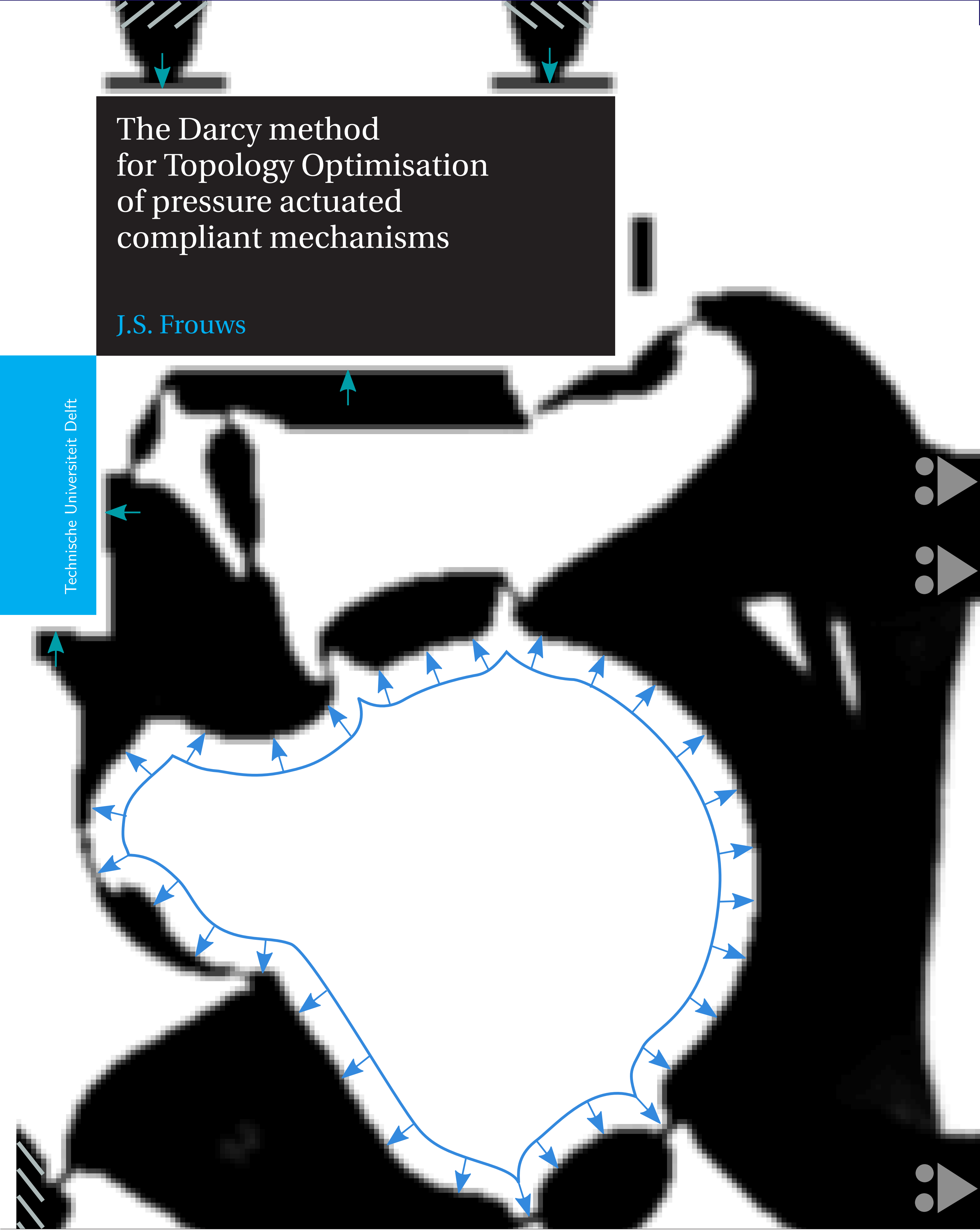
The Darcy method for Topology Optimisation of pressure actuated compliant mechanisms

J.S. Frouws

Technische Universiteit Delft



TU Delft Delft University of Technology

Challenge the future

Department of Precision and Microsystems Engineering

The Darcy method for Topology Optimisation of pressure actuated compliant mechanisms

J.S. Frouws

Report no	: 2017.060
Supervisor	: dr. ir. M. Langelaar
Specialisation	: Engineering Mechanics (EM)
Type of report	: Master Thesis
Date	: 2017-12-01

The Darcy method for Topology Optimisation of pressure actuated compliant mechanisms

by

J.S. Frouws

to obtain the degree of Master of Science at the Delft University of Technology,
to be defended publicly on Tuesday December, 19 2017 at 1:00 PM.

Student number:	4334094	
Project duration:	December 2016 – December 2017	
Supervisor and chair:	dr. ir. Matthijs Langelaar,	TU Delft
Committee members:	dr. ir. Marcel Tichem	TU Delft
	dr. ir. Prabhat Kumar	TU Delft
	ir. Joep Nijssen,	TU Delft

This thesis is confidential and cannot be made public until June 19, 2018.

Table of contents

Acknowledgements

To my beloved Iris, for her unconditional care and support.

Dear reader,
first of all my sincere thanks to you, who appears to be tempted to read this thesis work or who seeks its name in this page of fame. Maybe you want to continue on this wonderful topic and experience the same pleasure I had, discovering the vast world of Topology Optimisation! Especially after having to start at the bottom with the advise to do VMBO-b, but I was probably stupid back then.

During my thesis I had many help from different angles. First, I would like to thank Iris for always giving me a listening ear, for her loving care in busy times and her unconditional support in all the frustrating times. And I should mention the dishes, I guess. I also want to thank my family: mama Ria, for the always proud and enthusiastic calls which always left me in a better mood! My twin, Siete, for the time and effort you put in helping me with the Matlab coding and for the enthusiasm with which we could talk about my graduation topic! Thanks also for the remote access to your computer that designed up to half the content of this report. Hopefully more whiskeys will flow in the future. My loving sis Ankje, at this point a master in writing reports (and marine biology, and soon to be married to a Kenyan), thanks for reading my thesis and providing me with a lot of feedback! It was nice to have you in the Netherlands for a while. Then there is my big brother, Marten, thanks for the distractions now and then by helping me ride dinosaurs and build a Frouws dynasty, next time stay off the narcotics. Thanks also for building the 'super-' computer that is responsible for designing the other half of the designs that are found in this report. At last, I would like to thank my father for his support during my academic career and the nice lunches we had together.

Next, being the only one who could remotely follow my mumbling, Matthijs. You have my gratitude for the good conversation, guidance and checking my work so thoroughly. You were an excellent supervisor to me and I thank you for all your effort. I would also like to thank Prabhat for the nice conversations and the wish to continue on this work, and Emiel for helping me get my TO working and in the derivation of the sensitivities!

Throughout the master I always found myself in the company of like-minded students, real β's. Irene, thanks for helping me throughout the courses and checking the structure of my report! Sarah, you are always a delight, it was a pleasure to sit in your office and take an afternoon walk every day. Thanks also for taking a close look at my introduction!

This is by far not all, it would be more complete if I extend my gratitude to all my fellow master students of PME for the good times, and the Leidsche Instrument makers School (LIS) with its great teachers that gave me the opportunity to build a large appreciation for technology, mathematics and engineering.

Finally my sincere thanks to Eveline Matroos (the master coordinator) for caring about the students and having the desire to make life easier for us by being helpful e.g. by taking over part of the paperwork. I am glad to have you as the new face of the department!

Abstract

This work introduces a new method to deal with design dependent pressure loads in Topology Optimisation (TO) using the SIMP material model. A pronounced focus is on optimising pressure actuated compliant mechanisms. The difficulty herein is the interpretation of the pressure boundary in a TO design. In TO the boundaries are blurry, because of the filtering of the design variables, which is necessary to prevent checkerboarding. Another reason why the boundary is poorly defined is that the optimisation starts from an equally distributed grey, where black and white respectively are solid and void, so the boundary is either the domain boundary or not defined in early iterations of the optimisation. The methods proposed in literature often try to find the void-solid interface exposed to the pressure source to apply the loading from a pressure line directly. The method proposed in this work, appropriately called the Darcy method, first calculates the pressure field by using Darcy's law governing the flow through porous media and the associated pressure drop. A flow coefficient is introduced that decreases if the virtual element density increases. This results in a design dependent pressure field that can be solved using the Finite Element Method (FEM), which can then be translated to consistent nodal forces that are applied to the TO problem. A drainage coefficient has also been introduced to make sure that the pressure is drained entirely to the environment pressure over the first encountered void-solid interface exposed to the pressure source. The Darcy method has proven to function well in several test cases. The method has been thoroughly tested using several parameter sweeps on a clamping problem objective. The parameters whose influence is examined are: the initial condition, the density threshold value, flow coefficient gradient at the threshold, output spring stiffness and the volume fraction. Subsequently, some alternative TO problems are solved showing the diversity of the method. In perspective of future research, the Darcy method can function as a great tool to research load sensitivities by tuning the pressure field control parameters. The extension of the Darcy method to 3D or to several load cases comes naturally but has not been tested in this work, this is also recommended for future research.

Keywords: Topology Optimisation, design dependent loading, pressure load, Darcy's law,
compliant mechanisms, soft robotics.

Nomenclature

Symbol	Description	Unit
β_k	Parameter to control the rate of change of the flow coefficient with respect to ρ_e at $\rho_e = \eta_k$	-
$\boldsymbol{\eta}$	$= [\xi, \eta]^\top$ Coordinates in reference frame	-
η_k	Smooth density threshold to control when the flow coefficient changes from k_{void} to k_{mat}	-
κ	Permeability of a material	m^2
μ	Fluid viscosity	Pa s
ν	Parametrisation to perform a boundary integral with	m
$\boldsymbol{\rho}$	Vector of virtual densities, $[n_\text{e} \times 1]$	-
ρ_d	Virtual density of the domain not prescribed as an initial design of density ρ_i	-
ρ_e	Virtual density of element e	-
ρ_i	Virtual density of the domain prescribed as an initial design	-
ρ_T	Density threshold value (on which the pressure boundary is defined)	-
ψ	Energy efficiency value Equation (4.12)	-
$\mathbf{A}$	Global flow matrix, $[n \times n]$	$\text{m}^5\,\text{N}^{-1}\,\text{s}^{-1}$
$\boldsymbol{b}$	Body forces acting on the domain	$\text{N}\,\text{m}^{-3}$
$\mathbf{B}_p(\boldsymbol{\xi})$	$= \boldsymbol{N}_{,\boldsymbol{x}}$ Matrix of spatial derivatives of the vector of shape functions	m^{-1}
$\mathbf{B}_u(\boldsymbol{\xi})$	Matrix of spatial derivatives of the matrix of shape functions	m^{-1}
C	Objective function, could be: F_{out}, ψ or C (compliance), for respectively the OF, EE or MC objectives	
$\boldsymbol{f}$	Global 'volume flow' loading vector, $[n \times 1]$	$\text{m}^3\,\text{s}^{-1}$
$\boldsymbol{f}_{gp}$	Global contribution of prescribed pressures $[n \times 1]$	$\text{m}^3\,\text{s}^{-1}$
$\boldsymbol{F}$	Consistent nodal force vector, $[nd \times 1]$	N
$\boldsymbol{F}^e$	Consistent element load vector, $[8 \times 1]$	N
F_{out}	Output force of the structure, which equals the reaction force of the virtual spring	N
g_j	The j'th Constraint function, each is to remain below zero	-
h_{mat}	Drainage coefficient on material elements (where $\rho_e = 1$)	$\text{m}^2\,\text{N}^{-1}\,\text{s}^{-1}$
h_{void}	Drainage coefficient on void elements (where $\rho_e = 0$)	$\text{m}^2\,\text{N}^{-1}\,\text{s}^{-1}$
$\mathbf{H}$	Global conversion matrix from nodal pressures $\boldsymbol{p}$ to the nodal loads $\boldsymbol{F}$, $[nd \times n]$	m^2
$H(\rho_e)$	Drainage coefficient	$\text{m}^2\,\text{N}^{-1}\,\text{s}^{-1}$
$\mathbf{J}$	Jacobian matrix	-
J	Determinant of the Jacobian matrix	-
k_{mat}	Flow coefficient on material elements (where $\rho_e = 1$)	$\text{m}^4\,\text{N}^{-1}\,\text{s}^{-1}$
k_{void}	Flow coefficient on void elements (where $\rho_e = 0$)	$\text{m}^4\,\text{N}^{-1}\,\text{s}^{-1}$
$\mathbf{K}$	Global stiffness matrix, $[nd \times nd]$	$\text{N}\,\text{m}^{-1}$
$K(\rho_e)$	$= \frac{\kappa}{\mu}$, Flow coefficient	$\text{m}^4\,\text{N}^{-1}\,\text{s}^{-1}$
$\boldsymbol{n}(\boldsymbol{\xi})$	Unit normal vector, normal to a boundary Γ in the $\boldsymbol{\xi}$ domain	-
$\boldsymbol{n}(\boldsymbol{x})$	Boundary normal vector	-
$\boldsymbol{N}(\boldsymbol{\xi})$	Vector of Shape functions	-
$\mathbf{N}(\boldsymbol{\xi})$	Matrix of Shape functions	-
$N_i(\boldsymbol{\xi})$	Individual shape function, it is 1 at local node i and zero on the others	-
$\boldsymbol{p}^e$	Local nodal pressure values on element e	$\text{N}\,\text{m}^{-2}$
p_{E}	Penalisation factor in the SIMP material model	-
p_{in}	Input pressure at pressure source	$\text{N}\,\text{m}^{-2}$
p_{out}	Environmental pressure, in the aeronautical branch often called: p_∞ (meaning, far enough from the object, such that the object does not influence it)	$\text{N}\,\text{m}^{-2}$
$p(\boldsymbol{x})$	Pressure field, defined continuously and globally	$\text{N}\,\text{m}^{-2}$

p_0	Environmental pressure, here $p_0 = 0$	$\mathrm{N\,m^{-2}}$
$\boldsymbol{q}(\boldsymbol{x})$	$= -\frac{\kappa}{\mu}\nabla p$ Volume velocity through porous medium (or heat flux)	$\mathrm{m\,s^{-1}}$
$\boldsymbol{q}_\Gamma$	prescribed Darcy flux at the boundary	$\mathrm{m\,s^{-1}}$
Q	Volume flow, fluid volume per second	$\mathrm{m^3\,s^{-1}}$
$\tilde{Q}_{\text{drain}}$	Fluid volume drainage per second per unit volume	$\mathrm{s^{-1}}$
Q^e_{in}	Volume flowing into the infinitesimal element, per second	$\mathrm{m^3\,s^{-1}}$
Q_{out}	Volume outflow (or flowing out) through the outer boundary Γ_{p_0} of the domain, per second	$\mathrm{m^3\,s^{-1}}$
Q^*_{out}	Maximum volume flow through the outer boundary	$\mathrm{m^3\,s^{-1}}$
Q^e_{out}	Volume flowing out of the infinitesimal element, per second	$\mathrm{m^3\,s^{-1}}$
r	Fraction of input pressure at penetration depth Δs	-
Δs	Penetration distance in material at which the drainage term drained the pressure to $p(\Delta s) = r\,p_{\text{in}}$	m
$\boldsymbol{S}$	Vector of sensitivities	-
S_e	Sensitivity of element e, $S_e = \frac{\partial C}{\partial \rho_e}$	-
t	Thickness of the element in a 2D problem statement, in a 3D problem, Δz is used.	m
T	Temperature	K
$\boldsymbol{u}$	Vector of nodal displacements, $[nd \times 1]$	m
$V^*_{\max}$	Maximum volume fraction in the design domain	-
ΔV	Element volume	$\mathrm{m^3}$
$\boldsymbol{x}$	$= [x,\ y]^\top$ Coordinates in global frame	m
$\boldsymbol{x}^e(\boldsymbol{\xi})$	Coordinates of element e in global frame	m
$\boldsymbol{x}^e_n$	coordinates of the nodes of element e used to map $\boldsymbol{x}$ on $\boldsymbol{\xi}$ and define the Jacobian, $[8 \times 1]$	m
Δx	Element width	m
Δy	Element height	m
Δz	Element thickness, or in 3D, the element depth	m
∇	Spatial gradient operator	$\mathrm{m^{-1}}$

Index variables

e	Element index parameter
i	General index parameter
$\mathbf{L}^e_p$	Global indexing matrix, $[4 \times n]$
$\mathbf{L}^e_u$	Global indexing matrix, $[4d \times nd]$
n	Number of nodes in the domain
n_{e}	Number of elements in the domain

Sets and domains (as defined in Section 2.1.2)

Γ	$\Gamma_p \cup \Gamma_{p_0} \cup \Gamma_u \cup \Gamma_{\text{s}}$ Complete boundary
Γ_u	Boundary with fixed displacements
Γ_p	Boundary of applied pressure (where$p = p_{\text{in}}$)
Γ_{p_b}	Void-solid interface exposed to the pressure source, location where pressure acts
Γ_{p_0}	Boundary where outflow pressure is defined as $p = p_{\text{out}}$
Γ_{s}	Boundary with symmetry conditions
Ω	$\Omega_{\text{v}} \cup \Omega_{\text{m}}$ Total domain
Ω_e	Reference element domain
Ω_{f}	Domain with fixed void or material (fixed design variables)
Ω_{m}	Domain with material ($\rho_e \approx 1$)
Ω_{v}	Domain with void ($\rho_e \approx 0$)
d	Number of spatial dimensions considered ($d = 2$ in this work, as all examples are in 2D)
$\mathbb{D}_e$	Set of n_{e} indices of all elements
$\mathbb{D}_p$	Set of n indices of all pressure DOF's
$\mathbb{D}_u$	Set of nd indices of all displacement DOF's (all n nodes have a DOF in d free spatial directions)

List of abbreviations

Abbreviations	**Description**
BIM	Boundary Identification Methods
D	Dimensional
DDL	Design dependent load
DOF('s)	Degree(s) of freedom
DRLSE	Distance Regularised Level Set Evolution
EE	Energy efficiency
FE	Finite element
FEA	Finite element analysis
FEM	Finite element method
LSF	Level set function
MA	Mechanical advantage
MC	Minimum compliance
MMA	Method of Moving Asymptotes
ODE	Ordinary differential equation
OF	Output force
PDE	Partial differential equation
SE	Strain energy
SF	Shape functions
SIMP	Solid Isotropic Material with Penalisation
TO	Topology Optimisation
WP	Working principle, the mechanical way of a compliant mechanism to satisfy its objective

List of Figures

1 Introduction

1.1. Motivation and goal of this thesis

Nowadays, there is a wide application for pressure driven actuators and mechanisms, either pneumatic or hydraulic. On a large scale, heavy lifting and digging machines often work with hydraulic cylinders and on a smaller scale, pneumatics are often used to guarantee the safe interaction with humans. Pneumatic actuators are by definition less stiff because of the compressibility of air and will naturally operate in a lower loading range with pressures of about $1 \times 10^5\,\mathrm{N\,m^{-2}}$. Hydraulic actuators can be extremely stiff and work with pressures in the order of $1 \times 10^8\,\mathrm{N\,m^{-2}}$.

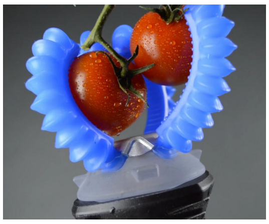


Figure 1.1: Soft gripper designed by Soft Robotics Inc.

A distinction can be made in the group of pressure driven mechanisms between *classic mechanisms*, with sliding components like a piston and cylinder, and *compliant mechanisms* that bend, deform and/or inflate to perform their task e.g. the finger like grippers in Figure 1.1. This thesis focusses on the latter. The pressure driven compliant mechanisms are commonly used in the field of soft robotics where, among others, they are deployed for the safe interaction with humans or handling vegetables without bruising them, as shown in Figure 1.1. However, research to the optimal shape (or topology) of such mechanisms has only been performed to a limited extend. Optimisation could bring us new designs that are more space and energy efficient. It might also find topologies, not thought of before and form a source of inspiration for the creative mind.

This thesis aims to construct a Topology Optimisation (TO) method that calculates the optimal design of such a pressure-driven compliant mechanism. Topology Optimisation is introduced in Section 1.3. A few of such methods already exist, but, as will be explained in Section 1.2, improvements can be made with respect to existing methods. The focus of this work is in particular on defining a design dependent and continuous pressure field $p(x, y)$ in order to improve the optimisation behaviour of a pressure actuated compliant mechanism. Additionally, this work shows an interest in finding single-walled design solutions common in the soft robotics field, like the inflatables in Figure 1.1.

Henceforth, the following **research questions** are posed:
1) Can a differentiable and globally defined pressure field formulation be used to perform a Topology Optimisation of a pressure actuated compliant mechanism?
2) Does this formulation improve the chance of finding a better optimum?
3) Can the parameters in this formulation be tuned such that soft robotic like[1] shapes can be extracted?

Optimisation of a pressure loaded structure introduces a few challenges. One being that the surface on which the pressure load acts is design dependent and not clearly defined in the popular TO method that is used in this work. In other words, the pressure acts on the unclear and evolving inner boundary of the design. As a

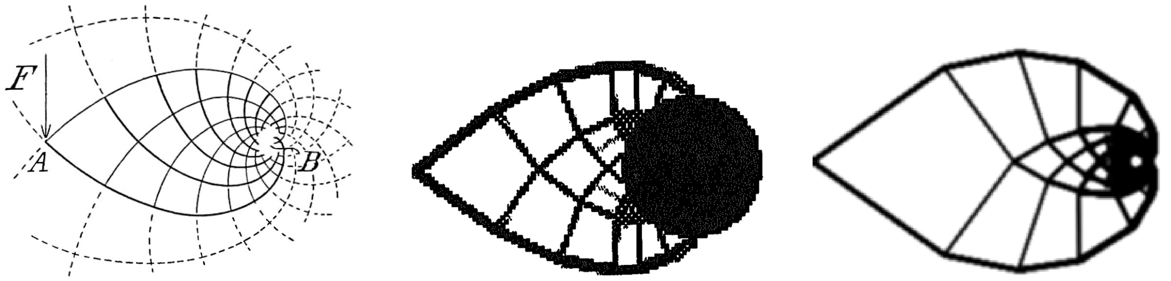
[1] The field of soft robotics often uses tentacle like structures of a single material, they often comprise of a single enclosing wall (they are single walled) and deflect by inflation.

direct consequence, finding the optimal design is for a large part about positioning the pressure boundary to the ideal location.

The author of this work, J.S. Frouws, conducted a literature review prior to this work, with the title: *'Topology Optimisation of compliant mechanisms using design dependent pressure loading'*. In this study, the application of design dependent pressure loads is investigated. The aim of the literature review was to gather the available boundary identification methods in literature which identify the inner pressure boundary[2] (at each design iteration) and see how the load is applied in the right location. Consequently, the methods are evaluated on their performance. The results and considerations of the literature study are further explained in Section 1.5.

1.2. History of structural optimisation

The optimisation of the shape of an object to increase its performance is a topic that dates back more than a century. An early example in the field is the paper by **A. Michell, 1904** [21]. He mathematically derived the stiffest configuration of trusses (or beams) for several load cases[3]. These configurations are often referred to as 'Michell trusses', one example is shown in Figure 1.2a. It was much later that the field gained more momentum with the rise of modern computers. In 1988, **M. Bensøe and N. Kikuchi** [3] introduced the homogenisation method. This method allows holes to be added and removed in a topology as opposed to the (at that time more commonly known) shape optimisation methods where the amount of holes is prescribed. An example of using the homogenisation method is shown in Figure 1.2b. Another great benefit is the straightforward implementation in 3D. Bensøe published a follow-up paper [2] in 1989 and in 1992 in which he introduced a simplified version of the method, namely the SIMP method, of which an example is shown in Figure 1.2c. This method has virtual densities on each element that make up the topology.



(a) 1904: A truss configuration optimized for the stiffest design with load F and cilindrically fixed at 'B', derived by Anthony (A.G.M.) Michell [21].

(b) 1991: A topology optimized for the stiffest design using a homogenisation method with 110 × 80 = 8800 elements [27]. The method is proposed by M. Bensøe and N. Kikuchi [3].

(c) 2017: A self written Matlab implementation of the SIMP method. The volume fraction $V^*_{\max} = 0.1$, square mesh with $220^2 = 48400$ elements and loading conditions similar to the Michell truss in (a).

Figure 1.2: Examples of the development from early structural optimisation to the SIMP method.

The SIMP method gained more ground in the 1990's as can be seen by the steep rise of published papers. Nowadays, Topology Optimisation is an active field of research where the popular SIMP method plays a key role.

1.3. Topology Optimisation

A Topology Optimisation (TO) can find the optimal design or distribution of material, given a design domain, constraints and an objective that describes the performance of the system. The most common objective in literature is minimisation of compliance and thus maximisation of stiffness. When using a density based method like 'Solid Isotropic Material with Penalisation' (SIMP), the domain is first divided or meshed into n_e

[2] With the inner pressure boundary, the interface between the void and the solid phase is meant which is exposed to the pressure source.

[3] A load case is a certain collection of forces (or loads) to which a structure is exposed in a given case. For example, an aircraft should be able to withstand each extreme weather conditions, each weather condition puts other loads on the structure, these are the load cases for which a structural analysis is performed.

number of elements. Each element in the domain is then given its own virtual density value ρ_e, which has a value between 0 and 1. If $\rho_e = 1$ there is material present on the element e and if $\rho_e = 0$ there is a void. Therefore, the Young's modulus (or Elastic modulus) is related to ρ_e in the SIMP material model by:

$$E^e(\rho_e) = E_{\min} + \rho_e^{p_{\mathrm{E}}}(E - E_{\min}), \qquad \rho_e \in (0,1), \tag{1.1}$$

where p is a penalisation factor which is chosen to equal $p = 3$ in this work. For $0 < \rho_e < 1$ there is a continuous shift in elastic modulus from $E_{\min}$ (required for numerical stability) to E, which is the Young's modulus of the modelled material. The minimum compliance objective can now be written as:

$$\begin{aligned} \min_{\boldsymbol{\rho}} \; & C(\boldsymbol{\rho}) = \boldsymbol{u}^{\top}\boldsymbol{F}(\boldsymbol{\rho}), \\ \text{subject to} \;\; & \mathbf{K}(\boldsymbol{\rho})\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{\rho}), \end{aligned} \tag{1.2}$$

where:

C	=	compliance, which is the current objective function (N m),
$\boldsymbol{\rho}$	=	vector of virtual densities,
$\mathbf{K}$	=	global stiffness matrix (N m^{-1}),
$\boldsymbol{u}$	=	displacement field (m),
$\boldsymbol{F}$	=	global force vector (N).

To determine whether material needs to be added or removed by the optimiser the TO needs sensitivities (collected in the vector: $\boldsymbol{S}$). The sensitivities are simply the derivatives of the objective function C to the vector of design variables $\boldsymbol{\rho}$: $S_e = \frac{\partial C}{\partial \rho_e}$ (using index notation, e is the element index). $C(\boldsymbol{\rho})$ can be seen as an n_e dimensional objective plane[4]. Imagine n_e being 2 here (for ease of imagining), such that there are two variables that can be changed to optimise the objective $C(\rho_1, \rho_2)$. Now, the objective surface can for example look like Figure 1.3. The optimiser, in search for the minimum objective value, uses the sensitivities (represented by the arrows on the $C = 0$ plane) as a direction to move in. This means moving the solution in the direction of greatest descent. Although n_e is typically in the order of tens of thousands, the method remains the same. Note that starting at different initial conditions, $\boldsymbol{\rho}_i$, causes the optimiser to converge to different local minima.

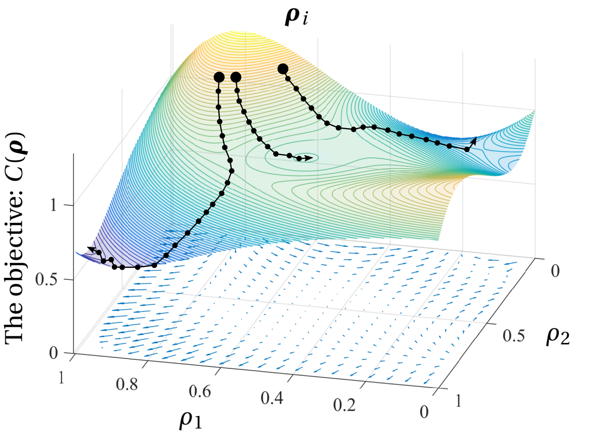


Figure 1.3: The objective plane with several possible optimisation paths. Three different initial conditions, $\boldsymbol{\rho}_i$, converge to different minima.

All TO's start from an initial condition (or design). Often the initial condition is chosen to be an equally distributed virtual density value $\boldsymbol{\rho}$ of approximately the maximum allowed volume fraction $V^*_{\max}$ in the domain, so $\boldsymbol{\rho}_i \approx V^*_{\max}\mathbf{1}$ where $\mathbf{1}$ is a vector of length n_e, filled with ones.

A TO uses the Finite Element Method (FEM) to calculate the nodal displacements (and possibly other properties) of the proposed design or material distribution to evaluate the objective and sensitivities. A more elaborate description of the Topology Optimisation method and its implementation is provided in Appendix A.

1.4. Design dependent pressure loads

In this work, the domain is exposed to a pressure source that exerts a hydrostatic pressure load on the design where the design is defined as a density field between 0 and 1. The difficulty when applying a pressure load to a design in TO the boundary on which it acts is poorly defined. The void-solid interface needs to be found that is exposed to the pressure source.

[4] n_e is the amount of elements in which the domain is divided or meshed. It is also the amount of design variables in the vector $\boldsymbol{\rho}$.

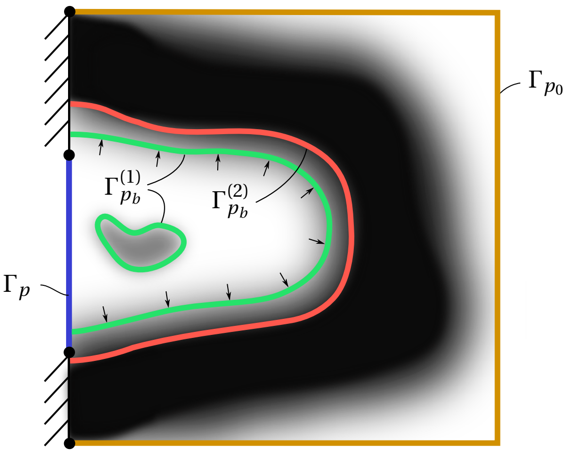


Figure 1.4: Illustration indicating the unclear inner boundary on which the pressure acts.

One focal area in the research on TO using the SIMP method, is the application of design-dependent loading (DDL). DDL can either mean applying a point load from which the location depends on the design or applying a hydrostatic pressure load that acts on the evolving inner boundary of the design. Here, the inner boundary is defined as the void-solid interface exposed to the pressure source. This seems to be a clear and foolproof description until this is put in the context of TO. In TO, boundaries are generally not well defined. The inner pressure boundary is called Γ_{p_b}, as depicted in Figure 1.4. Two other boundary domains are Γ_p and Γ_{p_0} which respectively are the boundaries where the pressure is applied (i.e. where $p = 1 \times 10^5$ Pa) and the boundary where ambient pressure occurs (i.e. where $p = 0$ Pa). Note that the pressure interface is not clearly defined when using the SIMP method, the location of the interface depends highly on the method that is used to find the boundary, the so called 'Boundary Identification Method' (BIM). Some of the BIM's use a density threshold value ρ_T to pinpoint the location of the boundary. Depending on ρ_T the interface Γ_{p_b} can for example be at $\Gamma_{p_b}^{(1)}$ or at $\Gamma_{p_b}^{(2)}$, for respectively a low, or a high value. Section 1.2 gives the recent historic background of shape and topology optimisation and Section 1.5 discusses the BIM's found in scientific literature.

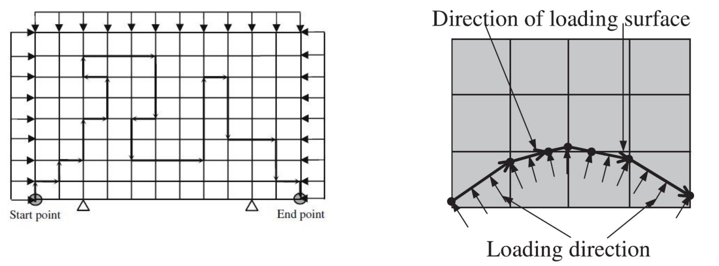
The sensitivity vector $\boldsymbol{S}$ consists of two contributions: the 'structural sensitivities' and the 'load sensitivities'. They respectively promote structural improvements to the objective, and improvements to the loading location and magnitude. The structural sensitivities are extensively studied and commonly appear in papers. However, the contribution of load sensitivities in TO is occasionally suppressed[5] [33] or, not embraced as an essential part of the sensitivities. Assume that a method is used to apply the hydrostatic pressure load to the Γ_{p_b} boundary and which incorporates load sensitivities. Then the optimiser knows what the contribution is of moving the Γ_{p_b} boundary to the objective, and the optimisation path might for example not get stuck in the minimum located near the 'saddle point' but escape that minimum and converge to a better one. In Figure 1.3, the saddle point is located at the end of the central optimisation path.

1.5. Boundary Identification Methods in literature

Since the first paper on the topic of Boundary Identification Methods (BIM's) was published by **Hammer and Olhoff, 2000** [14], several methods have been proposed for detecting the pressure boundary Γ_{p_b}, upon which the pressure acts. The challenge was in the fact that that the material distribution is poorly defined in the beginning of the optimization procedure and that the boundary Γ_{p_b} can change location at every design iteration.

Most BIM's look for an *iso-density line* based on a threshold density value (ρ_T). This method basically requires a starting point on Γ_{p_b} and starts walking along Γ_{p_b} using the density threshold to define the iso-density direction until reaching either another specified point or closing the boundary Γ_{p_b}. It is very similar to using a contour line of the virtual density field. This can either be applied *on the element boundary* as proposed by **Zhang et al., 2008** [33] (see figure 1.5a) or more smoothly *using nodal densities* (as the average of surrounding element densities) as proposed by **Hammer et al., 2000** [14]. The nodal density method was later revisited and improved by **Du and Olhoff, 2004** [10] (see figure 1.5b). **Lee and Martins, 2012** [19] improved on the start conditions of the boundary search by proposing a method that does not need to define the start and endpoint of the boundary, but can start somewhere halfway. These methods are scalable to 3D, but not as straightforward as desired.

[5] In [33] the pressure boundary Γ_{p_b} is moved to element boundaries making the load location discontinuous, virtually eliminating the load sensitivity.



(a) Element boundary walker method, [33]

(b) Isodensity line using nodal densities, [10]

Figure 1.5: Iso-density search algorithms where (b) uses nodal densities (which is the average density of the surrounding elements) and (a) uses element densities to determine the marching direction to find the iso-density boundary (similar to the definition of a contour line).

Pushing the boundary Γ_{p_b} to the edges of the elements makes the 'load sensitivity' vanish because the location of the loads becomes discrete. Then Γ_{p_b} is not sensitive to infinitesimal changes in the density field, unless the threshold value ρ_T is passed, then the nodal loads 'jump' to the next element. This is undesirable as the very reason that SIMP uses continuous densities is to have differentiability and a greater chance of finding a better optima (or the global one).

Another group of methods is characterized by the additional variables that need to be solved. **Fuchs and Shemesh, 2004** [11] proposed a method where the Γ_{p_b} is predefined using an additional set of variables. These variables are then added to the optimization algorithm and optimized together with the design variables.

Yet another method sees the density-field (in the SIMP material model) as a grey-scale image. Therefore the application of image segmentation techniques comes naturally. One of the most recent papers by **Wang et al., 2016** [32] introduced the Distance Regularized Level Set Evolution (DRLSE) (proposed by Li et al., 2010 [20]) to the boundary search problem of TO. They use a level set function (LSF) ϕ to describe the boundary. However, the downfall is that there is no analytical load-sensitivity analysis presented (due to the difficult analytical relation between the LSF and the density distribution) forcing the user to apply the expensive finite-difference method. The paper greatly reduced the amount of elements that need local sensitivity analysis by noting that the load sensitivity is only affected by closely oriented elements as observed earlier by [10][6]

Finally there are the alternative BIM's, **Zheng et al., 2009** [34] introduces a pseudo electric potential (of $V_0|_{\Gamma_p} = 1$) as analogous to a pressure source and models the material as isolator. On all other boundaries, the 'zero pressure' boundary condition is realised by setting the potential to zero ($V|_{\Gamma_{p_0}} = 0$). The design dependent pressure loading can now be described as an iso-potential line in the potential-field, the paper uses a threshold of $V = 0.95$. The electric field can be found by solving a PDE (very similar to that of the pressure distribution as described in this work, in Section 3.1) over the same mesh as the mechanics problem. Also in this paper the pressure boundary is moved to the element boundaries where the *load sensitivities vanish*.

Where normally the two phases are defined as material and void, **Sigmund and Clausen, 2007** [26] model the void as an incompressible fluid. This allows the pressure to be applied on the designated surface Γ_p and to be 'transferred' through the incompressible void to the internal material boundary Γ_{p_b} without any need for special load surface parametrizations (or load sensitivities). One problem that is introduced, being the presence of (incompressible) fluid filled cavities, is later avoided by making it a three phase problem: material, fluid and void, doubling the design variables [26]. The correct discretisation requires a special variational formulation and elements. To have control over the formation of fluid filled cavities, a maximum fluid fraction is introduced that limits the fluid volume.

The three phase method proposed by Sigmund and Clausen is successfully used by **H. Panganiban et al., 2010** and **S. Vasista et al., 2012** [22, 31] to solve a compliant mechanism design. It allowed for an easy integration over the pressure boundary in order to define an alternative 'input displacement' constraint for the compliant mechanism design [22].

[6] The author would like to emphasize that this highly depends on the BIM.

After the paper by Sigmund and Clausen [26], several papers appeared on topology optimization of incompressible media [7, 16] and the application to pressure load problems [6]. When using normal displacement based elements it is numerically difficult to model an incompressible medium. To aid in the numerical solution, when using this method, a 'truly-mixed' variational formulation resulting from the variational principle of Hellinger-Reissner is used, together with a discretisation based on the Johnson and Mercier finite element [6].

1.6. The research aim, and a quick glance at the report

A lot of solutions have been proposed for applying the design dependent load (DDL) and these are all tested on a minimum compliance problem. Only one method is also used in a pressure actuated compliant mechanism design, namely the three phase method in [26]. Although the method showed satisfying results, is stable and can be extended to 3D, it requires special elements and an adjusted variational formulation[6], and it is difficult to apply pressure sources inside the domain. Analysing the general concept of a BIM, the used BIM ideally:

1. offers an explicit formulation of the nodal forces,
2. has stable and reliable execution,
3. provides a high probability of finding the global optimum, and therefore should have load sensitivities that preferably range further then only elements directly adjacent to the pressure boundary (improving the convexity of the problem),
4. has not too many tuning parameters,
5. inheres general applicability[7] (e.g. 2D, 3D and can naturally be extended to several load cases),
6. is simple to implement (as a non-intervening 'overlay' type of solution, not altering the FEM),
7. is computationally cheap.

In literature there has not been a pronounced focus on this, or any, combined set of properties. Every proposed method in the scientific literature has some shortcomings, and that is why in this thesis a new method is proposed with the focus on the properties listed here. Since the design of a pressure driven compliant mechanism has already been done, the focus of this thesis is narrowed down to finding a boundary-identification method that stimulates differentiability, more globally present load sensitivities and single walled structures (like inflatables).

Therefore the research questions are as they are, as described in Section 1.1. In this work a new method is introduced to define the pressure boundary. Where most other methods try to find a pressure line to calculate the nodal forces equivalent to the pressure, this work solves a design dependent pressure field using FEM and then translates the pressure drop to loads on the elements. This two step process allows for a very flexible and tunable method to apply the pressure loads and, in a satisfactory degree, covers all perks mentioned above.

Chapter 2 introduces the mathematical notation and definitions, together with the names assigned to the different volumetric domains and boundary domains. It is advised to read this introductory chapter to get acquainted with the notations. Subsequently, Chapter 3 explains why the Darcy flow can realistically model the pressure field. It also discusses the relation between the flow parameters and the virtual density ρ_e on each element. The chapter ends by deriving the relation between pressure drop and resulting nodal forces. The proposed BIM is from here on called the Darcy method. Chapter 4 explains the compliant mechanism objectives that are used for the optimisation of pressure actuated compliant mechanisms. It closes with the sensitivities of the different objective functions, taking into account the Darcy method. Chapter 5 provides two additional constraints with their sensitivities to suppress some undesirable design options in the optimisation. Then, Chapter 6 contains the results of several different parameter sweeps. Section 6.3.2 to 6.3.5 respectively sweep over several flow coefficient and drainage parameters, several output spring stiffness values, different initial conditions, different volume fractions. The chapter ends with Section 6.4 which offers some alternative soft robotic like solutions. Finally, Chapter 7 discusses the results and Chapter 8 offers a conclusion and recommendations for future research.

[7] The term 'inheres' is used to infer the design of a method with the specific property 'build-in' in its basic working mechanism.

2

Mathematical preliminaries

This chapter introduces the equations and notation associated with the Finite Element Analysis (FEA). A full derivation of a FEA of both a scalar field variable and a displacement field is provided in Appendix B. It is advised to read this chapter as it will help the reader to become familiar with the notations.

2.1. Notation and definitions

2.1.1. Notation

This section introduces the notation for vectors and matrices used throughout this thesis. First, a scalar value is indicated as a cursive letter, like t, ξ or x. A column or row vector is indicated as a cursive, bold letter like $\boldsymbol{F}$ or $\boldsymbol{b}$ and a unit vector has a hat: $\hat{\boldsymbol{e}}_i$. When indicating a matrix (excluding matrices of the size of a column or row vector) a bold Capital letter is used like $\mathbf{A}$ or $\mathbf{K}$. In some cases the dimension is indicated behind the equation between straight brackets like:

$$\boldsymbol{x}^e = \mathbf{N}(\xi,\eta)\boldsymbol{x}_n^e, \quad [2\times 8]\,[8\times 1] \tag{2.1}$$

where $\boldsymbol{x}^e = \begin{bmatrix} x \\ y \end{bmatrix}$ is of the dimension $[2\times 1]$ which can be directly read from the bracket notation (being the first and last size indications). The sub or super index e always refers to the element index, often at element level. In some occasions a sub-index is used behind a bracket: $\left(c_i \boldsymbol{N}_i f_i\right)_{i=1,\ldots,4}$, this indicates that the equation f_i can be evaluated for each $i \in \{1,2,3,4\}$, making it a set of equations. Finally the comma notation indicates the derivative to a certain variable like: $N_{1,\xi} \equiv \frac{\partial N_1}{\partial \xi}$ or $\xi_{,x} \equiv \frac{\partial \xi}{\partial x}$ and if the variable to which we differentiate is clear from context, the apostrophe is used: $S_e = C' = \frac{\partial C}{\partial \rho_e}$.

