on the design domain can provide an indication for a good starting guess for the field variables. (a) Case 1: No contact is expected in this situation. (b) Case 2: Contact is expected.

1. If the body is expected not to make contact with the rigid support, such as in the configuration of Figure 5.4(a), the initial guess should satisfy:

\[ P_n = 0 \]
\[ C_n^A d - g^A > 0 \]  \hspace{1cm} (5.24)

2. If the body is expected to be in contact with the rigid support, such as in the configuration of Figure 5.4(b), the initial guess should satisfy:

\[ P_n > 0 \]
\[ C_n^A d - g^A = 0 \]  \hspace{1cm} (5.25)

After each update of the design variables via MMA, the governing equations have to be solved again since the design variables change. One might think that the last solution \( x_k \) could be a new good initial guess for the Newton’s method instead of the original guess \( x_0 \). However, the design variables \( \tilde{\rho} \) have changed and the new starting guess might not respect the empiric rules mentioned above anymore, at least locally. Therefore, the initial guess \( x_0 \) is kept the same all through the topology optimization.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulus of Elasticity of Material 1</td>
<td>$E_1$</td>
<td>2</td>
</tr>
<tr>
<td>Modulus of Elasticity of Material 2</td>
<td>$E_2$</td>
<td>2</td>
</tr>
<tr>
<td>Poisson’s ratio</td>
<td>$\nu$</td>
<td>0.3</td>
</tr>
<tr>
<td>Volume Fraction of void</td>
<td>$1 - f_0$</td>
<td>0.5</td>
</tr>
<tr>
<td>Volume Fraction of material 1</td>
<td>$f_0 \times f_1$</td>
<td>0.2</td>
</tr>
<tr>
<td>Volume Fraction of material 2</td>
<td>$f_0 \times (1 - f_1)$</td>
<td>0.3</td>
</tr>
<tr>
<td>Filter’s radius</td>
<td>$r_{\text{min}}$</td>
<td>1.2</td>
</tr>
<tr>
<td>Mesh resolution</td>
<td>$N_x \times N_y$</td>
<td>80x40</td>
</tr>
</tbody>
</table>

Table 5.1: Standard values for the parameters in the simulations for the structural compliance minimization problem for a two-material approach using a mechanical contact model.

5.2.3 Tests and Validation

This model was tested on simple mechanical compliance problem and the results compared to previous problems using the one-material and two-material mechanical optimization approaches. The objective was to determine whether this contact approach could reproduce the results obtained by simple algorithms.

This contact model was originally developed for a 1-material model. 2-material results are used here to demonstrate the ability to generate topologies similar to those presented in previous sections. The same standard values, repeated in Table 5.1, are used again for the following simulations. The colors of each material in the generated topologies are also the same: red for the material with the higher modulus of elasticity, blue for the other material.

The optimal designs obtained in Section 2.1 and 4.3.1 were exactly reproduced using the contact model, with one example shown in Figure 5.5. The boundary conditions have been slightly modified, so that the vertical displacement of the bottom right corner node of the domain is constrained: a rigid support is placed at a distance $g = 0$ from the initial position of the domain. As the domain is forced downwards under the forcing in the top left corner, the rigid support constrains the vertical displacement of the node in contact to zero.
Figure 5.5: Topology optimization of a two-material mechanical model with contact at the bottom right corner. The parameters and results are the same as Figure 4.1. (a) Boundary conditions. (b) Undeformed optimal topology for $E_1 = 2$ and $E_2 = 1$. Colors: white is associated with void elements, red with material 1 and blue with material 2.

The relative difference in the objective function between the simpler model and the contact model is in general less than 0.05%. The behaviors of the objective function and the density change are also very similar, as depicted in Figure 5.6. In particular, the behavior of the algorithm is identical through the first 50 iterations of the optimization, and slight differences appear around the optimum value.

These slight differences are most likely explained by the technique employed to solve the governing equations. The field variables present infinitesimal differences from the solutions provided by direct methods that arise from the Newton’s Method. Another advantage of using the Newton’s search method is to have access to the value of the nodal forces at the contact interface. In the
example shown in Figure 5.5, the normal contact force at the bottom right corner is equal to the opposite force exerted at the top left corner. This result is not surprising given the boundary conditions on the domain. In fact, the maximum deviation observed for the value of $P_n^A$ at this node is: $P_n^A = F \pm 10^{-11}$. Overall, the results of the contact model and the classical non-contact model are essentially equivalent.

Several additional configurations were tested. As shown in Figure 5.7, different locations for the support were chosen. In this particular case, the gap with the support on the bottom right corner does not seem to affect the optimal topology of the undeformed structure (Figure 5.7(c)), as well as the final mechanical compliance value, so long as contact is made. However, the vertical displacements and the deformed shape of the optimal topology are clearly different (Figure 5.7(d) and (e)). For both cases, the nodal forces at the contact interface satisfy the equality $P_n^A = F/2 \pm 10^{-11}$.

Claiming that the optimum is independent from the distance at which the support is located would be a hasty conclusion. Indeed, Figure 5.8 indicates a clear difference of topology between: a “balanced” situation (Figure 5.8(a)) and an “unbalanced” situation (Figure 5.8(b)). The configuration where the supports are level has a final compliance value $c_0 = 15.73$, while the optimal compliance for the “unbalanced” configuration is slightly higher ($c_0 = 16.6$).

The nodal forces at the contact boundary can be quite different according to the distribution of the material and the forcing on the structure. On Figure 5.9, the forces are quasi-null at the nodes of void elements and much more important at the edges of the rigid support. Finally, summing all the contact forces at the bottom, the equality $\sum (P_n^A) = 3F$ is respected. The access to the values of these contact forces will be later used for the implementation of a TCR model, which depends on the pressure values at the interface.

All these tests were generated by choosing carefully the initial guesses for the field variables when solving the governing equations with the Newton’s method. Figure 5.10 shows the same configuration as in Figure 5.8(a) with the three forces pulling upwards instead of down. Also the vertical and horizontal displacements are constrained at the left and right bottom nodes of the domain, to ensure that a feasible solution exists for the FEM problem. It is normally expect no
contact to happen at the bottom boundary. If the initial guesses for the design variable are kept the same, a physically non realizable solution is obtained for the governing equations. As a result, the solution converges towards a design close to the one presented in Figure 5.8(b). If the initial guess respects the conditions stated in Section 5.2.2, realistic solutions are obtained. In fact, this solution corresponds to the one observed for a 2-material mechanical model without contact. Also, the contact forces are equal to exactly zero in this configuration.

The following tests demonstrated the ability of this contact model to converge towards satisfying optima, provided that the boundary conditions and initial guess for the governing equations are chosen carefully. This model can now be extended to implement thermal contact at the boundary.
Figure 5.6: Behavior of the convergence in the minimization of compliance problem of Figure 5.5. (a) Objective function value at each iteration. (b) Density changes in the domain. Red: Contact model. Blue: Non contact model.
Figure 5.7: Influence of the distance between the supports and the initial "free" domain. For two different boundary conditions (a) and (b), the undeformed optimal topologies are identical (c), though the deformed shapes and displacement values are different (d) and (e). Displacement contours are given for non-dimensional displacement values.
Figure 5.8: Influence of the distance between the supports. For 2 different boundary conditions (a) and (b), the undeformed optimal topologies (respectively (c) and (d)) and deformed optimal topologies (respectively (e) and (f)) present clear dissimilarities. Their vertical displacement contours (non-dimensional) over the domain (respectively (g) and (h)) affect the design optima.
Figure 5.9: Values of the normal contact forces at each node at the contact boundary. (a) Normal contact force distribution at steady state for the optimal design in Figure 5.8(c). (b) Normal contact force distribution at steady state for the optimal design in Figure 5.8(d).
Figure 5.10: Optimum design when no contact is made for the boundary conditions in (a). (b) Undeformed optimum. (c) Deformed optimum. (d) Profile of the vertical displacement in the structure (non-dimensional).
5.3 Implementation of a Thermal Contact Model

5.3.1 Modeling the Contact Pressure

In the case of contact, the mechanical formulation of the contact presented in Section 5.2 provides the value for the nodal forces at the interface. Using these values, it is possible to estimate the pressure along the contact interface. In particular, those pressure values could be used further in the implementation of the thermal contact resistance (TCR).

Several models could be assumed for the pressure distribution. Assuming a polynomial pressure distribution over an element can provide relatively accurate results, but can be quite complex and hard to implement. In this thesis, the formulation of the TCR, which depends on the pressure at the interface, makes it difficult to deal with complex pressure distributions. For this reason, an element-wise uniform pressure is assumed in order to simplify implementation of contact. In fact, the normal nodal forces $P_n$ represent the resulting force of the pressure exercised on half of two-neighboring elements, as described in Figure 5.12. For a normal contact force $P_i$ at a node $i$, the pressure $p^*$ along the neighboring elements $e$ and $e - 1$ is equal to:

$$p^* = \frac{P_i}{t \frac{2a}{2}}$$

(5.26)

Figure 5.11: Piecewise uniform pressure distribution at the interface in the FEM.
where $t$ is the thickness of the elements and $a$ the semi-length of an element. Therefore, following the description in Figure 5.12, the element $e$ sees a discontinuity in pressure at the interface: $p_i^* = P_i/(t \cdot 2a)$ on its left half bottom face and $p_{i+1}^* = P_{i+1}/(t \cdot 2a)$ on his right half bottom face. This approximation, however coarse, can be eased by using a finer mesh. Also, this formulation simplifies the formulation of the TCR in the next section.

5.3.2 Thermal Contact Resistance (TCR) Model

The implementation of a TCR is the last feature to be added to the model. As discussed earlier in Section 1.1.3, the TCR is characterized by a discontinuity in temperature at the contact interface such that:

$$q = \frac{1}{TCR} \times (T - T_{sink})$$

(5.27)

with $T_{sink}$ the temperature of the rigid support with which the object is in contact, and $T$ the temperature at the body’s surface. In this work, $T_{sink}$ is a known, prescribed temperature.

The heat transfer $q$ flowing out of the domain depends on several factors. The pressure $p^*$ at the interface is the main factor that can be influenced by the design shape of the C3M. Also, the TCR depends on material properties such as the material hardness $H_a$, the surface roughness or the harmonic conductivity of the materials\(^1\). Many models have been explored; most of them are

\(^1\)The harmonic mean conductivity at the contact interface usually consists of an average of the conductivities of
empiric and determined through experimental analysis [5]. Depending on the type of material and pressure ranges, one model might prove better than another, though no model seems to stand out. Analytical formulation are available for very simple case problems [40], but usually suppose the knowledge of many parameters (temperature discontinuity, heat flux at the interface, etc.). Since the temperature is a field variable and the heat flux at the interface is likely to depend on the topology, a closed form solution would be a preferred choice for the implementation of the TCR in this case. Following Stavely’s approach [3] for the design of the C3Ms, the TCR model adopted has been proposed by Antonetti [41] and can be stated in the following simplified form:

\[
\frac{1}{TCR(p^*)} = h_c = h_{c0} \left( \frac{p^*}{H_a} \right)^\beta
\]  

(5.28)

where \( h_{c0} \) is the nominal thermal contact conductance at pressure \( p^* = H_a \) and \( \beta \in [0, 1] \) a coefficient usually equal to 0.95. This empirical relation was derived from experiments and proved reliable for metal-to-metal interfaces. This formulation is convenient for the problem studied in this thesis, since it constitutes a closed form expression and it relies on only one variable (pressure). Using the pressure formulation of Section 5.3.1 and Equation (2.25), the heat flux at the interface is treated as a boundary condition such that, for an element \( e \) at the boundary with nodes \( i \) and \( i + 1 \) at the boundary:

\[
K_{th,e}(\beta_t)T_e = -\frac{ta}{2} \begin{bmatrix} \bar{q}_i \\ \bar{q}_{i+1} \\ 0 \\ 0 \end{bmatrix}
\]  

(5.29)

where \( T_e = \{T_i, T_{i+1}, T_{i+2}, T_{i+3}\}^T \) is the element temperature vector. Also, using Equation (5.27) the expression of the heat flux \( \bar{q} \) can be replaced so that the boundary conditions at the interface become:

\[
\text{the two material in contact such as } k^* = \frac{k_1 k_2}{(k_1 + k_2)}
\]
where $1/TCR$ is known through equation (5.28). Finally, Equation (5.30) can be written as:

$$\bar{K}_{th,e} T_e = F_{TCR,e}$$

(5.31)

where

$$\bar{K}_{th,e} = K_{th,e} + \frac{ta}{2 TCR} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} T_e$$

and

$$F_{TCR,e} = \frac{ta}{2 TCR} \begin{bmatrix} T_{sink} \\ T_{sink} \\ 0 \\ 0 \end{bmatrix}$$

(5.32)

The global conductivity matrix is obtained by summing over the element conductivity matrices:

$$\bar{K}_{th} T = F_{th}$$

(5.33)

where $F_{th} = F_s + F_{TCR}$ contains the other boundary conditions of the thermal system ($F_s$). Also, since $F_{TCR} = F_{TCR}(P_n)$, the conductivity matrix depends on the normal contact forces ($P_n$). This non-linearity justifies once more the use of the Newton’s method to solve the system of governing equations.

5.3.3 Thermal Contact Implementation for a Thermo-mechanical System

Because of the contact conditions, the governing equations of the thermal and mechanical systems cannot be solved separately such as in Chapter 3. The mechanical system is coupled with the thermal system via the TCR’s pressure dependence and the coefficients of thermal expansion. The number of variables to solve for is denoted as $nd$ displacements ($d$), $nc$ normal contact forces ($P_n$).
and \( nth \) temperatures \((T)\) \(^2\). Following this notation, the quasi-static system of equations to be solved can be rearranged using equations (5.15) and (5.33):

\[
H = \begin{bmatrix}
K_m & C_n^T & -C_{thm} \\
\bar{C}_n & \bar{1}_{nc} & 0_{nc \times nth} \\
0_{nth \times nd} & 0_{nth \times nc} & \bar{K}_{th}
\end{bmatrix}
\begin{bmatrix}
d \\\nP_n \\
T
\end{bmatrix}
- \begin{bmatrix}
F_p \\
\bar{g} \\
F_{th}
\end{bmatrix}
= 0 \tag{5.34}
\]

where \( 0 \) represent all-zero matrices. \( C_{thm} \) is the matrix of size \( nd \times nth \) that contains the thermal expansion terms such that

\[
C_{thm} T = F_{\epsilon} \tag{5.35}
\]

with \( F_{\epsilon} \) defined in Equation (3.6) in Section 3.1.

This system of equation is non-linear because of the TCR expression: \( \tilde{K}_{th} = \tilde{K}_{th}(P_n) \). Therefore, using the notation \( x = (d, P_n, T) \), the Jacobian used for the Newton’s search is modified:

\[
J(x) = \begin{bmatrix}
K_m & C_n^T & -C_{thm} \\
\bar{C}_n & \bar{1}_{nc} & 0_{nc \times nth} \\
0_{nth \times nd} & \tilde{K}_{th}' & \tilde{K}_{th}
\end{bmatrix} \tag{5.36}
\]

\( \tilde{K}_{th}' \) contains the derivatives of \( \tilde{K}_{th} \) and \( F_{th} \) with respect to \( P_n \). The Jacobian now depends on the variables \( x \), and is updated at each iteration of the Newton-Armijo search.

### 5.4 Tests and Validation

Before testing the algorithm using the thermal contact model, this model successfully reproduced the results presented in Chapter 4 and Section 5.2.3. The Newton’s method proved accurate in solving the governing equations, even with the inclusion of the material thermal expansion.

The TCR model was tested on simple thermal problem, where the pressure was assumed uniform at the interface. The test was based on a thermal compliance optimization problem. The expected behavior was observed: for a high value of the thermal contact conductance (TCC) \( h_{c0} \), the contact was considered ideal, and the boundary conditions were close to a prescribed temper-

\(^2\text{Note: } nd = 2 \times nth \text{ in this problem. } nc \text{ depends on the contact interface location.} \)
Figure 5.13: Boundary conditions used for the test. (a) Mechanical boundary conditions. (b) Thermal boundary conditions the middle area at the bottom corresponds to an insulated face. The high conductive layer introduced in Section 3.4.2 is solid non-designed.

ature boundary condition. When the TCC was low, the discontinuity in temperature with $T_{\text{sink}}$ was increased.

However, optimization issues were encountered once the TCR model was incorporated in the complete thermo-mechanical model. The mechanical compliance was used for the tests. The optimal topologies in Figure 5.14 were obtained using the same values as the parameters as in Table 4.4, with a heat flux of $q = 0.5$, and for the boundary conditions of Figure 5.13. For a small TCR (high TCC), the optimal topology does not present any particular intermediate densities, and converges towards nearly the same optimum as for a prescribed temperature condition (Figure 5.14(a)). However, if the TCR is significant, the algorithm does not converge as smoothly (Figure 5.14(b)): areas composed of intermediate densities appear near the contact boundary and the volume fraction of the materials are not respected.

In Figures 5.15 and 5.16, the nodal temperatures at the bottom do not seem to coincide with the normal force distribution. According to the formulation of the TCR model, at nodes where the normal force is high, the temperature at this node should be close to the $T_{\text{sink}} = 0$ value (cold). These expectations are not met, since void areas present the lowest temperatures. Also in Figure 5.16, the sum of the normal forces do not equal the forcing values on top ($\sum P_n = 3F$) and the
normal forces present clear discontinuities near the edge of the domain.

If the temperatures at the contact boundary become too hot, huge deviations in the thermal strains can lead to a failure in the solution of the governing equations. Therefore, the choice of the values for the parameters have to be balanced accordingly for the algorithm to converge. For this reason, the use real dimensional quantities can be difficult, since ill-conditioned matrices restrain convergence.

Extended research on the thermal contact explored the possibility of interpolating the values of the thermal contact conductance \((h_c)\) following a SIMP method. Indeed, the thermal contact conductance depends on the harmonic conductivity of the two materials in contact. The TCR would then depend on the conductivity of the elements in contact. A possible formulation could take the form:

\[
\frac{1}{TCR}(\rho_0^6, \rho_1^6) = \hat{h}_{c,0} \left( \frac{p^*}{H_a} \right)^\beta (\rho_0^6)^{p_0}((\rho_1^6)^{p_1}k_1 + (1 - \rho_1^6)^{p_1}k_2)\frac{k_{sink}}{\max(k_1, k_2) + k_{sink}} \tag{5.37}
\]

Initial tests did not prove conclusive: no improvement on the convergence was obtain, and similar problems as those mentioned above were observed for high input flux conditions. Extended research would be necessary to make this model usable for high heat input values.
Figure 5.15: Forces and temperature at the nodes at the bottom boundary for the design in Figure 5.14(a). (a) Normal contact forces. (b) Temperatures
Figure 5.16: Forces and temperature at the nodes at the bottom boundary for the design in Figure 5.14(b).  (a) Normal contact forces.  (b) Temperatures
Chapter 6

Design Optimization of Cellular Contact-Aided Compliant Mechanisms

Using the algorithm presented in the previous sections, possible optimal topologies have been generated for the design of the cellular contact-aided compliant mechanisms (C3Ms). Rather than aiming at designing complete detailed structures, these optimal topologies are intended to highlight possible features that could be included in the C3M designs. The mechanical loadings correspond to the thermal strain from the thermal expansion of the materials. A poor choice of the objective function and/or the boundary conditions can lead to failure in the convergence.

6.1 Design Considerations

The design of a C3M depends on the reference temperature $T_{ref}$, at which the thermal strains are considered null. As described by Stavely [3], in an ideal case, the C3M should be designed such that a quasi-zero pressure contact is made at $T_{ref}$. Therefore, a drop in temperature from the initial $T_{ref}$ results in breaking contact, while an increase in temperature from $T_{ref}$ enhances the contact pressure at the interface. In reality, the thermal displacements are expected to be too small in the temperature range considered ($290K \pm 30K$). Consequently, the initial design of the C3M should consider to be built with an initial non-zero nominal pressure at the interface at
the reference temperature $T_{\text{ref}}$. Under varying temperature conditions, the pressure decreases (in a low conductivity mode) or increases (high conductivity mode) to modify the thermal contact resistance. This concept of a “thermal valve”, where the conduction heat transfer is continuously adjusted with gradual changes in pressure, would best describe the behavior of the C3Ms in this application.

From a topology optimization stand-point, accurate representation require dimensional parameters and a good model for the thermal contact in order to produce efficient, reliable designs that follow this “thermal valve” concept. In this thesis, the contact model is not mature enough to allow such accuracy, and finite element solutions should be implemented using more sophisticated commercial packages. Nonetheless, the present algorithm is used to provide design guidance, and highlights potential patterns or features characteristic of an optimal design for this application. Although quantifying the results remains difficult at this stage, a key objective at this stage is to evaluate the importance of the parameters on a possible optimum design.

6.2 How to Choose a “Good” Objective Function?

In a strict mathematical sense, the objective function should reflect a global criterion of the structure, as explained by Bendsøe and Sigmund [7]: “the optimization should be global criteria, e.g. compliance, volume, average stress, etc., so that the effectiveness of the dual optimizers can be maintained by treating problems with a moderate number of constraints”. They also emphasized that the choices of objective functions and constraints require “many experimentations and tries before arriving to a “good” formulation which makes physical sense” [7]. Failure to provide a satisfactory objective function translates into poor design optima at best, or the algorithm fails to converge at worse.

Compliance optimization problems are quite popular and ensure a relatively robust approach. However, the goal is to maintain the electronics within an allowable temperature range. The challenge here consists of identifying the best formulation for the objective function to describe this goal. Stavely pointed out [3] that the goal for the design optimization of the C3Ms could consist of
maximizing the effective conductivity ratio \( k_{\text{high}}^* / k_{\text{low}}^* \) of the cells: when the top side of the domain is hot, the objective is to maximize the effective conductivity \( k_{\text{high}}^* \) in order to enhance the heat transfer at the interface. Conversely, in the case where the top boundary is below a set temperature, the objective would be to minimize the heat transfer at the interface, which comes down to minimizing the effective conductivity \( k_{\text{low}}^* \). In fact, an ideal objective could be to maximize the conduction heat transfer between the top and bottom boundaries of the design domain for high temperature conditions, and minimize it for low temperature conditions.

The challenge is to find a set of objective functions that reflect both the design objective of the C3Ms for two different load cases and the requirements for the algorithm to convergence.

