Eindhoven University of Technology

MASTER

Multi-Scale Material Modelling
Application in Topology Optimization of Structures

Bol, Rowin J.M.

Award date:
2022

MASTER THESIS

Multi-Scale Material Modelling: Application in Topology Optimization of Structures

Rowin Johannes Marinus Bol

1005069

Eindhoven University of Technology

Department of the Built Environment

Master Track Architecture, Building & Planning

Unit Structural Engineering & Design

Chair of Applied Mechanics

Supervisors: dr. P. Poorsolhjouy
dr.ir. H. Hofmeyer
prof.dr.ir. A.S.J. Suiker

May 2022

Acknowledgements

In front of you is the final result of the graduation project ”Multi-Scale Material Modelling: Application in Topology Optimization of Structures”. This research has been carried out within the framework of the master track Structural Engineering & Design of the master program Architecture, Building & Planning in the Unit Structural Engineering & Design at the Chair of Applied Mechanics of the Department of the Built Environment at the Eindhoven University of Technology, Eindhoven, the Netherlands. The research has been performed between November 2020 and May 2022.

Simultaneously with the start of my graduation at the 26th of November 2020, a large tumor was found inside the spinal cord in the neck of my girlfriend, Renske Pertijs. On the 20th of January 2021, the tumor was successfully removed in a high-risk surgery which took over 13 hours - the longest day of my life. Afterwards it became clear that Renske had gotten a high, incomplete spinal cord injury (C4-5) as a result of the surgery. During the first couple of months, she was unable to move any part of her body, except for her face. Therefore, I slept at two different hospitals for about a month before she was transferred to Rotterdam for a very intensive rehabilitation process. After approximately half a year of very hard work, she had significantly improved and was discharged from the rehabilitation center. At that time, she was able to walk a few hundred meters with crutches whilst still being wheelchair bound, but she had largely regained her independence. Unfortunately, about a year after the discovery of the tumor, Renske fell backwards with her wheelchair causing additional loss of functioning in the right side of her body. To this day, the question remains how further recovery (if any) will proceed.

I would like to sincerely thank my graduation committee, dr. P. Poorsolhjouy, dr.ir. H. Hofmeyer and prof.dr.ir. A.S.J. Suiker for their guidance during the entire course of my graduation. Without their input, suggestions and knowledge in their respective fields of expertise, this research would not have been possible. In view of my personal circumstances, I am extremely thankful for their understanding and flexibility with respect to my graduation. I would like to give special thanks to my first supervisor, dr. P. Poorsolhjouy, for his involvement and thorough understanding of my situation as well as his confidence in me. In difficult times he was always there to support me. I am incredibly grateful for everything he has done for me. I would also like to thank the department's academic advisor, J.A.M. Pulles, for her support during these difficult times.

Finally, I would like to thank my family and close friends for their unconditional support throughout this period, especially my mother, Hélène. A special thanks goes to my girlfriend, Renske, who has always been my biggest supporter despite her incredibly difficult medical circumstances.

I wish you a pleasant reading,

R.J.M. Bol

Eindhoven, May 2022

Abstract

Topology optimization is a conceptual design tool which aims to find the most optimal material distribution of a structure within a given space. Therefore, this tool may be used to attain structural designs with higher stiffness or strength-to-weight ratios by avoiding the presence of redundant material. A vast majority of topology optimization methods is handled through a Finite Element Method (FEM) based analysis of a structure, followed by optimizing a certain property of that structure for a given constraint. Two popular topology optimization methodologies have been considered here, which are the compliance minimization problem for a given volume constraint and the stress-constrained volume minimization problem under a certain stress constraint. The focus of this research is on topology optimization of cementitious materials, such as 3D Concrete Printing (3DCP), to incorporate their specific properties and evolution of microstructure during construction and loading. To this end, the main objective has been to see the effects of non-linear and anisotropic behaviour of these cement-based materials on their optimized topologies. In order to implement a multi-scale material model to do so, a non-linear FEM was required that has been developed using the Newton-Raphson iterative method. The Granular Micromechanics Approach (GMA) has enabled incorporation of the asymmetric nature of particle interactions into the continuum model in a computationally affordable manner through which the model is capable of capturing the anisotropic evolution of a material's microstructure. The Proportional Topology Optimization (PTO) method has been used to perform two-dimensional as well as three-dimensional optimizations. The resulting topology optimization algorithms have been used to study the effect of asymmetric stiffness and strength on optimized topology. The obtained results for the compliance minimization problem clearly demonstrate the effects of asymmetric stiffness on the optimized topologies. Generally, tensile regions gain a larger size compared to compressive regions, because lower stiffness (tensile) regions require more cross-sectional area to transfer applied loads and satisfy static equilibrium. In the absence of bending mechanisms, the optimized topology will provide compression-only structures for a lower tensile stiffness. On the other hand, the results for the stress-constrained problem clearly demonstrate that the PTO method obtains less optimal topologies under asymmetric strength. Namely, the method does provide tensile results as opposed to the expected compression-dominant structures. As a result, the tensile topologies do require a larger structural volume in order to prevent exceedance of the material's tensile stress limit, which makes these structures less optimal in terms of volume minimization. Therefore, it is recommended to use the Traditional Topology Optimization (TTO) method for stress-constrained volume minimization under asymmetric strength. Nevertheless, several improvements to topology optimization have been achieved. First of all, the implementation of a non-linear FEM using the GMA enables to capture the anisotropic evolution of a material's microstructure, thus providing a more realistic macroscopic material behaviour. Furthermore, strength-related topology optimization is able to fully exploit the GMA, because it is able to describe the macroscopic material behaviour up to failure in a non-linear framework and to predict failure points to obtain failure envelopes as a result of natural failure. This means that failure is identified based on the actual material behaviour as opposed to (classical) failure criteria where failure is imposed at predefined stress levels.

Nomenclature

Abbreviations

3DCP	3D Concrete Printing
AM	Additive Manufacturing
ASTM	American Society for Testing Materials
CAD	Computer-Aided Design
DOF	Degree Of Freedom
ESO	Evolutionary Structure Optimization
FDP	Failure Data Point
FEA	Finite Element Analysis
FEM	Finite Element Method
FE	Finite Element
GMA	Granular Micromechanics Approach
MBB	Messerschmitt-Bölkow-Blohm
OC	Optimality Criteria
PTO	Proportional Topology Optimization
PVW	Principle of Virtual Work
SIMP	Solid Isotropic Material with Penalization
TTO	Traditional Topology Optimization
TU/e	Eindhoven University of Technology

Greek Upper Case Symbols

Ω_e	Domain of element e

Greek Lower Case Symbols

α	History coefficient
$\bar{\boldsymbol{\sigma}}$	Macroscopic stress related to deformations (i.e. strain)
β	Inter-granular shear to normal stiffness ratio
$\boldsymbol{\sigma}$	Macroscopic stress tensor
$\boldsymbol{\sigma}^{lim}$	Maximum allowable stress or stress limit

δ_i	Relative displacement
$\delta_y^{i\alpha}$	Relative displacement of α^{th} contact at yielding
δ_{ij}	Kronecker delta
$\dot{\sigma}_{ij}$	Rate of change in macroscopic stress
η	Accuracy of evaluation
γ	Strength ratio
λ	$k_{\mathrm{n}}^{T\alpha}/k_{\mathrm{n}}^{C\alpha}$ ratio
λ	Scalar quantity for controlling load incrementation
λ_1	Yield point or $\delta_y^{i\alpha}/\delta_y^{C\alpha}$ ratio
λ_2	Hardening or $k_i^{h\alpha}/k_i^{0\alpha}$ ratio
λ_{min}	Minimal eigenvalue
ν	Poisson's ratio
ϕ	Azimuth angle
ρ	Volumetric number density of contacts in a material point
σ_{ij}	Macroscopic (Cauchy) stress tensor
θ	Polar angle
$\tilde{\varepsilon}_{ij}$	Penalized strain
ε	Relaxation coefficient
ε_{ij}	Macroscopic strain tensor
ξ	Directional density distribution function

Latin Upper Case Symbols

$\mathbf{J_e}$	Jacobian of transformation matrix of element e
$\mathbf{B}$	Strain-displacement matrix
$\mathbf{C}$	Material tangent operator or tangential stiffness tensor
$\mathbf{D}$	Tangent modulus or tangential stiffness tensor for two-dimensional problem
$\mathbf{E}_m$	In-plane strain tensor
$\mathbf{E}_z$	Out-of-plane strain tensor
$\mathbf{F}$	External force vector

$\mathbf{I}_1$	First invariant of stress tensor
$\mathbf{I}_2$	Second deviatoric stress invariant
$\mathbf{K}$	Stiffness matrix or Jacobian
$\mathbf{K}_e$	Stiffness matrix of element e
$\mathbf{R}$	Out-of-balance forces or residual
$\mathbf{S}_m$	In-plane stress tensor
$\mathbf{S}_z$	Out-of-plane stress tensor
A	Cross-sectional area
c	Compliance or strain energy
C_{ijkl}	Macroscopic tangential stiffness tensor for a material point
C_{ij}	Macroscopic stiffness tensor
C_{ij}^0	Macroscopic stiffness tensor for solid material
C_{ij}^F	Macroscopic stiffness tensor in failed state
E	Young's modulus
E_0	Young's modulus of solid material
E_{min}	Minimal Young's modulus
F	Externally applied force
H	Filtering weight
K_i	Average inter-granular stiffness
K_{ij}^α	Inter-granular stiffness tensor of α^{th} contact in the global coordinate system
L	Element length
N^ρ	Total number of contacts for a given solid angle
N_c	Total number of contacts in a material point
N_e	Total number of elements
N_i	Total number of (load) increments
N_p	Number density of contacts or number of contacts in a unit volume of material
N_p^0	Number density of contacts for solid material

R_e	Stress constraint function of element e
T_{ij}	Rotation tensor
U	Strain energy
V	Total structural volume
V	Volume
V_0	Volume of entire design domain
W	Macroscopic strain energy density
W	Total work
W^α	Microscopic internal strain energy of α^{th} contact
W_p	Weighting factors

Latin Lower Case Symbols

$\mathbf{e}_j$	jth unit vector
$\mathbf{f}$	Internal force vector
$\mathbf{u}$	Displacement vector
$\mathbf{x}$	Vector containing all element densities
$\mathbf{x}_p^j$	Perturbed elemental density
$\tilde{x}_e$	Filtered physical density of element e
b	Element width
f	Volume fraction
f_c	Uniaxial compressive strength
f_i^α	Inter-granular force component of α^{th} contact
f_{cy}	Compressive strength
f_{ty}	Tensile strength
h	Element height
h	Perturbation parameter
k	Stiffness measure
k_i	Spring stiffness
k_i^α	Microscopic or inter-granular stiffness coefficient of α^{th} contact

k_{pq}	Diagonal inter-granular stiffness tensor in a grain's local coordinate system
l	Average branch length or grain size
l	Distance between the centroids of two neighboring grains
l_j^α	Inter-granular branch vector joining the centroids of α^{th} grain-pair
m_c	Current material amount
m_r	Remaining material amount
m_t	Target material amount
n_i	Unit vector in the direction of l_i (normal direction)
n_p	Gauss quadrature points
p	Penalization power
q	Proportion exponent
r_0	Filter radius
r_{ij}	Euclidean distance between the centers of to elements
s_i	Unit vector in the direction orthogonal to l_i (tangential direction)
t	Element thickness
t_i	Unit vector in the direction orthogonal to l_i (tangential direction)
u	Strain energy density
u_i	Particle displacement
$u_{i,j}$	Displacement gradient
v_e	Volume of element e
x_e	Density or design variable of element e
x_e^{new}	New density of element e for next iteration
x_e^{opt}	Optimized density of element e
x_e^{prev}	Density of element e from previous iteration
x_j	Position vector of grain centroid
x_{min}	Minimal density value

Other Symbols

$\mathbb{F}$	General yield function or failure criterion

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Topology optimization is a conceptual design tool which aims to find the most optimal material distribution of a structure within a given space [15, 19, 20]. Therefore, this tool may be used to attain structural designs with higher stiffness or strength-to-weight ratios [21, 22], for example, by avoiding the presence of redundant structural elements. Subsequently, topology optimization methods could be beneficial for the construction industry in its process to design lighter and less resource intensive structures [5, 23].

A vast majority of topology optimization methods is handled through a Finite Element Method (FEM) based analysis of a structure, followed by optimizing a certain property of that structure for a given constraint. Two popular topology optimization methodologies are the minimum compliance problem, which aims to minimize structural compliance for a given volume constraint, and the stress-constrained problem, which aims to minimize the structural volume while satisfying a certain stress constraint [21, 22].

Recently popular Additive Manufacturing (AM) techniques, such as 3D printing, appreciate the importance of topology optimization since it can potentially provide the ability to manufacture (porous) structural designs having complicated geometries [9, 21]. According to the American Society for Testing Materials (ASTM) [24], AM is defined as a "process of joining materials to make object from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies". This layer-by-layer manufacturing process allows for direct fabrication from CAD models such that AM techniques can build structures with virtually any shape. On that score, 3D Concrete Printing (3DCP) is rapidly gaining interest as a possible alternative construction method to traditional cast-in-situ concrete. Moreover, using multiple materials is feasible within AM. This could result in multi-functional structural designs having load bearing 3DCP at highly stressed regions and materials with low structural performance but other (e.g. thermal) properties in locations under low stress [25]. To this end, the focus of the present research is on topology optimization of cementitious materials (such as 3DCP) to incorporate their specific properties and evolution of microstructure during construction and loading.

Cementitious materials exhibit several interesting behavioral features that make modelling their behaviour difficult from a macroscopic continuum point of view. To derive the macroscopic response of these materials, the behaviour of particle interactions must be included: the normal interactions between particles are different under tension and compression whilst their tangential interactions also depend on the normal interaction. Therefore, the material's microstructure will evolve anisotropic: contacts in compression become stronger, while contacts in tension become weaker. As a result, the material will exhibit anisotropic behaviour when loaded, which is called *induced anisotropy* [14].

Various approaches at different length scales are available for incorporating the effects of a material's microstructure into the macroscopic behaviour. Methods in the smallest length scales having a discrete nature yield results with a high degree of exactness, but these are limited in their applicability to large-scale problems due to their prohibitively large computational demand. In the larger length scales continuum modelling suffers from losing characteristic features due to neglecting the effect of particle interactions, while being computationally affordable [1]. The Granular Micromechanics Approach (GMA) can be classified as a meso-scale approach that bridges the gap between continuum and discrete models [1, 26]. This multi-scale material model enables incorporation of the asymmetric nature of particle interactions into the continuum model [14] in a computationally affordable manner [1]. Thereby, GMA makes it possible to incorporate information from the microstructure into the material's macroscopic behaviour. Accordingly, the model is capable of capturing the anisotropic response of materials [2].

The added value of the GMA to topology optimization is that it enables derivation of optimized topologies in a manner that is in accordance with the actual behaviour of cementitious (and other granular) materials. Therefore, a more realistic macroscopic response is taken into account during the optimization process. Besides, there is yet another advantage specifically related to stress-constrained volume minimization. Using the multi-scale material model, the tangential stiffness tensor of the material is fully known at any time during the loading process. This stiffness tensor is consistent with the behaviour of particle interactions that form the material. Accordingly, material failure can be defined in a manner that is consistent with the behaviour of the material, in contrast to using arbitrary failure criteria for strength analysis. As a result, one can *obtain* failure envelopes that are consistent with the material behaviour, rather than using *predefined* failure envelopes. This allows strength-related optimizations to fully exploit the GMA.

The main objective of this graduation thesis is to see the effects of non-linear and anisotropic behaviour of cement-based materials on their optimized topologies. To implement the GMA to do so, a non-linear FEM is required that has been developed using the Newton-Raphson iterative method. All the codes and algorithms used and referred to throughout this work are modelled using the platform `MATLAB`, which is a tool for numerical computation and visualization. Afterwards, the developed non-linear FEM codes with incorporation of the GMA were implemented into the Proportional Topology Optimization (PTO) method to perform both two-dimensional as well as three-dimensional optimizations. In the end, the resulting topology optimization algorithms have been used to study the effect of the GMA on optimized topology through the following research question:

> *What is the effect of asymmetric behaviour of cementitious materials under tension and compression on the optimized topology?*

This main research question has been subdivided into two sub questions related to the minimum compliance and stress-constrained problems, respectively:

- *What is the effect of asymmetric stiffness on the optimized topology?*
- *What is the effect of asymmetric strength on the optimized topology?*

During the current research, these research questions have been answered through implementation of the developed non-linear FEM with incorporation of the GMA into the PTO method and investigating several benchmark problems.

2 Literature Review

The aim of the research presented here is to see the effects of non-linear and anisotropic behaviour of cement-based materials on their optimized topologies for which a non-linear Finite Element Method (FEM) is required. Therefore, the literature review consists of these three consecutive parts that have been necessary for this research: (i) Granular Micromechanics Approach (GMA), (ii) non-linear FEM and (iii) topology optimization. This first part introduces the reader to the multi-scale material model used in the present work, including its derivation. The second part describes the procedure used to trace a structural model's response that exhibits non-linear material behaviour, as well as various required numerical integration implementations to do so. The third and final part discusses a conceptual design tool to find a structure's most optimal material distribution for a given design domain and boundary conditions, together with several optimization problems and the method to achieve this.

2.1 Granular Micromechanics Approach

The macroscopic mechanical behaviour of a material is highly dependent on its microstructure [14]. This is most clear in granular materials where the collective effect of displacement, deformation and rearrangement of particles during loading represents the evolution of microstructure that will in turn define the macroscopic behaviour of the material [2]. In other words, the macro-scale behaviour of granular materials is directly linked to the grain-scale through their granular mechano-morphology that is represented by their collective mechanics (microstructure) and interaction mechanisms (micromechanics) [1]. Various approaches at different length scales are available for incorporating the effects of a material's microstructure and its corresponding micromechanical properties into the macroscopic behaviour. These approaches are depicted in Figure 2.1, wherein the horizontal and vertical axes represent the length scale and computational demand, respectively [26]. Globally, these can be divided into two modelling categories [27].

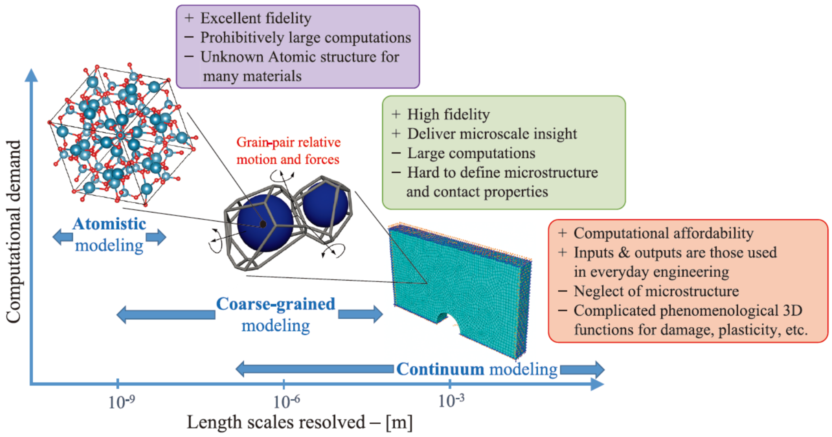

+ Excellent fidelity
− Prohibitively large computations
− Unknown Atomic structure for many materials
+ High fidelity
+ Deliver microscale insight
− Large computations
− Hard to define microstructure and contact properties
+ Computational affordability
+ Inputs & outputs are those used in everyday engineering
− Neglect of microstructure
− Complicated phenomenological 3D functions for damage, plasticity, etc.
Grain-pair relative motion and forces
Atomistic modeling
Coarse-grained modeling
Continuum modeling
Computational demand
10^{-9}
10^{-6}
10^{-3}
Length scales resolved – [m]

Figure 2.1: Modelling length scales and corresponding computational demand [1]

Methods in the smallest length scales include atomistic modelling, which relies on an explicit representation of the material's micro- or nano-structure and on enforcing mechanical equilibrium on each grain, yield results with a high degree of exactness (i.e. fidelity). However, it should be noted that these models with a discrete nature are limited in their applicability to large-scale problems due to their prohibitively large computational demand, even with the present computation and memory capacities. Besides, a reliable description of a material's microstructure is often not attainable for various granular materials, such as concrete, resulting from the challenges of grain-scale verification and validation (e.g. characterization such as particle sizes and shapes as well as specification of local interaction laws) [1, 2, 14, 26]. In the larger length scales, continuum modelling aims to describe a material's behaviour at a spatial scale, in which the grains and their motions remain concealed within a material point of a continuous system. However, these classical continuum models suffer from losing characteristic features due to neglecting grain scale phenomena (in terms of both the microsctructure and its evolution, as well as the inter-granular mechanisms), while being computationally affordable [1].

The Granular Micromechanics Approach (GMA) can be classified as a meso-scale approach that bridges the gap between continuum and discrete models [1,26], referred to as coarse-grained modelling in Figure 2.1. In this approach, the material point is envisioned as a collection of grains, each interacting with its neighbors through different inter-granular mechanisms [9]. All grain interactions contribute in defining the macroscopic material behaviour. GMA incorporates data from the material microstructure (and micromechanics) into the continuum model [14] in a computationally affordable manner [1]. In this method, it is neither needed nor intended to investigate the exact behaviour of all grains within a material point. Instead, the macroscopic behaviour of the material is derived by investigating the average behaviour of grain-pair interactions in different directions. Thereby, GMA makes it possible to incorporate information from the micro-scale into the material's macroscopic behaviour without enforcing explicit realization of all grains and their interactions [2].

Within GMA, the internal energy created by all the grain-pair interactions (i.e. the volumetric average of inter-granular deformation energies in all generic directions) is equated with the macro-scale energy created by the macroscopic stress and strain tensors (i.e. strain energy density). Thereby, the method automatically incorporates the effect of the material's microstructure, micromechanical properties and micro-scale phenomena taking place at the interacting grains [14]. The internal energy in each direction is defined through an inter-granular stiffness coefficient that governs the average force-displacement behaviour of the inter-granular contacts in that direction. It needs to be noted that these stiffness coefficients do not represent the stiffness of two isolated grains interacting with one another, but that these represent the behaviour of two grains embedded in a microstructure and incorporate the effects of the neighboring grains [2]. This is similar to the treatment of self-hardening and cross-hardening in crystal plasticity methods [28].

This section first derives the GMA for modelling a continuum material point, followed by macroscopic stiffness tensors for both plane stress and plane strain.

2.1.1 Granular Micromechanics Approach

The derivation of the GMA presented here substantially inherits from contributions by Poorsolhjouy and Misra in [2, 9, 14, 29], among others, in which the first step in the micro-macro identification is to identify the kinematics of grains inside a material point. However, these show large similarities with other works on micromechanical models for granular materials [27, 30, 31] and references therein. Consider two neighboring grains in a material point, n and p, as shown in Figure 2.2. Their relative displacement δ_i, ignoring grain rotations, is written as

$$\delta_i = u_i^p - u_i^n \,, \tag{2.1}$$

where u_i is the particle displacement and superscripts refer to the interacting particles. Subscripts follow conventions of index notation and summation over repeated indices is implied throughout this section, unless explicitly stated otherwise. For obtaining a continuum description of this discrete model, a Taylor series expansion of the displacement of grain n is used such that the displacement of grain p may be written as

$$u_i^p = u_i^n + u_{i,j}^n (x_j^p - x_j^n) \,, \tag{2.2}$$

where higher order terms are neglected to derive the first gradient behaviour (i.e. the effects of higher gradients of grain motion and nonaffine movements are neglected [31]) and the point of expansion is chosen as the centroid of grain n. In Equation (2.2), $u_{i,j} = \partial u_i / \partial x_j$ is the displacement gradient and x_j^n and x_j^p are the position vectors of the centroids of grains n and p, respectively. Because the anti-symmetric part of the displacement gradient does not contribute to energy and therefore stress, the displacement gradient $u_{i,j}$ in Equation (2.2) can be substituted with strain, ε_{ij}. Thus, by substituting Equation (2.2) into Equation (2.1) the relative displacement δ_i, representing the α^{th} inter-granular interaction, is given by

$$\delta_i^\alpha = u_{i,j}^n (x_j^p - x_j^n) = u_{i,j}^n l_j^\alpha \approx \varepsilon_{ij} l_j^\alpha \,, \tag{2.3}$$

where l_j^α is an inter-granular branch vector joining the centroids of grain-pair n and p and superscripts α correspond to the α^{th} contact of the grain. From Equation (2.3) it can be seen that the relative displacement of two grains inside the material is derived as the projection of the macroscopic strain tensor ε_{ij} in the direction of the grain-pair interaction line and multiplied by the length of the branch vector l_j^α. This assumption is generally known as the *kinematic* constraint.

To facilitate defining inter-granular force-displacement relationships, also referred to as inter-granular force-laws or microscopic constitutive relationships, a local coordinate system is introduced for each grain-pair interaction. This coordinate system is composed of three mutually orthogonal unit vectors, n_i, s_i and t_i. The unit vector n_i is in the direction of l_i (the vector joining the two particle centroids) and is referred to as the normal direction. The two other unit vectors, s_i and t_i, are orthogonal and lie in the plane whose normal vector is n_i. Two interacting grains and the direction of their local coordinate axes with respect to the global coordinate system are depicted in Figure 2.2.

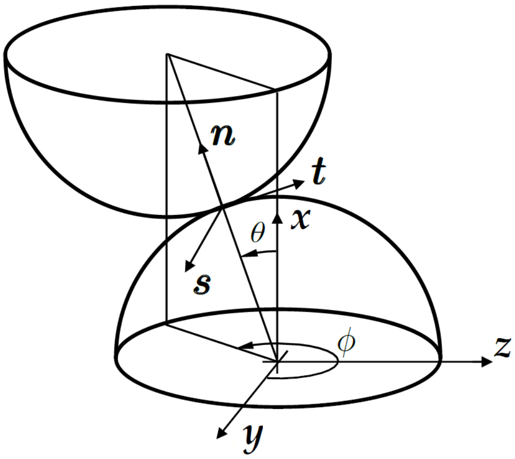


Figure 2.2: Grains in contact with local and global spherical coordinate systems [2]

In general, vectors s_i and t_i can be chosen arbitrarily in the tangential plane with normal vector n_i. Nevertheless, using the spherical coordinate system as shown in Figure 2.2, the following three unit vectors are used in future derivations:

$$\begin{aligned} n_i &= \langle \cos\theta, \sin\theta\cos\phi, \sin\theta\sin\phi \rangle \,, \\ s_i &= \langle -\sin\theta, \cos\theta\cos\phi, \sin\theta\sin\phi \rangle \,, \\ t_i &= \langle 0, -\sin\phi, \cos\phi \rangle \,, \end{aligned} \tag{2.4}$$

where θ is the polar angle measured from the positive x axis and ϕ is the azimuth angle, of the projection on the yz plane, measured from the positive y axis. Now, the relative displacement vector δ_i can be decomposed into components along the local coordinate axes as

$$\begin{cases} \delta_\mathrm{n} = \delta_i n_i = \varepsilon_{ij} l_j n_i \,, \\ \delta_\mathrm{s} = \delta_i s_i = \varepsilon_{ij} l_j s_i \,, \\ \delta_\mathrm{t} = \delta_i t_i = \varepsilon_{ij} l_j t_i \,, \end{cases} \tag{2.5}$$

where δ_n is the component in the direction of the vector joining the particle centroids, representing the relative normal displacement between two grains, and the other two, δ_s and δ_t, representing the relative displacements in the two tangential directions. Note that the variables δ_n, δ_s and δ_t are *scalar* values and, therefore, subscripts n, s and t *do not* follow index summation conventions throughout this section.

The decomposed relative displacement components will result in conjugate inter-granular force components at each grain-pair interaction. These can be obtained from inter-granular force-laws, which will be discussed later. To this end, normal and tangential stiffness coefficients are used for defining the normal and tangential inter-granular force components

$$\begin{Bmatrix} f^\alpha_\mathrm{n} \\ f^\alpha_\mathrm{s} \\ f^\alpha_\mathrm{t} \end{Bmatrix} = \begin{bmatrix} k^\alpha_\mathrm{n} & 0 & 0 \\ 0 & k^\alpha_\mathrm{s} & 0 \\ 0 & 0 & k^\alpha_\mathrm{t} \end{bmatrix} \begin{Bmatrix} \delta^\alpha_\mathrm{n} \\ \delta^\alpha_\mathrm{s} \\ \delta^\alpha_\mathrm{t} \end{Bmatrix} \,, \tag{2.6}$$

where k_n, k_s and k_t are microscopic or inter-granular stiffness coefficients in directions n, s and t, respectively. Recall that the inter-granular force, stiffness and displacement components used in Equation (2.6) are *scalar* values such that the indices n, s and t *do not follow* summation convention. For continuation of the derivation, the inter-granular force-displacement relationships are written in the global coordinate system by applying a coordinate transformation using the following rotation tensor for each inter-granular contact:

$$T_{ij}^{\alpha} = \begin{bmatrix} n_1 & s_1 & t_1 \\ n_2 & s_2 & t_2 \\ n_3 & s_3 & t_3 \end{bmatrix} . \tag{2.7}$$

Therefore, the inter-granular force-displacement relationships in the global coordinate system can be written as

$$f_i^{\alpha} = K_{ij}^{\alpha} \delta_j^{\alpha} \qquad \text{with} \qquad K_{ij}^{\alpha} = T_{ip}^{\alpha} k_{pq}^{\alpha} T_{jq}^{\alpha} \, , \tag{2.8}$$

where K_{ij}^{α} is the inter-granular stiffness tensor in the global coordinate system xyz, while k_{pq}^{α} represents the diagonal inter-granular stiffness tensor in a grain's local coordinate system nst as stated in Equation (2.6).