2.1.2. Set and domain assignments

In this section all domains are assigned a name to aid the reader. Figure 2.1 shows the different domains and boundaries where:

Ω_m	=	Domain with material ($\rho_e \approx 1$),
Ω_v	=	Domain with void ($\rho_e \approx 0$),
Γ_p	=	Boundary of applied pressure,
Γ_{p_0}	=	Boundary where outflow pressure is defined as p_out,
Γ_u	=	Boundary with fixed displacements,
Γ_s	=	Boundary with symmetry conditions,
Ω	=	$\Omega_\mathrm{v} \cup \Omega_\mathrm{m}$ Total domain,
Γ	=	$\Gamma_p \cup \Gamma_{p_0} \cup \Gamma_u \cup \Gamma_\mathrm{s}$ Complete boundary,
$\boldsymbol{b}$	=	Body loads ($\mathrm{N\,m^{-3}}$).

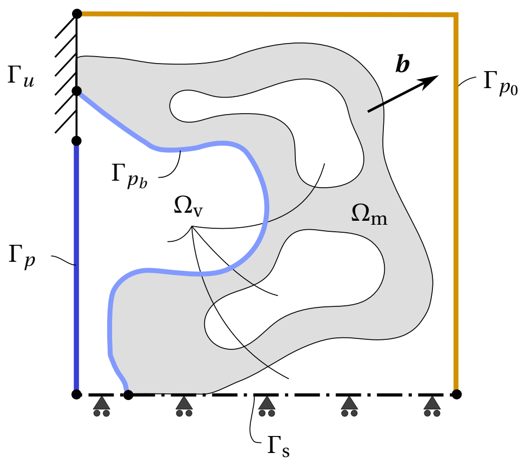


Figure 2.1: The design domain with indicated void, material, boundaries and fixtures.

The total domain Ω is divided in n_e number of adjacent square (or cubic) elements. On the four corners of these elements we define the nodes, such that each node is part of four elements. The number of nodes is n. A physical field can now be expressed on these nodes as discrete nodal values. Then shape functions are used to interpolate the physical field on the rest of the elements.

If temperature is expressed in nodal values, each node has one temperature value (i.e. a scalar field). But if a displacement vector is expressed on a node, the node has as many unknowns as it has spatial dimensions it can move in (i.e. a vector field). So a displacement field in 2D has two unknown Degrees of Freedom (DOF's) per node. As such, the DOF's can be subdivided in the following sets:
$\mathbb{D}_p = \{1, 2, ..., n\}$ are all DOF's indices of a pressure field,
$\mathbb{D}_u = \{1, 2, ..., nd\}$ are all DOF's indices of a displacement field (all n nodes have a DOF in d free spatial directions),
$\mathbb{D}_e = \{1, 2, ..., n_\mathrm{e}\}$ consists of all the element indices.

2.2. Finite Element Method

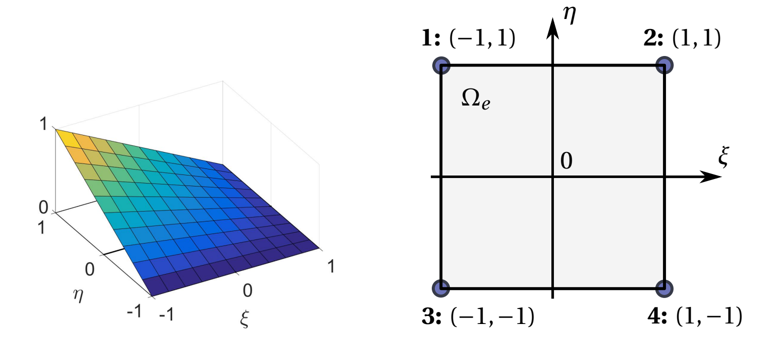
In this thesis, only a rectangular 2D domain is used for simplicity and to cut computational costs (so $d = 2$). The width and height of the domain and the elements can be adjusted. The division of the domain in elements (the discretisation) is necessary to derive the finite element (FE) equations. The full derivation is provided in Appendix B.

2.2.1. Shape functions

The shape functions (or basis functions) interpolate the physical field on the element using the discrete nodal values. For this work, only bilinear shape functions are used for simplicity [1]:

$$\boldsymbol{N}(\boldsymbol{\xi}) = \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4}(1-\xi)(1+\eta) & \frac{1}{4}(1+\xi)(1+\eta) & \frac{1}{4}(1-\xi)(1-\eta) & \frac{1}{4}(1+\xi)(1-\eta) \end{bmatrix}. \quad [1 \times 4] \tag{2.2}$$

The first of the four shape functions, N_1, is displayed in Figure 2.2a. The local numbering of the nodes of a standard element (of size 2×2 in the ξ, η plane) is shown in Figure 2.2b. Notice that N_1 is only one at node 1 and zero on the other three. When $N_1(\boldsymbol{\xi})$ is multiplied by the nodal value of a physical quantity of node 1, the shape function shows the linear interpolation, and thus the contribution within the element domain Ω_e.



(a) Shape function: N_1.

(b) Standard element domain and node numbering.

Figure 2.2: (a) The first shape function over a 'standard' unit element. (b) indicates the local numbering of the nodes of a standard element and shows its domain Ω_e in the ξ, η plane. If desired, one can look up all the linear shape functions in Figure B.1 in Appendix B.2.1.

The shape functions can also be organised in a matrix in order to express vector fields on each node instead

[1] All methods that are proposed in this thesis can handle higher order shape functions.

of scalar fields. The matrix of shape functions is:

$$\mathbf{N}(\boldsymbol{\xi}) = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \quad [d \times 4d]. \tag{2.3}$$

This matrix can, for example, be used to express a displacement field or the mapping from the x, y plane (within an element in the physical domain) to the standard element Ω_e in the ξ, η plane as shown in Figure 2.3. As the nodal coordinates in the physical (x, y) domain are known from the discretisation, the continuous description (for d=2) is:

$$\boldsymbol{x}^e(\boldsymbol{\xi}) = \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{N}(\boldsymbol{\xi})\boldsymbol{x}_n^e = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix}, \tag{2.4}$$

where x_i indicates the x coordinate of the i-th node, the same goes for y_i. Sub-index n indicates the vector of nodal coordinates. Equation (2.4) is valid for $-1 \le \xi \le 1$ and $-1 \le \eta \le 1$.

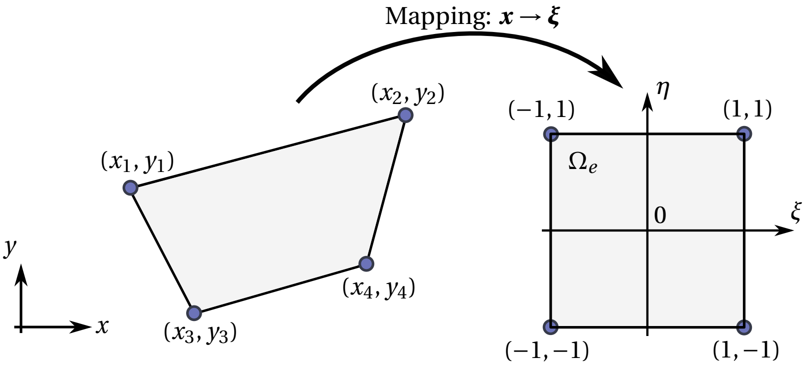


Figure 2.3: Mapping from a deformed element in the x, y plane to the local ξ, η plane (or standard element).

2.2.2. The 'nabla' operator: ∇

The *nabla operator* is used in different operations and is expressed as a vector: $\nabla \equiv \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right]^T$ for $d = 3$. This notation aids in explaining several mathematical concepts like the Gauss theorem and the derivation of FEA. ∇ can be used, among others, in the following expressions:

$$\begin{cases} \nabla T = \text{grad } T & = \left[\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z}\right]^\top, & \mathbb{R} \rightarrow \mathbb{R}^3 \\ \nabla^\top \boldsymbol{q} = \nabla \cdot \boldsymbol{q} = \text{div } \boldsymbol{q} & = \left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial q_z}{\partial z}\right), & \mathbb{R}^3 \rightarrow \mathbb{R} \\ \nabla \times \boldsymbol{q} = \text{curl } \boldsymbol{q} & = \left[\left(\frac{\partial q_z}{\partial y} - \frac{\partial q_y}{\partial z}\right), \left(\frac{\partial q_x}{\partial z} - \frac{\partial q_z}{\partial x}\right), \left(\frac{\partial q_y}{\partial x} - \frac{\partial q_x}{\partial y}\right)\right] & \mathbb{R}^3 \rightarrow \mathbb{R}^3 \\ \Delta T = \nabla^2 T = \text{div } (\text{grad } T) & = \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right), & \mathbb{R} \rightarrow \mathbb{R} \end{cases} \tag{2.5}$$

where $\boldsymbol{q}(x, y, z) = \left[q_x, \; q_y, \; q_z\right]^\top$ is a vector field (e.g. a heat flux) and $T(x, y, z)$ a scalar field (e.g. a temperature field). These are all continuous expressions of the fields.

3

Pressure boundary identification and implementation

One of the main problems in this thesis is to obtain the discretised force vector $\boldsymbol{F}$ that results from the pressure load that is applied on the design to actuate the mechanism. This force vector is commonly called the consistent load vector[1] of size $[nd \times 1]$ as mentioned in [8]. With the method proposed in this chapter it takes two steps to obtain the consistent load vector with nodal loads. First, the distribution of pressure, i.e. the pressure field, is simulated over the domain using a chosen physical model. Second, the nodal forces are calculated using the obtained pressure field. The first step requires a physical model of the pressure that depends on the changing topology, i.e. making the pressure drop when it encounters material.

3.1. Modelling the pressure field

To simulate a pressure field in a TO problem, Darcy's law is chosen. Darcy's law normally models fluid flow through a porous medium, like water flow through soil or sandstone. In order to use Darcy's law in a TO, it is adjusted to respond to the design. Therefore the porosity is made density dependent. We are free to choose any other model as long as it models the pressure distribution in a physically correct manner and exhibits the following properties:

1. the pressure drops from p_{in} to $p_{\text{out}} = 0$ over the boundary or wall when the topology is fully developed,
2. the pressure acts only on a thin layer on the inside of the boundary,
3. results in closed pressure boundaries and prevents leakage,
4. it is easily expandable to 3D, different domains, different boundary conditions (e.g. a symmetry condition)
5. especially within the context of this thesis, the model should allow the generation of single walled (soft robotic like) solutions if that is most ideal.

Within these conditions the method can be tuned to behave well in the optimisation. Darcy's law is chosen because it offers a tunable differentiability that allows for more optimal solutions (i.e. better decision making) by allowing more spread out load sensitivities, especially in the early stage of optimisation. In the following subsection, Darcy's law is introduced to calculate the pressure field. This calculation is done in each iteration.

[1] It is called the consistent load vector because in the discretisation its entries are consistent with the integral of the loads on the surrounding elements.

3.1.1. Darcy's law

Darcy's law can be stated as:

$$\boldsymbol{q}(\boldsymbol{x}) = -\frac{\kappa}{\mu}\,\nabla p(\boldsymbol{x}) \quad = -K\,\nabla p(\boldsymbol{x}), \tag{3.1}$$

where K, in $\mathrm{m^4\,N^{-1}\,s^{-1}}$, is introduced here as a flow coefficient[2],
and where:

$\boldsymbol{q}$	=	Darcy flux ($\mathrm{m\,s^{-1}}$),
κ	=	permeability ($\mathrm{m^2}$),
μ	=	fluid viscosity ($\mathrm{Pa\,s}$),
∇p	=	pressure gradient ($\mathrm{N\,m^{-3}}$),

This law states that the flow of (incompressible) fluid through a unit area is proportional to the pressure drop per unit length and inversely proportional to the resistance of the porous medium on the fluid. Through Darcy's law, the pressure will gradually drop from the inner boundary Γ_{p_b} to the outer boundary Γ_{p_0} (where the prescribed pressure: $p_{\text{out}} = 0\,\mathrm{Pa}$) as illustrated in a 1D example in Figure 3.1a. The flow coefficient $K(=\frac{\kappa}{\mu})$ can now be related to the material density: $K(\boldsymbol{\rho})$. A smooth Heaviside function is chosen to be used that can be adjusted with two parameters, η and β. This relation, shown in Figure 3.2a, is stated as:

$$K(\rho_e) = k_{\text{void}} - (k_{\text{void}} - k_{\text{mat}})\frac{\tanh(\beta_{\text{k}}\eta_{\text{k}}) + \tanh(\beta_{\text{k}}(\rho_e - \eta_{\text{k}}))}{\tanh(\beta_{\text{k}}\eta_{\text{k}}) + \tanh(\beta_{\text{k}}(1-\eta_{\text{k}}))}, \tag{3.2}$$

where η_{k} can be interpreted as a smooth threshold density, analogous to the threshold density used in literature (as explained in Section 1.2). Hence, η_{k} can be tuned to alter the position of the step, $K(\eta_{\text{k}}) = \frac{1}{2}(k_{\text{void}} + k_{\text{mat}})$ and β_{k} to control the slope through: $\frac{dK}{d\rho_e}(\eta_{\text{k}}) = k_{\text{mat}}\beta_{\text{k}}/2$. The parameters k_{void} and k_{mat} are used to control the flow coefficient in void ($\rho_e = 0$) and full material ($\rho_e = 1$) respectively.

The parameter values can be deduced from actual viscosity and permeability values. The permeability of an impervious material is roughly $1\times10^{-15}\,\mathrm{m^2}$, and the viscosity of, lets say air, is in an order of magnitude of $1\times10^{-5}\,\mathrm{Pa\,s}$. The flow coefficient in material is now chosen to be $k_{\text{mat}} = 1\times10^{-10}\,\mathrm{m^4\,N^{-1}\,s^{-1}}$ and k_{void} is then chosen to be large to mimic a free flow with low resistance in the void region. If k_{void} is chosen large enough there is no significant pressure drop over the void region. $k_{\text{void}} = 1\times10^{-3}\,\mathrm{m^4\,N^{-1}\,s^{-1}}$ is generally used in this work, any variations of this value will be indicated accordingly.

The fact that this model does not allow a pressure drop to occur instantly over a sharp boundary, but over a finite layer of material, causes the equivalent nodal loads to be present either inside the material as well as at the very boundary, as shown in Figure 3.1. This penetrating pressure load, caused by the Darcy method, is a smeared-out version of a pressure on a sharp boundary or interface which is used in most other 'boundary identification methods' mentioned in Section 1.2. When all the contributions of the penetrating load are summed up, it matches the single boundary load[3]. In other words, it is assumed in this work that local differences in load application have no significant effect on the global behaviour of the structure, which is supported by the Saint-Venant principle[4].

3.1.2. Similar models, different physics

Using Darcy's law introduces a weakness in the model which is illustrated in Figure 3.1. When the topology has more then one 'wall' between p_{in} and p_{out} the pressure does not completely drop over the first boundary as illustrated in Figure 3.1b. To solve this issue we turn to physical models that are directly comparable to Equation (3.1).

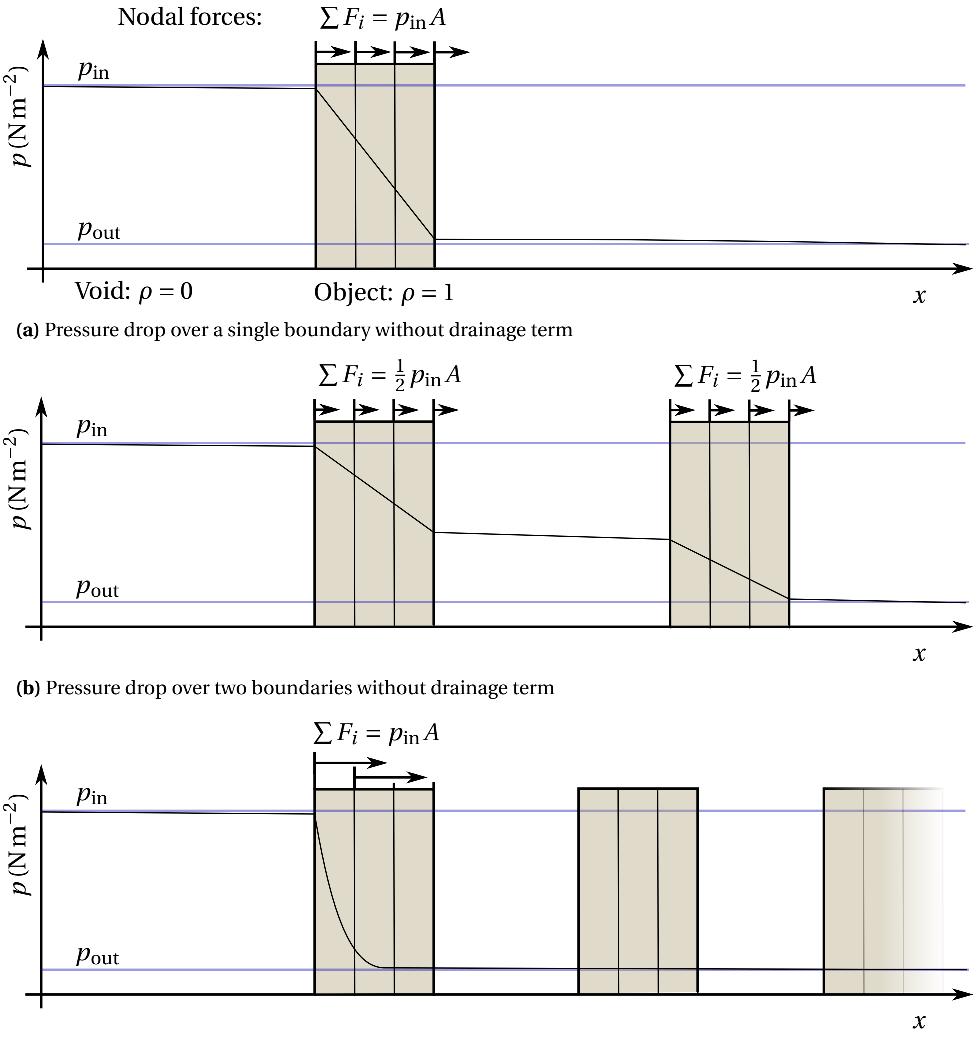
Darcy's law is similar to Fourier's law for heat flow (where $K \sim k$, the heat conductivity) and Ohm's law for electrical conduction (where $K \sim \epsilon$, the dielectric coefficient)[5].

[2] The parameter $K = \frac{\kappa}{\mu}$ is introduced here as the 'flow coefficient', despite the fact that this terminology is sometimes used in literature with a different meaning. In this work, K is a measure of the ability to flow, a low K value means a high volumetric resistance, causing a large pressure drop.

[3] This is checked by hand with a boundary of a topology that is designed using a self-written TO algorithm in Matlab©.

[4] The Saint-Venant principle states that the global deformation due to local forces is similar to the global deformation due to the resultant of the local forces

[5] Ohm's law is used by Zheng et al., 2009 [34] to identify the boundary. The main difference between the method introduced in this thesis


Nodal forces:
$\sum F_i = p_{\text{in}} A$
p_{in}
p_{out}
$p\,(\text{N}\,\text{m}^{-2})$
Void: $\rho = 0$
Object: $\rho = 1$
x
$\sum F_i = \frac{1}{2} p_{\text{in}} A$
$\sum F_i = \frac{1}{2} p_{\text{in}} A$
p_{in}
p_{out}
$p\,(\text{N}\,\text{m}^{-2})$
x
$\sum F_i = p_{\text{in}} A$
p_{in}
p_{out}
$p\,(\text{N}\,\text{m}^{-2})$
x

(a) Pressure drop over a single boundary without drainage term

(b) Pressure drop over two boundaries without drainage term

(c) pressure drop over multiple boundaries with drainage term

Figure 3.1: These illustrations show the behaviour of a one-dimensional pressure field when using Darcy's law and porous objects (consisting here of 3 elements). This behaviour is independent of the choice of k_{void} and k_{mat}. (a) shows the pressure drop over a single object. (b) shows an undesirable condition where the pressure drop happens over multiple objects or walls. In (c) an additional drainage term is used (analogous to a convective term to $p_{\text{out}} = 0$) such that the full pressure difference acts on the first wall exposed to the pressure source.

To overcome the issue with Darcy's law when encountering more than one boundary, a phenomena is used that is better explained using a heat flow analogy. Fourier's law is:

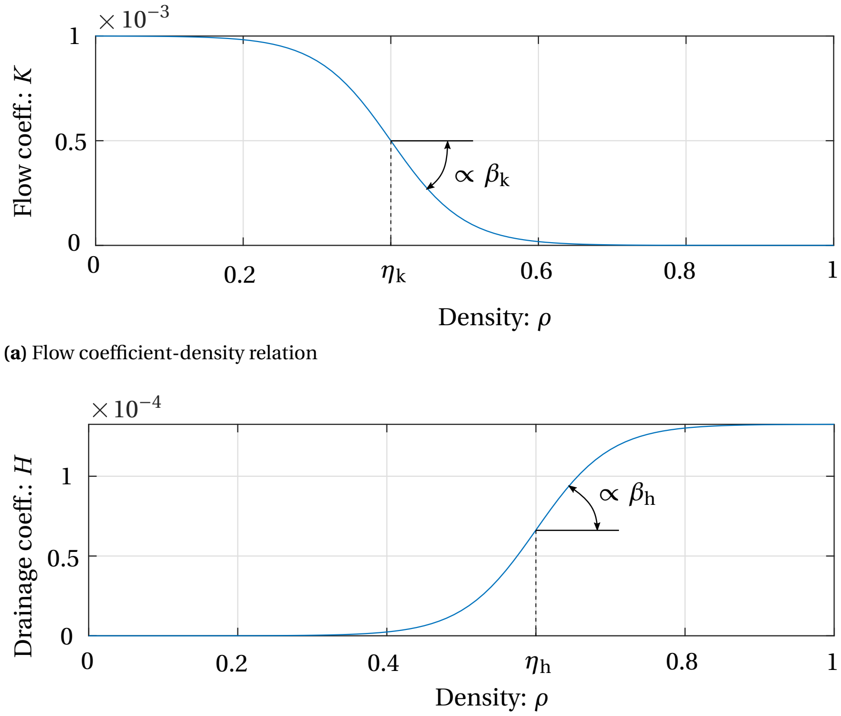
$$\boldsymbol{q} = -k\,\nabla T, \tag{3.3}$$

where:

$\boldsymbol{q}$	=	heat flux ($\text{W}\,\text{m}^{-2}$),
k	=	heat conductivity ($\text{W}\,\text{m}^{-1}\,\text{K}^{-1}$),
∇T	=	pressure gradient ($\text{K}\,\text{m}^{-1}$).

In a heat flow model, heat can be generated (e.g. inductive heating) or dissipated (e.g. a heat sink) in the middle of a domain. For example, convective heat loss is governed by: $Q_{\text{conv}} = \int h(T - T_0)\,t\,d\Gamma$. *The effect that is desired in the Darcy model is a volumetric, material dependent, pressure loss.* This would be mathematically analogous to the inverse of inductive heating, so like volumetric cooling. From here on this is called the

and the method Zheng uses, is that Zheng drives the loads to the edges of the elements which undesirably eliminates load sensitivities. This thesis promotes the use of a physically correct and intuitive model to define the pressure field and provide the TO with sufficiently far reaching load sensitivities.



(a) Flow coefficient-density relation

(b) Drainage coefficient-density relation

Figure 3.2: A smooth Heaviside function is used for $K(\rho)$ and $H(\rho)$. So if the TO converges towards a 0-1 design, the flow coefficient is almost zero and the drainage very large.

'drainage' term and is added as:

$$\int_\Omega N_i \tilde{Q}_\mathrm{drain}\ d\Omega = -\int_\Omega N_i \left(H(\rho_e)(p - p_\mathrm{out})\right)\ d\Omega, \quad \text{[scalar]} \tag{3.4}$$

where:
H = drainage coefficient ($\mathrm{m^2\,N^{-1}\,s^{-1}}$),
p = continuous pressure field ($\mathrm{N\,m^{-2}}$),
p_out = external pressure ($\mathrm{N\,m^{-2}}$) (in this work $p_\mathrm{out} = 0$),
$\tilde{Q}_\mathrm{drain}$ = Volumetric drainage per second per unit volume) ($\mathrm{m^3\,s^{-1}m^{-3} = s^{-1}}$).

We can use a density dependent drainage term in our Darcy flux model to make the pressure drop to zero when $\rho_e = 1$, as shown in Figure 3.1c. The relation for $H(\rho_e)$ is chosen to be similar to Equation (3.2), being the smooth Heaviside function. $H(\rho_e)$ is plotted in Figure 3.2b and can be written as:

$$H(\rho_e) = h_\mathrm{mat} \frac{\tanh\left(\beta_\mathrm{h}\eta_\mathrm{h}\right) + \tanh\left(\beta_\mathrm{h}(\rho_e - \eta_\mathrm{h})\right)}{\tanh\left(\beta_\mathrm{h}\eta_\mathrm{h}\right) + \tanh\left(\beta_\mathrm{h}(1 - \eta_\mathrm{h})\right)}. \tag{3.5}$$

Where again, β_h and η_h are chosen similarly to β_k and η_k, i.e. to control the location and slope: $\frac{dH}{d\rho_e}(\eta_\mathrm{h}) = h_\mathrm{mat}\beta_\mathrm{h}/2$. The drainage term can be adjusted through the drainage coefficient h_mat. With this coefficient one can adjust the thickness of the pressure-penetration layer. Now, the optimisation algorithm can effectively control the location, and depth of penetration of the applied pressure.

It is undesirable to have an excessive number of parameters, for that reason we can relate h_mat directly to the desirable penetration depth using:

$$h_\mathrm{mat} = \left(\frac{\ln r}{\Delta s}\right)^2 k_\mathrm{mat}, \tag{3.6}$$

where:
h_mat = drainage coefficient in material ($\mathrm{m^2\,N^{-1}\,s^{-1}}$),
r = ratio of input pressure to pressure at depth Δs, i.e.: $p(\Delta s) = r\,p_\mathrm{in}$,
Δs = penetration depth of pressure (m),
k_mat = flow coefficient in material ($\mathrm{m^4\,N^{-1}\,s^{-1}}$).

This is derived in Appendix C. In this work $r = 0.1$ and $k_{\text{mat}} = 1 \times 10^{-10}\,\text{m}^4\,\text{N}^{-1}\,\text{s}^{-1}$. The value for Δs can intuitively be set to the width or height of a few elements, in this work Δs typically is $2\Delta x = 2\,\text{mm}$ or two times the element width. For clarity, Material is defined as the location where $\rho_e = 1$.

3.1.3. Deriving the state equation

To solve the pressure field using the Darcy model requires an equation that describes the state of an element. In this case the law of 'conservation of mass' is used. Because the fluid is assumed incompressible, this can be simplified to a conservation of volume.

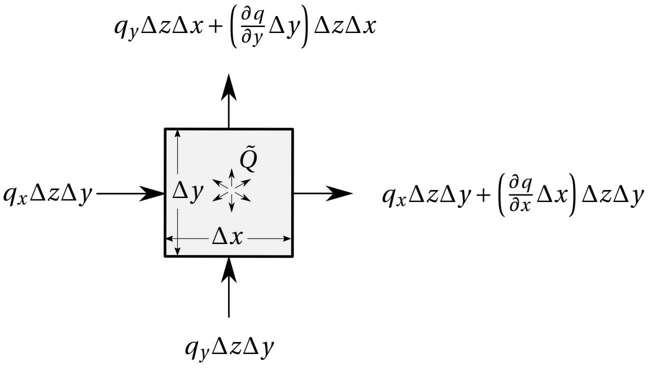

$q_y\Delta z\Delta x + \left(\frac{\partial q}{\partial y}\Delta y\right)\Delta z\Delta x$
$\tilde{Q}$
$q_x\Delta z\Delta y$
Δy
$q_x\Delta z\Delta y + \left(\frac{\partial q}{\partial x}\Delta x\right)\Delta z\Delta y$
Δx
$q_y\Delta z\Delta y$

Figure 3.3: Equilibrium of incompressible volume flow

Figure 3.3 shows the in- and outflow of a volume element[6]. The combined inflow and added volume $\tilde{Q}$ should now equal the outflow, for 2D this is:

$$\begin{aligned} Q^e_{\text{in}} &= Q^e_{\text{out}}, \\ q_x\Delta z\Delta y + q_y\Delta z\Delta x + \tilde{Q}\Delta x\Delta y\Delta z &= q_x\Delta z\Delta y + q_y\Delta z\Delta x + \left(\frac{\partial q}{\partial x}\Delta x\right)\Delta z\Delta y + \left(\frac{\partial q}{\partial y}\Delta y\right)\Delta z\Delta x, \\ \tilde{Q}\Delta x\Delta y\Delta z &= \left(\frac{\partial q}{\partial x}\Delta x\right)\Delta z\Delta y + \left(\frac{\partial q}{\partial y}\Delta y\right)\Delta z\Delta x, \\ \tilde{Q}\Delta V &= \left(\frac{\partial q}{\partial x} + \frac{\partial q}{\partial y}\right)\Delta V, \end{aligned} \tag{3.7}$$

where $\tilde{Q}$ (s^{-1}) is externally added volume per second per unit volume (or removed if negative). This is added to later operate as a 'pressure dissipation' or (more appropriately) 'drainage' term and works analogous to a heat sink (as explained in section 3.1.2). When taking the limit: $\Delta V \to 0$, Equation (3.7) can be written as:

$$\begin{aligned} \int \left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y}\right) dv - \int \tilde{Q}\,dv = 0, \\ \int \nabla \cdot \boldsymbol{q}\,dv - \int \tilde{Q}\,dv = 0. \end{aligned} \tag{3.8}$$

Because $\boldsymbol{q}$ contains first derivatives of the pressure, Equation (3.8) requires second spatial derivatives of the pressure field which is undesirable when using linear shape functions. To overcome this problem the Galerkin method is used, Equation (3.8) is multiplied by each shape function and integrated over the entire volume:

$$\sum_{e=1}^{n_e} \left(\int_{\Omega_e} \left(\nabla \cdot \boldsymbol{q} - \tilde{Q}\right) N_i(\boldsymbol{\xi})\,d\Omega \right)_{i=1,..,4} = 0. \quad \text{[scalar]} \tag{3.9}$$

Because the shape function is zero outside the element, the integration is limited to the integration over the element, Ω_e. The next steps are described in Appendix B.3, where the differentiation order is reduced and the pressure is discretised by using the shape functions contained in $\boldsymbol{N}$:

$$p(\boldsymbol{x}) = \sum_{e=1}^{n_e} \boldsymbol{N}(\xi,\eta)\boldsymbol{p}^e, \qquad [1 \times 4]\,[4 \times 1] \quad \text{where} \quad \boldsymbol{p}^e = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}. \tag{3.10}$$

[6] The Darcy flux $\boldsymbol{q}$ is equal to the fluid velocity if there is no porous medium in that volume, if there is, the average fluid velocity would be higher because there is less area for the volume flow to go through

Using this the derivation arrives at the FEA equations Equation (B.25), repeated here: $\mathbf{A}\boldsymbol{p} = \boldsymbol{f} =$

$$\underbrace{\sum_{e=1}^{n_e}\int_{\Omega_e}\mathbf{L}_p^{e\top}\left(K\,\mathbf{B}_p{}^{\top}\mathbf{B}_p + H\,\boldsymbol{N}^{\top}\boldsymbol{N}\right)\mathbf{L}_p^{e}\;d\Omega}_{\mathbf{A}}\;\;\boldsymbol{p} = \underbrace{\sum_{e=1}^{n_e}\mathbf{L}_p^{e\top}\left(\int_{\Omega_e}H\,\boldsymbol{N}^{\top}p_0\;d\Omega - \int_{\Gamma}\boldsymbol{N}^{\top}\boldsymbol{q}_{\Gamma}\cdot\boldsymbol{n}\;d\Gamma\right)}_{\boldsymbol{f}}, \tag{3.11}$$

with $\boldsymbol{p}$ being the global pressure vector of length n and $\mathbf{L}_p^e$ the index matrix that points to the correct location of the local element DOF's in the global DOF vector [7]. The variables are defined as:

$\boldsymbol{N}(x,y)$	=	vector of shape functions,	
$\mathbf{B}_p(x,y)$	=	matrix of spatial derivatives $= \boldsymbol{N}_{,\boldsymbol{x}}$ (m^{-1})	
$\boldsymbol{n}$	=	boundary normal vector,	
$\mathbf{L}_p^e$	=	global indexing matrix, $[n \times 4]$,	
K	=	flow coefficient,	($\mathrm{m}^4\,\mathrm{N}^{-1}\,\mathrm{s}^{-1}$)
H	=	drainage coefficient,	($\mathrm{m}^2\,\mathrm{N}^{-1}\,\mathrm{s}^{-1}$)
$\mathbf{A}$	=	global flow matrix, $[n \times n]$	($\mathrm{m}^5\,\mathrm{N}^{-1}\,\mathrm{s}^{-1}$)
$\boldsymbol{f}$	=	global loading vector, $[n \times 1]$	($\mathrm{m}^3\,\mathrm{s}^{-1}$)
p_0	=	external drain pressure, here $p_0 = 0$,	($\mathrm{N}\,\mathrm{m}^{-2}$)
$\boldsymbol{q}_\Gamma$	=	prescribed Darcy flux at the boundary.	($\mathrm{m}\,\mathrm{s}^{-1}$)

3.2. Pressure field to consistent nodal loads

Now that the pressure field can be calculated and controlled, it can be translated into consistent nodal loads. In a FEA, loads can be implemented in two ways. There are the boundary forces that require a surface integration, and the body forces that require a volume integration. In this section the force resulting from the change in a pressure field is expressed as an equivalent body force which is easily translated into consistent nodal loads using standard FEM methods as discussed in Appendix B.4. To find the equivalent body force vector $\boldsymbol{b}$, an infinitesimal volume element is considered with pressures acting on it.

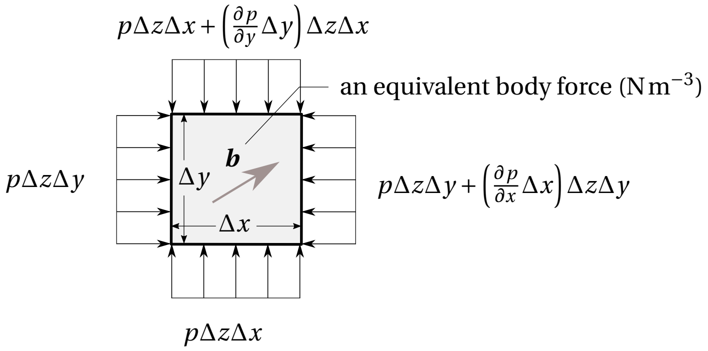

$p\Delta z\Delta x + \left(\frac{\partial p}{\partial y}\Delta y\right)\Delta z\Delta x$
an equivalent body force ($\mathrm{N}\,\mathrm{m}^{-3}$)
$p\Delta z\Delta y$
$\boldsymbol{b}$
Δy
Δx
$p\Delta z\Delta y + \left(\frac{\partial p}{\partial x}\Delta x\right)\Delta z\Delta y$
$p\Delta z\Delta x$

Figure 3.4: Forces on an infinitesimal volume.

Figure 3.4 shows an element of volume: $\Delta V = \Delta x\Delta y\Delta z$, the pressure loads on the boundaries of the element, and the body force that we want to set equal to the pressure forces on the volume. From this the following can be stated:

$$\begin{bmatrix}\sum F_x\\ \sum F_y\\ \sum F_z\end{bmatrix} = \begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}\Delta V \quad\Rightarrow\quad \begin{bmatrix}p\Delta z\Delta y - p\Delta z\Delta y - \left(\frac{\partial p}{\partial x}\Delta x\right)\Delta z\Delta y\\ p\Delta z\Delta x - p\Delta z\Delta x - \left(\frac{\partial p}{\partial y}\Delta y\right)\Delta z\Delta x\\ p\Delta x\Delta y - p\Delta x\Delta y - \left(\frac{\partial p}{\partial z}\Delta z\right)\Delta x\Delta y\end{bmatrix} = \begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}\Delta V, \tag{3.12}$$

[7] The indexing matrix is not specifically described in this work, the reader is referred to [8] for the conversion from the local to the global frame. The indexing matrix serves here as a mathematical tool and is normally not directly implemented this way due to its high computational expenses.

where in the 2D case, Δz is the thickness t and $\frac{\partial p}{\partial z} = 0$. Equation (3.12) can be simplified to get:

$$\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \Delta V = - \begin{bmatrix} \frac{\partial p}{\partial x} \Delta x \Delta z \Delta y \\ \frac{\partial p}{\partial y} \Delta y \Delta z \Delta x \\ \frac{\partial p}{\partial z} \Delta z \Delta x \Delta y \end{bmatrix} = - \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{bmatrix} \Delta V, \tag{3.13}$$

then by taking the limit: $\Delta V \rightarrow 0$ we can define the relation to be:

$$\boldsymbol{b} dV = -\nabla p dV. \tag{3.14}$$

In discretized form, Equation (3.14) becomes: $-\nabla p dV = -\mathbf{B}_p \boldsymbol{p}^e dV$. The equivalent body force $\boldsymbol{b}$ can now directly be substituted in the expression for the nodal force vector Equation (B.38), repeated here:

$$\boldsymbol{F}^e = \int_\Gamma \mathbf{N}^\top \boldsymbol{t}(\boldsymbol{x}) \, dA + \int_\Omega \mathbf{N}^\top \boldsymbol{b}(\boldsymbol{x}) \, dV. \tag{B.38}$$

Because in this work we do not consider boundary tractions $\boldsymbol{t} = \mathbf{0}$, we can simplify Equation (B.38) to:

$$\boldsymbol{F}^e = -\int_\Omega \mathbf{N}^\top \nabla p dV = -\int_{\Omega_e} \mathbf{N}^\top \mathbf{B}_p \boldsymbol{p}^e dV, \quad [8 \times 2][2 \times 4][4 \times 1] \tag{3.15}$$

and in global form:

$$\boldsymbol{F} = -\underbrace{\sum_{e=1}^{n_e} \mathbf{L}_u^{e\top} \int_{\Omega_e} \mathbf{N}^\top \mathbf{B}_p dV \, \mathbf{L}_p^e}_{\mathbf{H}} \; \boldsymbol{p}, \tag{3.16}$$

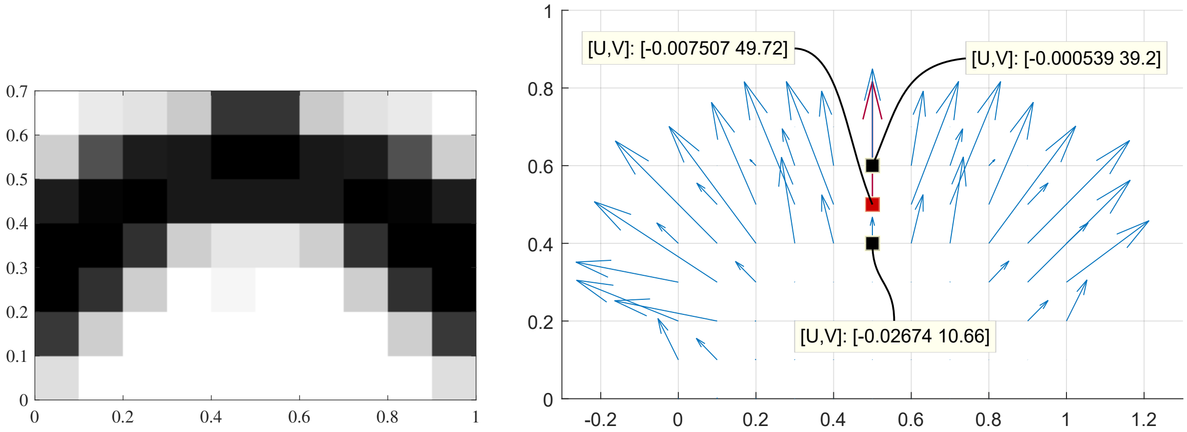
where:

$\boldsymbol{F}^e$	=	consistent load vector of size [8 × 1] (N),
$\mathbf{N}(\xi,\eta)$	=	matrix of shape functions used to express the displacement field,
$\mathbf{B}_p(\xi,\eta)$	=	matrix of spatial derivatives $= \boldsymbol{N}_{p,\boldsymbol{\xi}}$ (m^{-1}),
$\boldsymbol{p}^e$	=	nodal pressure values on element e ($\mathrm{N m^{-2}}$),
$\mathbf{H}$	=	global conversion matrix (m^2),
$\mathbf{L}_u^e$	=	indexing matrix $[4d \times nd]$,
$\mathbf{L}_p^e$	=	indexing matrix $[4 \times n]$,

$\mathbf{L}_u^e$ and $\mathbf{L}_p^e$ are indexing matrices in which every row contains a single entry of value 1. The location of this entry gives the location of the local DOF in the global displacement vector.

The subscript p in $\mathbf{B}_p$ indicates that this matrix of derivatives originates from the pressure field derivation in Appendix B.3.

To show that this derivation holds correct values, Figure 3.5 shows a 'minimum compliance' example with fixed points at $\boldsymbol{x} = [0,\ 0.2]^\top$ and $\boldsymbol{x} = [1,\ 0.2]^\top$. A pressure of 1 bar is applied to the bottom, which equals to $1 \times 10^5\,\mathrm{Pa} = 1 \times 10^5\,\mathrm{N m^{-2}}$. The thickness of the elements are set to: $t = 0.01\,\mathrm{m}$ and the element width is $\Delta x = 0.1\,\mathrm{m}$, as is the element height. The arrows in Figure 3.5b indicate the consistent nodal force vector $\boldsymbol{F}$. The three vertical arrows in the centre have labels indicating the horizontal and vertical component ([U, V]) of the load vector. The sum of these three vectors should (roughly) correspond with the pressure times the element area. So the vertical nodal loads in the centre column add up to: $0.05 + 0.10 + 0.09 + 0.08 + 10.66 + 49.72 + 39.20 + 0.1 = 100\,\mathrm{N}$, now the pressure force contributes: $p\Delta A = pt\Delta x = 1 \times 10^5\,\mathrm{N m^{-2}} \times 1 \times 10^{-3}\,\mathrm{m}^2 = 100\,\mathrm{N}$, thereby confirming that the pressure field is correctly translated into consistent nodal forces using the global conversion matrix $\mathbf{H}$.



(a) Solution of a minimum compliance problem

(b) Consistent nodal load vector representing the pressure field

Figure 3.5: The solution of a minimum compliance problem with few elements to check the translation between pressure field and consistent nodal force vector.

4

Compliant mechanism objectives

A compliant mechanism objective is some relevant relation between an output port deflection and an input force. Often, a (virtual) spring is considered at the output port with a certain spring stiffness, and the compression of this spring is part of the objective. There are numerous papers that propose different compliant mechanism objectives. Deepak et al, 2009, [9] did a comparative study between five methods and concluded that for a sufficiently high spring stiffness the objectives all converge to more or less the same topology. Therefore, this work only considers two proposed methods, one aims to optimise the *mechanical advantage* (MA) proposed by O. Sigmund, 1997 [24] and the other optimises the *energy efficiency* (EE) of the load transfer by L. Howell 2002 [15]. This chapter explains both objectives and derives their sensitivities. If the reader is unfamiliar with compliant mechanism objectives, Appendix D offers a more complete introduction and explanation of the mathematics behind it.