6.3 Loading & Boundary Conditions

The topology is subject to change depending on the resolution of the mesh and the size of the domain. The boundary conditions reproduce the C3M environment both thermally and mechanically. The optimization was performed on a square domain with boundary conditions similar to those of the problems explored in previous sections.

The general approach for the mechanical boundary conditions is given in Figure 6.1(a). The horizontal displacements on the sides and all displacements of the top nodes of the domain are constrained and null. Unilateral contact is considered at the bottom of the domain, which constrains the vertical displacement in the case of contact. Note that no forcing boundary conditions is exerted on the domain. The thermal expansion of the materials is the unique “source” of strains in the domain. Therefore, the deformations directly depend on the choice of the thermal boundary conditions.

Multiple choices of thermal boundary conditions could represent the C3M environment. However, the topology of the structure is built in response to the loads (mechanical and thermal) exerted on the domain. In the absence of mechanical forces on the domain, thermal loadings become essential to trigger both thermal and mechanical response of the structure. Following Stavely’s approach [3], a heat flux input flows into domain from the top boundary, while the sides of the domain are insulated and thermal contact is made at the bottom, as described in Figure 6.1(b). Also, passive,
Figure 6.1: Boundary conditions considered for the C3M design. (a) Mechanical boundary conditions. (b) Thermal boundary conditions.

solid-designed elements with high conductivity are placed on top of the domain to simulate a conductive face-sheet.

These boundary conditions assume that contact is always performed at the boundary, and that the heat flux flowing out of the domain depends on the normal contact forces. This configuration is likely to happen in a high conductivity mode, for a high heat flux flowing inside the domain: the materials are expected to expand and, given the displacement constraints, the structure is likely to be in contact with the bottom boundary. It corresponds to a situation where the electronics on the exterior of the spacecraft reach high temperatures, and will be further referred to as a "hot" case.

In the case of a low heat input however, the materials expansion might not be sufficient to generate contact or significant pressure during the optimization process. This corresponds to a situation where the temperature of the electronics is low, and will be further referred to as a "cold" case. In this case, other boundary conditions must be defined for the bottom side of the domain to maintain the existence of a steady-state solution for the governing equations. The choice of the boundary conditions can be made as follows:
• If no contact is made at a node, the boundary condition at this node corresponds to an insulated boundary condition.

• If the entire set of nodes at the bottom of the domain do not establish contact with the support boundary, an insulated boundary condition on the entire face is obtained using the above rule. As a result, no steady-state solution exists since heat is brought into the domain from the top. Topology optimization in the present work does not handle transient problems. Instead, the temperature at the bottom nodes is chosen to be prescribed, to ensure that a steady-state solution exists.

The prescribed temperature could be defined as a constant, uniform temperature distribution at the bottom boundary.

6.4 Hot Case

In this section, the structure is assumed in the configuration of a high conductivity mode: the nodal temperatures globally are higher than the reference temperature $T_{ref}$. The materials expansion is important, and high contact pressure is achieved at the bottom. The value of the nominal thermal contact conductance ($h_c$) is manually adjusted to simulate the so-called “quality” of the contact, namely the finish quality of the surface and the material’s hardness.

In this hot case, a good objective function to study is the thermal compliance of the structure, presented in Section 2.2 and formulated in Equation (2.24): in the case of the boundary conditions presented in Figure 6.1, this objective consists of maximizing the heat transfer from the top to the bottom boundary. In this special case, results showed that this objective function is equivalent to minimizing the average temperature at the nodes of the top elements.

6.4.1 Results using a Two-Material Thermal Model

To avoid possible misleading results obtained with a thermal contact model, the optimization was performed on a single two-material thermal model such as presented in Section 4.3.2. In this
| Conductivity of material 1 (red) | $k_1$ | 2 |
| Conductivity of material 2 (blue) | $k_2$ | 1 |
| Reference Temperature | $T_{ref}$ | 0 |
| Sink Temperature | $T_{sink}$ | 10 |
| Volume Fraction of void | $1 - f_0$ | 0.6 |
| Volume Fraction of material 1 | $f_0 \times f_1$ | 0.2 |
| Volume Fraction of material 2 | $f_0 \times (1 - f_1)$ | 0.2 |
| Filter’s radius | $r_{min}$ | 1.2 |
| Mesh resolution | $N_x \times N_y$ | 50x50 |

Table 6.1: Standard values for the parameters in the simulations for the C3M designs.

section, the mechanical behavior of the structure is not studied, and a uniform contact pressure was assumed at the interface. The value of the TCR was adjusted to study its effect on a thermal based optimum design. The standard values of the parameters used in these simulations are given in table 6.1 and the boundary conditions are given in Figure 6.1(b).

This problem is highly symmetrical, and several optima are likely to be close in the space of solutions for the design variables. In this case, the initial guess could be important. Again in this approach, the initial guess for the design variables is a uniform distribution, in which every element contains a portion of void, and a portion of each material. A rounded structure is obtained in each topology presented in Figure 6.2, with the highly conductive material (in red) placed in the center of the design. Also, a very good contact (TCR=1) results in more direct heat conductive paths between the two boundaries than the other configurations. This case can be referred to as an “ideal” contact condition, where the temperature boundary condition corresponds to a prescribed temperature $T \approx T_{sink}$, as the temperature profile within the structure suggests (Figure 6.2(d)). Higher values in TCR tend to increase the average temperature in the structure and force the placement of material near the contact boundary (Figure 6.2(b),(c) and their temperature profile in Figure 6.2(e),(f) respectively).

In the case where the volume fraction of each material is unconstrained and only the void fraction is constrained, logically, the material with the lowest conductivity does not appear in the
Figure 6.2: Minimum thermal compliance topologies using a two-material thermal model depending on the change in the thermal contact resistance (TCR) at the bottom boundary and the corresponding temperature distribution within the solid structure. Colors: white is associated with void elements, red with material 1 (high conductivity) and blue with material 2 (low conductivity).

final designs (Figure 6.3). Similarly to the fully constrained case, for a perfect contact, direct conductive heat paths are obtained, as depicted in Figure 6.3(a). Higher thermal resistance leads the algorithm to distribute more material near the heat flux boundaries.

6.4.2 Results using a Two-material Thermo-mechanical Model with Contact

In this section, the topologies presented were obtained using a thermo-mechanical model with contact. The only mechanical loading on the structure comes from the expansion of the structure. Under hot temperature conditions (above $T_{ref}$), the structure expands under the effects of thermal expansion, but is constrained within the space defined by the boundary conditions presented in Figure 6.1(a). Therefore, contact is made at the bottom boundary. The thermal boundary conditions used are given in Figure 6.1. Again, the objective was the minimization of the thermal compliance.
Figure 6.3: Minimum thermal compliance topologies using a two-material thermal model depending on the change in the thermal contact resistance (TCR) at the bottom boundary. The volume fractions of the materials are unconstrained.

Figure 6.4: Undeformed optimal designs for a thermal compliance optimization problem using different values for the TCR.

of the structure. The values of the parameters used in the simulations are given in Table 6.2.

Figure 6.4 show that for high values of the thermal contact resistance (or low values for $h_{c0}$), the high conductivity material is placed near the heat flux boundary. The results are similar to those obtained using the thermal model in Section 6.4.1. Also, when the volume fraction of each material is not constrained, the material with the the higher conductivity is chosen, and the other abandoned, even for a high mismatch in the CTEs. This result suggests that the objective function does not take into consideration the effect of the thermal contact condition at the bottom, and therefore needs to be redefined.
Table 6.2: Standard values for the parameters in the simulations for a C3M design using a thermo-mechanical model with contact.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulus of elasticity of material 1</td>
<td>$E_1$</td>
</tr>
<tr>
<td>Modulus of elasticity of material 2</td>
<td>$E_2$</td>
</tr>
<tr>
<td>Conductivity of material 1</td>
<td>$k_1$</td>
</tr>
<tr>
<td>Conductivity of material 2</td>
<td>$k_2$</td>
</tr>
<tr>
<td>CTE of material 1</td>
<td>$\alpha_1$</td>
</tr>
<tr>
<td>CTE of material 2</td>
<td>$\alpha_2$</td>
</tr>
<tr>
<td>Poisson’s ratio</td>
<td>$\nu$</td>
</tr>
<tr>
<td>Sink temperature</td>
<td>$T_{sink}$</td>
</tr>
<tr>
<td>Reference temperature</td>
<td>$T_{ref}$</td>
</tr>
<tr>
<td>Heat flux input</td>
<td>$q$</td>
</tr>
<tr>
<td>Volume Fraction of void</td>
<td>$1 - f_0$</td>
</tr>
<tr>
<td>Volume Fraction of material 1</td>
<td>$f_0 \times f_1$</td>
</tr>
<tr>
<td>Volume Fraction of material 2</td>
<td>$f_0 \times (1 - f_1)$</td>
</tr>
<tr>
<td>Filter’s radius</td>
<td>$r_{min}$</td>
</tr>
<tr>
<td>Mesh resolution</td>
<td>$N_x \times N_y$</td>
</tr>
</tbody>
</table>

6.5 Cold Case

In this situation, no contact or low pressure exists at the interface with the thermal bus, and the temperature within the structure is below the reference temperature $T_{ref}$. The objective could be to isolate the electronics from the thermal bus. The challenge here is to define an objective function which is consistent with the topology optimization process. Most of the formulations experimented focused on the maximization and/or minimization of nodal temperatures in some specified areas of the domain. None of the configurations tested converged: void areas are usually introduced near the top and/or bottom boundaries because of their low conductivity value, leading to large deviations in temperatures over the domain.

Also, the final structure should be optimal considering both “hot” and “cold cases”. This sug-
gests that a multiple-load problem should be considered in the design of the C3Ms, and possibly a multi-objective topology optimization problem. However, the absence of a good formulation for both the boundary conditions and the objective function in the “cold case” makes this multi-case approach difficult to implement at this stage.
Chapter 7

Conclusions & Future Work

7.1 Summary of Key Results

7.1.1 Development of a Thermo-mechanical Topology Optimization Algorithm

Based on a one-material topology optimization problem introduced by Sigmund [9], a finite element thermal model was implemented and successfully reproduced results available in the literature. By coupling both mechanical and thermal systems of equations, a thermo-mechanical topology optimization algorithm was developed to evaluate the effect of material thermal expansion on optimal designs. Inspired by the work of Sigmund [30], this model was tested on mechanical compliance problems by using a uniform temperature offset from a reference temperature, at which the thermal strains are null. Convincing results were obtained by comparison with available work in the literature.

This thermo-mechanical algorithm was tested using thermal heat input boundary conditions, and was shown to exhibit large deviations in temperatures in void elements when no treatment applied. It was shown that this phenomenon could lead to convergence and topology problems. A high conductive layer at the heat flux boundary was introduced to alleviate these hot-spots problems and help propagate the heat further into the structure. This implementation improved the optimal topologies along with the convergence behavior. This high conductive layer can be seen as face-sheet that could be incorporated in an eventual sandwich panel design when constructing cellular contact-aided compliant mechanisms (C3Ms) for thermal control.
7.1.2 Analysis of the Behavior of a Two-Material Model

Original one-material algorithms were extended to a two-material approach, where the effect of the parameter values of the two materials, and in particular their ratios, were analyzed. It was shown that in most cases, for constrained volume fractions of the materials, the best material with respect to the objective function was placed in priority in key areas.

When the volume fractions of the material phases were not constrained, the best material with respect to the objective function was chosen, while the other material was left out of the topology. However, for thermo-mechanical problems, tests on the minimization of the mechanical compliance showed that this other material could be placed for some special boundary conditions and load cases. For the tests on the maximization of the structure’s stiffness, the material with the higher modulus of elasticity is usually the preferred choice in most configurations, though special combinations of conductivities and coefficients of thermal expansion values can lead to the placement of the other material in specific locations.

7.1.3 Development and Implementation of a Contact Model

Complete modeling of the C3Ms behavior involve the formulation of a contact model. This was realized by using a unilateral contact model adapted from Strömberg [38]. This method used a Newton’s method combined with an Armijo line search to determine the normal contact forces at each node at the contact interface, and the contact defined by Signorini’s contact conditions. This method proved advantageous because:

- It can solve multi-linear and non-linear governing equations. In this case, the normal contact forces are seen as new variables that need to be determined.

- Values for the normal contact forces are accessible and can be utilized in the development of a thermal contact model.

- Convergence towards a feasible solution is relatively quick, provided a good starting guess and well-posed Jacobian.
Also, it was found that the initial guess for the solution of the governing equation was essential to converge towards a feasible solution. In the optimization loop, using the solution of the previous iteration does not constitutes a good starting guess for the governing equations. Also, this starting guess must satisfy certain conditions to converge towards a satisfying solution and avoid algorithm to crash.

The thermal contact model presented constitutes a new development that could potentially produce novel topologies. Initially, this approach generated satisfactory results when mechanical contact was excluded from the model, and a constant uniform pressure was simulated using a two-material thermal system only. For a complete thermo-mechanical model, tests demonstrated the ability of the model to generate designs for low thermal contact resistance values, effecting a prescribed temperature boundary condition. Suggestions on how to improve the model could reveal useful in future tests. However, when no other mechanical loading is applied to the structure other than the thermal expansion of the materials, the structure successfully realized the contact condition.

7.1.4 On the Design of the Cellular Contact-Aided Compliant Mechanisms for Thermal Control

The use of a multi-objective multi-load case approach that could optimize the design for both “hot” and “cold” configurations are limited because of the exigences of the topology optimization algorithm when using a SIMP interpolation scheme combined with primal-dual methods like MMA. This problem is reinforced by the difficult to determine a good objective function for the “cold case”.

Designs were generated for simple thermal compliance model performed using a thermal model and a thermo-mechanical model. The suggested features enhances the heat transfer between the top and bottom boundaries of the domain. More material is placed near the heat flux boundaries in case of poor contact, while straight path are preferable with good contact conditions. These findings needs to be verified by extensive experiments, possibly using higher mesh resolutions.
7.2 Limitations

7.2.1 Assumptions and Approximations

Topology optimization in this context is considered as a strict mathematical tool to suggest possible good designs. Physical interpretations are complicated by the mathematical formulation and in particular the SIMP approach. The possibility for some elements to contain an average of the material properties of each phase does not apply as an accurate description of the real physics in the optimal topology.

Other important assumptions lead to simplifications that weaken the fidelity of comparisons to reality. The absence of internal contact between two different materials might have a significant impact by locally modifying the material properties, especially the conductivities.

The void element properties can affect both the solution of the governing equations and the optimization process, as experienced with heat flux boundary conditions in chapter 2.

However, these assumptions are very convenient, if not essential, to the topology optimization process.

Concerning the contact model, the finite element formulation for the thermal contact resistance (TCR) could be improved. The model chosen based on Antonetti’s work [41], however simple, is difficult to manipulate in the governing equations. A smooth, continuous pressure distribution would be a much more accurate representation of the contact pressure, and by transitivity, of the TCR. Other mechanical aspects of the contacts, such as friction, could have been considered [42], but may not be fundamental at this point for the design of the C3Ms.

Finally, the SIMP formulation used in this thesis did not take into account the Hashin-Shtrikman bounds for a two-material approach, which sets the limits for the achievable isotropic material properties from two linear elastic, isotropic materials [7]. As a result, some of the topologies generated might not be practically realizable. Assuming that the material properties do not depend on the temperature might affect the results too, though extensive dimensional study is required to evaluate this.
7.2.2 Dimensionality

In this study, the dimensionality of the parameters in this study is one of the main issues. Here, the values for the parameters were chosen coarsely, without clearly defined units, in order to assess possible outcomes with the algorithm. Consequently, the results need to be carefully interpreted since no proper dimensional study has been performed, especially for the thermal and mechanical couplings. The difficulty when trying to assign dimensions is that ill-conditioned matrices in the system of equations can lead to optimization failure or the incapacity of the Newton’s method to solve the governing equations. For example, the moduli of elasticity of traditional materials are of the order of several GPa’s, namely $10^9$ Pa. For the mechanical system alone, this means that important forcing terms are necessary for the optimization to numerically converge. Considering a multi-physics approach, especially using the Newton’s method in chapter 5, a solution for the governing equations using dimensional units is unlikely to be found. By introducing characteristic quantities, it should be possible to achieve a non-dimensionalized system of equations, which would respect the algorithm tolerance in terms of quantities’ magnitudes. Nonetheless, in the actual state of the algorithm, the system of equations usually requires unrealistic dimensional forcing. This makes it difficult to evaluate the influence of the thermal expansion of the materials and the couplings between both mechanical and thermal systems.

7.2.3 Computational Efficiency

The current study was performed using the numerical programming language Matlab®. A custom square element formulation has been used instead of a commercial finite element package. If this offers more flexibility and understanding in the implementation of the model, and in particular the contact implementation, the performance of this formulation cannot reach the capabilities of software such as ANSYS® or ABAQUS®. Indeed, each topology provided in this thesis needs to be retreated before attempting any practical experimentation. In particular, a finer mesh could be achieved and some commercial packages include contact models that could be used for this analysis.

The algorithm CPU time mainly depends on the machine and the mesh size. The machine used for these simulations was composed of a standard processor (3.16 GHz) and 4.0 GB of RAM. For
the single physics problems such as in Chapter 2, and a mesh resolution of $80 \times 40$, the optimal design on a well posed problem is obtained after less than 100 iterations, after an average of 30 seconds. When the domain reached around 15000 elements, the calculation times were increased significantly (1 iteration after more than one minute). If the number of elements over the domain is $N = N_x \times N_y$, the number of nodal displacement is $nd = 2(N_x + 1)(N_y + 1)$, and the number of nodal temperatures $nth = (N_x + 1)(N_y + 1)$. The number of equations to be solved after each update of the design variables $\tilde{\rho}$ is equivalent to $3N_x \times N_y$, and the number of elements stored in the system matrices is $\sim 9 \times N^2$. The efficiency of the code is directly proportional to this number. Besides, the contact forces and conditions have to be solved and the MMA updates $2N$ design variables for a two-material problems. More efficient techniques for the optimization could be explored to improve computational efficiency, but this topic is beyond the scope of this thesis. Also, the Matlab algorithms implemented in this thesis could certainly be improved.

### 7.2.4 Optimization Method

The optima obtained using the MMA do not necessarily constitute global optima. The local minima could be different depending on the initial guess for the material distribution over the domain. When the volume of material is constrained, a uniform distribution seemed to offer the best approach to converge towards a reasonable local minimum, and is traditionally used in topology optimization. A highly non-uniform distribution could very well lead the topology to converge towards a different distribution, though this was not tested. For partially constrained problems in two-material approaches, where only volume constraint is the void fraction, the initial guess of the material volume fraction can matter, though no significant differences were seen in the configurations tested. Also, the differences between two local minima can be very small, which increases oscillatory behaviors in the density changes and in the objective function near the optimum value. In particular, the existence of a unique optimum is not guaranteed in some highly symmetrical problems [43]. However, the algorithm eventually met low density change stopping criteria for each simulation.
7.3 Recommendations and Future Work

According to the author, two major future directions emanate from the present work. One focuses mainly on the topology optimization method, and the other focuses on the C3Ms design for thermal control, although these directions are not completely disjoint.

7.3.1 On the Topology Optimization Method

Considering the algorithms developed, the topology optimization can be enhanced. This mostly involves mathematical and computational improvements. The list below summarizes the improvements and development of the algorithm that could be made. In the author’s opinion, the list is ordered by decreasing degree of priority:

1. **Dimensionalizing the quantities**
   
   Topology optimization has been considered from a strict theoretical standpoint, where the dimension of the quantities have not been studied. Real dimensional quantities would provide much more realistic results than the estimations proposed in this study. To avoid instability problems with the topology optimization, a possible solution would be to non-dimensionalize the equations using characteristic quantities.

2. **Perform the FEM on a commercial package**
   
   This constitutes potentially the greatest improvement of the results alongside the first item listed. Some commercial packages do incorporate contact models and modeling thermal contact in finite elements using these packages has been already studied [44]. Despite the usual steep learning curve behind the use of these packages, the improvement on the finite element analysis would be dramatic: finer mesh, more accurate results, more adaptable and less error-prone constructions are a few arguments that would justify this direction.

3. **Implementation of the Hashin-Shtrikman bounds**
   
   The Hashin-Shtrikman bounds would ensure that the optimal topology is realistically feasible when using two-material thermo-mechanical models. How much of an impact this would have on the final results is uncertain. This addition to the code would be relatively computationally cheap and easy, since this is a well documented problem. For this reason, this modification could also be done before all the other items listed here.
4. **Transient problems in topology optimization**

The governing equations in all the approaches in this study have steady-state solutions. However, the dynamic behavior of the structure might be of comparable interest. In particular, thermal conditions can vary quite abruptly in space. The optimal topology for a given set of static situations might differ from that of a dynamic problem. This would lead to quite drastic changes in the governing equations and regime extensive computational resources. Using a finite element package would be preferable for this task.

5. **Test other contact models**

The contact model presented here suffers from evident drawbacks: poor representation of the contact pressure at the interface, no frictional contact and a TCR model only suited for metal-metal interfaces and particular types of material properties. Several finite element models for the TCR such as the work of Zhangi [45] and Zheng [46] might be appropriate.

### 7.3.2 On the Cellular Contact-Aided Compliant Mechanism Designs

Further treatment of the topologies obtained here for the thermally actuated C3Ms constitutes another research direction. This direction corresponds to more of an engineering approach. A summary of various topics are listed below in decreasing order of priority:

1. **Post-processing of the topologies**

The topologies generated in this study rely on the adjustment of various parameters to follow the optimization process. The results are essentially an indication of the areas to fill with a specific material or the patterns to follow in order to enhance the thermal control of the C3Ms. A finer, realistic and feasible structure inspired from these patterns could emerge and then be tested. Post-treatment through a finite element package would provide valuable information regarding the behavior of the simulated final structure under various conditions. These would confirm or inform the choice of the topologies or material distribution trends proposed in this thesis, and would allow for possible corrections or adjustment in the algorithm.

2. **Comparison with existing designs**

Stavely [3] proposed a design presented in Figure 7.1 for the C3Ms. The low conductive and low coefficient of thermal expansion (CTE) material (gray) constitutes a center pillar
Figure 7.1: Proposed design for the C3Ms. Boundary conditions on the structure (a) and heat conduction paths in both “hot” case (b) and “cold” case (c). [3]

while the high conductive and high CTE material (blue) can create contact with the bottom surface. This design would be able to produce the necessary displacements that regulate the contact pressure at the interface. Nothing suggests that this design is optimal with respect to the thermal control objectives, though it provides effective thermal control in response to the various thermal conditions that can be encountered on the spacecraft. Any optimal topology generated should be compared to the performance of this already existing solution to evaluate the benefits of the topology optimization method.