The final step in the micro-macro identification is to set the macroscopic strain energy density equal to the volumetric average of deformation energy of all inter-granular contacts. This is done by applying the Principle of Virtual Work (PVW) to all of the grain-pairs. The incremental microscopic internal strain energy of the α^{th} contact, $\mathrm{d}W^{\alpha}$, resulting from the contact's relative displacements and its conjugate inter-granular forces can be formulated as

$$\mathrm{d}W^{\alpha} = f_\mathrm{n}^{\alpha} \mathrm{d}\delta_\mathrm{n}^{\alpha} + f_\mathrm{s}^{\alpha} \mathrm{d}\delta_\mathrm{s}^{\alpha} + f_\mathrm{t}^{\alpha} \mathrm{d}\delta_\mathrm{t}^{\alpha} = f_i^{\alpha} \mathrm{d}\delta_i^{\alpha} \, . \tag{2.9}$$

Next, the incremental Helmholtz free energy density in a the material point, defined as the volumetric average of the energy in all the grain-pair interactions, is equal to the macroscopic strain energy density. The incremental macroscopic strain energy density is in its turn defined as the scalar product of the (Cauchy) stress tensor and the incremental strain tensor:

$$\mathrm{d}W = \frac{1}{V} \sum_{\alpha=1}^{N_c} \mathrm{d}W^{\alpha} = \sigma_{ij} \mathrm{d}\varepsilon_{ij} \, , \tag{2.10}$$

where the summation over α goes over all grain-pair interactions, N_c is the total number of contacts in a material point and V its volume. It is in this step that the effect of a material's microstructure, micromechanical properties and micro-scale phenomena taking place at the interacting grains is incorporated in its macroscopic mechanical behaviour, by which GMA bridges the gap between continuum and discrete models. This is done by substituting the results of Equation (2.9) and then Equation (2.3) into Equation (2.10), thus equating the macroscopic strain energy density with the microscopic internal energy created by all the grain-pair interactions (i.e. the volumetric average of inter-granular deformation energies in all generic directions):

$$\mathrm{d}W = \underbrace{\sigma_{ij}\mathrm{d}\varepsilon_{ij}}_{macroscopic} = \underbrace{\frac{1}{V}\sum_{\alpha=1}^{N_c}\mathrm{d}W^{\alpha} = \frac{1}{V}\sum_{\alpha=1}^{N_c} f_i^{\alpha}\mathrm{d}\delta_i^{\alpha} = \frac{1}{V}\sum_{\alpha=1}^{N_c} f_i^{\alpha}\mathrm{d}\left(\varepsilon_{ij}l_j^{\alpha}\right)}_{microscopic} . \tag{2.11}$$

From rewriting Equation (2.10) and substitution of the right-hand side of Equation (2.11), it follows that the macroscopic stress tensor can be obtained in terms of the inter-granular force vectors as

$$\sigma_{ij} = \frac{\mathrm{d}W}{\mathrm{d}\varepsilon_{ij}} = \frac{1}{V}\sum_{\alpha=1}^{N_c} f_k^{\alpha}\frac{\mathrm{d}\left(\varepsilon_{kl}l_l^{\alpha}\right)}{\mathrm{d}\varepsilon_{ij}} = \frac{1}{V}\sum_{\alpha=1}^{N_c} f_k^{\alpha}\delta_{ik}\delta_{jl}l_l^{\alpha} = \frac{1}{V}\sum_{\alpha=1}^{N_c} f_i^{\alpha}l_j^{\alpha} , \tag{2.12}$$

where δ_{ij} represents the Kronecker delta. The macroscopic *tangential* stiffness tensor for a material point can now be derived using the local stiffness parameters through incorporation of the inter-granular force-laws (substituting Equation (2.8)) and then the *kinematic* assumption (Equation (2.3)) into Equation (2.12) as

$$\begin{aligned} C_{ijkl} &= \frac{\mathrm{d}\sigma_{ij}}{\mathrm{d}\varepsilon_{kl}} = \frac{1}{V}\sum_{\alpha=1}^{N_c}\frac{\mathrm{d}f_i^{\alpha}}{\mathrm{d}\varepsilon_{kl}}l_j^{\alpha} = \frac{1}{V}\sum_{\alpha=1}^{N_c}\frac{\mathrm{d}f_i^{\alpha}}{\mathrm{d}\delta_p^{\alpha}}\frac{\mathrm{d}\delta_p^{\alpha}}{\mathrm{d}\varepsilon_{kl}}l_j^{\alpha} \\ &= \frac{1}{V}\sum_{\alpha=1}^{N_c}K_{ip}^{\alpha}\delta_{pk}\delta_{ql}l_q^{\alpha}l_j^{\alpha} = \frac{1}{V}\sum_{\alpha=1}^{N_c}K_{ik}^{\alpha}l_l^{\alpha}l_j^{\alpha} . \end{aligned} \tag{2.13}$$

To recapitulate, in traditional continuum mechanics the macroscopic stress tensor is directly calculated from the macroscopic strain tensor by means of tensorial constitutive equations. However, in the GMA presented here, using the *kinematic* approach, the inter-granular displacement vectors are defined as projections of the macroscopic strain tensor in the direction of the grain-pair interaction line and then scaled by the length of the contact line (known as the *kinematic* assumption). From the inter-granular displacement vectors, inter-granular force vectors are found using inter-granular force-laws or microscopic constitutive laws. The inter-granular force and displacement vectors are then used to calculate the internal energy of each inter-granular contact. Using the PVW, the macroscopic strain energy density is set equal to the volumetric average of all grain-pair interactions from which the macroscopic stress and stiffness tensors are derived. A schematic representation of the GMA is shown in Figure 2.3.

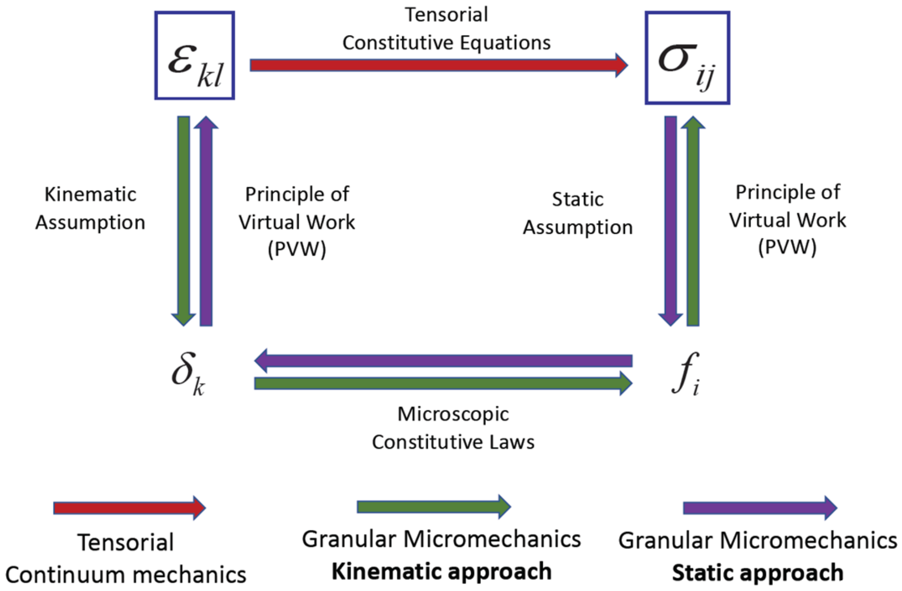


Figure 2.3: Schematic description of the Granular Micromechanics Approach [3]

Besides the GMA presented here, using the *kinematic* assumption, Figure 2.3 also displays another one using a *static* assumption. This method is used for cases where initially the stress tensor is known, such as load controlled experiments. However, in the current research only the GMA using the *kinematic* assumption is of interest. For the static approach, the interested reader is referred to [14,27,30,31] as well as references provided in there.

The derivation of the GMA continues with considering a density distribution function. So far, the derivation resulted in the summation formula for calculating the macroscopic *tangential* stiffness tensor (Equation (2.13)) of which the end result is repeated here for convenience:

$$C_{ijkl} = \frac{1}{V} \sum_{\alpha=1}^{N_c} K_{ik}^{\alpha} l_l^{\alpha} l_j^{\alpha} \, . \tag{2.14}$$

It should be noted that in this summation formula the inter-granular stiffness tensor K_{ik}^{α} and the branch vector l_j^{α} are defined separately for each individual contact, which is not efficient. Therefore, the summation will be transformed to integration with a density distribution function which enables GMA to derive the macroscopic behaviour in a more computationally efficient manner while still allowing the model to capture the (anisotropic) evolution of the microstructure and its effects on the macroscopic response [2]. This transition from summation to integration will be elaborated here, based on [14].

Note that in Equation (2.14) l_i can be replaced by ln_i, where l represents the distance between the centroids of two neighboring grains and n_i represents the unit vector in that direction [2, 14]. Therefore, Equation (2.14) can written as

$$C_{ijkl} = \frac{1}{V} \sum_{\alpha=1}^{N_c} \left(l^{\alpha}\right)^2 n_j^{\alpha} n_l^{\alpha} K_{ik}^{\alpha} \, . \tag{2.15}$$

Equation (2.15) can be written in expanded form by writing the grain-pair stiffness tensor K_{ik}^{α} in terms of its normal and tangential components using Equation (2.8) as

$$C_{ijkl} = \frac{1}{V}\sum_{\alpha=1}^{N_c} n_j^{\alpha} n_l^{\alpha} \left((l^{\alpha})^2 K_{\mathrm{n}}^{\alpha} n_i^{\alpha} n_k^{\alpha} + (l^{\alpha})^2 K_{\mathrm{s}}^{\alpha} s_i^{\alpha} s_k^{\alpha} + (l^{\alpha})^2 K_{\mathrm{t}}^{\alpha} t_i^{\alpha} t_k^{\alpha} \right) . \tag{2.16}$$

Recall that the subscripts n, s and t *do not follow* index notation conventions since these represent the normal and tangential stiffness coefficients. The terms inside the summation in Equation (2.16) are products of direction cosines, the branch length and inter-granular stiffness coefficients which can be grouped into discrete solid angles. Therefore, the summation over all contacts in Equation (2.16) can be rewritten as a summation over θ and ϕ as

$$\begin{aligned} C_{ijkl} = \frac{1}{V}\sum_{\theta}\sum_{\phi} \Bigg[& \left(\sum_{\rho}^{N^{\rho}} (l^{\rho})^2 K_{\mathrm{n}}^{\rho} \right) n_j n_l n_i n_k \\ & + \left(\sum_{\rho}^{N^{\rho}} (l^{\rho})^2 K_{\mathrm{s}}^{\rho} \right) n_j n_l s_i s_k + \left(\sum_{\rho}^{N^{\rho}} (l^{\rho})^2 K_{\mathrm{t}}^{\rho} \right) n_j n_l t_i t_k \Bigg] , \end{aligned} \tag{2.17}$$

where $N^{\rho}(\theta, \phi)$ is the total number contacts for a given solid angle defined by θ and ϕ, such that

$$N = \sum_{\theta}\sum_{\phi} N^{\rho}(\theta, \phi) , \tag{2.18}$$

and the summation over ρ is the sum of the branch length squared and the inter-granular stiffness for all contacts in that solid angle. Now the directional density distribution function $\xi(\theta, \phi)$ is introduced, which is defined as

$$\xi(\theta, \phi) = \frac{\sum_{\rho}^{N^{\rho}} (l^{\rho})^2 K_{\mathrm{n}}^{\rho}}{\sum_{\alpha=1}^{N_c} (l^{\alpha})^2 K_{\mathrm{n}}^{\alpha}} . \tag{2.19}$$

The density distribution function in Equation (2.19) represents the relative measure of a material's stiffness in a given direction, resulting from a combination of grain-size, number of contacts and inter-granular stiffness. For materials composed of a large number of grain sizes and compositions, it is practical to define an average product of the branch length squared and the inter-granular stiffness, $l^2 K_{\mathrm{n}}$, as

$$l^2 K_{\mathrm{n}} = \frac{\sum_{\alpha=1}^{N_c} (l^{\alpha})^2 K_{\mathrm{n}}^{\alpha}}{N} \quad \text{with} \quad N = V N_p , \tag{2.20}$$

where l may be considered as the average branch length, K_{n} as the material's average inter-granular normal stiffness and N_p is the number density of contacts or the number of contacts in a unit volume of the material [9]. By doing the same operations for the tangential terms of the stiffness tensor in Equation (2.17), Equations (2.17) through (2.20) can be combined and written in integral form as follows:

$$\begin{aligned} C_{ijkl} &= l^2 N_p \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \left(K_{\mathrm{n}} n_j n_l n_i n_k + K_{\mathrm{s}} n_j n_l s_i s_k + K_{\mathrm{t}} n_j n_l t_i t_k \right) \xi(\theta, \phi) \sin\theta \mathrm{d}\phi \mathrm{d}\theta \\ &= l^2 N_p \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \left(K_{ik} n_j n_l \right) \xi(\theta, \phi) \sin\theta \mathrm{d}\phi \mathrm{d}\theta \,. \end{aligned} \tag{2.21}$$

The density distribution function $\xi(\theta, \phi)$ used in the integral form in Equation (2.21) is thus only a function of the number of contacts, their inter-granular stiffness and contact length in each direction which determines the directional distribution of contact properties. In addition to transforming the summation in Equation (2.14) into the integration in Equation (2.21), it may also facilitate modelling inherently anisotropic materials (materials in which the contact properties in different directions are not identical [2]).

However, for modelling isotropic materials, such as concrete (which is the aim of the present work), there should be no directional preference for the grain-pair interactions and their properties. Therefore, the number density of contacts, their stiffness and lengths in different directions should be independent of the direction such that the probability of having contacts in any direction is equal. In that case, for an isotropic material, the density distribution function $\xi(\theta, \phi)$ reduces to a mere normalization constant [27]:

$$\xi(\theta, \phi) = \frac{1}{4\pi} \Rightarrow \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} \xi(\theta, \phi) \sin\theta \mathrm{d}\phi \mathrm{d}\theta = 1 \,. \tag{2.22}$$

Since isotropic materials have identical behaviour in the two shear directions [14, 26], the tangential components s and t of the relative displacement vector δ_i in Equation (2.5) can be combined as follows:

$$\left. \begin{aligned} \delta_{\mathrm{n}} &= \delta_i n_i \\ \delta_{\mathrm{w}} &= \sqrt{\left(\delta_i s_i\right)^2 + \left(\delta_i t_i\right)^2} \end{aligned} \right\} \text{ with } \delta_i = \varepsilon_{ij} l_j \,. \tag{2.23}$$

Hereafter, subscripts n and w imply components in normal and tangential directions, respectively, and *do not follow* summation convention. For an isotropic material, the integrals presented in Equation (2.21) can be computed analytically. These closed-form solutions yield the following relationships for Young's modulus E and Poisson's ratio ν [14]

$$\begin{aligned} E &= \frac{l^2 N_p k_{\mathrm{n}}}{3} \frac{2k_{\mathrm{n}} + 3k_{\mathrm{w}}}{4k_{\mathrm{n}} + k_{\mathrm{w}}} \,, \\ \nu &= \frac{k_{\mathrm{n}} - k_{\mathrm{w}}}{4k_{\mathrm{n}} + k_{\mathrm{w}}} \,, \end{aligned} \tag{2.24}$$

or alternatively formulated as in [2]

$$\begin{aligned} E &= k_{\mathrm{n}}\mathcal{C}\,, & \nu &= \frac{1-\beta}{4+\beta}\,, \\ \mathcal{C} &= \frac{l^2\rho}{3}\frac{2+3\beta}{4+\beta}\,, & \beta &= \frac{k_{\mathrm{w}}}{k_{\mathrm{n}}}\,, \end{aligned} \tag{2.25}$$

where $\rho = N_c/V$ is the volumetric number density of contacts in a material point. However, even though a material may be initially isotropic when unloaded, since within GMA the inter-granular interactions are experiencing a different loading history and are generally non-linear, contacts that are oriented in all different directions will have their inter-granular stiffness evolve depending on their orientation and loading path. Therefore, a material will exhibit anisotropic behaviour when loaded, which is called *induced* anisotropy [14]. Accordingly, the model is capable of capturing the history-dependent, anisotropic response of materials [2].

For modelling materials with *inherent* anisotropy, other forms of the probability density distribution function $\xi(\theta, \phi)$ need to be defined. By including appropriate terms within a density distribution function expressed as spherical harmonic expansion, materials with all different levels of *inherent* anisotropy ranging from completely isotropic to completely anisotropic can be modelled. For more details, reference is made to [14].

The macroscopic stiffness tensor, C_{ijkl}, has a number of symmetries that reduce its number of independent components [3]. These include the two *minor symmetries* and *major symmetry* as discussed in, for example, [3, 32]. In three-dimensional analysis, C_{ijkl} has 81 independent components [32]. In fact, the two *minor symmetries* imply that the macroscopic stiffness tensor can be represented by a 6 × 6 matrix with 36 independent components, relating the 6 strain components and 6 stress components. The *major symmetry* implies that the effect of the i^{th} strain component on the j^{th} stress component equals the effect of the j^{th} strain component on the i^{th} stress component. To avoid the associated redundancy in its components, the 6 × 6 matrix becomes symmetric having only 21 independent components. The aforementioned symmetries are handled after deriving the macroscopic stiffness tensor through Equation (2.21) as:

$$C_{ij} = \frac{C_{ijkl} + C_{ijlk} + C_{jikl} + C_{jilk}}{4}\,. \tag{2.26}$$

While the analysis of three-dimensional material behaviour is useful for many applications, there is a wide range of engineering applications in which a structure can be adequately modelled in two dimensions. Moreover, the Finite Element Analysis (FEA) of two-dimensional structures is much less computationally expensive and therefore more suitable for implementations within topology optimization algorithms. This is partly the reason why most topology optimization benchmark problems are also available in two-dimensional format [5, 9, 15, 20–23, 29, 33–40]. Accordingly, in what follows the specifications that need to be applied on GMA to allow for two-dimensional analysis of structures with different boundary conditions will be presented.

2.1.2 Stiffness Tensors for Plane Stress & Plane Strain

For adequately modelling structures in two dimensions, one should start by introducing the structure's geometry and its corresponding axes. For example, consider the simple plate structure as shown in Figure 2.4. The left side of the figure illustrates that in this particular structure the x and y axes are in the in-plane directions, while the z axis is in the normal or out-of-plane direction. The right side of Figure 2.4 depicts the in-plane directions, which are the only directions considered in both plane stress and plane strain problems.

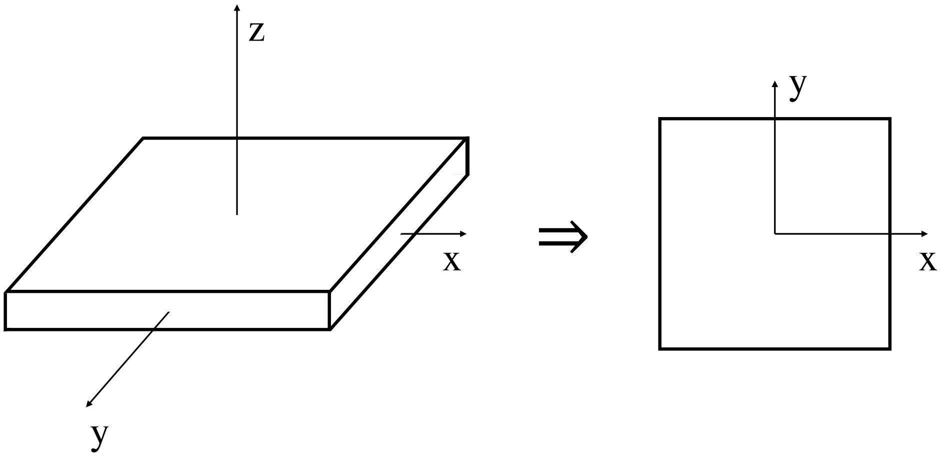


Figure 2.4: Geometry and corresponding axes of a simple plate structure

In both plane stress and plane strain problems, the displacement field is uniquely given by the horizontal and vertical displacements in the directions of the x and y axes. Therefore, only the three in-plane strain and stress components have to be considered. By definition, in plane stress all the other components of stress are equal to zero while in plane strain the stress in the direction perpendicular to the xy plane is non-zero such that the strain in that direction is zero [41].

- *Plane stress*

  Since in plane stress problems only the strain components in the xy plane are calculated using the aforementioned displacement field and the out-of-plane components of the stress tensor are assumed to be zero, as a result, stress-strain relationships constitute a mixed boundary condition. Since the inter-granular force-laws (see Equation (2.6)) are in general non-linear, forward Euler technique is utilized for solving the implicit set of equations [2]. In this approach, the three-dimensional constitutive equation is partitioned as follows [42]

$$\begin{pmatrix} \mathrm{d}\mathbf{S}_m \\ \mathrm{d}\mathbf{S}_z \end{pmatrix} = \begin{pmatrix} \mathbf{C}_{mm} & \mathbf{C}_{mz} \\ \mathbf{C}_{zm} & \mathbf{C}_{zz} \end{pmatrix} \begin{pmatrix} \mathrm{d}\mathbf{E}_m \\ \mathrm{d}\mathbf{E}_z \end{pmatrix} , \tag{2.27}$$

in which $d\mathbf{S}_m$ and $d\mathbf{E}_m$ denote the incremental in-plane stress and strain tensors that can be directly found from the non-linear FEM, respectively. $d\mathbf{S}_z$ and $d\mathbf{E}_z$ are the out-of-plane stress and strain components, respectively. In this case, $\mathrm{d}\mathbf{S}_m$ and $\mathrm{d}\mathbf{S}_z$ are defined as

$$\mathrm{d}\mathbf{S}_m = \begin{Bmatrix} \mathrm{d}\sigma_{11} \\ \mathrm{d}\sigma_{22} \\ \mathrm{d}\sigma_{12} \end{Bmatrix} , \qquad \mathrm{d}\mathbf{S}_z = \begin{Bmatrix} \mathrm{d}\sigma_{33} = 0 \\ \mathrm{d}\sigma_{23} = 0 \\ \mathrm{d}\sigma_{13} = 0 \end{Bmatrix} . \tag{2.28}$$

For completeness, $\mathbf{C}$ is the material tangent modulus or tangential stiffness tensor. From Equation (2.27), $\mathrm{d}\mathbf{E}_z$ and $\mathrm{d}\mathbf{S}_m$ are found as

$$\begin{gathered} \mathrm{d}\mathbf{E}_z = -\mathbf{C}_{zz}^{-1}\mathbf{C}_{zm}\mathrm{d}\mathbf{E}_m \, , \\ \mathrm{d}\mathbf{S}_m = \mathbf{D}\mathrm{d}\mathbf{E}_m \, , \\ \mathbf{D} = \left[\mathbf{C}_{mm} - \mathbf{C}_{mz}\mathbf{C}_{zz}^{-1}\mathbf{C}_{zm}\right] \, , \end{gathered} \tag{2.29}$$

where $\mathbf{D}$ is the tangent modulus or the tangential stiffness tensor for a two-dimensional problem, in this case the plane stress condition. The above forward Euler technique is implemented within the non-linear FEM and the out-of-plane stress components are preserved to be zero by the global Newton-Raphson iterative method.

- *Plane strain*

  In plane strain, by definition, the strain components in the out-of-plane directions are considered to be zero. Therefore, the set of equations can be rather straightforward explicitly solved such that there is no need to utilize forward Euler technique as in plane stress. Partitioning the three-dimensional constitutive equation in a similar fashion as in Equation (2.27), where in this case $\mathrm{d}\mathbf{E}_m$ and $\mathrm{d}\mathbf{E}_z$ are defined as:

$$\mathrm{d}\mathbf{E}_m = \begin{Bmatrix} \mathrm{d}\varepsilon_{11} \\ \mathrm{d}\varepsilon_{22} \\ \mathrm{d}\varepsilon_{12} \end{Bmatrix} , \qquad \mathrm{d}\mathbf{E}_z = \begin{Bmatrix} \mathrm{d}\varepsilon_{33} = 0 \\ \mathrm{d}\varepsilon_{23} = 0 \\ \mathrm{d}\varepsilon_{13} = 0 \end{Bmatrix} . \tag{2.30}$$

As a result of Equations (2.27) and (2.30), the tangent modulus or tangential stiffness tensor for the plane strain condition can be simply obtained by eliminating the out-of-plane components such that

$$\mathrm{d}\mathbf{S}_m = \mathbf{D}\mathrm{d}\mathbf{E}_m \quad \text{with} \quad \mathbf{D} = \mathbf{C}_{mm} \, . \tag{2.31}$$

2.2 Non-Linear Finite Element Method

The main strategy in non-linear Finite Element Methods (FEM) is to discretize the loading history into a series of piece-wise linear or weakly non-linear steps. To arrive at the final solution, a combination of load or displacement incrementation and iteration procedures are used [11]. The system of equations to be solved can be expressed as follows:

$$\mathbf{F} = \mathbf{f}(\mathbf{u}) \,, \tag{2.32}$$

where $\mathbf{F}$ is a column vector containing the applied external forces and $\mathbf{f}(\mathbf{u})$ represents the internal system forces as a function of the displacements $\mathbf{u}$. In the case of linear-elasticity, Equation 2.32 can be solved directly as it reduces to $\mathbf{F} = \mathbf{Ku}$, where $\mathbf{K}$ is a square matrix often referred to as the stiffness matrix. However, for non-linear material behaviour as well as geometric non-linearities more sophisticated solution strategies are required.

This section starts with an elaborate explanation of the Newton-Raphson iterative method for load control, including brief references for alternatives. Afterwards, numerical integration will be introduced briefly, followed by several worked out numerical integration implementations used in this research.

2.2.1 Newton-Raphson Iterative Method

In non-linear FEA the objective is to trace the response of a structural model subjected to a given loading history which is usually achieved by an *incremental-iterative procedure* [43]. In such procedures, the total load to be applied is divided into small steps or increments which are applied individually. One of the most common numerical methods for solving non-linear problems is the Newton-Raphson iterative method [12], also known as Newton's method, which is at the core of many solver methods [4]. This method is commonly used to find the roots of an equation [44] and works by initially guessing a trial solution and successively improves this 'guess' by using the slope of the load-displacement curve. This non-linear curve is approximated by a series of tangents [11]. Convergence of the method largely depends on the following: (i) how close the starting solution is to the final solution, i.e. the initial guess should not be very "far" from the exact solution [44], and (ii) the smoothness of the non-linear function, i.e. the function should not be discontinuous [18]. In the present study, the Newton-Raphson iterative method will be used for solving non-linear FEM problems. The implementation of this method considered here is referred to as *load control* and will be discussed next.

In load control, the vector containing the applied external forces $\mathbf{F}$ is monotonically increased from zero up until a desired, predefined load value. In structural analysis, the objective of the Newton-Raphson iterative method is to find displacements $\mathbf{u}$ that satisfy

$$\mathbf{f}(\mathbf{u}) - \lambda \mathbf{F} = \mathbf{0} \,, \tag{2.33}$$

where $\mathbf{f}(\mathbf{u})$ represents the internal system forces as a function of the displacements $\mathbf{u}$, $\mathbf{F}$ the externally applied forces and λ a scalar quantity which allows for controlling the load incrementation [4, 12]. Therefore, the desired, predefined load value is reached when $\lambda = 1$. To satisfy equilibrium between the internal and externally applied forces, iterations should be performed to ensure that the out-of-balance forces or the residual $\mathbf{R(u)}$ remains small (i.e. below a specified tolerance). Therefore, Equation (2.33) can be written as

$$\mathbf{R(u)} = \mathbf{f(u)} - \lambda\mathbf{F} = \mathbf{0}\,. \tag{2.34}$$

Further following the procedure as presented in [12], the displacements $\mathbf{u}$ and scalar quantity λ are of interest in the system of equations presented in Equation (2.34). The value of λ is monotonically increased in each increment to find $\mathbf{u}$ such that equilibrium is satisfied. Assuming that $\{\mathbf{u}_0, \lambda_0\}$ is the last converged solution, the load increment is initiated by imposing

$$\lambda' = \lambda_0 + \Delta\lambda\,, \tag{2.35}$$

where $\Delta\lambda$ is a known predefined incrementation parameter, which immediately violates equilibrium. Therefore, the displacements $\mathbf{u}_0$ require updating

$$\mathbf{u}' = \mathbf{u}_0 + \Delta\mathbf{u}\,, \tag{2.36}$$

such that Equation (2.34) can be updated as

$$\mathbf{R(u')} = \mathbf{R}(\mathbf{u}_0 + \Delta\mathbf{u}) = \mathbf{0} \Rightarrow \mathbf{f}(\mathbf{u}_0 + \Delta\mathbf{u}) - (\lambda_0 + \Delta\lambda)\mathbf{F} = \mathbf{0}\,. \tag{2.37}$$

However, the portion related to the internal forces $\mathbf{f}(\mathbf{u}_0 + \Delta\mathbf{u})$ can be expressed in terms of $\mathbf{f}(\mathbf{u}_0)$ by means of a first-order Taylor series expansion provided the increments are sufficiently small. This linearization about the local point $\mathbf{u}'$ is required for solving the set of non-linear equations [4]. Therefore, the internal forces related portion becomes:

$$\mathbf{f}(\mathbf{u}_0 + \Delta\mathbf{u}) = \mathbf{f}(\mathbf{u}_0) + \left[\frac{\partial\mathbf{f(u)}}{\partial\mathbf{u}}\right]_{\mathbf{u}_0} \cdot \Delta\mathbf{u} = \mathbf{f}(\mathbf{u}_0) + \mathbf{K}_{\mathbf{u}_0} \cdot \Delta\mathbf{u}\,, \tag{2.38}$$

where $\mathbf{K} = [\partial\mathbf{f(u)}/\partial\mathbf{u}]$ is typically referred to as the "Jacobian" or tangential stiffness matrix (*stiffness matrix* in short) since it represents a tangent line to the equilibrium path at $\mathbf{u}$. Now the system can be solved for $\Delta\mathbf{u}$ by combining Equations (2.37) and (2.38) as:

$$\mathbf{f}(\mathbf{u}_0) + \mathbf{K}_{\mathbf{u}_0} \cdot \Delta\mathbf{u} - (\lambda_0 + \Delta\lambda)\mathbf{F} = \mathbf{0} \Rightarrow \overbrace{\mathbf{f}(\mathbf{u}_0) - \lambda_0\mathbf{F}}^{\mathbf{0}} + \mathbf{K}_{\mathbf{u}_0} \cdot \Delta\mathbf{u} - \Delta\lambda\mathbf{F} = \mathbf{0} \Rightarrow$$
$$\Delta\mathbf{u} = \mathbf{K}_{\mathbf{u}_0}^{-1} \cdot (\Delta\lambda\mathbf{F})\,. \tag{2.39}$$

The displacement correction $\Delta\mathbf{u}$ can be calculated from Equation (2.39). However, even though it is imposed that $\Delta\mathbf{u}$ would satisfy Equation (2.37) the linearized Taylor series expansion approximation prohibits achieving equilibrium immediately (linear-response). Therefore, a non-zero residual vector $\hat{\mathbf{R}}(\mathbf{u}')$ will be obtained if the system in Equation (2.37) is evaluated at the new point $(\mathbf{u}', \lambda')$. Nevertheless, the residual (a measure of error) can be used to find corrections to the solution and subsequently update the linear model for the next step. Using this, a new displacement correction $\delta\mathbf{u}$ can be computed as follows:

$$\begin{gathered}
\mathbf{R}(\mathbf{u}_0+\Delta\mathbf{u}+\delta\mathbf{u})=\mathbf{0}\Rightarrow\mathbf{f}(\mathbf{u}_0+\Delta\mathbf{u})+\mathbf{K}_{\mathbf{u}'}\cdot\delta\mathbf{u}-(\lambda+\Delta\lambda)\mathbf{F}=\mathbf{0}\Rightarrow\\
\mathbf{K}_{\mathbf{u}'}\cdot\delta\mathbf{u}=-(\mathbf{f}(\mathbf{u}_0+\Delta\mathbf{u})-(\lambda+\Delta\lambda)\mathbf{F})\Rightarrow\mathbf{K}_{\mathbf{u}'}\cdot\delta\mathbf{u}=-\hat{\mathbf{R}}(\mathbf{u}')\Rightarrow\\
\delta\mathbf{u}=-\mathbf{K}_{\mathbf{u}'}^{-1}\cdot\hat{\mathbf{R}}(\mathbf{u}')\,.
\end{gathered}\tag{2.40}$$

Once the new displacement correction is found using Equation (2.40) and the system in Equation (2.37) is evaluated, the new points $(\mathbf{u}'+\delta\mathbf{u},\lambda')$ would generally result in a new and smaller residual vector $\hat{\mathbf{R}}(\mathbf{u}'')$. This process is repeated in an iterative method by providing displacement corrections until a specified tolerance on error or an iteration limit has been reached to prevent an infinite loop if the solver were to diverge [4]. The error is often measured as the Euclidean norm of the residual vector which should be less than a specified tolerance, referred to as the convergence criterion. The convergence in this research is defined as the ratio of vector norms, $\|\Delta\mathbf{u}\|/\|\mathbf{u}\|$, to check if $\Delta\mathbf{u}$ is small with respect to $\mathbf{u}$. The convergence criterion is always set to 10^{-8}, unless stated otherwise. If the error is smaller than the convergence criterion, the iterative procedure is terminated and the method proceeds to the next increment. The analysis is finished once the total load to be applied is reached and equilibrium is satisfied (i.e. the residual is below the specified tolerance for all increments). The total deformation of the structure is found by summing all the incremental deformations associated with the load increments. A graphical representation of the Newton-Raphson iterative method is shown in Figure 2.5.