4.1. Types of objectives

The strain energy in a single linear spring is given as $\frac{1}{2}ku^2$, with k being the spring constant and u the deflection. The strain energy (SE) of a deformed linear elastic continuum body can be stated as [15]:

$$\mathrm{SE} = \frac{1}{2}\int_V \sigma\varepsilon dV. \tag{4.1}$$

In a finite element format (see Appendix B.4 or [15]) the strain energy is expressed as:

$$\mathrm{SE} = \frac{1}{2}\boldsymbol{u}^\top \mathbf{K}\boldsymbol{u}. \tag{4.2}$$

In a basic compliant mechanism objective the evaluation of two load cases are required. The first one only considers a point-load P_1 at the input port (port 1) as shown in Figure 4.1b, and the second only considers a dummy point-load P_2 at the output port (port 2), that specifies the location and direction of the output port as shown in Figure 4.1c. The load cases satisfy:

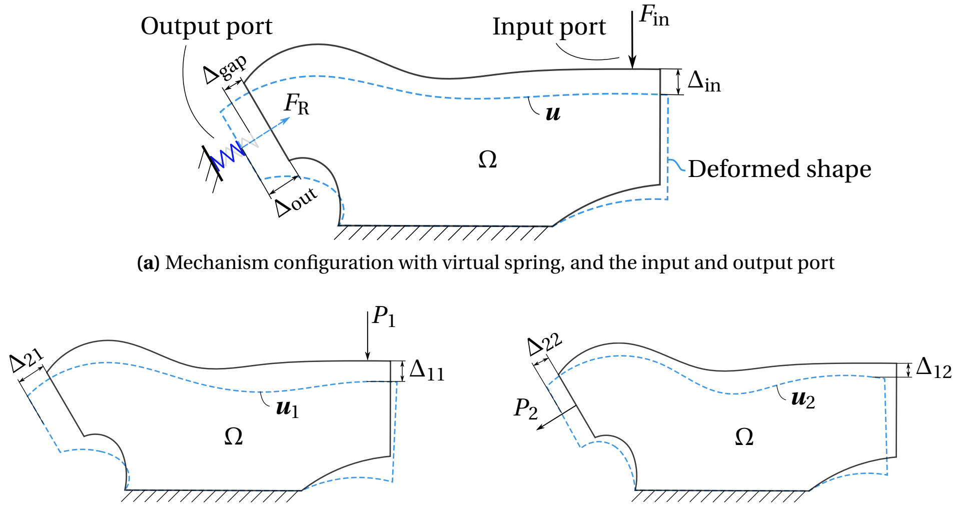
$$\mathbf{K}\boldsymbol{u}_1 = \boldsymbol{F}_1(P_1) \quad \text{and} \quad \mathbf{K}\boldsymbol{u}_2 = \boldsymbol{F}_2(P_2). \tag{4.3}$$

These two load cases are necessary to calculate the objective functions, for example the 'Mechanical Advantage' (MA) objective [24] that aims to maximise $M = F_{\text{out}}/F_{\text{in}}$.

At the output port, a virtual spring is considered with spring stiffness k_s. It is positioned such that there is a gap between the solid and the spring of size Δ_{gap}, see Figure 4.1a. To express the displacement at port i due to load case j, the following can be stated:

$$\Delta_{ij} = \boldsymbol{u}_j^\top \hat{\boldsymbol{e}}_i = \boldsymbol{u}_j^\top \frac{\boldsymbol{F}_i}{P_i} = \boldsymbol{u}_j^\top \mathbf{K}\boldsymbol{u}_i \frac{1}{P_i}, \qquad \forall \begin{Bmatrix} i \in \{1,2\} \\ j \in \{1,2\} \end{Bmatrix}, \quad \text{(no summation over } i,j\text{)}, \tag{4.4}$$

where $\hat{\boldsymbol{e}}_i$ is a unit vector that points to the desired DOFs which designate the direction of port i. The variable Δ_{ij} is clearly illustrated in Figure 4.1b and 4.1c for all combinations of i and j. Note that the final form of Equation (4.4) is very similar to the SE expression Equation (4.2). Because $\boldsymbol{F}_i$, by definition, has a total length of P_i, $\hat{\boldsymbol{e}}_i = \frac{\boldsymbol{F}_i}{P_i}$.



(a) Mechanism configuration with virtual spring, and the input and output port

(b) Load case 1 with an input load

(c) Load case 2 with a dummy output load

Figure 4.1: (a) The overall load configuration defined on the design domain Ω consisting of an input load F_{in} and the reaction load F_{R}[24]. (b) and (c) show two load cases. A linear superposition of the two displacement fields $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ resulting from the load cases gives the total displacement field $\boldsymbol{u}$.

Because of the superposition theorem of linear problems [24], we can state that the spring deflection is:

$$\Delta_s = \Delta_{\text{out}} - \Delta_{\text{gap}} = \Delta_{21} - c\Delta_{22} - \Delta_{\text{gap}}, \tag{4.5}$$

where c is related to the spring force that is present because of the deformation. c is a proportionality constant that indicates the contribution of the second (dummy) load case to the total displacement field and is derived in Appendix D. It is repeated here for convenience:

$$c = \frac{\Delta_{21} - \Delta_{\text{gap}}}{\Delta_{22} + \frac{P_2}{k_s}}.$$

In this work, however, the input load $\boldsymbol{F}_1$ is not formed by a single point load P_1 but rather by a distributed, design dependent, pressure field. So $\boldsymbol{F}_1$ is the consistent nodal load vector containing the pressure field $\boldsymbol{p}$, and $\boldsymbol{F}_2$ only contains one or two non-zero entries to indicate the location and direction of the output port. $\boldsymbol{F}_2$ should have a total length equal to P_2.

The unit vector $\hat{\boldsymbol{e}}_i$, from Equation (4.4), is used in the notation for the mechanical advantage. However, $\hat{\boldsymbol{e}}_1$ (the unit vector of F_1) is poorly defined because P_1 does not exist[1]. Hence, Δ_{11} and Δ_{12} are not defined. Notice that if we would simply chose some P_1 value, Δ_{11} can be used to express the strain energy:

$$\text{SE}_{\text{input}} = \frac{1}{2}\Delta_{11}P_1, \tag{4.6}$$

where the division by P_1 (in Δ_{11}) is corrected by the multiplication. This allows the Δ_{ij} formulation to be used in both the mechanical advantage (MA) formulation and the energy efficiency formulation.

The classical MA (that requires maximisation) is given by:

$$M = \frac{F_{\text{out}}}{F_{\text{in}}} = \frac{\Delta_s k_s}{P_1}, \tag{4.7}$$

[1] $\hat{\boldsymbol{e}}_1$ cannot be defined here because the length of $\boldsymbol{F}_1$ depends on the surface area of the interface Γ_{p_b} it encounters. Besides that, F_1 has non-zero components that act on more than one node, making the existence of the unit vector of $\boldsymbol{F}_1$ arguable.

where F_{in} is in our case undefined. Several ways can be concocted to cope with this issue. Two questions are important here: 1) is a design dependent F_{in} desired? 2) can the MA objective value M still be an intuitive value when using a pressure load at the input? Imagine F_{in} would be chosen as: $F_{\text{in}} = p\int_{\Gamma_{p_b}} dA$, then the variable p would be a constant and the only design dependent aspect here would be the exposed area A. So if improvements to the objective are sought by the optimiser, it could choose to minimise the exposed area which is often undesirable, especially in the perspective of single walled soft robotics like designs. Calculation of F_{in} is attempted by using the force vector $\boldsymbol{F}_1$ which contains the integrated pressure field:

$$\begin{aligned} F_{\text{in}} &= \sum_{k=1}^{n} \sqrt{F_{2k-1}^2 + F_{2k}^2} && \text{for } d = 2, \\ F_{\text{in}} &= \sum_{k=1}^{n} \sqrt{F_{3k-2}^2 + F_{3k-1}^2 + F_{3k}^2} && \text{for } d = 3, \end{aligned} \tag{4.8}$$

where F_i, $\forall\, i \in \mathbb{D}_u$, are components of $\boldsymbol{F}_1$. The derivative of Equation (4.8) to $\boldsymbol{\rho}$, however, requires the derivative of each individual component F_i of $\boldsymbol{F}_1$, which requires nd Lagrange multiplier vectors to be solved and is computationally very expensive. This is explained and derived in Appendix G.

The formulation could be simplified by proposing: $F_{\text{in}} = \sum_{e=1}^{n} |F_i| = \text{sgn}\big(\boldsymbol{F}_1\big)^\top \boldsymbol{F}_1$, where $\text{sgn}\big(\boldsymbol{F}_1\big)^\top$ is used to cover the absolute sign. This formulation is easily differentiable, it only requires one Lagrange multiplier solution per iteration, but it is biased to vertical and horizontal walls of the pressure boundary (Because F_{in} is now 1.41 times higher at any 45 degree angle), and it still lacks meaning.

As a result of these considerations, and by lack of meaning in a pressure case, the MA objective will not be used. Instead, the output force (**OF**) is used as the objective, defined as the spring deflection times the spring constant:

$$F_{\text{out}} = \Delta_s k_s, \tag{4.9}$$

this will promote a maximum use of the pressure.

As an alternative objective, the energy efficiency (EE) is considered, defined as the energy in the spring divided by the input energy. The input energy is equal to the input load vector times the total displacement field, where the latter is a combination of the displacement fields of both load cases. The value of c from Equation (4.1) is used to calculate the total displacement field: $\boldsymbol{u} = \boldsymbol{u}_1 - c\boldsymbol{u}_2$. The input energy is now given as:

$$E_{\text{in}} = \boldsymbol{u}^\top \boldsymbol{F}_1 = \left(\boldsymbol{u}_1{}^\top - c\boldsymbol{u}_2{}^\top\right)\boldsymbol{F}_1 = \left(\boldsymbol{u}_1{}^\top - c\boldsymbol{u}_2{}^\top\right)\mathbf{K}\boldsymbol{u}_1 = \boldsymbol{u}_1{}^\top\mathbf{K}\boldsymbol{u}_1 - c\boldsymbol{u}_2{}^\top\mathbf{K}\boldsymbol{u}_1 = (\Delta_{11} - c\Delta_{12})P_1, \tag{4.10}$$

The output energy can simply be found using the expression for energy in a linear spring, hence:

$$E_{\text{out}} = \frac{1}{2}\Delta_s^2 k_s. \tag{4.11}$$

where Δ_s is the compression of the spring. Note that this energy is positive for both pulling and pushing, to correct for this, $\text{sgn}(\Delta_s)$ is used in the EE objective that will give a negative energy for a negative deflection.

The **EE** objective is given by:

$$\frac{E_{\text{out}}}{E_{\text{in}}} = \frac{\frac{1}{2}\Delta_s^2 k_s}{\frac{1}{2}(\Delta_{11} - c\Delta_{12})P_1}, \quad \Rightarrow \quad \psi = \frac{\text{sgn}(\Delta_s)\Delta_s^2 k_s}{(\Delta_{11} - c\Delta_{12})P_1}, \tag{4.12}$$

where,

k_s	=	output spring constant ($\text{N}\,\text{m}^{-1}$),
P_1	=	the magnitude of the input point-load if used, (N) otherwise set: $P_1 = 1$,
c	=	proportionality constant,
Δ_s	=	spring deflection (m),
Δ_{11}, Δ_{12}	=	input energy per applied pressure (m^3),
Δ_{ij}	=	displacement at port i due to load case j (m),
E_{in}	=	strain energy by the pressure load (J),
E_{out}	=	strain energy in the output spring (J).

Notice that OF is not dimensionless while EE is. However, both formulations work properly and give satisfactory results. Both the OF and the EE need to be maximized, therefore a minus sign is added in the objective function to convert it to a minimization problem.

$$\begin{aligned}
\min_{\boldsymbol{\rho}}: \quad & C = -F_{\text{out}}(\boldsymbol{\rho}) \quad \text{or} \quad C = -\psi(\boldsymbol{\rho}), \\
\text{subject to}: \quad & \mathbf{A}\boldsymbol{p} = \boldsymbol{f}, \\
& \mathbf{K}\boldsymbol{u}_1 = \boldsymbol{F}_1(\boldsymbol{\rho}) = -\mathbf{H}(\boldsymbol{\rho})\boldsymbol{p}, \\
& \mathbf{K}\boldsymbol{u}_2 = \boldsymbol{F}_2, \\
& \mathbf{0} \leq \boldsymbol{\rho} \leq \mathbf{1}, \\
& g_1 = \frac{\sum_e \rho_e}{n_{\text{e}} V^*} - 1 < 0,
\end{aligned} \tag{4.13}$$

where V^* is the allowed volume fraction ($V^* = 0.5$ means that a maximum of 50% of the domain may be filled with material). Both of these expressions require more constraints to achieve a more fail proof optimisation, this is treated in Chapter 5.

4.2. Sensitivity analysis

Sensitivities are required by the optimization algorithm to give it a sense of direction. The sensitivities are calculated by taking the derivative of the objective function to each design variable:

$$S_e = \frac{dC}{d\rho_e}. \tag{4.14}$$

To achieve this, the derivative $\Delta'_{ij} \equiv \frac{d\Delta_{ij}}{d\rho_e}$ is required[2] for each combination of i and $j \in (1,2)$.

$\Delta_{ij}(\boldsymbol{\rho})$ can be differentiated using the adjoint method [30] with four adjoint equations (with Lagrange multipliers). These four adjoint (algebraic systems of) equations are: the two load cases Equation (4.3), and the two pressure fields Equation (3.11) for both i and j. The reason why both port i and port j are considered to possibly have a pressure field is because of two reasons: one, they both are used when calculating Δ'_{11}, and two, a pressure field could also be specified to exist on the output port. The derivation of Δ'_{ij} is provided in Appendix E, it also contains the expressions of c', and Δ'_s.

The derivatives of the three objective functions can now be expressed in terms of Δ'_{ij} for $i, j \in (1,2)$, c' and $\Delta s'$.

The derivative of the minimum compliance (**MC**) objective, Equation (4.6) is:

$$\frac{dC}{d\rho_e} = P_1 \Delta'_{11}, \tag{4.15}$$

The derivative of the **OF**, Equation (4.9) is:

$$\frac{dF_{\text{out}}}{d\rho_e} = k_s \Delta'_s = k_s \left(\Delta'_{21} - c'\Delta_{22} - c\Delta'_{22}\right), \tag{4.16}$$

The derivative of the **EE**, Equation (4.12) is:

$$\frac{d\psi}{d\rho_e} = \text{sgn}(\Delta_s) \frac{2\Delta_s \Delta'_s \left(\Delta_{11} - c\Delta_{12}\right) - \Delta_s^2 \left(\Delta'_{11} - c'\Delta_{12} - c\Delta'_{12}\right)}{\left(\Delta_{11} - c\Delta_{12}\right)^2} \frac{k_s}{P_1}. \tag{4.17}$$

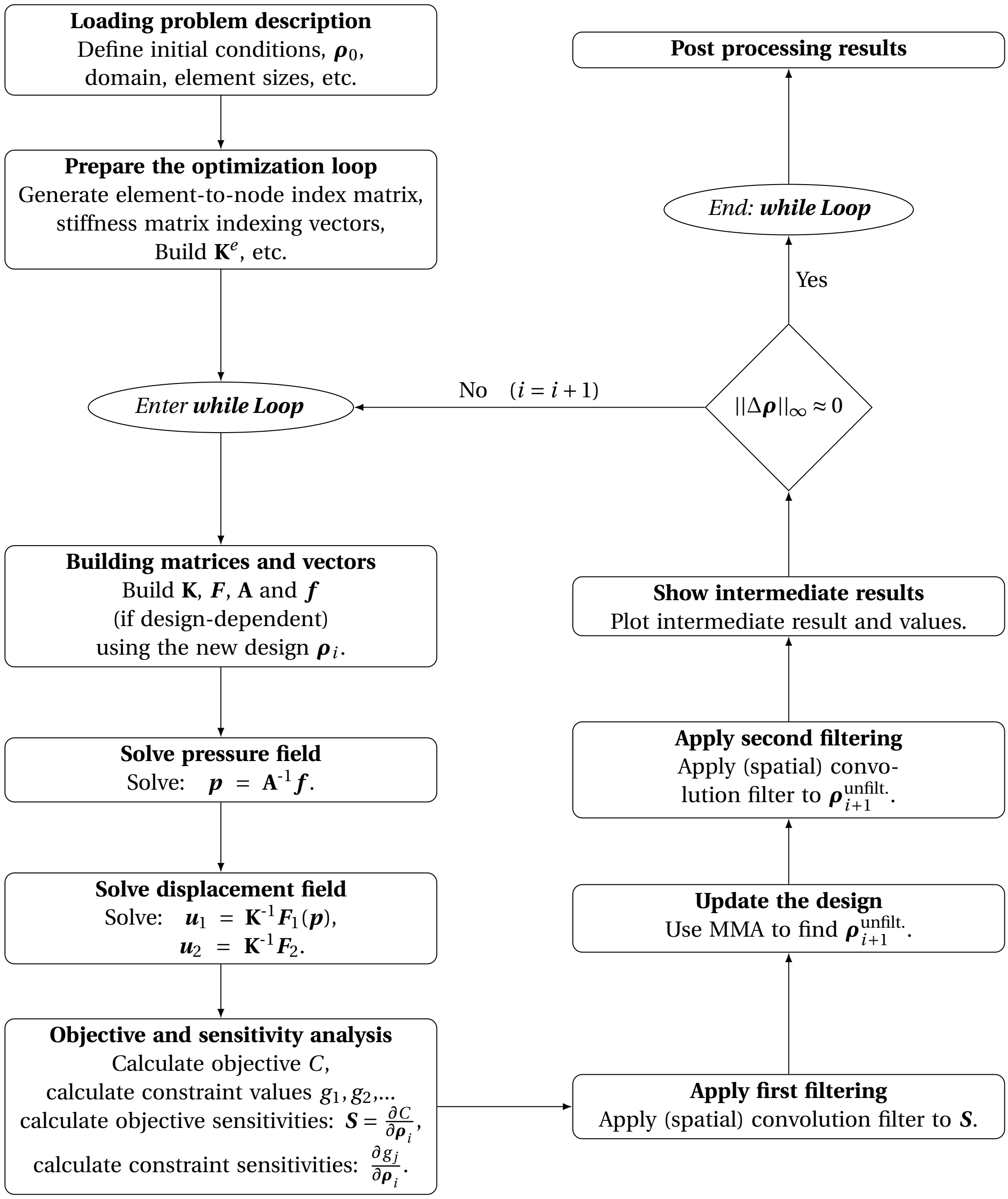
Appendix E elaborates on these sensitivities further and offers the full derivation. Every analytically derived sensitivity is checked using a finite difference method.

[2] Where the apostrophe notation is introduced in Section 2.1.1.

4.3. The Topology Optimisation cycle

This section gives a brief explanation of the optimisation cycle. The flow chart displayed below indicates the required steps for a TO program. Notice that solving the pressure field is a standalone part of the optimisation cycle and does not invoke radical changes in the work flow. Because density filtering is applied, the new design (returned by the MMA optimiser) is called $\boldsymbol{\rho}_{i+1}^{\text{unfilt.}}$, after filtering this becomes $\boldsymbol{\rho}_{i+1}$. To adjust the sensitivities accordingly, the chain-rule is implicitly applied by filtering the sensitivities in the first filtering step. Where the bottom four boxes are iterated to update the design and make it converge.

The convergence is implicitly checked by looking at the inf-norm of the design change $||\Delta\boldsymbol{\rho}||_\infty$ (being the largest absolute value of $\Delta\boldsymbol{\rho}$) where $\Delta\boldsymbol{\rho} = \boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i$, if this is smaller than a given value the optimization procedure is assumed to have sufficiently converged and stops.


Loading problem description
Define initial conditions, $\boldsymbol{\rho}_0$,
domain, element sizes, etc.
Post processing results
Prepare the optimization loop
Generate element-to-node index matrix,
stiffness matrix indexing vectors,
Build $\mathbf{K}^e$, etc.
End: ***while Loop***
Yes
No $(i = i + 1)$
Enter ***while Loop***
$||\Delta\boldsymbol{\rho}||_\infty \approx 0$
Building matrices and vectors
Build $\mathbf{K}$, $\boldsymbol{F}$, $\mathbf{A}$ and $\boldsymbol{f}$
(if design-dependent)
using the new design $\boldsymbol{\rho}_i$.
Show intermediate results
Plot intermediate result and values.
Apply second filtering
Apply (spatial) convolution filter to $\boldsymbol{\rho}_{i+1}^{\text{unfilt.}}$.
Solve pressure field
Solve: $\boldsymbol{p} = \mathbf{A}^{-1}\boldsymbol{f}$.
Update the design
Use MMA to find $\boldsymbol{\rho}_{i+1}^{\text{unfilt.}}$.
Solve displacement field
Solve: $\boldsymbol{u}_1 = \mathbf{K}^{-1}\boldsymbol{F}_1(\boldsymbol{p})$,
$\boldsymbol{u}_2 = \mathbf{K}^{-1}\boldsymbol{F}_2$.
Objective and sensitivity analysis
Calculate objective C,
calculate constraint values g_1, g_2,...
calculate objective sensitivities: $\boldsymbol{S} = \frac{\partial C}{\partial \boldsymbol{\rho}_i}$,
calculate constraint sensitivities: $\frac{\partial g_j}{\partial \boldsymbol{\rho}_i}$.
Apply first filtering
Apply (spatial) convolution filter to $\boldsymbol{S}$.

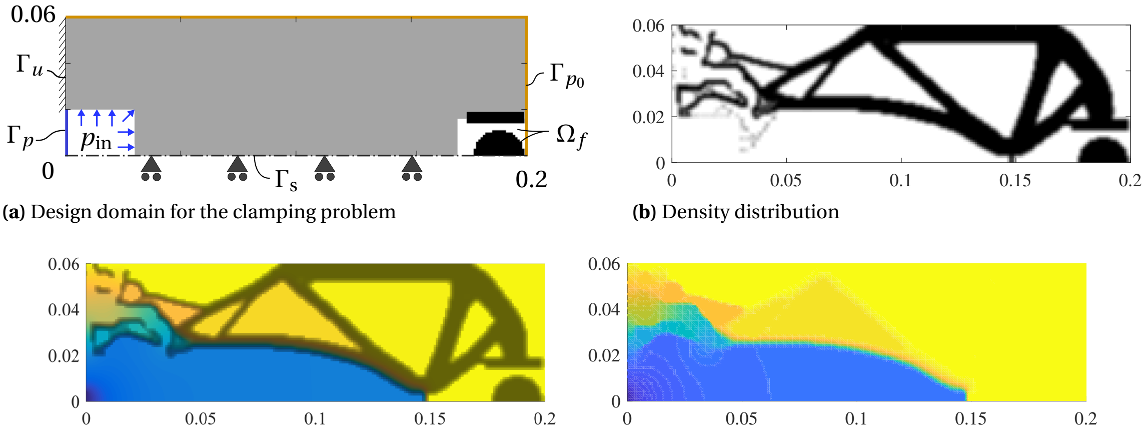
5

Adding model constraints

In theory, the optimisation problem is completely defined. However, the proposed method still has some shortcomings which it can exploit. The constraints, introduced in this chapter, aim to suppress these weaknesses.

5.1. A hitch in the model

One thing the model does not take care of is the creation of open boundaries as can be seen in Figure 5.1, the algorithm can exploit open boundaries as they give no mechanical resistance and only leak slightly. Another example is the extreme input deflection that is sometimes beneficial to the objective. To prevent this, two additional constraints are proposed. One is the input deflection constraint, for which several varieties are possible, and the second is to prevent the outgoing volume flow by constraining it.



(a) Design domain for the clamping problem

(b) Density distribution

(c) Pressurefield with design overlay

(d) Pressure field

Figure 5.1: The optimisation algorithm exploits the fact that an open boundary allows frictionless bending and that the design can still realize a large pressure drop. (a) the design space with boundary conditions. In (b) the density distribution (or design) is given after 800 optimisation iterations. (c) shows combined plot where the design is displayed on top of the pressure field. (d) the pressure field calculated as a function of the current density distribution.

5.2. Input deflection constraint

In the classical mechanical advantage (MA) objective proposed in [24] the maximum internal stress is indirectly limited by limiting the deflection at the input port. Because this work uses a similar objective: the OF objective, this section aims to provide the input deflection constraint for pressure load problems.

Using an input deflection constraint in a pressure load problem can have the side effect of limiting the surface area of the pressure interface Γ_{p_b}, whilst this is sometimes not desired.

Sigmund [24], uses the following equation for the input deflection:

$$\Delta_{\text{in}} = \Delta_{11} - c\Delta_{12} = (\boldsymbol{u}_1^\top - c\boldsymbol{u}_2^\top)\hat{\boldsymbol{e}}_1 \qquad \text{where}: \quad \hat{\boldsymbol{e}}_1 = \frac{\boldsymbol{F}_1}{P_1}, \tag{5.1}$$

$\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are the displacement fields of the first and second load case and $\hat{\boldsymbol{e}}_1$ is the unit vector indicating the direction of the input port load. When working with a pressure load, P_1 and therefore $\hat{\boldsymbol{e}}_1$ are undefined and cannot simply be replaced as explained in Section 4.1. To circumvent this problem we can use the input strain energy (SE), Equation (4.6) as a measure for input deflection. The SE of the input loading depends on p_{in}, $\boldsymbol{n}$ and $\boldsymbol{u}$, being the pressure load, surface normal and displacement of that element respectively. The continuous expression of the SE is the integral:

$$\text{SE}_{\text{in}} = p_{\text{in}} \int_{\Gamma_{p_b}} \boldsymbol{u} \cdot \boldsymbol{n} \, dA, \tag{5.2}$$

over the inner pressure interface Γ_{p_b}, and where the combined, total deformation field: $\boldsymbol{u} = \boldsymbol{u}_1 - c\boldsymbol{u}_2$. Because the applied pressure, p_{in}, is constant, the constraint on the SE limits the exposed area and its deformation. A discretised form of the SE is simply:

$$\text{SE}_{\text{in}} = \boldsymbol{u}^\top \boldsymbol{F}_1 = \left(\boldsymbol{u}_1 - c\boldsymbol{u}_2\right)^\top \mathbf{K}\boldsymbol{u}_1 = \boldsymbol{u}_1^\top \mathbf{K}\boldsymbol{u}_1 - c\boldsymbol{u}_2^\top \mathbf{K}\boldsymbol{u}_1 = \left(\Delta_{11} - c\Delta_{12}\right) P_1, \tag{5.3}$$

where $\boldsymbol{F}_1$ is the input load vector and in the case of a pressure field, the point-load: $P_1 = 1$ (or any random value as it cancels anyway). Doing so allows the general use of the Δ notation. The constraint then takes the form:

$$g_2 = \frac{\text{SE}_{\text{in}}}{\text{SE}_{\text{max}}} - 1 \leq 0. \tag{5.4}$$

SE_{max} is a parameter to control the maximum input energy. Using Equation (E.10), the sensitivities can be written as:

$$\frac{dg_2}{d\rho_e} = \frac{P_1}{\text{SE}_{\text{max}}}(\Delta'_{11} - c'\Delta_{12} - c\Delta'_{12})$$

5.3. Outflow constraint

To prevent the open boundary issue, shown in Figure 5.1, a constraint can be used to limit the volume flux through the design. This is achieved by limiting the volume flux through the outer boundaries[1] Γ_{p_0} (where the prescribed pressure is $P_{\text{out}} = 0$), the different boundary types were introduced in Section 2.1.2 and the relevant ones are repeated in Figure 5.2. Since Γ_{p_0} is the only location where the volume flow can leave the domain it is not possible to set the allowed volume flux to zero, this would (in the model) require an infinite amount of material between the inflow and outflow boundaries or a flow coefficient of zero. If a boundary does not have a prescribed pressure or outflow, by way of formulation, it automatically satisfies a symmetry condition which means no outflow is present.

Tuning such a constraint can cause the boundaries to be closed and thin hinges to be thickened and elongated (to respectively prevent leaking and maintain flexibility). The outflow is the integral:

$$Q_{\text{out}} = \int_{\Gamma_{p_0}} \boldsymbol{q}(\boldsymbol{x}) \cdot \boldsymbol{n} \, d\Gamma \tag{5.5}$$

This is now discritised by substituting the element wise expression for $\boldsymbol{q} = -K(\rho_e)\mathbf{B}_p(\boldsymbol{x})\boldsymbol{p}^e$, and sum over all elements:

$$\begin{aligned} Q_{\text{out}} &= \sum_{e=1}^{n_e} \int_{\Gamma_{p_0}} \left(\left(-K(\rho_e)\mathbf{B}_p(\boldsymbol{x})\boldsymbol{p}^e\right) \cdot \boldsymbol{n}(\boldsymbol{x})\right) d\Gamma, \\ &= \underbrace{-\sum_{e=1}^{n_e} K(\rho_e) \int_{\Gamma_{p_0}} \left(\boldsymbol{n}^\top(\boldsymbol{x}) \, \mathbf{B}_p(\boldsymbol{x})\right) d\Gamma \boldsymbol{p}^e.}_{\boldsymbol{k}^\top \mathbf{G}\boldsymbol{p}} \end{aligned} \tag{5.6}$$

[1] This can, theoretically, also be done by integrating over the input boundary or volume flux source and minimizing the inflow from this source. However, this would have a different effect because the 'drained' fluid (using the drainage term) is 'pulled' from the source, adding to the inflow, while the outflow would be lower when all volume flow is drained.

The outflow can be formulated as an integral over the boundary, for this the parametrisation v is introduced in the ξ, η domain to integrate over, see Figure 5.2. If the integration goes from node 2 to node 4 (the right line), v can be expressed as $\boldsymbol{N}(v) = \boldsymbol{N}(\xi = 1, \eta)$. Note that $\boldsymbol{x}(\boldsymbol{\xi})$ and $\boldsymbol{\xi}(v)$, using this parametrisation the integral can be written as:

$$\begin{aligned} Q_{\text{out}} &= -\sum_{e=1}^{n_{\text{e}}} K(\rho_e) \int_{\Gamma_{p_0}} \left(\boldsymbol{n}^{\top}(\boldsymbol{\xi})\, \mathbf{J}^{\text{-}1}(\boldsymbol{\xi}) \mathbf{B}_p(\boldsymbol{\xi}) \right) \sqrt{\left(\frac{dx(v)}{dv}\right)^2 + \left(\frac{dy(v)}{dv}\right)^2}\, dv \boldsymbol{p}^e, \\ &= -\sum_{e=1}^{n_{\text{e}}} K(\rho_e) \int_{\Gamma_{p_0}} \left(\boldsymbol{n}^{\top}(v)\, \mathbf{J}^{\text{-}1}(v) \mathbf{B}_p(v) \right) \sqrt{\left(\frac{dx(v)}{dv}\right)^2 + \left(\frac{dy(v)}{dv}\right)^2}\, dv \boldsymbol{p}^e, \end{aligned} \tag{5.7}$$

where,

Q_{out}	=	the volume flow through the boundary Γ_{p_0} ($\text{m}^3\,\text{s}^{-1}$),
Γ_{p_0}	=	all boundaries where p is set to zero (so where flow can leave the domain,
$\boldsymbol{n}$	=	surface normal vector,
$\mathbf{B}_p$	=	matrix of derivatives of shape functions (m^{-1}),
$\boldsymbol{p}^e$	=	vector of nodal pressures on an element ($\text{N}\,\text{m}^{-2}$).

The integral in Equation (5.7) can be solved using the quadrature rule, with abscissae v_i and weights W_i,

$$Q_{\text{out}} = -\sum_{e=1}^{n_{\text{e}}} \left(K(\rho_e) \sum_i \left(W_i \left(\boldsymbol{n}^{\top}(v_i)\, \mathbf{J}^{\text{-}1}(v_i) \mathbf{B}_p(v_i) \right) \sqrt{\left(\frac{dx(v_i)}{dv}\right)^2 + \left(\frac{dy(v_i)}{dv}\right)^2} \right) \boldsymbol{p}^e \right). \tag{5.8}$$

The unit normal vector $\boldsymbol{n}(\boldsymbol{\xi})$ has a length of one and in the ξ, η plane it always is in a beneficial, single coordinate, direction (when using a square local element). Because the implementation in this work is that of a rectangular mesh and rectangular elements, it is possible to find the analytical expressions for each of the four possible direction and speed up the computations. This way the boundary integral is replaced by the summation of small areas times the flow that goes through each area. The outflow constraint is given by:

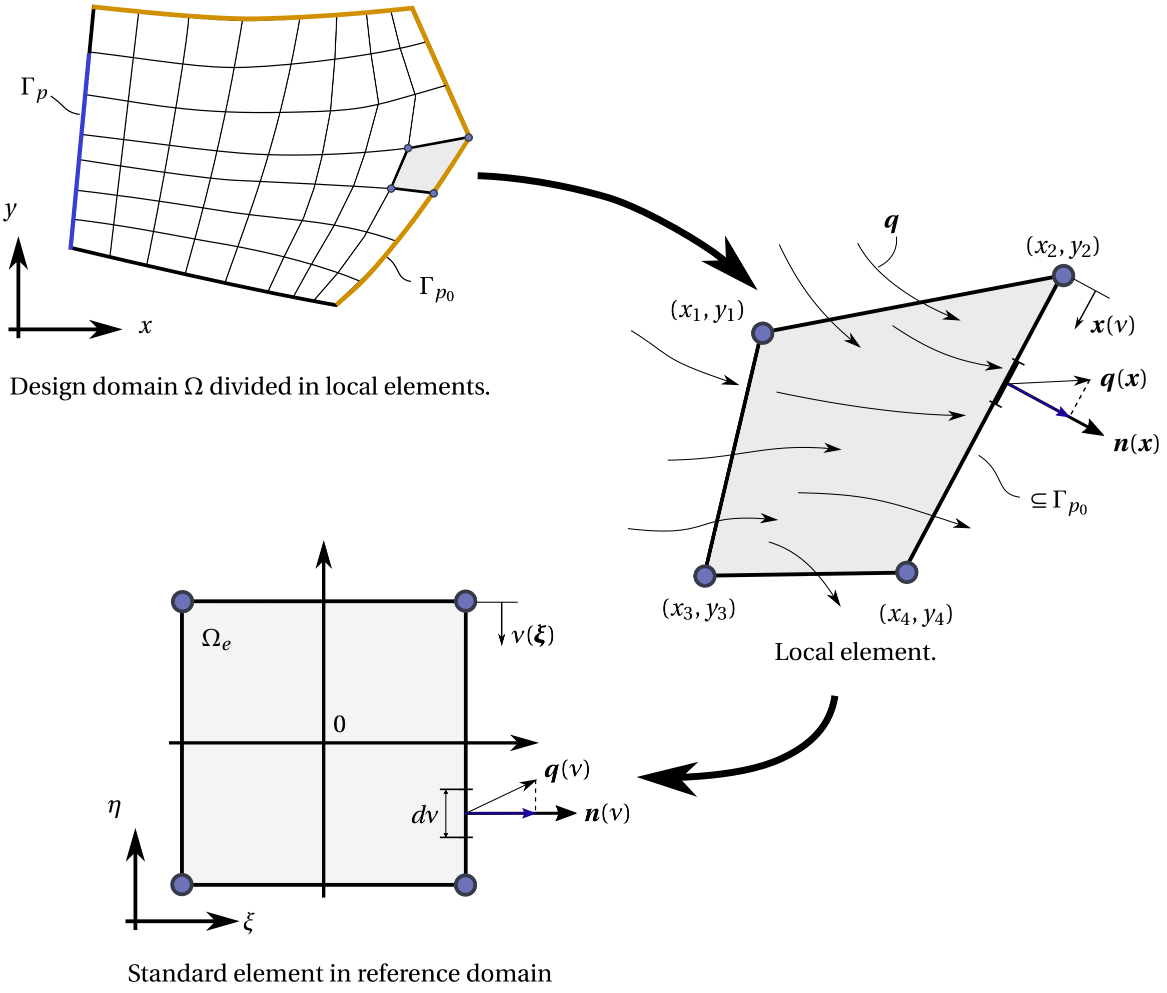
$$g_3 = \frac{Q_{\text{out}}}{Q^*_{\text{out}}} - 1. \tag{5.9}$$

The outflow constraint is very effective in preventing leaking hinges and pressure drops inside the pressurised domain. Bear in mind, however, that the optimiser will first try to satisfy this constraint before it tries to optimise the objective. Typically this means that at first, a tight enclosure is formed around the pressure application (see Figure 6.7i for clarification) before starting to optimise the objective. To prevent this, it is found to be very effective to start at a high value of $Q^*_{\text{out}} \approx 40$ and half this value each 20-30 iterations to a minimum of about $Q^*_{\text{out}} = 1$. Doing so allows the optimiser to first find the ideal location of the boundary and then close down any leakages to the outside. This method is not applied in the first few sections of the results, when it is used, the additional parameters will be specified.

The sensitivities of this constraint, $\frac{dg_3}{d\boldsymbol{\rho}}$, can be calculated by rewriting the integral equation into a matrix multiplication on global scale. The matrix $\mathbf{G}$ is introduced in Equation (5.6) as the global contribution matrix that contains the Gauss quadrature weights and abscissae to perform the boundary integration. $\boldsymbol{k}$ is defined as the vector, $k_e = K(\rho_e)$, of flow coefficients using index notation. Using the flow coefficient vector $\boldsymbol{k}$, the global contribution matrix $\mathbf{G}$, and the global pressure vector $\boldsymbol{p}$, of which $\mathbf{G}$ is independent of the design variable $\boldsymbol{\rho}$, the outflow is expressed as:

$$Q_{\text{out}} = \boldsymbol{k}(\rho)^{\top} \mathbf{G} \boldsymbol{p}(\rho), \tag{5.10}$$

The reader interested in applying this constraint is referred to Appendix F. The translation from the integral formulation in Equation (5.6) to the matrix multiplication Equation (5.10) is thoroughly described there. It also contains the sensitivities which are derived using the adjoint formulation with Lagrange multipliers.



Design domain Ω divided in local elements.

Local element.

Standard element in reference domain

Figure 5.2: The global design domain is meshed into local element and these can be mapped to the reference domain. The outflow through the boundary Γ_{p_0}, physically present in the global domain, is integrated over the element boundaries $\subseteq \Gamma_{p_0}$ of all adjacent elements. This is done using the mapping to the reference domain in which the integration is performed.

6 Results

This chapter discusses the method performance and shows resulting designs. The parameters that are used in the optimisation are introduced in the first section and every deviations from it are noted in the proceeding examples.

6.1. Baseline for the used parameters

Because there are a lot of parameters that influence the optimisation algorithm, this chapter first introduces all of the parameters such that only the deviation from this baseline needs to be stated in the proceeding examples. Table 6.1 lists all used parameters.

The optimiser will stop by default after either 1200 iterations or if the length of the difference-vector (defined as the difference between two design iterations) is below 0.005, i.e.

$$\text{if } \sqrt{\sum_i \left(|\rho^i| - |\rho^{i-1}| \right)^2} < 0.005.$$

And by default the energy efficiency (EE) objective will be used and no initial design is used, i.e. the design variables, by default, all start from an unbiased, equally distributed density that satisfies the volume fraction. Note that $h_{\text{void}}(=0)$ is not included, that is because this variable is non-existent because drainage is simply zero on the void region.

Table 6.1: The baseline with the parameters used in the optimisation examples. Any deviations from it will be specified.

Description	Variable	Value
Geometric parameters		
Width	w	200 mm
Height	h	120 mm (symmetric)
Thickness	t	10 mm
number of elements in W	N_W	200
number of elements in H	N_H	60
Material parameters		
Young's modulus	E	$3 \times 10^9\,\mathrm{N\,m^{-2}}$ (Similar to that of nylon or PET)
Poisson's ratio	ν	0.4
Optimisation parameters		
Penalisation (for SIMP)	P	3
Filtering radius	r_f	2 mm
Minimum density value	$\rho_{\min}$	1×10^{-5}
Minimum Young's modulus	$E_{\min}$	$E \times 10^{-5}\mathrm{N\,m^{-2}}$
Volume fraction	$V_{\max}$	35%
Maximum change in density	$\Delta\rho_e$	0.2 per iteration
Maximum # of iterations	n_{maxIter}	1500
Objective parameters		
Input pressure load	p_{in}	$1 \times 10^5\,\mathrm{N\,m^{-2}}$ (1 bar of pressure)
Output load	p_{out}	10 N
Output spring stiffness	k_s	$1 \times 10^4\,\mathrm{N\,m^{-1}}$
Output gap (as in [24])	Δ_{gap}	0 m
Darcy model parameters		
Allowed leakage	$Q_{\max}$	$1\,\mathrm{m^3\,s^{-1}}$
$K(\rho)$ step location	η_k	0.4
$K(\rho)$ slope at step	β_k	10
$H(\rho)$ step location	η_h	0.6
$H(\rho)$ slope at step	β_h	10
Conductivity in material	k_{mat}	$1 \times 10^{-10}\,\mathrm{m^4\,N^{-1}\,s^{-1}}$
Conductivity in void	k_{void}	$1 \times 10^{-3}\,\mathrm{m^4\,N^{-1}\,s^{-1}}$
Drainage from material	h_{mat}	given by Equation (3.6) using r and Δs, and has the unit $\mathrm{m^2\,N^{-1}\,s^{-1}}$
Remainder of input pressure at Δs	r	0.1 (so $p(\Delta s) = 0.1P_1$ after flowing trough Δs material)
Depth at which limit r is reached	Δs	2 mm

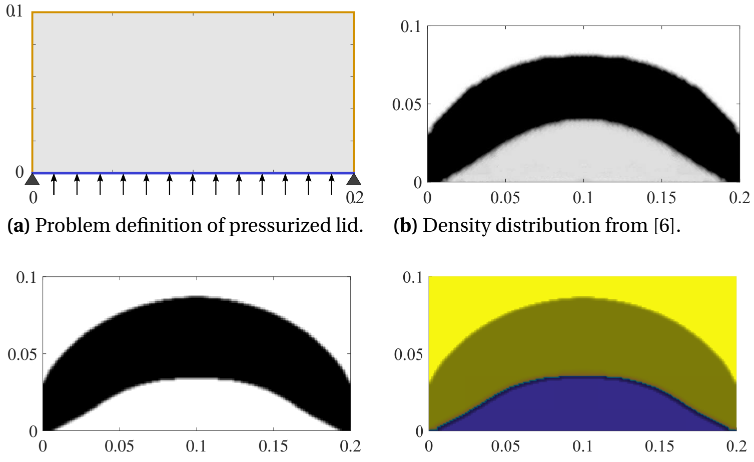
6.2. Minimum compliance solutions

A popular design problem in literature that has a design dependent load is the minimisation of compliance of a lid, with from the bottom boundary a pressure load as shown in Figure 6.1.

The two results in Figure 6.1 are very similar, a slight difference can be noted at the bottom where the Darcy approach seems to result in a bit more of a blunt design. The reason for the small difference is probably in the choice of mesh and the post processing in [6].

Another popular benchmark problem is the piston design also solved in [6]. For this analysis the Darcy approach is used with [$\eta_k = 0.2$, $\eta_h = 0.3$] and varying values of β_k and β_h. Several different results are shown in Figure 6.2. Notice that the design in Figure 6.2c allows the fluid to go through and equalises the pressure in the orifices behind the inner boundary, preventing these thin solid layers from taking up the pressure load while still allowing them to provide bending stiffness.