3. **Integration of the C3Ms design into the interface**

Once the final design of the structure determined, the C3Ms must be assembled together to constitute the thermal contact interface. 3D modeling might be needed and the assembled C3Ms structure behavior analyzed. Other considerations such as vibratory and dynamic response to thermal variations could be of interest too.

4. **Fabrication of the C3Ms**

The fabrication of a two-material C3Ms is actually the subject of ongoing research with our collaborators Dr. James Adair and his students at the Pennsylvania State University. The fabrication technique investigated is referred to as a lost mold-rapid infiltration forming (LM-RIF) process [47] [48]. This process is well suited for the manufacturing of mesoscale parts (1mm or less in the largest dimension) arranged in large arrays with high aspect ratios or fine edge resolution. These parts are constituted of ceramic or metallic particulates that can be combined to form composites. 3D parts can be creating by stacking and laminating the molds in which the parts are casted.
The fabrication via the LM-RIF process is summarized in Figure 7.2. Molds in which the parts are casted are build using a modified UV-lithography. The molds are then filled with colloidal suspension (gel). The manufactured part is obtained after the sintering process, during which the molds are lost. The mold, but also the processing parameters for the colloidal suspension, can be greatly influenced by the topology of the compliant mechanism, especially with intricate designs.

C3Ms mechanisms have already been fabricated using zirconia and stainless steel [49]. Zirconia could be chosen as a low conductive material, while stainless steel, such as an alpha-aluminum oxide would constitutes the high conductive material. Recent research would also consider copper as the high conductive material. 3D Printing techniques are being investigated as a possible manufacturing process. Current issues concern the resolution of the
manufactured part, and obtaining a good surface roughness.

7.4 Final Comment

C3Ms constitutes a novel avenue for passive thermal control in space applications and beyond. Passive thermal conductive interfaces could be potentially applicable to nearly every thermal control subsystem. Energy and engineering costs engendered by complex and active thermal devices could be reduced. Successful design of such an interface could prove critical in the energy management, and thereby to mission of the spacecrafts.
Appendix A: Armijo Line Search

The Newton-Armijo model used in this thesis was developed by Kelly [39]. The Newton step is computed using a LU factorization for the Jacobians, and the step size is determined using an Armijo line search. In Section 5.2.2, the system of equations $\mathbf{H}$ (Equation (5.15)) is solved through a minimization problem, where the objective is equivalent to minimizing the potential energy such that [38]:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} d^T K_m d + (C_n^T P_n - F_p)^T d \\
\text{subject to} & \quad P_n(C_n d - g) = 0
\end{align*}
\]

Equation (1)

with the notation proposed in Section 5.2.2. The generic line search method is given as follow:

1. Choose an initial guess for the variables $x_0$

2. Until the stopping criterion $||\mathbf{H}_k|| < \text{Abstol} + \text{Reltol} ||\mathbf{H}_k||$ is met,
   
   (a) Calculate the search direction (direction of descent): $z_k = -J_k^{-1} \mathbf{H}_k$.

   (b) Compute the suitable step length $\delta_k$ such that $\mathbf{H}(x_k + \delta_k z_k) < \mathbf{H}(x_k)$. The determination of $\delta_k$ is called a line search.

   (c) Set $x_{k+1} = x_k + \delta_k z_k$ and $k \rightarrow k + 1$.

The Armijo line search performs the task 2(b). Compared to so-called “exact” line search approaches, it presents the advantage to prevent very small reductions in $\mathbf{H}$ by ensuring a sufficient decrease condition [33]:

\[
\mathbf{H}(x_k + \delta z_k) \leq \gamma \delta J z_k
\]

Equation (2)
where $0 < \gamma < 1$

It consists of a nested iterative approach which, given $\gamma, x_k, H_k$ and $J_k$, proceeds as follows:

1. Set $\delta = 1$

2. Calculate $H(x_k + \delta z_k)$

3. If the condition (.2) is satisfied, the step size for the current step is determined ($\delta_k = \delta$); else, set $\delta = \delta/2$ and come back to step 2.

Illustration of a one dimensional example is given in Figure .3. The principle of the Armijo’s line search is to avoid too long or too short step sizes using the constraint in Equation (.2), and helps the optimization process to converge faster.
Figure 3: One dimensional representation of the line search. (a) Typical behavior of a too long step size. (b) Typical behavior of a too short step size. (c) Principle of Armijo’s safeguarding technique.
Appendix B: Matlab Codes

1-material Mechanical and Thermal Algorithm (Used in Chapter 2)

Main Program

```matlab
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1-material Topology Optimization for mechanical and thermal problems
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

by Pierre Thurier, <pfthurier@gmail.com>

Last Update: July 2014

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

DESCRIPTION:
This algorithms performs optimizes a 1-material topology using a
mechanical and thermal system. The objective function is the mechanical
thermal compliance This algorithms uses the SIMP interpolation for
the material properties and the Method of Moving Asymptotes (MMA)
subroutine developed by Krister Svanberg.

EXECUTE using the matlab editor (run command)

INPUT : User choose a model (mechanical ot thermal)

OUTPUT : Topologies at each iterations of the optimization process,
final convergence plots, final field variables. It is possible to plot
```
the field variables contour on the domain at each iterations by setting
plott=1;
In a mechanical problem, plots the deformed optimal topology and the
corresponding displacement fields on the structure.
It is possible to plot the optimal stress distribution on the domain by
using the SigmaX, SigmaY, SigmaXY matrices (not completed in this
version).

CALLS:
- FEMel.m: builds the element stiffness matrix/conductivity matrix
- FEM.m: builds the global stiffness/conductivity matrix, defines
the boundary conditions and solve the governing equations.
- mmasub.m: MMA subroutine implemented by K. Svanberg.
- PlotDesignMesh.m: Plots the deformed topology (black and white
  solution)
- PlotFieldonDefoMesh.m: Plots field variables contours on the
deformed topology.

USAGE:
Common usage of this code requires the definition of parameters
nelx, nely, volfrac, penal and rmin.
Initialization is performed before the optimization loop: Prepare
filter, element matrices, etc. The material properties (\{nu,E\} or k)
must be specified.
The topology optimization loop goes on until the criterion for the
density variables ("change") is met. The definition of the boundary
conditions and loading are done through the call of FEM.m, where the
governing equations are solved. The sensitivity analysis is performed
following Sigmund's approach ("A 99 line topology optimization code
written in Matlab", 2001). The update of the design variables is
performed using the MMA subroutine (mmasub.m) developed and implemented by
Svanberg. Use of this subroutine requires the permission from the
author K. Svanberg (<krille@math.kth.se>).
%}
clear all;
clc;clf;close all;

disp('***** 1-Material Model for Thermal and Mechanical Systems *****')
disp('***** What model do you want to use? *****')
disp('***** Type 0 to quit *****')
var = input('***** For Mechanical type: 1 || for Thermal tpye: 2 => '); 
while var~==1 || var == 2
    if var == 0
        disp('Exiting Program...');
        break;
    end
    disp('**** Incorrect input. Insert 1 or 2 or q *****');
    var = input(' What model do you want to use (For Mechanical type 1, for Thermal tpye 2) => '); 
end 
if var == 0
    break;
end

plott = 1; % plott=1 : plot the field variables at each iterations (slows down the algorithm)

% Parameters
nelx=30; % Number of elements in the horizontal direction
nely=30; % Number of elements in the horizontal direction
volfrac=0.5; % Volume fraction of material
penal=3; % Penaty term
rmin=2; % Filter's radius
E=1.; % Modulus of elasticity
nu=0.3; % Poisson's ration
k = 1.; % Conductivity

% Vector of design variables (x-uho)
x = volfrac*ones(nely,nelx);

change = 1.; % Density change critirion
m=1; % Number of constraints (volume constraints)
n=nely*nelx; % Total number of elements
iter=0; % Number of iterations

% Parameters needed for the call of the MMA subroutine
xmin = 1e-3; % Densities' Lower bound
xmax = 1; % Densities' Upper bound
low = xmin;
upp = xmax;
xold1 = reshape(x,n,1);
xold2 = reshape(x,n,1);
a0=1;
a = zeros(m,1);
c = 10000*ones(m,1);
d= zeros(m,1);

if var == 1
    param = [nu, E];
    % Create a structure array:
    Parameters = struct('nelx',nelx,'nely',nely,'volfrac',volfrac,'E',E,...
                        'nu',nu,'penal',penal,'rmin',rmin)
else
    param = k;
    Parameters = struct('nelx',nelx,'nely',nely,'volfrac',volfrac,'k',k,...
                        'penal',penal,'rmin',rmin)
end

% PREPARE FE ANALYSIS
[KE] = FEMel( param );

% PREPARE FILTER (%%% From top88.m by Andreassen, Nov, 2010 %%%
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
kk = 0;
for il = 1:nelx
    for jl = 1:nely
        el = (il-1)*nely+jl;
...
for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        kk = kk+1;
        iH(kk) = e1;
        jH(kk) = e2;
        sH(kk) = max(0, rmin-sqrt((i1-i2)^2+(j1-j2)^2));
    end
end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);

%% TOPOLOGY OPTIMIZATION PROCESS %%

% START ITERATION
while or(change > 0.01 , iter < 30)
    iter = iter+1;
    % FE-ANALYSIS
    [K, U, F, fixeddofs, alldofs, freedofs] = FEM( nely, nelx, penal, var, x, KE );
    % PLOT FIELD VARIABLES
    if plott==1
        if var == 1
            Disp_x = sparse(nely+1,nelx+1);
            Disp_y = sparse(nely+1,nelx+1);
            for j=1:nelx+1
                for i=1:nely+1
                    Disp_x(i,j)=U(2*(nely+1)*(j-1)+2*i-1);
                    Disp_y(i,j)=U(2*(nely+1)*(j-1)+2*i);
                end
            end
            xx=1:nelx+1;
        end
    end
yy = -(1:nely+1);

% Plot final Ux displacement
figure(1);
contourf(xx,yy,Disp_x,50)
% surf(yy,xx,Disp_x);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')
colorbar;

% Plot final Uy displacement
figure(2);
contourf(xx,yy,Disp_y,50)
% surf(yy,xx,Disp_y);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')
colorbar;

elseif var == 2

T = zeros(nely+1,nelx+1);
for j = 1:nelx+1
    for i = 1:nely+1
        T(i,j) = U((nely+1)*(j-1)+i);
    end
end
figure(1);
xx = 1:nelx+1;
yy = 1:nely+1;
contourf(xx,yy,T,50);
ylabel('x');
xlabel('y');
colorbar;
% PREPARE SENSITIVITY ANALYSIS
f0val = 0;
df0dx_mat = zeros(nely,nelx);
df0dx2_mat = zeros(nely,nelx);
xval = reshape(x,n,1);

% DEFINE CONSTRAINTS
volconst = sum(sum(x))/(volfrac*n) - 1;
fval = volconst; % Column vector of size (1xm)!!
dfdx(1,:) = (1/(volfrac*n))*ones(1,n);

% CALCULATE SENSITIVITIES
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)*elx +ely;
        if var ==1
            edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
        else
            edof = [n1+1; n2+1; n2; n1];
        end
        Ue = U(edof,1);
        f0val = f0val + x(ely,elx)ˆpenal*Ue'*KE*Ue;
        df0dx_mat(ely,elx) = -penal*x(ely,elx)ˆ(penal-1)*Ue'*KE*Ue;
    end
end
df0dx = reshape(df0dx_mat,nely*nelx,1);

% FILTERING OF SENSITIVITIES
df0dx = H*(xval.*df0dx)./Hs./max(1e-3,xval);

% UPDATE DESIGN VARIABLES USING MMA
% Store previous density variables for next iteration
if iter>2
    xold2 = xold1;
    xold1 = xval;
elseif iter>1
    xold1 = xval;
end
x = reshape(xmma,nely,nelx);

% PRINT RESULTS
change = max(max(abs(xmma-xold1))); % Maximum density change
change_evol(iter) = change; % Store densities for future plots
OBJ(iter) = f0val; % Store Objective function value for future plots
disp([' It.: ' sprintf('%4i',iter) ' Obj.: ' sprintf('%10.4f',f0val) ...
    ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
    ' ch.: ' sprintf('%6.3f',change )])

% PLOT DENSITIES
figure(3);
colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);

end

% PLOT RESULTS %

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONVERGENCE AND DENSITIES PLOTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure();
plot(1:iter,OBJ,'LineWidth',1,...
    'Marker','o',...
    'MarkerSize',3,...
    'MarkerFaceColor','b');
ylabel('c (obj. function)');
xlabel('iteration');

% title('Convergence of Objective function')

figure();

plot(1:iter,change_evol,'LineWidth',1,...
    'Marker','o',...
    'MarkerSize',3,...
    'MarkerFaceColor','b');

ylabel('Max( |\rho_{new} - \rho_{old}| )');

xlabel('iteration');

% title('Density Change')

hold on

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PREPARE THE DATA TO PLOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if var == 1 % Mechanical model
    nd = length(U);

    % B matrix in FEM (Derivatives of the shape functions
    B = [-4 0 4 0 4 0 -4 0;
        0 -4 0 -4 0 4 0 4;
        -4 -4 -4 4 4 4 4 -4];

    xvoid = 1-x;

    % Initialize vectors of future plot
    SigmaX = zeros(nely,nelx); % Stress in x deirection
    SigmaY = zeros(nely,nelx); % Stress in y deirection
    SigmaXY = zeros(nely,nelx); % Shear stress
    XY_undeformed = zeros((nely+1)*(nelx+1),2); % Undeformed structure representation (node coordinates)
    XY_deformed = XY_undeformed;
    Nodes = zeros((nely+1)*(nelx+1)); % Deformed structure representation (node coordinates)
    Udisp = zeros(nd/2,2);
    Unorm = zeros(nd/2,1); % Vector of displament's norm.
    x_deformed = zeros(1,nely*nelx); % Vector of density variables.
init_pos_x = zeros(nelx+1,1); \hspace{1cm} \text{% Initial (horizontal) position of the free structure}
init_pos_y = zeros(nely+1,1); \hspace{1cm} \text{% Initial (vertical) position of the free structure}
Lx = nelx; \hspace{1cm} \text{% Horizontal length of the domain}
Ly = nely; \hspace{1cm} \text{% Vertical length of the domain}

for j = 1:nelx
    init_pos_x(j+1) = Lx*j/(nelx);
end
for i = 1:nely
    init_pos_y(nely+1-i) = Ly*i/(nely);
end

for j=1:nelx+1
    for i=1:nely+1
        if j<nelx+1 && i<nely+1
            n1 = (nely+1)*(j-1)+i;
            n2 = (nely+1)*j +i;
            Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
            % Calculate stress
            stress = B*Ue;
            SigmaX(i,j) = stress(1);
            SigmaY(i,j) = stress(2);
            SigmaXY(i,j) = stress(3);
            clear stress Ue;
        end
        Disp_x(i,j)=U(2*(nely+1)*(j-1)+2*i-1);
        Disp_y(i,j)=U(2*(nely+1)*(j-1)+2*i);
    end
end

XY_undeformed((nely+1)*(j-1)+i,:) = [ init_pos_x(j) , init_pos_y(i) ];
XY_deformed((nely+1)*(j-1)+i,:) = ... 
    XY_undeformed((nely+1)*(j-1)+i,:) + ...
    [ Disp_x(i,j), Disp_y(i,j) ];
Udisp((nely+1)*(j-1)+i,:) = [ Disp_x(i,j), Disp_y(i,j) ];
Unorm((nely+1)*(j-1)+i) = (Disp_x(i,j)^2 + Disp_y(i,j)^2)^(1/2);
if i<nely+1 && j<nelx+1
    n1 = (nely+1)*(j-1)+i;
    n2 = (nely+1)* j +i;
    Nodes(n1,n1+1) = 1;
    Nodes(n1+1,n2+1) = 1;
    Nodes(n2+1,n2) = 1;
    Nodes(n2,n1) = 1;
    x_deformed((nely)*(j-1)+i) = xvoid(i,j);
end
end

xx=l:nelx+1;
yy=-(1:nely+1);
SigmaVM = ( SigmaX.^2 + SigmaY.^2 - SigmaX.*SigmaY + 3*SigmaXY.^2 ).^(1/2);
DispNorm = ( Disp_x.^2 + Disp_y.^2 ).^(1/2);

%%%%%%%%%%%%%%%%%%% PLOT DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%

% Plot final Ux displacement
figure();
contourf(xx,yy,Disp_x,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')

% Plot final Uy displacement
figure();
contourf(xx,yy,Disp_y,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')
% Define the structure (elements with a minimum of 50% of material)
listallnodes = zeros((nely)*(nelx),4);
countint = 0;
for elx=1:nelx
    for ely=1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        if xvoid(ely,elx)<0.5
            countint = countint + 1;
            nodes(countint,1:4) = [n1; n2; n2+1; n1+1];
        end
        listallnodes((nely)*(elx-1)+ely,1:4) = [n1;n2;n2+1;n1+1];
    end
end

% generation of coordinates and connectivities
coordinates = [XY_deformed(:,1) XY_deformed(:,2)];
coordinates2 = [(1:length(XY_undeformed))' XY_undeformed];
nodes2 = [(1:length(nodes))' nodes];

% Plot displacement Uy in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,2))
title('Profile of U_y on deformed Mesh') ;
axis image;

% Plot displacement Ux in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,1))
title('Profile of U_x on deformed Mesh') ;
axis image;

% Plot displacements norm |U| in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Unorm)
title('Profile of |U| on deformed Mesh') ;
axis image;

% Plot deformed optimal topology
allnodes2 = [(1:length(listallnodes))' listallnodes];

PlotDesignMesh(coordinates2,allnodes2,1,XY_deformed,1-x_deformed)
title('Deformed Design')
axis image;

elseif var ==2 % Thermal model

T = zeros(nely+1,nelx+1);
% Plot temperature distribution
for j=1:nelx+1
    for i=1:nely+1
        T(i,j)=U((nely+1)*(j-1)+i);
    end
end
figure();
xx=[1:nelx+1];
yy=[1:nely+1];
contourf(xx,-yy,T,30)
axis image;
ylabel('y');
xlabel('x');
title('Temperature distribution')

end

%%%%%%%%%%%%%%%%%%%%%%%%% SAVE WORKSPACE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
saving = input([' *** Do you want to save the workspace as a' ...
'.mat file in the working directory? *** (y for yes) => ',s']);

if saving == 'y'
    filename = input([' *** Insert name xxxxxx of the file' ...
function [KE] = FEMel( param )

% This function builds the element stiffness matrix/conductivity matrix
% INPUT:  param : a list containing the mechanical (nu, E) or thermal
%           (k) properties
% OUTPUT : KE   : element stiffness/conductivity matrix

if length(param)==2
  nu = param(1);
  E = param(2);

  % Construct element stiffness matrix
  Kel=[ 1/2-nu/6  1/8+nu/8 -1/4-nu/12 -1/8+3 *nu/8 ...
       -1/4+nu/12 -1/8-nu/8 nu/6  1/8-3*nu/8];
  KE = E/(1-nu^2)*[ Kel(1) Kel(2) Kel(3) Kel(4) Kel(5) Kel(6) Kel(7) Kel(8) ...
                    Kel(2) Kel(1) Kel(8) Kel(7) Kel(6) Kel(5) Kel(4) Kel(3) ...
                    Kel(3) Kel(8) Kel(1) Kel(6) Kel(7) Kel(4) Kel(5) Kel(2) ...
                    Kel(4) Kel(7) Kel(6) Kel(1) Kel(8) Kel(3) Kel(2) Kel(5) ...
                    Kel(5) Kel(6) Kel(7) Kel(8) Kel(1) Kel(2) Kel(3) Kel(4) ...
                    Kel(6) Kel(5) Kel(4) Kel(3) Kel(2) Kel(1) Kel(8) Kel(7) ...
                    Kel(7) Kel(4) Kel(5) Kel(2) Kel(3) Kel(8) Kel(1) Kel(6) ...
                    Kel(8) Kel(3) Kel(2) Kel(5) Kel(4) Kel(7) Kel(6) Kel(1)];

elseif length(param) == 1

k = param;

% Construct element conductivity matrix
KE = (k/6)*... 
[4 -1 -2 -1
-1 4 -1 -2
-2 -1 4 -1
-1 -2 -1 4];

end

FEM Subroutine

function [K, U, F, fixeddofs, alldofs, freedofs] = FEM( nely, nelx, penal, var, x, KE )

% by Pierre Thurier, <pfthurier@gmail.com>
Last Update: July 2014

DESCRIPTION:
This function completes the formulation of the Finite Element Model for the
the topology optimization by completing the different tasks:
- Builds the global stiffness/conductivity matrix K by summing the
element matrices KE, with respect to the densities x and the penalty
penal.
- Defines the boundary conditions (fixeddofs) and the loading on the
domain (F).
- Solves the governing equations (KU = F)

INPUTS:
  nely : Number of vertical elements
  nelx : Number of horizontal elements
  penal : Penalty term
  var  : Defines which model should be used such as:
1 == Mechanical model
2 == Thermal model

x : Design variables (or densities)
KE : Element stiffness/conductivity matrix

OUTPUTS:
K : Global stiffness/conductivity matrix
U : Solution of the governing equations (vector)
    (Displacements/Temperatures)
F : Loading vector
fixeddofs : Array of constrained with prescribed boundary conditions
allnodes : Array of all the degrees of freedoms the domain
freedofs : Array of the "free" degrees of freedoms in the structure

if var==1
  % Define free and fixed degrees of freedoms
  fixeddofs = union(1:2:2*(nely+1),2*(nelx+1) *(nely+1));
  alldofs   = 1:2*(nely+1) *(nelx+1);
  freedofs  = setdiff(alldofs,fixeddofs);
  K = zeros(2*(nely+1) *(nelx+1), 2*(nely+1) *(nelx+1));  % Stiffness Matrix
  F = zeros(2*(nely+1) *(nelx+1),1);  % Vector of nodal forces
  U = zeros(2*(nely+1) *(nelx+1),1);  % DVector of nodal displacements
  for elx = 1:nelx
    for ely = 1:nely
      n1 = (nely+1)* (elx-1)+ely;
      n2 = (nely+1)* elx    +ely;
      edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
      K(edof,edof) = K(edof,edof) + x(ely,elx) ^penal*KE;
    end
  end
  % DEFINE LOADS
F(2,1) = -1;

% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:) = 0;

elseif var == 2

% Nodes at the boundary of the domain
top = 1:nely+1:(nely+1)*nelx+1;
bottom = nely+1:nely+1:(nely+1)*(nelx+1);
left = 1:nely+1;
right = (nely+1)*nelx+1:(nely+1)*(nelx+1);

fixeddofs = [left,right,top,bottom];
alldofs = 1:(nelx+1)*(nely+1);
freedofs = setdiff(alldofs,fixeddofs);

% Boundary conditions
K = zeros((nely+1)*(nelx+1), (nely+1)*(nelx+1));
F = 10^-2*ones((nelx+1)*(nely+1),1);
% U = ones((nelx+1)*(nely+1),1);
for ii=1:length(left)
    dof=left(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
end

for ii=1:length(right)
    dof=right(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
end
for ii=1:length(top)
    dof=top(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
end

for ii=1:length(bottom)
    dof=bottom(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
end

% Build Conductivity Matrix:
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        edof = [n1+1; n2+1; n2; n1];
        K(edof,edof) = K(edof,edof) + x(ely,elx)ˆpenal*KE;
    end
end
U = K \ F;

PlotFieldonDefoMesh Subroutine

function PlotFieldonDefoMesh(coordinates,nodes,factor,depl,component)
%--------------------------------------------------------------------------
% Purpose:
% To plot the profile of a component on deformed mesh
% Synopsis :
% ProfileonDefoMesh(coordinates,nodes,component)
% Variable Description:
coordinates - The nodal coordinates of the mesh
-----> coordinates = [node X Y Z]

nodes - The nodal connectivity of the elements
-----> nodes = [elementNo node1 node2......]

factor - Amplification factor (Change accordingly, trial)
depl - Nodal displacements
-----> depl = [UX UY UZ]

component - The components whose profile to be plotted
-----> components = a column vector in the order of node numbers

% NOTE : Please note that in coordinates , displacements first column is node number and in nodes forst column is element number .