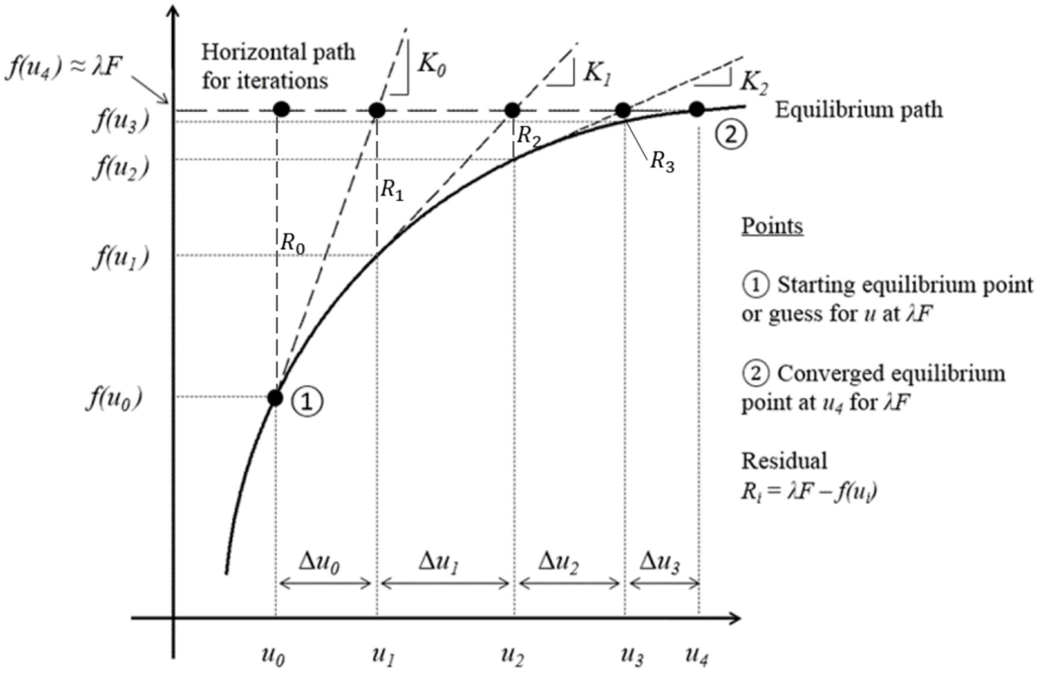


Figure 2.5: Newton-Raphson iterative method [4]

Load control is also associated with a major drawback: any solution strategy using load control, independent of the choice of iterative algorithm, fails if the prescribed load cannot be equilibrated by the resistance of the structure. This usually happens when the load-carrying capacity of the structure is exhausted (i.e. a peak is reached in the load-displacement curve) while the load remains incrementally increasing [43]. In other words: load control does not work when the load-displacement tangent becomes (almost) horizontal, because the load increments enforce the load to increase while in such cases the load should remain constant or decrease to satisfy equilibrium. Points at which the tangent stiffness reaches zero are called critical or limit points, after which load control fails to accurately follow the 'equilibrium' path because singularity of $\mathbf{K}$ will cause the Newton-Raphson iterative method to fail or diverge from finding a solution since it requires matrix inversion for updating $\mathbf{u}$ [4]. This occurs for material behaviour with strain softening, for example. In such problems *displacement control* should be used [11], which utilizes displacement increments instead of load increments for specified Degrees Of Freedom (DOFs). In this way, the displacements $\mathbf{u}$ can be continuously increased while remaining on the equilibrium curve [12].

In order to study a structure's post-peak branch of the load-displacement curve to clearly demonstrate the presence of a limit point or to obtain insight in it's ductility and energy absorption capability, displacement control is also considered in the present work. For more information on displacement control in general, its implementation of the Newton-Raphson iterative method and a comparison between load and displacement control, the reader is referred to Appendix A. Elaborated examples of both the load and displacement control variations of the Newton-Raphson iterative method can also be found in there, as well as brief references for a more complex solution procedure.

In the current implementation of the Newton-Raphson iterative method, the stiffness matrix is updated for each iteration. In this way, the solution procedure exhibits second order or quadratic convergence near the solution [44] which yields guaranteed convergence within few iterations [12] and thus faster convergence [45]. The stepwise procedure for a parameterized, load control Newton-Raphson iterative method is provided in Algorithm 1 while the actual codes can be found in Appendix G.2.1.

Algorithm 1 Newton-Raphson iterative method for load control

```
1. Initialization
    (a) Define tolerance
    (b) Define load increments ($\lambda\mathbf{F}$)
2. Next load increment $t \geq 1$
    (a) Read $\lambda_t\mathbf{F}$ from incremental loading
    (b) Provide an estimate or initial guess: $\mathbf{u}_i = \mathbf{u}_{t-1}$
3. Next iteration $i \geq 1$
    (a) Find $\mathbf{K}_i$ or the matrix of first-order partial derivatives with respect to state variables in
        vector $\mathbf{u}_i$ as given by Equation (2.38)
    (b) Find system vector $\mathbf{f}(\mathbf{u}_i)$                                ▷ See Section 2.2.2
    (c) Find the residual or difference between $\lambda_t\mathbf{F}$ and $\mathbf{f}(\mathbf{u}_i)$ where $\mathbf{R}_i = \lambda_t\mathbf{F} - \mathbf{f}(\mathbf{u}_i)$ (Eq. (2.34))
    (d) Find $\Delta\mathbf{u}_i$ where $\Delta\mathbf{u}_i = \mathbf{K}_i^{-1}\mathbf{R}_i$ (Equation (2.40))
    (e) Update $\mathbf{u}_i$ by adding $\Delta\mathbf{u}_i$ to the current value of $\mathbf{u}_i$ where $\mathbf{u}_i = \mathbf{u}_i + \Delta\mathbf{u}_i$
    (f) Check if $\Delta\mathbf{u}_i$ is small with respect to $\mathbf{u}_i$
        i. if $\|\Delta\mathbf{u}_i\|/\|\mathbf{u}_i\| >$ tolerance then $\Longrightarrow$ go to next iteration 3
        ii. else if $\|\Delta\mathbf{u}_i\|/\|\mathbf{u}_i\| \leq$ tolerance then store $\mathbf{u}_i \Longrightarrow$ go to next load increment 2
```

2.2.2 Numerical Integration Implementations

As discussed in the previous section, the stiffness matrix $\mathbf{K}$ is tangent to the load-displacement curve for each linearized increment. From this definition it is inherent that $\mathbf{K}$ changes with respect to the increments. Therefore, non-linear FEM requires solving several integral terms for each increment within the process of calculations, for which numerical integration is used. For this purpose, it is advantageous to use numerical integration. For that matter, the nodal coordinates of the finite elements have been transformed into natural coordinates.

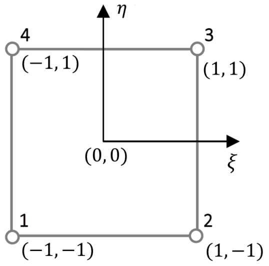


Figure 2.6: Natural coordinates

For the evaluation of two-dimensional integration problems, integration over the element domain Ω_e is required. Therefore, it is advantageous to transform the coordinate system into a natural coordinate system ($\xi - \eta$) [16]:

$$\int_{(\Omega_e)} g(\mathbf{X})\mathrm{d}A = \int_{-1}^{+1}\int_{-1}^{+1} g(\xi,\eta)\det\mathbf{J_e}\mathrm{d}\xi\mathrm{d}\eta\,, \tag{2.41}$$

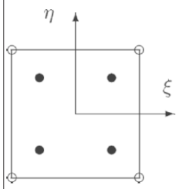
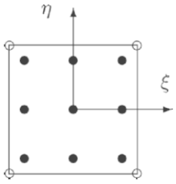
where $g(\xi, \eta)$ is the function to be integrated and $\mathbf{J_e}$ the *Jacobian* of the transformation to the reference configuration (i.e. the natural coordinate system). The

Jacobian is computed as follows for a two-dimensional problem:

$$\mathbf{J_e}(\xi,\eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} . \tag{2.42}$$

Integration is carried out by means of Gaussian quadrature. Due to their accuracy, Gauss rules are usually applied in finite element computations [16]. The weighting factors W_p and the coordinates of the quadrature points ξ_p and η_p are contained in Table 2.1 below for a Gauss quadrature up to a number of $n_p = 3 \times 3$ points. All two-dimensional Gauss quadrature for rectangular elements in Table 2.1 have been implemented in the two-dimensional non-linear FEM algorithms (see Appendix G.1).

Table 2.1: Two-dimensional Gauss quadrature for rectangular elements [16]

m	n_p	p	ξ_p	η_p	W_p	Position of points
1	1	1	0	0	4	
3	4	1	$-1/\sqrt{3}$	$-1/\sqrt{3}$	1	
		2	$+1/\sqrt{3}$	$-1/\sqrt{3}$	1	
		3	$-1/\sqrt{3}$	$+1/\sqrt{3}$	1	
		4	$+1/\sqrt{3}$	$+1/\sqrt{3}$	1	
5	9	1	$-\sqrt{3/5}$	$-\sqrt{3/5}$	25/81	
		2	0	$-\sqrt{3/5}$	40/81	
		3	$+\sqrt{3/5}$	$-\sqrt{3/5}$	25/81	
		4	$-\sqrt{3/5}$	0	40/81	
		5	0	0	64/81	
		6	$+\sqrt{3/5}$	0	40/81	
		7	$-\sqrt{3/5}$	$+\sqrt{3/5}$	25/81	
		8	0	$+\sqrt{3/5}$	40/81	
		9	$+\sqrt{3/5}$	$+\sqrt{3/5}$	25/81	

The coordinate transformation is now complete. However, the integral in Equation (2.41) is rather complicated. Therefore, it will be solved numerically which means that it will be transformed into a summation. By doing so, the following is obtained:

$$\int_{-1}^{+1}\int_{-1}^{+1} g(\xi,\eta) \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \approx \sum_{p=1}^{n_p} g(\xi_p,\eta_p) \det \mathbf{J_e}(\xi_p,\eta_p) W_p \, . \tag{2.43}$$

Similarly, three-dimensional integration problems can be evaluated by adding the third dimension using a $\xi - \eta - \zeta$ coordinate system. Now $\mathbf{J_e}$ is the three-dimensional *Jacobian*, computed as in [46]:

$$\mathbf{J_e}(\xi,\eta,\zeta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} . \tag{2.44}$$

The location and weights of the hexahedral elements are contained in Table 2.2 below for one ($1 \times 1 \times 1 = 1$), two ($2 \times 2 \times 2 = 8$) and three ($3 \times 3 \times 3 = 27$) integration points in each direction [17,18]. All three-dimensional integration point configurations for hexahedral elements in Table 2.2 have been implemented in the three-dimensional non-linear FEM algorithms (see Appendix G.1).

Table 2.2: Three-dimensional integration points for hexahedral elements [17, 18]

Scheme	Location (ξ_i, η_i, ζ_i) and any perturbation	Number	Weight
$1 \times 1 \times 1$	$(0, 0, 0)$	1	8
$2 \times 2 \times 2$	$\left(\pm\frac{1}{\sqrt{3}}, \pm\frac{1}{\sqrt{3}}, \pm\frac{1}{\sqrt{3}}\right)$	8	1
$3 \times 3 \times 3$	$\left(\pm\sqrt{\frac{3}{5}}, \pm\sqrt{\frac{3}{5}}, \pm\sqrt{\frac{3}{5}}\right)$	8	$\left(\frac{5}{9}\right)^3$
	$\left(0, \pm\sqrt{\frac{3}{5}}, \pm\sqrt{\frac{3}{5}}\right)$	12	$\left(\frac{8}{9}\right)\left(\frac{5}{9}\right)^2$
	$\left(0, 0, \pm\sqrt{\frac{3}{5}}\right)$	6	$\left(\frac{8}{9}\right)^2\left(\frac{5}{9}\right)$
	$(0, 0, 0)$	1	$\left(\frac{8}{9}\right)^3$

The coordinate transformation is now completed once more. Again, the resulting integral is complicated. Therefore, it will also be solved numerically which means a transformation into a summation. By doing so, the following is obtained:

$$\int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} g(\xi,\eta,\zeta)\det\mathbf{J_e}\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \approx \sum_{p=1}^{n_p} g(\xi_p,\eta_p,\zeta_p)\det\mathbf{J_e}(\xi_p,\eta_p,\zeta_p)W_p\,. \tag{2.45}$$

Now that the general format for numerical integration is presented, the numerical integration implementations and formulations used in the non-linear FEM algorithms in this research will be discussed. These implementations include the element stiffness matrices, internal force vectors and compliance values for both two-dimensional as well as three-dimensional formulations. For full elaborations, the reader is referred to Appendix B. Codes are provided in Appendix G.1.

- *Element Stiffness Matrices*

  In two-dimensional topology optimization codes the design domain is often discretized by rectangular finite elements, see for example [20, 33–36]. The rectangular finite elements used in these references are four-node bi-linear elements, see Figure 2.7 below. These elements are also referred to as bi-linear quadrilateral or Q4 elements.

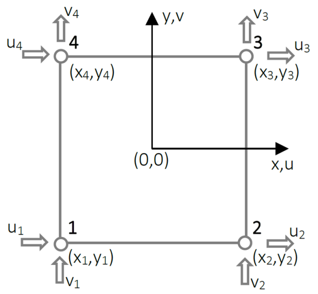


Figure 2.7: Four-node bi-linear element [5]

As shown by [5], the element stiffness matrix of such elements can be derived by exact integration solving the following integral using the strain-displacement matrix $\mathbf{B}$, the constant constitutive matrix $\mathbf{D}$ for a linear elastic material and constant thickness t:

$$\mathbf{K}_e = t\int_A \mathbf{B}^T\mathbf{D}\mathbf{B}\mathrm{d}A\,. \tag{2.46}$$

However, the constitutive matrix $\mathbf{D}$ becomes anisotropic and a function of strain in GMA. Hence, its final form is unknown beforehand and it is not a function of Young's modulus E and Poisson's ratio ν as it is in previous works on topology optimization [5,23,47]. Therefore, the constitutive matrix should in this case be taken from Equation (2.29) for plane stress conditions or from Equation (2.31) for plane strain. Since these constitutive matrices change with respect to loading and deformation (strain) for every element, the integral in Equation (2.46) should be evaluated for all elements in every increment of the analysis. Following from Equation (2.43), the integral for deriving the element stiffness matrix in Equation (2.46) can be evaluated numerically as:

$$\begin{aligned}\mathbf{K}_e &= t\int_{-1}^{+1}\int_{-1}^{+1}\mathbf{B}(\xi,\eta)^T\mathbf{D}\mathbf{B}(\xi,\eta)\det\mathbf{J_e}\mathrm{d}\xi\mathrm{d}\eta\\ &\approx t\sum_{p=1}^{n_p}\mathbf{B}(\xi,\eta)^T\mathbf{D}\mathbf{B}(\xi,\eta)\det\mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p})W_p\,.\end{aligned} \tag{2.47}$$

By comparing the exact stiffness matrix as obtained by [5,23,33] from Hooke's law and the stiffness matrix obtained from the FEM algorithm for a linear-elastic analysis, it was found that for $n_p = 4$ the element stiffness matrix is computed sufficiently accurate with a relative difference of

$$\frac{||\mathbf{K}_e^{numerical} - \mathbf{K}_e^{exact}||}{||\mathbf{K}_e^{exact}||} = 10^{-16}\,.$$

This is similar to the definition used by ABAQUS for "fully integrated", which refers to the number of Gauss points required to integrate the polynomial terms in an element's stiffness matrix exactly. Therefore, fully integrated, two-dimensional, quadrilateral elements use two integration points in each direction in ABAQUS [48].

When proceeding with three-dimensional topology optimization, a similar coding structure to the 99-line code by Sigmund [36] and the 88-line code by Andreassen et al. [33] was found in the 169-line program [6]. In this program the design domain is discretized by eight-node hexahedral elements, see Figure 2.8 below. These elements are also referred to as hex or (eight-node) brick elements.

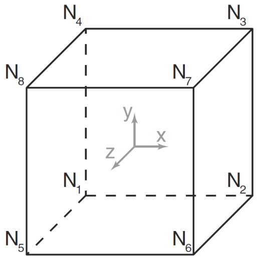


Figure 2.8: Eight-node hexahedral element [6]

Similar to the two-dimensional derivation, the element stiffness matrix of three-dimensional elements can be derived by means of exact integration as shown by [6]. In general this yields solving the following integral, as described in [17,46]:

$$\mathbf{K}_e = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} \mathrm{d}V \,. \tag{2.48}$$

As explained before, in GMA the constitutive matrix is a function of strain. Again, since the constitutive matrix changes with respect to the strain in every element, the integral in Equation (2.48) should be evaluated for all elements in every increment of the analysis. In this case the nodal coordinates of the finite elements are already written in natural coordinates, which is advantageous for performing numerical integration to derive the element stiffness matrix of the eight-node hexahedral elements.

Similar to the two-dimensional derivation, the integral for deriving the three-dimensional element stiffness matrix in Equation (2.48) can be transformed according to Equation (2.45). Numerical evaluation yields:

$$\begin{aligned}\mathbf{K}_e &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta,\zeta)^T \mathbf{C}\mathbf{B}(\xi,\eta,\zeta) \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \\ &\approx \sum_{p=1}^{n_p} \mathbf{B}(\xi,\eta,\zeta)^T \mathbf{C}\mathbf{B}(\xi,\eta,\zeta) \det \mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p},\zeta_\mathbf{p}) W_p \,. \end{aligned} \tag{2.49}$$

Analogous to the two-dimensional case, after comparing the stiffness matrices as obtained by [6] from Hooke's law and the FEM algorithm it was found that for two integration points in each direction ($2 \times 2 \times 2 = 8$) the element stiffness matrix is computed sufficiently accurate with a relative difference of

$$\frac{\|\mathbf{K}_e^{numerical} - \mathbf{K}_e^{exact}\|}{\|\mathbf{K}_e^{exact}\|} = 10^{-16}\,.$$

Coherent to the previous section this is also similar to the definition used by ABAQUS for "fully integrated" elements, referring to the required amount of Gauss points to exactly integrate the polynomial terms in the stiffness matrix of an element. Hence, fully integrated, three-dimensional, hexahedral elements use a $2 \times 2 \times 2$ array of integration points in ABAQUS [48].

- *Internal Force Vectors*

  In non-linear FEM, the elemental internal force vector can be obtained by solving the following integral using the strain-displacement matrix $\mathbf{B}$ and the stress vector $\boldsymbol{\sigma}$, as stated in for example [12, 38–40, 43, 49]:

$$\mathbf{f} = \int_V \mathbf{B}^T \boldsymbol{\sigma} \mathrm{d}V\,. \tag{2.50}$$

  In the case of linear-elasticity, Equation (2.50) simply reduces to $\mathbf{f} = \mathbf{K}\mathbf{u}$. Similar to the element's stiffness matrices being a function of strain in GMA, as a result the stress vector is also a function of strain. Therefore, the stress vector also changes with respect to the strain in every increment causing that the integral in Equation (2.50) should be evaluated for all elements in every increment of the analysis as well. Therefore, numerical integration is used once again to derive the elemental internal force vector of the elements, starting with the two-dimensional one:

$$\begin{aligned}\mathbf{f} &= t\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta)^T \boldsymbol{\sigma} \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \\ &\approx t\sum_{p=1}^{n_p} \mathbf{B}(\xi,\eta)^T \boldsymbol{\sigma} \det \mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p}) W_p\,. \end{aligned} \tag{2.51}$$

  Similarly, the three-dimensional elemental internal force vector can be derived:

$$\begin{aligned}\mathbf{f} &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta,\zeta)^T \boldsymbol{\sigma} \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \\ &\approx \sum_{p=1}^{n_p} \mathbf{B}(\xi,\eta,\zeta)^T \boldsymbol{\sigma} \det \mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p},\zeta_\mathbf{p}) W_p\,. \end{aligned} \tag{2.52}$$

- *Compliance*

  Compliance is equal to the overall strain energy [50]. Therefore, minimizing compliance is equivalent to maximizing global stiffness and thus related to minimizing strain energy [5, 23]. Besides, strain energy is equal to internal work [51, 52], such that minimizing compliance also means minimizing the work done by external forces [6]. However, the aforementioned is valid for load controlled analyses. For displacement controlled problems, maximizing structural stiffness means maximizing the total external work which is equal to the total strain energy under quasi-static condition [13].

  Often it may be convenient to have an expression for the strain energy per unit volume, also known as strain energy density. Strain energy density can be identified as the area underneath the stress-strain curve [53], in Figure 2.9 denoted by $W(\varepsilon)$. This yields solving the following integral [54] for every element:

$$u = \frac{U}{V} = \int_0^{\varepsilon_{ij}} \sigma_{ij} \mathrm{d}\varepsilon_{ij} \, , \tag{2.53}$$

  where U is the strain energy, V the volume, ε_{ij} the strain and σ_{ij} the stress. As a matter of fact, compliance (or strain energy) can also be computed via another approach, namely through strain energy. The implementation of this second approach is elaborated in Appendix B. During the development of the non-linear FEM algorithms, both approaches have been implemented to verify if similar results were obtained.

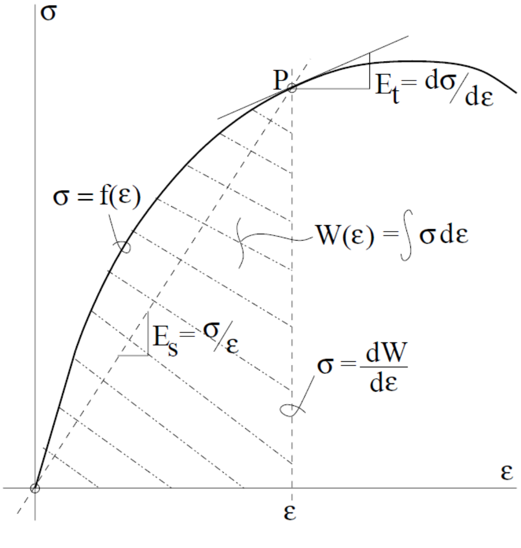

σ
P
$E_t = d\sigma/d\varepsilon$
σ = f(ε)
W(ε) = ∫ σ dε
$E_s = \sigma/\varepsilon$
$\sigma = \frac{dW}{d\varepsilon}$
ε
ε

Figure 2.9: Non-linear stress-strain relation including strain energy density [7]

  Since strain energy density is the amount of energy per unit volume, Equation (2.53) should be multiplied by the volume to obtain the actual strain energy [52]:

$$c = uV = V \int_0^{\varepsilon_{ij}} \sigma_{ij} \mathrm{d}\varepsilon_{ij} \, . \tag{2.54}$$

In the non-linear FEM, Equation (2.54) is implemented using a Riemann sum (midpoint rule) approximation. However, as discussed previously the stress vector also changes with respect to the strain in every increment causing that the integral in Equation (2.54) should be evaluated for all elements in every increment of the analysis as well. Therefore, numerical integration is used once more to derive the elemental compliance values. Again, starting with the two-dimensional problem:

$$\begin{aligned} c &= \sum_{i=1}^{N_i} \left[t \int_{-1}^{+1} \int_{-1}^{+1} \left(\sigma_{ij}^{i} - \frac{\mathrm{d}\sigma_{ij}}{2} \right) \mathrm{d}\varepsilon_{ij} \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \right] \\ &\approx \sum_{i=1}^{N_i} \left[t \sum_{p=1}^{n_p} \left(\sigma_{ij}^{i} - \frac{\mathrm{d}\sigma_{ij}}{2} \right) \mathrm{d}\varepsilon_{ij} \det \mathbf{J_e}(\xi_p, \eta_p) W_p \right] , \end{aligned} \tag{2.55}$$

where σ_{ij}^{i} is the stress tensor in the current increment, $\mathrm{d}\sigma_{ij}$ the incremental stress tensor (i.e. $\mathrm{d}\sigma_{ij} = \sigma_{ij}^{i} - \sigma_{ij}^{i-1}$) and $\mathrm{d}\varepsilon_{ij}$ the incremental strain tensor (i.e. $\mathrm{d}\varepsilon_{ij} = \varepsilon_{ij}^{i} - \varepsilon_{ij}^{i-1}$). Similarly, the compliance for the three-dimensional problem can be derived:

$$\begin{aligned} c &= \sum_{i=1}^{N_i} \left[\int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \left(\sigma_{ij}^{i} - \frac{\mathrm{d}\sigma_{ij}}{2} \right) \mathrm{d}\varepsilon_{ij} \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \right] \\ &\approx \sum_{i=1}^{N_i} \left[\sum_{p=1}^{n_p} \left(\sigma_{ij}^{i} - \frac{\mathrm{d}\sigma_{ij}}{2} \right) \mathrm{d}\varepsilon_{ij} \det \mathbf{J_e}(\xi_p, \eta_p, \zeta_p) W_p \right] . \end{aligned} \tag{2.56}$$

The advantage of the strain energy density approach is that it is independent of the type of finite element analysis (since the strains and stresses are calculated anyway) and thus that, again, the input for Equations (2.55) and (2.56) is readily available. Another difference between the two approaches is that the strain energy approach (see Appendix B) computes compliance *globally*, i.e. it is based on the global force and displacement vectors, and that the approach presented here (strain energy density) computes compliance *locally*, i.e. it is based on the elemental stress and strain tensors. Therefore, the strain energy density approach is used in the non-linear FEM modules for the topology optimization algorithms, since these require (local) elemental compliance values as an input to solve the well defined topology optimization problem of compliance minimization as addressed in [6,33,36], among others (see Appendix G.1). In the particular case when Hooke's law is valid, as is the case in the references mentioned, the formulation for compliance simply reduces to $c = \mathbf{u}^T\mathbf{K}\mathbf{u} = \mathbf{F}^T\mathbf{u}$.

2.3 Topology Optimization

Topology optimization is a conceptual design tool which aims to find the most optimal material distribution of a structure within a certain predefined design domain or space [15,19,20]. Besides a given design domain, the boundary conditions (i.e. the load and support conditions) are defined. Therefore, topology optimization may be used to attain structural designs with higher stiffness or strength-to-weight ratios [21,22], for example, by avoiding the presence of redundant structural elements within the design domain. Subsequently, topology optimization methods could be beneficial for the construction industry in its process to design lighter and less resource intensive structures [5,23]. In addition, recently popular Additive Manufacturing (AM) techniques, such as 3D (concrete) printing, appreciate the importance of topology optimization since it can potentially provide the ability to manufacture (porous) structural designs having complicated geometries [9,21].

Nowadays, a large variety of methods is available that address topology optimization design problems in various engineering fields. However, the present research is limited to the *density-based approach* to topology optimization, which is popularly applied [20] and therefore one of the most widely used approaches [19]. In this approach, the design domain is generally discretized by Finite Elements (FE) and handled through the FEM [37]. A so-called density x_e is assigned to each element, also referred to as the design variable, which controls the distribution of material within the design domain ranging from solid ($x_e = 1$) to void ($x_e = 0$) [55]. Intermediate densities (grey elements) are permitted as well to obtain a continuous optimization problem, but these are penalized by a penalization power: the mechanical properties of each element, i.e. the stiffness tensor, are determined using a power-law interpolation function between void and solid [6]. This is what relates the FEM with the topology optimization algorithm. In other words, elements get mechanically activated in correspondence to their density values: elements close to unit density contribute to the global stiffness matrix while elements being void are practically unresponsive. Using the "power-law approach", the penalization power does only affect (i.e. reduce) intermediate densities while leaving solid and void elements unaffected. Therefore, the penalization power may *implicitly* penalize elements having intermediate density values by which it steers the optimization towards a black-and-white configuration as much as possible [5,23].

A highly favored [21,22] and therefore one of the most utilized penalization techniques [37] is usually referred to as the *Solid Isotropic Material with Penalization* (SIMP) method [6]. This method is based on a heuristic relation between an element's density x_e and Young's modulus E_e (i.e. stiffness) and is described as

$$E_e = E_e(x_e) = x_e^p E_0 \quad \text{with} \quad 0 < x_{min} \leq x_e \leq 1\,, \tag{2.57}$$

where x_e is the density of element e, p is the penalization power (usually $p \geq 3$), E_0 the Young's modulus of solid material and x_{min} a small, non-zero value to prevent singularity of the stiffness matrix (typically 10^{-3}) [5,23,36]. However, a modified SIMP approach is given as in [6,33] by

$$E_e = E_e(x_e) = E_{min} + x_e^p\,(E_0 - E_{min}) \quad \text{with} \quad x_e \in [0,1]\,, \tag{2.58}$$

where E_{min} is a very small, non-zero stiffness assigned to void elements, again to prevent the stiffness matrix from becoming singular (typically 10^{-9}) [21, 22]. This modified SIMP approach makes it unnecessary to set a lower bound x_{min} for the element density x_e since E_{min} already serves this purpose. Nevertheless, an advantage of the modified approach in Equation (2.58) is the independency between a void element's stiffness E_{min} and the penalization power p [6]. However, one could argue that the value for E_{min} in the modified SIMP approach to some extend inherits from the SIMP method in Equation (2.57), since in pioneering works (such as [33, 36]) a solid material's Young's modulus is often set equal to one ($E_0 = 1$), resulting in $E_{min} = E_e(x_{min}) = 1(10^{-3})^3 = 10^{-9}$.

Most applications of topology optimization are focused on isotropic materials and therefore use a classical continuum mechanics approach with a given elastic stiffness tensor, without considering dependency on its microstructure [9, 29]. However, as addressed earlier in section 2.2.2, in GMA the constitutive matrix or stiffness tensor generally becomes anisotropic and is a function of strain, hence its final form is unknown beforehand. Therefore, in contrast to isotropic materials, the stiffness tensor is not a function of Young's modulus E and Poisson's ratio ν (or alternatively Lame's constants). As a result, penalization cannot be performed by means of the SIMP method or modified SIMP approach. To this end, an alternative penalization strategy is developed in this research, which will be addressed in a later section. Nevertheless, this strategy is inspired by the SIMP method and is formulated in a similar manner for which the SIMP approach is included in this literature review.