The values of η_k and η_h control the density threshold (ρ_T). If β_k and β_h are chosen to be very large (say > 150) the proposed method becomes very similar to the method proposed by Zheng et al. 2009 [34] which uses an electric potential field to identify the pressurized elements. The load is then applied to the element boundaries, making the sensitivities discontinuous and not differentiable. This is a strange way of cooping



(a) Problem definition of pressurized lid.

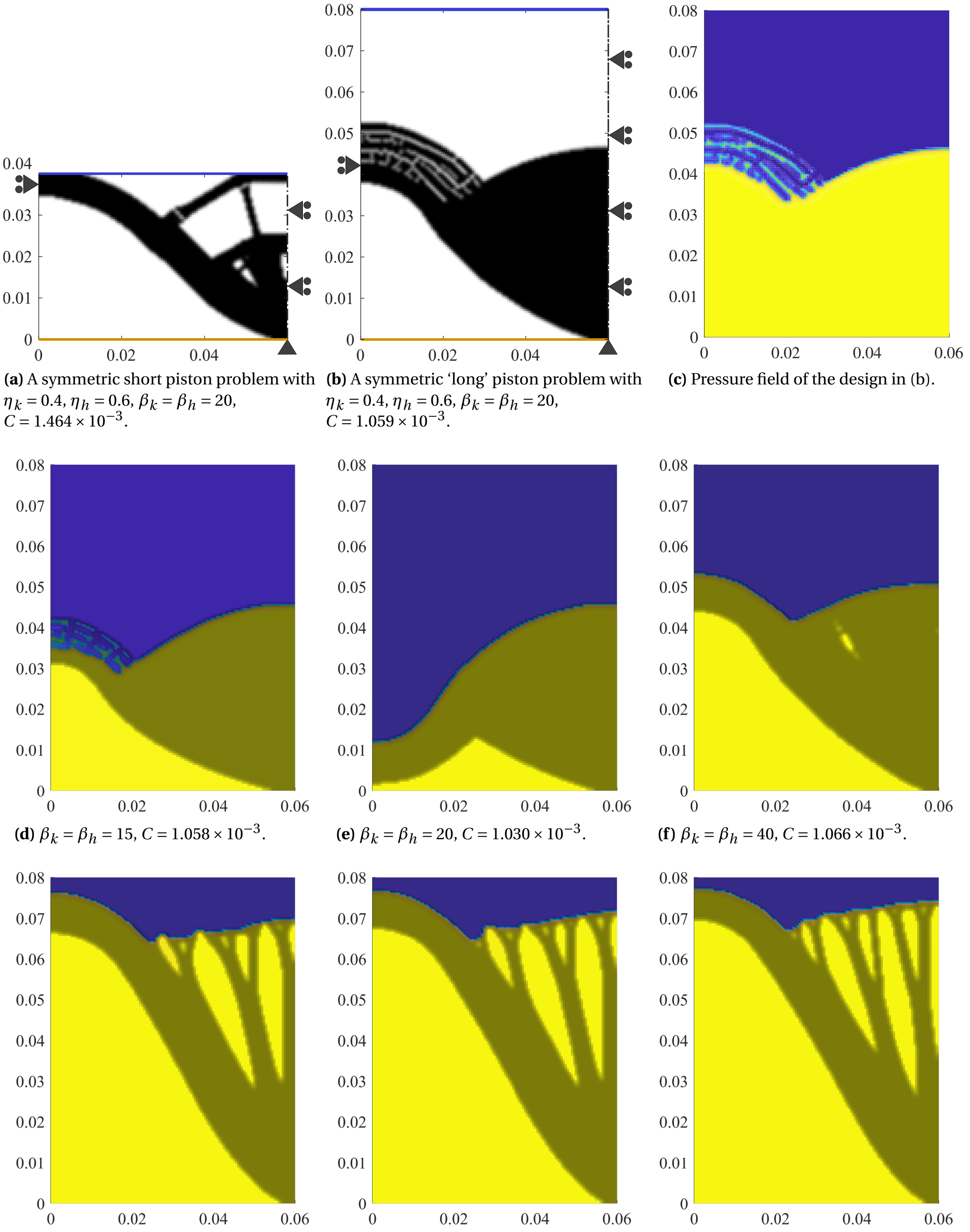
(b) Density distribution from [6].

(c) Result using the Darcy approach.

(d) Overlay plot of the Darcy approach.

Figure 6.1: [$V_{\max} = 50\%$, $N_W = 200$, $N_H = 100$]. Result of the common benchmark problem for design dependent TO. (a) and (b) show the problem definition and solution by Brugge et al. 2009, similar to a lot of other papers on design dependent loads. (c) shows the Darcy approach, the outflux constraint is not used here as the MC objective naturally discourages leakage. (d) shows the pressure field with a transparent design overlay.

with sensitivities because of the following reason. A design dependent optimisation solves two problems, the first is finding the optimised location to ‘catch’ the pressure load, and the second is finding the stiffest design to catch that load. By eliminating the load sensitivities the loading location is fully determined by stiffness considerations, if the optimiser makes a small step towards a stiffer design, it can just as well allow the load to distribute in a worse way, effectively worsening the solution.



(a) A symmetric short piston problem with $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = \beta_h = 20$, $C = 1.464 \times 10^{-3}$.

(b) A symmetric 'long' piston problem with $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = \beta_h = 20$, $C = 1.059 \times 10^{-3}$.

(c) Pressure field of the design in (b).

(d) $\beta_k = \beta_h = 15$, $C = 1.058 \times 10^{-3}$.

(e) $\beta_k = \beta_h = 20$, $C = 1.030 \times 10^{-3}$.

(f) $\beta_k = \beta_h = 40$, $C = 1.066 \times 10^{-3}$.

(g) $\beta_k = \beta_h = 80$, $C = 1.218 \times 10^{-3}$.

(h) $\beta_k = \beta_h = 100$, $C = 1.234 \times 10^{-3}$.

(i) $\beta_k = \beta_h = 200$, $C = 1.251 \times 10^{-3}$.

Figure 6.2: Two examples of a piston problem and a parameter sweep with a MC objective, C is the compliance.
(a) uses $[w = 6\,\text{cm},\ h = 4\,\text{cm},\ N_W = 120,\ N_H = 80]$ and is similar to examples from literature.
(e) uses $[w = 6\,\text{cm},\ h = 8\,\text{cm},\ N_W = 120,\ N_H = 160]$. Figures (d)-(i) use $[\eta_k = 0.2,\ \eta_h = 0.3]$ for improved results.

6.3. Compliant mechanism solutions

First the reader will be introduced to a symmetric design problem for a clamping mechanism. This design problem is then put to the test by solving this problem using different parameters and objectives. Finally, some examples are given to proof the broad range of design problems that such a method can solve.

6.3.1. Introducing a clamping problem

A typical design example in compliant mechanism designs is the clamping mechanism. The logical extension of this example to a design dependent load problem is a pressure actuated clamp or gripper. Figure 6.3 shows the full design problem with a small ball that is clamped or a walnut that needs to be cracked, the walnut stiffness is that of the output spring and is brought into the model via the objective. The figure also indicates the input pressure, the boundary where $p_{\text{out}} = 0$, and the location of the output port with the arrow indicating the $\boldsymbol{F}_2$ vector. Figure 6.4 shows half the design domain in which the symbols (indicating the prescribed boundaries) were introduced in Section 2.1.2. Because of symmetry, only half the design domain is considered in the optimisation. The line (or surface) of symmetry Γ_s now requires two boundary conditions (BC), namely a (zero outflow) Neumann BC for the pressure field and a (zero vertical displacement) Dirichlet BC for the displacement field of which the outflow condition is naturally imposed [1].

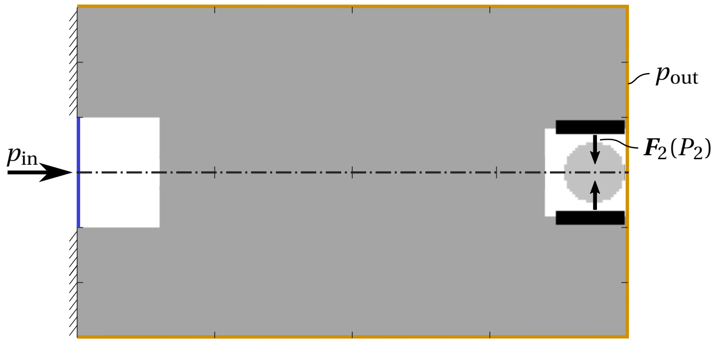


Figure 6.3: The symmetric design domain for the clamping problem with indication of the output port and the input pressure.

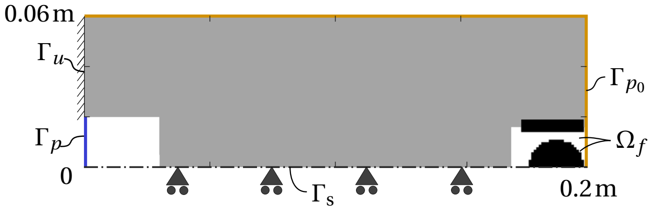


Figure 6.4: Design domain for the clamping problem with fixed void and solid regions Ω_f indicated as black and white areas, the grey area is the free design area. Imposing the void region in the lower left corner allows the pressure, defined on Γ_p, to at least cover this small region in every design iteration.

[1] Prescribing the outflow at a boundary, in the FEM formulation for the pressure field, is done by adding $-\int_\Gamma \boldsymbol{N}^\top \boldsymbol{q}_\Gamma \cdot \boldsymbol{n}\ d\Gamma$ to $\boldsymbol{f}$. The zero outflow ($\boldsymbol{q}_\Gamma = \boldsymbol{0}$) constraint in now imposed by adding zero, hence, it is naturally imposed.

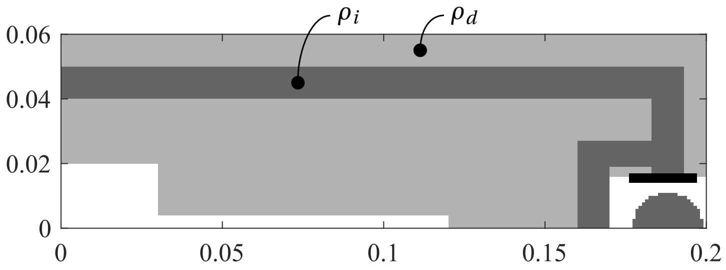


Figure 6.5: Illustration of the initial design using a parameter to prescribe the density ρ_i at this initial shape. The dependence of the optimisation algorithm to the initial design can be tested by changing ρ_i.

To experiment with initial conditions in a systematic way we also introduce Figure 6.5, where ρ_i can be changed to alter the initial design. The program adjusts ρ_d accordingly to satisfy the volume fraction constraint. The design that follows from this default baseline using an initial condition of $\rho_i = 0.6$ can be seen in Figure 6.9f.

6.3.2. Parameter sweep: flow resistance and drainage parameters

To indicate the behavioural differences of the Darcy method with changing parameters, ten TO's are executed using the EE objective. The results are shown in Figure 6.7 along with the used parameters. These figures show the capability of finding the optimum given different settings, notice that (h) has reached the highest ψ value. These results can be compared with the piston analysis shown in Figure 6.2 where the values [$\eta_k = 0.2$, $\eta_h = 0.3$ and $\beta_k = \beta_h = 20$] (used to make Figure 6.2e) resulted in the minimum compliance. It can be concluded from these two parameter sweeps that it is desirable to choose the η's slightly under the volume fraction $V_{\max}$, i.e. $\eta_k = 0.2$, $\eta_h = 0.3$ and the β's to be around 20, this being a good trade-off between differentiability (having load sensitivity) and decisiveness in defining the boundary.

The behaviour of the Darcy method is highly dependent on the η and β parameters. Leaking of the inner boundary, for example, happens in Figure 6.7: a, b, d, e, i and j. Is this a bad thing? In these 2D TO's an engineer is simplifying a 3D problem, if the η and β parameters are chosen such that 'inner boundary leakage' exists, we are using the fact that a boundary (in 3D) can both bring structural support and allow fluid to enter the next chamber, which is simply realised by drilling some holes in the membrane. As such, choosing the correct parameters provides the user with a method that is trustworthy and finds better and more versatile solutions than other methods where differentiability is not stimulated.

The behaviour of the optimisation procedure can be seen in Figure 6.6, containing the normalised objective values at each iteration. The vertical axis shows the $\psi/\psi_{\max}$ values, where $\psi_{\max}$ is the highest value of ψ that the optimisation reached. Notice that E and F take-off a lot later that the other optimisation runs, and E clearly behaves less smoothly. It is unclear what the reason is for the long delay, each optimisation first starts by blocking and dissipating the pressure to satisfy the outflow constraint (as mentioned at the end of Section 5.3) and effectively concentrate the forces it can work with. Then, in consequent iterations, the design rapidly bridges the gap between the pressure source and the output port and it lifts the blockade allowing the pressure to spread out. This only happens if it is allowed by the input displacement constraint and outflow constraint(!), this tipping point in the early stage of the design has significant influence on the final design. Concerning analysis E and F, the blockade appeared and an equally distributed grey area filled the rest of the domain (till the output port), then the design seems to be on a flat part in the objective plane as it is unable to decide where to go. Finally, just like the others, after the tipping point it quickly finds a route to satisfy the objective the most, and updates the design accordingly. It is clear that in most cases the first 40 iterations give a good indication of the final design.

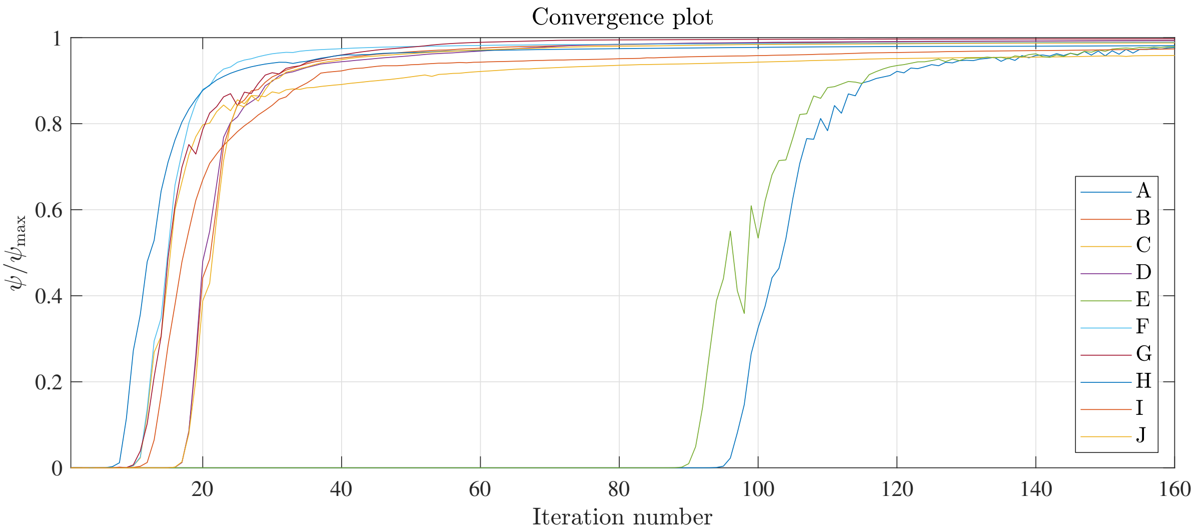
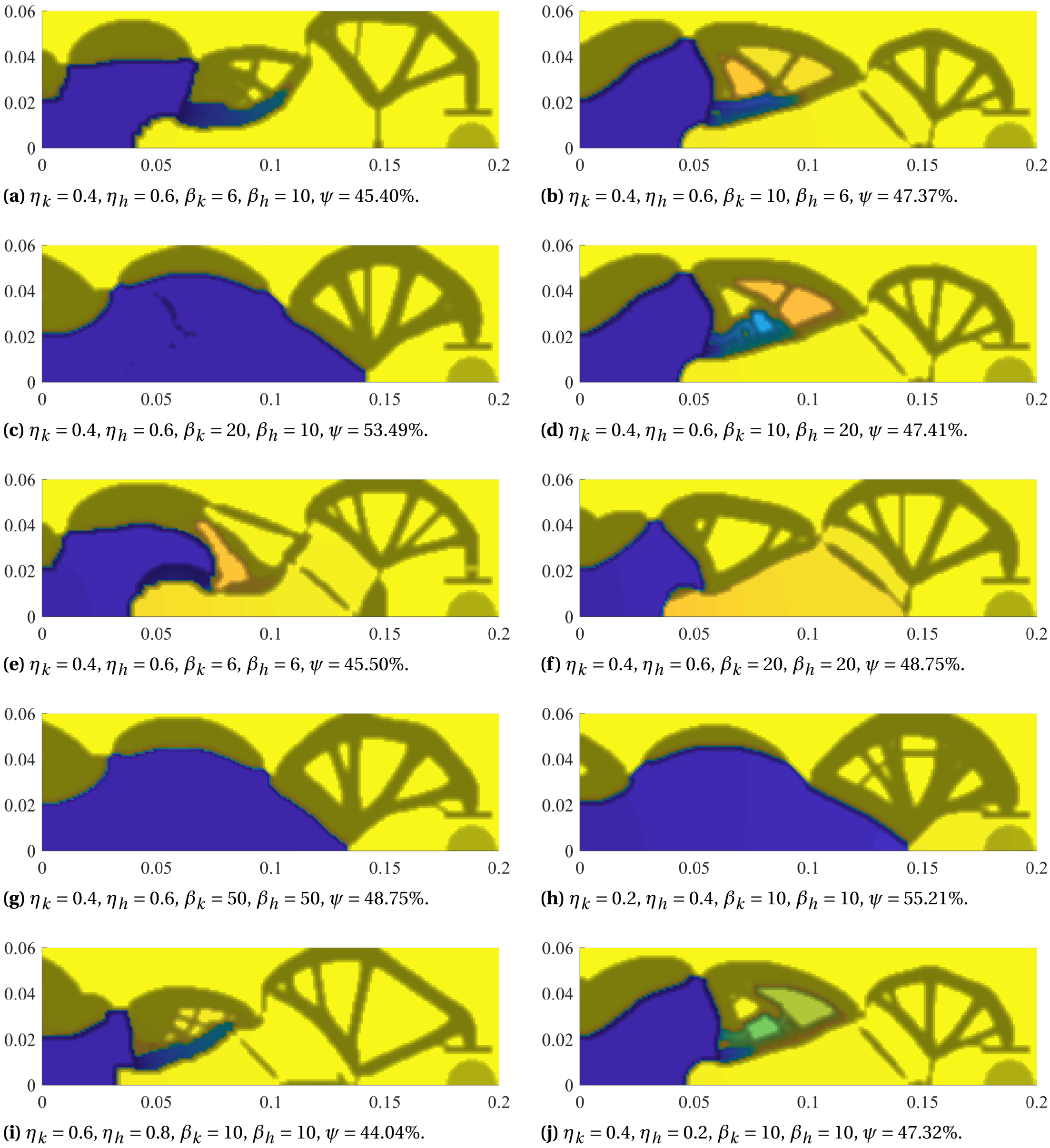


Figure 6.6: Convergence plot of all topology optimisations in Figure 6.7. The labels A-I refer to different sets of settings.



(a) $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = 6$, $\beta_h = 10$, $\psi = 45.40\%$.

(b) $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = 10$, $\beta_h = 6$, $\psi = 47.37\%$.

(c) $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = 20$, $\beta_h = 10$, $\psi = 53.49\%$.

(d) $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = 10$, $\beta_h = 20$, $\psi = 47.41\%$.

(e) $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = 6$, $\beta_h = 6$, $\psi = 45.50\%$.

(f) $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = 20$, $\beta_h = 20$, $\psi = 48.75\%$.

(g) $\eta_k = 0.4$, $\eta_h = 0.6$, $\beta_k = 50$, $\beta_h = 50$, $\psi = 48.75\%$.

(h) $\eta_k = 0.2$, $\eta_h = 0.4$, $\beta_k = 10$, $\beta_h = 10$, $\psi = 55.21\%$.

(i) $\eta_k = 0.6$, $\eta_h = 0.8$, $\beta_k = 10$, $\beta_h = 10$, $\psi = 44.04\%$.

(j) $\eta_k = 0.4$, $\eta_h = 0.2$, $\beta_k = 10$, $\beta_h = 10$, $\psi = 47.32\%$.

Figure 6.7: Ten designs resulting from different settings of the Darcy method. The objective solved for is the maximisation of energy efficiency (ψ).

The outflow constraint introduced in Section 5.3 has a clear effect on the results. Some of the designs still show the blockade that appears in the beginning which effectively forces a poor initial design to appear before the actual optimisation of the objective is done. In these examples it would be very beneficial to allow a higher outflow: $Q^*_{\text{out}} \approx 40$, at the beginning as explained in the end of Section 5.3.

6.3.3. Parameter sweep: output spring stiffness

The output spring stiffness is added via the objective function. The parameter sweep is shown in Figure 6.9 where the EE objective is used that aims to maximise ψ.

The energy efficiency objective aims to minimise mechanical losses, this can be clearly seen in (a) where the boundary is formed by a very thin balloon like membrane that literally inflates. As the spring stiffness increases the bulk of material (between $2 < x < 6$ cm) distributes more evenly to have more input compliance transferred to strain energy in the spring. High spring stiffness values cause the inner boundary to consist of segmented parts connected through mechanical hinges. Because the output port is almost held steady, the algorithm minimises the input compliance in order to still maximise the energy efficiency. In other words, the EE objective can end up with any mechanical advantage $M(=\frac{F_{\text{out}}}{F_{\text{in}}})$, which might be undesired by the user because a very low M value is very impractical.

The results, using the OF objective and different output stiffness values, are shown in Figure 6.11. Note that Figure 6.11a shows less of a thin membrane than Figure 6.9a using the same spring stiffness.

In Figure 6.8 the convergence plot of the EE objective spring stiffness sweep is provided. It can be clearly seen that for two of the ten runs, i.e. B and C, the shape was not quite determined after 40 iterations. Closer inspection of the behaviour reveals that it mainly is the shaping of the thin balloon like membrane (at $x =$ 8 cm) that happens smoothly and thus takes some iterations to converge. Furthermore, Figure 6.8 shows satisfying behaviour and smooth convergence of all optimisations.

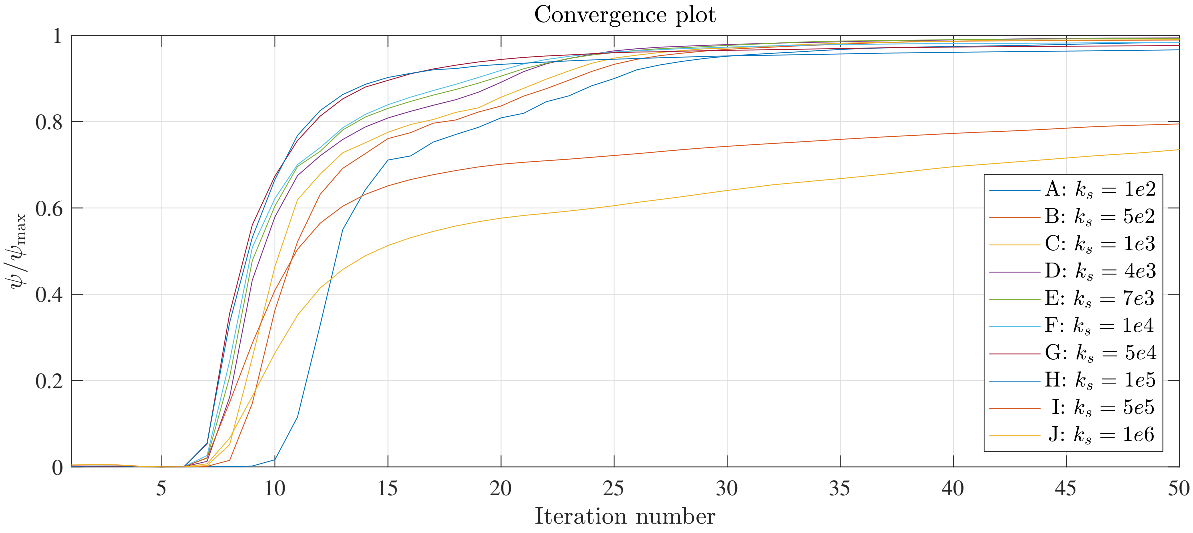
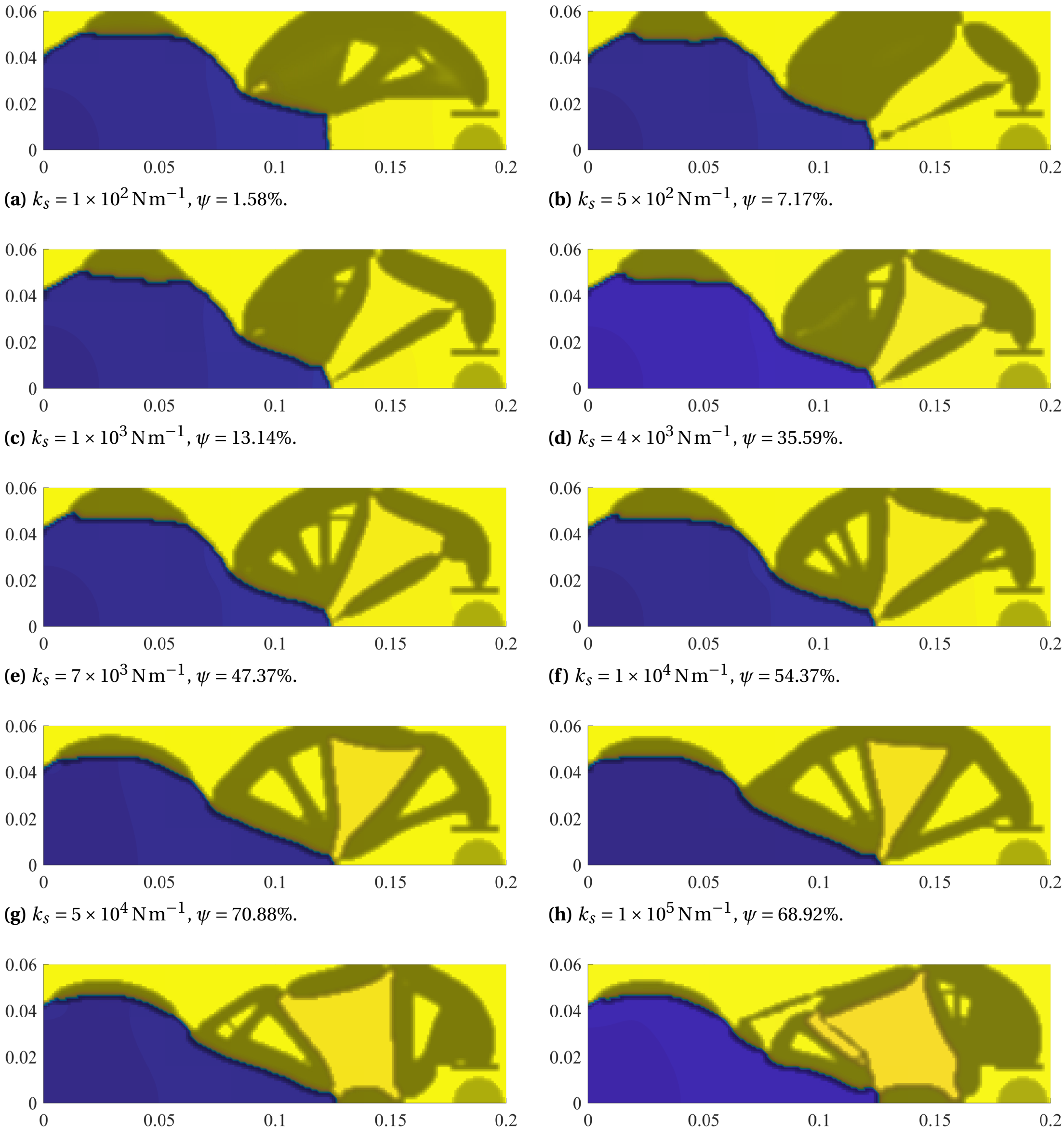


Figure 6.8: Convergence plot of all topology optimisations in Figure 6.9. They use the EE objective and an initial design with $\rho_i = 0.6$. Lines B and C continue linearly to 2000 iterations, the maximum allowed number of iterations.



(a) $k_s = 1 \times 10^2\,\mathrm{N\,m^{-1}}$, $\psi = 1.58\%$.

(b) $k_s = 5 \times 10^2\,\mathrm{N\,m^{-1}}$, $\psi = 7.17\%$.

(c) $k_s = 1 \times 10^3\,\mathrm{N\,m^{-1}}$, $\psi = 13.14\%$.

(d) $k_s = 4 \times 10^3\,\mathrm{N\,m^{-1}}$, $\psi = 35.59\%$.

(e) $k_s = 7 \times 10^3\,\mathrm{N\,m^{-1}}$, $\psi = 47.37\%$.

(f) $k_s = 1 \times 10^4\,\mathrm{N\,m^{-1}}$, $\psi = 54.37\%$.

(g) $k_s = 5 \times 10^4\,\mathrm{N\,m^{-1}}$, $\psi = 70.88\%$.

(h) $k_s = 1 \times 10^5\,\mathrm{N\,m^{-1}}$, $\psi = 68.92\%$.

(i) $k_s = 5 \times 10^5\,\mathrm{N\,m^{-1}}$, $\psi = 59.75\%$.

(j) $k_s = 1 \times 10^6\,\mathrm{N\,m^{-1}}$, $\psi = 51.83\%$.

Figure 6.9: The results of the TO using different output spring stiffness values and an initial design with $\rho_i = 0.6$. The energy efficiency ψ is maximised, thus a higher value is a higher load transfer efficiency.

Figure 6.11 shows the optimised designs for the OF objective as opposed to the designs in Figure 6.9. In Figure 6.11a the output spring stiffness is very low, $k_s = 1 \times 10^2\,\mathrm{N\,m^{-1}}$, and the only way to gain output force is by maximising the output deflection. table 6.2 shows the data that is used, including the output deflection resulting from the division of spring stiffness and force. The deflection barely seems to be effected by the spring in the first three optimisations. Note however that this is a linear analysis and that deflections simulated here could overestimate the actual values[2]. A remarkable result in this parameter sweep is the optimisation for the spring stiffness of $5 \times 10^4\,\mathrm{N\,m^{-1}}$, shown in (i). The optimisation only took 241 iterations to converge, and the objective was at 99% of the final objective at iteration 33 and does not show a single bump in the convergence plot. The two adjacent spring stiffness settings also required less iterations.

Table 6.2: Results of the spring stiffness sweep using the OF objective.

k_s in $\mathrm{N\,m^{-1}}$	F_{out} in N	Δ_s in mm	# iter.
1×10^2	1.09	10.9	1500
5×10^2	5.27	10.5	1500
1×10^3	10.0	10.0	1500
4×10^3	33.0	8.25	1500
7×10^3	50.1	7.16	1500
1×10^4	64.5	6.45	1293
5×10^4	169.2	3.39	241
1×10^5	241.7	2.42	1136
5×10^5	488.1	0.98	1500
1×10^6	583.3	0.58	1500

Figure 6.11(a), (b) and (c) show a ‘fluid’ filled region between $3 < x < 10\,\mathrm{cm}$. This often appears when a local structure is required to be stiff in bending but does not have sufficient material to do so without creating a large cavity in the middle. If there would be no pressure, the local structure would be stiff in bending but weak against sideways compression. The final design solves this by leaking the pressure to the inside, allowing the two walled solution with a significant gap in between to be stiff in bending and preventing the sideways compression by equalizing the pressure between the inside and the outside of the cavity or chamber.

Throughout these solutions, one can notice the shift between two design types, characterised by the designs in (d) and (j). The first uses a membrane that is pushed away and pulls the ‘lever’ to crush the ball, and the other tries to maximise the use of the input load by making a stiffer structure to transfer the load.

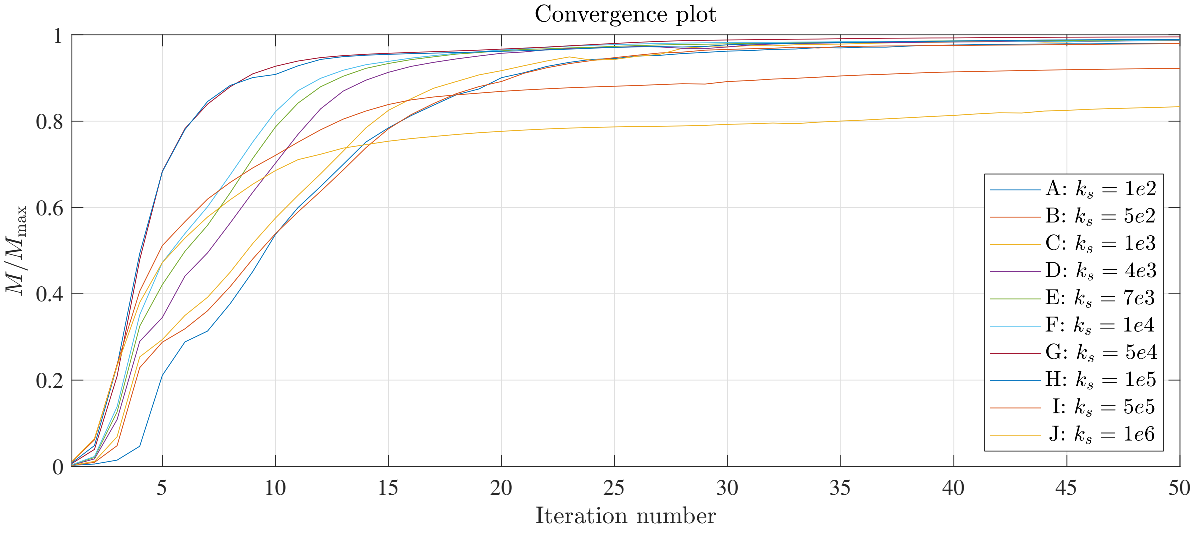
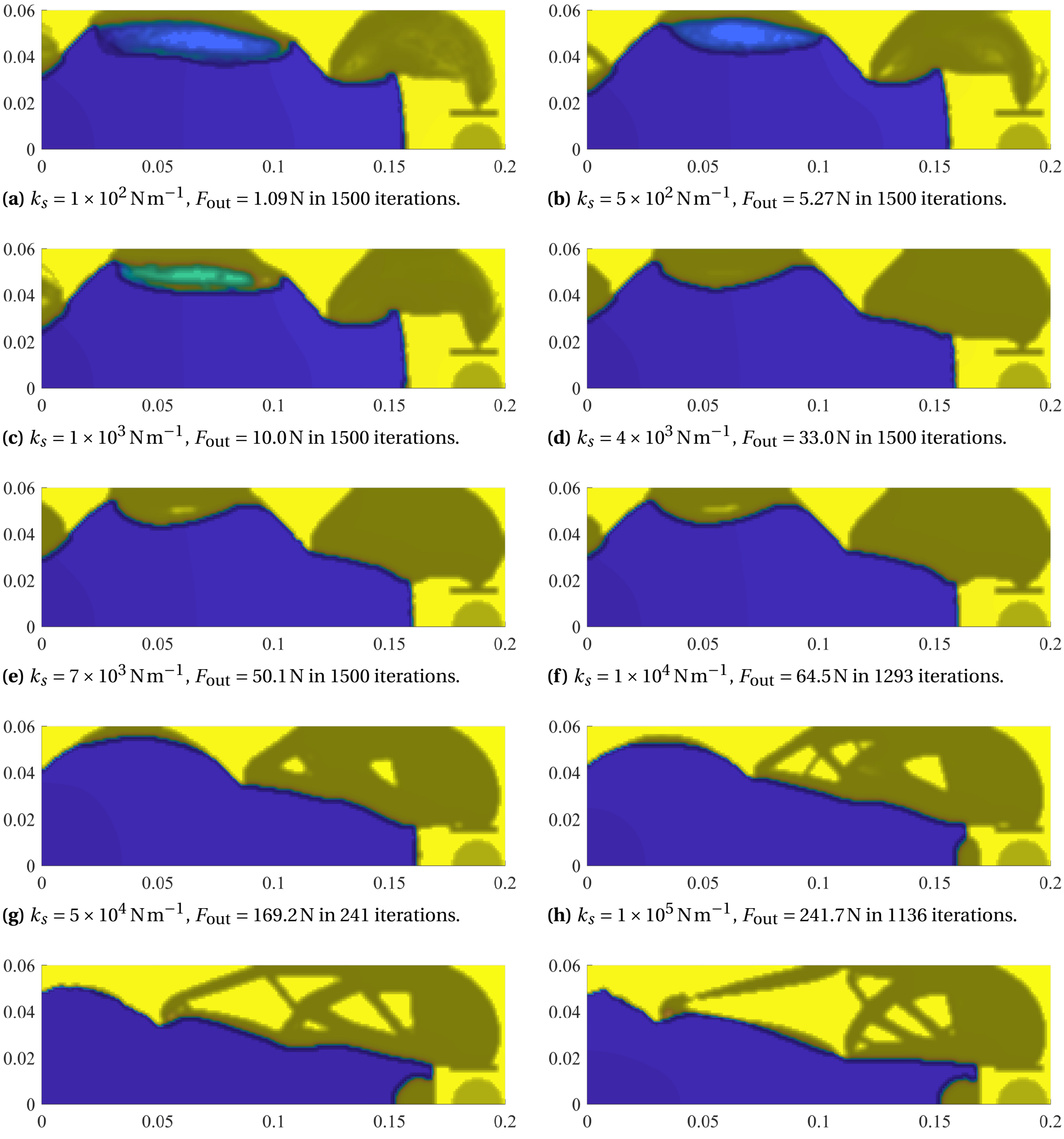


Figure 6.10: Convergence plot of all topology optimisations in Figure 6.11.

[2] The linear analysis provides a deformation field that is formed by scaling the deformation field that results from an infinite decimal load.



(a) $k_s = 1 \times 10^2\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 1.09\,\mathrm{N}$ in 1500 iterations.

(b) $k_s = 5 \times 10^2\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 5.27\,\mathrm{N}$ in 1500 iterations.

(c) $k_s = 1 \times 10^3\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 10.0\,\mathrm{N}$ in 1500 iterations.

(d) $k_s = 4 \times 10^3\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 33.0\,\mathrm{N}$ in 1500 iterations.

(e) $k_s = 7 \times 10^3\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 50.1\,\mathrm{N}$ in 1500 iterations.

(f) $k_s = 1 \times 10^4\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 64.5\,\mathrm{N}$ in 1293 iterations.

(g) $k_s = 5 \times 10^4\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 169.2\,\mathrm{N}$ in 241 iterations.

(h) $k_s = 1 \times 10^5\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 241.7\,\mathrm{N}$ in 1136 iterations.

(i) $k_s = 5 \times 10^5\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 488.1\,\mathrm{N}$ in 1500 iterations.

(j) $k_s = 1 \times 10^6\,\mathrm{N\,m^{-1}}$, $F_{\text{out}} = 583.3\,\mathrm{N}$ in 1500 iterations.

Figure 6.11: Topologies resulting from a parameter sweep over the output spring stiffness values, using $\eta_k = 0.2$, $\eta_h = 0.3$, with initial topology of $\rho_i = 0.6$ to aid the low stiffness output spring designs.

6.3.4. Parameter sweep: initial condition

The initial design (so the design the optimisation algorithm starts with) appears to have quite some effect on the outcome, Section 6.3.1 introduces a method to alter the initial design by altering ρ_i, this method is applied here. Figure 6.13 shows the results of various initial designs using both the EE and the MA objective, the left and right column respectively.

Lowering the η values provides a clearer boundary description in the earlier stage of optimisation but makes it difficult to 'sense' beyond these boundaries because the load sensitivity at $\rho_e = 1$ is almost zero for low η values. Another clear observation is that having an initial condition does not do much good, the highest objective values can be found at $\rho_i = 0.4$ and 0.6. There is a chance that other, more creative, initial conditions would do a better job.

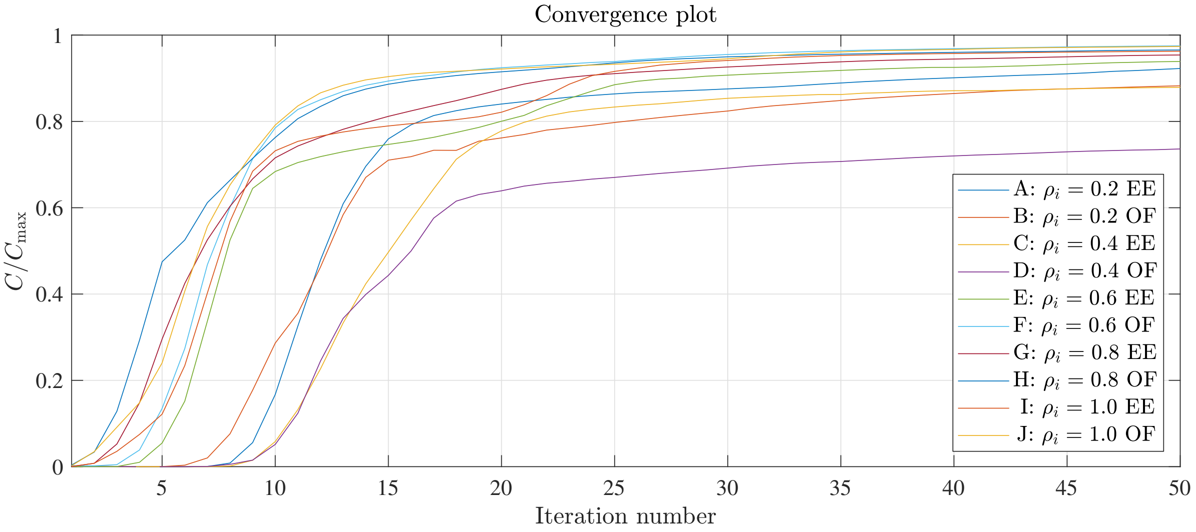
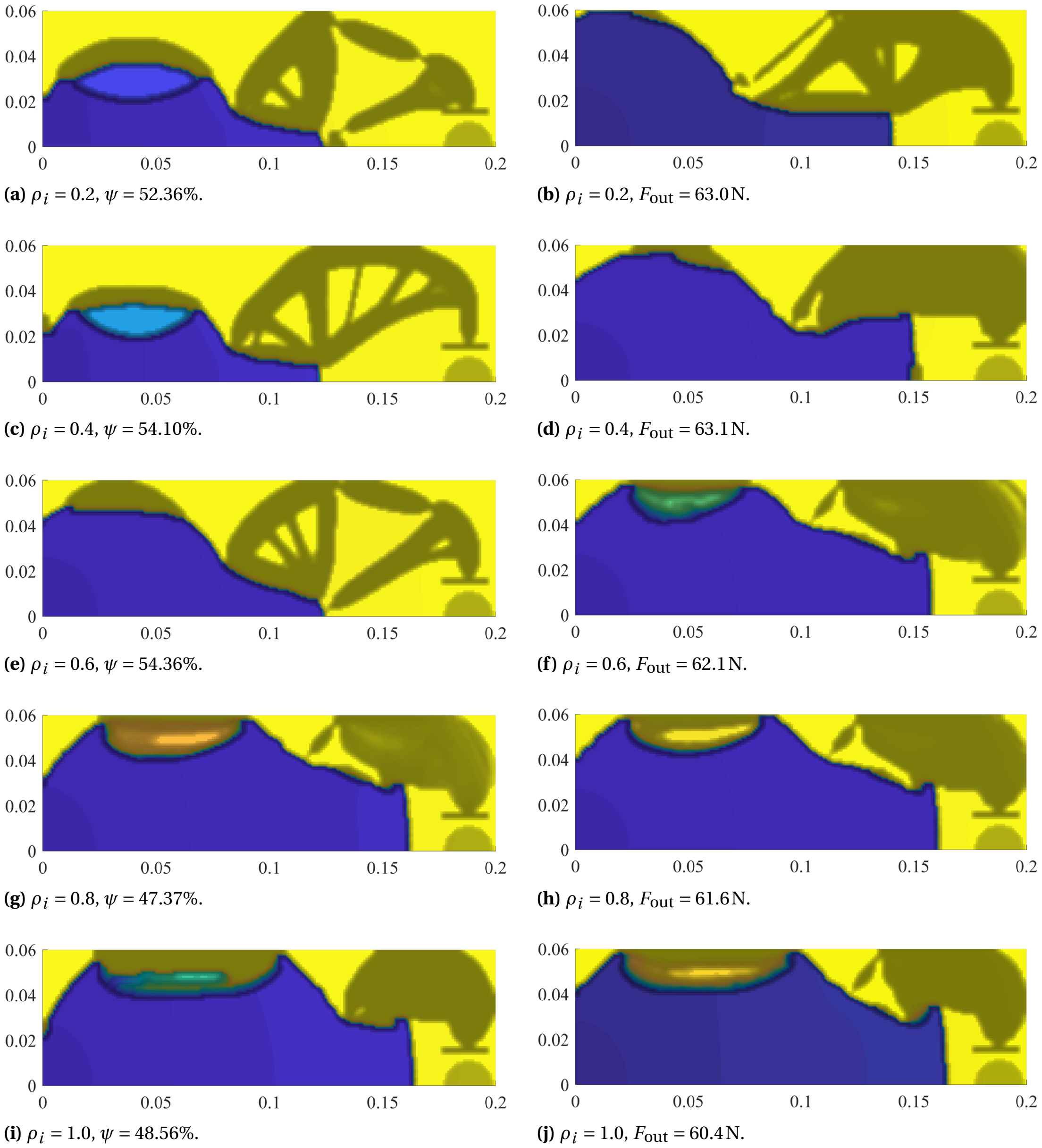


Figure 6.12: Convergence plot of all topology optimisations in Figure 6.13.