%--------------------------------------------------------------------------

dimension = size(coordinates(:,2:end),2) ; % Dimension of the mesh

nel = length(nodes) ; % number of elements
nnode = length(coordinates) ; % total number of nodes in system
nnel = size(nodes,2)-1; % number of nodes per element

% Initialization of the required matrices
X = zeros(nnel,nel) ;
Y = zeros(nnel,nel) ;
Z = zeros(nnel,nel) ;
profile = zeros(nnel,nel) ;

% if dimension == 3 % For 3D plots
for iel=1:nel
    for i=1:nnel
        nd(i)=nodes(iel,i+1); % extract connected node for (iel)-th element
        X(i,iel)=coordinates(nd(i),2); % extract x value of the node
        Y(i,iel)=coordinates(nd(i),3); % extract y value of the node
        Z(i,iel)=coordinates(nd(i),4); % extract z value of the node

    end
    ux = depl(:,1)

```text

```
uy = depl(:,2) ;
uz = depl(:,3) ;

UX(:,iel) = ux(nd') ;          % extract displacement value's of the node
UY(:,iel) = uy(nd') ;
UZ(:,iel) = uz(nd') ;
profile(:,iel) = component(nd') ;

end

% Plotting the profile of a property on the deformed mesh
defoX = X+factor*UX ;
defoY = Y+factor*UY ;
defoZ = Z+factor*UZ ;
figure
plot3(defoX,defoY,defoZ,'k')
fill3(defoX,defoY,defoZ,profile)
title('Profile of component on deformed Mesh') ;
rotate3d on ;
axis off ;
% Colorbar Setting
SetColorbar

elseif dimension == 2 % For 2D plots

for iel=1:nel
    for i=1:nnel
        nd(i)=nodes(iel,i+1);        % extract connected node for (iel)-th element
        X(i,iel)=coordinates(nd(i),2);  % extract x value of the node
        Y(i,iel)=coordinates(nd(i),3);  % extract y value of the node
    end
    ux = depl(:,1) ;
    uy = depl(:,2) ;

    UX(:,iel) = ux(nd') ;
    UY(:,iel) = uy(nd') ;
    profile(:,iel) = component(nd') ;
end
PlotDesignMesh Subroutine

Same as PlotFieldonDefoMesh but change line 87 with:

```matlab
1 colormap(gray)
2 fill(defoX,defoY,1-component)
```

1-material Thermo-mechanical algorithm (Used in Chapter 3)

Main Program

```matlab
% "1-material Topology Optimization for thermo-mechanical problems

by Pierre Thurier, <pfthurier@gmail.com>
Last Update: July 2014
```
DESCRIPTION:
This algorithm performs optimizes a 1-material topology for
thermo-mechanical systems. The objective function is the mechanical
compliance. This algorithm uses the SIMP interpolation for
the material properties and the Method of Moving Asymptotes (MMA)
subroutine developed by Krister Svanberg.

EXECUTE using the matlab editor (run command)

INPUT: User choose a model (mechanical or thermal)

OUTPUT: Topologies at each iterations of the optimization process,
final convergence plots, final field variables. It is possible to plot
the field variables contours on the domain at each iterations by setting
plot=1;
In a mechanical problem, plots the deformed optimal topology and the
corresponding displacement fields on the structure.
It is possible to plot the optimal stress distribution on the domain by
using the SigmaX, SigmaX, SigmaXY matrices (not completed in this
version).

CALLS:
- FEMelthm.m: builds the element stiffness and conductivity
matrices
- FEMthm.m: builds the global stiffness and conductivity matrices
- mmasub.m: MMA subroutine implemented by K. Svanberg.
- PlotDesignMesh.m: Plots the deformed topology (black and white
solution)
- PlotFieldonDefoMesh.m: Plots field variables contours on the
dehomed topology.

USAGE:
Common usage of this code requires the definition of parameters
nelx, nely, volfrac, penal and rmin.
Initialization is performed before the optimization loop: Prepare
filter, element matrices, etc. The material properties (\(nu,E,k,\alpha\)) must be specified.

The topology optimization loop goes on until the criterion for the density variables ("change") is met. The definition of the boundary conditions and loading are done through the call of FEMthm.m, where the governing equations are solved. The sensitivity analysis is performed using an adjoint method. The update of the design variables is performed using the MMA subroutine (mmasub.m) developed and implemented by Svanberg. Use of this subroutine requires the permission from the author K. Svanberg (<krille@math.kth.se>).

```matlab
clear all;
clc;clf;close all;

clear all;
clc;clf;close all;

disp('***** 1-Material Model for Thermo-mechanical Systems *****')

plott = 1; % plott=1 : plot the field variables at each iterations (slows down the algorithm)

% Parameters
nelx=60; % Number of elements in the horizontal direction
nely=20; % Number of elements in the horizontal direction
volfrac=0.4; % Volume fraction of material
penal=3; % Penalty term
rmin=1.2; % Filter's radius

E=1.; % Modulus of elasticity
nu=0.3; % Poisson's ratio
k = 1.; % Conductivity
alpha = 5e-4; % Coefficient of thermal expansion (CTE)
Tref = 0; % Reference Temperature

% Vector of design variables (x-rho)
x = volfrac*ones(nely,nelx);

change = 1.; % Density change criterion
m=1; % Number of constraints (volume constraints)
n=nely*nelx; % Total number of elements
iter=0; % Number of iterations

% Parameters needed for the call of the MMA subroutine
xmin = 1e-3; % Densities' Lower bound
xmax = 1; % Densities' Upper bound
low = xmin;
upp = xmax;
xold1 = reshape(x,n,1);
xold2 = reshape(x,n,1);
a0=1;
a = zeros(m,1);
c = 10000*ones(m,1);
d = zeros(m,1);

Parameters = struct('nelx',nelx,'nely',nely,'volfrac',volfrac,'E',E,'nu',nu,'k',k,'CTE',alpha,'penal',penal,'rmin',rmin)

% PREPARE FE ANALYSIS
param = [nu,E,k,alpha];
[KE,KEth,CEthm] = FEMelthm( param );

% Set passive parts <= Fixed material densities
valx = 1;
pasv = reshape(passive,n,1);

% PREPARE FILTER (%% From top88.m by Andreassen, Nov, 2010 %%%
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
kk = 0;
for il = 1:nelx
    for jl = 1:nely
        e1 = (il-1)*nely+jl;
        for i2 = max(il-(ceil(rmin)-1),1):min(il+(ceil(rmin)-1),nelx)
            for j2 = max(jl-(ceil(rmin)-1),1):min(jl+(ceil(rmin)-1),nely)
                e2 = (i2-1)*nely+j2;
                kk = kk+1;
                iH(kk) = e1;
                jH(kk) = e2;
                sH(kk) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
            end
        end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);

\% TOPOLOGY OPTIMIZATION PROCESS \%

\% START ITERATION
while or(change > 0.01 , iter<30)
    iter = iter + 1;
    \% FE-ANALYSIS
    [Km, Kth, Um, Uth, Fm, Fth, dCthm fixeddofsm, alldofsm, freedofsm... fixeddofsth, alldofsth, freedofsth] = ...
    FEMthm( nely,nelx,penal,x,KE,KEth,CEthm,Tref );

\% PLOT FIELD VARIABLES
if plott==1
    Disp_x = zeros(nely+1,nelx+1);
Display = zeros(nely+1,nelx+1);
for j=1:nelx+1
  for i=1:nely+1
    Disp_x(i,j)=Um(2*(nely+1)*(j-1)+2*i-1);
    Disp_y(i,j)=Um(2*(nely+1)*(j-1)+2*i);
  end
end
xx=1:nelx+1;
yy=-(1:nely+1);

% Plot final Ux displacement
figure(1);
contourf(xx,yy,Disp_x,50)
% surf(yy,xx,Disp_x);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')
colorbar;

% Plot final Uy displacement
figure(2);
contourf(xx,yy,Disp_y,50)
% surf(yy,xx,Disp_y);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')
colorbar;

T = zeros(nely+1,nelx+1);
for j=1:nelx+1
  for i=1:nely+1
    T(i,j)=Uth((nely+1)*(j-1)+i);
  end
end
figure(3);
xx=1:nelx+1;
yy=1:nely+1;
contourf(xx,yy,T,50);
axis image;
ylabel('x');
xlabel('y');
title('Temperature distribution')
colorbar;
end

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS

% SOLVE ADJOINT EQUATIONS
lamm = zeros(2*(nely+1)*(nelx+1),1);
Flam = Fm;
lamm(freedofsm,:) = - Km(freedofsm,freedofsm) \ Flam(freedofsm,:);
lamm(fixeddofsm,:) = 0;

lamth = Kth \ ...
     (lamm'*dCthm-Um'*dCthm)';

% PREPARE SENSITIVITY ANALYSIS
f0val = 0;
df0dx_mat = zeros(nely,nelx);
df0dx2_mat = zeros(nely,nelx);
xval = reshape(x,n,1);

% DEFINE CONSTRAINTS
volconst = sum(sum(x))/(volfrac*(n)) - 1;
fval = volconst;  % Column vector of size (1xm)!!
dfdx(1,:) = (1/(volfrac*n))'*ones(1,n);

% CALCULATE SENSITIVITIES
for elx = 1:nelx
    for ely = 1:nely
\begin{verbatim}
229 n1 = (nely+1)*(elx-1)+ely;
230 n2 = (nely+1)* elx +ely;
231 edof4 = [n1+1; n2+1; n2; n1];
232 edof8 = [2*n1+1; 2*n1+2; 2*n2+1; 2*n2+2; 2*n2-1; 2*n2; 2*n1-1; 2*n1];
233
234 Ume = Um(edof8,1);
235 Uthe = Uth(edof4,1);
236 lamme = lamm(edof8,1);
237 lamthe = lamth(edof4,1);
238
239 f0val = f0val + x(ely,elx)ˆpenal*Ume'*KE*Ume;
240 df0dx_mat(ely,elx) = ...
241 penal*x(ely,elx)ˆ(penal-1)*Ume'*CEthm*Uthe + ...
242 lamme'*{(...
243 penal*x(ely,elx)ˆ(penal-1)*KE*Ume - ... 
244 penal*x(ely,elx)ˆ(penal-1)*CEthm*Uthe) + ... 
245 lamthe'*{penal*x(ely,elx)ˆ(penal-1)*KEth*Uthe); 
246    end 
247 end 
248
250 df0dx = reshape(df0dx_mat,nely*nelx,1);
251
252 % FILTERING OF SENSITIVITIES
253 df0dx = H*(xval.*df0dx)./Hs./max(1e-3,xval);
254
255 % UPDATE DESIGN VARIABLES USING MMA
256 [xmma,ymma,zzma,lam,ksi,eta,mu,zet,s,low,upp] = ...
257 mmasub(m,n,iter,xval,xmin,xmax,xold1,xold2, ...
258 f0val,df0dx,fval,dfdx,low,upp,a0,a,c,d);
259
260 if iter>2
261    xold2 = xold1;
262    xold1 = xval;
264 elseif iter>1
265    xold1 = xval;
\end{verbatim}
x = reshape(xmma,nely,nelx);

% PRINT RESULTS
change = max(max(abs(xmma-xold1))); % Maximum density change
change_evol(iter) = change; % Store densities for future plots
OBJ(iter) = f0val; % Store Objective function value for future plots
disp([' It.: ' sprintf('%4i',iter) ' Obj.: ' sprintf('%10.4f',f0val) ...
    ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
    ' ch.: ' sprintf('%6.3f',change )])

% PLOT DENSITIES
figure(4);
colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);

end

% PLOT RESULTS %

%%% CONVERGENCE AND DENSITIES PLOTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure();
plot(1:iter,OBJ,'LineWidth',1,...
    'Marker','o',...
    'MarkerSize',3,...
    'MarkerFaceColor','b');
ylabel('c (obj. function)');
xlabel('iteration');
% title('Convergence of Objective function')

figure();
plot(1:iter,change_evol,'LineWidth',1,...
    'Marker','o',...
    'MarkerSize',3,...
    'MarkerFaceColor','b');
ylabel('Max( |\rho_{new} - \rho_{old}| )');
xlabel('iteration');
% title('Density Change')
hold on

% Mechanical model
nd = length(Um);

% B matrix in FEM (Derivatives of the shape functions
B = [-4 0 4 0 4 0 -4 0;
     0 -4 0 -4 0 4 0 4;
    -4 -4 -4 4 4 4 4 -4];

xvoid = 1-x;

% Initialize vectors of future plot
SigmaX = zeros(nely,nelx); % Stress in x direction
SigmaY = zeros(nely,nelx); % Stress in y direction
SigmaXY = zeros(nely,nelx); % Shear stress
XY_undeformed = zeros((nely+1)*(nelx+1),2); % Undeformed structure representation (node coordinates)
XY_deformed = XY_undeformed; % Deformed structure representation (node coordinates)
Nodes = zeros((nely+1)*(nelx+1));
Udisp = zeros(nd/2,2);
Unorm = zeros(nd/2,1); % Vector of displacement's norm.
x_deformed = zeros(1,nely*nelx); % Vector of density variables.

init_pos_x = zeros(nelx+1,1); % Initial (horizontal) position of the free structure
init_pos_y = zeros(nely+1,1); % Initial (vertical) position of the free structure
Lx = nelx;
Ly = nely;

for j = 1:nelx
    init_pos_x(j+1) = Lx*j/(nelx);
end

for i = 1:nely
    init_pos_y(nely+1-i) = Ly*i/(nely);
end

PREPARE THE DATA TO PLOT
for j=1:nelx+1
    for i=1:nely+1
        if j<nelx+1 && i<nely+1
            n1 = (nely+1)*(j-1)+i;
            n2 = (nely+1)* j +i;
            Ue = Um([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
            % Calculate stress
            stress = B*Ue;
            SigmaX(i,j) = stress(1);
            SigmaY(i,j) = stress(2);
            SigmaXY(i,j) = stress(3);
            clear stress Ue;
        end
        Disp_x(i,j)=Um(2*(nely+1)*(j-1)+2*i);  
        Disp_y(i,j)=Um(2*(nely+1)*(j-1)+2*i);
        XY_undeformed((nely+1)*(j-1)+i,:) = [ init_pos_x(j) , init_pos_y(i) ];
        XY_deformed((nely+1)*(j-1)+i,:) = ...
          XY_undeformed((nely+1)*(j-1)+i,:) + ...
           [ Disp_x(i,j) , Disp_y(i,j) ];
        Udisp((nely+1)*(j-1)+i,:) = [ Disp_x(i,j) , Disp_y(i,j) ];
        Unorm((nely+1)*(j-1)+i) = (Disp_x(i,j)^2 + Disp_y(i,j)^2)^(1/2);
        if i<nely+1 && j<nelx+1
            n1 = (nely+1)*(j-1)+i;
            n2 = (nely+1)* j +i;
            Nodes(n1,n1+1) = 1;
            Nodes(n1+1,n2+1) = 1;
            Nodes(n2+1,n2) = 1;
            Nodes(n2,n1) = 1;
            x_deformed((nely)*(j-1)+i) = xvoid(i,j);
        end
    end
end
xx=1:nelx+1;
yy=-(1:nely+1);
SigmaVM = ( SigmaX.ˆ2 + SigmaY.ˆ2 - SigmaX.*SigmaY + 3*SigmaXY.ˆ2 ).ˆ(1/2);
DispNorm = ( Disp.x.ˆ2 + Disp.y.ˆ2 ).ˆ(1/2);

%%%%%%%%%%%%%%%%%%%%% PLOT DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%

% Plot final Ux displacement
figure();
contourf(xx,yy,Disp_x,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')

% Plot final Uy displacement
figure();
contourf(xx,yy,Disp_y,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')

% Define the structure (elements with a minimum of 50% of material)
listallnodes = zeros((nely)*(nelx),4);
countint = 0;
for elx=1:nelx
    for ely=1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        if xvoid(ely,elx)<0.5
            countint = countint + 1;
            nodes(countint,1:4) = [n1; n2; n2+1; n1+1];
        end
    end
end
listallnodes((nely)*(elx-1)+ely,1:4) = [n1;n2;n2+1;n1+1];
end

% generation of coordinates and connectivities
coordinates = [XY_deformed(:,1) XY_deformed(:,2)];
coordinates2 = [(1:length(XY_undeformed))' XY_undeformed];
nodes2 = [(1:length(nodes))' nodes];

% Plot displacement Uy in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,2))
title('Profile of U_y on deformed Mesh');
axis image;

% Plot displacement Ux in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,1))
title('Profile of U_x on deformed Mesh');
axis image;

% Plot displacements norm |U| in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Unorm)
title('Profile of |U| on deformed Mesh');
axis image;

%%%%%%%%%%%%%%%%%%%%%% PLOT TEMPERATURES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Thermal model
T = zeros(nely+1,nelx+1);
% Plot temperature distribution
for j=1:nelx+1
    for i=1:nely+1
        T(i,j)=Uth((nely+1)*(j-1)+i);
    end
end
figure();
xx=[1:nelx+1];
yy=[1:nely+1];
contourf(xx,-yy,T,30)
axis image;
ylabel('y');
xlabel('x');
title('Temperature distribution')

% Plot temperature distribution on deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Uth)
title('Temperature Profile on deformed Mesh')
axis image;

%%%%%%%%%%%%%%%%%%%%%% PLOT DEFORMED TOPOLOGY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
allnodes2 = [(1:length(listallnodes))' listallnodes];
PlotDesignMesh(coordinates2,allnodes2,1,XY_deformed,1-x_deformed)
title('Deformed Design')
axis image;

%%%%%%%%%%%%%%%%%%%%%%%%% SAVE WORKSPACE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
saving = input([' *** Do you want to save the workspace as a' ...
'.mat file in the working directory? *** (y for yes) => ',s');
if saving == 'y'
    filename = input([' *** Insert name xxxxxx of the file' ...
'(format: xxxxxx.mat) *** => ',s');
    fname = pwd;
    filename_param = [fname '\' filename '.mat'];
    save(filename_param);
end
FEMthm Subroutine

function [Km, Kth, Um, Uth, Fm, Fth, dCthm, fixeddofsm, alldofsm, freedofsm...
    fixeddofsth, alldofsth, freedofsth] = ...
    FEMthm( nely,nelx,penal,x,KE,KEth,CEthm, Tref )
%
by Pierre Thurier, <pfthurier@gmail.com>
Last Update: July 2014

DESCRIPTION:
This function completes the formulation of the Finite Element Model for the
topology optimization by completing the different tasks:
- Builds the global stiffness/conductivity matrix $K$ by summing the
element matrices $KE$, with respect to the densities $x$ and the penalty
penal.
- Defines the boundary conditions (fixeddofs) and the loading on the
domain ($F$).
- Solves the governing equations ($KU = F$)

INPUTS:
- nely : Number of vertical elements
- nelx : Number of horizontal elements
- penal : Penalty term
- $x$ : Design variables (or densities)
- $KE$ : Element stiffness matrix
- $KEth$ : Element conductivity matrix
- $CEthm$ : Element coupling matrix
- $Tref$ : Reference Temperature

OUTPUTS:
- Km : Global stiffness matrix
- Kth : Global conductivity matrix
- Um : Displacement vector
- Uth : Temperature vector
- Fm : Mechanical Loading vector
Fth : Thermal Loading vector
dCthm : Derivatives of Cthm with respect to Uth
fixeddofsm : Array of constrained with prescribed mechanical boundary conditions
allnodesm : Array of all the mechanical degrees of freedoms the domain
freedofsm : Array of the "free" mechanical degrees of freedoms in the structure
fixeddofsth : Array of constrained with prescribed thermal boundary conditions
allnodesth : Array of all the thermal degrees of freedoms the domain
freedofsth : Array of the "free" thermal degrees of freedoms in the structure

PARAMETERS:
Tsink : Sink Temperature
q : Heat flux at the boundary
%

%% THERMAL GOVERNING EQUATIONS

Tsink = 0; % Sink Temperature
q = 1; % Heat flux

% Nodes at the boundary of the domain
topth = 1:nely+1:(nely+1)*nelx+1;
bottomth = nely+1:nely+1:(nely+1)*(nelx+1);
leftth = 1:nely+1;
rightth = (nely+1)*nelx+1:(nely+1)*(nelx+1);

fixeddofsth = [leftth,rightth,topth,bottomth];
alldofsth = 1:(nelx+1)*(nely+1);
freedofsth = setdiff(alldofsth,fixeddofsth);

Kth = sparse((nely+1)*(nelx+1), (nely+1)*(nelx+1));
Fth = 0*ones((nelx+1)*(nely+1),1);

% Uth = ones((nelx+1)*(nely+1),1);

% Build Conductivity Matrix:
for elx = 1:nelx
    for ely = 1:nely
        if ely==1 % Set high conductivity material on top
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1) * elx + ely;
            edof = [n1+1; n2+1; n2; n1];
            Kth(edof,edof) = Kth(edof,edof) + 100*KEth;
        else % Set normal material elsewhere
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1) * elx + ely;
            edof = [n1+1; n2+1; n2; n1];
            Kth(edof,edof) = Kth(edof,edof) + x(ely,elx)^penal*KEth;
        end
    end
end

% Boundary conditions
for ii=1:length(leftth)
    dof=leftth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=0;
end

for ii=1:length(rightth)
    dof=rightth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=0;
end

for ii=1:length(topth)
    dof=topth(ii);

% Kth(dof,:) = 0;
% Kth(dof,dof) = 1;
Fth(dof,1) = q;
end

for ii=1:length(bottomth)
dof = bottomth(ii);
Kth(dof,:) = 0;
Kth(dof,dof) = 1;
Fth(dof,1) = Tsink;
end

Uth = Kth \ Fth;

%% MECHANICAL SYSTEM

% Define free and fixed degrees of freedoms
fixeddofsm = union(1:2*(nely+1), 2*(nely+1)*nelx+1:2*(nely+1)*(nelx+1));
alldofsm = 1:2*(nely+1)*(nelx+1);
freedofsm = setdiff(alldofsm, fixeddofsm);

Km = sparse(2*(nely+1)*(nelx+1), 2*(nely+1)*(nelx+1));  % Stiffness Matrix
Feps = zeros(2*(nely+1)*(nelx+1), 1);  % Thermal expansion forcing
Fp = sparse(2*(nely+1)*(nelx+1), 1);  % Vector of nodal forces
Fm = sparse(2*(nely+1)*(nelx+1), 1);  % Vector of nodal displacements
dCthm = zeros(2*(nely+1)*(nelx+1), (nely+1)*(nelx+1));

for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)*elx + ely;
        edof4 = [n1+1; n2+1; n2; n1];
        edof8 = [2*n1+1; 2*n1+2; 2*n2+1; 2*n2+2; 2*n2-1; 2*n2; 2*n1-1; 2*n1];
        Km(edof8, edof8) = Km(edof8, edof8) + x(ely, elx)^penal*KE;
Uthe = Uth(edof4,1);
Feps(edof8) = Feps(edof8) + x(ely,elx)^(penal)*CEthm*(Uthe-Tref);
dCthm(edof8,edof4) = dCthm(edof8,edof4) + x(ely,elx)^(penal)*CEthm;

% DEFINE LOADS
if mod(nelx+1,2)==0
  Fp(2*(nely+1)*(nelx+1)/2) = -0.5;
  Fp(2*(nely+1)*(nelx+1)/2+2*(nely+1)) = -0.5;
  L14 = ceil((nelx+1)/4);
  Fp(2*(nely+1)*(L14)) = -0.5;
  L34 = ceil(3*nelx/4);
  Fp(2*(nely+1)*(L34+1)) = -0.5;
else
  Fp(2*(nely+1)*(nelx+1)/2+1+2*floor((nely+1)/2)) = -1;
  L14 = floor(nelx/4);
  Fp(2*(nely+1)*(L14+1)) = -0.5;
  L34 = ceil(3*nelx/4);
  Fp(2*(nely+1)*(L34+1)) = -0.5;
end

Fm = Fp + Feps;

% SOLVING
Um(freedofsm,:) = Km(freedofsm,freedofsm) \ Fm(freedofsm,:);
Um(fixeddofsm,:) = 0;

end

FEMelthm Subroutine

function [KE, KEth, CEthm] = FEMelthm( param )
% This function builds the element stiffness matrix/conductivity matrix
% INPUT:  param : a list containing the material properties
% OUTPUT : KE : element stiffness matrix
%          KEth : element conductivity matrix
%          CEthm : element coupling matrix (thermal expansion)

nu = param(1);
E = param(2);
k = param(3);
alpha = param(4);

% Construct element stiffness matrix
Kel=[ 1/2-nu/6  1/8+nu/8  -1/4-nu/12  -1/8+3*nu/8 ... 
     -1/4+nu/12  -1/8-nu/8  nu/6  1/8-3*nu/8];
KE = E/(1-nu^2)*[ Kel(1) Kel(2) Kel(3) Kel(4) Kel(5) Kel(6) Kel(7) Kel(8) 
                  Kel(2) Kel(1) Kel(8) Kel(7) Kel(6) Kel(5) Kel(4) Kel(3) 
                  Kel(3) Kel(8) Kel(1) Kel(6) Kel(7) Kel(4) Kel(5) Kel(2) 
                  Kel(4) Kel(7) Kel(6) Kel(1) Kel(8) Kel(3) Kel(2) Kel(5) 
                  Kel(5) Kel(6) Kel(7) Kel(8) Kel(1) Kel(2) Kel(3) Kel(4) 
                  Kel(6) Kel(5) Kel(4) Kel(3) Kel(2) Kel(1) Kel(8) Kel(7) 
                  Kel(7) Kel(4) Kel(5) Kel(2) Kel(3) Kel(8) Kel(1) Kel(6) 
                  Kel(8) Kel(3) Kel(2) Kel(5) Kel(4) Kel(7) Kel(6) Kel(1)];

% Construct element conductivity matrix
KEth = (k/6)*... 
[ 4  -1  -2  -1 
 -1  4  -1  -2 
 -2  -1  4  -1 
-1  -2  -1  4 ];

% Element coupling matrix (thermal expansion)
CEthm = (E*alpha/(6*(1-nu)))*... 
[ -2  -2  -1  -1 ];
2-material Single Physics Algorithm (Used in Chapter 4)

Main Program