In addition to the above, most topology optimization formulations are based on the linear-elastic assumption, using a linear material model. While this assumption is applicable to a large range of problems it is not always valid [55], because in practice materials often exhibit non-linear behaviour [56]. However, as stated in the latter reference, the main challenge of implementing a non-linear material model into topology optimization algorithms lies in the computational efficiency of the sensitivity analysis. This issue will be addressed in a later section as well. Moreover, it should be clarified that the non-linear FEM codes converge for all load increments before returning the converged solution of the final load increment to the topology optimization algorithm. Only afterwards the design variables are updated via the topology optimization algorithm, similar to other implementations of material non-linearities such as [13, 39, 49, 57]. The presented topology optimization algorithms accommodate material non-linearities in their structural analyses.

This last section related to literature initiates by introducing two popular topology optimization problems, namely compliance minimization under a volume constraint and stress-constrained volume minimization. Subsequently, the Proportional Topology Optimization (PTO) method is discussed, including its pros and cons.

2.3.1 Topology Optimization Methodologies

Two popular topology optimization methodologies or problems are the minimum compliance problem, which aims to minimize structural compliance for a given structural volume (often referred to as the volume fraction), and the stress-constrained problem, which aims to minimize the structural volume while satisfying a certain stress constraint [21,22]. However, the latter problem has the implicit aim to utilize the material to its maximum extent. A vast majority of earlier topology optimization researches address the compliance minimization design problem [15, 20, 37]. By using the mean compliance as the inverse of structural stiffness, it is intended to enhance the overall stiffness of a structure. Accordingly, the compliance minimization problem for non-linear material behaviour in this research reads as follows:

$$\begin{cases} \min c = V \int_0^{\varepsilon_{ij}} \sigma_{ij} \mathrm{d}\varepsilon_{ij} \,, \\ \text{subject to} \begin{cases} \int_V \mathbf{B}^T \boldsymbol{\sigma} \mathrm{d}V = \mathbf{f} \,, \\ \frac{V(\mathbf{x})}{V_0} = f \quad \text{with} \quad V(\mathbf{x}) = \sum_{e=1}^{N_e} x_e v_e \,, \\ 0 < x_{min} \leq x_e \leq 1 \,, \end{cases} \end{cases} \tag{2.59}$$

where $V(\mathbf{x})$ is the total structural volume where all densities x_e are combined in vector $\mathbf{x}$, V_0 the volume of the entire design domain, f the prescribed volume fraction (i.e. the constraint), N_e the number of elements in the discretized design domain (i.e. the FE mesh) and v_e the volume of element e. However, a major drawback of the compliance problem is that it does not guarantee that in the final topology none of the stresses exceed their limits such that significant post-processing is still required [5, 23] since most materials in real structural design problems may suffer from strength failure [15]. Therefore, stress analysis is more significant to structural designers [21,22] since it is paramount from an engineering point of view [20] to utilize the material to its full potential. From the viewpoint of classical failure criteria, the stress-constrained problem can be generally described by the following model:

$$\begin{cases} \min V(\mathbf{x}) = \sum_{e=1}^{N_e} x_e v_e \,, \\ \text{subject to} \begin{cases} \int_V \mathbf{B}^T \boldsymbol{\sigma} \mathrm{d}V = \mathbf{f} \,, \\ \mathbb{F}\left(\sigma_{ij}(\mathbf{x})\right) \leq 0 \,, \\ 0 < x_{min} \leq x_e \leq 1 \,, \end{cases} \end{cases} \tag{2.60}$$

where, in addition to the previous nomenclature, $\mathbb{F}\left(\sigma_{ij}(\mathbf{x})\right)$ is a general yield function or failure criterion (i.e. the constraint) as a function of the stress tensor σ_{ij} that depends on the density distribution $\mathbf{x}$ for which no failure will occur if it is less or equal to zero. Most available stress-constrained topology optimization problems make use of the von Mises stress criterion as a failure constraint, which is more suitable for metals rather than cementitious materials [37], such as concrete. In that case, the general failure criterion in Equation (2.60) could be replaced by the classical von Mises stress criterion. Please note that contrary to the minimum compliance problem, the stress-constrained problem does not optimize stiffness in any way. In fact, when considering material non-linearities the stiffness can even reduce (significantly) when approaching the limit stress.

However, materials such as concrete are characterized by an increase in shear strength as a result of increasing hydrostatic pressure. These pressure-dependent materials typically exhibit different stress limits in tension and compression for which the von Mises criterion is not appropriate. One of the simplest available plasticity yield models that is able to manage asymmetric stress limits in tension and compression is the Drucker-Prager criterion, defined in terms of stress invariants [15]. This criterion has been used in a previous master graduation research at the Eindhoven University of Technology (TU/e) on the topic of topology optimization for concrete structures [5, 23] and is elaborately discussed in Appendix F. The (classical) failure criterion predicts how close a material point is to failure according to its stress state and is the most frequently utilized yield criterion in the field of geotechnical engineering. The Drucker-Prager criterion will be used to show the form of the yield surface, since it is recognized to provide numerical predictions which are in good agreement with experimental results for identifying the failure behaviour of, in particular, concrete. Besides, the criterion will be used for comparison reasons as well as relating this research to the previous master graduation thesis [5]: the topology optimization algorithms used in the present research have emerged from the ones presented in there.

Materials like concrete can be characterized by means of a compressive and tensile strength, f_{cy} and f_{ty}, respectively. For that matter, the two-dimensional Drucker-Prager yield surface can be obtained for a plane stress situation (i.e. $\sigma_3 = 0$), which can be seen in Figure 2.10. According to [5, 23], a so-called strength ratio $\gamma = f_{cy}/f_{ty}$ can be defined by which the areas of the compressive and tensile zones in the two-dimensional yield surface can be altered. However, the strength ratios used in that research have been limited ($1/3 \leq \gamma \leq 3$) to prevent problems in the topology optimization algorithms.

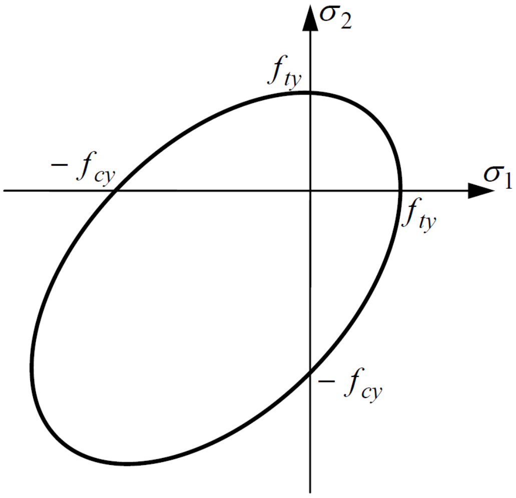


Figure 2.10: Drucker-Prager yield surface for a plane stress situation [8]

2.3.2 Proportional Topology Optimization

The research field of topology optimization comprises a variety of methods that are available in common literature. However, in view of a previous research that investigated what topology optimization method obtains the most optimal results for materials with unequal stress limits in tension and compression [5, 23], the current research should consider the Traditional Topology Optimization (TTO) method for stress-constrained volume minimization. To be specific, from the investigated methods TTO was found to be the only one to produce structures having more compressive elements compared to tensile elements, which is intuitively more logical for materials like concrete. Additionally, it has been concluded that the TTO method provides solutions with a lower structural volume compared to other topology optimization methods for materials governed by asymmetric stress limits. Therefore, it is recommended to use TTO as the topology optimization method for materials like concrete. Besides, this is currently the most frequently utilized density-based topology optimization method. However, as stated by Jia, Misra, Poorsolhjouy and Liu in [9], the derivative of the objective function with respect to the design variable cannot be obtained analytically for the multi-scale material model (the GMA derived in Section 2.1). Since TTO solves topology optimization problems by means of mathematical solvers that rely on these so-called *sensitivities* or *gradients* of the objective- and constraint functions [5, 23], an alternative approach to obtain sensitivity information will be addressed later.

On that score, a simple and efficient non-sensitivity method that has also been considered in the previous research [5,23], called Proportional Topology Optimization (PTO), has been used here instead for the first implementations of the non-linear FEM using the multi-scale model into topology optimization algorithms for both the compliance minimization and stress-constrained problems. Besides, sensitivity information is accompanied by additional computations yielding a larger computational demand, is even more difficult to derive analytically for non-linear analysis (as in this research) and may bring along implementation concerns, i.e. its computation could become impracticable due to a large number of imposed (stress) constraints [21, 22]. Despite this, it should be noted that sensitivity information is helpful to accelerate and steer a topology optimization algorithm. In conclusion, there is a trade-off between sensitivity and non-sensitivity methods as regards their computational or implementation's efficiency and complexity.

According to the aforementioned research [5, 23] and references therein [21, 22], PTO is a relatively recent topology optimization method that distributes the material in the design domain proportional to the element's objective function values without the use of sensitivity information. Notably, PTO algorithms only impose constraints *globally*, thus on the system as a whole. Therefore, the method *globally* controls the material's proportional distribution to the elements, for which it can be considered as an Optimality Criteria (OC) method [21]. Even though it is recognized that the PTO method is very heuristic, this is what makes it easy to understand and implement. In addition, it was found in [5, 23] that even though TTO yields more optimal results it is also more time-consuming than PTO, up to 264 times, which makes it a good method to initially implement the non-linear

FEM using the GMA into. Similar to TTO, PTO utilizes the modified SIMP approach for a linear-elastic material model as described in Equation (2.58).

The first step in both the compliance minimization and stress-constrained problems using the PTO method, hereafter referred to as the PTOc and PTOs algorithms, respectively, is the determination of the target material amount (m_t). Since the target material amount for the compliance minimization problem is equal to a fixed, predefined volume fraction (i.e. $m_t = N_e \times f$), the PTOc algorithm does only have to calculate this once at the beginning of the optimization process. However, for the stress-constrained problem, in which it is the objective to minimize the structural volume, the target material amount should be determined at the beginning of each iteration. In that case, the target material will become the new material amount. Worded differently, the current material amount (m_c) is going to be updated to the target material amount. If an element's stress value violates the failure criterion, i.e. $\mathbb{F}\left(\sigma_{ij}\right) > 0$, the current material amount is increased by a so-called "material move amount" and vice versa. This amount is scaled with the number of elements in the design domain which in the present work, contrary to the implementations by [5, 21–23], is taken as $0.01 \times N_e$:

$$m_t = \begin{cases} \left(\sum_{e=1}^{N_e} x_e + 0.01 N_e\right) v_e & \text{if} \quad \mathbb{F}\left(\sigma_{ij}\right) > 0\,, \\ \left(\sum_{e=1}^{N_e} x_e - 0.01 N_e\right) v_e & \text{if} \quad \mathbb{F}\left(\sigma_{ij}\right) \leq 0\,. \end{cases} \tag{2.61}$$

Afterwards, both algorithms distribute the target material amount to the design domain, i.e. the FE mesh, which can only be done iteratively for reasons to be explained. In the iterative inner loop, the material amount to be distributed will be referred to as the remaining material amount (m_r) which is initially set equal to the remaining material amount. The distribution is executed proportionally to either the elemental compliance values c_e (PTOc) or the element stress values $\boldsymbol{\sigma}_e$ (PTOs) and is extended to a power q, as suggested by Biyikli and To [21, 22], to enhance the performance of the algorithms:

$$x_e^{opt} = \frac{m_r}{\sum_{e=1}^{N_e} c_e^q v_e} c_e^q\,, \qquad\qquad x_e^{opt} = \frac{m_r}{\sum_{e=1}^{N_e} \boldsymbol{\sigma}_e^q} \boldsymbol{\sigma}_e^q\,, \tag{2.62}$$

where x_e^{opt} are the optimized elemental densities and q is the proportion exponent. It appears that the proportional distribution can theoretically assign density values that exceed the density limits, i.e. values smaller than zero or larger than one. To prevent this from happening, the density limits are enforced by trimming the distributed material amount. Therefore, the actual material amount (m_a) differs from the target material amount which is the reason why the iterative inner loop is required. As a result, the remaining material amount should be determined several times until it is sufficiently small.

The last step in the PTO algorithms is updating the elemental densities. This is done through linearly blending the element densities from the previous iteration with the optimized ones in Equation (2.62) from the current iteration:

$$x_e^{new} = \alpha x_e^{prev} + (1 - \alpha) x_e^{opt}\,, \tag{2.63}$$

where x_e^{new} is the new elemental density for the next iteration, x_e^{prev} the element density from the previous iteration and α the history coefficient which has no unit and determines the ratio of dependency of the next iteration's densities on the previous and current iterations. For illustration: when $\alpha = 0$ the portion of Equation (2.63) related to the previous iteration vanishes, meaning that the new densities are solely dependent on the current iteration.

However, it should be noted that Biyikli and To [22] have performed a parametric study of the proportion exponent q and history coefficient α for both the PTOc and PTOs algorithms for linear analysis. For the PTOc algorithm $\alpha = 0.5$ and $q = 1$ were found to be reasonable values, which results in exclusion of the proportion exponent in Equation (2.62). The PTOs algorithm found the smallest volume for the settings $\alpha = 0$ and $q = 2$, which results in exclusion of the history coefficient in Equation (2.63). Therefore, combining Equations (2.62) and (2.63) with the above parameters for both algorithms yields

$$x_e^{new} = \frac{1}{2} x_e^{prev} + \frac{1}{2}\left(\frac{m_r}{\sum_{e=1}^{N_e} c_e v_e} c_e \right) , \qquad x_e^{new} = \frac{m_r}{\sum_{e=1}^{N_e} \boldsymbol{\sigma}_e^2} \boldsymbol{\sigma}_e^2 . \tag{2.64}$$

For defining a suitable stress failure state within a fictitious porous microstructure of an intermediate-density element, a simple power-law relationship between this microscopic *local* stress and the macroscopic stress at the material points is assumed as in [5, 15, 23, 58]. This reads as

$$\boldsymbol{\sigma}(x_e) = \frac{\bar{\boldsymbol{\sigma}}(x_e)}{x_e^q} , \tag{2.65}$$

where $\bar{\boldsymbol{\sigma}}(x_e)$ is the macroscopic stress related to the deformations (i.e. strains), which for the SIMP approach becomes $\bar{\boldsymbol{\sigma}}(x_e) = x_e^p \mathbf{C} \bar{\boldsymbol{\varepsilon}}(x_e)$ with $q \geq 1$. For linear-elasticity, it is often mentioned that the power q should be chosen to be equal to the penalization power p for micro-structural considerations. As a result, the density-related term in the *local* stress formulation vanishes for the SIMP method. Combining Equation (2.65) with the above, the *local* stress assuming solid material properties becomes

$$\boldsymbol{\sigma}(x_e) = \frac{x_e^p}{x_e^q} \mathbf{C} \bar{\boldsymbol{\varepsilon}}(x_e) = x_e^{p-q} \mathbf{C} \bar{\boldsymbol{\varepsilon}}(x_e) = \mathbf{C} \bar{\boldsymbol{\varepsilon}}(x_e) \quad \text{for} \quad p = q . \tag{2.66}$$

The relationship given in Equation (2.66) is referred to as "local stress interpolation" and has been successfully implemented in various stress-constrained problems in topology optimization. This *local* stress is taken into account in the failure criterion.

Even though failure criteria defined in terms of this *local* stress measure are consistent with micro-structural considerations, these still encounter problems with low-density elements. These difficulties arise in low-density elements having a significantly lower stiffness (due to the SIMP method) which leads to very high, non-negligible strains and thus stresses to occur. Therefore, the stress values in low-density elements may violate the failure criterion which causes their stress constraints to prevent complete removal (i.e. reducing the element density to the minimal value (x_{min})). This is a well-known problem in stress-constrained topology optimization that is referred to as the ”stress singularity phenomenon”, usually referred to as the *singularity phenomenon* for short [5, 15, 23, 35, 58]. To circumvent this phenomenon, the *local* stress constraints are often relaxed for elements having a low density. There are various relaxation techniques available, but the present research limits itself to the same approach used in a previous research [5, 23], which utilized the epsilon- or ε-relaxation technique. Applying this relaxation technique for a general, normalized failure criterion [58] leads to the following expression for the stress constraint function for each element e, using the same terminology as in [5, 23]:

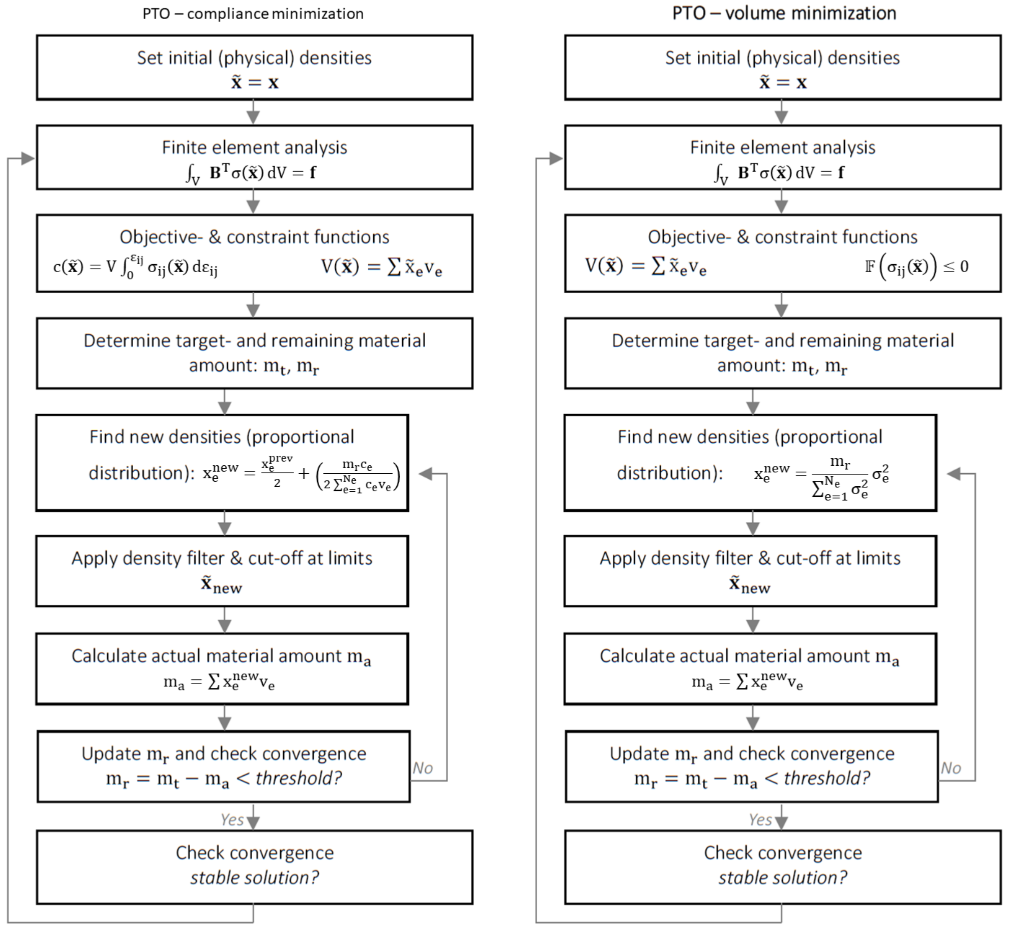
$$R_e(\boldsymbol{\sigma}(x_e), x_e) = \frac{\|\boldsymbol{\sigma}(x_e)\|}{\|\boldsymbol{\sigma}^{lim}\|} - 1 - \frac{\varepsilon\,(1 - x_e)}{x_e} \leq 0\,, \tag{2.67}$$

where $\|\boldsymbol{\sigma}(x_e)\|$ is the chosen stress measure (e.g. von Mises or Drucker-Prager) and $\|\boldsymbol{\sigma}^{lim}\|$ is the maximum allowable stress in the same direction. The third term in Equation (2.67) represents the relaxation coefficient in which $\varepsilon > 0$ is the relaxation parameter, a prescribed small positive real number that controls the amount of relaxation. The portion $(1 - x_e)$ ensures that no relaxation occurs for solid elements ($x_e = 1$), such that Equation (2.67) reduces to the original, normalized stress constraint regardless of the relaxation parameter. However, for low-density elements the failure criterion gets significantly relaxed, which implies enlargement of the yield surface. For larger values of the relaxation parameter, the stress constraints of low-density elements are further relaxed. In the current research, the relaxation parameter is taken as $\varepsilon = \sqrt{x_{min}}$, which is similar to the previous research [5, 23] as well as the publication by Luo & Kang [15].

Furthermore, topology optimization methods are prone to numerical issues like mesh-dependency, checkerboard patterns and local minima [6]. To prevent these difficulties, regularization techniques have been proposed of which density filters are most usually applied. A density filter rescales the density of each element within the design domain (i.e. the FE mesh) with respect to its neighbor’s densities [37]. As a result, filtered *physical densities* ($\tilde{x}_e$) are originated that are different from the *design variables* (x_e). A standard filter density function filters the design variables by taking a weighted average of the local densities within a certain filter radius [5, 15, 21–23, 33, 37]:

$$\tilde{x}_i = \frac{\sum_{j=1}^{N_e} H_{ij} x_j}{\sum_{j=1}^{N_e} H_{ij}} \quad \text{with} \quad H_{ij} = \begin{cases} \frac{r_0 - r_{ij}}{r_0} & \text{if } r_{ij} < r_0\,, \\ 0 & \text{if } r_{ij} \geq r_0\,, \end{cases} \tag{2.68}$$

where $\tilde{x}_i$ is the (filtered) *physical density* of element i, H_{ij} the filtering weight of elements i and j, x_j the (non-filtered) design variable of element j, r_{ij} the Euclidean distance between the centers of elements i and j and r_0 the filter radius. The weight H_{ij} is inversely proportional to the distance between an element and its neighbors, such that the density filter essentially does nothing more than local averaging while preserving the volume [21, 22]. The *physical densities* obtained from Equation (2.68) are utilized in the non-linear FEM algorithms. Since PTO is a non-sensitivity method, it is obvious that this method can only utilize density filtering. The new elemental densities that are obtained from the proportional distribution in Equation (2.64) are therefore filtered using the basic density filter given in Equation (2.68). As a result, the density filter ensures consistency in the topology optimization algorithm, since the *physical densities* are used for the non-linear FEM algorithm while the *design variables* are utilized for the optimization process. In Figure 2.11, the solution procedures for (a) the compliance minimization problem (PTOc) as well as (b) the stress-constrained problem (PTOs) using the PTO method are schematically presented.


PTO – compliance minimization
Set initial (physical) densities $\tilde{\mathbf{x}} = \mathbf{x}$
Finite element analysis $\int_V \mathbf{B}^T \sigma(\tilde{\mathbf{x}})\,dV = \mathbf{f}$
Objective- & constraint functions
$c(\tilde{\mathbf{x}}) = V \int_0^{\varepsilon_{ij}} \sigma_{ij}(\tilde{\mathbf{x}})\,d\varepsilon_{ij}$
$V(\tilde{\mathbf{x}}) = \sum \tilde{x}_e v_e$
Determine target- and remaining material amount: m_t, m_r
Find new densities (proportional distribution): $x_e^{new} = \frac{x_e^{prev}}{2} + \left(\frac{m_r c_e}{2\sum_{e=1}^{N_e} c_e v_e}\right)$
Apply density filter & cut-off at limits $\tilde{\mathbf{x}}_{new}$
Calculate actual material amount m_a
$m_a = \sum x_e^{new} v_e$
Update m_r and check convergence
$m_r = m_t - m_a <$ *threshold?*
No
Yes
Check convergence
stable solution?
PTO – volume minimization
Set initial (physical) densities $\tilde{\mathbf{x}} = \mathbf{x}$
Finite element analysis $\int_V \mathbf{B}^T \sigma(\tilde{\mathbf{x}})\,dV = \mathbf{f}$
Objective- & constraint functions
$V(\tilde{\mathbf{x}}) = \sum \tilde{x}_e v_e$
$\mathbb{F}\left(\sigma_{ij}(\tilde{\mathbf{x}})\right) \leq 0$
Determine target- and remaining material amount: m_t, m_r
Find new densities (proportional distribution): $x_e^{new} = \frac{m_r}{\sum_{e=1}^{N_e} \sigma_e^2} \sigma_e^2$
Apply density filter & cut-off at limits $\tilde{\mathbf{x}}_{new}$
Calculate actual material amount m_a
$m_a = \sum x_e^{new} v_e$
Update m_r and check convergence
$m_r = m_t - m_a <$ *threshold?*
No
Yes
Check convergence
stable solution?

(a) Schematic overview of PTOc

(b) Schematic overview of PTOs

Figure 2.11: Schematic overview of Proportional Topology Optimization methods

The PTO method has been successfully applied to the stress-constrained problem for linear-elastic analysis of materials exhibiting symmetric stress limits, e.g. using the von Mises failure criterion [21, 22]. However, for pressure-dependent materials having a lower stress limit in tension compared to compression (such as concrete), it was found by [5, 23] that the method tends to prefer tensile elements over compressive elements resulting in solutions with a higher structural volume to prevent the exceedance of the material's tensile stress limit. The reason for this has been explained in [5, 23] by means of a numerical example using the Drucker-Prager yield function. Nevertheless, a brief and general illustration will be provided here for convenience using the same failure criterion. Consider two two-dimensional plane stress elements, I and II, that are made of a material having a strength ratio $1 < \gamma \leq 3$ such that it has a lower tensile stress limit compared to its compressive stress limit. Suppose element I is under a pure tensile load (no shear) and element II is subjected to a pure compressive load of the exact same magnitude (but in opposite direction) as element I, such that both stress states can be visualized within the Drucker-Prager yield surface in the two-dimensional principal stress space. This is depicted in Figure 2.12 in which the dotted line goes through the stress states of the elements and the origin, representing the direction of loading.

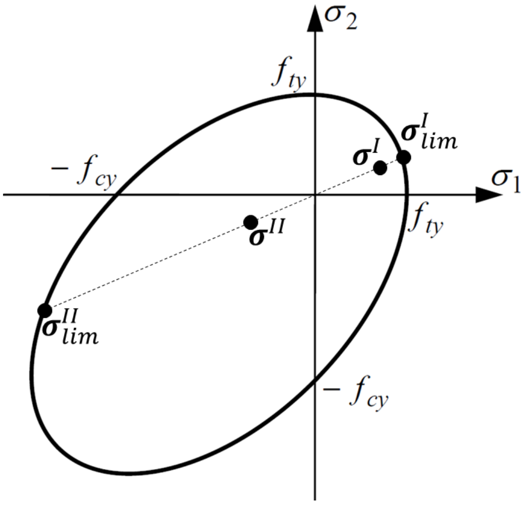

σ_2
f_{ty}
$\boldsymbol{\sigma}^I_{lim}$
$\boldsymbol{\sigma}^I$
$-f_{cy}$
σ_1
f_{ty}
$\boldsymbol{\sigma}^{II}$
$\boldsymbol{\sigma}^{II}_{lim}$
$-f_{cy}$

Figure 2.12: Stress states of elements I and II in principal stress space

The Drucker-Prager yield criterion returns the ratio between the obtained stress state and the limit stress state. In the case of Figure 2.12, it describes the ratios $\|\boldsymbol{\sigma}^I\|/\|\boldsymbol{\sigma}^I_{lim}\|$ and $\|\boldsymbol{\sigma}^{II}\|/\|\boldsymbol{\sigma}^{II}_{lim}\|$. It can be clearly seen that this ratio is larger for the tensile element (I) compared to the compressive element (II), from which it can be concluded that the stress state of element I is closer to yielding than element II. Since the PTOs algorithm distributes the material proportionally to the constraint function values (i.e. the failure criterion), the tensile element will receive a larger portion (if not all) of the material to be distributed.

In view of the observation that the TTO method was found to be most optimal for materials having asymmetric stress limits in tension and compression and the critical flaw of the PTO method for pressure-dependent materials as described above, an alternative approach to obtain sensitivity information has been considered in Appendix F. Because the derivative of the objective function with respect to the design variable cannot be obtained analytically for the multi-scale material model, numerical sensitivity information could be obtained from the GMA and used in the TTO method. However, the implementation of numerical sensitivity information in combination with the mathematical solvers that this method uses will be left for follow up work with respect to the reserved time for the current research. For more information on the TTO method, numerical sensitivity information and a preliminary implementation, the reader is referred to Appendix F and references therein.

3 Methodology

The methodology section presents how the theories, methods and techniques as described in Section 2 are applied in the current research. The first part describes how the multi-scale material model has been incorporated into the non-linear FEM, together with several validations. The second part discusses the implementation of the developed non-linear FEM codes into topology optimization algorithms, as well as an alternative penalization approach and several microscopic constitutive relationships. The third and final part gives a description of the first implementations of the developed non-linear FEM using the multi-scale material model into proportional topology optimization algorithms for both the compliance minimization and stress-constrained problems. All the codes and algorithms used and referred to throughout this work are modelled using the platform `MATLAB`.

3.1 Non-Linear Finite Element Method & Multi-Scale Model

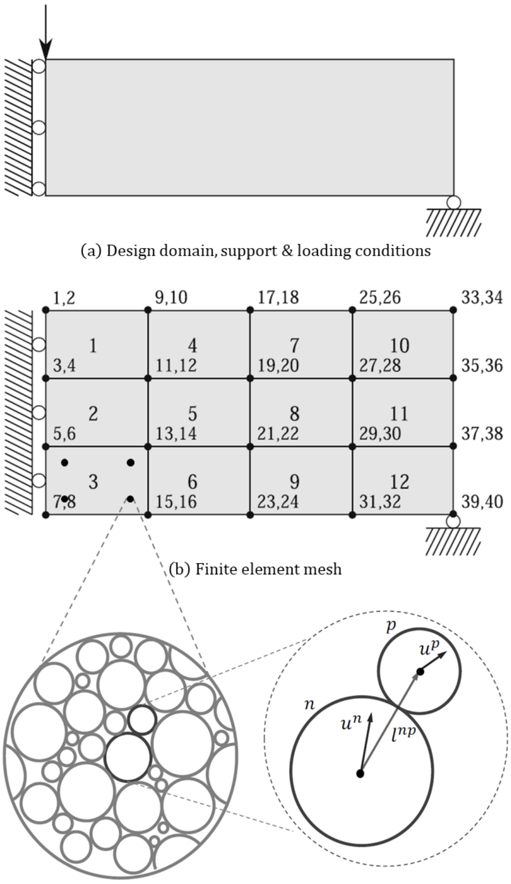
The first step in this research was to develop a non-linear FEM code that incorporated the multi-scale material model, namely the GMA as discussed in Section 2.1. With regard to a previous master graduation research conducted at the TU/e on the topic of topology optimization for concrete structures using `MATLAB` codes having a linear-elastic FEM [5, 23], the FEM portions of the codes used in there have been taken as a starting point. Most of these codes strongly inherit from the publicly available 99-line and 88-line `MATLAB` implementations for two-dimensional topology optimization by Sigmund [36] and Andreassen et al. [33], respectively. However, when proceeding with three-dimensional topology optimization, a similar coding structure was found in the 169-line program by Liu & Tovar [6]. Besides, GMA had already been coded in `MATLAB` by Poorsolhjouy [14].