(a) $\rho_i = 0.2$, $\psi = 52.36\%$.

(b) $\rho_i = 0.2$, $F_{\text{out}} = 63.0\,\text{N}$.

(c) $\rho_i = 0.4$, $\psi = 54.10\%$.

(d) $\rho_i = 0.4$, $F_{\text{out}} = 63.1\,\text{N}$.

(e) $\rho_i = 0.6$, $\psi = 54.36\%$.

(f) $\rho_i = 0.6$, $F_{\text{out}} = 62.1\,\text{N}$.

(g) $\rho_i = 0.8$, $\psi = 47.37\%$.

(h) $\rho_i = 0.8$, $F_{\text{out}} = 61.6\,\text{N}$.

(i) $\rho_i = 1.0$, $\psi = 48.56\%$.

(j) $\rho_i = 1.0$, $F_{\text{out}} = 60.4\,\text{N}$.

Figure 6.13: Topologies resulting from different initial designs, i.e. different ρ_i and using $\eta_k = 0.4$, $\eta_h = 0.6$ and $k_s = 1 \times 10^4\,\text{N}\,\text{m}^{-1}$. At the left hand the solutions of the EE objective are shown and at the right hand the OF objective.

6.3.5. Parameter sweep: volume fraction

Different allowable volume fractions ($V_{\max}$) can give very different results. Figure 6.15 shows the results of this parameter sweep. It seems in Figure 6.15a that a low volume fraction would allow fluid to flow through more easily, but comparing to other designs, obtained using the Darcy approach, this largely depends on the used K and H relations. When a soft robotic, single layer, inflatable kind of design is desired, allowing a low volume fraction and high input compliance helps. Typically, the design only needs 1 or 2 element thick boundaries to fully dissipate the pressure field. By tuning the η_k and η_h parameters properly (and maybe the β parameters), each of these designs can be tuned to behave like Figure 6.15c, with the highest ψ value.

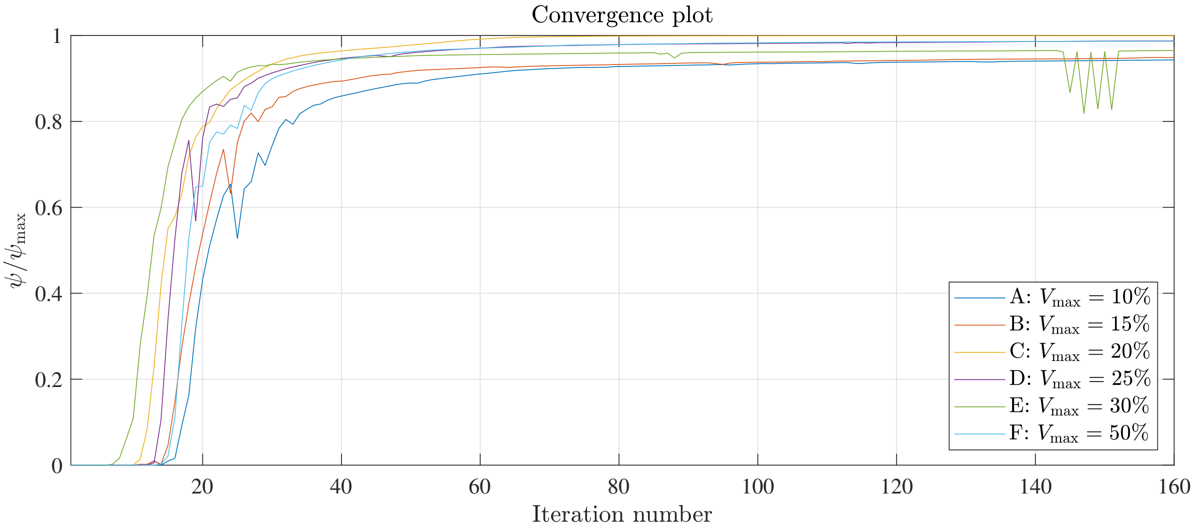
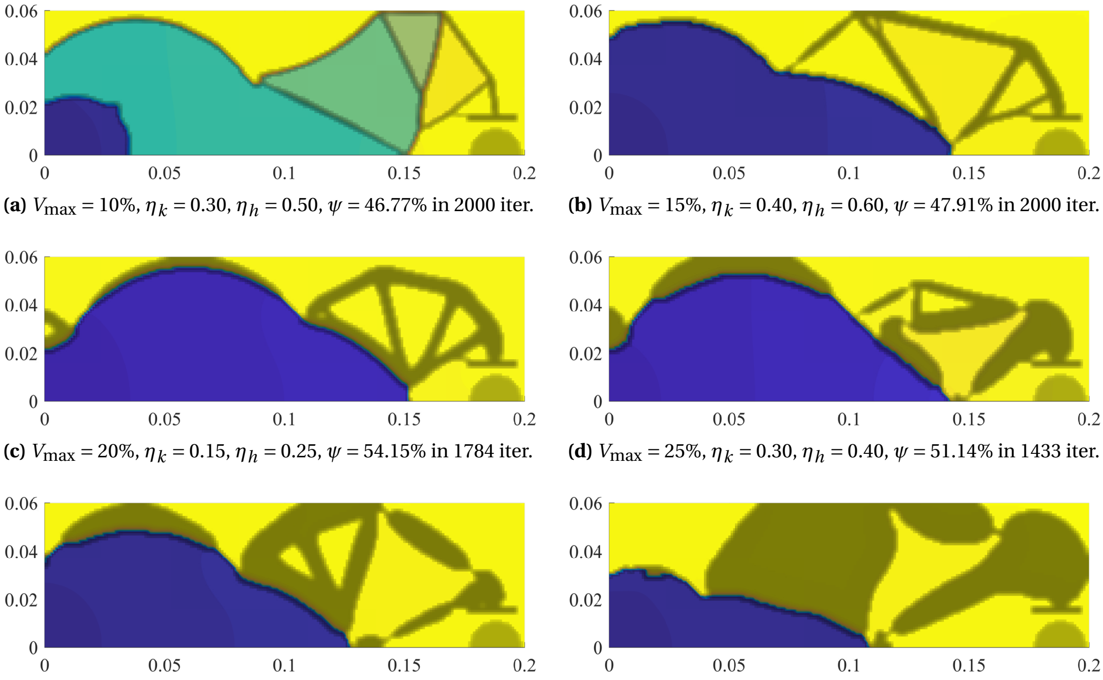


Figure 6.14: Convergence plot of all topology optimisations in Figure 6.15.



(a) $V_{\max} = 10\%$, $\eta_k = 0.30$, $\eta_h = 0.50$, $\psi = 46.77\%$ in 2000 iter.

(b) $V_{\max} = 15\%$, $\eta_k = 0.40$, $\eta_h = 0.60$, $\psi = 47.91\%$ in 2000 iter.

(c) $V_{\max} = 20\%$, $\eta_k = 0.15$, $\eta_h = 0.25$, $\psi = 54.15\%$ in 1784 iter.

(d) $V_{\max} = 25\%$, $\eta_k = 0.30$, $\eta_h = 0.40$, $\psi = 51.14\%$ in 1433 iter.

(e) $V_{\max} = 30\%$, $\eta_k = 0.30$, $\eta_h = 0.40$, $\psi = 53.43\%$ in 853 iter.

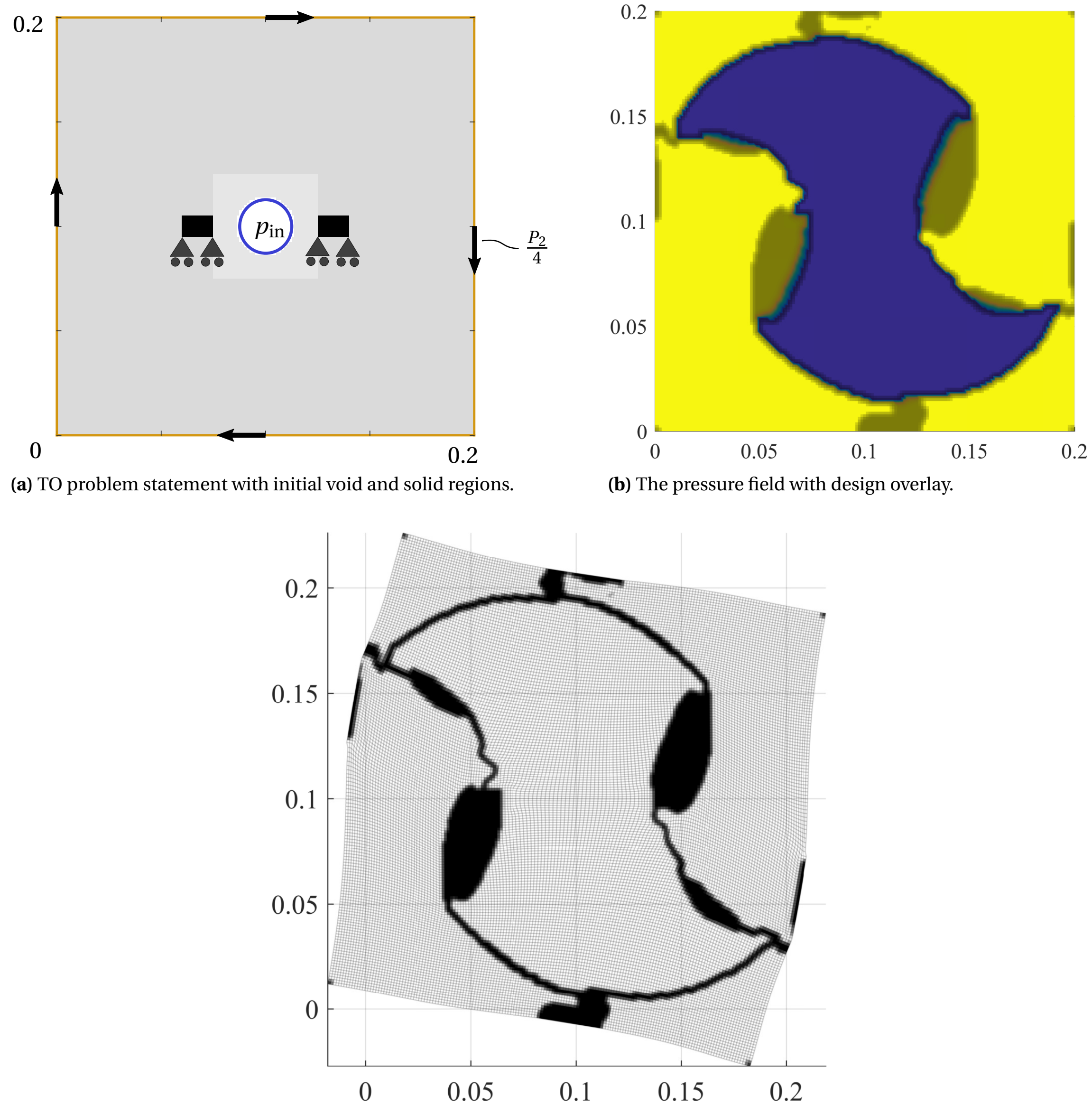
(f) $V_{\max} = 50\%$, $\eta_k = 0.30$, $\eta_h = 0.40$, $\psi = 51.49\%$ in 1876 iter.

Figure 6.15: Topologies resulting from different volume fractions, $V_{\max}$. The η values are also adjusted accordingly.

The convergence plot of the optimisation procedure is shown in Figure 6.14. Note that three of the six optimisations show the same bump in the beginning around iteration 20, i.e. A, B and D. This has to do with the decision process whether or not to fill the next chamber and whether or not to preserve the structural internal boundary.

6.4. Alternative design problems and single walled solutions

In order to show the versatility of the Darcy method, this section offers some alternative solutions. The cover image of this work is the first example of an alternative problem. In this example, a pressure source is used in the centre of the bottom half of the page and the objective was to push the frames in place as the arrows indicate.



(a) TO problem statement with initial void and solid regions.

(b) The pressure field with design overlay.

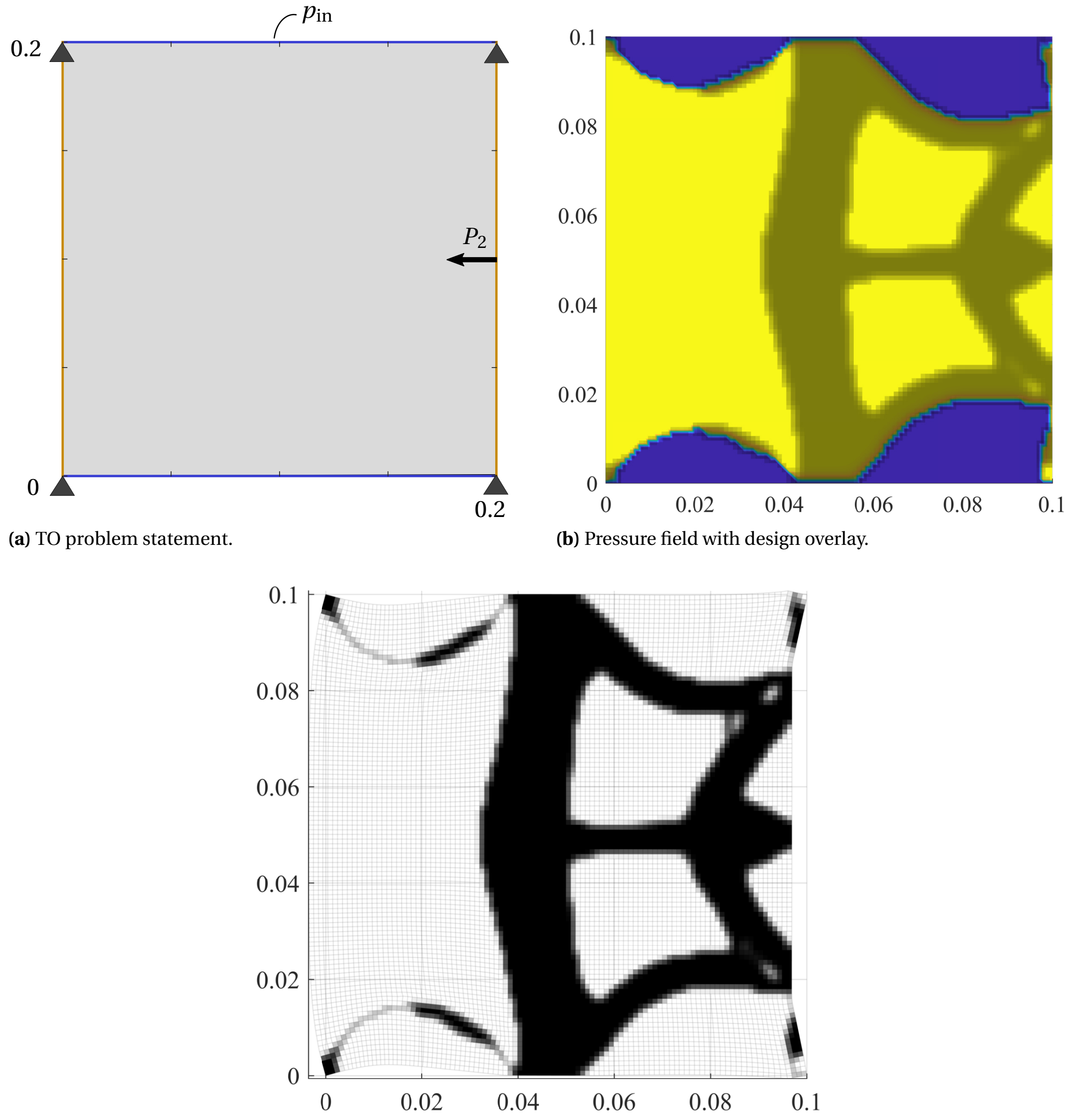
(c) Displacement field of (b)

Figure 6.16: A rotational motion is requested by four clockwise oriented loads at the middle of each side. The outflow constraint is imposed slowly, halving Q^*_{out} every 50 iterations from $40\,\text{m}^3\,\text{s}^{-1}$ to a minimum of $1\,\text{m}^3\,\text{s}^{-1}$. $\boldsymbol{F}_2$ contains the four loads of length $\frac{P_2}{4}$. The EE objective value is $\psi = 62.8\%$.

In Figure 6.16 a rotational output motion is provoked by the four clockwise oriented loads displayed in (a).

The result is symmetrical and it can be recognised that at $\boldsymbol{x} = [x,\, y]^\top = [0.06,\, 0.12]^\top$ and $\boldsymbol{x} = [0.14,\, 0.07]^\top$ two wrinkled walls occur, this effectively lowers the resistance against elongation of the wall, which is desired here. The desired motions at the middle of each side are realised by a pushing action on the top and bottom boundaries and a pulling action on the left and right ones. The two black blocks, visible in (a), are supported by roller constraints. In (b) and (c) these blocks are part of the two lumps of material close to the centre. Note that, because of the roller constraints, these two blocks cannot rotate and the optimiser benefits from that by having this big lump pull on the arc shapes at the top and bottom.

The deformation field is visualised by plotting the design on transformed coordinates, the new coordinates are: $\boldsymbol{x}_d = \boldsymbol{x}_g + \boldsymbol{u}_u$ and $\boldsymbol{y}_d = \boldsymbol{y}_g + \boldsymbol{u}_v$ where $\boldsymbol{u}_u$ and $\boldsymbol{u}_v$ refer to the x- and y-direction components of the total displacement vector $\boldsymbol{u}$. The variables $\boldsymbol{x}_g$ and $\boldsymbol{y}_g$ are the collection of coordinates and $\boldsymbol{x}_d$ and $\boldsymbol{y}_d$ are the displaced coordinates of all nodes in the global domain.



(a) TO problem statement.

(b) Pressure field with design overlay.

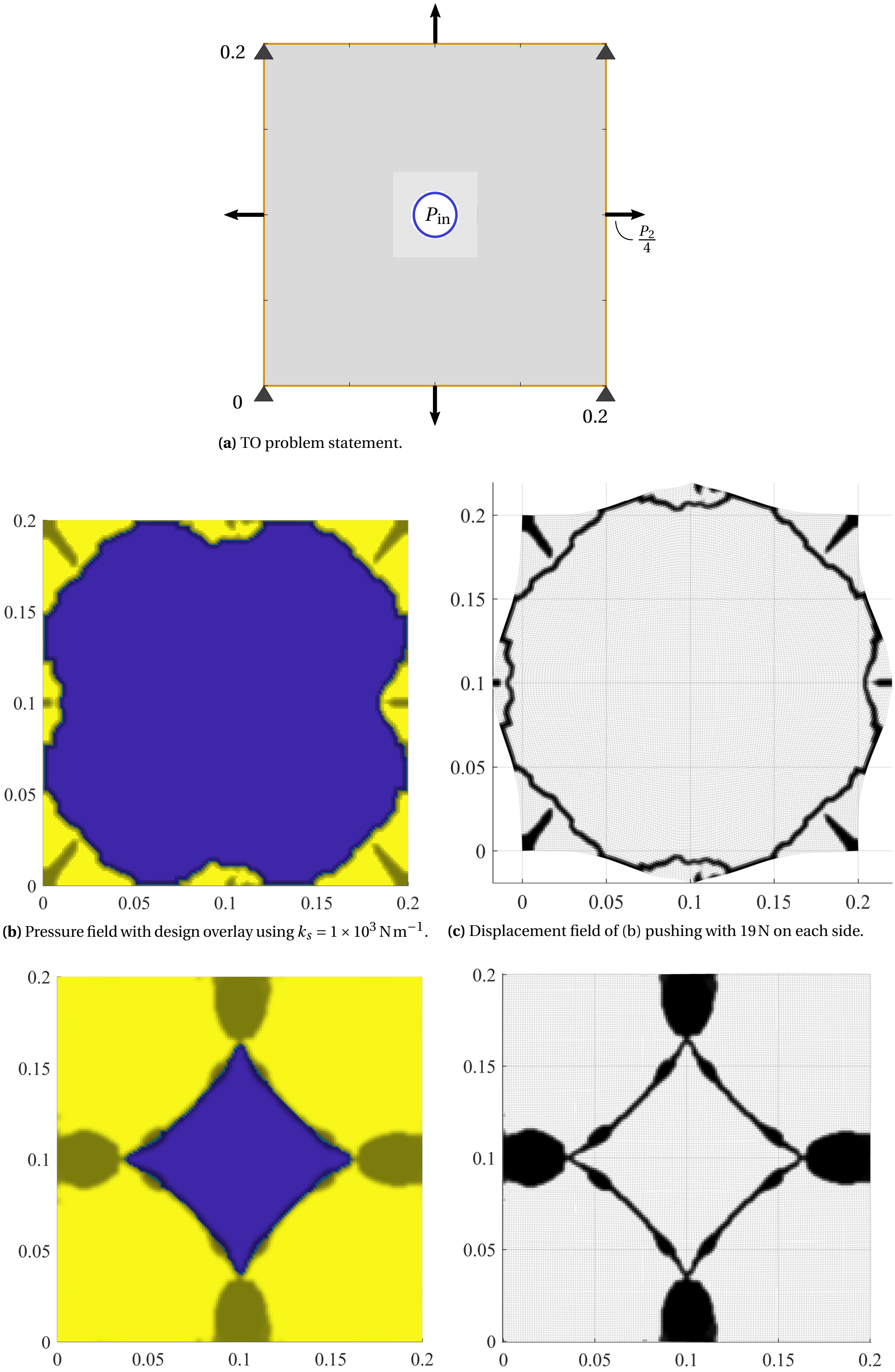
(c) Displacement field of (b)

Figure 6.17: A motion to the left is sought as a response to a pressure load on the top and bottom. The spring stiffness is $k_s = 1 \times 10^3\,\mathrm{N\,m^{-1}}$ and the outflow constraint is imposed where $Q^*_{\text{out}} = 1$. $\boldsymbol{F}_2$ contains the output load: P_2. The final EE objective value is $\psi = 66.62\%$.

In Figure 6.17 a TO uses a top and bottom pressure to compress a spring of stiffness $k_s = 1 \times 10^3\,\mathrm{N\,m^{-1}}$ positioned on the arrow with length P_2 which is the only entry in $\boldsymbol{F}_2$. The environmental pressure, p_{out}, is imposed on the left and right boundaries. The converged shape uses the right part to block the pressure and the left part to deform a membrane that pulls the structure to the left. The design reaches a remarkable efficiency of $\psi = 66.62\%$.

Two separate optimisations using $k_s = 1 \times 10^3\,\text{N}\,\text{m}^{-1}$ and $k_s = 1 \times 10^6\,\text{N}\,\text{m}^{-1}$ are conducted in Figure 6.18. The problem statement is shown in (a). In (c), the deformation plot, there is at the output ports, barely a connection visible that transfers the force. There are two reasons for this, first the output stiffness is relatively low and second the bulk modulus of an element still aids in pushing the mesh outwards at these points.

Figure 6.19 shows two optimisations of a contracting mechanism problem, they both use $k_s = 1 \times 10^3\,\text{N}\,\text{m}^{-1}$. The first one, displayed in (b) and (c), optimised the EE objective to a value of $\psi = 23.40\%$. The second one, displayed in (d) and (e), optimised the OF objective to a value of $F_{\text{out}} = 35.6\,\text{N}$ with the help of an added drain or sink in the centre of the square domain to p_{out}. The first optimisation run closed the top and bottom lid in an early stage and had difficulty achieving any of the objective at first, a bit later it developed the buckling knees on the top and bottom to satisfy the objective. The moment that the mechanism, through which motion occurs, is developed, it is unlikely to change. This is the reason why it is easy to find local optima and near impossible to find the global one. The second optimisation has got some artefacts which do not seem to add much to the optimisation. It is interesting to note that the MMA optimiser sometimes chose to completely remove the artefacts and put them back again, removing them had a slightly negative impact on the objective.



(a) TO problem statement.

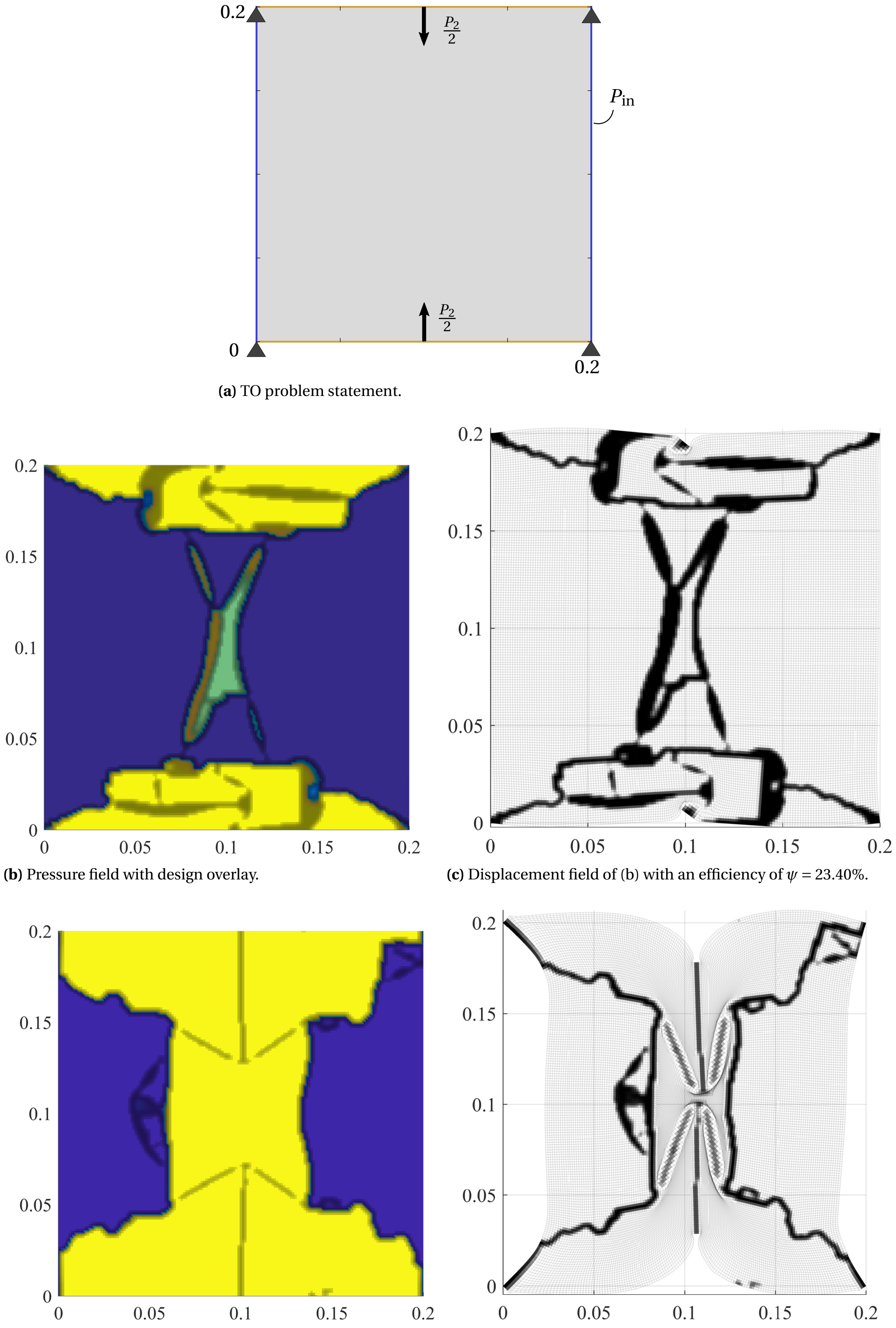
(b) Pressure field with design overlay using $k_s = 1 \times 10^3\,\mathrm{N\,m^{-1}}$.

(c) Displacement field of (b) pushing with 19 N on each side.

(d) Pressure field with design overlay using $k_s = 1 \times 10^6\,\mathrm{N\,m^{-1}}$

(e) Displacement field of (d) pushing with 92 N on each side.

Figure 6.18: An expanding motion is requested by four outward loads at the middle of each side. The spring stiffness is indicated in the sub-captions and the outflow constraint is imposed slowly, halving Q^*_{out} every 50 iterations from $40\,\mathrm{m^3\,s^{-1}}$ to a minimum of $1\,\mathrm{m^3\,s^{-1}}$. $\boldsymbol{F}_2$ contains the four loads: $\frac{P_2}{4}$.



(a) TO problem statement.

(b) Pressure field with design overlay.

(c) Displacement field of (b) with an efficiency of $\psi = 23.40\%$.

(d) Pressure field with design overlay.

(e) Displacement field of (d) pushing with 17.8 N on each side.

Figure 6.19: A contracting motion is requested by two inward loads at the middle of the top and bottom side. The spring stiffness is $k_s = 1 \times 10^3\,\mathrm{N\,m^{-1}}$ and the outflow constraint is imposed slowly, halving Q^*_{out} every 50 iterations from $40\,\mathrm{m^3\,s^{-1}}$ to a minimum of $1\,\mathrm{m^3\,s^{-1}}$. $\boldsymbol{F}_2$ contains the two loads: $\frac{P_2}{2}$. The difference between (b) and (d) is the use of the EE and OF objective respectively and, in (d), the node in the centre of the square is prescribed to be p_{out}. This is analogous to adding a sink in the middle.

7 Discussion

In the previous chapter the reader is familiarised with the behaviour of the Darcy method using different settings. This chapter will try to put the results and behaviour of the pressure modelling method in perspective and discusses several trade-off's.

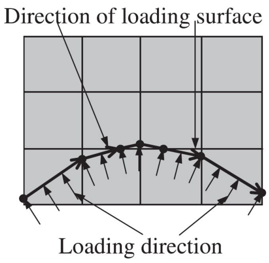


Figure 7.1: Iso-density boundary search method, [10].

So, what does this method bring to the table? The original aim was to find a way to explicitly define the pressure field such that it is differentiable and contained an adjustable amount of wall penetration, meaning that the sharp pressure interface could be smeared out if the user desires more load sensitivity. This work offers a new perspective by using Darcy's law governing the flow through porous media. Henceforth, the proposed method is called here(x,y,z) the Darcy's method. Darcy's method allows for these features and thus for better, more widely defined, load sensitivities that aid the behaviour of the optimisation in a satisfactory way. At this point the Darcy method is fully capable to solve pressure actuated TO problems. It can also be used as a research tool to investigate the behaviour of the load sensitivities of design dependent loads.

One of the methods proposed in literature used nodal densities to find the iso-density pressure interface, as shown in Figure 7.1. This iso-density method also has load sensitivities [10, 14, 19]. In this case, however, the sensitivities are only present on, and directly adjacent to, the elements where the pressure boundary Γ_{p_b} is positioned. There is no two phase method in literature that can sense beyond a boundary, this is where the Darcy method gains ground. When using the Darcy method, it might be less predictable (at this point in research) whether the solutions are clean and desirable. It will, however, increase the chance of finding the optimal position to apply the load and allow the optimiser to remove, build and shift the boundary as it pleases. This is unlike most other methods proposed in scientific literature, the only method that inherits load sensitivities and the option to remove and build boundaries is the three phase, (fluid, void, and solid) TO proposed by Sigmund **Sigmund and Clausen, 2007**.

This work relates both the flow and drainage coefficients, K and H, to $\boldsymbol{\rho}$ through a smooth Heaviside function. Other relations have been attempted in an early stage of development but the smooth Heaviside was the most stable at that point. It is very interesting to experiment with different relations to expose the optimisation to different load sensitivities. Different relations for $K(\boldsymbol{\rho})$ and $H(\boldsymbol{\rho})$ can be used to stimulate smaller or larger surface areas Γ_{p_b} exposed to pressure or to achieve better optima. In 2011, B.S. Lazarov and O. Sigmund said that 'Topology optimisation as a design process has matured over the years' [18]. This paragraph shows that this cannot be said about the methods to work with design dependent pressure loads. More research is required to find more about the local and global optima and how load sensitivities can be tuned to find them.

The designs shown in Section 6.4 are all thin and single walled. It thus seems that the Darcy method has a natural tendency to create soft robotic like designs. Examples in literature of pressure actuated compliant mechanisms show inflating geometries with trusses attached that finally perform the task as shown in Figure 7.2, where the three phase BIM is used. In the Darcy method, the combination of a pressure field formulation and the outflow constraint increase the chance of finding single walled, largely fluid filled designs.

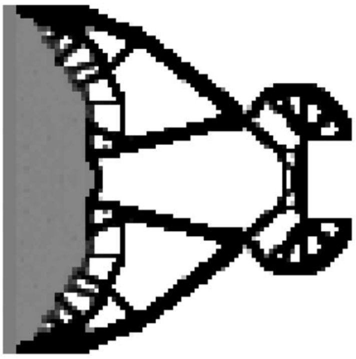


Figure 7.2: A pressure actuated compliant mechanism using same design problem as in the previous sections, with different domain size. Image from [23].

In a compliant mechanism design, the final result often seems to be determined in the first 30-50 iterations. This is intuitive because a TO narrows down on a 'working principle' (WP), i.e. in Figure 7.2 the WP is inflating a balloon like structure and inverse the outward motion to a clamping motion using trusses and hinges. When a WP is found by the optimiser it is locked-on to a local optima, the design rarely finds a different WP (or a local optima). In general, the TO still needs a few hundred iterations to actually converge, it makes the solid parts stiffer and the hinges thinner and longer. The Darcy method is sometimes able to remove a boundary if leaking the pressure to the next boundary is beneficiary. This way it actually shifts the focus on one WP and converges to another, which is in this case desirable.

In some problems, to which the Darcy method in combination with the EE objective has been exposed behaved undesirably. The EE objective tries to maximise ψ, the ratio between output and input energy. Therefore, if it is unable to find a path to satisfy the objective a little bit, it starts limiting the input compliance. If the boundary where the pressure is applied is also fixed in the degree of freedom perpendicular to the boundary, the optimiser is likely to generate material against this boundary. This virtually causes E_{in} to go to zero and is that way still able to raise ψ a bit to some sort of local optimum.

Panganiban [23] states that 'An alternative approach without introducing a pressure as an additional field variable can be formulated with the use of nonconforming finite elements', this being the three phase method of Sigmund [26]. What they do is doubling the design variables instead of adding one pressure DOF per node which is independent of the displacement DOF's. The flow matrix that is solved for the pressure field is of the size $[n \times n]$, while the stiffness matrix is of the size $[nd \times nd]$. A common rule of thumb is that solution time scales quadratic with size[8]. The solution of a pressure field takes approximately d^2 times less time compared to the displacement field calculation, which, in the cases presented in this work, only took 1-2 seconds. The optimisation step however, using the MMA optimiser, takes longer as the design converges. This generally takes 0.5-8 seconds per iteration, it takes longer when the design changes become smaller. Therefore it seems that solving the pressure field even be a computationally cheaper way to go.

One could argue whether the smeared out pressure boundary is similar to the sharp interface used in other methods. This work does not provide a comparative study to define the difference. It is, however, common practice to translate the line load $\boldsymbol{F}_p(s)$ to a consistent force vector $\boldsymbol{F}$ by performing an integration over this line:

$$\boldsymbol{F}^e = \int_{\Gamma_{p_b}} \mathbf{N}^\top \boldsymbol{F}_p(s)\, dA, \tag{7.1}$$

where s is a location on the line and $\mathbf{N}$ is the matrix of shape functions. The position of the line, in this case, depends on all surrounding element densities and shifts with every change in the density. There is no method in literature that can make the pressure interface Γ_{p_b} sharper than this. The Darcy approach can be configured to have the consistent forces in a smeared out region to have further carrying load sensitivities but it can also be configured or converge to configurations such that the pressure drops over a single element, which is about as sharp as the FEM allows an interface to be translated. This, together with the theory of Saint-Venant (that the resultant of a group of locally acting forces causes the same global deformation as the local forces would), allows the use of smeared out consistent forces.

When using the smooth Heaviside relations for K and H, the choice of η_k, η_h, β_k, and β_h determines the reaction of the pressure field to the material. If the last two are considered constant: $\beta_k = 10$ and $\beta_h = 10$, the response of the pressure field to the material can be tuned by shifting η_k and η_h. In other words, at what value of ρ_e does the flow coefficient K decrease (and the resistance of flow in material increase)? Note that this is analogous to the density threshold, except that it is not a sharp 'threshold' line (like a normal Heaviside function). **Low values** of η_k and η_h will allow an early and strong definition of the boundary with little chance of leakage, but limit the sensitivity beyond that boundary. When using **higher values** of η_k and η_h, the chance of leakage of the inner boundaries increases but this also means that the optimiser can choose to get rid of the inner boundaries as it has more sensitivity beyond the first encountered boundary. Leakage through the outer boundary is in any case suppressed by the outflow constraint.

One might wonder if the flow coefficient is actually required when the drainage coefficient does most of the work? If the flow coefficient is kept unaltered or independent of density, such that $k_{\text{mat}} = k_{\text{void}}$, the only design dependent parameter is the drainage. When a TO is attempted this way, the design is seen to create branches to 'absorb' the pressure in order to use it. This is very similar to the branches that occur in the TO of a thermal heat exchanger where the boundary is being maximised by the optimiser to guide away as much heat as possible. Concluding, the Darcy method requires both these parameters in order to function properly. This might change is other objectives are used that would not benefit from the creation of branches.

8 Conclusion and recommendations

Research questions
The introduction, Chapter 1, states the goal of the thesis as the development of a method to deal with pressure loads in the mathematical framework of topology optimisation using SIMP. The method should aid in calculating the optimal design of pressure actuated mechanisms. To this end, a few research questions were posed, repeated here: **1)** Can a differentiable and globally defined pressure field formulation be used to perform a Topology Optimisation of a pressure actuated compliant mechanism? **2)**Does this formulation improve the chance of finding a better optima? And **3)** Can the parameters in this formulation be tuned such that soft robotic like shapes can be extracted?

Conclusion
1) An explicitly defined, design dependent pressure field, that is solved using a FEM formulation, can be used to optimise pressure actuated compliant mechanisms. This follows clearly from the designs shown in Chapter 6 and Section 6.4. The boundary identification problem is dealt with in two steps: finding a design dependent pressure field and apply the forces resulting from a pressure drop to the structural problem. This two-step approach is not yet used in scientific literature. Doing this, however, offers some large advantages: it inheres generality in its application (can be expanded to 3D or other problems or physics), it supports different relations between the pressure field and the design (i.e. by altering $K(\boldsymbol{\rho})$ and $H(\boldsymbol{\rho})$), and it can be extended to locally have different pressures. It might even be possible to optimise a design that attempts to use an input pressure and convert it to a higher output pressure like a 'pressure booster'.

2) The performance of the structural optimisation in combination with the Darcy method can be seen to increase if the parameters are set, such that differentiability and smeared out pressure interfaces are promoted, i.e. by choosing $\beta_k = \beta_h \approx 10$. It can thereby be concluded that having control over $K(\boldsymbol{\rho})$ and $H(\boldsymbol{\rho})$ allows for finding better optima. In general, a TO is ran multiple times with different initial conditions to find the best (local) optimum. The Darcy method allows the user to make significant changes by simply altering the method parameters. It provides the TO user with the tools to alter the spread of the load sensitivities, which can be used, instead of altering the initial condition, in the search for the best optimum.

3) The field of soft robotics often uses tentacle like structures of a single material, they often comprise of a single enclosing wall (they are single walled) and deflect by inflation. In Section 6.4 the reader can find several soft robotic like designs which are optimised using the Darcy method and varying parameters. At this point in research it is not clear exactly what parameters are required to promote single walled designs. The method generates solutions with large pressurised areas as well as smaller ones.

Recommendations
At this point, only one relation for $K(\boldsymbol{\rho})$ and $H(\boldsymbol{\rho})$ is tested more thoroughly. The method is very tunable like changing the penalisation p_E in the SIMP method, although, for p_E it is clear that $p_\mathrm{E} = 3$ is the magic number while in the Darcy method different parameters offer different solutions. Tuning the parameters can even control whether flow is allowed to pass through an almost solid wall without a pressure drop. Other pressure field formulations (than the Darcy method) can be tried as long as they simulate a realistic pressure field that drops entirely over the outer boundary of the topology.

The output pressure p_{out} is prescribed at the outer boundaries Γ_{p_0} which causes for a naturally decreasing pressure field in the early iterations where the topology still forms a grey field. The closing of the boundary is now guaranteed by the outflow constraint, but this seems to work strongly against the convexity of the problem. When some inner boundary is closed it cannot be opened as easily as one would desire because the volume flow that flows through the inner boundary is likely not sufficiently blocked by the outer boundary, as such, the constraint will not be satisfied any more. Solvers tend to optimise tangent to the constraint 'line' and will, as such, not open a boundary (cross the constraint line) if that is not immediately allowed by the constraint. To circumvent this harsh constraint, it can be advised to look into different objective functions that would benefit from having closed boundaries. Another option is to soften the constraint by inserting it into the objective function, by adding it with a weight factor or as a multiplication with a certain adjustable penalisation, automatically stimulating closed boundaries.