```matlab
end
```

DESCRIPTION:

This algorithm performs optimizes a 2-material topology using a mechanical and thermal system. The objective function is the mechanical/thermal compliance. This algorithm uses the SIMP interpolation for the material properties and the Method of Moving Asymptotes (MMA) subroutine developed by Krister Svanberg.

EXECUTE using the matlab editor (run command)

INPUT: User choose a model (mechanical or thermal)
OUTPUT: Topologies at each iterations of the optimization process, final convergence plots, final field variables. It is possible to plot the field variables contour on the domain at each iterations by setting `plott=1;`

In a mechanical problem, plots the deformed optimal topology and the corresponding displacement fields on the structure. It is possible to plot the optimal stress distribution on the domain by using the SigmaX, SigmaX, SigmaXY matrices (not completed in this version).

CALLS:
- FEM_2mat.m: builds the global stiffness/conductivity matrix, defines the boundary conditions and solve the governing equations.
- mmasub.m: MMA subroutine implemented by K. Svanberg.
- PlotDesignMesh.m: Plots the deformed topology (black and white solution)
- PlotFieldonDefoMesh.m: Plots field variables contours on the deformed topology.

USAGE:
Common usage of this code requires the definition of parameters `nelx,nelly,volfrac,penal and rmin`.

Initialization is performed before the optimization loop: Prepare filter, element matrices, etc. The material properties (`{nu,E}` or `k`) must be specified.

The topology optimization loop goes on until the criterion for the density variables ("change") is met. The definition of the boundary conditions and loading are done through the call of FEM_2mat.m, where the governing equations are solved. The sensitivity analysis is performed following Sigmund's approach ("A 99 line topology optimization code written in Matlab", 2001). The update of the design variables is performed using the MMA subroutine (mmasub.m) developed and implemented by Svanberg. Use of this subroutine requires the permission from the author K. Svanberg (<krille@math.kth.se>).

%}
clear all;
clc;clf;close all;

disp('***** 2-Material Model for Thermal and Mechanical Systems *****')
disp('***** What model do you want to use? *****')
disp('***** Type 0 to quit *****')
var = input('***** For Mechanical type: 1 | | for Thermal type: 2 => ');
while var~=1 && var ~= 2
    if var == 0
        disp('Exiting Program...');
        break;
    end
    disp('**** Incorrect input. Insert 1 or 2 or q *****');
    var = input(' What model do you want to use (For Mechanical type 1, for Thermal type 2) = ');
end
if var == 0
    break;
end
plott = 1; % plott=1 : plot the field variables at each iterations (slows down the algorithm)

%%% Setting Parameters
nelx=80; % Number of elements in the horizontal direction
nely=40; % Number of elements in the horizontal direction
volfrac=[0.5 0.2]; % Control the volume fractions of material
penal1=3; % Penaty term (void/material variable)
penal2=1; % Penaty term (material1/material2 variable)
penal = [penal1,penal2];
rmin=1.2; % Filter's radius

e = [2 1]; % Modulus of elasticity
k = [2 1]; % Conductivity
nu=0.3; % Poisson's ration

% Vector of design variables (x~rho)
x1 = volfrac(1)*ones(nely,nelx); % Density variable (void/material)
x2 = volfrac(2)*ones(nely,nelx); % Design variable (material1/material2)
change = 1.; % Density change criterion
m=2; % Number of constraints (volume constraints)
n=nely*nelx; % Total number of elements
iter=0; % Number of iterations

% Parameters needed for the call of the MMA subroutine
xmin = 1e-3; % Densities' Lower bound
xmax = 1; % Densities' Upper bound
low = ones(2*n,1)*xmin;
upp = ones(2*n,1)*xmax;
x1old1 = reshape(x1,n,1);
x1old2 = reshape(x1,n,1);
x2old1 = reshape(x2,n,1);
x2old2 = reshape(x2,n,1);
xold1 = [x1old1; x2old1];
xold2 = [x1old2; x2old2];
a0=1;
a = zeros(m,1);
c = 10000*ones(m,1);
d = zeros(m,1);

if var == 1
    Parameters = struct('nelx',nelx,'nely',nely,'volfrac',volfrac,'E',e,...
        'nu',nu,'penal',penal,'rmin',rmin)
else
    Parameters = struct('nelx',nelx,'nely',nely,'volfrac',volfrac,'k',k,...
        'penal',penal,'rmin',rmin)
end

% Set passive parts <= Fixed material densities
valx = [1 0.5];
for ely = 1:nely
    for elx = 1:nelx
        if sqrt((ely-nely/2.)^2+(elx-nelx/3.)^2) < nely/3.
130 \% passive(ely,elx) = 0;
131 \% xl(ely,elx) = valx;
132 \% else
133 \% passive(ely,elx) = 0;
134 \% end
135 \% end
136 pasv = reshape(passive,nely*nelx,1);
137
138 \% PREPARE FILTER (%%%% From top88.m by Andreassen, Nov, 2010 %%%
139 iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
140 jH = ones(size(iH));
141 sH = zeros(size(iH));
142 kk = 0;
143 for il = 1:nelx
144 \ for jl = 1:nely
145 \ \ el = (il-1)*nely+jl;
146 \ \ for i2 = max(il-(ceil(rmin)-1),1):min(il+(ceil(rmin)-1),nelx)
147 \ \ \ for j2 = max(jl-(ceil(rmin)-1),1):min(jl+(ceil(rmin)-1),nely)
148 \ \ \ \ e2 = (i2-1)*nely+j2;
149 \ \ \ \ kk = kk+1;
150 \ \ \ \ iH(kk) = el;
151 \ \ \ \ jH(kk) = e2;
152 \ \ \ \ sH(kk) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
153 \ \ \ end
154 \ \ end
155 \ end
156 \ H = sparse(iH,jH,sH);
157 \ Hs = sum(H,2);
158 \% START ITERATION
159 \ while or(change > 0.01 , iter<30)
160 \ \ iter = iter + 1;
161
% SIMP INTERPOLATION and FEA

if var == 1

    E = x1.ˆpenal1.*(e(1)*x2.ˆpenal2 + e(2)*x1.ˆ(1-x2).ˆpenal2);
    dEdx1 = penal1*x1.ˆ(1-penal1).*
      (e(1)*x2.ˆpenal2 + e(2)*x1.ˆ(1-x2).ˆpenal2);
    dEdx2 = penal2*x1.ˆpenal1.*
      (e(1)*x2.ˆpenal2- e(2)*x1.ˆ(1-x2).ˆpenal2-1));

% FE-ANALYSIS

[K, KE, U, F, fixeddofs, alldofs, freedofs] = ...
    FEM_2mat( nely, nelx, penal, E, x1, x2, var );

elseif var == 2

    kx = x1.ˆpenal1.*(k(1)*x2.ˆpenal2 + k(2)*(1-x2).ˆpenal2);
    dkdx1 = penal1*x1.ˆ(1-penal1).*
      (k(1)*x2.ˆpenal2 + k(2)*(1-x2).ˆpenal2);
    dkdx2 = penal2*x1.ˆpenal1.*
      ((k(1)*x2.ˆpenal2-1) - k(2)*x1.ˆ(1-x2).ˆpenal2-1));

% FE-ANALYSIS

[K, KE, U, F, fixeddofs, alldofs, freedofs] = ...
    FEM_2mat( nely, nelx, penal, kx, x1, x2, var );

end

% PLOT FIELD VARIABLES

if plott==1

    if var == 1

        Disp_x = zeros(nely+1,nelx+1);
        Disp_y = zeros(nely+1,nelx+1);
        for j=1:nelx+1
            for i=1:nely+1
                Disp_x(i,j)=U(2*(nely+1)*(j-1)+2*i-1);
                Disp_y(i,j)=U(2*(nely+1)*(j-1)+2*i-1);

        end
    end

168
end
end
xx=1:nelx+1;
yy=-1:nely+1;

% Plot final Ux displacement
figure(1);
contourf(xx,yy,Disp_x,50)
% surf(yy,xx,Disp_x);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')
colorbar;

% Plot final Uy displacement
figure(2);
contourf(xx,yy,Disp_y,50)
% surf(yy,xx,Disp_y);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')
colorbar;

elseif var ==2

T = zeros(nely+1,nelx+1);
for j=1:nelx+1
    for i=1:nely+1
        T(i,j)=U((nely+1)*(j-1)+i);
    end
end
figure(1);
xx=1:nelx+1;
yy=1:nely+1;
contourf(xx,yy,T,50);
ylabel('x');
xlabel('y');
colorbar;
end
end

% PREPARE SENSITIVITY ANALYSIS
f0val = 0;
df0dx_mat = zeros(nely,nelx);
xval1 = reshape(x1,n,1);
xval2 = reshape(x2,n,1);
xval = [xval1; xval2];

% DEFINE CONSTRAINTS
volconst1 = sum(sum(x1))/n - volfrac(1);
volconst2 = sum(sum(x1.*x2))/(n) - volfrac(2);
volconst3 = sum(sum(x1.*(1-x2)))/(n) - (1-volfrac(2));
fval = [volconst1; volconst2];

dfdx1(1,:) = ones(1,n)/(n);
dfx2(1,:) = zeros(1,n);
dfdx1(2,:) = xval2/(n);
dfdx2(2,:) = xval1/(n);

dfdx = [dfdx1(1,:), dfdx2(1,:)
        dfdx1(2,:), dfdx2(2,:)];

df0dx1_mat = zeros(nely,nelx);
df0dx2_mat = zeros(nely,nelx);

% CALCULATE SENSITIVITIES
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx  +ely;
        if var ==1
edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];

Ue = U(edof,1);

f0val = f0val + E(ely,elx)*Ue'*KE*Ue;
df0dx1_mat(ely,elx) = -dEdx1(ely,elx)*Ue'*KE*Ue;
df0dx2_mat(ely,elx) = -dEdx2(ely,elx)*Ue'*KE*Ue;

else
    edof = [n1+1; n2+1; n2; n1];
    Ue = U(edof,1);
    f0val = f0val + kx(ely,elx)*Ue'*KE*Ue;
df0dx1_mat(ely,elx) = -dkdx1(ely,elx)*Ue'*KE*Ue;
df0dx2_mat(ely,elx) = -dkdx2(ely,elx)*Ue'*KE*Ue;
end
end

end

df0dx1 = reshape(df0dx1_mat,nely*nelx,1);
df0dx2 = reshape(df0dx2_mat,nely*nelx,1);

% FILTERING OF SENSITIVITIES
df0dx1 = H*(xval(1:n).*df0dx1)./Hs./max(1e-3,xval(1:n));
df0dx2 = H*(df0dx2)./Hs;
df0dx = [df0dx1; df0dx2];

% UPDATE DESIGN VARIABLES USING MMA
[xmma,ymma,zmma,lambda,sigma,tau,eta,mu,nu,sigma,low,upp] = ...
    mmsubs(m,2*n,iter,xval,xmin,xmax,xold1,xold2, ...
    f0val,df0dx,fval,dffdx,low,upp,a0,a,c,d,pasv,valx);

% Store previous density variables for next iteration
if iter>2
    xold2 = xold1;
    xold1 = xval;
elseif iter>1
    xold1 = xval;
end
x1 = reshape(xmma(1:n),nely,nelx);
x2 = reshape(xmma(n+1:end),nely,nelx);
xmat1 = x1.*x2;
xmat2 = x1.*(1-x2);
xvoid = 1 - xmat1 - xmat2;

x_plot = zeros(nely,nelx,3);
xplot(:,:,1) = xvoid;
xplot(:,:,2) = xmat1;
xplot(:,:,3) = xmat2;

% PRINT RESULTS
change1 = max(max(abs(xmma(1:n)-xold1(1:n))));
change2 = max(max(abs(xmma(n+1:2*n)-xold1(n+1:2*n))));
change1_evol(iter) = change1;
change2_evol(iter) = change2;
change = max(change1);%,change2);
change_evol(iter) = change;
OBJ(iter) = f0val;
disp([' It.: ' sprintf('%4i',iter) ' Obj.: ' sprintf('%10.4f',f0val) ...
     ' Vol.: ' sprintf('%6.3f',sum(sum(xplot))/(n)) ...
     ' ch.: ' sprintf('%6.3f',change )])

% PLOT DENSITIES
I = make_bitmap (3,nelx,nely,xplot);
image(I), axis image off, drawnow;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CONVERGENCE AND DENSITIES PLOTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure();
plot(1:iter,OBJ,'LineWidth',1,...
     'Marker','o',...
     'MarkerSize',3,...
     'MarkerFaceColor','b');
ylabel('c (obj. function)');
xlabel('iteration');
% title('Convergence of Objective function')
figure();
plot(1:iter,change_evol,'LineWidth',1,...
'Marker','o',...
'MarkerSize',3,...
'MarkerFaceColor','b');
ylabel('Max( \mid \rho_{new} - \rho_{old}\mid )');
xlabel('iteration');
% title('Density Change')
hold on

%%%%%%%%%%%%%%%% PREPARE THE DATA TO PLOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if var ==1 % Mechanical model
    nd = length(U);
    % B matrix in FEM {Derivatives of the shape functions
    B = [-4 0 4 0 4 0 -4 0;  
         0 -4 0 -4 0 4 0 4;  
        -4 -4 -4 4 4 4 4 -4];
    % Initialize vectors of future plot
    SigmaX = zeros(nely,nelx); % Stress in x deirection
    SigmaY = zeros(nely,nelx); % Stress in y deirection
    SigmaXY = zeros(nely,nelx); % Shear stress
    XY_undeformed = zeros((nely+1)*(nelx+1),2); % Undeformed structure representation (node coordinates)
    XY_deformed = XY_undeformed;
    Nodes = zeros((nely+1)*(nelx+1));
    Udisp = zeros(nd/2,2); % Vector of displamement's norm.
    x_deformed = zeros(1,nely*nelx); % Vector of density variables.
    init_pos_x = zeros(nelx+1,1); % Initial (horizontal) position of the free structure
    init_pos_y = zeros(nely+1,1); % Initial (vertical) position of the free structure
\begin{verbatim}
Lx = nelx;    \hspace{0.5cm} \% Horizontal length of the domain
Ly = nely;    \hspace{0.5cm} \% Vertical length of the domain
for j = 1:nelx
    init_pos_x(j+1) = Lx*j/(nelx);
end
for i = 1:nely
    init_pos_y(nely+1-i) = Ly*i/(nely);
end

for j=1:nelx+1
    for i=1:nely+1
        if j<nelx+1 && i<nely+1
            n1 = (nely+1)*(j-1)+i;
            n2 = (nely+1)* j +i;
            Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
            \% Calculate stress
            stress = B*Ue;
            SigmaX(i,j) = stress(1);
            SigmaY(i,j) = stress(2);
            SigmaXY(i,j) = stress(3);
            clear stress Ue;
        end
        Disp_x(i,j)=U(2*(nely+1)*(j-1)+2*i-1);
        Disp_y(i,j)=U(2*(nely+1)*(j-1)+2*i);
        XY_undeformed((nely+1)*(j-1)+i,:) = [ init_pos_x(j) , init_pos_y(i) ];
        XY_deformed((nely+1)*(j-1)+i,:) = ...
        \% [ Disp_x(i,j), Disp_y(i,j) ];
        Udisp((nely+1)*((j-1)+i, :) = [ Disp_x(i,j), Disp_y(i,j) ];
        Unorm((nely+1)*(j-1)+i) = (Disp_x(i,j)^2 + Disp_y(i,j)^2)^(1/2);
    end
end
\end{verbatim}
n2 = (nely+1) * j + i;
Nodes(nl,nl+1) = 1;
Nodes(nl+1,n2+1) = 1;
Nodes(n2+1,n2) = 1;
Nodes(n2,n1) = 1;
x_deformed((nely)*(j-1)+i) = xvoid(i,j);
end

end

xx=1:nelx+1;
yy=-(1:nely+1);
SigmaVM = ( SigmaX.^2 + SigmaY.^2 - SigmaX.*SigmaY + 3*SigmaXY.^2 ).^(1/2);
DispNorm = ( Disp_x.^2 + Disp_y.^2 ).^(1/2);

%%%%%%%%%%%%%%%%%%%%% PLOT DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Plot final Ux displacement
figure();
contourf(xx,yy,Disp_x,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')

% Plot final Uy displacement
figure();
contourf(xx,yy,Disp_y,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')

% Define the structue (elements with a minimum of 50% of material)
listallnodes = zeros((nely)*(nelx),4);
countint = 0;

for elx=1:nelx
    for ely=1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1) * elx + ely;
        if xvoid(ely,elx)<0.5
            countint = countint + 1;
            nodes(countint,1:4) = [n1; n2; n2+1; n1+1];
        end
    end
    listallnodes((nely)*(elx-1)+ely,1:4) = [n1;n2;n2+1;n1+1];
end

% generation of coordinates and connectivities
coordinates = [XY_deformed(:,1) XY_deformed(:,2)];
coordinates2 = [(1:length(XY_undeformed))' XY_undeformed];
nodes2 = [(1:length(nodes))' nodes];

% Plot displacement Uy in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,2))
title('Profile of $U_y$ on deformed Mesh');
axis image;

% Plot displacement Ux in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,1))
title('Profile of $U_x$ on deformed Mesh');
axis image;

% Plot displacements norm $|U|$ in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Unorm)
title('Profile of $|U|$ on deformed Mesh');
axis image;

% Plot deformed optimal topology
allnodes2 = [(1:length(listallnodes))' listallnodes];
PlotDesignMesh(coordinates2,allnodes2,1,XY_deformed,xplot)
if var == 2 % Thermal model

    T = zeros(nely+1,nelx+1);

    % Plot temperature distribution
    for j=1:nelx+1
        for i=1:nely+1
            T(i,j)=U((nely+1)*(j-1)+i);
        end
    end

    figure();
    xx=[1:nelx+1];
    yy=[1:nely+1];
    contourf(xx,-yy,T,30)
    axis image;
    ylabel('y');
    xlabel('x');
    title('Temperature distribution')

end

%%%%%%%%%%%%%%%%%%%%%%%%% SAVE WORKSPACE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

saving = input([" *** Do you want to save the workspace as a'...

'.mat file in the working directory? *** (y for yes) = '],[,'s']);

if saving == 'y'

    filename = input([" *** Insert name xxxxxx of the file'...

'(format: xxxxxx.mat) *** => '],[,'s]);

    fname = pwd;

end
function I = make_bitmap (p, nelx, nely, x)

% DESCRIPTION:
% This subroutine prepare the multimaterial color representation of the
topologies.

INPUT:   p      : Number of phases
         nelx    : Number of horizontal elements
         nely    : Number of vertical elements
         x       : Design variables

OUTPUT:  I      : An nelx by nely by 3 matrix.

%}
color = [ 1 1 1; 1 0 0; 0 0 .45; 0 0 0; 1 1 1];
I = zeros(nely*nelx,3);
alpha = zeros(nelx*nely,p);
for j = 1:p
    alpha(:,j) = reshape(x(:,:,j),nely*nelx,1);
    I(:,1:3) = I(:,1:3) + alpha(:,j)*color(j,1:3);
end
% I = imresize(reshape(I,nely,nelx,3),10,'bilinear');
I = reshape(I,nely,nelx,3);
end

% color = [ 1 0 0; 0 0 .45; 0 1 0; 0 0 0; 1 1 1];
% I = zeros(nx*ny,3);
% for j = 1:p
% I(:,1:3) = I(:,1:3) + alpha(:,j)*color(j,1:3);
% end
% I = imresize(reshape(I,nely,nelx,3),10,'bilinear');

FEM_2mat Subroutine

function [K, KE, U, F, fixeddofs, alldofs, freedofs] = FEM_2mat( nely,nelx,penal,Eorkx,x1,x2,var )
%
by Pierre Thurier, <pfthurier@gmail.com>
Last Update: July 2014

DESCRIPTION:
This function completes the formulation of the Finite Element Model for the
 topology optimization by completing the different tasks:
 - Builds the element stiffness/conductivity matrix
 - Builds the global stiffness/conductivity matrix K by summing the
   element matrices KE, with respect to the densities x and the penalty
   penal.
 - Defines the boundary conditions (fixeddofs) and the loading on the
   domain (F).
 - Solves the governing equations (KU = F)

INPUTS:
nely : Number of vertical elements
nelx : Number of horizontal elements
penal : Penalty term
var : Defines which model should be used such as:
      1 == Mechanical model
      2 == Thermal model
x1,x2 : Design variables (or densities)
KE : Element stiffness/conductivity matrix

OUTPUTS:
K : Global stiffness/conductivity matrix
U : Solution of the governing equations (vector)  
   (Displacements/Temperatures)
F : Loading vector
fixeddofs : Array of constrained with prescribed boundary  
   conditions
allnodes : Array of all the degrees of freedoms the domain
freedofs : Array of the "free" degrees of freedoms in the  
   structure
%

if var==1

   nu = 0.3;
   % Construct element stiffness matrix
   Kel = [ 1/2-nu/6  1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 
         -1/4+nu/12 -1/8-nu/8 nu/6  1/8-3*nu/8 ];
   KE = 1/(1-nu^2) * [ Kel(1) Kel(2) Kel(3) Kel(4) Kel(5) Kel(6) Kel(7) Kel(8) 
      Kel(2) Kel(1) Kel(8) Kel(7) Kel(6) Kel(5) Kel(4) Kel(3) 
      Kel(3) Kel(8) Kel(1) Kel(6) Kel(7) Kel(4) Kel(5) Kel(2) 