In this section, it is first discussed in general how FEA works and how this is numerically implemented using the platform `MATLAB`. This is followed by the performance of several FE formulation validations by means of numerical integration through comparison with commercial FE software. Finally, it is explained how the multi-scale material model is incorporated into the non-linear FEM codes that have been developed.

3.1.1 Finite Element Analysis & Numerical Implementations

Each FEA starts with defining a structure to consider. This is done through providing its corresponding design domain, support and loading conditions (see Figure 3.1(a)). As discussed in Section 2.2.2, in the two-dimensional codes the design domain is assumed to be rectangular and discretized by rectangular finite elements, composing the FE mesh. In this case, each FE has four nodes that have two Degrees of Freedom (DOFs). Identical to the two-dimensional `MATLAB` implementations by Sigmund [36] and Andreassen et al. [33], both the elements and nodes are numbered column-wise from left to right while the odd and even DOFs relate to the horizontal and vertical displacements, respectively. Element numbers and the numbers DOFs are given in Figure 3.1(b). As specified in Section

2.2.2, four material or integration points adequately describe the kinematics of the material in two-dimensional finite elements. To incorporate the micromechanics of the material, we use the multi-scale material model to implement this behaviour. In this model, all grain interactions contribute to the macroscopic response of the material (see Figure 3.1(c)). An illustration of the two-dimensional FEA including its numerical implementation is given in Figure 3.1 for a classical topology optimization problem, namely the Messerschmitt-Bölkow-Blohm (MBB) beam that is usually referred to as the (half) MBB beam.



(a) Design domain, support & loading conditions

(b) Finite element mesh

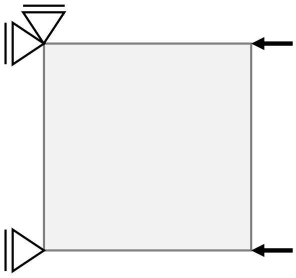
(c) Microstructure of material point & contact of two neighbor grains

Figure 3.1: Two-dimensional Finite Element Analysis & numerical implementation

Accordingly, Section 2.2.2 states that in the three-dimensional codes the FE mesh is composed of eight-node hexahedral elements. Now, each element has eight nodes having three DOFs that correspond to linear displacements in $x - y - z$ directions. Similar to the 169-line `MATLAB` program [6], the nodes are identified with a node ID and ordered column-wise top-to-bottom, left-to-right and back-to-front. For a more detailed description of the ordering and numerical implementation of the three-dimensional codes, the reader is referred to the original publication by Liu & Tovar [6]. Similarly as discussed in Section 2.2.2, each three-dimensional FE contains eight material points to adequately describe the material behaviour.

3.1.2 Linear-Elastic Element Formulations & Validations

In order to validate the element formulation as presented in Section 2.2.2 by means of numerical integration, linear-elastic analysis are performed from which the results have been compared to the commercial FE software ABAQUS. Various types of structures have been considered with different mesh refinements (i.e. number of elements and mesh size). The specifics and results of two particularly simple two-dimensional FEA are discussed here. Both structures are composed of only one element, which also simplified comparing the results. The first structure considered has one-dimensional homogeneous boundary conditions and is depicted including its specifics in Figure 3.2 while the results are listed in Table 3.1. The second structure has two-dimensional homogeneous boundary conditions as shown in Figure 3.3 along with its specifics and the results contained in Table 3.2. However, more sophisticated structures have been checked as well which are included in Appendix C. In addition, for quick visual validation (of larger structures) the author has written a post processing code in `MATLAB` that can plot the deformed structure by any scale factor and it can display any element variable of his choice in a contour plot. This has been showcased in Appendix C as well.

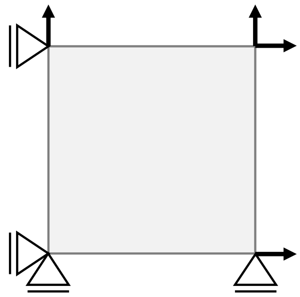


One-dimensional homogeneous		
Structure size	1×1	mm
Plane thickness	0.01	mm
Fixed DOFs	1,2,3	(left edge nodes)
Applied load	1E-05	N (DOFs 5 & 7)
Element type	CP4S	(Q4 plane stress element)
Mesh	1×1	(mesh size = 1×1 mm)
Young's modulus	1	MPa
Poisson's ratio	0.3	-

Figure 3.2: Structure with one-dimensional homogeneous boundary conditions

Table 3.1: Results one-dimensional homogeneous boundary conditions

`MATLAB`	**Results**	**Units**	**ABAQUS**	**Results**	**Units**
u_4	-0.0006	mm	N2 U22	-0.0006	mm
u_5	-0.002	mm	N3 U11	-0.002	mm
u_7	-0.002	mm	N4 U11	-0.002	mm
u_8	-0.0006	mm	N4 U22	-0.0006	mm
ε_{11}	-0.002	-	E11	-0.002	-
ε_{22}	0.0006	-	E22	0.0006	-
σ_{11}	-0.002	MPa	S11	-0.002	MPa



Two-dimensional homogeneous		
Structure size	1×1	mm
Plane thickness	0.01	mm
Fixed DOFs	1,3,4,8	(left & bottom edge nodes)
Applied load	1E-05	N (DOFs 5 & 7)
Element type	CP4S	(Q4 plane stress element)
Mesh	1×1	(mesh size = 1×1 mm)
Young's modulus	1	MPa
Poisson's ratio	0.3	-

Figure 3.3: Structure with two-dimensional homogeneous boundary conditions

Table 3.2: Results two-dimensional homogeneous boundary conditions

`MATLAB`	**Results**	**Units**	**ABAQUS**	**Results**	**Units**
u_2	0.0014	mm	N1 U22	0.0014	mm
u_5	0.0014	mm	N3 U11	0.0014	mm
u_6	0.0014	mm	N3 U22	0.0014	mm
u_7	0.0014	mm	N4 U11	0.0014	mm
ε_{11}	0.0014	-	E11	0.0014	-
ε_{22}	0.0014	-	E22	0.0014	-
σ_{11}	0.002	MPa	S11	0.002	MPa
σ_{22}	0.002	MPa	S22	0.002	MPa

From Tables 3.1 and 3.2 it can be seen that the developed `MATLAB` FEM code obtains similar results as ABAQUS. Please note that the omitted results are either zero or below machine precision. In both cases, displacements, strains and stresses are at least 17 orders of magnitude smaller than the other displacements, strains and stresses. This is below machine precision. Therefore, it can be concluded that the element formulations (by means of numerical integration) of the `MATLAB` FEM codes are correct. Additionally, three-dimensional element formulation validations can be found in Appendix C.

3.1.3 Implementation of Multi-Scale Material Model

As stressed in Section 2.2, the non-linear FEM code developed in this research utilizes the Newton-Raphson iterative method to trace the response of the structural models considered. Within this non-linear FEM algorithm, the macroscopic strain tensor is computed for all integration points in each element. Using the macroscopic strain tensor, the inter-granular displacements (Equation (2.3)), inter-granular stiffness tensor resulting from the inter-granular force-displacement relationships (Equation (2.6)) and macroscopic stiffness tensor (Equations (2.13) and (2.21)) are obtained successively. Therefore, the macroscopic stiffness tensor is a function of strain in GMA, as pointed out before.

Subsequently, the macroscopic stiffness tensors from all integration points are used for the calculation of each element's stiffness matrix $\mathbf{K}_e$ using numerical integration (see Equations (2.47) and (2.49)). Accordingly, the macroscopic stress tensor can be derived for all integration points from which the internal force vector $\mathbf{f}$ can be obtained for each element (Equations (2.51) and (2.52)). The aforementioned physical quantities as a function of the macroscopic strain tensor are required for computing the residual $\mathbf{R(u)}$ (Equation (2.34)) and displacement corrections (Equations (2.40) and (A.7)) thereafter for the Newton-Raphson iterative method. As discussed in Section 2.2, this process is repeated in an iterative method for each load increment until convergence (or an iteration limit) has been reached. The non-linear FEM is finished once the total load to be applied is reached and equilibrium is satisfied (i.e. the residual is below the specified tolerance for all increments).

The connectivity between the non-linear FEM and GMA `MATLAB` codes will be explained next. The GMA codes have been written as a `function` in `MATLAB`, such that it can be implemented as a material model module in any (non-linear) FEM code. These `function` codes essentially only require a material point's macroscopic strain tensor as an input. However, the plane stress implementation requires some additional information for obtaining its stiffness tensor using the Forward Euler technique as discussed in Section 2.1.2. After receiving the macroscopic strain tensor from the non-linear FEM `MATLAB` code, the GMA `function` codes start with computing the inter-granular displacements using the kinematic assumption (Equation (2.3)). Once this is done, the inter-granular forces are obtained from the microscopic constitutive laws (Equation (2.6)). Afterwards, the macroscopic stress and stiffness tensors are calculated using the PVW (Equations (2.13) and (2.21)). The aforementioned operations are all performed within the the GMA `function` codes. Once the GMA `function` codes are finished, the macroscopic stiffness tensor is returned to the non-linear FEM `MATLAB` code. During this research, three `MATLAB` implementations as GMA `function` codes have been developed: (i) `GMA_Plane_Stress.m` for two-dimensional FEA using plane stress conditions, (ii) `GMA_Plane_Strain.m` for two-dimensional FEA using plane strain conditions and (iii) `GMA_3D.m` for three-dimensional FEA. The three implementations work overall the same, but the main differences are the operations performed on the stiffness tensor for the two-dimensional implementations (see Section 2.1.2). The `MATLAB` codes can be found in Appendix G.3.1.

3.2 Non-Linear Finite Element Method & Topology Optimization

The next step in the current research was to implement the developed non-linear FEM codes with incorporation of the multi-scale material model into topology optimization algorithms. The added value of the GMA to topology optimization is that it incorporates a more realistic macroscopic material behaviour, since all grain-pair interactions are considered to contribute. All these particle contacts undergo a different loading history such that a material's micro-scale information (i.e. its microstructure and micromechanics) is captured in the macroscopic response. Similar to Section 3.1 in which the FEM portions of the `MATLAB` codes used in a previous master's study at TU/e [5, 23] served as starting points, the topology optimization codes used in that research are utilized here. In this case, it should be mentioned that these codes substantially inherit from the 88-line optimization code by Andreassen et al. [33] (combined with information from the publication by Luo & Kang [15]) and the `MATLAB` codes written by Biyikli & To [21, 22].

This section begins with the development and validation of an alternative penalization strategy that is compatible with the multi-scale material model. In what follows, several remarks concerning the minimum compliance problem and non-linear analysis are made, followed by a brief comment regarding different stress formulations and the non-linear FEM. Subsequently, the considered microscopic constitutive relationships are presented that are used in topology optimization. In conclusion, the implementation of the developed non-linear FEM algorithm into the PTO method is discussed.

3.2.1 Micro-Scale Penalization Technique

Previously, Section 2.3 addressed that the present research uses the *density-based approach* to topology optimization in which densities are assigned to each element that control the distribution of material within the design domain. Furthermore, it was stressed that intermediate densities (grey elements) are penalized by a penalization power to determine each element's mechanical properties, i.e. the stiffness tensor, using a power-law interpolation function. It is this "power-law approach" that relates the FEM with the topology optimization algorithm. Whereas most applications of topology optimization are focused on isotropic materials and therefore use a classical continuum mechanics approach with a given elastic stiffness tensor, in GMA the constitutive matrix or stiffness tensor generally becomes anisotropic and is a function of strain, hence its final form is unknown beforehand. Therefore, in contrast to isotropic materials, the stiffness tensor is not a function of Young's modulus E and Poisson's ratio ν (or alternatively Lame's constants). As a result, penalization cannot be performed by means of the SIMP method or modified SIMP approach that were discussed in Section 2.3. To this end, an alternative penalization strategy is developed here which is inspired by the SIMP method and is formulated in a similar manner.

Characteristics of the SIMP method are that (i) it does only affect intermediate densities while leaving solid and void elements unaffected and (ii) it mechanically activates elements in correspondence to their density values through affecting their stiffness tensors such that elements close to unit density contribute to the global stiffness matrix while elements being void are practically unresponsive. Therefore, the SIMP method with its penalization power may *implicitly* penalize elements having intermediate densities through which it steers the optimization towards black-and-white solutions. It was found that the alternative penalization strategy should exhibit these same characteristics.

Equations (2.57) and (2.58) show that both SIMP formulations can be regarded as penalization at macro-scale, since these affect a material's Young's modulus. However, in the presented multi-scale material model each material point's macroscopic stiffness tensor is derived from its average inter-granular stiffness coefficients (see Equation (2.20)). In that respect, the penalization should be brought down from macro-scale to micro-scale. While it is possible to penalize all inter-granular stiffness coefficients within each material point, a more computationally efficient strategy has been chosen here: instead of penalizing all inter-granular stiffness coefficients separately for each individual contact, the number density of contacts (N_p) is penalized. As a result, penalization is only required once per material point since N_p directly influences the macroscopic stiffness tensor (see Equation (2.21)). Equivalent to the SIMP method, the developed micro-scale penalization technique is formulated as a heuristic relation between an element's density x_e and its material point's number density of contacts N_p and is described as:

$$N_p = N_p(x_e) = x_e^p N_p^0 \quad \text{with} \quad 0 < x_{min} \leq x_e \leq 1\,, \tag{3.1}$$

where x_e is the density of element e, p is the penalization power (usually $p \geq 3$), N_p^0 the number density of contacts for solid material and x_{min} a small, non-zero value to prevent singularity of the stiffness matrix (typically 10^{-3}). Through the relation given by Equation (3.1), the micro-scale penalization technique only affects elements having intermediate densities by decreasing the number density of contacts. Consequently, a lower value for N_p explicitly takes a loss of particle contacts within a material point into account. Less inter-granular contacts eventually result in loss of stiffness. Therefore, the micro-scale penalization technique exhibits the same characteristics as the SIMP method.

Nevertheless, to be sure that the micro-scale penalization technique does indeed work similar as the SIMP method, linear-elastic analysis are performed for validation purposes. In this case, the results from the SIMP method have been compared to those from the micro-scale penalization technique for both the two- and three-dimensional FEM implementations. Two simple, single element structures have been considered from which the boundary conditions are not important here: only the effect of the element's densities on the macroscopic stiffness tensor are of importance. This effect has been checked as follows: the initial stiffness tensor for unit density ($x_e = 1$) is compared to the stiffness tensor for $x_e = 0.5$ while all other conditions remain unchanged (i.e. the boundary conditions and material properties). The first cases for $x_e = 1$ are not affected and serve as references. To be able to compare the linear-elastic FEM results using the SIMP method with the FEM results from the micro-scale penalization technique, a linear-elastic GMA has been considered. To this end, the material is assumed to be isotropic such that its Young's modulus and Poisson's ratio could be calibrated using the closed-form solutions in Equations (2.24) and (2.25). The material properties are chosen to represent those of concrete, with a Young's modulus of 30 GPa and Poisson's ratio of 0.2. The resulting macroscopic stiffness tensors are given in Equation (3.2) and (3.3) for the two-dimensional FEA, and Equations (3.4) and (3.5) for the three-dimensional FEA, respectively. For both penalization techniques, the penalization power is set equal to 3:

$$C_{ij}\left(x_e = 1\right) = \begin{bmatrix} 31250 & 6250 & 0 \\ 6250 & 31250 & 0 \\ 0 & 0 & 12500 \end{bmatrix}, \tag{3.2}$$

$$C_{ij}\left(x_e = 0.5\right) = \begin{bmatrix} 3906.25 & 781.25 & 0 \\ 781.25 & 3906.25 & 0 \\ 0 & 0 & 1562.5 \end{bmatrix}. \tag{3.3}$$

The two-dimensional results in Equations (3.2) and (3.3), and the three-dimensional results in Equations (3.4) and (3.5) yield the same prediction of properties for both the developed micro-scale penalization technique and the macro-scale SIMP method. In both cases, the power-law interpolation functions do apply to all macroscopic stiffness tensor entries, i.e. $C_{ij} = x_e^p C_{ij}^0$ where C_{ij}^0 is the macroscopic stiffness tensor for solid material. Please note that the micro-scale results are obtained numerically whereas the macro-scale SIMP method is computed analytically. Therefore, (some) entries that are zero in the SIMP method may have small, non-zero values in the microscopic penalization. However, these entries are negligibly small because they are below machine precision: similar as before, in both cases the small, non-zero numerical results are at least 17 orders of magnitude smaller than the other macroscopic stiffness tensor entries. In conclusion, the micro-scale penalization technique developed here has a similar effect on the macroscopic stiffness tensor as the SIMP method while being compatible with the multi-scale material model.

$$C_{ij}\left(x_e = 1\right) = 10^3 \begin{bmatrix} 33.3 & 8.33 & 8.33 & 0 & 0 & 0 \\ 8.33 & 33.3 & 8.33 & 0 & 0 & 0 \\ 8.33 & 8.33 & 33.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12.5 \end{bmatrix}, \tag{3.4}$$

$$C_{ij}\left(x_e = 0.5\right) = 10^3 \begin{bmatrix} 4.17 & 1.04 & 1.04 & 0 & 0 & 0 \\ 1.04 & 4.17 & 1.04 & 0 & 0 & 0 \\ 1.04 & 1.04 & 4.17 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.56 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.56 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.56 \end{bmatrix}. \tag{3.5}$$

3.2.2 Minimum Compliance Problem with Non-Linear Analysis

When considering the minimum compliance problem, which aims to minimize structural compliance for a given structural volume, one could argue that this means designing a structure for serviceability purposes since it is intended to limit the deformations. In this case, it is usually expected that materials remain in their linear-elastic branch rather than having non-linear or even plastic behaviour.

The focus of the present research is on topology optimization of cementitious materials (such as 3DCP) to incorporate their specific properties and evolution of microstructure during construction and loading. To this end, the main objective is to see the effects of non-linear and isotropic behaviour of these cement-based materials on their optimized topologies. Even though a material may be initially isotropic when unloaded, since within GMA the inter-granular interactions are experiencing a different loading history and are generally non-linear, contacts that are oriented in all different directions will have their inter-granular stiffness evolve depending on their orientation and loading path. Therefore, a material will exhibit anisotropic behaviour when loaded, which is called *induced* anisotropy [14]. As a result, the presence of a well-defined linear-elastic branch is not guaranteed since the evolution of microstructure may generally cause non-linear behaviour in the macroscopic response of the material from the beginning of loading. For this reason, topology optimization for the minimum compliance problem using non-linear analysis is required to address the non-linearities that arise in the multi-scale analysis. This is also seen in other topology optimization works that consider compliance minimization for non-linear multi-scale analysis [39, 40].

Moreover, it is feasible within AM to use multiple materials, which could result in multi-functional structural designs. To this end, the incorporation of material non-linearities could be advantageous in topology optimization design applications that include multiple general elastic and inelastic materials, as discussed in [38]. This can be a potential strength in solving topology optimization problems that involve more than two materials [59] and maybe even multiple objectives. For example, when optimizing a multi-functional structural design for both compliance and thermal performance, one might perform topology optimization to find an optimal distribution of load bearing structural material (e.g. behaving non-linearly), thermal insulation (e.g. behaving linearly) and voids [25]. Besides, composite materials can have elastic and inelastic characteristics on the micro-structural level.

Furthermore, more realistic designs are pursued when considering material non-linearities [39]. However, the use of a sophisticated multi-scale material model with incorporation of material non-linearities comes with an increase in computational demand. A large number of non-linear load increments needs to be calculated, which causes additional computation time. This means that in non-linear topology optimization it is required to solve the multi-scale problem (i.e. running the GMA) not once, but for numerous iterations of the structural topology. As a result, the topology optimization simulations performed in the current research are sound since all iterations of the structural topology are a result of converged non-linear FEM solutions with the inclusion of the non-linear and anisotropic material behaviour of cementitious materials.

Additionally, the added value of including the non-linear and anisotropic behaviour of cementitious materials for the minimum compliance problem, besides finding the optimized topology, is to know until what load level a certain optimized topology works. However, contrary to linear-elastic analysis, in this case the optimized material distribution is dependent on the magnitude of the applied load [60]. In the work presented here, the load level is determined by trial and error. Alternatively, the objective function could be altered to maximize the load factor of the system in equilibrium, which is presented in [61]. However, this is considered to be beyond the scope of the present research.

In conclusion, it is observed in literature that the background of deciding upon which topology optimization scheme to use is rarely explicitly explained. Most often there is little to no reflection on this choice. In practice, the main focus is usually to prevent failure in structural designs, which is related to the stress-constrained problem, while checking the deformations in the serviceability state is considered later, which is related to the minimum compliance problem. However, as mentioned in Section 2.3.1 a vast majority of earlier topology optimization researches address the compliance minimization design problem [15, 20, 37]. In "Topology Optimization: Theory, Methods, and Applications", Bendsøe & Sigmund [60] argue that the first problems in the field of topology optimization are considering the minimum compliance problem due to its simplicity in terms of objective and constraints. To this end, it has been used as a fundamental test case in the early days of topology optimization methods. Surprisingly, usually no guidance is provided in literature about why specific topology optimization schemes are used nor is any reasoning given with respect to the decision of their objectives and constraints. This could be related to the application of topology optimization in new areas where it is important to make sensible decisions and combinations of objective functions and constraints, which is one of the most challenging and difficult parts according to Bendsøe & Sigmund [60]. These should be physically meaningful and manageable for simulation and topology optimization algorithms.

3.2.3 Local Stress Interpolation with Non-Linear Analysis

As discussed in Section 2.3.2, a suitable stress failure state within a fictitious porous microstructure of an intermediate-density element is defined for the SIMP methods through Equation (2.66). However, in the SIMP approach this local stress interpolation usually yields that the density-related term in the *local* stress formulation vanishes within the linear-elastic FEM (see Equation (2.65)). Section 2.2 stressed that in the non-linear FEM the displacement corrections are computed using the residual, which is defined as the difference between the internal and external force vectors. Thereby, in Section 2.2.2 Equation (2.50) showed that the internal force vector depends on stress. Therefore, it should be clarified what stress formulation is used for which implementation in this research.

The non-linear FEM with the incorporated multi-scale material model utilizes the macroscopic stress related to the deformations (i.e. strains) for computing the internal force vector, that is the stress measure including density-related penalization effects ($\bar{\boldsymbol{\sigma}}(x_e)$ in Equation (2.66)). As a result, the residual and thus the displacement corrections are calculated based on the penalized stress measure as well. However, the topology optimization algorithms concerned with the stress-constrained problem use the stress failure state defined through local stress interpolation, thus the stress measure divided by x_e^q with $q \geq 1$. In this case, where material non-linearities are considered, it was found that using $q = 1$ worked most optimal in combination with the non-linear FEM. Therefore, it should be clear that the local stress interpolation is performed *outside* of the non-linear FEM scheme.

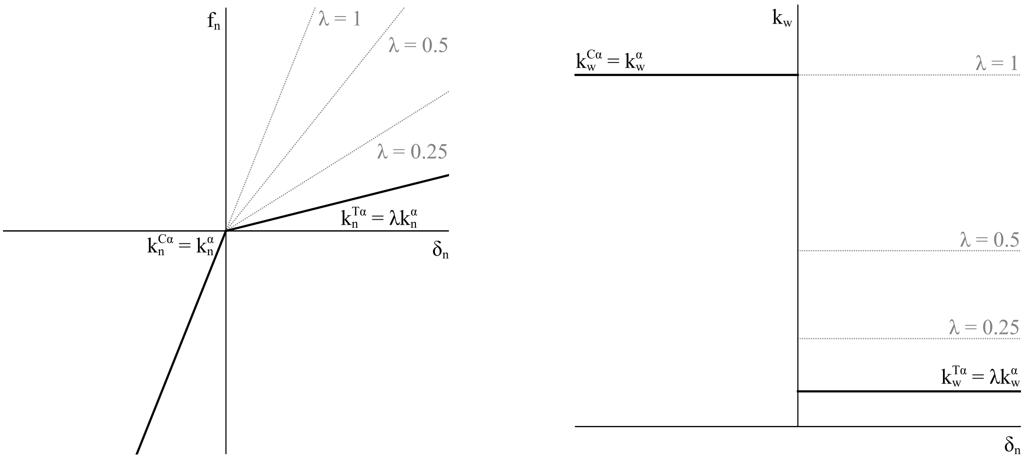
3.2.4 Microscopic Constitutive Relationships & Benchmarks

The considered microscopic constitutive relationships or inter-granular force-laws in the current research will be presented here. These inter-granular force-displacement relationships relate the decomposed relative displacement components and their conjugate inter-granular force components at each grain-pair interaction, as explained in Section 2.1. To this end, normal and tangential stiffness coefficients are used for defining the normal and tangential inter-granular force components. The present study considers two different inter-granular force-laws that are used in topology optimization.

The first microscopic constitutive relationship has been presented in a publication by H. Jia et al. [9] on "Optimal structural topology of material with micro-scale tension-compression asymmetry simulated using granular micromechanics". In this work, the macro-scale material behaviour is affected in a complex manner as a result of tension-compression asymmetry at micro- or grain-scale. Similar to the present research, the constitutive equations are derived using the GMA as discussed in Section 2.1. The topology optimization problem in [9] is considered to be the minimization of structural strain energy in order to obtain the stiffest structure possible for a given volume fraction. To this end, a bi-linear inter-granular force-displacement relationship having unequal microscopic stiffness coefficients in tension and compression is assumed. This assumption mimics a more realistic non-linear behaviour without the need for an iterative procedure. Therefore, the effect of tension-compression asymmetry at macro-scale is modelled by specifying the inter-particle stiffness as follows:

$$k_i^\alpha = \begin{cases} k_\mathrm{n}^{T\alpha} = \lambda k_\mathrm{n}^\alpha; \quad k_\mathrm{w}^{T\alpha} = \lambda k_\mathrm{w}^\alpha; & \text{if} \quad \delta_\mathrm{n}^\alpha > 0 \quad (\text{tension}), \\ k_\mathrm{n}^{C\alpha} = k_\mathrm{n}^\alpha; \quad k_\mathrm{w}^{C\alpha} = k_\mathrm{w}^\alpha; & \text{if} \quad \delta_\mathrm{n}^\alpha \leq 0 \quad (\text{compression}), \end{cases} \tag{3.6}$$

where the superscripts T and C refer to tension and compression, respectively, and λ is a constant between 0 and 1 that influences the material's tensile behaviour. If λ is set to 1, the material exhibits symmetric behaviour in tension and compression. When λ is reduced to zero, the material has no tensile stiffness (nor strength). Notice that both the normal and tangential inter-granular stiffness coefficients are functions of the normal component of the relative displacement δ_n^α. As a result, the material will exhibit *induced anisotropy* from the initiation of the loading process [9]. The inter-granular force-law is shown in Figure 3.4. Please note that the λ terms in Figures 3.4(a) and 3.4(b) are identical and refer to Equation (3.6).

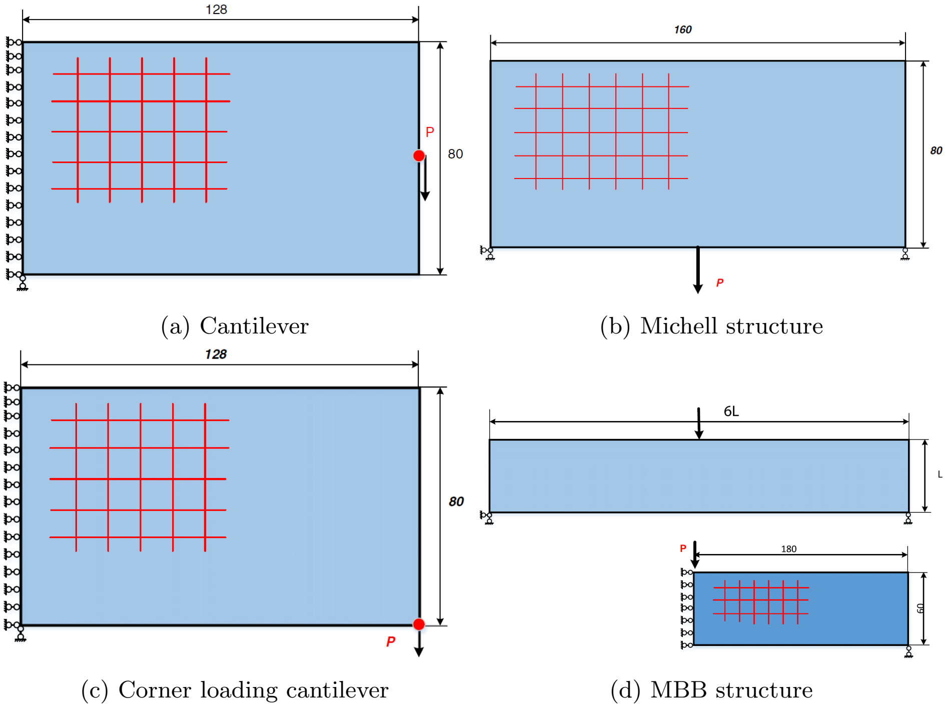


(a) Normal directions

(b) Tangential directions

Figure 3.4: Tension-compression asymmetric inter-granular force-law

The publication by H. Jia et al. [9] considers four benchmark examples to demonstrate the effectiveness of the mathematical formulation of topology optimization for the granular material structure, these can be found in Figure 3.5 including their corresponding number of elements in x and y directions. Furthermore, this work studies the effect of λ on the final topology to demonstrate the effect of different micro-scale tension-compression asymmetries on the optimized topology. In the present work, the same four benchmarks as well as numerical input as in [9] have been used in the topology optimization for the multi-scale material model to be able to compare results. Therefore, the material is assumed to be isotropic in the unloaded condition such that $\xi(\theta, \phi) = 1/4\pi$ having an unloaded Young's modulus of 52 GPa and a Poisson's ratio of 0.11 with the stiffness parameters $k_\mathrm{n} = 2$ kN/m and $k_\mathrm{w} = 1$ kN/m, while the grain size $l = 10$ μm and $N_p = 10^{18}$ m^{-3}. However, it should be noted that the results in [9] have been obtained through Evolutionary Structure Optimization (ESO) method which is yet another topology optimization method as the ones considered in the current research (see Section 2.3.2). For more information on the ESO method, the reader is referred to [5, 9, 23].