Now that a method has been formulated that provides the user with control over the continuity of the pressure field, it is interesting to investigate the effect of altering the pressure field response and thereby the load sensitivities. As mentioned earlier, this method lends itself perfectly for a wide study to load sensitivities and maybe a relation $p(\boldsymbol{\rho})$ can be concocted that causes the optimisation problem to be (more) convex, allowing for better, and less bound decision making.

A

Appendix: Topology Optimization applied

Topology optimization (TO) is a field in solid mechanics that deals with optimal material distributions for given physical problems and associated objectives. The objectives are related to the physics which are calculated by a FEA, and the physics depend on some chosen design variable (e.g. a density field). A topology optimization typically consists of the following elements: A physical problem is formulated in a FEA using the current design, the design variables differ per element, then the solution of the FEA provides the physical parameters (over those elements), these are required to evaluate the objective and calculate the direction of optimal design and then a small change in the material distribution is made in the right direction, this is repeated until the optimal design is found. The criteria for optimal design is usually the amount of change in material distribution, if it falls beneath a threshold the optimization loop will stop [4].

A.1. Objective and material models

There are two material descriptions which are frequently used. One is the level-set-method which uses a Level Set Function (LSF) ϕ defined over the entire domain. Material is now defined as the domain where $\phi > 0$ and the boundary of which is the contour line where $\phi = 0$. The second description is the Solid Isotropic Material with Penalization (SIMP). Each element e is assigned a density $\boldsymbol{\rho}$ (ranging from zero to one) that determines its Young's modulus E_e by

$$E^e(\rho_e) = E_{\min} + \rho_e^{p_{\mathrm{E}}}(E - E_{\min}), \qquad \rho_e \in (0,1). \tag{A.1}$$

$E_{\min}$ is required to prevent the stiffness matrix from becoming singular[1] [26], E is the Young's modulus and p_{E} is the penalization which pushes the optimal design in the direction of a clear material-void design without grey regions (where $\rho_e \neq \{0,1\}$) which is usually desired. A penalization of $p_{\mathrm{E}} \approx 3$ is found to be very effective, lower will often not give a clear material-void design and higher will drop the chance of finding (or getting close to) the global optimum. The SIMP material interpolation became much more popular in topology optimization than the level set function approach, the latter is rarely found in articles related to design dependent load. For that reason we will only consider the SIMP method in this introductory chapter on TO.

For this chapter we will only consider the *minimum compliance* (is maximum stiffness) objective as described in equation A.2 because it is the most commonly used objective. In chapter 4 a different objective is treated with the purpose of designing a compliant mechanism.

[1] Originally the $E^e \neq 0$ was assured using a minimum density $\rho_e \neq 0$, but this interpretation has some advantages, e.g. in filtering [26]

$$
\begin{aligned}
\min_{\boldsymbol{\rho}}: \quad & C = \boldsymbol{u}^{\top}\ \mathbf{K}(\boldsymbol{\rho})\ \boldsymbol{u} = \sum_{e=1}^{n_{\mathrm{e}}} \boldsymbol{u}^{e\top}\mathbf{K}^{e}\boldsymbol{u}^{e}, \\
\text{s.t.}: \quad & \mathbf{K}(\boldsymbol{\rho})\ \boldsymbol{u} = \boldsymbol{F}(\boldsymbol{\rho}) \quad \text{(Follows from a FEA)}, \\
& g_1 = \frac{V}{V_{\max}} - 1 = \frac{\sum_e \rho_e}{n_{\mathrm{e}} V^*} - 1 < 0, \\
& \mathbf{0} \le \boldsymbol{\rho} \le \mathbf{1},
\end{aligned}
\tag{A.2}
$$

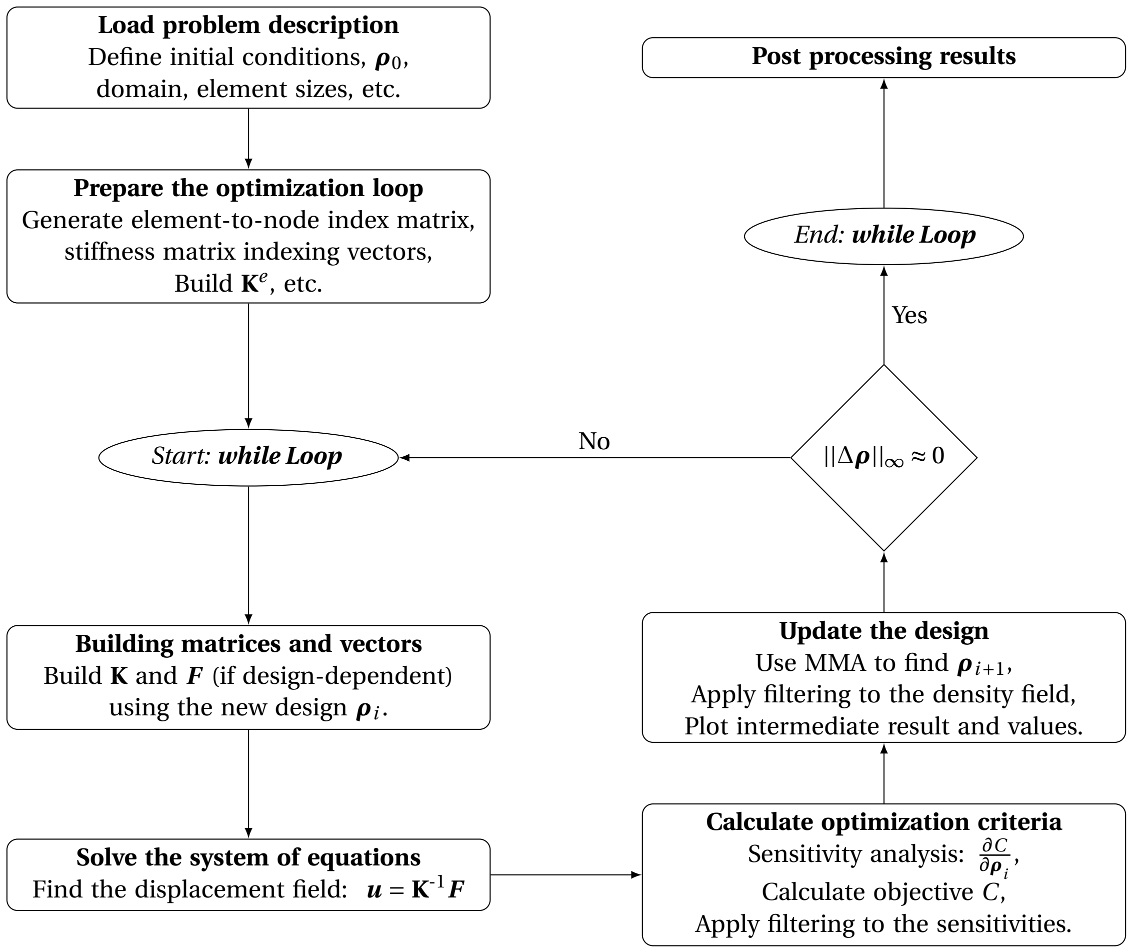
where $\boldsymbol{u}$ is an n dimensional displacement vector, $\mathbf{K}$ the global n by n stiffness matrix, g_1 the first constraint function (required to be below zero), v_e and ρ_e the element volume and density respectively.

A.2. How Topology Optimization works

In order to optimize the density distribution we require information about where we should add material and where to remove it. This is where the *sensitivities* come in, this is the gradient of the objective function C to the design variables ρ_e. From Equation (2.5) one can deduce that it should be of the following form:

$$
\frac{dC}{d\boldsymbol{\rho}} = \left.\frac{\partial C}{\partial \rho_e}\right|_{e=1,\ldots,n_{\mathrm{e}}} = \left[\frac{\partial C}{\partial \rho_1}, \frac{\partial C}{\partial \rho_2}, \ldots, \frac{\partial C}{\partial \rho_{n_e}}\right]^{\top}.
$$

The sensitivities are here assumed to be known. The typical build-up of a topology optimization is as follows:


Load problem description
Define initial conditions, $\boldsymbol{\rho}_0$,
domain, element sizes, etc.
Post processing results
Prepare the optimization loop
Generate element-to-node index matrix,
stiffness matrix indexing vectors,
Build $\mathbf{K}^e$, etc.
*End: **while Loop***
Yes
No
*Start: **while Loop***
$||\Delta\boldsymbol{\rho}||_\infty \approx 0$
Building matrices and vectors
Build $\mathbf{K}$ and $\boldsymbol{F}$ (if design-dependent)
using the new design $\boldsymbol{\rho}_i$.
Update the design
Use MMA to find $\boldsymbol{\rho}_{i+1}$,
Apply filtering to the density field,
Plot intermediate result and values.
Solve the system of equations
Find the displacement field: $\boldsymbol{u} = \mathbf{K}^{-1}\boldsymbol{F}$
Calculate optimization criteria
Sensitivity analysis: $\frac{\partial C}{\partial \boldsymbol{\rho}_i}$,
Calculate objective C,
Apply filtering to the sensitivities.

Where the bottom four boxes are iterated to update the design and make it converge. The convergence is implicitly checked by looking at the inf-norm of the design change $||\Delta\boldsymbol{\rho}||_\infty$ (or the largest absolute value of

$\Delta\boldsymbol{\rho}$) where $\Delta\boldsymbol{\rho} = \boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i$, if this is smaller than a given value the optimization procedure is assumed to have sufficiently converged and stops.

A.3. Filtering

To prevent instability in topology optimization solutions such as checkerboarding and mesh dependency, various filtering methods can be employed in the optimization procedure. Here two methods are described that use a convolution to 'smooth' or 'blur' the solution. One of these methods, the sensitivity filter, introduced by Sigmund, 1997 [24], is applied to the sensitivities and the other to the densities, which is called the density filter [4]. Other methods are to restrict the gradient directly (by adding gradient constraints) or use nodal densities as design variables to effectively prevent checkerboarding, proposed by O. Hammer [13]. Then the element densities are the average of its nodal densities. The density filter directly limits the gradient of the density ($\nabla\boldsymbol{\rho}$) by limiting the variations in the set of admissible designs. We will only elaborate on the sensitivity and density filter as they are most commonly applied.

A.3.1. Density filtering

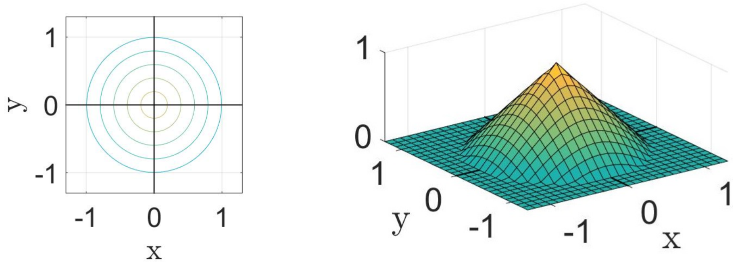
The density filter is described as [4]:

$$\begin{aligned} &E_e(x) = (\rho * K)^p E \quad \text{where} \quad (\rho * K)(x) = \int_\Omega \rho(y) K(x-y)\, dy, \\ &\text{s.t.} \quad \int_\Omega \rho(x) d\Omega \leq V; \quad 0 \leq \rho(x) \leq 1; \quad x \in \Omega, \end{aligned} \tag{A.3}$$

where $*$ is the convolution and $K(x)$ a convolution kernel that effectively averages the element density over directly surrounding elements (weighted through their absolute distance) in a given range or radius. K can for example be:

$$K(x) = \begin{cases} 1 - \frac{||\boldsymbol{x}||}{r} & \text{if } ||\boldsymbol{x}|| \leq r \\ 0 & \text{otherwise} \end{cases} \qquad \text{(See figure A.1)}$$

Where $||\boldsymbol{x}|| = \sqrt{x^2 + y^2}$ (length of the coordinate vector), when implemented in a discretized domain this, in fact, is the distance from the evaluated element to a surrounding element that is being looped over. The kernel function is shown in figure A.1.



(a) Contour plot

(b) Surface plot

Figure A.1: The plot shows the convolution kernel (or function) used to average the density distribution using surrounding elements within a desired radius ($r = 1$ is used in this example)

In a more efficient form for application in a discretized domain we can use:

$$\mathbf{H}(\boldsymbol{x}) = H_{ei}(\boldsymbol{x}_e, \boldsymbol{x}_i) = \begin{cases} 1 - \dfrac{||\boldsymbol{x}_e - \boldsymbol{x}_i||}{r} & \text{if } ||\boldsymbol{x}_e - \boldsymbol{x}_i|| \leq r \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all :} \quad i \in \mathbb{D}_{er} \quad \text{and} \quad e \in \mathbb{D}_e.$$

Where $\mathbb{D}_{er}$ is the set of elements in a given radius r of the element e: $\mathbb{D}_{er} = \left\{ \forall i \in \mathbb{D}_e : ||\boldsymbol{x}_e - \boldsymbol{x}_i|| \leq r \right\}$ for each $e \in \mathbb{D}_e$.

The application could look like (from the 88-line code [1]):

```
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx     %
  for j1 = 1:nely   % Sweeping over all the elements
    e1 = (i1-1)*nely+j1; % Element number in the rho or 'design' vector

    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) % Only loop over surrounding elements
          e2 = (i2-1)*nely+j2;                          % Element number in the rho or 'design' vector
          k  = k+1;
          iH(k) = e1;  % Selecting row in H
          jH(k) = e2;  % Selecting column in H
          sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2)); % Weight factors related to spatial coordinates
       end
    end

  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);          % normalization vector to compensate the weighting factors
```

A.3.2. Sensitivity filtering

A sensitivity filter works similarly to a density filter, only now the convolution is over the sensitivities. The expression for the weight factors on a discrete mesh is as follows:

$$
\begin{aligned}
\widehat{\frac{\partial C}{\partial \rho_e}} &= \frac{1}{\rho_e H_{es}} \sum_{i \in \mathbb{D}_{er}} H_{ei} \rho_i \frac{\partial C}{\partial \rho_i}, \\
H_{ei} &= \sum_{i \in \mathbb{D}_{er}} H_{ei} \quad \text{Which are the rows of } \mathbf{H}.
\end{aligned}
\tag{A.4}
$$

It depends on the numbering of the elements and their corresponding location what the matrix **H** looks like but it is always symmetric with centre-diagonal and off-diagonal rows of components.

A.4. In-loop solvers

In the article *'A 99 line topology optimization code written in Matlab'* by O. Sigmund [25] and later in the 88 line code [1] a bisection algorithm is used to find the new density distribution while satisfying the volume constraint (see g_1 in eq. A.2).

For problems with more constraints it is common to use a solver that returns the improved design using the information available each time it is called upon. Such a solver can be fitted inside the loop as it only does one optimization step allowing the user to recalculate the solver input with the modified design. An example of such a solver that can handle more then one constraint is the method of moving asymptotes (MMA) introduced by Svanberg, 1987 [28]. This method solves an equivalent quadratic problem that is build up using the sensitivities.

B

Appendix: Finite Element Method

The finite element method (FEM) is a numerical technique for solving partial differential equations (PDE's) or functional minimization problems (e.g. $\min_x f(x)$, where x is solved). A PDE is often a relation of a physical variable with space and time. A finite element analysis (FEA) is used to simulate the behaviour of a physical quantity in a given domain. The term FEM refers to the subdivision of the domain in small elements (finite elements of the domain). The nodes are located on the (boundary of the) elements. The physical field is expressed on the element trough discrete values on the nodes, called nodal values. This way a continuous physical problem is transformed into a discretized finite element problem with unknown nodal values. After adding the local contribution of every element in a global matrix we can start to solve the matrix problem: $\mathbf{A}\boldsymbol{x} = \boldsymbol{f}$, for the unknown nodal values in $\boldsymbol{x}$. The matrix $\mathbf{A}$ is usually sparse, symmetric and, in the case of a scalar field, of the size n by n where n is the number of nodes with unknown nodal values. This chapter is based on the book by Cook et al. [8] and establishes the equations, variables and their notation used throughout this thesis.

B.1. FEM in a nutshell

This section will give a brief summery of the steps required to solve a problem using FEM.

1. *Discretize the continuum domain*

   The domain is subdivided into elements and nodes on which we can define local interpolation functions.

2. *Select interpolation functions*

   Choose how to interpret the physical property locally within an element.

3. *Derive the element matrix*

   Transform the problem to a weak formulation and define locally the influence of connected nodes.

4. *Build the global matrix*

   Now loop over all elements and add their local contributions to the global matrix.

5. *Solve the global system of equations*

   Use for example a dedicated software package to solve the set of equations to find the nodal values.

6. *Post-processing of results*

   Calculate additional parameters and plot results.

In this work, we assume a simple rectangular domain with rectangular quadrilateral elements, therefore limited attention is paid on deriving bullets 1 and 2. We attend to bullets 3 and 4 by deriving two weak formulations, one from a partial differential equation (PDE) governing a diffusivity problem and one using the virtual work principle on a elastic deformation problem. The two derivations are treated in Appendix B.3 and Appendix B.4 respectively. The final two bullets depend on the application and will not be treated here.

B.2. The shape functions

In order to use FEM it is required to rewrite the problem into a minimization problem. When working with a differential equation, the *Galerkin method* is the most popular (for e.g. heat problems or other PDE's). If we already have a physical problem that can be formulated as a minimization the *variational formulation* of the finite element equations are usually used (for e.g. minimization of potential energy for elastic bodies). First, a continuous scalar field is described using shape functions and the discretized nodal values.

B.2.1. Choosing the element shape and shape function

In this general case we will look at a 2D domain. The element of interest is a rectangle with four nodes at the corners. We will interpolate this rectangular element using bilinear shape functions (SF) (or basis functions). A property (1) of SF's is that each SF has a value of 1 at a single node and is 0 at the other nodes. A second property (2) is that the sum of the four SF's should equal a surface at one on the entire element. The latter (2) is automatically satisfied by using linear SF's. A more complete list of requirements can be found in [8].

The SF's are derived here in a general framework. This means that first the SF's are defined on a standard element (with fixed dimensions) and second the actual elements are mapped to the standard element to perform integration and differentiation.

A coordinate transformation is later introduced that maps the physical domain with coordinates $\boldsymbol{x} = \begin{bmatrix} x & y \end{bmatrix}^\top$ to the standard element domain with coordinates $\boldsymbol{\xi} = \begin{bmatrix} \xi & \eta \end{bmatrix}^\top$, i.e. $\boldsymbol{x} \rightarrow \boldsymbol{\xi}$, see Figure B.2.

The SF's for a standard element are found by introducing the unknown coefficients **C** and solving them for condition (1). The unknown coefficients are used in the following matrix multiplication where the unit matrix **I** satisfies condition (1):

$$\mathbf{HC} = \mathbf{I} \quad \rightarrow \quad \begin{bmatrix} 1 & \xi_1 & \eta_1 & \xi_1\eta_1 \\ 1 & \xi_2 & \eta_2 & \xi_2\eta_2 \\ 1 & \xi_3 & \eta_3 & \xi_3\eta_3 \\ 1 & \xi_4 & \eta_4 & \xi_4\eta_4 \end{bmatrix} \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{B.1}$$

where **H** consists of ξ_i and η_i which are the known and fixed nodal coordinates on the standard element. The coefficients **C** are directly found by finding the inverse of the *node coordinate* matrix **H**:

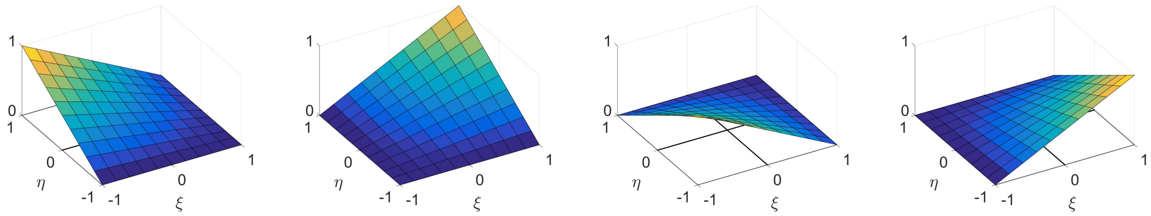
$$\mathbf{C} = \mathbf{H}^{-1}\mathbf{I} \quad \rightarrow \quad \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix}. \tag{B.2}$$

The SF's can now be written in terms of the local coordinates $\boldsymbol{\xi}$:

$$\boldsymbol{N}(\boldsymbol{\xi}) = \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4}(1-\xi)(1+\eta) & \frac{1}{4}(1+\xi)(1+\eta) & \frac{1}{4}(1-\xi)(1-\eta) & \frac{1}{4}(1+\xi)(1-\eta) \end{bmatrix}, \quad [1 \times 4] \tag{B.3}$$

for $-1 \leq \xi \leq 1$ and $-1 \leq \eta \leq 1$. Otherwise, outside the element under consideration: $\boldsymbol{N} = \boldsymbol{0}$ (the null-vector). In Figure B.1 the SF's are plotted on the standard element.

To perform the coordinate mapping, the node coordinates in the real domain $\boldsymbol{x}_n^e = [x_1, y_1, x_2, y_2, ...]^\top$ are re-



(a) Shape function 1 **(b)** Shape function 2 **(c)** Shape function 3 **(d)** Shape function 4

Figure B.1: Shape functions plotted on a standard element. The local node numbering is the same as the shape funcion numbering.

quired. These are defined in the discretisation of the physical domain. The mapping now becomes:

$$\boldsymbol{x}^e = \begin{bmatrix} x = Q(\xi,\eta) \\ y = R(\xi,\eta) \end{bmatrix} = \mathbf{N}(\boldsymbol{\xi})\boldsymbol{x}^e_n = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \end{bmatrix}, \tag{B.4}$$

where the notation '**N**' is introduced in Equation (2.3). The functions Q and R in Equation (B.4) map $\boldsymbol{x} \rightarrow \boldsymbol{\xi}$, this can be used to evaluate equations in terms of ξ and η (e.g. in Equation (B.9)). The mapping is illustrated in Figure B.2. The reverse mapping is not easily defined (and often the mapping does not exist in closed form) [12]. It can however be approximated using a Taylor expansion:

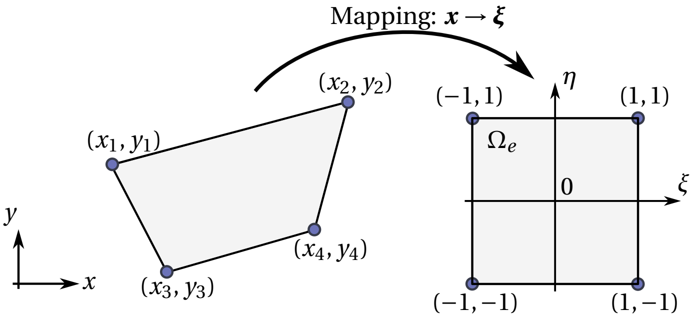


Figure B.2: The mapping: $\boldsymbol{x} \rightarrow \boldsymbol{\xi}$. Left is the real domain and right is the standard element.

$$\begin{gathered} \boldsymbol{\xi} \approx \boldsymbol{\xi}_0 + \left.\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}_0}(\boldsymbol{x}-\boldsymbol{x}_0) + \left.\frac{\partial^2\boldsymbol{\xi}}{\partial\boldsymbol{x}^2}\right|_{\boldsymbol{x}_0}\frac{(\boldsymbol{x}-\boldsymbol{x}_0)^2}{2} = \boldsymbol{\xi}_0 + \mathbf{J}_{\boldsymbol{x}_0}(\boldsymbol{x}-\boldsymbol{x}_0) + \mathbf{H}_{\boldsymbol{x}_0}\frac{(\boldsymbol{x}-\boldsymbol{x}_0)^2}{2}, \\ \text{when evaluated at } \boldsymbol{\xi}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{ and if } \quad \mathbf{H}_{\boldsymbol{x}_0} = \mathbf{0}, \quad \text{we get}: \quad \boldsymbol{\xi} \approx \mathbf{J}_{\tilde{\boldsymbol{x}}}(\boldsymbol{x}-\tilde{\boldsymbol{x}}). \end{gathered} \tag{B.5}$$

Where $\tilde{\boldsymbol{x}}$ is the average of x_i and y_i or the centre of the standard element and $\mathbf{H}_{\boldsymbol{x}_0}$ is the Hessian matrix (of second order partial derivatives) evaluated at $\boldsymbol{x}_0$.

For the transformation to be valid the Jacobian **J** of the transformation Equation (B.4) must be non-singular

$(\det \mathbf{J} \neq 0)$.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}, \qquad \mathbf{J}^{\text{-}1} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix},$$

$$\begin{aligned} \tfrac{\partial x}{\partial \xi} &= \frac{1}{4}\Big(-x_1 + x_2 - x_3 + x_4 + \eta\left(-x_1 + x_2 + x_3 - x_4\right)\Big), \\ \tfrac{\partial x}{\partial \eta} &= \frac{1}{4}\Big(\;\; x_1 + x_2 - x_3 - x_4 + \xi\left(-x_1 + x_2 + x_3 - x_4\right)\Big), \\ \tfrac{\partial y}{\partial \xi} &= \frac{1}{4}\Big(-y_1 + y_2 - y_3 + y_4 + \eta\left(-y_1 + y_2 + y_3 - y_4\right)\Big), \\ \tfrac{\partial y}{\partial \eta} &= \frac{1}{4}\Big(\;\; y_1 + y_2 - y_3 - y_4 + \xi\left(-y_1 + y_2 + y_3 - y_4\right)\Big). \end{aligned} \tag{B.6}$$

We can check the matrix by calculating the determinant of $\mathbf{J}$. Later we will also need this to numerically integrate over the standard element.

$$\begin{gathered} J = \det(\mathbf{J}) = \frac{1}{4^2}\Big(\left(-x_1 + x_2 - x_3 + x_4 + A_x\eta\right)\left(y_1 + y_2 - y_3 - y_4 + A_y\xi\right) - \left(x_1 + x_2 - x_3 - x_4 + A_x\xi\right)\left(-y_1 + y_2 - y_3 + y_4 + A_y\eta\right)\Big), \\ \text{where :} \\ A_x = \left(-x_1 + x_2 + x_3 - x_4\right), \\ A_y = \left(-y_1 + y_2 + y_3 - y_4\right). \end{gathered} \tag{B.7}$$

B.2.2. Describing physical fields using shape functions

Using the SF's, which form a continuous description on Ω_e, we can express the derivative of some field quantity by using the matrix of derivatives. This matrix is commonly named $\mathbf{B}$, but because we will first introduce a pressure field problem and then a deformation problem, the matrix of derivatives for the pressure field is called $\mathbf{B}_p$, while the matrix $\mathbf{B}$ is used to express the strains due to a displacement field. Because $\boldsymbol{N}$ is expressed in local coordinates ξ, η we can apply a chain rule to calculate the derivatives with respect to the global coordinates x, y. We can simply use the Jacobian to convert the derivatives from one coordinate systems to the other, and back.

$$\begin{aligned} \mathbf{B}_p(\boldsymbol{\xi}) = \nabla \boldsymbol{N}(\boldsymbol{\xi}) &= \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} = \frac{1}{4}\begin{bmatrix} -1-\eta & 1+\eta & -1+\eta & 1-\eta \\ 1-\xi & 1+\xi & -1+\xi & -1-\xi \end{bmatrix}, \\ \mathbf{B}_p(\boldsymbol{x}) = \mathbf{J}^{\text{-}1}\mathbf{B}_p(\boldsymbol{\xi}) &= \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix}. \end{aligned} \tag{B.8}$$

A continuous scalar field like pressure, p, can now be expressed using $\boldsymbol{N}$ in the following way:

$$p(\boldsymbol{x}) = p\left(Q(\xi,\eta), R(\xi,\eta)\right) = \sum_{e=1}^{n_e} \boldsymbol{N}(\xi,\eta)\boldsymbol{p}^e \qquad [1 \times 4]\,[4 \times 1] \quad \text{where} \quad \boldsymbol{p}^e = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \tag{B.9}$$

where $\boldsymbol{p}^e$ are the nodal values of the pressure field on element e. The same can be done for a displacement field $u(\boldsymbol{x})$, $v(\boldsymbol{x})$ which are the displacements in x and y direction respectively. When expressing a vector field one needs the matrix of shape functions as introduced in Equation (2.3).

B.3. Solving a pressure field

In order to explain Galerkin's method, it is applied to a PDE that describes a diffusivity problem. In particular the diffusion of a pressure field through a domain of varying porosity as described by Darcy's law. This PDE is transformed to the weak formulation such that it can be solved using FEM. The static PDE that governs the Darcy flux Equation (3.8) is derived in Section 3.1.3 and is repeated here:

$$\nabla \cdot \boldsymbol{q} - Q = 0, \tag{B.10}$$

where:

$\boldsymbol{q}$	=	Darcy flux ($\mathrm{m\,s^{-1}}$),
$\tilde{Q}$	=	fluid source ($\mathrm{s^{-1}}$).

The Darcy flux is related to the spatial derivative of the temperature as given by Darcy's law, repeated here:

$$\boldsymbol{q} = -\frac{\kappa}{\mu}\nabla p \quad = -K\nabla p, \tag{B.11}$$

where:

$\boldsymbol{q}$	=	Darcy flux ($\mathrm{m\,s^{-1}}$),
κ	=	permeability ($\mathrm{m^2}$),
μ	=	fluid viscosity ($\mathrm{Pa\,s}$),
∇p	=	pressure gradient ($\mathrm{N\,m^{-3}}$),
K	=	conduction coefficient ($\mathrm{N\,m^4\,s^{-1}}$).

If we substitute eq. B.11 into eq. B.10 we will end up with a second order PDE. because we want to use linear basis functions this is undesirable in a FEA. Therefore we introduce the weak formulation to lower the order analytically. This is where the Galerkin method comes in to rewrite the PDE as a residual minimization by multiplying all terms by shape functions and integrating over the domain. The integrated PDE is called a weak formulation, and requires a lower order derivative to solve. (Or more intuitively, the integral of a function is a weaker 'function constraint' than the PDE). Finally, after discretisation, the residual (which is the interpolation error) is equated to zero forming the minimization problem. So First put everything to one side:

$$\nabla \cdot \boldsymbol{q} - \tilde{Q} = 0. \quad \text{[scalar]} \tag{B.12}$$

Multiply the equation by the shape function $N_i(\boldsymbol{\xi})$ [scalar] for each shape function on an element ($i = 1, ..., 4$) and integrate over the entire domain (Ω). Because the shape function is only non-zero on the element under consideration, the integration is only nonzero over that element domain (Ω_e).

Summed over each element and for each shape function N_i we get:

$$\sum_{e=1}^{n_e}\left(\int_{\Omega_e}\left(\nabla \cdot \boldsymbol{q} - \tilde{Q}\right)N_i(\boldsymbol{\xi})d\Omega\right)_{i=1,..,4} = 0. \quad \text{[scalar]} \tag{B.13}$$

B.3.1. Weak formulation

Often the expression in the form of a PDE includes higher order derivatives which cannot directly be modelled using linear shape functions. However, the weak formulation provides numerical benefits as it allows us to reduce the order of differentiation. This way, simple linear shape functions can be used that are continuous and only require to be differentiable once.

We will now introduce the divergence theorem or *Gauss theorem* that applies to any vector field $\boldsymbol{u}(\boldsymbol{x})$ in a closed and continuous domain (like Ω):

$$\int_\Omega \nabla \cdot \boldsymbol{u}\ d\Omega = \int_\Gamma \boldsymbol{u} \cdot \boldsymbol{n}\ d\Gamma, \tag{B.14}$$

where Γ is the boundary of the domain Ω and $\boldsymbol{n}$ is the normal to that boundary defined everywhere on Γ. Another theorem we have to introduce defines the divergence of the product of a scalar field $c(\boldsymbol{x})$ and vector field $\boldsymbol{u}(\boldsymbol{x})$:

$$\nabla \cdot (c\boldsymbol{u}) = \nabla c \cdot \boldsymbol{u} + c\ (\nabla \cdot \boldsymbol{u}). \quad \text{[scalar]} \tag{B.15}$$

Integrating eq. B.15 over the whole domain and applying the divergence theorem, eq. B.14, gives us *Green's theorem* [17]:

$$\int_{\Omega} c\,(\nabla\cdot\boldsymbol{u})\ d\Omega = -\int_{\Omega}(\nabla c)\cdot\boldsymbol{u}\ d\Omega + \int_{\Gamma} c\,\boldsymbol{u}\cdot\boldsymbol{n}\ d\Gamma. \quad \text{[scalar]} \tag{B.16}$$

We can directly apply this to our problem Equation (B.13), where we substitute for $\boldsymbol{u}$ the Darcy flux $\boldsymbol{q}$ and for c the individual shape functions N_i. Rewriting Equation (B.13) gives:

$$\sum_{e=1}^{n_e}\left(\int_{\Omega_e} N_i\left(\nabla\cdot\boldsymbol{q}\right)\ d\Omega - \int_{\Omega_e} N_i\tilde{Q}\ d\Omega\right)_{i=1,..,4} = 0. \tag{B.17}$$

After applying Green's theorem we have:

$$\sum_{e=1}^{n_e}\left(-\int_{\Omega_e}\left(\nabla N_i\right)\cdot\boldsymbol{q}\ d\Omega + \int_{\Gamma} N_i\boldsymbol{q}_{\Gamma}\cdot\boldsymbol{n}\ d\Gamma - \int_{\Omega_e} N_i\tilde{Q}\ d\Omega\right)_{i=1,..,4} = 0, \tag{B.18}$$

where $\boldsymbol{q}$ is renamed: $\boldsymbol{q}_{\Gamma}$ because it only requires to be evaluated at the boundary (as a result of Green's theorem, and because in the dot-product the rest would become zero) . Now that we have derived the first order weak form we can back-substitute Darcy's law Equation (B.11) in Equation (B.18):

$$\sum_{e=1}^{n_e}\left(\int_{\Omega_e} K\left(\nabla N_i(\boldsymbol{\xi})\right)\cdot\nabla p(\boldsymbol{x})\ d\Omega\right)_{i=1,..,4} = \sum_{i=1}^{4}\left(\int_{\Omega_e} N_i(\boldsymbol{\xi})\tilde{Q}\ d\Omega - \int_{\Gamma} N_i\boldsymbol{q}_{\Gamma}\cdot\boldsymbol{n}\ d\Gamma\right)_{i=1,..,4}, \quad \text{[scalar]} \tag{B.19}$$

where the right hand terms $\tilde{Q}$ and $\boldsymbol{q}_{\Gamma}\cdot\boldsymbol{n}$ can cover (or house) several different boundary conditions. In this derivation, a draining term (analogous to a convective heat flow term) is added by making the following replacement:

$$\int_{\Omega} N_i\tilde{Q}\ d\Omega = -\int_{\Omega} N_i\left(H(p-p_0)\right)\ d\Omega, \quad \text{[scalar]} \tag{B.20}$$

where:

H	=	drainage coefficient ($\mathrm{m^2\,s^{-1}\,N^{-1}}$),
p	=	continuous pressure field ($\mathrm{N\,m^{-2}}$),
p_0	=	external pressure ($\mathrm{N\,m^{-2}}$) (in this thesis p_0 is set to 0),
$\tilde{Q}$	=	volumetric draining term per unit volume ($\mathrm{s^{-1}}$).

Now in the topology optimization, Equation (B.20) can be tuned through H to effectively alter the pressure field by implementing local (design dependent) drainage. To let the pressure drop to zero, we choose $p_0 = 0$.

B.3.2. Discretisation of the weak form

The weak form still contains the unknown continuous scalar field $p(\boldsymbol{x})$. To be able to numerically solve the pressure distribution we will discretize by substituting Equation (B.9) into Equation (B.19).

$$\sum_{e=1}^{n_e}\left(\int_{\Omega_e} K\left(\nabla N_i\right)\cdot\nabla\boldsymbol{N}\boldsymbol{p}^e\ d\Omega\right)_{i=1,..,4} = \sum_{e=1}^{n_e}\left(-\int_{\Omega_e} H\,N_i(\boldsymbol{N}\boldsymbol{p}^e - p_0)\ d\Omega - \int_{\Gamma} N_i\boldsymbol{q}_{\Gamma}\cdot\boldsymbol{n}\ d\Gamma\right)_{i=1,..,4}, \tag{B.21}$$

where the gradients are defined as follows:

$$\begin{aligned}
\nabla\boldsymbol{N}\boldsymbol{p}^e = \mathbf{B}_p\boldsymbol{p}^e = \begin{bmatrix}\frac{\partial[p]}{\partial x}\\ \frac{\partial[p]}{\partial y}\\ \frac{\partial[p]}{\partial z}\end{bmatrix} &= \begin{bmatrix}\frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x}\\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y}\\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \frac{\partial N_3}{\partial z} & \frac{\partial N_4}{\partial z}\end{bmatrix}\boldsymbol{p}^e, \qquad [3\times 4]\,[4\times 1]\\
\nabla N_i &= \begin{bmatrix}\frac{\partial N_i}{\partial x}\\ \frac{\partial N_i}{\partial y}\\ \frac{\partial N_i}{\partial z}\end{bmatrix}. \qquad [3\times 1]
\end{aligned} \tag{B.22}$$

Now for the final step we have four independent equations, one for each evaluated i. To clarify this further, notice that evaluating the left hand term in Equation (B.21) for N_i results in a vector multiplication of the form: $K\int_{\Omega_e}[1\times 4]_i\,\boldsymbol{p}^e d\Omega$, when evaluated for each i, this gives the desired local matrix of the size: $K\int_{\Omega_e}[4\times 4]\,\boldsymbol{p}^e d\Omega$, providing four equations for the four unknowns in $\boldsymbol{p}^e$.

$$\begin{aligned}
\left(\nabla N_i\cdot\nabla \boldsymbol{N}\boldsymbol{p}^e\right)_{i=1,..,4} &= \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial z} \\ \frac{\partial N_2}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial z} \\ \frac{\partial N_3}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial z} \\ \frac{\partial N_4}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial z} \end{bmatrix} \mathbf{B}_p\ \boldsymbol{p}^e = \mathbf{B}_p^\top(\boldsymbol{x})\,\mathbf{B}_p(\boldsymbol{x})\ \boldsymbol{p}^e = \mathbf{B}_p^\top(\boldsymbol{\xi})\,\mathbf{J}^{\text{-}\top}\,\mathbf{J}^{\text{-}1}\,\mathbf{B}_p(\boldsymbol{\xi})\ \boldsymbol{p}^e, && [4\times 3]\,[3\times 4]\,[4\times 1] \\
\left(N_i\,\boldsymbol{q}(\boldsymbol{x})\cdot\boldsymbol{n}\right)_{i=1,..,4} &= \boldsymbol{N}^\top\left(q_x n_x + q_y n_y + q_z n_z\right)_\Gamma, && [4\times 1]
\end{aligned} \tag{B.23}$$

where $(\mathbf{J}^{\text{-}1})^\top$ is abbreviated by $\mathbf{J}^{\text{-}\top}$. So that we may arrive at the final form, after bringing the pressure dependent terms to the left, Equation (B.23) becomes:

$$\underbrace{\sum_{e=1}^{n_e}\int_{\Omega_e}\mathbf{L}^{e\top}_p\left(K\,\mathbf{B}_p^\top\mathbf{B}_p + H\,\boldsymbol{N}^\top\boldsymbol{N}\right)\mathbf{L}_p^e\ d\Omega}_{\mathbf{A}}\ \boldsymbol{p} = \underbrace{\sum_{e=1}^{n_e}\mathbf{L}^{e\top}_p\left(\int_{\Omega_e} H\,\boldsymbol{N}^\top p_0\ d\Omega - \int_\Gamma \boldsymbol{N}^\top\boldsymbol{q}_\Gamma\cdot\boldsymbol{n}\ d\Gamma\right)}_{\boldsymbol{f}}, \tag{B.24}$$

with $\boldsymbol{p}$ being the global pressure vector of length n and $\mathbf{L}_p^e$ the index matrix that points to the correct location of the local element DOF's in the global DOF vector.

$$\mathbf{A}\boldsymbol{p} = \boldsymbol{f} + \boldsymbol{f}_p. \tag{B.25}$$

Finally, there is a need to prescribe the possible boundary conditions. Γ_p is the boundary of prescribed (nonzero) pressures, they are transferred to the load vector using: $\boldsymbol{f}_p = -\mathbf{A}\boldsymbol{p}_p$. The now fixed pressure DOF's (both $\forall\boldsymbol{p}_p = 0$ and $\forall\boldsymbol{p}_p \neq 0$) are excluded from calculations (such that only the free degrees of freedom are solved for). This boundary condition is called a *Dirichlet* boundary condition.

When a pressure is not prescribed at the wall, $\boldsymbol{f}$ can pose a *Neumann* boundary condition trough $\boldsymbol{q}_\Gamma$, which is the (Darcy) flux trough the boundary. If $\boldsymbol{q}_\Gamma = 0$ and $p_0 = 0$ then the right hand side only contains the contribution of the prescribed pressures: $\boldsymbol{f}_p$. In that case we have a simple static problem where the boundaries without prescribed pressure have zero outgoing flux because $\boldsymbol{q}_\Gamma = 0$. This way we can satisfy the symmetry condition at Γ_s.

B.3.3. Numerical integration

The integrals are often hard to solve analytically so it is common to use numerical integration. We will use the *Legendre-Gauss quadrature* rule to evaluate the integrals. Following the method we evaluate the function at *Gauss points* or *abscissa*[1] in the domain and then we multiply them by corresponding weight factors. First let us formulate the transformation of integrals from a global coordinate system to a local coordinate system using the determinant of the Jacobian.

$$\begin{aligned}
dV = t\,dx\,dy &= \det(\mathbf{J})\ (1)\,d\xi\,d\eta, \\
\int_{\Omega_e(\boldsymbol{x})} f(x,y)\ t\,dx\,dy &= \int_{-1}^{1}\int_{-1}^{1} f(p(\xi,\eta),Q(\xi,\eta))\ \det(\mathbf{J})\ d\xi\,d\eta.
\end{aligned} \tag{B.26}$$

In this case most functions are already defined in the local frame or standard element. Now Gauss's method goes as follows:

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\ \det(\mathbf{J})\ d\xi\,d\eta = \sum_{i=1}^{n}\sum_{j=1}^{n} f(\xi_i,\eta_j)\ \det\!\left(\mathbf{J}(\xi_i,\eta_j)\right)\ w_i w_j. \tag{B.27}$$

It is worth noticing that the numerical integration by the Gauss quadrature rule of the order n is precise for functions of the polynomial order $n+1$. For elements that are only scaled to a width 'a' and height 'b' the application of the Gauss quadrature rule to the right side of eq. B.24 comes down to this:

[1] These are the roots of n degree Lagrange polynomials

Table B.1: Abscissae and Weights of Gauss quadrature

n	ξ_i & η_i	wi
1	0	2
2	$\pm 1/\sqrt{3}$	1
3	0 $\pm\sqrt{3/5}$	8/9 5/9
4	±0.3400 ±0.8611	0.6521 0.3479

```
%% Jacobian calculation of scaled element
J     = [a/2, 0; 0, b/2];
J_inv = [2/a, 0; 0, 2/b];
J_det = a*b/4;

%% Legendre-Gauss integration:
Ae = zeros(4,4);
GaussPoints = [-1/sqrt(3), 1/sqrt(3)];                      % Gauss points
Weights     = [1, 1];                                       % Corresponding Weights
for i = 1:size(GaussPoints);
    eta = GaussPoints(i);
    W_i = Weights(i);
    for j = 1:size(GaussPoints);
        xi  = GaussPoints(j);
        W_j = Weights(j);
        B   = 1/4*[-(1+eta), (1+eta), -(1-eta), (1-eta);
                    (1-xi),  (1+xi),  -(1-xi), -(1+xi)];
        Ae  = Ae+(B'*J_inv')*J_inv*B*J_det*W_i*W_j;         % Element matrix for a  heat flow problem
    end
end
```

This is already enough to solve the steady Darcy-flux problem.