      Kel(4) Kel(7) Kel(6) Kel(1) Kel(8) Kel(3) Kel(2) Kel(5) 
      Kel(5) Kel(6) Kel(7) Kel(8) Kel(1) Kel(2) Kel(3) Kel(4) 
      Kel(6) Kel(5) Kel(4) Kel(3) Kel(2) Kel(1) Kel(8) Kel(7) 
      Kel(7) Kel(4) Kel(5) Kel(2) Kel(3) Kel(8) Kel(1) Kel(6) 
      Kel(8) Kel(3) Kel(2) Kel(5) Kel(4) Kel(7) Kel(6) Kel(1) ];

   % Define free and fixed degrees of freedoms
   fixeddofs = union(1:2:2*(nely+1), 2*(nelx+1) *(nely+1));
   alldofs = 1:2*(nely+1) *(nelx+1);
   freedofs = setdiff(alldofs, fixeddofs);

   E = Eorkx;
   % FE-ANALYSIS
K = sparse(2*(nely+1)*(nelx+1),1); % Stiffness Matrix
F = sparse(2*(nely+1)*(nelx+1),1); % Vector of nodal forces
% U = zeros(2*(nely+1)*(nelx+1),1); % Vector of nodal displacements

for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
        K(edof,edof) = K(edof,edof) + E(ely,elx)*KE;
    end
end

% DEFINE LOADS
F(2,1) = -1;

% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:) = 0;

elseif var == 2

% Construct element conductivity matrix
KE = (1/6)*[
    4  -1  -2  -1  
   -1  4  -1  -2  
    -2  -1  4  -1  
   -1  -2  -1  4];

% Nodes at the boundary of the domain
top    = 1:nely+1:(nely+1)*nelx+1;
bottom = nely+1:nely+1:(nely+1)*nelx+1;
left   = 1:nely+1;
right  = (nely+1)*nelx+1:(nely+1)*nelx+1;
fixeddofs = [left,right,top,bottom];
alldofs = 1:(nelx+1)*(nely+1);
freedofs = setdiff(alldofs,fixeddofs);
kx = E*0.5x;

% FE-ANALYSIS
K = sparse((nely+1)*(nelx+1), (nely+1)*(nelx+1));  % Conductivity matrix
F = 10^-2*ones((nelx+1)*(nely+1),1);                  % Load vector
% U = ones((nelx+1)*(nely+1),1);                      % Temperatures

% Boundary Conditions
for ii=1:length(left)
    dof=left(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
end

for ii=1:length(right)
    dof=right(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
end

for ii=1:length(top)
    dof=top(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
end

for ii=1:length(bottom)
    dof=bottom(ii);
    K(dof,:) =0;
    K(dof,dof) =1;
    F(dof,1)=100;
% Build global Conductivity Matrix:
for elx = 1:nels
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        edof = [n1+1; n2+1; n2; n1];
        if ely == 1; % Introduction of the high conductive layer
            K(edof,edof) = K(edof,edof) + 10*KE;
        else
            K(edof,edof) = K(edof,edof) + kx(ely,elx)*KE;
        end
    end
end

U = K \ F;

2-material Thermo-mechanical algorithm without Contact (Used in Chapter 4)

Main Program

%{
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2-material Topology Optimization for thermo-mechanical problems
by Pierre Thurier, <pfthurier@gmail.com>
Last Update: July 2014
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
}
DESCRIPTION:

This algorithm performs optimizes a 2-material topology for thermo-mechanical systems. The objective function is the mechanical compliance. This algorithm uses the SIMP interpolation for the material properties and the Method of Moving Asymptotes (MMA) subroutine developed by Krister Svanberg.

EXECUTE using the matlab editor (run command)

INPUT: User choose a model (mechanical or thermal)

OUTPUT: Topologies at each iterations of the optimization process, final convergence plots, final field variables. It is possible to plot the field variables contour on the domain at each iterations by setting plott=1;

In a mechanical problem, plots the deformed optimal topology and the corresponding displacement fields on the structure.

CALLS:
- FEMthm_2mat.m: builds the global stiffness and conductivity matrices
- mmasub.m: MMA subroutine implemented by K. Svanberg.
- PlotDesignMesh.m: Plots the deformed topology (black and white solution)
- PlotFieldonDefoMesh.m: Plots field variables contours on the deformed topology.

USAGE:
Common usage of this code requires the definition of parameters nelx, nely, volfrac, penal and rmin.
Initialization is performed before the optimization loop: Prepare filter, element matrices, etc. The material properties (nu, E, k, alpha) must be specified.
The topology optimization loop goes on until the criterion for the
density variables ("change") is met. The definition of the boundary
conditions and loading are done through the call of FEMhm_2mat.m, where the
governing equations are solved. The sensitivity analysis is performed
using an adjoint method. The update of the design variables is
performed using the MMA subroutine (mmasub.m) developed and implemented by
Svanberg. Use of this subroutine requires the permission from the
author K. Svanberg (<krille@math.kth.se>).
%
clear all;
clc;clf;close all;

clear all;
clc;clf;close all;

disp('***** 2-Material Model for Thermo-mechanical Systems *****')

plott = 1; % plott=1 : plot the field variables at each iterations (slows down the algorithm)

% Parameters
nelx=60; % Number of elements in the horizontal direction
nely=20; % Number of elements in the horizontal direction
volfrac=[0.4 0.2]; % Volume fraction of material
penal1=3; % Penaty term
penal2=1; % Penaty term
penal = [penal1, penal2];
rmin=1.2; % Filter's radius

e = [2 1]; % Modulus of elasticity
nu=0.3; % Poisson's ration
k = [2 1]; % Conductivity
alpha = [5e-4 2e-4]; % Coefficient of thermal expansion (CTE)
Tref = 0; % Reference Temperature

% Vector of design variables (x-rho)
x1 = volfrac(1)*ones(nely,nelx); % Density variable (void/material)
x2 = volfrac(2)*ones(nely,nelx); % Design variable (material1/material2)
% Densitiy change critirion
change = 1.;

% Number of constraints (volume constraints)
m=2;

% Total number of elements
n = nely*nelx;

% Number of iterations
iter = 0;

% Parameters needed for the call of the MMA subroutine
xmin = 1e-3; % Densities' Lower bound
xmax = 1; % Densities' Upper bound
low = xmin;
upp = xmax;
x1old1 = reshape(x1,n,1);
x1old2 = reshape(x1,n,1);
x2old1 = reshape(x2,n,1);
x2old2 = reshape(x2,n,1);
xold1 = [x1old1; x2old1];
xold2 = [x1old2; x2old2];
a0 = 1;
a = zeros(m,1);  
c = 10000*ones(m,1);  
d = zeros(m,1);

Parameters = struct('nelx',nelx,'nely',nely,'volfrac',volfrac,'E',e,...  
                    'nu',nu,'k',k,'CTE',alpha,'penal',penal,'rmin',rmin,'Tref',Tref)

% Set passive parts <= Fixed material densities
valx = [1 0.5];

passive = sparse(nely,nelx);

% passive(1,:) = 1;

% x1(1,:) = valx(1);
% x2(1,:) = valx(2);

pasv = reshape(passive,nely*nelx,1);

% PREPARE FILTER (% From top88.m by Andreassen, Nov, 2010 %

iH = ones(nely*nelx*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));

sH = zeros(size(iH));
kk = 0;
for il = 1:nelx
  for j1 = 1:nely
    e1 = (il-1)*nely+j1;
    for i2 = max(il-(ceil(rmin)-1),1):min(il+(ceil(rmin)-1),nelx)
      for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        kk = kk+1;
        iH(kk) = e1;
        jH(kk) = e2;
        sH(kk) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
      end
    end
  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);

%% TOPOLOGY OPTIMIZATION PROCESS %%

% START ITERATION
while or(change > 0.01 , iter<30)
  iter = iter + 1;
  % SIMP INTERPOLATION
  E = x1.^penal1.*(e(1)*x2.^penal2 + e(2)*(1-x2).^penal2);
  dEdx1 = penal1*x1.^(penal1-1).*...
    (e(1)*x2.^penal2 + e(2)*(1-x2).^penal2);
  dEdx2 = penal2*x1.^(penal1.*...
    (e(1)*x2.^(penal2-1)- e(2)*(1-x2).^(penal2-1));
  kx = x1.^penal1.*(k(1)*x2.^penal2 + k(2)*(1-x2).^penal2);
  dkd1x1 = penal1*x1.^(penal1-1).*...
    (k(1)*x2.^penal2 + k(2)*(1-x2).^penal2);
  dkd2x2 = penal2*x1.^(penal1.*...
    ((k(1)*x2.^(penal2-1) - k(2)*(1-x2).^(penal2-1)));
\[ \text{Alpha} = x_2 \cdot \alpha(1) + (1-x_2) \cdot \alpha(2); \]
\[ d\text{Alpha}_{x2} = \alpha(1) - \alpha(2); \]

\% FE-ANALYSIS

\[ [K_E, K_{Eth}, C_{Eth}, K_m, K_{th}, U_m, U_{th}, F_m, F_{th}, dC_{thm}, \text{fixeddofsm}, \ldots \]
\[ \text{alldofsm}, \text{freedofsm}, \text{fixeddofsth}, \text{alldofsth}, \text{freedofsth}] = \ldots \]
\[ \thetaEC_{thm} \cdot 2 \text{mat}(nely, nelx, nu, E, kx, \text{Alpha}, Tref); \]

\% PLOT FIELD VARIABLES

\text{if} \; \text{plott}==1
\text{Disp}_x = \text{zeros}(\text{nely}+1, \text{nelx}+1);
\text{Disp}_y = \text{zeros}(\text{nely}+1, \text{nelx}+1);
\text{for} \; j=1: \text{nelx}+1
\text{for} \; i=1: \text{nely}+1
\text{Disp}_x(i,j)=\text{Um}(2*(\text{nely}+1)*(j-1)+2*i-1);
\text{Disp}_y(i,j)=\text{Um}(2*(\text{nely}+1)*(j-1)+2*i);
\text{end}
\text{end}
\text{xx}=1: \text{nelx}+1;
\text{yy}=-(1: \text{nely}+1);

\% Plot final Ux displacement
\text{figure}(1);
\text{contourf}(\text{xx}, \text{yy}, \text{Disp}_x, 50)
\% \text{surf}(\text{yy}, \text{xx}, \text{Disp}_x);
\text{axis} \; \text{image};
\text{ylabel}('y');
\text{xlabel}('x');
\text{title}('Virtual Displacement Ux')
\text{colorbar};

\% Plot final Uy displacement
\text{figure}(2);
\text{contourf}(\text{xx}, \text{yy}, \text{Disp}_y, 50)
\% \text{surf}(\text{yy}, \text{xx}, \text{Disp}_y);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')
colorbar;

T = zeros(nely+1,nelx+1);
for j=1:nelx+1
    for i=1:nely+1
        T(i,j)=Uth((nely+1)*(j-1)+i);
    end
end

figure(3);
xx=1:nelx+1;
yy=1:nely+1;
contourf(xx,yy,T,50);
axis image;
ylabel('x');
xlabel('y');
title('Temperature distribution')
colorbar;

%%%%%%%%%%%%%%%% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS %%%%%%%%%%%%

% SOLVE ADJOINT EQUATIONS
lamm = zeros(2*(nely+1)*(nelx+1),1);
Flam = Fm;
lamm(freedofsm,:) = - Km(freedofsm,freedofsm) \ Flam(freedofsm,:);
lamm(fixeddofsm,:) = 0;

lamth = Kth \ ...
    (lamm'*dCthm-Um'*dCthm)';

% PREPARE SENSITIVITY ANALYSIS
f0val = 0;
df0dx_mat = zeros(nely,nelx);
xval1 = reshape(x1,n,1);
xval2 = reshape(x2,n,1);
xval = [xval1; xval2];

% DEFINE CONSTRAINTS
volconst1 = sum(sum(x1))/n - volfrac(1);
volconst2 = sum(sum(x1.*x2))/(n) - volfrac(2);
% volconst3 = sum(sum(x1.*(1-x2)))/(n) - (1-volfrac(2));
fval = [volconst1; volconst2];

dfdx1(1,:) = ones(1,n)/(n);
dfdx2(1,:) = zeros(1,n);
dfdx1(2,:) = xval2/(n);
dfdx2(2,:) = xval1/(n);

dfx = [dfdx1(1,:), dfdx2(1,:)
       dfdx1(2,:), dfdx2(2,:)];

df0dx1_mat = zeros(nely,nelx);
df0dx2_mat = zeros(nely,nelx);

% CALCULATE SENSITIVITIES
for elx = 1:nelx
  for ely = 1:nely
    n1 = (nely+1)*(elx-1)+ely;
    n2 = (nely+1)* elx  +ely;
    edof4 = [n1+1; n2+1; n2; n1];
    edof8 = [2*n1+1; 2*n1+2; 2*n2+1; 2*n2+2; 2*n2-1; 2*n2; 2*n1-1; 2*n1];
    Ume = Um(edof8,1);
    Uthe = Uth(edof4,1);
    lamme = lamm(edof8,1);
    lamthe = lamth(edof4,1);
    f0val = f0val + E(ely,elx)*Ume'*KE*Ume;
    df0dx1_mat(ely,elx) = ...
  end
end
dEdx1(ely, elx) * Alpha(ely, elx) * Ume' * CEthm * Uthe + ...
1amma' *(...

lamme' *(...
dEdx2(ely, elx) * KE * Ume - ...

lamthe' * (dkdx2(ely, elx) * KEth * Uthe);

end

end

df0dx1 = reshape(df0dx1_mat, nely*nelx, 1);
df0dx2 = reshape(df0dx2_mat, nely*nelx, 1);

% FILTERING OF SENSITIVITIES
df0dx1 = H*(xval(1:n).*df0dx1 ./Hs./max(1e-3,xval(1:n)));
df0dx2 = H*(df0dx2)./Hs;
df0dx = [df0dx1; df0dx2];

% UPDATE DESIGN VARIABLES USING MMA
[xmma, ymma, zmma, lam, xsi, eta, mu, zet, s, low, upp] = ...

mmasub(m, 2*n, iter, xval, xmin, xmax, xold1, xold2, ...
f0val, df0dx, fval, dfdx, low, upp, a0, a, c, d, pasv, valx);

% Store previous density variables for next iteration
if iter>2
    xold2 = xold1;
    xold1 = xval;
elseif iter>1
    xold1 = xval;
end
x1 = reshape(xmma(1:n),nely,nelx);

x2 = reshape(xmma(n+1:end),nely,nelx);

xmat1 = x1.*x2;

xmat2 = x1.*(1-x2);

xvoid = 1 - xmat1 - xmat2;

x_plot = zeros(nely,nelx,3);

xplot(:,:,1) = xvoid;

xplot(:,:,2) = xmat1;

xplot(:,:,3) = xmat2;

% PRINT RESULTS
change1 = max(max(abs(xmma(1:n)-xold1(1:n))));
change2 = max(max(abs(xmma(n+1:2*n)-xold1(n+1:2*n))));
change1_evol(iter) = change1;
change2_evol(iter) = change2;
change = max(change1);%change2);
change_evol(iter) = change;
OBJ(iter) = f0val;
disp([' It.: ' sprintf('%4i',iter) ' Obj.: ' sprintf('%10.4f',f0val) ...
     ' Vol.: ' sprintf('%6.3f',sum(sum(xplot))/(n)) ...
     ' ch.: ' sprintf('%6.3f',change )])

% PLOT DENSITIES
I = make_bitmap (3,nelx,nely,xplot);
figure(4);
image(I), axis image off, drawnow;

end

%%% PLOT RESULTS %%%

% CONVERGENCE AND DENSITIES PLOTS %%%%%%%%%%%%%%%%%%%%
figure();
plot(1:iter,OBJ,'LineWidth',1,...
     'Marker','o',...%
     'MarkerSize',3,...
% MarkerFaceColor', 'b');
ylabel('c (obj. function)');
xlabel('iteration');

figure();
plot(1:iter, change_evol, 'LineWidth', 1,
'Marker', 'o', ... % MarkerSize', 3, ...
'MarkerFaceColor', 'b');
ylabel('Max( |\rho_{new} - \rho_{old}| )');
xlabel('iteration');

hold on

%%%%%%%%%%%%%%%% PREPARE THE DATA TO PLOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Mechanical model
nd = length(Um);

% B matrix in FEM (Derivatives of the shape functions
B = [-4 0 4 0 4 0 -4 0;
     0 -4 0 -4 0 4 0 4;
     -4 -4 -4 4 4 4 4 -4];

% Initialize vectors of future plot
SigmaX = zeros(nely, nelx); % Stress in x deirection
SigmaY = zeros(nely, nelx); % Stress in y deirection
SigmaXY = zeros(nely, nelx); % Shear stress
XY_undeformed = zeros((nely+1)*(nelx+1), 2); % Undeformed structure representation (node coordinates)
XY_deformed = XY_undeformed; % Deformed structure representation (node coordinates)
Nodes = zeros((nely+1)*(nelx+1));
Udisp = zeros(nd/2, 2);
Unorm = zeros(nd/2, 1); % Vector of displacement's norm.
x_deformed = zeros(1, nely*nelx); % Vector of density variables.
% Initial (horizontal) position of the free structure
init_pos_x = zeros(nelx+1,1);
% Initial (vertical) position of the free structure
init_pos_y = zeros(nely+1,1);
% Horizontal length of the domain
Lx = nelx;
% Vertical length of the domain
Ly = nely;

for j = 1:nelx
    init_pos_x(j+1) = Lx*j/(nelx);
end

for i = 1:nely
    init_pos_y(nely+1-i) = Ly*i/(nely);
end

for j=1:nelx+1
    for i=1:nely+1
        if j<nelx+1 && i<nely+1
            n1 = (nely+1)*(j-1)+i;
            n2 = (nely+1)* j + i;
            Ue = Um([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
            % Calculate stress
            stress = B*Ue;
            SigmaX(i,j) = stress(1);
            SigmaY(i,j) = stress(2);
            SigmaXY(i,j) = stress(3);
            clear stress Ue;
        end
    end
    Disp_x(i,j)=Um(2*(nely+1)*(j-1)+2*i-1);
    Disp_y(i,j)=Um(2*(nely+1)*(j-1)+2*i);

    XY_undeformed((nely+1)*(j-1)+i,:) = [ init_pos_x(j) , init_pos_y(i) ];
    XY_deformed((nely+1)*(j-1)+i,:) = ...
    [ Disp_x(i,j), Disp_y(i,j) ];
    Udisp((nely+1)*(j-1)+i,:) = [ Disp_x(i,j), Disp_y(i,j) ];
    Unorm((nely+1)*(j-1)+i) = (Disp_x(i,j)^2 + Disp_y(i,j)^2)^(1/2);
if i<nely+1 && j<nelx+1
    n1 = (nely+1)*(j-1)+i;
    n2 = (nely+1)* j +i;
    Nodes(n1,n1+1) = 1;
    Nodes(n1+1,n2+1) = 1;
    Nodes(n2+1,n2) = 1;
    Nodes(n2,n1) = 1;
    x_deformed((nely)*(j-1)+i) = xvoid(i,j);
end
end
end

xx=1:nelx+1;
yy=-(1:nely+1);
SigmaVM = ( SigmaX.^2 + SigmaY.^2 - SigmaX.*SigmaY + 3*SigmaXY.^2 ).^(1/2);
DispNorm = ( Disp_x.^2 + Disp_y.^2 ).^(1/2);

% PLOT DISPLACEMENTS

% Plot final Ux displacement
figure();
contourf(xx,yy,Disp_x,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')

% Plot final Uy displacement
figure();
contourf(xx,yy,Disp_y,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')
% Define the structure (elements with a minimum of 50% of material)
listallnodes = zeros((nely)*(nelx),4);
countint = 0;
for elx=1:nelx
  for ely=1:nely
    n1 = (nely+1)*(elx-1)+ely;
    n2 = (nely+1)* elx +ely;
    if xvoid(ely,elx)<0.5
      countint = countint + 1;
      nodes(countint,1:4) = [n1; n2; n2+1; n1+1];
    end
    listallnodes((nely)*(elx-1)+ely,1:4) = [n1;n2;n2+1;n1+1];
  end
end
% generation of coordinates and connectivities
coordinates = [XY_deformed(:,1) XY_deformed(:,2)];
coordinates2 = [(1:length(XY_deformed))' XY_undeformed];
nodes2 = [(1:length(nodes))' nodes];
% Plot displacement U_y in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,2))
title('Profile of U_y on deformed Mesh') ;
axis image;
% Plot displacement U_x in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Udisp(:,1))
title('Profile of U_x on deformed Mesh') ;
axis image;
% Plot displacements norm |U| in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Unorm)
title('Profile of |U| on deformed Mesh') ;
axis image;
% Thermal model
T = zeros(nely+1,nelx+1);

% Plot temperature distribution
for j=1:nelx+1
    for i=1:nely+1
        T(i,j)=Uth((nely+1)*(j-1)+i);
    end
end

figure();
xx=[1:nelx+1];
yy=[1:nely+1];
contourf(xx,-yy,T,30)
axis image;
ylabel('y');
xlabel('x');
title('Temperature distribution')

% Plot temperature distribution on deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Uth)
title('Temperature Profile on deformed Mesh')
axis image;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
allnodes2 = [(1:length(listallnodes))' listallnodes];
PlotDesignMesh(coordinates2,allnodes2,1,XY_deformed,xplot)
title('Deformed Design')
axis image;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
saving = input([' *** Do you want to save the workspace as a' ...
'.mat file in the working directory? *** (y for yes) ='],'s');
if saving == 'y'
    filename = input([' *** Insert name xxxxxx of the file' ... 
                      '(format: xxxxxx.mat) *** => ',s');
    fname = pwd;
    filename_param = [fname '\' filename '.mat'];
    save(filename_param);
end

FEMthm_2mat Subroutine

function [KE, KEth, CEthm, Km, Kth, Um, Uth, Fm, Fth, dCthm,...
       alldofsm, freedofsm, fixeddofsth, alldofsth, freedofsth] = ...