(a) Cantilever

(b) Michell structure

(c) Corner loading cantilever

(d) MBB structure

Figure 3.5: Geometry and boundary conditions of four benchmark examples [9]

The second microscopic constitutive relationship is more focused on strength rather than stiffness compared to the first one. Whereas the first inter-granular force-law is used for the stiffness-related compliance minimization problem in topology optimization, the second one is used in the strength-related stress-constrained problem. The micro-scale tension-compression asymmetric constitutive law is not suited for strength-related problems, because it does not exhibit a loss in stiffness and therefore imposes infinite strength. Therefore, the second inter-granular force-law considers a bi-linear microscopic force-displacement relationship with hardening which is defined as:

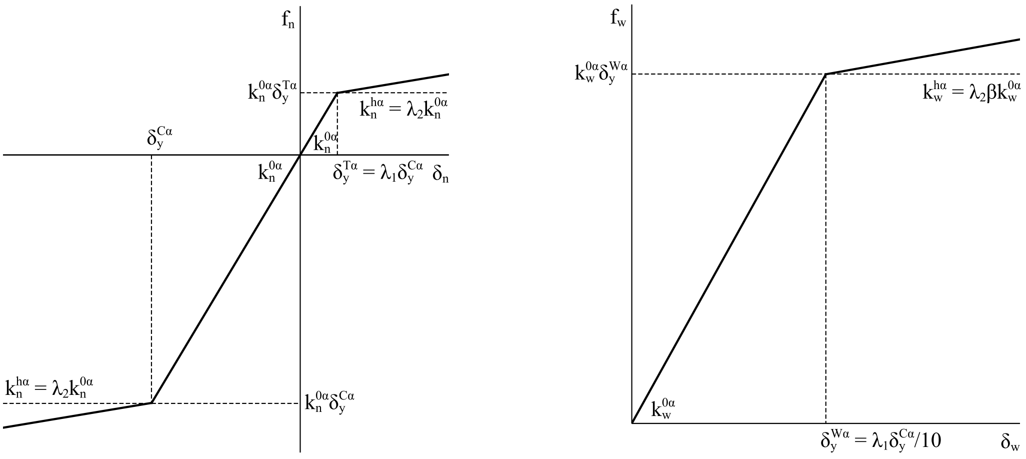
$$
\begin{aligned}
f_{\mathrm{n}}^{\alpha} &= \begin{cases} k_{\mathrm{n}}^{0\alpha}\delta_{\mathrm{n}}^{\alpha} & \text{if } \delta_y^{C\alpha} \leq \delta_{\mathrm{n}}^{\alpha} \leq \delta_y^{T\alpha} \quad \text{(prior to hardening)}\,, \\ k_{\mathrm{n}}^{0\alpha}\delta_y^{T\alpha} + k_{\mathrm{n}}^{h\alpha}\left(\delta_{\mathrm{n}}^{\alpha} - \delta_y^{T\alpha}\right) & \text{if } \delta_{\mathrm{n}}^{\alpha} > \delta_y^{T\alpha} \quad \text{(tensile hardening)}\,, \\ k_{\mathrm{n}}^{0\alpha}\delta_y^{C\alpha} + k_{\mathrm{n}}^{h\alpha}\left(\delta_{\mathrm{n}}^{\alpha} - \delta_y^{C\alpha}\right) & \text{if } \delta_{\mathrm{n}}^{\alpha} < \delta_y^{C\alpha} \quad \text{(compressive hardening)}\,, \end{cases} \\
f_{\mathrm{w}}^{\alpha} &= \begin{cases} k_{\mathrm{w}}^{0\alpha}\delta_{\mathrm{w}}^{\alpha} & \text{if } \delta_{\mathrm{w}}^{\alpha} \leq \delta_y^{W\alpha} \quad \text{(prior to hardening)}\,, \\ k_{\mathrm{w}}^{0\alpha}\delta_y^{W\alpha} + k_{\mathrm{w}}^{h\alpha}\left(\delta_{\mathrm{w}}^{\alpha} - \delta_y^{W\alpha}\right) & \text{if } \delta_{\mathrm{w}}^{\alpha} > \delta_y^{W\alpha} \quad \text{(shear hardening)}\,, \end{cases}
\end{aligned}
\tag{3.7}
$$

where the microscopic stiffness coefficients $k_i^{0\alpha}$ and $k_i^{h\alpha}$ provide the initial inter-granular stiffness and micro-scale stiffness after hardening, respectively, and the constants $\delta_y^{i\alpha}$ are related to the different relative displacements at yielding. The superscript W refers to shear. The aforementioned parameters are given by:

$$\delta_y^{i\alpha} = \lambda_1 \delta_y^{C\alpha} = \begin{cases} \delta_y^{T\alpha} = \lambda_1 \delta_y^{C\alpha}\,, \\ \delta_y^{W\alpha} = \frac{\lambda_1}{10} \delta_y^{C\alpha}\,, \end{cases} \qquad k_i^{h\alpha} = \lambda_2 k_i^{0\alpha} = \begin{cases} k_\mathrm{n}^{h\alpha} = \lambda_2 k_\mathrm{n}^{0\alpha}\,, \\ k_\mathrm{w}^{h\alpha} = \lambda_2 \beta k_\mathrm{w}^{0\alpha}\,, \end{cases} \tag{3.8}$$

where the constants λ_1 and λ_2 can be regarded as yield point ($\delta_y^{i\alpha}/\delta_y^{C\alpha}$) and hardening ($k_i^{h\alpha}/k_i^{0\alpha}$) ratios, respectively, both having values between 0 and 1 (similar to λ in Equation (3.6)). β describes the inter-granular shear to normal stiffness ratio as defined in Equation (2.25). If λ_1 is set to 1 the material exhibits symmetric behaviour in tension and compression and when λ_1 is zero tensile hardening occurs immediately. Making λ_2 equal to 1 results in linear-elastic material behaviour without the occurrence of hardening and if λ_2 equals zero ideal plastic behaviour will be obtained. Even though it is possible to make λ_2 a negative value to obtain softening, in the present research its value is limited to $0 \leq \lambda_2 \leq 1$. The resulting inter-granular force-displacement relationships typically exhibit hardening as shown in Figure 3.6. Please note once more that λ_1 and λ_2 are also identical in Figures 3.6(a) and 3.6(b) and refer to Equation (3.8). Parameters $\delta_y^{i\alpha}$ ($\delta_y^{T\alpha}$, $\delta_y^{C\alpha}$ and $\delta_y^{W\alpha}$) correspond to the inter-granular displacements at yielding of the inter-granular force-displacement curves in tension, compression and shear. The inter-granular stiffness can be derived through direct differentiation of Equation (3.7):

$$\begin{aligned} k_\mathrm{n}^{\alpha} &= \begin{cases} k_\mathrm{n}^{0\alpha} & \text{if} \quad \delta_y^{C\alpha} \leq \delta_\mathrm{n}^{\alpha} \leq \delta_y^{T\alpha} & \text{(prior to hardening)}\,, \\ k_\mathrm{n}^{h\alpha} & \text{if} \quad \delta_\mathrm{n}^{\alpha} < \delta_y^{C\alpha} \vee \delta_\mathrm{n}^{\alpha} > \delta_y^{T\alpha} & \text{(post hardening)}\,, \end{cases} \\ k_\mathrm{w}^{\alpha} &= \begin{cases} k_\mathrm{w}^{0\alpha} & \text{if} \quad \delta_\mathrm{w}^{\alpha} \leq \delta_y^{W\alpha} & \text{(prior to hardening)}\,, \\ k_\mathrm{w}^{h\alpha} & \text{if} \quad \delta_\mathrm{w}^{\alpha} > \delta_y^{W\alpha} & \text{(shear hardening)}\,. \end{cases} \end{aligned} \tag{3.9}$$


f_n
$k_n^{0\alpha}\delta_y^{T\alpha}$
$k_n^{h\alpha} = \lambda_2 k_n^{0\alpha}$
$\delta_y^{C\alpha}$
$k_n^{0\alpha}$
$k_n^{0\alpha}$
$\delta_y^{T\alpha} = \lambda_1 \delta_y^{C\alpha}$
δ_n
$k_n^{h\alpha} = \lambda_2 k_n^{0\alpha}$
$k_n^{0\alpha}\delta_y^{C\alpha}$
f_w
$k_w^{0\alpha}\delta_y^{W\alpha}$
$k_w^{h\alpha} = \lambda_2 \beta k_w^{0\alpha}$
$k_w^{0\alpha}$
$\delta_y^{W\alpha} = \lambda_1 \delta_y^{C\alpha}/10$
δ_w

(a) Normal directions

(b) Tangential directions

Figure 3.6: Bi-linear inter-granular force-law with hardening

The multi-scale material model used in the present research takes the response of contacts in all different directions into account for deriving the macroscopic material behaviour. All particle contacts undergo a different loading history, i.e. some are in tension while others are in compression and some might still be in the initial stiffness branch while others are already hardening. Therefore, similar to the particle-binder composite described by Poorsolhjouy & Gonzalez [2], it should be noted that even though the inter-granular force-law presented in Figure 3.6 is piecewise continuous, the macroscopic material behaviour is smooth.

3.2.5 Implementation of Non-Linear Finite Element Method

Section 2.3 states that topology optimization is used to find the most optimal material distribution of a structure given its design domain and boundary conditions. To this end, the current research utilizes the *density-based approach* through the PTO method. Within this method, the design variables (element densities) are updated via different computational approaches (see Figure 2.11). Nevertheless, the developed micro-scale penalization technique (Equation (3.1)) is used to penalize a material point's macroscopic stiffness tensor through penalizing the number density of contacts (N_p). Accordingly, the penalized macroscopic stiffness tensors are used in the developed non-linear FEM algorithms as explained before. These return either the considered objective function value, i.e. the overall compliance for the minimum compliance problem, or the constraint function values, i.e. the stress constraint function values for the stress-constrained problem in order to update the design variables.

Now the connectivity between the topology optimization and non-linear FEM `MATLAB` codes will be explained. Identical to the GMA codes, the non-linear FEM codes have been written as a `function` in `MATLAB` to be implemented as a non-linear FEM module in any topology optimization code. In this case, these `function` codes essentially only require the density distribution as an input. However, the topology optimization `MATLAB` codes have been written as the main codes such that the boundary conditions should be passed as well. After receiving the density distribution from the topology optimization `MATLAB` code, the non-linear FEM `function` codes initiate by penalizing the macroscopic stiffness tensors through penalizing the number density of contacts (grain-pair interactions) in each element based on its related density value (Equation (3.1)). Once this is done, the Newton-Raphson iterative method starts as discussed in Section 2.2. During this iterative procedure, the required objective or constraint function values for updating the design variables are obtained, i.e. either the overall compliance or the elemental stress values. The aforementioned operations are all performed within the non-linear FEM `function` codes. When the non-linear FEM `function` codes are done (i.e. convergence has been reached), the objective and constraint function values are returned to the topology optimization `MATLAB` code. Over the course of this research, two `MATLAB` implementations as non-linear FEM `function` codes have been developed for both three-dimensional as well as two-dimensional topology optimization. These implementations are available in Appendix G.2.1.

In conclusion, it should be made clear that the non-linear FEM codes converge for all load increments before returning the converged solution of the final load increment to the topology optimization algorithm. To be more specific: the non-linear FEM utilizes the same density distribution throughout the entire Newton-Raphson iterative method. Only after the non-linear FEM codes have fully converged, the design variables are updated via the topology optimization algorithm. As a result, within each iteration of the topology optimization algorithm the non-linear FEM is ran once.

3.3 Proportional Topology Optimization Using Multi-Scale Model

The first implementations of the developed non-linear FEM using the multi-scale material model into topology optimization algorithms have been performed through the PTO method, for reasons explained in Section 2.3.2. First of all, PTO is a non-sensitivity method in contrast to the TTO method. Therefore, it does not require the derivative of the objective function with respect to the design variables which cannot be obtained analytically for the considered multi-scale model. It was also found that PTO is up to 264 times faster compared to TTO. As a result, PTO was considered to be a good method to initially implement the non-linear FEM using the GMA into. All optimizations used and referred to throughout this work are performed using load controlled FEA. Two-dimensional FEA is always performed under plane stress conditions.

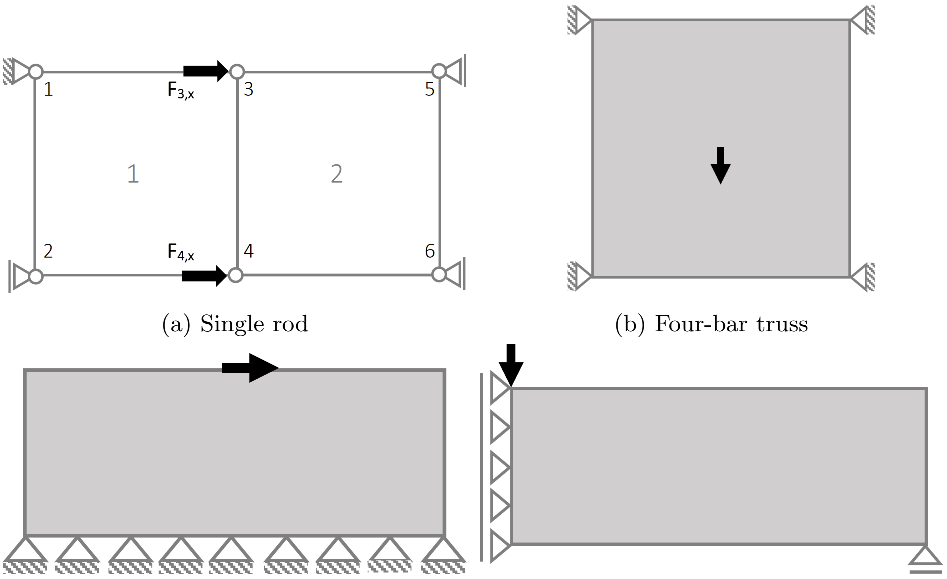
This section reviews the two discussed topology optimization problems, starting with the minimum compliance problem. Afterwards, the stress-constrained problem is addressed.

3.3.1 Proportional Topology Optimization Using Compliance

The implementations of the developed non-linear FEM using the GMA into the PTO method started with the compliance minimization problem (see Appendix G.1). As explained in Section 2.3.1, the aim of the minimum compliance problem is to minimize structural compliance for a given volume fraction through which it is intended to enhance the overall stiffness of a structure. The problem for non-linear material behaviour in this research is presented in Equation (2.59). According to Section 3.2.1, intermediate densities (grey elements) are penalized by the developed micro-scale penalization technique (Equation (3.1)). The microscopic constitutive relationship used for the compliance minimization problem is the one presented by H. Jia et al. [9], in which the macro-scale material behaviour is affected via tension-compression asymmetry at grain-scale as described in Section 3.2.4 (see Equation (3.6)). The publication also considers the minimization of structural strain energy in order to obtain the stiffest structure possible for a given volume fraction provided a different topology optimization method (ESO) is used therein. However, it should be noted that the considered tension-compression asymmetric inter-granular force-law does not exploit the incremental-iterative procedure of the developed non-linear FEM algorithms.

For the compliance minimization problem using the PTO method (PTOc), initially the same four benchmark examples as in [9] are considered (see Figure 3.5) to demonstrate the effect of the multi-scale material model on topology optimization. Similarly, the effect of λ in Equation (3.6) is studied as well to demonstrate the effect of different micro-scale tension-compression asymmetries on the optimized topology. To be able to compare results, the same numerical input has been used for the GMA. Therefore, the material is assumed to be isotropic in the unloaded condition such that $\xi(\theta, \phi) = 1/4\pi$ having an unloaded Young's modulus of 52 GPa and a Poisson's ratio of 0.11 with the stiffness parameters $k_\mathrm{n} = 2$ kN/m and $k_\mathrm{w} = 1$ kN/m, while the grain size $l = 10$ μm and $N_p = 10^{18}$ m^{-3}. All structures are subjected to a concentrated point load of 1 N with the volume constraint set at 60% of the total design domain. The filter radius is set to 1.5 and kept constant throughout this work, unless stated otherwise. It should be noted that both the normal and tangential inter-granular stiffness coefficients are functions of the normal component of the relative displacement δ_n^α in Equation (3.6) for which the convergence criterion needed to be relaxed to 10^{-2} for the tension-compression asymmetry inter-granular force-law in order to achieve convergence in the non-linear FEM. In the end, the obtained results have been validated by means of a simple comparison study with the commercial software ABAQUS.

However, the previously considered four benchmark examples (Figure 3.5) are each composed of a rather large number of elements, resulting in large computational demand of the non-linear FEM for non-linear microscopic constitutive relationships that do require utilization of the Newton-Raphson iterative method. Therefore, except for PTOc using the tension-compression asymmetric inter-granular force-law, the current research only considers four smaller benchmark examples (see Figure 3.7) as presented in a previous research conducted at the TU/e [5, 23]. Please note that the MBB structure was already included in Figure 3.5(d), but it is repeated here because a smaller number of elements will be considered now. To be more specific, the design domains of the benchmark examples included in Figure 3.7 are discretized into the following number of elements in x and y directions, respectively, in accordance with the previous research: the single rod benchmark (Figure 3.7(a)) has 1×2 elements, the four-bar truss benchmark (Figure 3.7(b)) has 20×20 elements, the two-bar truss benchmark (Figure 3.7(c)) has 20×80 elements and the (half) MBB beam benchmark now has 60×20 elements. Because these four benchmark examples have been reported for linear-elastic FEA with the stress-constrained problem using the TTO method (TTOs) in the latter reference, it is still possible to compare the results obtained here. Besides, the four benchmark examples allow for quick visual inspection of the occurrence of either tensile or compressive elements. In this case, the volume fraction (i.e. the constraint) is set to 0.25, meaning that only 25% of the design domain may be occupied by material. However, the single rod benchmark uses a volume fraction of 0.5 and has a different filter radius than the standard value of 1.5: this specific benchmark example in Figure 3.7(a) ignores filtering by setting r_0 equal to 1.



(a) Single rod

(b) Four-bar truss

(c) Two-bar truss

(d) MBB beam

Figure 3.7: Geometry and boundary conditions of four benchmark examples [5]

To illustrate the abilities of the developed non-linear FEM using the GMA to be expanded and utilized for three-dimensional topology optimization applications, a three-dimensional representation of the single rod benchmark is considered as well (see Figure 3.8). In this case, the design domain is discretized into 1 × 2 × 1 elements in x, y and z directions, respectively. Similarly, the benchmark example allows for quick visual inspection of the occurrence of either tensile or compressive elements. The same volume fraction of 0.5 is used and filtering is ignored by using a filter radius of 1.

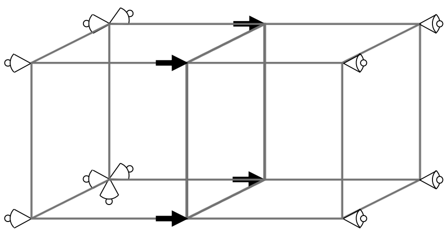


Figure 3.8: Three-dimensional representation of single rod benchmark

3.3.2 Proportional Topology Optimization Using Stress

Even though it was found in a previous research [5, 23] that the TTO method obtains the most optimal results for materials with unequal strength limits for stress-constrained volume minimization, the PTO method has also been applied to the stress-constrained problem because of its simplicity and no use of sensitivity information (see Appendix G.1). The aim of the stress-constrained problem is to minimize the structural volume while satisfying a certain stress constraint, as stressed in Section 2.3.1. From the viewpoint of classical failure criteria, the problem is generally described by Equation (2.60). Identical to the minimum compliance problem using PTO, the micro-scale penalization technique developed in Section 3.2.1 is used according to Equation (3.1). The bi-linear inter-granular force-law with hardening, as described by Equations (3.7) to (3.9), is the microscopic constitutive relationship used for the stress-constrained problem. This non-linear microscopic constitutive relationship does require utilization of the Newton-Raphson iterative method of the developed non-linear FEM algorithms.

As discussed in Section 2.3.2, ”local stress interpolation” is used to define a suitable stress failure state within a fictitious porous microstructure of elements having intermediate densities. For isotropic materials, using the SIMP method, it is argued that the power q in Equation (2.66) should be chosen to be equal to the penalization power p, which is usually equal to 3. However, for the non-linear FEM with incorporation of the multi-scale material model $q = 1$ was found to be a reasonable value. Furthermore, ε-relaxation is applied to the stress constraint function according to Equation (2.67) which is reformulated for the PTO implementation. Inspired by the implementation of [5, 23], the normalized stress constraint function is rewritten as follows:

$$R_e(\boldsymbol{\sigma}(x_e), x_e) = \left[\frac{\|\boldsymbol{\sigma}(x_e)\|}{\|\boldsymbol{\sigma}^{lim}\|} - \frac{\varepsilon\,(1 - x_e)}{x_e} \right]_+ \leq 1\,, \tag{3.10}$$

which implies that the stress constraint function here can be regarded as an indicator of how sensible or close the material is to failure. Please note that in contrast to [5, 23], not the homogenized macroscopic stress but microscopic *local* stress (Equation (2.66)) is used to determine the stress constraint function. Negative stress constraint function values can occur for significantly small densities through which the ε-related term in Equation (3.10) becomes smaller than -1. In order to prevent negative stress constraint function values causing problems in the proportional distribution algorithm, these are replaced by a small, non-zero value equal to x_{min}, i.e. $[.]_+ = \max(x_{min}, .)$.

The four benchmark examples in Figure 3.7 are considered to showcase the effect of the GMA. The numerical input used for the GMA has been chosen to mimic the material characteristics of concrete class C20/25 according to Table 3.1 of Eurocode 2 [62]. Therefore, the material is assumed to be isotropic in the unloaded condition such that such that $\xi(\theta, \phi) = 1/4\pi$ having an unloaded Young’s modulus of 30 GPa and a Poisson’s ratio of 0.2 with the stiffness parameters $k_\text{n} = 5$ MN/m and $k_\text{w} = 0.83$ MN/m, while the grain size $l = 0.1$ mm and $N_p = 3 \cdot 10^{12}$ m^{-3}. Identical to Section 3.3.1, the abilities of the developed non-linear FEM using the GMA to be expanded and utilized for three-dimensional topology optimization applications are illustrated by means of the three-dimensional representation of the single rod benchmark in Figure 3.8.

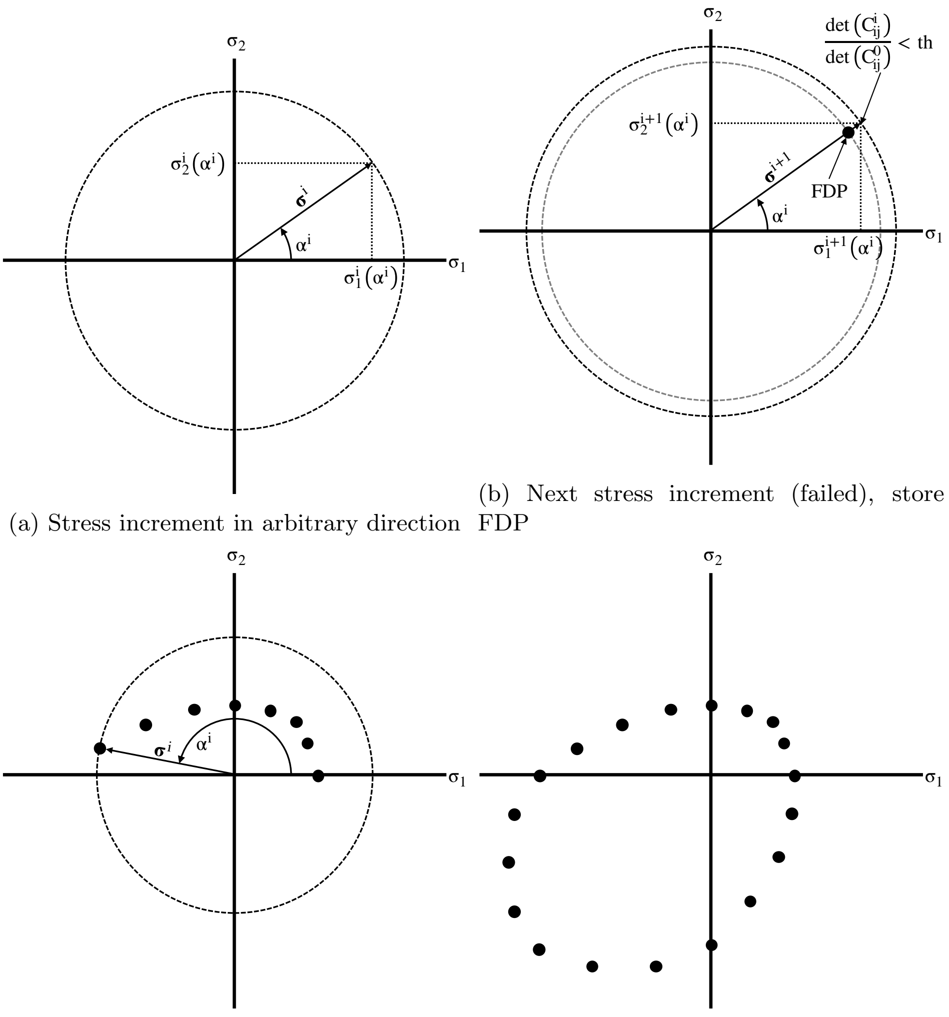
Besides the general added value of the GMA to topology optimization by providing a more realistic macroscopic material behaviour through incorporation of its microstructure and micromechanics, there is yet another advantage with respect to stress-constrained volume minimization. Instead of using predefined failure envelopes resulting from classical failure criteria (e.g. the Drucker-Prager yield criterion) that are chosen to match the macroscopic material behaviour from experimental data, the multi-scale model can predict failure points to construct failure envelopes that are an exact match for any given inter-granular force-law. This allows strength-related topology optimization (i.e. the stress-constrained problem) to fully exploit the multi-scale model, because the GMA is able to both (i) describe the macroscopic material behaviour up to failure in a non-linear framework and to (ii) predict failure points to obtain tailored failure envelopes for any given microscopic constitutive relationship as a result of natural failure.

However, in order to implement GMA-based failure envelopes into stress-constrained volume minimization, one should be able to determine their corresponding constraint function values (i.e. the failure criterion values). Since the multi-scale model can only predict Failure Data Points (FDPs), in contrast to classical failure criteria that are defined as functions of stress, predicting how close a material point is to failure requires two steps. The first step is to construct the yield surface or failure envelope from a given inter-granular force-law by means of a sufficient number of FDPs. Afterwards, the constraint function values, referred to as *failure sensitivities*, can be obtained through linear interpolation. Both steps are elaborated below.

- Construction of failure envelope from inter-granular force-law

  The GMA-based failure envelopes are constructed from a sufficient number of FDPs as a result of the natural failure for a given inter-granular force-law. To this end, the principal stress space is discretized into a sufficient number of directions for which the FDPs will be computed. For each direction, the corresponding principal stress state will be gradually increased via stress increments by means of a stress controlled analysis. The determinant of the macroscopic stiffness tensor, $\det(C^i_{ij})$, is calculated for each stress increment and compared to its unloaded state, $\det(C^0_{ij})$. This comparison is performed by computing the determinant ratio $\det(C^i_{ij})/\det(C^0_{ij})$ and checking if it is below a predefined threshold for all stress increments. This threshold can essentially be regarded as a convergence criterion, similar to the non-linear FEM as discussed in Section 2.2. However, the threshold for the determinant ratio determines the loss of initial stiffness that is considered as failure. Similar to [14], the threshold is usually set to 10^{-3}. This threshold means that a loss of $(1 - 0.001) \times 100\% = 99.9\%$ of the initial stiffness is regarded as failure of the material. However, some microscopic constitutive relationships may require the threshold value to be raised, such as the bi-linear inter-granular force-law. In this case, the determinant of the macroscopic stiffness tensor might not be able to sufficiently decrease for higher values of λ_2. For example, $\lambda_2 = 0.25$ requires a threshold that is about 50 times larger in order to obtain a reasonable failure envelope, meaning that only a loss of $(1 - 0.05) \times 100\% = 95\%$ of the initial stiffness can be reached. If the determinant ratio is above the threshold, meaning no failure, the principal strains are updated using the forward Euler technique as $\mathrm{d}\varepsilon_{ij} = C_{ij}^{-1}\mathrm{d}\sigma_{ij} \rightarrow \varepsilon_{ij}^{i} = \varepsilon_{ij}^{i-1} + \mathrm{d}\varepsilon_{ij}^{i}$. When the determinant ratio is below the threshold, meaning failure has occurred, the previous (unfailed) principal stress state is stored as FDP. The procedure is repeated for all directions of the discretized principal stress space.

The aforementioned procedure can be easily visualized for a two-dimensional problem in two-dimensional principal stress space, as provided in Figure 3.9. Figure 3.9(a) depicts a certain stress increment $\boldsymbol{\sigma}^i$ for an arbitrary direction α^i. The corresponding in-plane principal stresses, σ_1^i and σ_2^i, can be computed accordingly. Subsequently, the stiffness tensor determinant and determinant ratio are calculated. In the case of Figure 3.9(a) it is found that the determinant ratio is above the threshold, meaning that no failure has occurred yet. Therefore, the principal strains are updated for the next stress increment $\boldsymbol{\sigma}^{i+1}$, as shown in Figure 3.9(b). The previous steps are repeated and this time it is found that the determinant ratio is below the threshold, meaning failure, as illustrated in Figure 3.9(b). Following the described procedure, the previous (unfailed) in-plane principal stress state $\boldsymbol{\sigma}^i = \left(\sigma_1^i(\alpha^i), \sigma_2^i(\alpha^i)\right)$ is stored as FDP. This procedure is repeated for all directions of the discretized principal stress space, as visualized in Figure 3.9(c) for eight directions in the top half of the failure envelope. For a sufficient number of directions, a tailored failure envelope corresponding to a specific inter-granular force-law is constructed (see Figure 3.9(d)).


σ_2
σ_1
$\sigma_2^i(\alpha^i)$
$\boldsymbol{\sigma}^i$
α^i
$\sigma_1^i(\alpha^i)$
$\frac{\det(C_{ij}^i)}{\det(C_{ij}^0)} < \mathrm{th}$
$\sigma_2^{i+1}(\alpha^i)$
$\boldsymbol{\sigma}^{i+1}$
FDP
$\sigma_1^{i+1}(\alpha^i)$

(a) Stress increment in arbitrary direction

(b) Next stress increment (failed), store FDP

(c) FDPs for eight directions in top half

(d) Constructed tailored failure envelope

Figure 3.9: Construction of failure envelope from inter-granular force-law

Similar to the two-dimensional procedure as explained above, three-dimensional failure envelopes can be obtained by also considering the third dimension of the three-dimensional principal stress space, σ_3. The generalized procedure is given in Algorithm 2 while the actual codes are provided in Appendix G.3.2.