B.4. Solid mechanics

Solid mechanics is about the static deformations and dynamic motions of an elastic or solid body. In this section we will only derive the static (or time independent) elastic deformation. Variational calculus is used to derive the expression of virtual work which, inherited in the method, needs to be minimized to find the displacement field of our elastic body. For this section the book by Ir. J. Blaauwendraad and Ir. A.W.M. Kok, 1973 [5] and additionally the book by R.D. Cook, 2001 [8] are used as reference.

B.4.1. Principle of virtual work

Lets consider a body of solid isotropic material covering the domain Ω with the domain boundary being the sub-domain: $(\delta\Omega =)\Gamma$. External forces act on Ω in the form of body forces ($\boldsymbol{b}$) and on Γ as (Cauchy) tractions ($\boldsymbol{t}$). Imagine a virtual motion of the solid in Ω characterised by arbitrary virtual displacements $\boldsymbol{\delta u}$ which form a displacement field. Then we can write the external work done by this virtual motion $\boldsymbol{\delta u}$ as:

$$\delta W_{\text{ext}} \, (= \boldsymbol{F}\cdot\delta\boldsymbol{u}) \; = \int_{\Gamma} \boldsymbol{t}\cdot\delta\boldsymbol{u}\, dA + \int_{\Omega} \boldsymbol{b}\cdot\delta\boldsymbol{u}\, dV. \tag{B.28}$$

Note that the external work is zero if the body does not deform. The internal stresses are related to the internal traction as: $\boldsymbol{t} = \sigma\boldsymbol{n}$, where σ is the internal stress tensor and $\boldsymbol{n}$ is the unit normal vector of a cross section inside the material. If we cut the solid we expose an inner surface Γ_{cut} that allows us to evaluate the internal stress, we can write:

$$\int_{\Gamma_{\text{cut}}} \boldsymbol{t}\cdot\delta\boldsymbol{u}\, dA = \int_{\Gamma_{\text{cut}}} \sigma\boldsymbol{n}\cdot\delta\boldsymbol{u}\, dA. \tag{B.29}$$

Because the total internal work is the sum of all these cross sections (proof is omitted) we can write:

$$\begin{aligned} \delta W_{\text{int}} &= \int_\Omega \delta \boldsymbol{\varepsilon}^{e\top} \boldsymbol{\sigma}^e \, dV, \\ \text{(for a plane)} \quad \delta W_{\text{int}} &= \int_\Omega \left(\sigma_x \delta\varepsilon_x + \sigma_y \delta\varepsilon_y + \tau_{xy} \delta\gamma_{xy} \right) dV. \end{aligned} \tag{B.30}$$

The material will always deform such that the energy is minimal, e.g. if there is an abundance of internal energy somewhere it is intuitive that it will 'flow' to a place of less energy (or deformation) if physically possible. This automatically satisfies the stress equilibrium, stress-strain relation and continuity. As a result the principle of virtual work states that:

$$\delta W_{\text{ext}} - \delta W_{\text{int}} = 0. \tag{B.31}$$

B.4.2. Discretization

The displacement field can be discretized by using the square elements proposed in section B.2.1 and the associated shape functions given in equation B.3. The matrix of shape functions required to represent a vector field or displacement field is given in equation 2.3 and used here for the displacement field $\boldsymbol{u}(x,y)$:

$$\boldsymbol{u}(\boldsymbol{x}) = \begin{bmatrix} u(P,Q) \\ v(P,Q) \end{bmatrix} = \sum_{e=1}^{n_e} \mathbf{N}(\boldsymbol{\xi}) \boldsymbol{u}^e, \quad [2\times 8]\,[8\times 1] \quad \text{where} \quad \boldsymbol{u}^e = \begin{bmatrix} u_1 & v_1 & u_2 & v_2 & u_3 & v_3 & u_4 & v_4 \end{bmatrix}^\top,$$

and (x,y) and (ξ,η) are related through:

$$\boldsymbol{x}^e = \begin{bmatrix} x = P(\xi,\eta) \\ y = Q(\xi,\eta) \end{bmatrix} = \mathbf{N}(\boldsymbol{\xi}) \mathbf{x}_n^e \quad \text{where} \quad \boldsymbol{x}_n^e = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 & x_3 & y_3 & x_4 & y_4 \end{bmatrix}^\top. \tag{B.32}$$

We can now define the strains as derivatives of the displacement field: $\varepsilon_x = \frac{\partial u(\boldsymbol{x})}{\partial x}$, $\varepsilon_y = \frac{\partial v(\boldsymbol{x})}{\partial y}$ and $\gamma_{xy} = \frac{\partial u(\boldsymbol{x})}{\partial y} + \frac{\partial v(\boldsymbol{x})}{\partial x}$. This can be expressed in the matrix **B** of derivatives of shape functions, using the short hand notation: $N_{1,\xi} = \frac{\partial N_1}{\partial \xi}$ and $u_{,x} = \frac{\partial u}{\partial x}$, note that the **B** derived for solid mechanics purposes is different than that for a pressure distribution problems.

$$\boldsymbol{\varepsilon}^e(\boldsymbol{\xi}) = \mathbf{B}(\boldsymbol{\xi}) \boldsymbol{u}^e, \qquad [3\times 8]\,[8\times 1]$$

where **B** can be expressed by combining the following equations:

$$\boldsymbol{\varepsilon}^e = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_{,x} \\ u_{,y} \\ v_{,x} \\ v_{,y} \end{bmatrix},$$

$$\begin{bmatrix} u_{,x} \\ u_{,y} \\ v_{,x} \\ v_{,y} \end{bmatrix} = \begin{bmatrix} \mathbf{J}^{-1} & 0 \\ 0 & \mathbf{J}^{-1} \end{bmatrix} \begin{bmatrix} u_{,\xi} \\ u_{,\eta} \\ v_{,\xi} \\ v_{,\eta} \end{bmatrix}, \quad [4\times 4]\,[4\times 1]$$

$$\begin{bmatrix} u_{,\xi} \\ u_{,\eta} \\ v_{,\xi} \\ v_{,\eta} \end{bmatrix} = \begin{bmatrix} N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} & 0 \\ N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} & 0 \\ 0 & N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} \\ 0 & N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} \end{bmatrix} \boldsymbol{u}^e,$$

where $\mathbf{B}(\xi,\eta)$ can be evaluated at the nodes to find the nodal strains (and later stresses) and **J** is the Jacobian matrix defined in equation B.6. We can write **B** now explicitly by combining the last three equations:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \xi_{,x} & \eta_{,x} & 0 & 0 \\ \xi_{,y} & \eta_{,y} & 0 & 0 \\ 0 & 0 & \xi_{,x} & \eta_{,x} \\ 0 & 0 & \xi_{,y} & \eta_{,y} \end{bmatrix} \begin{bmatrix} N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} & 0 \\ N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} & 0 \\ 0 & N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} \\ 0 & N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} \end{bmatrix}, \tag{B.33}$$

or in shorter, but maybe less general formulation,

$$\mathbf{B} = \begin{bmatrix} \xi_{,x} & \eta_{,x} & 0 & 0 \\ 0 & 0 & \xi_{,y} & \eta_{,y} \\ \xi_{,y} & \eta_{,y} & \xi_{,x} & \eta_{,x} \end{bmatrix} \begin{bmatrix} N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} & 0 \\ N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} & 0 \\ 0 & N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} \\ 0 & N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} \end{bmatrix}_i \quad [3 \times 4][4 \times 8]. \tag{B.34}$$

Now that the strain is expressed in terms of the discrete displacement field we use Hook's law to find the internal stresses, were E^e is the Young's modulus that can be chosen to vary over different elements.

$$\boldsymbol{\sigma}^e = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} = E^e \frac{1}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2} - \frac{1}{2}\nu \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = E^e \hat{\mathbf{K}} \boldsymbol{\varepsilon}^e, \tag{B.35}$$

where in topology optimization we have a different 'density' (and thus stiffness) for each element. Therefore we introduce the 'hat' ($\hat{\mathbf{K}}$) notation to indicate a unit stiffness matrix (analogous to a unit vector) that requires multiplication by a Young's modulus E^e.

Now that the quantities are described on each element we can substitute this in the energy balance, equation B.31 and rewrite it such that only the discrete displacement field remains unknown.

$$\delta W^e_{\text{int}} = \int_\Omega \delta \boldsymbol{\varepsilon}^{e\top} \boldsymbol{\sigma}^e dV = \int_\Omega \delta \boldsymbol{\varepsilon}^{e\top} \left(E^e \hat{\mathbf{K}} \boldsymbol{\varepsilon}^e \right) dV = E^e \int_\Omega \delta \boldsymbol{u}^{e\top} \mathbf{B}^\top \hat{\mathbf{K}}\, \mathbf{B} \boldsymbol{u}^e dV. \tag{B.36}$$

Notice that the integral is only nonzero on the element in consideration and that $\boldsymbol{u}^e$ is independent of spatial coordinates. We can simplify the expression by defining the stiffness matrix:

$$\begin{aligned} \delta W^e_{\text{int}} &= \delta \boldsymbol{u}^{e\top} \mathbf{K} \boldsymbol{u}^e \\ \mathbf{K} &= E^e \int_\Omega \mathbf{B}^\top \hat{\mathbf{K}}\, \mathbf{B} dV. \end{aligned} \tag{B.37}$$

For the external work we only have to discretize the virtual displacement field in equation B.28.

$$\begin{aligned} \delta W_{\text{ext}} &= \int_\Gamma \delta \boldsymbol{u}^\top \boldsymbol{t}\, dA + \int_\Omega \delta \boldsymbol{u}^\top \boldsymbol{b}\, dV, \\ W^e_{\text{ext}} &= \int_\Gamma \delta \boldsymbol{u}^{e\top} \mathbf{N}^\top \boldsymbol{t}\, dA + \int_\Omega \delta \boldsymbol{u}^{e\top} \mathbf{N}^\top \boldsymbol{b}\, dV = \delta \boldsymbol{u}^{e\top} \boldsymbol{F}^e, \\ \boldsymbol{F}^e &= \int_\Gamma \mathbf{N}^\top \boldsymbol{t}(\boldsymbol{x})\, dA + \int_\Omega \mathbf{N}^\top \boldsymbol{b}(\boldsymbol{x})\, dV, \end{aligned} \tag{B.38}$$

where $\boldsymbol{b}^e = \begin{bmatrix} b_x(x,y) \\ b_y(x,y) \end{bmatrix}$ are the body forces per unit volume acting on the element, like gravitation:

$$\int_{\Omega_e} \mathbf{N}^\top \boldsymbol{b}^e dV = \int_{\Omega_e} \mathbf{N}^\top (\rho g \boldsymbol{e}_2) dV.$$

To add the element contributions to the global stiffness matrix and load vector we define a (binary) location matrix $\mathbf{L}^e_u$ that reshapes the stiffness matrix and force vector such that it acts upon the correct degrees of freedom in the global displacement vector $\boldsymbol{u}$. This allows us to define the element contribution in a global sense:

$$\begin{aligned} \mathbf{K}^e_0 &= \mathbf{L}^{e\top}_u \mathbf{K}^e \mathbf{L}^e_u, \\ \boldsymbol{F}^e_0 &= \mathbf{L}^{e\top}_u \boldsymbol{F}^e. \end{aligned} \tag{B.39}$$

Note that this is rarely the way it is implemented because that would be computationally expensive. We described the internal and external work in terms of the global discrete displacement field. Substitution in the energy balance, equation B.31 holds:

$$\delta \boldsymbol{u}^\top \sum_{e=1}^{n_e} \left(\mathbf{K}^e_0 \boldsymbol{u} - \boldsymbol{F}^e_0 \right) = 0. \tag{B.40}$$

Which should equal zero for every virtual displacement field we choose, thus the terms in brackets should be zero to satisfy the equation.

$$\begin{aligned} \sum_{e=1}^{n_e} \left(\mathbf{K}_0^e \boldsymbol{u} - \boldsymbol{F}_0^e\right) &= 0, \\ \mathbf{K}\boldsymbol{u} &= \boldsymbol{F}. \end{aligned} \tag{B.41}$$

Notice that the factor E^e can be brought out and be multiplied by a unit element stiffness matrix, this is important for the topology optimization using the SIMP method.

$$\sum_{e=1}^{n_e} \left(E^e \hat{\mathbf{K}}_0^e \boldsymbol{u} - \boldsymbol{F}_0^e\right) = 0. \tag{B.42}$$

C

Appendix: Relation between drainage and penetration depth

To have a more intuitive grip on the drainage coefficient h_{mat}, the ordinary differential equation (ODE) is analytically solved for a 1D flow problem trough the wall. The ODE for the adjusted Darcy flow model is:

$$K(\rho_e)\frac{d^2p}{ds^2} = pH(\rho_e), \tag{C.1}$$

where K is the flow coefficient, p the pressure, s the penetration depth and H th drainage coefficient. Because we are simulating the behaviour of the pressure penetrating material we are only considering $\rho_e = 1$. So, Equation (C.1) can be rewritten to[1]:

$$k_{\text{mat}}\frac{d^2p}{ds^2} = ph_{\text{mat}}. \tag{C.2}$$

The boundary conditions (BC's) can be used to express h_{mat} in more intuitive parameters like the penetration depth Δs and the ratio 'r' of input pressure that is left there. Two BC's are required because it is a 2nd order ODE, so we choose:

$$\begin{aligned} \lim_{s\to\infty} p = p_{\text{out}} = 0 \qquad & \text{BC } 1, \\ p(s=0) = p_{\text{in}} \qquad & \text{BC } 2. \end{aligned} \tag{C.3}$$

As a trial solution we choose:

$$p(s) = ae^{-bs} + ce^{bs}, \tag{C.4}$$

where e is Euler's number and a, b and c are unknown coefficients to be determined. Substituting Equation (C.4) in Equation (C.2) holds:

$$\begin{aligned} k_{\text{mat}}b^2(ae^{-bs} + ce^{bs}) &= (ae^{-bs} + ce^{bs})h_{\text{mat}}, \\ \text{that is: } \quad k_{\text{mat}}b^2 &= h_{\text{mat}}, \end{aligned} \tag{C.5}$$

such that:

$$b = \sqrt{\frac{h_{\text{mat}}}{k_{\text{mat}}}}. \tag{C.6}$$

Applying both BC's now gives:

$$\begin{aligned} p(s\to\infty) = \lim_{s\to\infty}\left(ae^{-bs} + ce^{bs}\right) = 0 \qquad &\Rightarrow \qquad c = 0, \\ p(0) = a + c = p_{\text{in}} \qquad &\Rightarrow \qquad a = p_{\text{in}}. \end{aligned} \tag{C.7}$$

[1] The actual value of the flow coefficient k_{mat} depends on the function: $K(\rho_e)$ and how it overlaps the curve $H(\rho_e)$. However, in this work, β_{k} and η_{k} are chosen such that $K(\rho) = k_{\text{mat}}$ before the drainage, $H(\rho)$, significantly kicks in. See Figure 3.2 in Section 3.1.2 for an indication of what the curves may look like and how they could overlap.

The final form now becomes:

$$p(s) = p_{\text{in}} e^{-\sqrt{\frac{h_{\text{mat}}}{k_{\text{mat}}}}s}. \tag{C.8}$$

We now want to find h_{mat} such that: $p(s = \Delta s) = r\,p_{\text{in}}$. Substitution holds:

$$\begin{aligned} r\,p_{\text{in}} &= p_{\text{in}} e^{-\sqrt{\frac{h_{\text{mat}}}{k_{\text{mat}}}}\Delta s}, \\ r &= e^{-\sqrt{\frac{h_{\text{mat}}}{k_{\text{mat}}}}\Delta s}. \end{aligned} \tag{C.9}$$

This can be rewritten to get:

$$h_{\text{mat}} = \left(\frac{\ln r}{\Delta s}\right)^2 k_{\text{mat}}, \tag{C.10}$$

where:

h_{mat}	=	drainage coefficient in material ($\text{m}^2\,\text{N}^{-1}\,\text{s}^{-1}$),
r	=	ratio of input pressure at Δs: $p(\Delta s) = r\,p_{\text{in}}$, ($r = 0.1$ in this work)
Δs	=	penetration depth of pressure (m),
k_{mat}	=	flow coefficient in material (s^{-1}) ($k_{\text{mat}} = 1$ in this work).

D

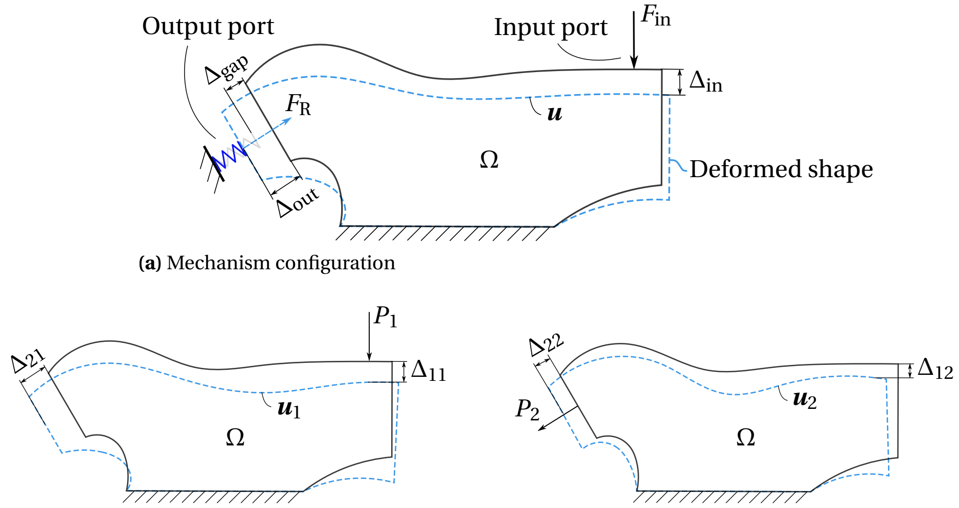
Appendix: A compliant Mechanism formulation

A compliant mechanism is a monolithic structure that inherits its degrees of freedom (DOF's) by removing material in specifically chosen places. This chapter is based on the work of O. Sigmund, 1997 [24]. We can design a compliant mechanism by altering the topology optimization objective and finding the associated sensitivities.

D.1. Objective and constraints

The objective we are interested in is to maximize the "mechanical advantage" M, defined as:

$$M = \frac{F_{\text{out}}}{F_{\text{in}}} = \frac{F_{\text{R}}}{F_{\text{in}}}. \tag{D.1}$$



(a) Mechanism configuration

(b) Load case 1 with an input load

(c) Load case 2 with a dummy output load

Figure D.1: The overall load configuration defined on the design domain Ω can be subdivided into two dummy load cases [24] that need to be solved. The superposition theorem can be applied here to calculate the mechanical advantage and input displacement as a result of the load cases.

In Figure D.1a one can see an example of a compliant mechanism set up. For the purpose of generality we model the output port with a gap and a spring. The output force F_{R} is to be maximized with respect to the input force F_{in}, while taking into account the spring constant k_s and the gap Δ_{gap}. As a constraint of the design we will also require the input displacement Δ_{in}. To find M and Δ_{in} we define two load cases with dummy loads P_1 and P_2 in $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ respectively, see Figure D.1b and Figure D.1c. And we solve both matrix equations for the displacement vector:

$$\mathbf{K}\boldsymbol{u}_1 = \boldsymbol{F}_1 \qquad \text{and} \qquad \mathbf{K}\boldsymbol{u}_2 = \boldsymbol{F}_2. \tag{D.2}$$

Then we define the 'unit load' displacements Δ_{ij} where i and j are the location and load case respectively as indicated in Figure D.1b and Figure D.1c.

$$\Delta_{ij} = \boldsymbol{u}_j{}^\top \hat{\boldsymbol{e}}_i = \boldsymbol{u}_j{}^\top \frac{\boldsymbol{F}_i}{P_i} = \boldsymbol{u}_j{}^\top \mathbf{K}\boldsymbol{u}_i \frac{1}{P_i}, \qquad \forall \begin{Bmatrix} i \in \{1,2\} \\ j \in \{1,2\} \end{Bmatrix} \quad \text{(no summation over } i,j\text{)}. \tag{D.3}$$

The output force F_R can now be expressed as: $F_{\text{R}} = (\Delta_{\text{out}} - \Delta_{\text{gap}})k_s$, assuming the outward deflection is greater than the gap. Rewriting this holds: $\Delta_{\text{out}} = \Delta_{\text{gap}} + \frac{F_{\text{R}}}{k_s}$. We can now use the superposition theorem for linear systems to add load cases and associated forces linearly.

$$\Delta_{21} - c\Delta_{22} = \Delta_{\text{out}}. \tag{D.4}$$

Which states that the output displacement is the displacement at the output due to a force at the input minus c times the displacement at the output due to the reaction force $F_{\text{R}} = cP_2$, where c is a constant.

$$c = \frac{\Delta_{21} - \Delta_{\text{gap}} - \frac{F_{\text{R}}}{k_s}}{\Delta_{22}}. \tag{D.5}$$

$$F_{\text{R}} = P_2 c \; = \; \frac{P_2(\Delta_{21} - \Delta_{\text{gap}})}{\Delta_{22}} - \frac{F_{\text{R}} P_2}{\Delta_{22} k_s} = \; P_2 \frac{\Delta_{21} - \Delta_{\text{gap}}}{\Delta_{22} + \frac{P_2}{k_s}}.$$

Note that the objective function is: $M = \frac{F_R}{P_1}$, so we can write M as:

$$M = \frac{P_2}{P_1} \frac{\Delta_{21} - \Delta_{\text{gap}}}{\Delta_{22} + \frac{P_2}{k_s}}. \tag{D.6}$$

The input displacement is the same linear combination as Equation (D.4), defined as follows:

$$\Delta_{\text{in}} = \Delta_{11} - c\Delta_{12} = \Delta_{11} - \frac{\Delta_{12}}{\Delta_{22}} \left(\Delta_{21} - \Delta_{\text{gap}} - \frac{F_{\text{R}}}{k_s} \right). \tag{D.7}$$

Now that the objective function is quantified in terms of the displacement field of two load cases we can start deriving the sensitivities. Note that, if desired, the gap can be removed by setting $\Delta_{gap} = 0$ and the outward deflection can be completely blocked by letting $k_s \to \infty$, effectively modelling a pressing load on a rigid body.

The minimization problem is now as follows:

$$\begin{aligned} \min_{\boldsymbol{\rho}}: \quad & -M(\boldsymbol{\rho}, \boldsymbol{u}_{1,2}, \boldsymbol{F}_{1,2}), \\ \text{s.t.}: \quad & g_1 = \frac{\Delta_{\text{in}}}{\Delta_{\text{in}}^*} - 1 \leq 0, \\ & g_2(\boldsymbol{\rho}) = \frac{V(\boldsymbol{\rho})}{V_{\max}} - 1 = \frac{\sum_{e=1}^{N_e} (v_e \rho_e)}{V_{\max}} - 1 \leq 0, \\ & \mathbf{K}\,\boldsymbol{u}_1 = \boldsymbol{F}_1, \quad \mathbf{K}\,\boldsymbol{u}_2 = \boldsymbol{F}_2, \\ & 0 \leq \rho_i \leq 1 \quad \forall i \in \mathbb{D}_e, \end{aligned} \tag{D.8}$$

where Δ_{in}^* is the maximum allowable input displacement and the set $\mathbb{D}_e = \{1, 2, ..., N_e\}$.

D.2. Sensitivities

The sensitivities are the total derivatives of the objective to the individual design variables, $\boldsymbol{S} = \frac{dM(\boldsymbol{\rho},\boldsymbol{u}_{1,2}(\boldsymbol{\rho}),\boldsymbol{F}_{1,2}(\boldsymbol{\rho}))}{d\boldsymbol{\rho}}$. Using the quotient rule on Equation (D.6) gives:

$$S_e = \frac{dM}{d\rho_e} = \frac{P_2}{P_1}\left(\frac{\frac{\partial\Delta_{21}}{\partial\rho_e}\left(\Delta_{22}+\frac{P_2}{k_s}\right) - \frac{\partial\Delta_{22}}{\partial\rho_e}\left(\Delta_{21}-\Delta_{\text{gap}}\right)}{\Delta_{22}+\frac{P_2}{k_s}}\right), \qquad \forall e \in \mathbb{D}_e, \tag{D.9}$$

with k_s, Δ_{gap}, P_1 and P_2 constant w.r.t. ρ_e. The partial derivatives $\frac{\partial\Delta_{ij}}{\partial\rho_e}$ can be found using the adjoint method. Introducing the Lagrange multipliers λ_i and λ_j we can write Δ_{ij} as:

$$\Delta^*_{ij} = \frac{1}{P_i}\boldsymbol{u}_j{}^\top\mathbf{K}\boldsymbol{u}_i + \lambda_i{}^\top\left(\mathbf{K}\,\boldsymbol{u}_i - \boldsymbol{F}_i\right) + \lambda_j{}^\top\left(\mathbf{K}\,\boldsymbol{u}_j - \boldsymbol{F}_j\right). \tag{D.10}$$

Upon differentiation we get a rather lengthy equation, collecting terms it becomes:

$$\frac{\partial\Delta^*_{ij}}{\partial\rho_e} = \left(\frac{1}{P_i}\boldsymbol{u}_i{}^\top\mathbf{K}^\top + \lambda_i\mathbf{K}\right)\frac{\partial\boldsymbol{u}_j}{\partial\rho_e} + \left(\frac{1}{P_i}\boldsymbol{u}_j{}^\top\mathbf{K} + \lambda_j\mathbf{K}\right)\frac{\partial\boldsymbol{u}_i}{\partial\rho_e} + \frac{1}{P_i}\boldsymbol{u}_j{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_i - \lambda_i{}^\top\frac{\partial\boldsymbol{F}_i}{\partial\rho_e} - \lambda_j{}^\top\frac{\partial\boldsymbol{F}_j}{\partial\rho_e} + \lambda_i{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_i + \lambda_j{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_j. \tag{D.11}$$

Where we can choose the Lagrange multipliers to be: $\lambda_i = -\frac{1}{P_i}\boldsymbol{u}_j$ and $\lambda_j = -\frac{1}{P_i}\boldsymbol{u}_i$ to extinguish the derivatives: $\frac{\partial\boldsymbol{u}_i}{\partial\rho_e}$ and $\frac{\partial\boldsymbol{u}_j}{\partial\rho_e}$. We can substitute this and rewrite Equation (D.11) to get:

$$\frac{\partial\Delta^*_{ij}}{\partial\rho_e} = \frac{1}{P_i}\boldsymbol{u}_j{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_i - \frac{1}{P_i}\boldsymbol{u}_j{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_i - \frac{1}{P_i}\boldsymbol{u}_i{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_j + \frac{1}{P_i}\boldsymbol{u}_j{}^\top\frac{\partial\boldsymbol{F}_i}{\partial\rho_e} + \frac{1}{P_i}\boldsymbol{u}_i{}^\top\frac{\partial\boldsymbol{F}_j}{\partial\rho_e}. \tag{D.12}$$

Using the fact that $\frac{\partial\mathbf{K}}{\partial\rho_e} = \frac{\partial\mathbf{K}^\top}{\partial\rho_e}$ because $\mathbf{K}$ is symmetric, and the fact that these are scalar terms, we can transpose terms at will and rewrite to the following form:

$$\frac{\partial\Delta^*_{ij}}{\partial\rho_e} = \frac{1}{P_i}\left(-\boldsymbol{u}_j{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_i + \boldsymbol{u}_j{}^\top\frac{\partial\boldsymbol{F}_i}{\partial\rho_e} + \boldsymbol{u}_i{}^\top\frac{\partial\boldsymbol{F}_j}{\partial\rho_e}\right) \qquad \forall\begin{Bmatrix} i \in \{1,2\} \\ j \in \{1,2\}\end{Bmatrix}. \tag{D.13}$$

For the two derivatives required in Equation (D.9), this becomes:

$$\begin{aligned}\frac{\partial\Delta_{21}}{\partial\rho_e} &= \frac{1}{P_2}\left(-\boldsymbol{u}_1{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_2 + \boldsymbol{u}_1{}^\top\frac{\partial\boldsymbol{F}_2}{\partial\rho_e} + \boldsymbol{u}_2{}^\top\frac{\partial\boldsymbol{F}_1}{\partial\rho_e}\right),\\ \frac{\partial\Delta_{22}}{\partial\rho_e} &= \frac{1}{P_2}\left(-\boldsymbol{u}_2{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_2 + \boldsymbol{u}_2{}^\top\frac{\partial\boldsymbol{F}_2}{\partial\rho_e} + \boldsymbol{u}_2{}^\top\frac{\partial\boldsymbol{F}_2}{\partial\rho_e}\right),\end{aligned}$$

where the derivative $\frac{\partial\mathbf{K}}{\partial\rho_e}$ follows from the chosen material model, which is the SIMP model in this work. In the absence of design-dependent loads the derivatives $\frac{\partial\boldsymbol{F}_{1,2}}{\partial\rho_e} = 0$ and the sensitivities S_e simplify to:

$$S_e = \frac{dM}{d\rho_e} = \frac{1}{P_1\left(\Delta_{22}+\frac{P_2}{k_s}\right)}\left(\boldsymbol{u}_2{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_2\left(\Delta_{21}-\Delta_{\text{gap}}\right) - \boldsymbol{u}_1{}^\top\frac{\partial\mathbf{K}}{\partial\rho_e}\boldsymbol{u}_2\left(\Delta_{22}+\frac{P_2}{k_s}\right)\right), \qquad \forall e \in \mathbb{D}_e \tag{D.14}$$

where Δ_{ij} is as in Equation (D.3) and $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ follow from Equation (D.2).

D.3. Actuation by pressure loading

In the history of literature about topology optimization there are very few articles that discuss the topic of compliant mechanism design with design dependent loading and its associated problems.

H. Panganiban et al., 2010 [22] used a method proposed by Sigmund [26] (described in Section 1.2) by using a three phase design problem (fluid, material and void) to design a compliant mechanism. [22] redefined the input displacement constraint as this displacement is defined in terms of a single point. Instead of constraining the input displacement at a single point, Panganiban constraints the mean compliance due to the applied pressure.

$$\begin{aligned} &U = \int_{\Gamma_p} \boldsymbol{u}(x,y)\cdot \boldsymbol{p} d\Gamma \leq \int_{\Gamma_p} \boldsymbol{u}_{\max}\cdot \boldsymbol{p} d\Gamma = U_{\max}, && \text{(where } \boldsymbol{u} \text{ is a continuous displacement field)} \\ &U \approx \boldsymbol{u}^\top \boldsymbol{F}_{\text{input}}, && \text{(where } \boldsymbol{u} \text{ are the discritized nodal displacements).} \end{aligned} \tag{D.15}$$

A second article by S. Vasista and L. Tong, 2012 [31] designs pressure actuated cells which, by combining, form morphing structures (like in plants). Vasista and Tong also use the incompressible medium and the mixed u/P finite element formulation together with SIMP as described in [26, 35] and additionally together with the moving isosurface threshold (MIST) topology optimization method [29].

E

Appendix: Sensitivity derivation using the adjoint method

Sensitivities, are required by the optimization algorithm to give it a sense of direction. The sensitivities are calculated by taking the derivative of the objective function to each design variable:

$$S_e = \frac{\partial C}{\partial \rho_e}. \tag{E.1}$$

The optimisation problem can, without additional[1] constraints, be written as:

$$\begin{array}{rll} \min\limits_{\boldsymbol{\rho}}: & & C_1 = -F_{\text{out}}(\boldsymbol{\rho}) \quad \text{or} \quad C_2 = -\psi(\boldsymbol{\rho}), \\ \text{subject to}: & 1) & \mathbf{A}\boldsymbol{p} = \boldsymbol{f}, \\ & 2) & \mathbf{K}\boldsymbol{u}_1 = \boldsymbol{F}_1(\boldsymbol{\rho}) = -\mathbf{H}(\boldsymbol{\rho})\boldsymbol{p}, \\ & 3) & \mathbf{K}\boldsymbol{u}_2 = \boldsymbol{F}_2, \\ & & \mathbf{0} \leq \boldsymbol{\rho} \leq \mathbf{1}, \\ g_1 = & & \dfrac{\sum_e \rho_e}{n_{\text{e}} V^*} - 1 < 0, \end{array} \tag{E.2}$$

where first the FEA of the pressure field (1) is solved and then the two load cases (2 and 3) are solved. The global stiffness matrix **K** is the same in both load cases (and throughout this work). The global stiffness matrix and conductivity matrix can respectively be split in submatrices containing contributions to free and prescribed DOF's, like:

$$\left(\mathbf{K}\boldsymbol{u}_i = \boldsymbol{F}_i\right) \Rightarrow \begin{bmatrix} \mathbf{K}_{\text{ff}} & \mathbf{K}_{\text{fp}} \\ \mathbf{K}_{\text{pf}} & \mathbf{K}_{\text{pp}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{\text{f}_i} \\ \boldsymbol{u}_{\text{p}_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_{\text{f}_i} \\ \boldsymbol{F}_{\text{p}_i} \end{bmatrix}, \qquad \left(\mathbf{A}\boldsymbol{p}_i = \boldsymbol{f}_i\right) \Rightarrow \begin{bmatrix} \mathbf{A}_{\text{ff}} & \mathbf{A}_{\text{fp}} \\ \mathbf{A}_{\text{pf}} & \mathbf{A}_{\text{pp}} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_{\text{f}_i} \\ \boldsymbol{p}_{\text{p}_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{\text{f}_i} \\ \boldsymbol{f}_{\text{p}_i} \end{bmatrix}, \tag{E.3}$$

where the same goes for sub-index j. The matrix **H** can also be split, such that the load vector can be given by the following:

$$\begin{array}{rcl} \left(\boldsymbol{F}_i = -\mathbf{H}\boldsymbol{p}_i\right) \Rightarrow \begin{bmatrix} \boldsymbol{F}_{\text{f}_i} \\ \boldsymbol{F}_{\text{p}_i} \end{bmatrix} & = - & \begin{bmatrix} \mathbf{H}_{\text{ff}} & \mathbf{H}_{\text{fp}} \\ \mathbf{H}_{\text{pf}} & \mathbf{H}_{\text{pp}} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_{\text{f}_i} \\ \boldsymbol{p}_{\text{p}_i} \end{bmatrix}, \\ \text{such that}: \qquad \boldsymbol{F}_{\text{f}_i} & = & -\,\mathbf{H}_{\text{ff}}\,\boldsymbol{p}_{\text{f}_i} - \mathbf{H}_{\text{fp}}\,\boldsymbol{p}_{\text{p}_i}, \end{array} \tag{E.4}$$

where again the same goes for sub-index j and where we can note that the prescribed displacements are zero (fixed), so $\boldsymbol{u}_{\text{p}_i} = \mathbf{0}$. The values in the prescribed pressure vector $\boldsymbol{p}_{\text{p}_i}$ can either be zero or P_{in}. The derivatives of both prescribed vectors: $\frac{d\boldsymbol{u}_{\text{p}_i}}{d\boldsymbol{\rho}} = \mathbf{0}$ and $\frac{d\boldsymbol{p}_{\text{p}_i}}{d\boldsymbol{\rho}} = \mathbf{0}$ because they do not alter with changes in the design. In the rest of this appendix, a derivative will be indicated with an 'apostrophe': $\boldsymbol{u}'_{\text{f}_i} \equiv \frac{d\boldsymbol{u}_{\text{f}_i}}{d\boldsymbol{\rho}}$.

[1] 'Additional' refers to the constraints suggested in Section 5. This is the minimal working problem set-up. Adding constraints does not affect this derivation.

For the derivation of the sensitivities the tricky part is finding Δ'_{ij}. If that is derived we can directly formulate the derivative of the output force (OF) F'_{out}, of the Energy efficiency (EE) ψ' and even the sensitivities of the minimal compliance (MC) objective: $C_3 = (\boldsymbol{u}_1^\top \mathbf{K} \boldsymbol{u}_1)$.

$$\Delta_{ij} = \boldsymbol{u}_j^\top \hat{\boldsymbol{e}}_i = \boldsymbol{u}_j^\top \frac{\boldsymbol{F}_i}{P_i} = \boldsymbol{u}_j^\top \mathbf{K} \boldsymbol{u}_i \frac{1}{P_i}, \qquad \forall \begin{Bmatrix} i \in \{1,2\} \\ j \in \{1,2\} \end{Bmatrix}, \quad \text{(no summation over } i, j\text{)}, \tag{E.5}$$

where $\hat{\boldsymbol{e}}_i$ is a unit vector that points to the desired DOF that designates the direction of port i.

E.1. Adjoint method

For the derivation we will take the second term of Equation (E.5). Because the derivative $\boldsymbol{u}'$ and $\boldsymbol{p}'$ are explicitly defined, we need to use an adjoint equation (indicated by $(\ldots)^*$) to find the derivative. So, 'zero terms' are multiplied with Lagrange multipliers and added to the equation. Here, the formulation is kept general by introducing pressure fields for both an input (i) and an output (j) port, they can always be set to zero if they are unused.