       FEMthm_2mat( nely,nelx,nu,E,kx,Alpha,Tref )

%{

by Pierre Thurier, <pfthurier@gmail.com>
Last Update: July 2014

DESCRIPTION:
This function completes the formulation of the Finite Element Model for the
the topology optimization by completing the different tasks:
- Buils the global stiffness/conductivity matrix K by summing the
element matrices KE, with respect to the densities x and the penalty
penal.
- Defines the boundary conditions (fixeddofs) and the loading on the
domain (F).
- Solves the governing equations (KU = F)

INPUTS:
  nely : Number of vertical elements
  nelx : Number of horizontal elements
  nu   : Poisson's ratio
  E    : Matrix of moduli of elasticity (via SIMP)
  kx   : Matrix of conductivities (via SIMP)
  Alpha : Matrix of the coefficient of thermal expansion (via
SIMP)

Tref : Reference Temperature

OUTPUTS:

KE : Element stiffness matrix
KEth : Element conductivity matrix
CEthm : Element coupling matrix
Km : Global stiffness matrix
Kth : Global conductivity matrix
Um : Displacement vector
Uth : Temperature vector
Fm : Mechanical Loading vector
Fth : Thermal Loading vector
dCthm : Derivatives of Cthm with respect to Uth
fixeddofsm : Array of constrained with prescribed mechanical
            boundary conditions
allnodesm : Array of all the mechanical degrees of freedoms the
            domain
freedofsm : Array of the "free" mechanical degrees of freedoms in
            the structure
fixeddofsth : Array of constrained with prescribed thermal
            boundary conditions
allnodesth : Array of all the thermal degrees of freedoms the
            domain
freedofsth : Array of the "free" thermal degrees of freedoms in
            the structure

PARAMETERS:

Tsink : Sink Temperature
q : Heat flux at the boundary

}%

%% THERMAL GOVERNING EQUATIONS

% Set parameters
Tsink = 0; % Sink Temperature
q = 1.5; % Heat flux
% Construct element stiffness matrix

Kel=[ 1/2-nu/6  1/8+nu/8  -1/4-nu/12  -1/8+3*nu/8 ... 
     -1/4+nu/12  -1/8-nu/8  nu/6  1/8-3*nu/8];

KE = 1/(1-nuˆ2) * [ Kel(1) Kel(2) Kel(3) Kel(4) Kel(5) Kel(6) Kel(7) Kel(8) 
                    Kel(2) Kel(1) Kel(8) Kel(7) Kel(6) Kel(5) Kel(4) Kel(3) 
                    Kel(3) Kel(8) Kel(1) Kel(6) Kel(7) Kel(4) Kel(5) Kel(2) 
                    Kel(4) Kel(7) Kel(6) Kel(1) Kel(8) Kel(3) Kel(2) Kel(5) 
                    Kel(5) Kel(6) Kel(7) Kel(8) Kel(1) Kel(2) Kel(3) Kel(4) 
                    Kel(6) Kel(5) Kel(4) Kel(3) Kel(2) Kel(1) Kel(8) Kel(7) 
                    Kel(7) Kel(4) Kel(5) Kel(2) Kel(3) Kel(8) Kel(1) Kel(6) 
                    Kel(8) Kel(3) Kel(2) Kel(5) Kel(4) Kel(7) Kel(6) Kel(1)];

% Construct element conductivity matrix

KEth = (1/6) * 
[ 4  -1  -2  -1 
  -1  4   -1  -2 
  -2  -1  4    -1 
  -1  -2  -1  4 ];

% Element coupling matrix (thermal expansion)

CEthm = (1/(6*(1-nu))) * [-2   -1  -1  -2; 
                         -2   -1   1  1; 
                         -1  -2  -2  -1; 
                         1    1  2  2; 
                         1    2  2  1; 
                         -1  -1  -2  -2; 
                         2    1  1  2 ];

% Nodes at the boundary of the domain
topth = 1:nely+1:(nely+1)\*nelx+1;
bottomth = nely+1:nely+1:(nely+1)\*nelx+1;
leftth = 1:nely+1;
rightth = (nely+1)\*nelx+1:(nely+1)\*nelx+1;

fixeddofsth = [leftth,rightth,topth,bottomth];
alldofsth = 1:(nelx+1)\*nely+1;
freedofsth = setdiff(alldofsth,fixeddofsth);

% Boundary conditions
Kth = sparse((nely+1)\*nelx+1), (nely+1)\*nelx+1);
Fth = 0*ones((nelx+1)\*nely+1),1);

% Uth = ones((nelx+1)\*nely+1),1);
for ii=1:length(leftth)
    dof=leftth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=0;
end

for ii=1:length(rightth)
    dof=rightth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=0;
end

for ii=1:length(topth)
    dof=topth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=q;
end

for ii=1:length(bottomth)
    dof=bottomth(ii);
    Kth(dof,:) =0;
end
Kth(dof,dof) = 1;
Fth(dof,1) = Tsink;
end

% Build Conductivity Matrix:
for elx = 1:nelx
    for ely = 1:nely
        if ely == 1 % Set high conductivity material on top
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1)*elx + ely;
            edof = [n1+1; n2+1; n2; n1];
            Kth(edof,edof) = Kth(edof,edof) + 10*KEth;
        else % Set normal material elsewhere
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1)*elx + ely;
            edof = [n1+1; n2+1; n2; n1];
            Kth(edof,edof) = Kth(edof,edof) + kx(ely,elx)*KEth;
        end
    end
end
Kth = (Kth+Kth')/2;
Uth = Kth \ Fth;

%% MECHANICAL SYSTEM

% Define free and fixed degrees of freedoms
fixeddofsm = union(1:2*(nely+1),2*(nely+1)*(nelx)+1:2*(nely+1)*(nelx+1));
alldofsm = 1:2*(nely+1)*(nelx+1);
freedofsm = setdiff(alldofsm,fixeddofsm);

Km = sparse(2*(nely+1)*(nelx+1), 2*(nely+1)*(nelx+1)); % Stiffness Matrix
Feps = zeros(2*(nely+1)*(nelx+1),1); % Thermal expansion forcing
Fp = sparse(2*(nely+1)*(nelx+1),1);
Fm = sparse(2*(nely+1)*(nelx+1),1); % Vector of nodal forces
Um = zeros(2*(nely+1)*(nelx+1),1); % Vector of nodal displacements
dCthm = zeros(2*(nely+1)*(nelx+1),(nely+1)*(nelx+1));

for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        edof4 = [n1+1; n2+1; n2; n1];
        edof8 = [2*n1+1; 2*n1+2; 2*n2+1; 2*n2+2; 2*n2-1; 2*n2; 2*n1-1; 2*n1];
        Km(edof8,edof8) = Km(edof8,edof8) + E(ely,elx)*KE;
        Uthe = Uth(edof4,1);
        Feps(edof8) = Feps(edof8) + E(ely,elx)*Alpha(ely,elx)*CEthm*(Uthe-Tref);
        dCthm(edof8,edof4) = dCthm(edof8,edof4) + E(ely,elx)*Alpha(ely,elx)*CEthm;
    end
end
Km = (Km+Km')/2;

% DEFINE LOADS
if mod(nelx+1,2)==0
    Fp(2*(nely+1)*(nelx+1)/2) = -0.5;
    Fp(2*(nely+1)*(nelx+1)/2+2*(nely+1)) = -0.5;
    L14 = ceil((nelx+1)/4);
    Fp(2*(nely+1)*(L14)) = -0.5;
    L34 = ceil(3*nelx/4);
    Fp(2*(nely+1)*(L34+1)) = -0.5;
else
    Fp(2*(nely+1)*(nelx+1)/2+1+2*floor((nely+1)/2)) = -1;
    L14 = floor(nelx/4);
    Fp(2*(nely+1)*(L14+1)) = -0.5;
    L34 = ceil(3*nelx/4);
    Fp(2*(nely+1)*(L34+1)) = -0.5;
end
Fm = Fp + Feps;

% SOLVING
Um(freedofsm,:) = Km(freedofsm,freedofsm) \ Fm(freedofsm,:);

Um(fixeddofsm,:) = 0;

end

2-material Thermo-mechanical algorithm with Contact (Used in Chapter 5 and 6)

Main Program

{%

2-material Topology Optimization for thermo-mechanical problems with contact

by Pierre Thurier, <pfthurier@gmail.com>

Last Update: July 2014

DESCRIPTION:

This algorithms performs optimizes a 2-material topology for thermo-mechanical systems with contact. The objective function is the mechanical compliance. This algorithms uses the SIMP interpolation for the material properties and the Method of Moving Asymptotes (MMA) subroutine developed by Krister Svanberg.

EXECUTE using the matlab editor (run command)

INPUT : User choose a model (mechanical ot thermal)

OUTPUT : Topologies at each iterations of the optimization process,
final convergence plots, final field variables. It is possible to plot
the field variables contour on the domain at each iterations by setting
plott=1;
In a mechanical problem, plots the deformed optimal topology and the
Corresponding displacement fields on the structure.
It is possible to plot the optimal stress distribution on the domain by
using the SigmaX, SigmaX, SigmaXY matrices (not completed in this
version).

CALLS:
- mmasub.m : MMA subroutine implemented by K. Svanberg.
- nsold.m : Newton-Armijo non linear solver implementd by T. Kelley
  -> nsold.m calls GovEq, which builds the governing equations
  into matrix form
- PlotDesignMesh.m : Plots the deformed topology (black and white
  solution)
- PlotFieldonDefoMesh.m : Plots field variables contours on the
deformed topology.

USAGE:
Common usage of this code requires the definition of parameters
nelx, nely, volfrac, penal and rmin, Tref.
Initialization is performed before the optimization loop: Prepare
filter, element matrices, etc. The material properties (nu,E,k,alpha)
must be specified.
The topology optimization loop goes on until the criterion for the
density variables ("change") is met. The definition of the boundary
conditions and loading are done, and the governing equations are solved
using the Newton-Armijo solver nsold.m. The sensitivity analysis is
performed using an adjoint method. The update of the design variables is
performed using the MMA subroutine (mmasub.m) developed and implemented by
Svanberg. Use of this subroutine requires the permission from the
author K. Svanberg (<krille@math.kth.se>).
%}
clear all;
clc;clf;close all;
clear all;
clc;clf;close all;

disp('***** 2-Material Model for Thermo-mechanical Systems with contact *****')

plott = 1; % plott=1 : plot the field variables at each iterations (slows down the algorithm)

global Km Cn Constd Constn Fm gA nd nn nth Cndof fixeddofs TCC0 Tsink... Cthm Kth Fth Kthjac Cndofth Ksol bottomth

% Parameters

nelx=60; % Number of elements in the horizontal direction
nely=20; % Number of elements in the horizontal direction
volfrac=[0.4 0.2]; % Volume fraction of material
penal1=3; % Penalty term
penal2=1; % Penalty term
penal = [penal1, penal2];
rmin=1.2; % Filter's radius

% Modulus of elasticity
% Poisson's ration
% Conductivity
% Coefficient of thermal expansion (CTE)
% Reference Temperature
% Nominal thermal contact conductance

% Vector of design variables (xrho)
x1 = volfrac(1)*ones(nely,nelx); % Density variable (void/material)
x2 = volfrac(2)*ones(nely,nelx); % Design variable (material1/material2)

change = 1.; % Density change criterion
m=2; % Number of constraints (volume constraints)
n=nely*nelx; % Total number of elements
iter=0; % Number of iterations

% Parameters needed for the call of the MMA subroutine
xmin = 1e-3; % Densities' Lower bound
% Densities’ Upper bound
xmax = 1;
low = xmin;
upp = xmax;
x1old1 = reshape(x1,n,1);
x1old2 = reshape(x1,n,1);
x2old1 = reshape(x2,n,1);
x2old2 = reshape(x2,n,1);
xold1 = [x1old1; x2old1];
xold2 = [x1old2; x2old2];
a0 = 1;
a = zeros(m,1);
c = 10000*ones(m,1);
d = zeros(m,1);

% Parameters to call the nsold.m subroutine
tol = [1.d-6, 1.d-6];
parms = [40, 1, 0, 0];

Parameters = struct('nelx',nelx,'nely',nely,'volfrac',volfrac,'E',e,'nu',nu,'k',k,'CTE',alpha,'penal',penal,'rmin',rmin,'Tref',Tref)

% Set passive parts <= Fixed material densities
valx = [1 0.5];
passive = sparse(nely,nelx);

% passive(1,:) = 1;
% x1(1,:) = valx(1);
% x2(1,:) = valx(2);
pasv = reshape(passive,nely*nelx,1);

% PREPARE FILTER (%% From top88.m by Andreassen, Nov, 2010 %%%
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for il = 1:nelx
    for jl = 1:nely

el = (i1-1)*nely+j1;
for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        kk = kk+1;
        iH(kk) = el;
        jH(kk) = e2;
        sH(kk) = max(0,rmin-sqrt((i1-i2)ˆ2+(j1-j2)ˆ2));
    end
end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);

% DEFINE CONTACT LOCATION
Cndofs = [2*(nely+1):2*(nely+1):2*(nely+1)*floor((nelx+1)/3), ... 
        2*(nely+1)*2*floor((nelx+1)/3):2*(nely+1):2*(nely+1)\(nelx+1)];
Cndofs = [];  % Nodes in potential contact
Cndofth = Cndof/2;  % Number of nodes in potential contact
Cc = zeros(2*(nely+1)\(nelx+1),nn);  % Submatrix for the governing eq.
gA = guess*ones(nn,1);  % Algebraic Distance of the support from the free structure
for ii = 1:nn
    Cn(Cndof(ii),ii) = -1;
end
Cn = Cn';  % Submatrix for the governing eq.
n = 2*(nely+1)\(nelx+1);
nth = n/2;

% INITIAL GUESS FOR THE FIELD VARIABLES
yd = -ones(2*(nely+1)\(nelx+1),1);  % Displacements
% Contact forces
yn = 1*ones(nn,1);
% Temperatures
yth = 1*ones(nth,1);
y = [ yd ; yn ; yth];
y0 = y;
Constn = -eye(nn,nn); % Submatrix for the governing eq.

%% TOPOLOGY OPTIMIZATION PROCESS %%

% START ITERATION
while or(change > 0.01 , iter<30)
    iter = iter + 1;
    % SIMP INTERPOLATION
    E = x1.^penal1.*(e(1)*x2.^penal2 + e(2)*(-x2).^penal2);
    dEdx1 = penal1*x1.^(penal1-1).*...
                 (e(1)*x2.^penal2 + e(2)*(-x2).^penal2);
    dEdx2 = penal2*x1.^(penal1).*...
                 (e(1)*x2.^(penal2-1)- e(2)*(-x2).^penal2);
    kx = x1.^penal1.*(k(1)*x2.^penal2 + k(2)*(-x2).^penal2);
    dkdx1 = penal1*x1.^(penal1-1).*...(k(1)*x2.^penal2 + k(2)*(-x2).^penal2);
    dkdx2 = penal2*x1.^(penal1).*...
                 ((k(1)*x2.^(-penal2-1) - k(2)*(-x2).^penal2));
    Alpha = x2*alpha(1) + (1-x2)*alpha(2);
    dAlphax2 = alpha(1) - alpha(2);

%%%%% FE-ANALYSIS %%%

% THERMAL GOVERNING EQUATIONS

% Set parameters
Tsink = 50;  % Sink Temperature
q = 1.5;  % Heat flux
% Construct element stiffness matrix

Kel = [ 1/2-nu/6  1/8+nu/8  -1/4-nu/12  -1/8+3*nu/8 ... 
      -1/4+nu/12  -1/8-nu/8  nu/6  1/8-3*nu/8 ];

KE = 1/(1-nuˆ2)*[ Kel(1) Kel(2) Kel(3) Kel(4) Kel(5) Kel(6) Kel(7) Kel(8) 
                 Kel(2) Kel(1) Kel(8) Kel(7) Kel(6) Kel(5) Kel(4) Kel(3) 
                 Kel(3) Kel(8) Kel(1) Kel(6) Kel(7) Kel(4) Kel(5) Kel(2) 
                 Kel(4) Kel(7) Kel(6) Kel(1) Kel(8) Kel(3) Kel(2) Kel(5) 
                 Kel(5) Kel(6) Kel(7) Kel(1) Kel(2) Kel(3) Kel(4) Kel(8) 
                 Kel(6) Kel(5) Kel(4) Kel(7) Kel(1) Kel(2) Kel(1) Kel(8) 
                 Kel(7) Kel(4) Kel(5) Kel(2) Kel(3) Kel(8) Kel(1) Kel(6) 
                 Kel(8) Kel(3) Kel(2) Kel(5) Kel(4) Kel(7) Kel(6) Kel(1) ];

% Construct element conductivity matrix

KEth = (1/6)*[ 4 -1 -2 -1 
               -1 4 -1 -2 
               -2 -1 4 -1 
               -1 -2 -1 4 ];

% Element coupling matrix (thermal expansion)

CEthm = 1/(6*(1-nu))*[ -2 -2 -1 -1; 
                        -2 -1 -1 -2; 
                        2 2 1 1; 
                        1 1 2 2; 
                        1 2 2 1; 
                        -1 -1 -2 -2; 
                        2 1 1 2 ];

% Nodes at the boundary of the domain

topth = 1:nely+1:(nely+1)*nelx+1;
bottomth = nely+1:nely+1:(nely+1)*(nelx+1);
leftth = 1:nely+1;
rightth = (nely+1)*nelx+1:(nely+1)*(nelx+1);

fixeddofsth = [leftth,rightth,topth,bottomth];
alldofsth = 1:(nelx+1)*(nely+1);
freedofsth = setdiff(alldofsth,fixeddofsth);

% Boundary conditions
Kth = sparse((nely+1)*(nelx+1), (nely+1)*(nelx+1));
Fth = 0*ones((nelx+1)*(nely+1),1);

for ii=1:length(leftth)
    dof=leftth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=0;
end

for ii=1:length(rightth)
    dof=rightth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=0;
end

for ii=1:length(topth)
    dof=topth(ii);
    % Kth(dof,:) =0;
    % Kth(dof,dof) =1;
    Fth(dof,1)=q;
end

for ii=1:length(bottomth)
    dof=bottomth(ii);
    Kth(dof,:) =0;
    Kth(dof,dof) =1;
    Fth(dof,1)=Tsink;
% Build Conductivity Matrix:
for elx = 1:nelx
    for ely = 1:nely
        if ely==1 % Set high conductivity material on top
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1)* elx +ely;
            edof = [n1+1; n2+1; n2; n1];
            Kth(edof,edof) = Kth(edof,edof) + 10*KEth;
        else % Set normal material elsewhere
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1)* elx +ely;
            edof = [n1+1; n2+1; n2; n1];
            Kth(edof,edof) = Kth(edof,edof) + kx(ely,elx)*KEth;
        end
    end
end
Kth = (Kth+Kth')/2;

% MECHANICAL SYSTEM

% Define free and fixed degrees of freedoms
fixeddofsm = union(1:2:2*(nely+1)-1, 2*(nely+1)*(nelx+1));
alldofsm = 1:2*(nely+1)*(nelx+1);
freedofsm = setdiff(alldofsm,fixeddofsm);

Km = sparse(2*(nely+1)*(nelx+1), 2*(nely+1)*(nelx+1)); % Stiffness Matrix
Fp = sparse(2*(nely+1)*(nelx+1),1);
Fm = sparse(2*(nely+1)*(nelx+1),1); % Vector of nodal forces
Cthm = zeros(2*(nely+1)*(nelx+1), (nely+1)*(nelx+1)); % Coupling Matrix
dCthm = zeros(2*(nely+1)*(nelx+1), (nely+1)*(nelx+1));
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
edof4 = [n1+1; n2+1; n2; n1];
edof8 = [2*n1+1; 2*n1+2; 2*n2+1; 2*n2+2; 2*n2-1; 2*n2; 2*n1-1; 2*n1];
Km(edof8,edof8) = Km(edof8,edof8) + E(ely,elx)*KE;

Cthm(edof8,edof4) = Cthm(edof8,edof4) + E(ely,elx)*Alpha(ely,elx)*CEthm;
end
end
Km = (Km+Km')/2;

% DEFINE LOADS
Fm(2*(nely+1)*floor((nelx+1)/2)+2,1) = -1;
Fm(2,1) = -1;
Fm(2*(nely+1)*(nelx)+2,1) = -1;

% SOLVING using Newton's method
[sol, it_hist, ierr, x_hist] = nsold(y0,'GovEq_thm',tol,parms);

jacM = [Km, Cn; Constd, Constn];
Cthm_2 = [Cthm; zeros(nn,nth)];
jac = [jacM, -Cthm_2; Kthmjac, Kth_sol];
nullmat = zeros(nth,nd+nn);
Hjac = [jacM, -Cthm_2; nullmat, Kth_sol];
y = sol;

Um(:,1) = y(1:nd);
Un(:,1) = y((nd+1):(nd+nn));
Uth(:,1) = y((nd+nn)+1:end);
Pn(:,iter) = Un;
Um(fixeddofs,1)= 0;
Kth = Kth_sol;

% PLOT FIELD VARIABLES
if plott==1
    Disp_x = zeros(nely+1,nelx+1);
    Disp_y = zeros(nely+1,nelx+1);
for j=1:nelx+1
    for i=1:nely+1
        Disp_x(i,j)=Um(2*(nely+1)*(j-1)+2*i-1);
        Disp_y(i,j)=Um(2*(nely+1)*(j-1)+2*i);
    end
end

xx=1:nelx+1;
yy=-(1:nely+1);

% Plot final Ux displacement
figure(1);
contourf(xx,yy,Disp_x,50)
% surf(yy,xx,Disp_x);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')
colorbar;

% Plot final Uy displacement
figure(2);
contourf(xx,yy,Disp_y,50)
% surf(yy,xx,Disp_y);
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')
colorbar;

T = zeros(nely+1,nelx+1);
for j=1:nelx+1
    for i=1:nely+1
        T(i,j)=Uth((nely+1)*(j-1)+i);
    end
end

figure(3);
```matlab
xx=1:nelx+1;
yy=1:nely+1;
contourf(xx,yy,T,50);
axis image;
ylabel('x');
xlabel('y');
title('Temperature distribution')
colorbar;
end

% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS

% SOLVE ADJOINT EQUATIONS
lamm = zeros(2*(nely+1)*(nelx+1),1);
Flam = Fm + Cthm*Uth;

% lamm(freedofsm,:) = - Km(freedofsm,freedofsm) \ Flam(freedofsm,:);
lamm = - Um;
lamm(fixeddofsm,:) = 0;

lamth = Kth \ ...
(lamm'*dCthm-Um'*dCthm)';

% PREPARE SENSITIVITY ANALYSIS
f0val = 0;
df0dx_mat = zeros(nely,nelx);
xval1 = reshape(x1,n,1);
xval2 = reshape(x2,n,1);
xval = [xval1; xval2];

% DEFINE CONSTRAINTS
volconst1 = sum(sum(x1))/n - volfrac(1);
fval(1,1) = volconst1;
dfdx1(1,:) = ones(1,n)/(n);
dfdx2(1,:) = zeros(1,n);
if m >=2
    volconst2 = sum(sum(x1.*x2))/(n) - volfrac(2);
end
```
fval(2,1) = volconst2;
dfdx1(2,:) = xval2/(n);
dfdx2(2,:) = xval1/(n);
if m >= 3
    volconst3 = sum(sum(x1.*(1-x2)))/(n) - (1-volfrac(2));
    fval(3,1) = volconst3;
    dfdx1(3,:) = (1-xval2)/(n);
    dfdx2(3,:) = -xval1/(n);
end
end
dfdx = [dfdx1 , dfdx2];
df0dx1_mat = zeros(nely,nelx);
df0dx2_mat = zeros(nely,nelx);

% CALCULATE SENSITIVITIES
for elx = 1:nelx
    for ely = 1:nely
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        edof4 = [n1+1; n2+1; n2; n1];
        edof8 = [2*n1+1; 2*n1+2; 2*n2+1; 2*n2+2; 2*n2-1; 2*n2; 2*n1-1; 2*n1];
        Ume = Um(edof8,1);
        Uthe = Uth(edof4,1);
        lamme = lamm(edof8,1);
        lamthe = lamth(edof4,1);
        f0val = f0val + E(ely,elx)*Ume'*KE*Ume;
        df0dx1_mat(ely,elx) = ...
        dEdx1(ely,elx)*Alpha(ely,elx)*Ume'*CEthm*Uthe + ...
        lamme'*(...
        dEdx1(ely,elx)*KE*Ume - ...
        dEdx1(ely,elx)*Alpha(ely,elx)*CEthm*Uthe) + ...
        lamthe'*(dkdx1(ely,elx)*KEth*Uthe);
\[
\text{df0dx2} = \begin{bmatrix} \text{df0dx2}(\text{ely,elx}) \\
\end{bmatrix}
\]

\[
\begin{align*}
(\text{dEdx2}(\text{ely,elx}) \cdot \text{Alpha}(\text{ely,elx}) + \text{E}(\text{ely,elx}) \cdot \text{dAlphax2}) \cdot ... \\
\text{Ume}' \cdot \text{C Ethm} \cdot \text{Uthe} + ... \\
\text{lamme}' \cdot (... \\
\text{dEdx2}(\text{ely,elx}) \cdot \text{KE} \cdot \text{Ume} - ... \\
(\text{dEdx2}(\text{ely,elx}) \cdot \text{Alpha}(\text{ely,elx}) + \text{E}(\text{ely,elx}) \cdot \text{dAlphax2}) \cdot ... \\
\text{CEthm} \cdot \text{Uthe}) + ... \\
\text{lamthe}' \cdot (\text{dkdx2}(\text{ely,elx}) \cdot \text{KEth} \cdot \text{Uthe});
\end{align*}
\]

\[
\text{end}
\]

\[
\text{end}
\]

\[
\text{df0dx1} = \text{reshape(df0dx1, nely*nelx, 1)};
\]

\[
\text{df0dx2} = \text{reshape(df0dx2, nely*nelx, 1)};
\]

% FILTERING OF SENSITIVITIES

\[
\begin{align*}
\text{df0dx1} &= \text{H} \cdot (\text{xval}(1:n) \cdot \text{df0dx1}) / \text{Hs} / \text{max}(1e-3, \text{xval}(1:n)); \\
\text{df0dx2} &= \text{H} \cdot (\text{df0dx2}) / \text{Hs}; \\
\text{df0dx} &= [\text{df0dx1}; \text{df0dx2}];
\end{align*}
\]

% UPDATE DESIGN VARIABLES USING MMA

\[
\begin{align*}
[\text{xmma}, \text{ymma}, \text{zmma}, \text{lam}, \text{ksi}, \text{eta}, \text{mu}, \text{zet}, \text{s}, \text{low}, \text{upp}] &= ... \\
\text{mmasub}(m, 2*n, \text{iter}, \text{xval}, \text{xmin}, \text{ xmax}, \text{ xold1}, \text{ xold2}, ...) \\
&\quad \text{fval, df0dx, fval, dfdx, low, upp, a0, a, c, d, pasv, valx);}
\end{align*}
\]

% Store previous density variables for next iteration

\[
\text{if iter} > 2 \\
\text{xold2} = \text{xold1}; \\
\text{xold1} = \text{xval};
\]

\[
\text{else if iter} > 1 \\
\text{xold1} = \text{xval};
\]

\[
\text{end}
\]

\[
\text{x1} = \text{reshape(xmma(1:n), nely, nelx)};
\]

\[
\text{x2} = \text{reshape(xmma(n+1:end), nely, nelx)};
\]

\[
\text{xmat1} = \text{x1} \cdot \text{x2};
\]

\[
\text{xmat2} = \text{x1} \cdot (1 - \text{x2});
\]
xvoid = 1 - xmat1 - xmat2;
xplot = zeros(nely,nelx,3);
xplot(:,:,1) = xvoid;
xplot(:,:,2) = xmat1;
xplot(:,:,3) = xmat2;

% PRINT RESULTS
change1 = max(max(abs(xmma(1:n)-xold1(1:n))));
change2 = max(max(abs(xmma(n+1:2*n)-xold1(n+1:2*n))));
change1_evolve(iter) = change1;
change2_evolve(iter) = change2;
change = max(change1);,change2);
change_evolve(iter) = change;
OBJ(iter) = f0val;
disp([ ' It.: ' sprintf('%4i',iter) ' Obj.: ' sprintf('%10.4f',f0val) ... 
    ' Vol.: ' sprintf('%6.3f',sum(sum(xplot))/(n)) ... 
    ' ch.: ' sprintf('%6.3f',change )])

% PLOT DENSITIES
I = make_bitmap (3,nelx,nely,xplot);
figure(4);
image(I), axis image off, drawnow;

end

%% PLOT RESULTS %%

%%%%%%%%%%%%%%%% CONVERGENCE AND DENSITIES PLOTS %%%%%%%%%%%%%%%%%%%%%%%%%
figure();
plot(l:iter,OBJ,'LineWidth',1,...
    'Marker','o',...
    'MarkerSize',3,...
    'MarkerFaceColor','b');
ylabel('c (obj. function)');
xlabel('iteration');
% title('Convergence of Objective function')
figure();
plot(1:iter,change_evol,'LineWidth',1,...
    'Marker','o',...'
    'MarkerSize',3,...
    'MarkerFaceColor','b');
ylabel('Max( \rho_{new} - \rho_{old} | )');
xlabel('iteration');

%%%% PREPARE THE DATA TO PLOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Mechanical model
nd = length(Um);

% B matrix in FEM (Derivatives of the shape functions
B = [-4 0 4 0 4 -4 0;
     0 -4 0 -4 0 4 0 4;
     -4 -4 -4 4 4 4 4 -4];

% Initialize vectors of future plot
SigmaX = zeros(nely,nelx); % Stress in x deirection
SigmaY = zeros(nely,nelx); % Stress in y deirection
SigmaXY = zeros(nely,nelx); % Shear stress
XY_undeformed = zeros((nely+1)*(nelx+1),2); % Undeformed structure representation (node coordinates)
XY_deformed = XY_undeformed; % Deformed structure representation (node coordinates)
Nodes = zeros((nely+1)*(nelx+1)); % Deformed structure representation (node coordinates)
Udisp = zeros(nd/2,2); % Vector of displacement's norm.
x_deformed = zeros(1,nely*nelx); % Vector of density variables.

init_pos_x = zeros(nelx+1,1); % Initial (horizontal) position of the free structure
init_pos_y = zeros(nely+1,1); % Initial (vertical) position of the free structure
Lx = nelx; % Horizontal length of the domain
Ly = nely; % Vertical length of the domain
for j = 1:nelx
  init_pos_x(j+1) = Lx*j/(nelx);
end
for i = 1:nely
  init_pos_y(nely+1-i) = Ly*i/(nely);
end
for j=1:nelx+1
  for i=1:nely+1
    if j<nelx+1 && i<nely+1
      n1 = (nely+1)*(j-1)+i;
      n2 = (nely+1)* j +i;
      Ue = Um([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
      stress = B*Ue;
      SigmaX(i,j) = stress(1);
      SigmaY(i,j) = stress(2);
      SigmaXY(i,j) = stress(3);
      clear stress Ue;
    end
    Disp_x(i,j)=Um(2*(nely+1)*(j-1)+2*i-1);
    Disp_y(i,j)=Um(2*(nely+1)*(j-1)+2*i);
  end
  XY_undeformed((nely+1)*(j-1)+i,:) = [ init_pos_x(j) , init_pos_y(i) ];
  XY_deformed((nely+1)*(j-1)+i,:) = ...
    XY_undeformed((nely+1)*(j-1)+i,:) + ...
    [ Disp_x(i,j), Disp_y(i,j) ];
  Udisp((nely+1)*(j-1)+i,:) = [ Disp_x(i,j), Disp_y(i,j) ];
  Unorm((nely+1)*(j-1)+i) = (Disp_x(i,j)^2 + Disp_y(i,j)^2)^(1/2);
  if i<nely+1 && j<nelx+1
    n1 = (nely+1)*(j-1)+i;
    n2 = (nely+1)* j +i;
    Nodes(n1,n1+1) = 1;
    Nodes(n1+1,n2+1) = 1;
Nodes(n2+1,n2) = 1;
Nodes(n2,n1) = 1;
x_deformed((nely)*(j-1)+i) = xvoid(i,j);
end
end
end

xx=1:nelx+1;
yy=-(1:nely+1);

SigmaVM = ( SigmaX.^2 + SigmaY.^2 - SigmaX.*SigmaY + 3*SigmaXY.^2 ).^(1/2);
DispNorm = ( Disp_x.^2 + Disp_y.^2 ).^(1/2);

%%%%%%%%%%%%%%%%%%%%% PLOT DISPLACEMENTS %%%%%%%%%%%%%%%%%%%%%%%%%

% Plot final Ux displacement
figure();
contourf(xx,yy,Disp_x,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Ux')

% Plot final Uy displacement
figure();
contourf(xx,yy,Disp_y,50)
axis image;
ylabel('y');
xlabel('x');
title('Virtual Displacement Uy')

% Define the structure (elements with a minimum of 50% of material)
listallnodes = zeros((nely)*(nelx),4);
countint = 0;
for elx=1:nelx
    for ely=1:nely
\[ n_1 = (n_{ely}+1) \cdot (elx-1) + ely; \]
\[ n_2 = (n_{ely}+1) \cdot elx + ely; \]
\[ \text{if } \text{xvoid}(ely,elx) < 0.5 \]
\[ \text{countint} = \text{countint} + 1; \]
\[ \text{nodes}(\text{countint},1:4) = [n_1; n_2; n_2+1; n_1+1]; \]
\[ \text{end} \]
\[ \text{listallnodes}((n_{ely}) \cdot (elx-1) + ely,1:4) = [n_1; n_2; n_2+1; n_1+1]; \]
\[ \text{end} \]

% generation of coordinates and connectivities
coordinates = [XY_{deformed(:,1)} \ XY_{deformed(:,2)}];
coordinates2 = [(1:length(XY_{undeformed}))' \ XY_{undeformed}];
nodes2 = [(1:length(nodes))' \ nodes];

% Plot displacement U_y in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_{deformed},Udisp(:,2))
title('Profile of U_y on deformed Mesh')
axis image;

% Plot displacement U_x in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_{deformed},Udisp(:,1))
title('Profile of U_x on deformed Mesh')
axis image;

% Plot displacements norm \|U\| in deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_{deformed},Unorm)
title('Profile of \|U\| on deformed Mesh')
axis image;

% Thermal model
T = zeros(nely+1,nelx+1);
% Plot temperature distribution
for j=1:nelx+1
    for i=1:nely+1
        T(i,j)=Uth((nely+1)*(j-1)+i);
    end
end

figure();
xx=[1:nelx+1];
yy=[1:nely+1];
contourf(xx,-yy,T,30)
axis image;
ylabel('y');
xlabel('x');
title('Temperature distribution')

% Plot temperature distribution on deformed shape
PlotFieldonDefoMesh(coordinates2,nodes2,1,XY_deformed,Uth)
title('Temperature Profile on deformed Mesh')
axis image;

%%%%%%%%%%%%%%%%%%%%%% PLOT DEFORMED TOPOLOGY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
allnodes2 = [(1:length(listallnodes))' listallnodes];
PlotDesignMesh(coordinates2,allnodes2,1,XY_deformed,xplot)
title('Deformed Design')
axis image;

%%%%%%%%%%%%%%%%%%%%%%%%% SAVE WORKSPACE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
saving = input([' *** Do you want to save the workspace as a' ...
    '.mat file in the working directory? *** (y for yes) => '],'s');
if saving == 'y'
    filename = input([' *** Insert name xxxxxx of the file' ...
         '(format: xxxxxx.mat) *** => '],'s');
    fname = pwd;
GovEq.thm Subroutine

function [ f, jac ] = GovEq.thm( y )

% by Pierre Thurier, <pfthurier@gmail.com>
Last Update: July 2014
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% INPUT : The current solution given by the Newton's search.

% OUTPUT : f - The value of the objective function f to be minimized
jac - Jacobian of the governing equation problem

%}

%global Km Cn Constd Constn Fm gA nd nn nth Cndof fixeddofs TCC0 Tsink...
Cthm Kth Fth Kthjac Cndofth Kth_sol bottomth

yd = y(1:nd); % Displacements
yth = y((nd+nn)+1:end); % Temperatures
nullmat = zeros(nth,nd+nn);
Kthjac = nullmat; % Matrix of Derivatives with respect to the normal forces
Kth_sol = Kth;

% Beta = 0.95;
Beta = 0.95;
nA = -1; % Normal surface vector of the body

if nn > 0

yn = y((nd+1):nd+nn); % Normal forces
ydA = yd(Cndof); % Location of nodes in possible contact
r = 1e6; % Numerical factor
gi = yn;

% CONTACT CONDITION
for ii = 1:nn
    contact = r*(nA*ydA(ii) + gA(ii)); % contact
    if contact >= 0 % If contact is made
        Constd(ii,Cndof(ii)) = r; % Constrain displacement to zero
        Constn(ii,ii) = 0; % No constraint for the contact force
        gi(ii) = r*gA(ii);
        % disp(['Constn=0, iter=', num2str(ii)])
    else
        Constd(ii,Cndof(ii)) = 0; % No constraint for the displacement
        Constn(ii,ii) = -1; % Constrain the contact force to zero
        gi(ii) = 0;
        % disp(['Constn=-1, iter=', num2str(ii)])
    end
end

% THERMAL CONTACT MODEL
for ii=1:nn
    dof=Cndofth(ii);
    if Constn(ii,ii) == 0
        if yn(ii)<0 % Avoid negative normal forces
            Constn(ii,ii) = -1;
            Fth(dof,1)=0;
        else % Implement thermal contact
            kth_sol(dof,:) = 0;
        end
    end
Kth_sol(dof,dof) = 1;
Fth(dof,1) = Tsink;

% Kth_sol(dof,dof) = Kth_sol(dof,dof) + (TCC0*yn(ii)^Beta);
% Fth(dof,1) = + Tsink * (TCC0*yn(ii)^Beta);
% Kthmjac(dof,nd+ii) = + TCC0*Beta*(yth(dof)-Tsink)*yn(ii)^(Beta-1);

end

else
    Fth(dof,1) = 0;  % If no contact, set insulated BC
end

end

% If no contact is made at the bottom

if Constn == -eye(nn)
    for ii=1:nn
        dof=Cndofth(ii);
        Kth_sol(dof,:) =0;
        Kth_sol(dof,dof) =1;
        Fth(dof,1) = Tsink;
    end
    Kthmjac = zeros(nth,nd+nn);
    disp('no contact at the bottom boundary')
end

else  % If no contact condition considered
    for ii=1:length(bottomth)
        dof = bottomth(ii);
        Kth_sol(dof,:) =0;
        Kth_sol(dof,dof) =1;
        Fth(dof,1) = Tsink;
    end
    disp('no contact model')
end

Km(fixeddofs,:) = 0;
\begin{verbatim}
Km(fixeddofs,fixeddofs) = eye(length(fixeddofs),length(fixeddofs));

% FORMULATION OF THE GOVERNING EQUATIONS
if nn > 0
    FH = [ Fm ; gi ; Fth ];
    jacM = [ Km , Cn ; Constd , Constn ];
    Cthm_2 = [Cthm ; zeros(nn,nth)];
    jac = [ jacM , -Cthm_2 ; Kthmjac , Kth_sol ];
    Hjac = [ jacM , -Cthm_2 ; nullmat , Kth_sol ];
else
    FH = [ Fm ; Fth ];
    y = [ yd ; yth ];
    jacM = Km;
    jac = [ jacM , -Cthm ; Kthmjac , Kth_sol ];
    Hjac = [ jacM , -Cthm ; nullmat , Kth_sol ];
end

f = Hjac*y-FH;
end
\end{verbatim}
Bibliography


[27] Li, D., Zhang X. Guan Y. Zhan J., “Topology optimization of thermo-mechanical continuum structure,” IEEEASME International Conference on Advanced Intelligent Mechatronics AIM, pp. 403–408, July 2010


[38] Strömberg, N. and Klarbring, A., “Topology optimization of structures in unilateral contact,” *Structural and multidisciplinary optimization*, volume 41, no. 1, pp. 57–64, February 2010