Algorithm 2 Construction of failure envelope from inter-granular force-law

```
1. Initialization
    (a) Discretize the principal stress space into a sufficient number of directions
    (b) Define stress increments
    (c) Compute the determinant of the macroscopic stiffness tensor in the unloaded state ($\det(C^0_{ij})$)
2. Next direction $i_{dir} \geq 1$
    (a) Apply stress increments in current direction
3. Next stress increment $i_{\boldsymbol{\sigma}} \geq 1$
    (a) Compute principal stress state based on current stress increment ($\boldsymbol{\sigma}^i$)
    (b) Compute inter-granular displacements, internal stresses and macroscopic stiffness tensor from
        previous stress increment ($\boldsymbol{\sigma}^{i-1}$)
    (c) Compute stiffness tensor determinant and check if the determinant ratio $\det(C^i_{ij})/\det(C^0_{ij})$
        is below the specified threshold (failure)
        i. if $\det(C^i_{ij})/\det(C^0_{ij}) >$ threshold then update principal strains: $\mathrm{d}\varepsilon_{ij} = C^{-1}_{ij}\mathrm{d}\sigma_{ij} \to \varepsilon^i_{ij} =$
           $\varepsilon^{i-1}_{ij} + \mathrm{d}\varepsilon^i_{ij} \implies$ go to next stress increment 3
        ii. else if $\det(C^i_{ij})/\det(C^0_{ij}) \leq$ threshold then store previous (unfailed) principal stress state
            as FDP for the current direction $\implies$ go to next direction 2
```

- Determination of failure sensitivity

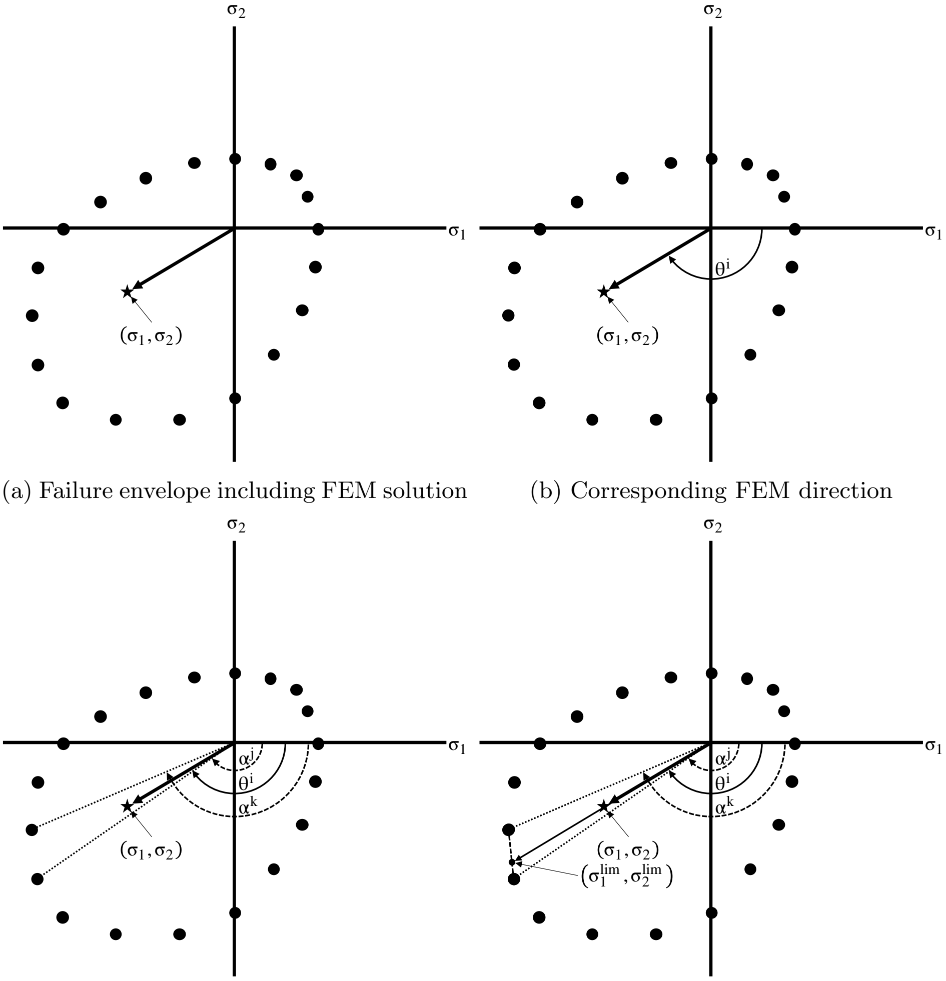
The constraint function values for strength-related topology optimization problems are generally described by classical yield functions or failure criteria, as discussed in Section 2.3.1. However, these classical failure criteria are (continuous) functions of the stress tensor, i.e. these are defined for any direction, in contrast to the GMA-based failure envelopes that are essentially point clouds composed of several FDPs in a number of directions. Therefore, the GMA-based failure envelopes require linear interpolation in order to predict failure in any direction. To this end, the procedure starts with computing the principal stresses from the converged non-linear FEM solution for each element. The corresponding FEM directions in the principal stress space are calculated next, based on the obtained principal stress states. The next step is to find the nearest FDP directions (and the corresponding FDPs) for each element. From the nearest FDPs, the FDPs corresponding to the FEM directions can be obtained through linear interpolation. Once the interpolated FEM FDPs are known, the so-called *failure sensitivities* can be determined.

The failure sensitivity is defined as the ratio of the principal stress state obtained from the converged non-linear FEM solution and the (linearly interpolated) FDP in that direction ($\|\boldsymbol{\sigma}\|/\|\boldsymbol{\sigma}^{lim}\|$). Therefore, the failure sensitivity can be thought of as the division in a normalized failure criterion, such as the first term in Equation (2.67). As a result, the failure sensitivity indicates how sensitive or close a material point is to failure according to its stress state, similar to the Drucker-Prager failure criterion reviewed in Section 2.3.1. Accordingly, the failure sensitivity varies between 0 and 1 provided no failure will occur. A failure sensitivity of 0 means that a material point is non-sensible or very far from failure (i.e. fully intact or unloaded conditions) and a value of 1 tells that a material point is very sensible or prone to failure (i.e. the yield surface's edge has been reached). Failure sensitivities larger than 1 postulate failure.

Identical to the procedure for constructing GMA-based failure envelopes, the procedure explained at the beginning of this section can be easily visualized for a two-dimensional problem in two-dimensional principal stress space (see Figure 3.10). Figure 3.10(a) depicts the tailored failure envelope from Figure 3.9(d) including a certain principal stress state (σ_1, σ_2) from a converged non-linear FEM solution. Subsequently, its corresponding FEM direction (θ^i) can be calculated as shown in Figure 3.10(b). In the case of two-dimensional analysis the two nearest FDP directions (α^j and α^k) and their corresponding FDPs should be found, as illustrated in Figure 3.10(c). From the nearest FDPs, the FDP corresponding to the FEM can be obtained through linear interpolation $(\sigma_1^{lim}(\theta^i), \sigma_2^{lim}(\theta^i))$ as visualized in Figure 3.10(d). Now that the interpolated FEM FDP is known, the failure sensitivity can be determined as $\|\boldsymbol{\sigma}\|/\|\boldsymbol{\sigma}^{lim}\| = \|(\sigma_1(\theta^i), \sigma_2(\theta^i))\|/\|(\sigma_1^{lim}(\theta^i), \sigma_2^{lim}(\theta^i))\|$.

Algorithm 3 Determination of two-dimensional failure sensitivity

```
1. Obtain converged non-linear FEM solution ($\sigma_{ij}$)
2. Compute two-dimensional principal stresses via $\sigma_1, \sigma_2 = \frac{\sigma_{xx}+\sigma_{yy}}{2} \pm \sqrt{\left(\frac{\sigma_{xx}-\sigma_{yy}}{2}\right)^2 + \tau_{xy}^2}$
3. Compute corresponding FEM directions in failure envelope through $\theta^i = \arctan \frac{\sigma_2}{\sigma_1} \left(\frac{180°}{\pi}\right)$
4. Find two nearest FDP directions ($\alpha^j$ & $\alpha^k$) and their corresponding FDPs from $\theta^i$
5. Linearly interpolate FDP: $\sigma_1^{lim}\left(\theta^i\right), \sigma_2^{lim}\left(\theta^i\right) = \boldsymbol{\sigma}^{lim}\left(\alpha^j\right) + \left(\boldsymbol{\sigma}^{lim}\left(\alpha^k\right) - \boldsymbol{\sigma}^{lim}\left(\alpha^j\right)\right)\left(\frac{\theta^i-\alpha^j}{\alpha^k-\alpha^j}\right)$
6. Determine failure sensitivity as $\frac{\|\boldsymbol{\sigma}\|}{\|\boldsymbol{\sigma}^{lim}\|} = \frac{\|(\sigma_1(\theta^i),\sigma_2(\theta^i))\|}{\|(\sigma_1^{lim}(\theta^i),\sigma_2^{lim}(\theta^i))\|}$
```



(a) Failure envelope including FEM solution

(b) Corresponding FEM direction

(c) Two nearest FDP directions and FDPs

(d) Linearly interpolated FEM FDP

Figure 3.10: Determination of failure sensitivity

Similarly, failure sensitivities can be obtained from three-dimensional failure envelopes. However, in the three-dimensional principal stress space four nearest FDP directions and their corresponding FDPs are used for linear interpolation of the FEM FDP: two nearest FDP directions related to the azimuth angle θ and two other corresponding to the polar angle ϕ. Hence finding these four nearest FDPs and subsequently performing the linear interpolation is more sophisticated. Therefore, the two- and three-dimensional procedures are provided separately in Algorithms 3 and 4, respectively. Their related codes are included in Appendix G.1.3.

Algorithm 4 Determination of three-dimensional failure sensitivity

```
1. Obtain converged non-linear FEM solution ($\sigma_{ij}$)
2. Compute three-dimensional principal stresses via $\det(\sigma_{ij} - \lambda I) = 0$
3. Compute corresponding FEM directions in failure envelope through
   $\theta^i = \arccos \frac{\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2+\sigma_3^2}} \left(\frac{180^\circ}{\pi}\right)$ & $\phi^i = \arctan \frac{\sigma_2}{\sigma_1} \left(\frac{180^\circ}{\pi}\right)$
4. Find four nearest FDP directions ($(\theta^j, \phi^j)$, $(\theta^j, \phi^k)$, $(\theta^k, \phi^j)$ & $(\theta^k, \phi^k)$) and their corresponding
   FDPs from $(\theta^i, \phi^i)$
5. Linearly interpolate FDP $x$, $y$ & $z$ coordinates for $(\theta^i, \phi^i)$ from four nearest FDP directions
6. Determine failure sensitivity as $\frac{\|\boldsymbol{\sigma}\|}{\|\boldsymbol{\sigma}^{lim}\|} = \frac{\|(\sigma_1(\theta^i,\phi^i),\sigma_2(\theta^i,\phi^i),\sigma_3(\theta^i,\phi^i))\|}{\|(\sigma_1^{lim}(\theta^i,\phi^i),\sigma_2^{lim}(\theta^i,\phi^i),\sigma_3^{lim}(\theta^i,\phi^i))\|}$
```

Nevertheless, as addressed at the end of Section 2.3.2 the PTO method tends to prefer tensile elements over compressive ones resulting in solutions with a higher structural volume to prevent exceedance of the material's tensile stress limit when applied to the stress-constrained problem for materials exhibiting asymmetric stress limits. Therefore, despite using non-linear FEA the PTOs algorithm will still distribute the material proportionally to the failure criterion values yielding a larger portion (if not all) to be assigned to tensile elements.

The two-dimensional GMA-based failure envelope corresponding to the strength-related inter-granular force-law, the bi-linear with hardening (Equations (3.7) to (3.9)), that is considered in the topology optimization algorithm is displayed below. The evolution of the determinant of the macroscopic stiffness tensor is plotted against the stress increments for five different directions in two-dimensional principal stress space as indicated in the figure. The GMA-based failure envelope has been obtained using $\lambda_1 = 0.1$, $\lambda_2 = 0.1$ and $\beta = 1/6$ (see Figure 3.11). However, for a reasonable failure envelope for that input the threshold for the determinant ratio had to be decreased to 10^{-2}, meaning a loss of $(1 - 0.01) \times 100\% = 99\%$ of the initial stiffness is regarded as failure of the material. The failure envelope is discretized into 72 directions (i.e. every 5 degrees) and normalized with respect to its corresponding uniaxial compressive strength, f_c. The corresponding three-dimensional GMA-based failure envelope that is obtained using the same numerical criterion can be found in Figure 3.12.

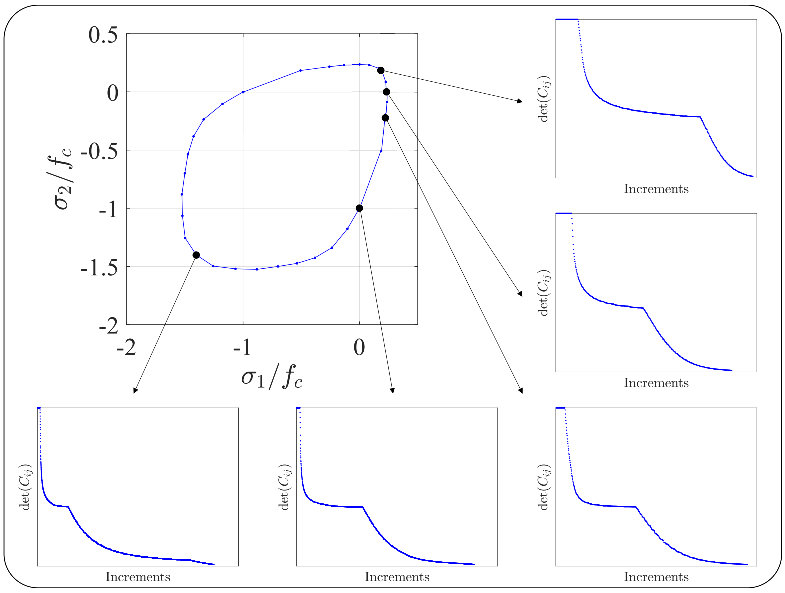
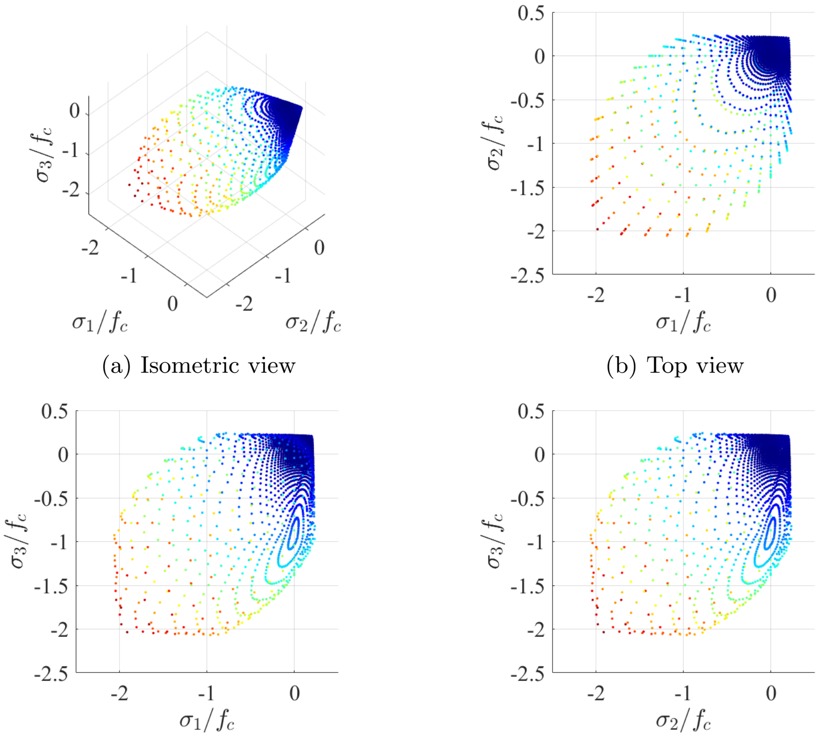


Figure 3.11: GMA-based failure envelope from bi-linear inter-granular force-law



(a) Isometric view

(b) Top view

(c) Front view

(d) Side view

Figure 3.12: GMA-based failure envelope from bi-linear inter-granular force-law

4 Results

This section presents the first implementations of the developed non-linear FEM using the multi-scale material model into topology optimization algorithms to demonstrate the effect of the GMA on topology optimization. To this end, the four benchmark examples as presented in Figure 3.5 are considered first for the compliance minimization problem. Afterwards, the four smaller benchmarks included in Figure 3.7 are considered for both the minimum compliance and stress-constrained problems using the PTO method. Each problem ends by considering the simple three-dimensional problem in Figure 3.8 to illustrate the abilities of the developed non-linear FEM using the multi-scale material model for three-dimensional topology optimization applications.

4.1 Proportional Topology Optimization Using Compliance

As provided in Section 3.3.1, the material is here assumed to be isotropic in the unloaded condition such that $\xi(\theta, \phi) = 1/4\pi$ having an unloaded Young's modulus of 52 GPa and a Poisson's ratio of 0.11 with the stiffness parameters $k_{\mathrm{n}} = 2$ kN/m and $k_{\mathrm{w}} = 1$ kN/m, while the grain size $l = 10$ μm and $N_p = 10^{18}$ m^{-3}. All structures are subjected to a concentrated point load of 1 N with the volume constraint set at 60% of the total design domain. Recall that the standard filter radius is set to 1.5 and that the convergence criterion needed to be relaxed to 10^{-2} for the tension-compression asymmetry inter-granular force-law in order to achieve convergence in the non-linear FEM. There are three stopping criteria for the compliance problem used in this research, these are: (i) the change in density between two successive iterations should be smaller than 0.1, (ii) the change in compliance between two successive iterations should be less than 0.1 and (iii) a minimum amount of 50 iterations should be satisfied.

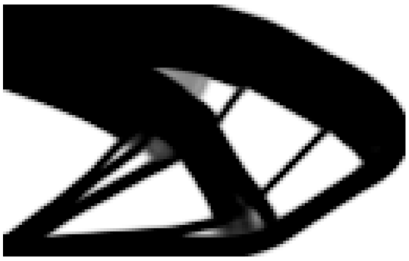


Figure 4.1: Optimized topology of cantilever beam problem for $\lambda = 0.1$

4.1.1 Cantilever Benchmark

Starting with the cantilever beam problem having 128 × 80 elements (see Figure 3.5(a)), which is a recognized benchmark structure in structural topology optimization problems. Figure 4.1 shows the structure's optimized topology for $\lambda = 0.1$, in which it can be seen that the tensile regions (having lower stiffness) have a larger size compared to the compressive regions. Actually, the optimized topology is composed of significantly more tensile (lower stiffness) elements and less compressive (higher stiffness) elements. This is a logical result, because lower stiffness regions require more cross-sectional area to transfer the applied load. In terms of accommodation of displacements, if the stiffness goes down by a factor α, stress goes down by the same factor such that the cross-sectional area should go up by a factor α. Another point of view is to use a spring analogy, in which the compressive regions can be considered to have a spring stiffness k_c and the tensile regions having a ten times smaller spring stiffness, i.e. $k_t = \lambda k_c = 0.1k_c$. In order to satisfy static equilibrium, the tensile regions should accommodate the same (spring) stiffness as the compressive regions. According to Hooke's law for springs in parallel, ten springs having spring stiffness k_t are required to reach an equivalent spring stiffness, i.e. $10k_t = k_c$. However, this specific result has also been validated by means of a simple comparison study with the commercial software ABAQUS (see Appendix D).

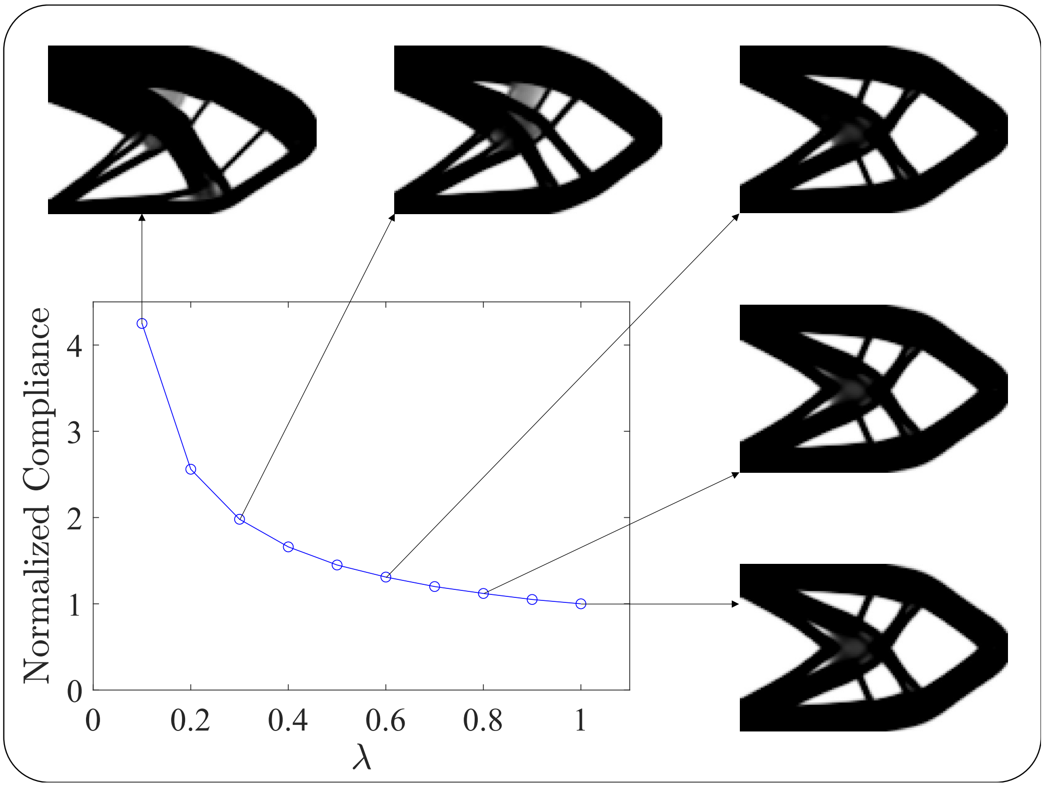


Figure 4.2: Topology variation and normalized objective function for different λ

Furthermore, the effect of different tension-to-compression stiffness ratios, λ, is illustrated in Figure 4.2 where the structure's optimized topologies are shown for varying values of λ. It is noteworthy that the asymmetry of the optimized topologies increases as λ decreases from 1 to 0.1 for the constant volume constraint of 60%. The tensile regions increase and the compressive regions decrease in size for lower values of λ. For equal microscopic stiffness in tension and compression ($\lambda = 1$), the structure is symmetric about the horizontal axis having equally sized tensile and compressive regions, as found in classical topology optimization results. Lower values of λ decrease the inter-granular stiffness in tension, which causes softening of the material at macro-scale. To minimize compliance, the optimization algorithm distributes more (softer) elements to the tensile regions causing removal of (stiffer) elements from the compressive regions. Moreover, Figure 4.2 shows that the overall compliance increases for lower values of λ, which is expected. Additionally, it is remarkable that the compliance non-linearly increases towards lower values of λ, where the lowest values cause the largest increase in compliance.

4.1.2 Corner Loading Cantilever Benchmark

Continuing with the corner loading cantilever benchmark example, also having 128 × 80 elements (see Figure 3.5(b)), the same material distribution appears to be happening as for the previously discussed cantilever beam problem. For this case, Figure 4.3 depicts the effect of λ on the final topology. The same trend is observed that if the tensile microscopic stiffness decreases for a constant volume constraint, more material gets distributed towards tensile regions and compressive regions are reduced.

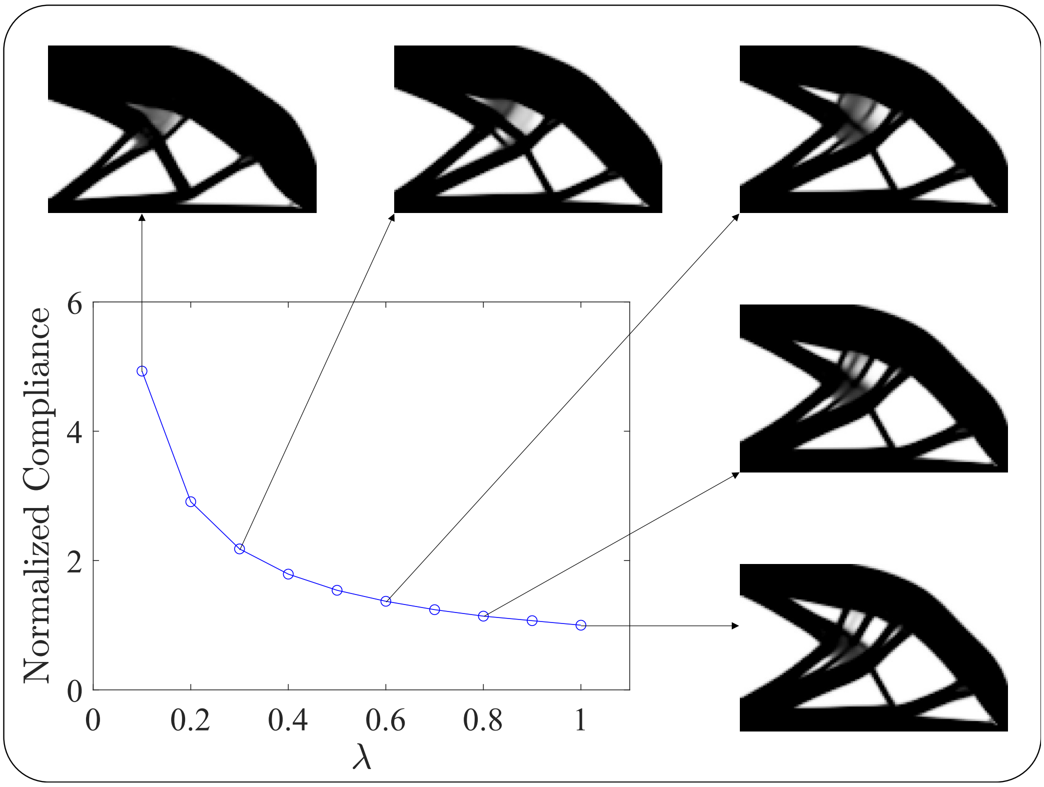


Figure 4.3: Topology variation and normalized objective function for different λ

4.1.3 Michell Structure Benchmark

Moving on to the Michell structure having 160 × 80 elements (see Figure 3.5(c)), another known benchmark problem in the field of topology optimization. Here the effect of the tension-to-compression stiffness ratio λ is provided in Figure 4.4. Again, this figure illustrates that the tensile regions are larger compared to the compressive regions for decreasing λ.

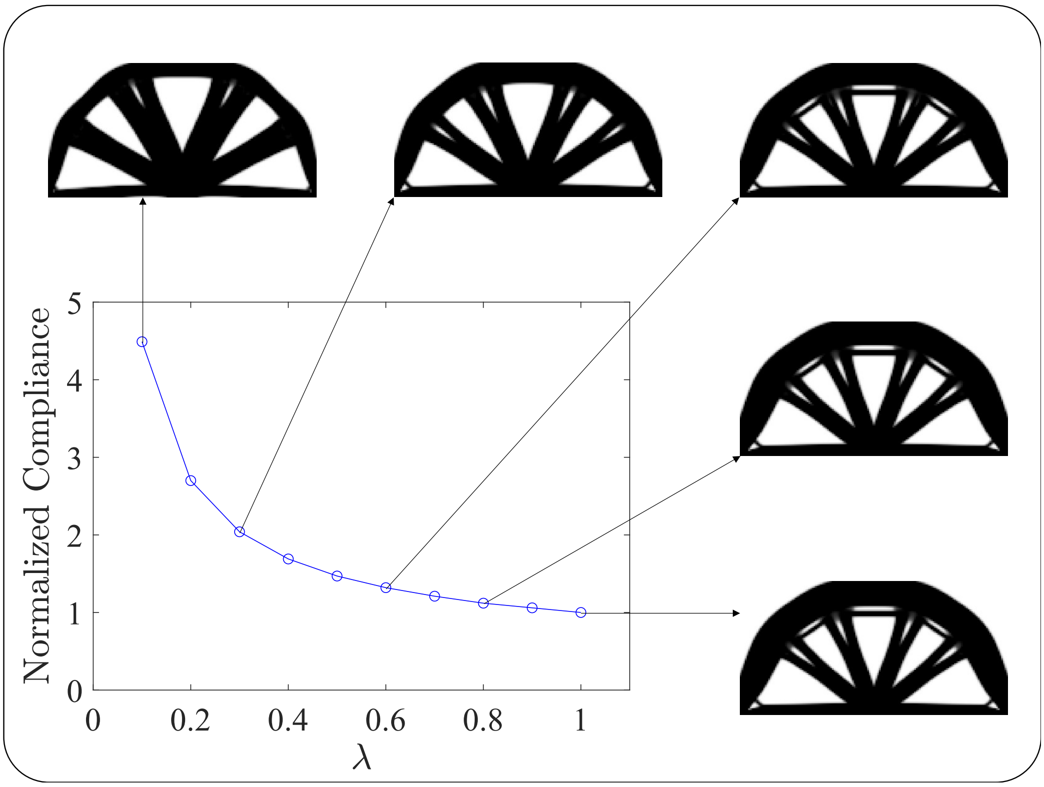


Figure 4.4: Topology variation and normalized objective function for different λ

4.1.4 MBB Structure Benchmark

The final large benchmark example considered for PTOc only, before going into the smaller benchmark problems (Figure 3.7), is the well-known MBB beam (see Figure 3.5(d)) having 180 × 60 elements. The height-to-length ratio for this problem is usually taken as 1/6 for a full beam, resulting in a ratio of 1/3 for one half. Figure 4.5 shows the effect of λ from which it can be concluded once more that tensile regions increase and compressive regions decrease in size for lower values of λ given a constant volume constraint.

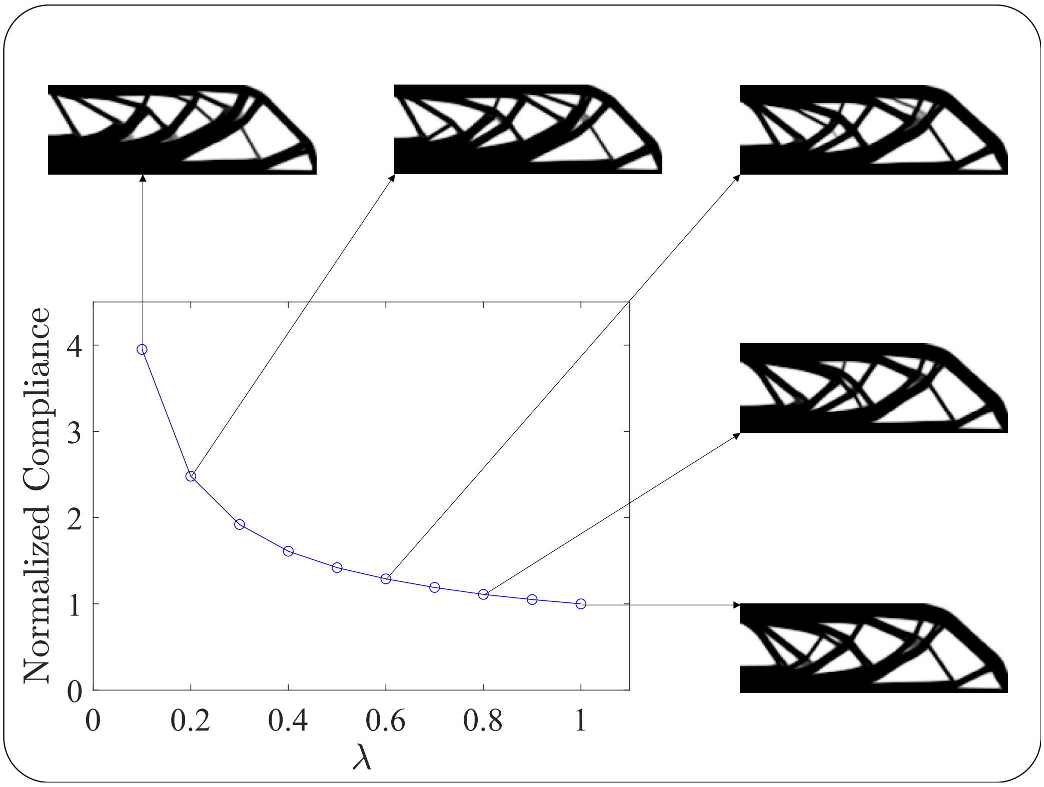


Figure 4.5: Topology variation and normalized objective function for different λ

From hereon, the smaller benchmark examples from Figure 3.7 are considered. Remember that for these cases, the volume fraction (i.e. the volume constraint) is set to 0.25, meaning that only 25% of the design domain may be occupied by material. However, the single rod benchmark uses a volume fraction of 0.5 and has a different filter radius than the standard value of 1.5: this specific benchmark example in Figure 3.7(a) ignores filtering by setting r_0 equal to 1.