Introducing the Lagrange multipliers $\boldsymbol{\lambda}_i$, $\boldsymbol{\lambda}_j$, $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ we can write Δ_{ij} Equation (E.5) as:

$$\begin{aligned} \Delta^*_{ij} = \frac{1}{P_i}\, {\boldsymbol{u}_{\text{f}_j}}^\top \mathbf{K}_{\text{ff}}\, \boldsymbol{u}_{\text{f}_i} + \boldsymbol{\lambda}_i^\top \left(\mathbf{K}_{\text{ff}}\, \boldsymbol{u}_{\text{f}_i} - \boldsymbol{F}_{\text{f}_i} \right) + \boldsymbol{\lambda}_j^\top \left(\mathbf{K}_{\text{ff}}\, \boldsymbol{u}_{\text{f}_j} - \boldsymbol{F}_{\text{f}_j} \right) + \\ \boldsymbol{\mu}_i^\top \left(\mathbf{A}_{\text{ff}}\, \boldsymbol{p}_{\text{f}_i} + \mathbf{A}_{\text{fp}}\, \boldsymbol{p}_{\text{p}_i} - \boldsymbol{f}_{\text{f}_i} \right) + \boldsymbol{\mu}_j^\top \left(\mathbf{A}_{\text{ff}}\, \boldsymbol{p}_{\text{f}_j} + \mathbf{A}_{\text{fp}}\, \boldsymbol{p}_{\text{p}_j} - \boldsymbol{f}_{\text{f}_j} \right), \end{aligned} \tag{E.6}$$

in which we now substitute $\boldsymbol{F}_{\text{f}_i} = -\,\mathbf{H}_{\text{ff}}\, \boldsymbol{p}_{\text{f}_i} - \mathbf{H}_{\text{fp}}\, \boldsymbol{p}_{\text{p}_i}$, i.e. the pressure loads, note that $\mathbf{H}_{\text{ff}}$ is of the size $[nd \times n]$ and the subscripts denote free displacement DOF's and free pressure DOF's respectively. Taking its derivative holds: $\boldsymbol{F}'_{\text{f}_i} = -\,\mathbf{H}_{\text{ff}}\, \boldsymbol{p}'_{\text{f}_i}$, because $\mathbf{H}$ does not depend on ρ_e and $\boldsymbol{p}_{\text{p}_i}{}' = \mathbf{0}$. Upon differentiating Equation (E.6) we get:

$$\begin{aligned} \frac{d\Delta^*_{ij}}{d\rho_e} = \frac{1}{P_i} \left({\boldsymbol{u}'_{\text{f}_j}}^\top \mathbf{K}_{\text{ff}}\, \boldsymbol{u}_{\text{f}_i} + {\boldsymbol{u}_{\text{f}_j}}^\top \mathbf{K}'_{\text{ff}}\, \boldsymbol{u}_{\text{f}_i} + {\boldsymbol{u}_{\text{f}_j}}^\top \mathbf{K}_{\text{ff}}\, \boldsymbol{u}'_{\text{f}_i} \right) + \boldsymbol{\lambda}_i^\top \left(\mathbf{K}'_{\text{ff}}\, \boldsymbol{u}_{\text{f}_i} + \mathbf{K}_{\text{ff}}\, \boldsymbol{u}'_{\text{f}_i} - \boldsymbol{F}'_{\text{f}_i} \right) + \boldsymbol{\lambda}_j^\top \left(\mathbf{K}'_{\text{ff}}\, \boldsymbol{u}_{\text{f}_j} + \mathbf{K}_{\text{ff}}\, \boldsymbol{u}'_{\text{f}_j} - \boldsymbol{F}'_{\text{f}_j} \right) + \\ \boldsymbol{\mu}_i^\top \left(\mathbf{A}'_{\text{ff}}\, \boldsymbol{p}_{\text{f}_i} + \mathbf{A}_{\text{ff}}\, \boldsymbol{p}'_{\text{f}_i} + \mathbf{A}'_{\text{fp}}\, \boldsymbol{p}_{\text{p}_i} + \mathbf{A}_{\text{fp}}\, \boldsymbol{p}_{\text{p}_i}{}' - \boldsymbol{f}'_{\text{f}_i} \right) + \boldsymbol{\mu}_j^\top \left(\mathbf{A}'_{\text{ff}}\, \boldsymbol{p}_{\text{f}_j} + \mathbf{A}_{\text{ff}}\, \boldsymbol{p}'_{\text{f}_j} + \mathbf{A}'_{\text{fp}}\, \boldsymbol{p}_{\text{p}_j} + \mathbf{A}_{\text{fp}}\, \boldsymbol{p}_{\text{p}_j}{}' - \boldsymbol{f}'_{\text{f}_j} \right), \end{aligned} \tag{E.7}$$

where it is assumed [2] that all $\boldsymbol{f}'_{\text{f}} = \mathbf{0}$, all $\boldsymbol{f}_{\text{p}}{}' = \mathbf{0}$ and all $\boldsymbol{p}_{\text{p}}{}' = \mathbf{0}$. By collecting the terms in Equation (E.7), it becomes:

$$\begin{aligned} \frac{d\Delta^*_{ij}}{d\rho_e} = \left(\frac{1}{P_i}\, {\boldsymbol{u}_{\text{f}_j}}^\top \mathbf{K}'_{\text{ff}}\, \boldsymbol{u}_{\text{f}_i} + \boldsymbol{\lambda}_i^\top \mathbf{K}'_{\text{ff}}\, \boldsymbol{u}_{\text{f}_i} + \boldsymbol{\lambda}_j^\top \mathbf{K}'_{\text{ff}}\, \boldsymbol{u}_{\text{f}_j} \right) + \left(\boldsymbol{\mu}_i^\top \mathbf{A}'_{\text{ff}}\, \boldsymbol{p}_{\text{f}_i} + \boldsymbol{\mu}_i^\top \mathbf{A}'_{\text{fp}}\, \boldsymbol{p}_{\text{p}_i} \right) + \left(\boldsymbol{\mu}_j^\top \mathbf{A}'_{\text{ff}}\, \boldsymbol{p}_{\text{f}_j} + \boldsymbol{\mu}_j^\top \mathbf{A}'_{\text{fp}}\, \boldsymbol{p}_{\text{p}_j} \right) + \\ \left(\frac{1}{P_i}\, {\boldsymbol{u}_{\text{f}_j}}^\top \mathbf{K}_{\text{ff}} + \boldsymbol{\lambda}_i^\top \mathbf{K}_{\text{ff}} \right) \boldsymbol{u}'_{\text{f}_i} + \left(\frac{1}{P_i}\, {\boldsymbol{u}_{\text{f}_i}}^\top \mathbf{K}_{\text{ff}} + \boldsymbol{\lambda}_j^\top \mathbf{K}_{\text{ff}} \right) \boldsymbol{u}'_{\text{f}_j} + \left(\boldsymbol{\mu}_i^\top \mathbf{A}_{\text{ff}} + \boldsymbol{\lambda}_i^\top \mathbf{H}_{\text{ff}} \right) \boldsymbol{p}'_{\text{f}_i} + \left(\boldsymbol{\mu}_j^\top \mathbf{A}_{\text{ff}} + \boldsymbol{\lambda}_j^\top \mathbf{H}_{\text{ff}} \right) \boldsymbol{p}'_{\text{f}_j}, \end{aligned} \tag{E.8}$$

where the last four brackets have to sum to zero in order to extinguish the explicit variables. From this we can state that:

$$\begin{aligned} \boldsymbol{\lambda}_i &= -\frac{1}{P_i}\, \boldsymbol{u}_{\text{f}_j}, \\ \boldsymbol{\lambda}_j &= -\frac{1}{P_i}\, \boldsymbol{u}_{\text{f}_i}, \\ \boldsymbol{\mu}_i &= -\,{\mathbf{A}_{\text{ff}}}^{-1}\, {\mathbf{H}_{\text{ff}}}^\top \boldsymbol{\lambda}_i = \frac{1}{P_i}\, {\mathbf{A}_{\text{ff}}}^{-1}\, {\mathbf{H}_{\text{ff}}}^\top \boldsymbol{u}_{\text{f}_j}, \\ \boldsymbol{\mu}_j &= -\,{\mathbf{A}_{\text{ff}}}^{-1}\, {\mathbf{H}_{\text{ff}}}^\top \boldsymbol{\lambda}_j = \frac{1}{P_i}\, {\mathbf{A}_{\text{ff}}}^{-1}\, {\mathbf{H}_{\text{ff}}}^\top \boldsymbol{u}_{\text{f}_i}, \end{aligned} \tag{E.9}$$

[2] If the pressure field calculation would be altered, this assumption cannot be made. In this work, the chosen methods allow us to assume this.

where we used the fact that $\mathbf{K}' = \mathbf{K}'^{\top}$ and $\mathbf{A}' = \mathbf{A}'^{\top}$ because they are symmetric. As all the encountered terms are scalars we can freely transpose them without transposing the whole equation. Now that the Lagrange multipliers are known, Equation (E.8) can be simplified by substituting $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_j$ ($\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are kept in place because they are solved separately):

$$\frac{d\Delta^*_{ij}}{d\rho_e} = \frac{1}{P_i}\left(\boldsymbol{u}_{\mathrm{f}_j}{}^{\top}\, \mathbf{K}'_{\mathrm{ff}}\, \boldsymbol{u}_{\mathrm{f}_i} - \boldsymbol{u}_{\mathrm{f}_j}{}^{\top}\, \mathbf{K}'_{\mathrm{ff}}\, \boldsymbol{u}_{\mathrm{f}_i} - \boldsymbol{u}_{\mathrm{f}_i}{}^{\top}\, \mathbf{K}'_{\mathrm{ff}}\, \boldsymbol{u}_{\mathrm{f}_j}\right) + \boldsymbol{\mu}_i^{\top}\left(\mathbf{A}'_{\mathrm{ff}}\, \boldsymbol{p}_{\mathrm{f}_i} + \mathbf{A}'_{\mathrm{fp}}\, \boldsymbol{p}_{\mathrm{p}_i}\right) + \boldsymbol{\mu}_j^{\top}\left(\mathbf{A}'_{\mathrm{ff}}\, \boldsymbol{p}_{\mathrm{f}_j} + \mathbf{A}'_{\mathrm{fp}}\, \boldsymbol{p}_{\mathrm{p}_j}\right),$$

now transposing the third term in the brackets, the final form becomes:

$$\frac{d\Delta^*_{ij}}{d\rho_e} = -\frac{1}{P_i}\, \boldsymbol{u}_{\mathrm{f}_j}{}^{\top}\, \mathbf{K}'_{\mathrm{ff}}\, \boldsymbol{u}_{\mathrm{f}_i} + \boldsymbol{\mu}_i^{\top}\left(\mathbf{A}'_{\mathrm{ff}}\, \boldsymbol{p}_{\mathrm{f}_i} + \mathbf{A}'_{\mathrm{fp}}\, \boldsymbol{p}_{\mathrm{p}_i}\right) + \boldsymbol{\mu}_j^{\top}\left(\mathbf{A}'_{\mathrm{ff}}\, \boldsymbol{p}_{\mathrm{f}_j} + \mathbf{A}'_{\mathrm{fp}}\, \boldsymbol{p}_{\mathrm{p}_j}\right), \tag{E.10}$$

comparing this with Equation (D.13), the direct derivation without substituted pressure loads from Section D.2, we can state the following: (where we substituted $\boldsymbol{\mu}_i$)

$$\boldsymbol{u}_{\mathrm{f}_j}{}^{\top} \frac{1}{P_i}\, \mathbf{H}_{\mathrm{ff}}\, \mathbf{A}_{\mathrm{ff}}{}^{\text{-1}}\left(\mathbf{A}'_{\mathrm{ff}}\, \boldsymbol{p}_{\mathrm{f}_i} + \mathbf{A}'_{\mathrm{fp}}\, \boldsymbol{p}_{\mathrm{p}_i}\right) = \boldsymbol{u}_{\mathrm{f}_j}{}^{\top}\, \boldsymbol{F}'_{\mathrm{f}_i}. \tag{E.11}$$

This comparison shows how the derivative of the load vector can be obtained directly[3]

$$\boldsymbol{F}_{\mathrm{f}_i} = -\,\mathbf{H}_{\mathrm{ff}}\, \boldsymbol{p}_{\mathrm{f}_i} - \mathbf{H}_{\mathrm{fp}}\, \boldsymbol{p}_{\mathrm{p}_i} \quad \Rightarrow \quad \boldsymbol{F}'_{\mathrm{f}_i} = \frac{1}{P_i}\, \mathbf{H}_{\mathrm{ff}}\, \mathbf{A}_{\mathrm{ff}}{}^{\text{-1}}\left(\mathbf{A}'_{\mathrm{ff}}\, \boldsymbol{p}_{\mathrm{f}_i} + \mathbf{A}'_{\mathrm{fp}}\, \boldsymbol{p}_{\mathrm{p}_i}\right). \tag{E.12}$$

E.2. The sensitivities of different objective functions

Because of the superposition theorem of linear problems [24] we can state that the spring deflection is:

$$\Delta_s = \Delta_{\mathrm{out}} - \Delta_{\mathrm{gap}} = \Delta_{21} - c\Delta_{22} - \Delta_{\mathrm{gap}}, \tag{E.13}$$

where c is related to the spring force that is present because of the deformation. c is derived in Appendix D and is the contribution of the second load case in the deformation problem. It is repeated here for convenience:

$$c = \frac{\Delta_{21} - \Delta_{\mathrm{gap}}}{\Delta_{22} + \frac{P_2}{k_s}},$$

$$c' = \frac{\Delta'_{21}\left(\Delta_{22} + \frac{P_2}{k_s}\right) - \left(\Delta_{21} - \Delta_{\mathrm{gap}}\right)\Delta'_{22}}{\left(\Delta_{22} + \frac{P_2}{k_s}\right)^2}, \tag{E.14}$$

In this work the reader can encounter three different objective functions, the first is minimisation of compliance (**MC**):

$$\min_{\boldsymbol{\rho}} \quad C = \boldsymbol{u}^{\top}\, \mathbf{K}\, \boldsymbol{u} = P_1\Delta_{11}, \tag{E.15}$$

the second is maximising the output force (**OF**):

$$M = \frac{F_{\mathrm{out}}}{F_{\mathrm{in}}} = \frac{\Delta_s k_s}{P_1}, \tag{E.16}$$

the third is maximising the Energy Efficiency (**EE**) objective is given by:

$$\psi = \frac{E_{\mathrm{out}}}{E_{\mathrm{in}}} = \frac{\frac{1}{2}(\Delta_s)^2 k_s}{\frac{1}{2}(\Delta_{11} - c\Delta_{12})P_1} = \frac{\mathrm{sgn}(\Delta_s)(\Delta_s)^2 k_s}{(\Delta_{11} - c\Delta_{12})P_1}, \tag{E.17}$$

[3] Obtaining this derivative directly is discouraged as it would be computationally expansive. Advised is to use Gaussian elimination, as is the default in MATLAB when using 'mldivide' or 'A/b', to solve the system of linear equations.

where,

k_s	=	output spring constant ($\mathrm{N\,m^{-1}}$),
$P_1 = p_{\mathrm{in}}$	=	input pressure applied to the system ($\mathrm{N\,m^{-2}}$),
c	=	contribution of the second load case in the system,
Δ_s	=	spring deflection (m),
Δ_{11}, Δ_{12}	=	input energy per applied pressure ($\mathrm{m^3}$),
Δ_{ij}	=	displacement at port i due to load case j (m),
E_{in}	=	strain energy by the pressure load (J),
E_{out}	=	strain energy in the output spring (J).

The derivatives of the three objective functions can now be expressed in terms of Δ'_{ij} for $i, j \in (1,2)$.

The derivative of the **MC** is:

$$\frac{dC}{d\rho_e} = P_1 \Delta'_{11}, \tag{E.18}$$

The derivative of the **OF** is:

$$\frac{dM}{d\rho_e} = \frac{k_s}{P_1}\left(\Delta'_{21} - c'\Delta_{22} - c\Delta'_{22}\right), \tag{E.19}$$

The derivative of the **EE** is:

$$\begin{aligned} \frac{d\psi}{d\rho_e} &= \mathrm{sgn}(\Delta_s)\frac{2\Delta_s\Delta'_s\left(\Delta_{11} - c\Delta_{12}\right) - \Delta_s^2\left(\Delta'_{11} - c'\Delta_{12} - c\Delta'_{12}\right)}{\left(\Delta_{11} - c\Delta_{12}\right)^2}\frac{k_s}{P_1} \\ &= \mathrm{sgn}(\Delta_s)\frac{2\left(\Delta_{21} - c\Delta_{22} - \Delta_{\mathrm{gap}}\right)\left(\Delta'_{21} - c'\Delta_{22} - c\Delta'_{22}\right)\left(\Delta_{11} - c\Delta_{12}\right) - \left(\Delta_{21} - c\Delta_{22} - \Delta_{\mathrm{gap}}\right)^2\left(\Delta'_{11} - c'\Delta_{12} - c\Delta'_{12}\right)}{\left(\Delta_{11} - c\Delta_{12}\right)^2}\frac{k_s}{P_1}, \end{aligned} \tag{E.20}$$

F

Appendix: Sensitivity of the outflow constraint

The outflow constraint, as introduced in Section 5.3, limits the outflow of volume flow out of the domain. The outflow is formulated as follows:

$$
\begin{aligned}
Q_{\text{out}} &= \int_{\Gamma_{p_0}} \boldsymbol{q}(\boldsymbol{x}) \cdot \boldsymbol{n} \, d\Gamma \\
&= \sum_{e=1}^{n_e} \left(\int_{\Gamma_{p_0}} \left(\left(-K(\rho_e) \mathbf{B}_p(\boldsymbol{x}) \boldsymbol{p}^e \right) \cdot \boldsymbol{n}(\boldsymbol{x}) \right) d\Gamma \right), \\
&= \sum_{e=1}^{n_e} \left(-K(\rho_e) \int_{\Gamma_{p_0}} \left(\boldsymbol{n}^\top(\boldsymbol{x}) \, \mathbf{B}_p(\boldsymbol{x}) \right) d\Gamma \boldsymbol{p}^e \right), \\
&= \sum_{e=1}^{n_e} \left(-K(\rho_e) \int_{\Gamma_{p_0}} \left(\boldsymbol{n}^\top(\boldsymbol{x}) \, \mathbf{B}_p(\boldsymbol{x}) \right) d\Gamma \, \mathbf{L}_p^e \right) \boldsymbol{p}, \\
&= \boldsymbol{k}^\top \underbrace{\sum_{e=1}^{n_e} \left(-\boldsymbol{L}^{e\top} \int_{\Gamma_{p_0}} \left(\boldsymbol{n}^\top(\boldsymbol{x}) \, \mathbf{B}_p(\boldsymbol{x}) \right) d\Gamma \, \mathbf{L}_p^e \right)}_{\mathbf{G}} \boldsymbol{p}, \\
&\quad [1 \times n_e] \; \big([n_e \times 1] \; [1 \times d] \; [d \times 4] \; [4 \times n] \big) \; [n \times 1]
\end{aligned}
\tag{F.1}
$$

where,

Q_{out}	=	the volume flow through the boundary Γ_{p_0} $(\mathrm{m^3\,s^{-1}})$,
Γ_{p_0}	=	all boundaries where p is set to zero (so where flow can leave the domain,
$\boldsymbol{n}$	=	surface normal vector $[d \times 1]$,
d	=	number of dimensions considered ($d = 2$ in this work as most examples are in 2D),
n	=	number of nodes in the total domain Ω,
n_e	=	number of elements in the total domain Ω,
$\mathbf{B}_p$	=	matrix of derivatives of shape functions $[d \times 4]$ $(\mathrm{m^{-1}})$,
$\boldsymbol{p}^e$	=	vector of nodal pressures on an element $[4 \times 1]$ $(\mathrm{N\,m^{-2}})$,
$\mathbf{L}_p^e$	=	indexing matrix $[4 \times n]$, that relates the local to the global pressure DOF's,
$\boldsymbol{L}^e$	=	indexing vector with a single non-zero entry of value 1 e-th row $[1 \times n_e]$,
$\boldsymbol{k}$	=	the vector of flow coefficients: $k_e = K(\rho_e)$, using index notation $[n_e \times 1]$ $(\mathrm{m^4\,N^{-1}\,s^{-1}})$,
$\mathbf{G}$	=	global contribution matrix,(m)
$\boldsymbol{p}$	=	global pressure vector $[n \times 1]$ $(\mathrm{N\,m^{-2}})$,

Which can thus be written as a matrix multiplication:

$$
Q_{\text{out}} = \boldsymbol{k}^\top \mathbf{G} \boldsymbol{p}, \tag{F.2}
$$

The integral in Equation (F.1) can be solved using the quadrature rule. The unit normal vector $\boldsymbol{n}(\boldsymbol{\xi})$ has a length of one and in the ξ,η plane it always is in a beneficial, single coordinate, direction (when using a square local element). Because the implementation in this work uses a rectangular mesh and rectangular elements, it is possible to find the analytical expressions for each of the four possible direction and speed up the computations. This way the boundary integral is replaced by the summation of small areas times the flow that goes through each area. The outflow constraint is given by:

$$g_3 = \frac{Q_{\text{out}}}{Q^*_{\text{out}}} - 1. \tag{F.3}$$

The sensitivities of this constraint, $\frac{dg_3}{d\boldsymbol{\rho}}$, can be calculated by rewriting the integral equation into a matrix multiplication on global scale. The matrix $\mathbf{G}$ is introduced in Equation (F.1) as the global contribution matrix that contains the Gauss quadrature weights and abscissae to perform the boundary integration. $\boldsymbol{k}$ is defined as the vector, $k_e = K(\rho_e)$, of flow coefficients using index notation. The outflow is then calculated by left-multiplying $\boldsymbol{k}^\top$ with the global contribution matrix $\mathbf{G}$ and then right-multiplying to the global pressure vector $\boldsymbol{p}$, of which $\mathbf{G}$ is independent of the design variable $\boldsymbol{\rho}$. The multiplication can take the form:

$$Q_{\text{out}} = \boldsymbol{k}(\rho)^\top \mathbf{G}\boldsymbol{p}(\rho), \tag{F.4}$$

The derivative to $\boldsymbol{\rho}$ can now be found using the adjoint method. First the contribution matrix $\mathbf{G}$ and the flow matrix $\mathbf{A}$ are divided in sub matrices that are multiplied to the free pressure DOF's $\mathbf{(...)}_{\text{f}}$ and prescribed pressure DOF's $\mathbf{(...)}_{\text{p}}$:

$$\left(\mathbf{A}\boldsymbol{p} = \boldsymbol{f}\right) \Rightarrow \begin{bmatrix} \mathbf{A}_{\text{ff}} & \mathbf{A}_{\text{fp}} \\ \mathbf{A}_{\text{pf}} & \mathbf{A}_{\text{pp}} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_{\text{f}} \\ \boldsymbol{p}_{\text{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{\text{f}} \\ \boldsymbol{f}_{\text{p}} \end{bmatrix}, \qquad \left(Q_{\text{out}} = \boldsymbol{k}^\top \mathbf{G}\boldsymbol{p}\right) \Rightarrow \boldsymbol{k}^\top \begin{bmatrix} \mathbf{G}_{\text{f}} & \mathbf{G}_{\text{p}} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_{\text{f}} \\ \boldsymbol{p}_{\text{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{\text{f}} \\ \boldsymbol{f}_{\text{p}} \end{bmatrix}. \tag{F.5}$$

Now, the adjoint form (which is the normal form with an added zero term) can be written as:

$$\begin{aligned} Q_{\text{out}} = Q^a_{\text{out}} &= \boldsymbol{k}^\top \mathbf{G}\boldsymbol{p} + \boldsymbol{\mu}^\top \left(\mathbf{A}\boldsymbol{p} - \boldsymbol{f}\right), \\ Q^a_{\text{out}} &= \boldsymbol{k}^\top\, \mathbf{G}_{\text{f}}\, \boldsymbol{p}_{\text{f}} + \boldsymbol{k}^\top\, \mathbf{G}_{\text{p}}\, \boldsymbol{p}_{\text{p}} + \boldsymbol{\mu}_{\text{f}}^\top \left(\mathbf{A}_{\text{ff}}\, \boldsymbol{p}_{\text{f}} + \mathbf{A}_{\text{fp}}\, \boldsymbol{p}_{\text{p}} - \boldsymbol{f}_{\text{f}}\right) + \boldsymbol{\mu}_{\text{p}}^\top \left(\mathbf{A}_{\text{pf}}\, \boldsymbol{p}_{\text{f}} + \mathbf{A}_{\text{pp}}\, \boldsymbol{p}_{\text{p}} - \boldsymbol{f}_{\text{p}}\right). \end{aligned} \tag{F.6}$$

Note that the derivative: $\boldsymbol{p}_{\text{p}}{}' = \frac{\partial\, \boldsymbol{p}_{\text{p}}}{\partial \rho_e} = 0$, because by definition these pressures are prescribed and do not change with the design. In this work, the derivative $\boldsymbol{f}' = 0$. Differentiating Equation (F.6) to ρ_e holds the following:

$$\begin{aligned} Q^a_{\text{out}} =& \boldsymbol{k}'^\top\, \mathbf{G}_{\text{f}}\, \boldsymbol{p}_{\text{f}} + \boldsymbol{k}'^\top\, \mathbf{G}_{\text{p}}\, \boldsymbol{p}_{\text{p}} + \boldsymbol{k}^\top\, \mathbf{G}_{\text{f}}\, \boldsymbol{p}'_{\text{f}} + \boldsymbol{k}^\top\, \mathbf{G}_{\text{p}}\, \boldsymbol{p}_{\text{p}}{}' + \\ & \boldsymbol{\mu}_{\text{f}}^\top \left(\mathbf{A}'_{\text{ff}}\, \boldsymbol{p}_{\text{f}} + \mathbf{A}_{\text{ff}}\, \boldsymbol{p}'_{\text{f}} + \mathbf{A}'_{\text{fp}}\, \boldsymbol{p}_{\text{p}} + \mathbf{A}_{\text{fp}}\, \boldsymbol{p}_{\text{p}}{}' - \boldsymbol{f}'_{\text{f}}\right) + \boldsymbol{\mu}_{\text{p}}^\top \left(\mathbf{A}'_{\text{pf}}\, \boldsymbol{p}_{\text{f}} + \mathbf{A}_{\text{pf}}\, \boldsymbol{p}'_{\text{f}} + \mathbf{A}'_{\text{pp}}\, \boldsymbol{p}_{\text{p}} + \mathbf{A}_{\text{pp}}\, \boldsymbol{p}_{\text{p}}{}' - \boldsymbol{f}_{\text{p}}{}'\right). \end{aligned} \tag{F.7}$$

Eliminating $\boldsymbol{p}_{\text{p}}{}'$ and $\boldsymbol{f}'$ for they are zero:

$$\begin{aligned} Q^a_{\text{out}} =& \boldsymbol{k}'^\top\, \mathbf{G}_{\text{f}}\, \boldsymbol{p}_{\text{f}} + \boldsymbol{k}'^\top\, \mathbf{G}_{\text{p}}\, \boldsymbol{p}_{\text{p}} + \boldsymbol{k}^\top\, \mathbf{G}_{\text{f}}\, \boldsymbol{p}'_{\text{f}} + \\ & \boldsymbol{\mu}_{\text{f}}^\top \left(\mathbf{A}'_{\text{ff}}\, \boldsymbol{p}_{\text{f}} + \mathbf{A}_{\text{ff}}\, \boldsymbol{p}'_{\text{f}} + \mathbf{A}'_{\text{fp}}\, \boldsymbol{p}_{\text{p}}\right) + \boldsymbol{\mu}_{\text{p}}^\top \left(\mathbf{A}'_{\text{pf}}\, \boldsymbol{p}_{\text{f}} + \mathbf{A}_{\text{pf}}\, \boldsymbol{p}'_{\text{f}} + \mathbf{A}'_{\text{pp}}\, \boldsymbol{p}_{\text{p}}\right). \end{aligned} \tag{F.8}$$

Now, collecting terms, holds:

$$\begin{aligned} Q^a_{\text{out}} =& \left(\boldsymbol{k}'^\top\, \mathbf{G}_{\text{f}} + \boldsymbol{\mu}_{\text{f}}^\top\, \mathbf{A}'_{\text{ff}} + \boldsymbol{\mu}_{\text{p}}^\top\, \mathbf{A}'_{\text{pf}}\right) \boldsymbol{p}_{\text{f}} + \\ & \left(\boldsymbol{k}'^\top\, \mathbf{G}_{\text{p}} + \boldsymbol{\mu}_{\text{f}}^\top\, \mathbf{A}'_{\text{fp}} + \boldsymbol{\mu}_{\text{p}}^\top\, \mathbf{A}'_{\text{pp}}\right) \boldsymbol{p}_{\text{p}} + \\ & \left(\boldsymbol{k}^\top\, \mathbf{G}_{\text{f}} + \boldsymbol{\mu}_{\text{f}}^\top\, \mathbf{A}_{\text{ff}} + \boldsymbol{\mu}_{\text{p}}^\top\, \mathbf{A}_{\text{pf}}\right) \boldsymbol{p}'_{\text{f}}. \end{aligned} \tag{F.9}$$

At this point it is clear that only one of the two Lagrange multipliers is necessary to eliminate the last term between brackets. Notice that $\boldsymbol{\mu}_{\text{p}}^\top$ is multiplied with $\mathbf{A}_{\text{pf}}$, which, in general, is not a square matrix. Therefore, $\boldsymbol{\mu}_{\text{p}}^\top = \mathbf{0}$ and the solving $\boldsymbol{\mu}_{\text{f}}$ is done by solving the following system of equations: $\boldsymbol{\mu}_{\text{f}}^\top\, \mathbf{A}_{\text{ff}} = -\boldsymbol{k}^\top\, \mathbf{G}_{\text{f}}$. The sensitivities now take the form:

$$\frac{\partial g_3}{\partial \rho_e} = \frac{1}{Q^*_{\text{out}}} \left(\left(\boldsymbol{k}'^\top\, \mathbf{G}_{\text{f}} + \boldsymbol{\mu}_{\text{f}}^\top\, \mathbf{A}'_{\text{ff}} + \boldsymbol{\mu}_{\text{p}}^\top\, \mathbf{A}'_{\text{pf}}\right) \boldsymbol{p}_{\text{f}} + \left(\boldsymbol{k}'^\top\, \mathbf{G}_{\text{p}} + \boldsymbol{\mu}_{\text{f}}^\top\, \mathbf{A}'_{\text{fp}} + \boldsymbol{\mu}_{\text{p}}^\top\, \mathbf{A}'_{\text{pp}}\right) \boldsymbol{p}_{\text{p}}\right), \tag{F.10}$$

in which $\boldsymbol{\mu}_{\text{f}}^\top$ is solved from:

$$\boldsymbol{\mu}_{\text{f}}^\top\, \mathbf{A}_{\text{ff}} = -\boldsymbol{k}^\top\, \mathbf{G}_{\text{f}}.$$

A finite difference analysis confirmed these sensitivities.

G

Appendix: Sensitivity derivation of the input load vector

The sensitivity of the input load vector F_{in} is derived in this appendix in an effort to proof that the computational costs are high of calculating the sensitivity of each component of $\boldsymbol{F}$. $\boldsymbol{F}$ is the consistent global force vector that contains the pressure load contributions, in this work it is referred to as: $\boldsymbol{F}_1$. First, remember that the shape of a vector derived to another vector is a matrix of derivatives:

$$\frac{\partial F_i}{\partial \rho_e} = P_{ie} \quad \mathbf{P} = \begin{bmatrix} \frac{\partial F_1}{\partial \rho_1} & \frac{\partial F_1}{\partial \rho_2} & \cdots & \frac{\partial F_1}{\partial \rho_{ne}} \\ \frac{\partial F_2}{\partial \rho_1} & \frac{\partial F_2}{\partial \rho_2} & \cdots & \frac{\partial F_2}{\partial \rho_{ne}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_{(nd)}}{\partial \rho_1} & \frac{\partial F_{(nd)}}{\partial \rho_2} & \cdots & \frac{\partial F_{(nd)}}{\partial \rho_{ne}} \end{bmatrix}. \tag{G.1}$$

Therefore, this appendix will use the index notation to aid the reader in following the derivation. The matrix is divided into sub matrices that are multiplied to the free pressure DOF's $\mathbf{(...)}^{\text{f}}$ and prescribed pressure DOF's $\mathbf{(...)}^{\text{p}}$ (where the subscript is reserved for the index notation):

$$\begin{aligned} &(\mathbf{A}\boldsymbol{p} = \boldsymbol{f}) \Rightarrow \begin{bmatrix} \mathbf{A}^{\text{ff}} & \mathbf{A}^{\text{fp}} \\ \mathbf{A}^{\text{pf}} & \mathbf{A}^{\text{pp}} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^{\text{f}} \\ \boldsymbol{p}^{\text{p}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{\text{f}} \\ \boldsymbol{f}^{\text{p}} \end{bmatrix}, \qquad (\boldsymbol{F} = -\mathbf{H}\boldsymbol{p}) \Rightarrow \begin{bmatrix} \boldsymbol{F}^{\text{f}} \\ \boldsymbol{F}^{\text{p}} \end{bmatrix} = -\begin{bmatrix} \mathbf{H}^{\text{ff}} & \mathbf{H}^{\text{fp}} \\ \mathbf{H}^{\text{pf}} & \mathbf{H}^{\text{pp}} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^{\text{f}} \\ \boldsymbol{p}^{\text{p}} \end{bmatrix}, \\ &\text{such that:} \quad \mathbf{A}^{\text{ff}}\,\boldsymbol{p}^{\text{f}} + \mathbf{A}^{\text{fp}}\,\boldsymbol{p}^{\text{p}} = \boldsymbol{f}^{\text{f}}, \qquad \text{and such that:} \quad \boldsymbol{F}^{\text{f}} = -\mathbf{H}^{\text{ff}}\,\boldsymbol{p}^{\text{f}} - \mathbf{H}^{\text{fp}}\,\boldsymbol{p}^{\text{p}}, \end{aligned} \tag{G.2}$$

Now the relation of F_{in}, as mentioned in Section 4.1, could be chosen to be:

$$\begin{aligned} F_{\text{in}} &= \sum_{k=1}^{n} \sqrt{F_{2k-1}^2 + F_{2k}^2} && \text{for } d = 2, \\ F_{\text{in}} &= \sum_{k=1}^{n} \sqrt{F_{3k-2}^2 + F_{3k-1}^2 + F_{3k}^2} && \text{for } d = 3, \end{aligned} \tag{G.3}$$

where F_i, $\forall i \in \mathbb{D}_u$, are components of $\boldsymbol{F}$. Deriving Equation (G.3) (for $d = 3$) to each component ρ_e in the vector of design variables $\boldsymbol{\rho}$ holds:

$$\frac{\partial F_{\text{in}}}{\partial \rho_e} = \frac{\partial}{\partial \rho_e} \sum_{k=1}^{n} \sqrt{F_{3k-2}^2 + F_{3k-1}^2 + F_{3k}^2}, \tag{G.4}$$

which is composed of:

$$\begin{aligned} F_{\text{in}}(U_k) &= \sum_{k=1}^{n} \sqrt{U_k}, \\ U_k(F_i) &= F_{3k-2}^2 + F_{3k-1}^2 + F_{3k}^2, \\ F_i(\rho_e) = F_i(\rho_e)^a &= -\,\mathbf{H}_{ij}^{\text{ff}}\,\boldsymbol{p}_j^{\text{f}} - \mathbf{H}_{ij}^{\text{fp}}\,\boldsymbol{p}^{\text{p}}{}_j + \zeta_{is}\left(\mathbf{A}_{sj}^{\text{ff}}\,\boldsymbol{p}_j^{\text{f}} + \mathbf{A}_{sj}^{\text{fp}}\,\boldsymbol{p}^{\text{p}}{}_j + \boldsymbol{f}_s^{\text{f}}\right), \end{aligned} \tag{G.5}$$

where the index notation is started to be used and an adjoint term is added to differentiate the implicit dependence of $\boldsymbol{p}^{\mathrm{f}}_j$ to the design variable ρ_e. The chain rule can now be used to get:

$$\begin{aligned}
\frac{dF_{\text{in}}}{d\rho_e} &= \frac{dF_{\text{in}}}{dU_k}\frac{dU_k}{dF_i}\frac{dF_i}{d\rho_e}, \\
\frac{dF_{\text{in}}}{dU_k} &= \sum_{k=1}^{n}\frac{1}{2\sqrt{U_k}}, \\
\frac{dU_k}{dF_i} &= 2F_{3k-2} + 2F_{3k-1} + 2F_{3k},
\end{aligned} \tag{G.6}$$

such that:

$$\frac{dF_{\text{in}}}{d\rho_e} = \sum_{k=1}^{n}\frac{\left(F_{3k-2} + F_{3k-1} + F_{3k}\right)}{\sqrt{F_{3k-2}^2 + F_{3k-1}^2 + F_{3k}^2}}F_{i,\rho},$$

where $\frac{\partial F_i}{\partial \rho_e} \equiv F_{i,\rho}$. The derivative $F_{i,\rho}$ is a bit more difficult to derive. The adjoint equations are already introduced in the function $F_i^a(\rho_e)$ in Equation (G.5), these are necessary to solve the sensitivities. Note that the matrix $\mathbf{H}$ is independent of the design parameter and that the derivative: $\boldsymbol{p}^{\mathrm{p}}{}_{,\rho} = \frac{\partial \boldsymbol{p}^{\mathrm{p}}}{\partial \rho_e} = 0$, because by definition these pressures are prescribed and do not change with changes in the design. And in this work, the derivative $\boldsymbol{f}_{,\rho} = 0$.

$$\begin{aligned}
\frac{dF_i^a}{d\rho_e} &= -\,\mathbf{H}^{\mathrm{ff}}_{ij}\,\boldsymbol{p}^{\mathrm{f}}_{j,\rho} - \mathbf{H}^{\mathrm{fp}}_{ij}\,\boldsymbol{p}^{\mathrm{p}}{}_{j,\rho} + \zeta_{is}\left(\mathbf{A}^{\mathrm{ff}}_{sj,\rho}\,\boldsymbol{p}^{\mathrm{f}}_j + \mathbf{A}^{\mathrm{fp}}_{sj,\rho}\,\boldsymbol{p}^{\mathrm{p}}{}_j + \mathbf{A}^{\mathrm{ff}}_{sj}\,\boldsymbol{p}^{\mathrm{f}}_{j,\rho} + \mathbf{A}^{\mathrm{fp}}_{sj}\,\boldsymbol{p}^{\mathrm{p}}{}_{j,\rho} + \boldsymbol{f}^{\mathrm{f}}_{s,\rho}\right), \\
&= -\,\mathbf{H}^{\mathrm{ff}}_{ij}\,\boldsymbol{p}^{\mathrm{f}}_{j,\rho} + \zeta_{is}\left(\mathbf{A}^{\mathrm{ff}}_{sj,\rho}\,\boldsymbol{p}^{\mathrm{f}}_j + \mathbf{A}^{\mathrm{fp}}_{sj,\rho}\,\boldsymbol{p}^{\mathrm{p}}{}_j + \mathbf{A}^{\mathrm{ff}}_{sj}\,\boldsymbol{p}^{\mathrm{f}}_{j,\rho}\right), \\
&= \zeta_{is}\left(\mathbf{A}^{\mathrm{ff}}_{sj,\rho}\,\boldsymbol{p}^{\mathrm{f}}_j + \mathbf{A}^{\mathrm{fp}}_{sj,\rho}\,\boldsymbol{p}^{\mathrm{p}}{}_j\right) + \left(\zeta_{is}\,\mathbf{A}^{\mathrm{ff}}_{sj} - \mathbf{H}^{\mathrm{ff}}_{ij}\right)\boldsymbol{p}^{\mathrm{f}}_{j,\rho}.
\end{aligned} \tag{G.7}$$

Note that from the beginning the Lagrange multiplier had to be chosen as a matrix, ζ_{is} (with two indices), in order to be able to left-multiply it to the adjoint function (that equates to zero) and be added in the original equation, Equation (G.2). To solve the Lagrange multiplier to find $F_{i,\rho}$, the system of equations is to be solved:

$$\zeta_{is}\,\mathbf{A}^{\mathrm{ff}}_{sj} = \mathbf{H}^{\mathrm{ff}}_{ij}, \tag{G.8}$$

which are nd solutions of linear sets of equations of approximately the size $[n \times n]$. So, in summery, in order to find the derivative (or sensitivity) of Equation (G.3) the derivative of each individual force vector component $F_{i,\rho}$ is required and that is computationally expensive because it requires approximately nd solutions of algebraic systems of the same size as the flow matrix.

Bibliography

[1] Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan S. Lazarov, and Ole Sigmund. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43(1):1–16, 2011. ISSN 1615147X. doi: 10.1007/s00158-010-0594-7.

[2] Martin Philip Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, 1989. ISSN 09344373. doi: 10.1007/BF01650949.

[3] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988. ISSN 00457825. doi: 10.1016/0045-7825(88)90086-2.

[4] Martin Philip Bendsøe and Ole Sigmund. *Topology Optimization, Theory, Methods and Applications*. Springer, 2003. ISBN 3540429921.

[5] J. Blaauwendraad and A.W.M. Kok. *Elementenmethode voor constructeurs*. Agon Elsevier, Amsterdam/Brussel, 1973. ISBN 90 10 10439 7.

[6] Matteo Bruggi and Carlo Cinquini. An alternative truly-mixed formulation to solve pressure load problems in topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20): 1500–1512, apr 2009. ISSN 00457825. doi: 10.1016/j.cma.2008.12.009. URL `http://dx.doi.org/10.1016/j.cma.2008.12.009`.

[7] Matteo Bruggi and Paolo Venini. Topology optimization of incompressible media using mixed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3151–3164, jul 2007. ISSN 00457825. doi: 10.1016/j.cma.2007.02.013. URL `http://linkinghub.elsevier.com/retrieve/pii/S0045782507001247`.

[8] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert E. Witt. *Concepts and Application of Finite Element Analysis*. John Wiley & Sons, Inc., 4th edition, 2001. ISBN 978-0-471-35605-9.

[9] Sangamesh R. Deepak, M. Dinesh, Deepak K. Sahu, and G. K. Ananthasuresh. A Comparative Study of the Formulations and Benchmark Problems for the Topology Optimization of Compliant Mechanisms. *Journal of Mechanisms and Robotics*, 1(1):011003, 2009. ISSN 19424302. doi: 10.1115/1.2959094. URL `http://mechanismsrobotics.asmedigitalcollection.asme.org/article.aspx?articleid=1484848`.

[10] J. Du and N. Olhoff. Topological optimization of continuum structures with design-dependent surface loading - Part I: New computational approach for 2D problems. *Structural and Multidisciplinary Optimization*, 27(3):151–165, 2004. ISSN 1615147X. doi: 10.1007/s00158-004-0379-y.

[11] M. B. Fuchs and N. N. Y. Shemesh. Density-based topological design of structures subjected to water pressure using a parametric loading surface. *Structural and Multidisciplinary Optimization*, 28(1):11–19, 2004. ISSN 1615147X. doi: 10.1007/s00158-004-0406-z.

[12] Michael Gosz. *Finite Element Method: Applications in Solids, Structures, and Heat Transfer*. Taylor & Francis Group, 2006. ISBN 9780849334078.

[13] V. B. Hammer. Checkmate? Nodal densities in topology optimization. In *Proc. II Max Planck workshop on engineering design optimization*. Dept. of Mathematics, DTU Denmark, 2001.

[14] V. B. Hammer and N. Olhoff. Topology optimization of continuum structures subjected to pressure loading. *Structural and Multidisciplinary Optimization*, 19(2):85–92, 2000. ISSN 1615147X. doi: 10.1007/s001580050088.

[15] Larry L Howell. *Compliant Mechanisms*. John Wiley & Sons, Inc., New York, 2002. ISBN 9780471384786.

[16] Gang Won Jang and Yoon Young Kim. Topology optimization with displacement-based nonconforming finite elements for incompressible materials. *Journal of Mechanical Science and Technology*, 23(2):442–451, 2009. ISSN 1738494X. doi: 10.1007/s12206-008-1114-1.

[17] van J. Kan, A. Segal, and F. Vermolen. *Numerical Methods in Scientific Computing*. Delft Academic Press, Delft, 2nd edition, 2014. ISBN 97890-6562-3638.

[18] B. S. Lazarov and O. Sigmund. Filters in topology optimization based on Helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86(6):765–781, may 2011. ISSN 00295981. doi: 10.1002/nme.3072. URL `http://doi.wiley.com/10.1002/nme.3072`.

[19] Edmund Lee and Joaquim R R A Martins. Structural topology optimization with design-dependent pressure loads. *Computer Methods in Applied Mechanics and Engineering*, 233-236:40–48, 2012.

[20] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D. Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 19(12):3243–3254, 2010. ISSN 10577149. doi: 10.1109/TIP.2010.2069690.

[21] A. G. M. Michell. The limits of economy of material in frame-structures. *Philosophical Magazine Series 6*, 8(47):589–597, 1904. ISSN 1941-5982. doi: 10.1080/14786440409463229. URL `http://www.tandfonline.com/doi/abs/10.1080/14786440409463229`.

[22] Henry Panganiban, Gang Won Jang, and Tae Jin Chung. Topology optimization of pressure-actuated compliant mechanisms. *Finite Elements in Analysis and Design*, 46(3):238–246, 2010. ISSN 0168874X. doi: 10.1016/j.finel.2009.09.005. URL `http://dx.doi.org/10.1016/j.finel.2009.09.005`.

[23] Henry Panganiban, Gang Won Jang, and Tae Jin Chung. Topology optimization of pressure-actuated compliant mechanisms. *Finite Elements in Analysis and Design*, 46(3):238–246, 2010. ISSN 0168874X. doi: 10.1016/j.finel.2009.09.005. URL `http://dx.doi.org/10.1016/j.finel.2009.09.005`.

[24] Ole Sigmund. On the Design of Compliant Mechanisms Using Topology Optimization. *Mechanics Based Design of Structures and Machines*, 25(4):493–524, 1997. ISSN 0890-5452. doi: 10.1080/08905459708945415.

[25] Ole Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001. ISSN 1615147X. doi: 10.1007/s001580050176.

[26] Ole Sigmund and P. M. Clausen. Topology optimization using a mixed formulation: An alternative way to solve pressure load problems. *Computer Methods in Applied Mechanics and Engineering*, 196(13-16):1874–1889, 2007. ISSN 00457825. doi: 10.1016/j.cma.2006.09.021.

[27] Katsuyuki Suzuki and Noboru Kikuchi. A homogenization method for shape and topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 93(3):291–318, dec 1991. ISSN 00457825. doi: 10.1016/0045-7825(91)90245-2. URL `http://linkinghub.elsevier.com/retrieve/pii/0045782591902452`.

[28] K. Svanberg. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987. ISSN 00295981. doi: 10.1002/nme.1620240207. URL `http://onlinelibrary.wiley.com/doi/10.1002/nme.1620240207/abstract`.

[29] Liyong Tong and Jiangzi Lin. Structural topology optimization with implicit design variable—optimality and algorithm. *Finite Elements in Analysis and Design*, 47(8):922–932, aug 2011. ISSN 0168874X. doi: 10.1016/j.finel.2011.03.004. URL `http://linkinghub.elsevier.com/retrieve/pii/S0168874X11000473`.

[30] Daniel A. Tortorelli and P Michaleris. Design sensitivity analysis: Overview and review. *Inverse Problems in Engineering*, 1(1):71–105, 1994. ISSN 10290281. doi: 10.1080/174159794088027573. URL `http://www.informaworld.com/10.1080/174159794088027573`.

[31] Srinivas Vasista and Liyong Tong. Design and Testing of Pressurized Cellular Planar Morphing Structures. *AIAA Journal*, 50(6):1328–1338, 2012. ISSN 0001-1452. doi: 10.2514/1.J051427.

[32] Cunfu Wang, Min Zhao, and Tong Ge. Structural topology optimization with design-dependent pressure loads. *Structural and Multidisciplinary Optimization*, 53(5):1005–1018, may 2016. ISSN 1615-147X. doi: 10.1007/s00158-015-1376-z. URL `http://dx.doi.org/10.1007/s00158-015-1376-z`.

[33] Hui Zhang, Xiong Zhang, and Shu-Tian Liu. A new boundary search scheme for topology optimization of continuum structures with design-dependent loads. *Structural and Multidisciplinary Optimization*, 37(2):121–129, 2008. ISSN 1615147X. doi: 10.1007/s00158-007-0221-4.

[34] Bin Zheng, Ching Jui Chang, and Hae Chang Gea. Topology optimization with design-dependent pressure loading. *Structural and Multidisciplinary Optimization*, 38(6):535–543, 2009. ISSN 1615147X. doi: 10.1007/s00158-008-0317-5.

[35] O.C. Zienkiewicz and Robert Leroy Taylor. *The finite element method*. Butterworth Heinemann, Oxford, 5 edition, 2000. ISBN 0750650559.