4.1.5 Single Rod Benchmark

The first smaller benchmark is the single rod structure with an axial load acting at mid-span (see Figure 3.7(a)). Figure 4.6 shows the effect of different tension-to-compression stiffness ratios, λ, on the structure's optimized topology. In this case, the compliance varies remarkably different with respect to λ and the material distribution towards tensile and compressive regions is manifested differently. This is caused by the boundary conditions of this particular problem having two symmetries, namely one about the horizontal axis and one about the vertical axis. Therefore, the mechanism of load bearing is illustrated because the structure is statically indeterminate. This means that the structure is able to make equilibrium by only having either (less stiff) tensile elements or (stiffer) compressive elements, in contrast to the larger benchmark problems discussed earlier that do require both tensile and compressive members because of bending mechanisms being present within these structures. For equal microscopic stiffness in tension and compression ($\lambda = 1$), the structure has a symmetry about both the horizontal and vertical axes, identical to its boundary conditions. However, Figure 4.6 shows that already for the smallest asymmetry in inter-granular stiffness (e.g. $\lambda = 0.9$), the optimized topology becomes a compression-only structure. As a result, it is noteworthy that the compliance is hardly affected by λ, which is logical since it has no effect on the microscopic compressive stiffness. For $\lambda = 1$, both elemental density values are a half ($x_e = 0.5$) causing penalization of both their stiffness tensors yielding larger displacements and therefore a higher overall compliance value.

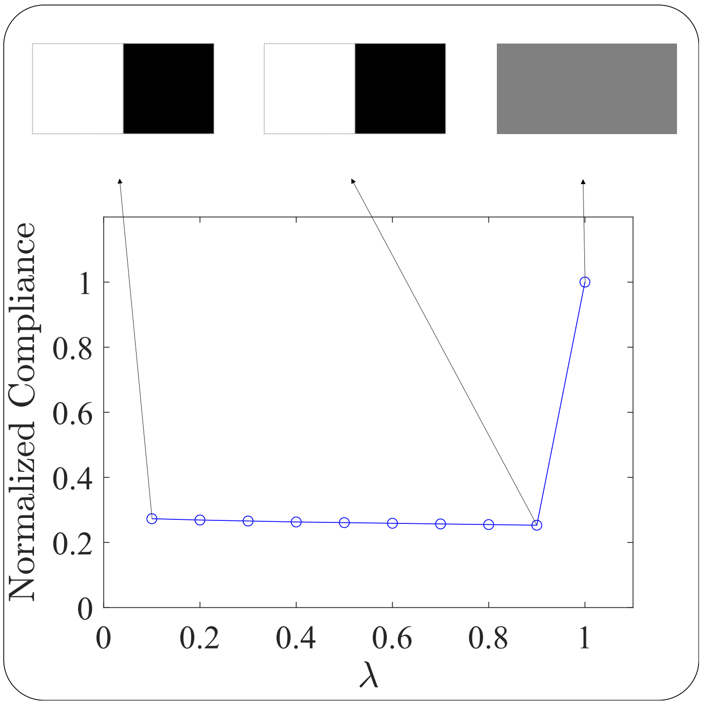


Figure 4.6: Topology variation and normalized objective function for different λ

4.1.6 Four-Bar Truss Benchmark

The second smaller benchmark example is a problem referred to as the ”four-bar truss” benchmark (see Figure 3.7(b)), named after the typical X-shaped optimized topology that is found for symmetric material behaviour. For this case, Figure 4.7 depicts the effect of λ on the final topology. Similar to the single rod benchmark, the variation of compliance and distribution of material are different here for the same reasons. The boundary conditions of the four-bar truss benchmark are also double-symmetric, having a symmetry about the horizontal and vertical axes, allowing for a compression-only load bearing mechanism due to the absence of bending mechanisms. Accordingly, for $\lambda = 1$ the optimized topology is double-symmetric as well as its boundary conditions. Analogous to Figure 4.6, Figure 4.7 illustrates that also in this case the optimized topology immediately becomes fully compressive for the slightest asymmetry in the microscopic constitutive relationship. However, contrary to the single rod structure, the compliance shows to be more affected by λ in Figure 4.7. The variation in compliance with respect to λ is related to the contribution of void elements: even though these might not necessarily contribute to the stiffness of the structure, their strain energies are computed and added to the overall compliance anyway. Therefore, tensile elements being void for higher values of λ (e.g. $\lambda = 0.9$) will have somewhat smaller displacements and thus compliance values compared to those for smaller λ values. This effect scales by the number of elements (being tensile), causing a minor variation in the overall compliance with respect to λ. As a result, it is hardly noticeable for the single rod benchmark in Figure 4.6 having only two elements. Moreover, the compliance values for $\lambda = 1$ and $\lambda = 0.1$ show to be similar. This observation has also been validated through comparison with ABAQUS (see Appendix D).

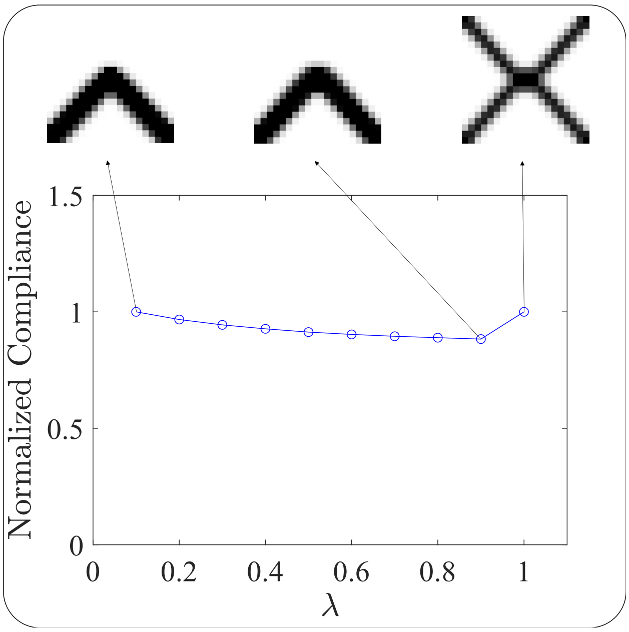


Figure 4.7: Topology variation and normalized objective function for different λ

4.1.7 Two-Bar Truss Benchmark

Proceeding with the two-bar truss benchmark where the design domain is fully fixed by means of hinged supports at each node of the bottom edge and loaded by a horizontal force at the center of the upper edge (see Figure 3.7(c)). The force is distributed over the three central nodes at the upper edge of the FE mesh. In this case, the load bearing mechanism requires the presence of both tensile and compressive members. The structure's optimized topologies are depicted for varying values of λ in Figure 4.8 to show its effect. Now the same trend as for the larger benchmark examples (see Figures 4.2 to 4.5) is observed that if the tensile inter-granular stiffness decreases for a constant volume fraction, more material is distributed towards tensile regions and compressive regions are reduced. For equal microscopic stiffness in tension and compression ($\lambda = 1$), the classical result having a symmetry about the vertical axis and equally sized tensile and compressive regions is found: both trusses have angles of 45° with respect to the horizontal. Identical to Figure 4.2, the asymmetry of the optimized topologies increases as λ decreases from 1 to 0.1 for a constant volume constraint. For $\lambda = 0.1$, there are more compressive than tensile elements with a tensile-to-compressive area ratio of 2/3. In this case, the width of the design domain allows the topology optimization algorithm to steer towards a more direct compressive load transfer by decreasing the angle of the compressive member and increasing the angle of the tensile member with respect to the horizontal. From nodal analysis (see Appendix E) one can see that by altering the angles as shown in Figure 4.8, the compressive force component increases while the tensile force component decreases from a tensile-to-compressive force component ratio of 1/1 for symmetric microscopic stiffness ($\lambda = 1$) to a ratio of 1/2 ($\lambda = 0.1$).

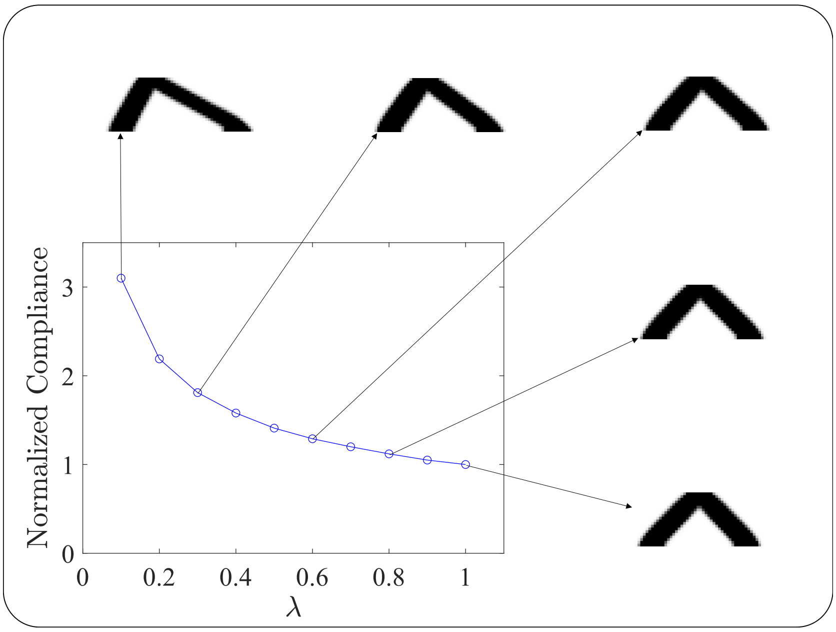


Figure 4.8: Topology variation and normalized objective function for different λ

4.1.8 MBB Beam Benchmark

The MBB beam is considered once more for a smaller number of elements, 60×20, and lower volume constraint of 25% (see Figure 3.7(d)). Figure 4.9 shows the effect of λ from which it can be concluded again that more material gets assigned to tensile regions and compressive regions are reduced for lower values of λ under a constant volume constraint. However, in this case the asymmetry in stiffness does not necessarily yield (large) differences in the global topology. The differences are more nuanced here, since these are mostly expressed in the element densities of different structural members (i.e. their grey scale): for $\lambda = 1$, one can clearly see that all trusses have at least one band of fully dense (black) elements while for $\lambda = 0.1$ the tensile trusses are fully solid (black) and compressive trusses have less material (grey). Furthermore, it is noteworthy that the non-linear compliance versus λ curves of both MBB benchmark analysis are exactly the same.

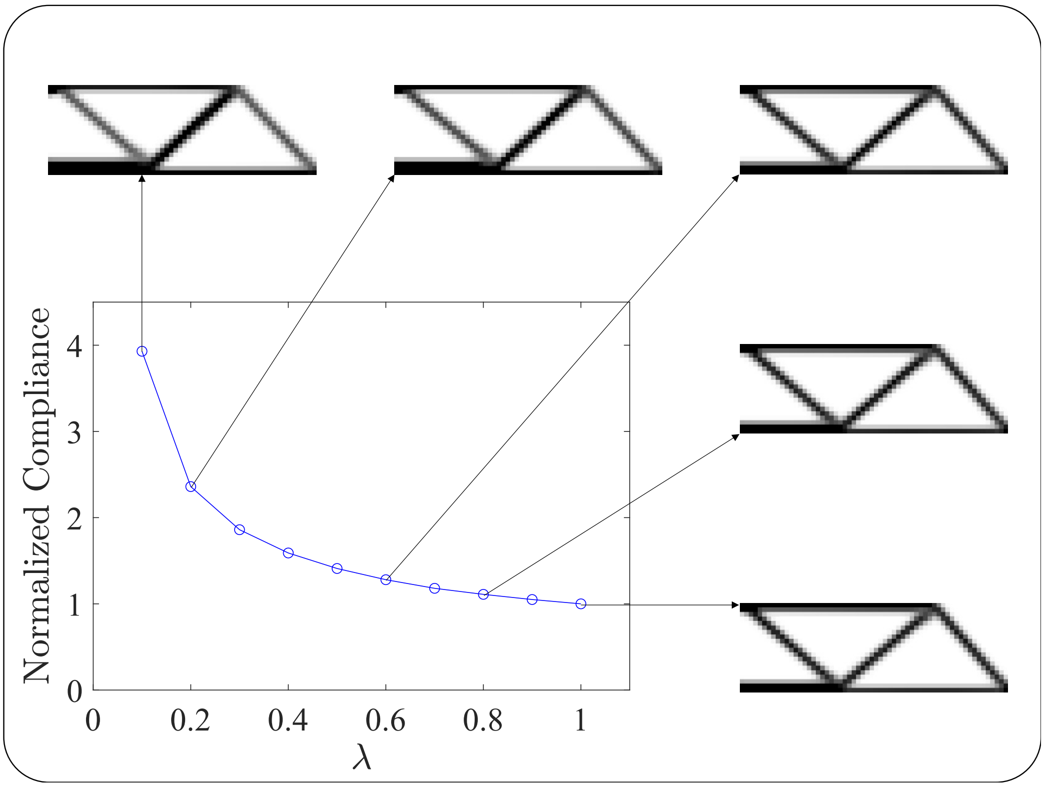


Figure 4.9: Topology variation and normalized objective function for different λ

4.1.9 Three-Dimensional Representation of Single Rod Benchmark

Finally, a three-dimensional representation of the single rod benchmark is considered (see Figure 3.8) to illustrate the abilities of the developed non-linear FEM using the GMA for three-dimensional applications. The three-dimensional single rod benchmark uses a volume fraction of 0.5 and ignores filtering by setting r_0 equal to 1. For the three-dimensional implementations, the stopping criteria are slightly different: (i) the change in density between two successive iterations should be smaller than 0.001 and (ii) the maximal amount of iterations equals 50. Figure 4.10 shows the effect of different tension-to-compression stiffness ratios, λ, on the structure's optimized topology. In this case, the compliance varies in a similar manner with respect to λ as for the two-dimensional problem (see Figure 4.6). This is caused by the same reasons as explained under the two-dimensional case. Therefore, equal microscopic stiffness in tension and compression ($\lambda = 1$) yields a double symmetric structure, identical to its boundary conditions. Furthermore, Figure 4.10 illustrates once more that the optimized topology becomes compressive for even the smallest asymmetry in the inter-granular stiffness (e.g. $\lambda = 0.9$). As a result, the compliance is hardly affected by λ since it has no effect on the microscopic compressive stiffness. In the case where $\lambda = 1$, both elemental density values are a half ($x_e = 0.5$) causing penalization of both their stiffness tensors yielding larger displacements and therefore a higher overall compliance value.

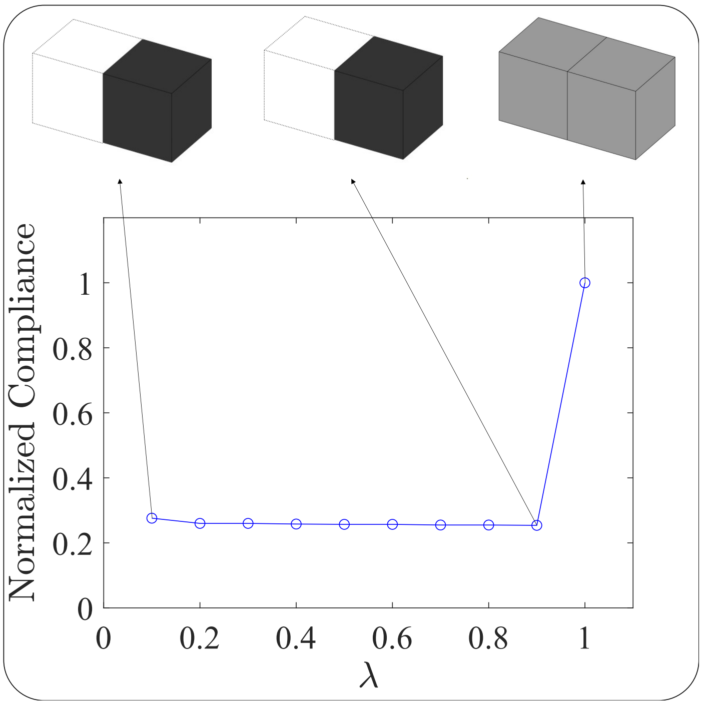


Figure 4.10: Topology variation and normalized objective function for different λ

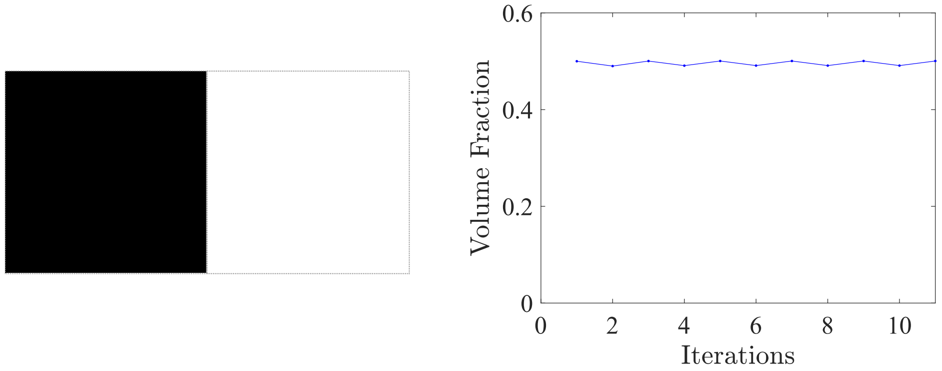
4.2 Proportional Topology Optimization Using Stress

Even though it was found in a previous research [5, 23] that the PTO method has a tendency to prefer tensile elements over compressive elements for pressure-dependent materials having unequal stress limits in tension and compression (such as concrete), the method has also been applied to the stress-constrained problem for completeness and comparison. Previous results using linear-elastic FEM have shown solutions having higher structural volume to prevent the exceedance of the material's tensile stress limit, as explained in Section 2.3.2. In the present work, the developed non-linear FEM algorithms using the GMA have been implemented into the topology optimization algorithm. However, despite using non-linear FEM with a sophisticated multi-scale material model the PTO method still obtains the less optimal, tensile topologies. This is demonstrated here by means of the four smaller benchmark examples depicted in Figure 3.7 and the simple three-dimensional one as shown in Figure 3.8.

In this case, the numerical input used for the GMA has been chosen to mimic the material characteristics of concrete for which the material is assumed to be isotropic in the unloaded condition such that such that $\xi(\theta, \phi) = 1/4\pi$ having an unloaded Young's modulus of 30 GPa and a Poisson's ratio of 0.2 with the stiffness parameters $k_\mathrm{n} = 5$ MN/m and $k_\mathrm{w} = 0.83$ MN/m, while the grain size $l = 0.1$ mm and $N_p = 3 \cdot 10^{12}$ m^{-3} (see Section 3.3.2). All structures initially start with a volume fraction set at 50% of the total design domain. The stopping criteria for the stress constraint problem used in this work are: (i) the change in volume between two successive iterations should be smaller than 0.1 and (ii) the maximal stress constraint function value of any element may not exceed its limit (i.e. 1 for the PTO method). The results presented here are obtained using the bi-linear inter-granular force-law with hardening.

4.2.1 Single Rod Benchmark

The single rod benchmark (Figure 3.7(a)) is considered here to have horizontal and vertical dimensions of 0.2 m and 0.1 m, respectively. The axial loads acting at mid-span have a magnitude of 2.6 kN each and are pointed towards the right. Figure 4.11 shows the optimized topology and evolution of the objective function for this particular configuration. Identical to the results for linear-elastic FEM, the PTOs result in Figure 4.11(a) depicts the less optimal tensile bar despite the material having a higher compressive stress limit. The final volume fraction yields 0.5, which means that 50% of the total design domain is occupied by material. The final maximal stress constraint function value equals 0.991, meaning that the material is exploited up to 99.1% of its ultimate strength. Furthermore, it is notable that the volume fraction evolves linearly with respect to the iterations.



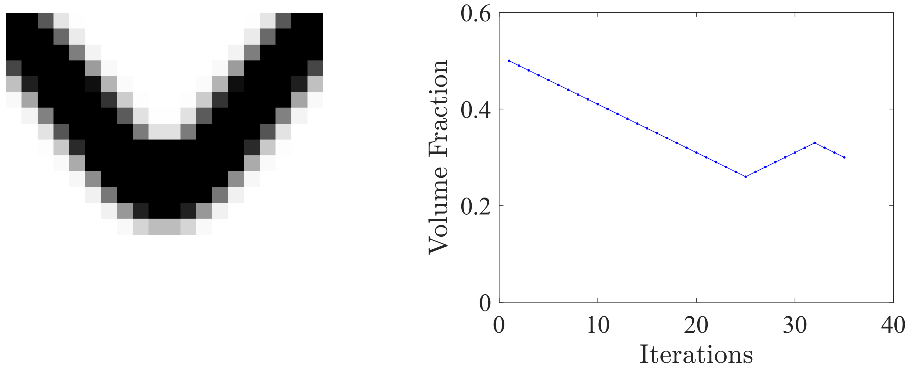
(a) Optimized topology (density distribution)

(b) Objective function versus iterations

Figure 4.11: Optimized topology and evolution of objective function

4.2.2 Four-Bar Truss Benchmark

The four-bar truss benchmark (Figure 3.7(b)) is modelled to have horizontal and vertical dimensions of both 0.2 m. The point load acting at its center has a magnitude of 1.325 kN pointing downwards. For this case, Figure 4.12 displays the obtained material distribution and objective function's evolution. Again, the density distribution in Figure 4.12(a) yields a tensile solution while the material is in fact stronger in compression. Now the final volume fraction is 0.3 with a maximal stress constraint function value that equals 0.993.



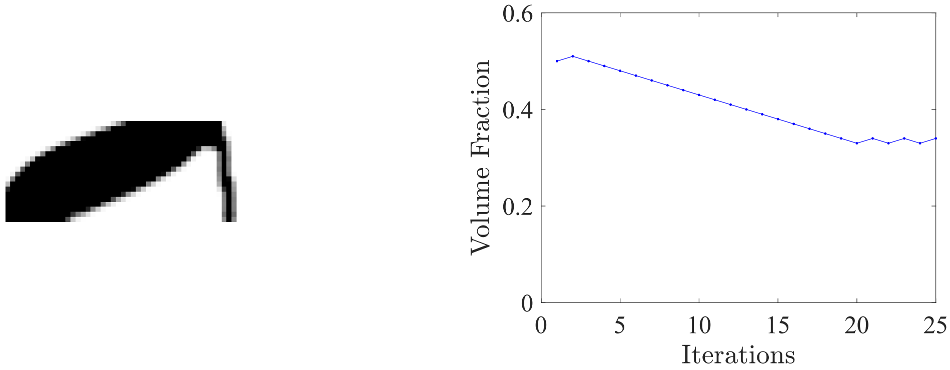
(a) Optimized topology (density distribution)

(b) Objective function versus iterations

Figure 4.12: Optimized topology and evolution of objective function

4.2.3 Two-Bar Truss Benchmark

Continuing with the two-bar truss benchmark (Figure 3.7(c)) having horizontal and vertical dimensions of 0.8 m and 0.2 m, respectively, and three distributed horizontal forces at the center mesh nodes of its upper edge with a magnitude of 1.575 kN each towards the right. The structure's optimized topology and evolution of its objective function are shown in Figure 4.13. Compared to the classical symmetric result having equally sized tensile and compressive regions with angles of 45° with respect to the horizontal, in this case the topology optimization algorithm steers towards a solution having a longer and thicker tensile member while the compressive member is shortened and slimmer (see Figure 4.13(a)). This is similar to the result found for linear-elastic FEM. Contrary to the results for compliance minimization (see Figure 4.8), the angle of the compressive member is significantly increased and the angle of the tensile member is decreased with respect to the horizontal due to the method's tendency to prefer tensile elements. Now the tensile-to-compressive area ratio yields roughly 9.5. Through nodal analysis (see Appendix E) it can be showed once more that by altering the angles as in Figure 4.13, the compressive force component decreases while the tensile force component increases which leads to a tensile-to-compressive force component ratio of about 2.2/1. Therefore, it makes sense that the compressive member is thinned while the tensile member is thickened in order to limit the normal stresses occurring in the members. Here the final volume fraction equals 0.34 having a maximal stress constraint function value of 0.994.



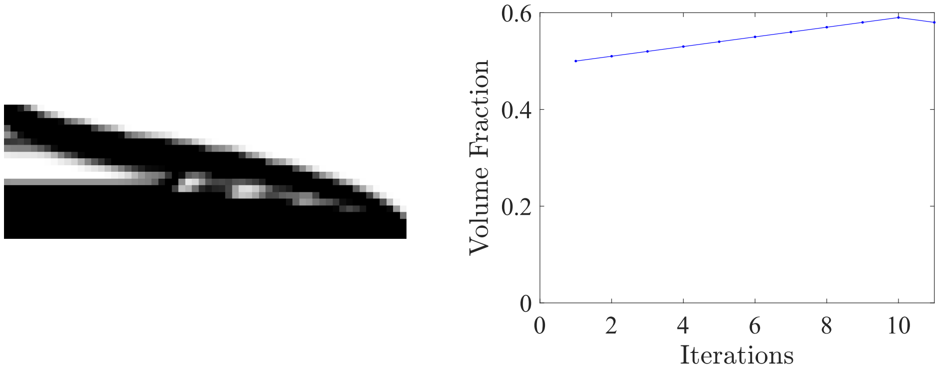
(a) Optimized topology (density distribution)

(b) Objective function versus iterations

Figure 4.13: Optimized topology and evolution of objective function

4.2.4 MBB Beam Benchmark

The smaller MBB beam benchmark (see Figure 3.7(d)) having 60 × 20 elements is considered again with horizontal and vertical dimensions of 0.6 m and 0.2 m, respectively. The point load acting downwards has a magnitude of 1.4 kN. Figure 4.14 depicts the structure's optimized topology and evolution of the objective function. It can be seen that in contrast to the other results in this section, the volume fraction is increased here in order to prevent exceedance of the stress constraint. In this case, the final volume fraction yields 0.58 with a maximal stress constraint function value equal to 0.958.



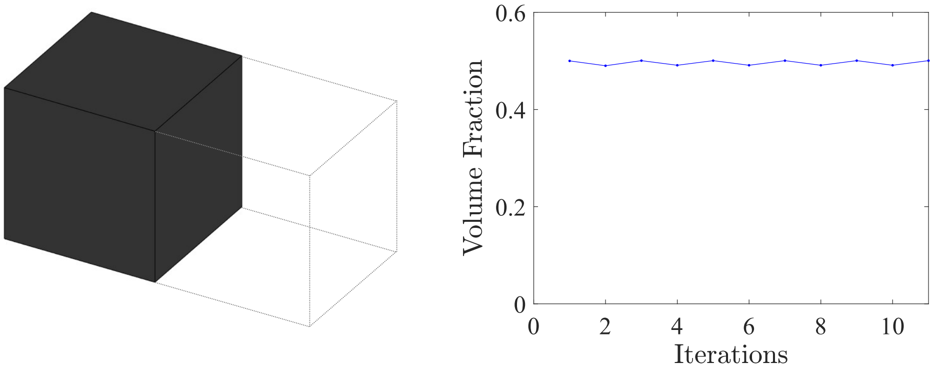
(a) Optimized topology (density distribution)

(b) Objective function versus iterations

Figure 4.14: Optimized topology and evolution of objective function

4.2.5 Three-Dimensional Representation of Single Rod Benchmark

Lastly, the three-dimensional representation of the single rod benchmark (Figure 3.8) is considered here to have dimensions of 0.2 m, 0.1 m and 0.1 m in the x, y and z directions, respectively. The four axial loads acting at mid-span have a magnitude of 130 kN each and are pointed towards the positive y direction. Recall that the stopping criteria for the three-dimensional implementations are slightly different. In this case, this yields: (i) the change in volume between two successive iterations should be less than 0.01 and (ii) the maximal stress constraint function value may not exceed its limit value of 1. Figure 4.15 shows the optimized topology and evolution of the objective function for this configuration. Similar to the results for the two-dimensional case, Figure 4.15(a) depicts a tensile solution despite the material having a higher compressive stress limit. This is what makes the found topology suboptimal. The final volume fraction yields 0.5, which means that exactly 50% of the total design domain is occupied by material. The final maximal stress constraint function value equals 0.983, meaning that the material is exploited up to 98.3% of its ultimate strength. Please note that the three-dimensional boundary conditions allow for the applied axial loads to be about two orders of magnitude larger compared to the two-dimensional problem. This is due to the plane stress conditions imposed on the two-dimensional benchmark, which has a small element thickness that is only one hundredth of the two other dimensions. As a result, the three-dimensional problem has a thickness and thus a cross-sectional area that is a hundred times larger which allows the axial loads to be about two orders of magnitude larger to end up with the same (normal) stress level.



(a) Optimized topology (density distribution)

(b) Objective function versus iterations

Figure 4.15: Optimized topology and evolution of objective function

5 Discussion

The focus of the presented research was on topology optimization of cementitious materials (such as 3DCP) to incorporate their specific properties and evolution of microstructure during construction and loading. To this end, the main objective has been to see the effects of non-linear and anisotropic behaviour of cement-based materials on their optimized topologies. The results found in previous sections will be evaluated next, together with several additional remarks concerning the inclusion of material non-linearities.

5.1 Compliance Minimization

For the compliance minimization problem, the nine considered benchmark problems clearly demonstrate the effects of the tension-compression asymmetric intergranular force-law on the optimized topologies. The optimized topologies found for asymmetric microscopic stiffness are very different from the classical optimized topologies found for symmetric behaviour in tension and compression that do not account for induced anisotropy. From the results in Section 4.1, it can be seen that generally the tensile regions have a larger size compared to the compressive regions under tension-compression asymmetry. However, in some particular cases the optimized topology will only contain compressive elements if the boundary conditions allow for this (i.e. statically indeterminacy or absence of bending mechanisms). By definition, the PTO method is tempted to distribute the material to the highest objective function values, which in this case are the elemental compliance values. Since the weakest (tensile) elements will yield the largest compliance values, the distribution algorithm will locate more material towards tensile regions. As a result, even though the PTO method does not rely on sensitivity information, it provides logical and justifiable results for the minimum compliance problem under tension-compression asymmetric material behaviour. However, it should be noted that sensitivity information is helpful to accelerate a topology optimization algorithm such that it may provide faster convergence.

5.2 Stress-Constrained Volume Minimization

On the other hand, for the stress-constrained problem the five considered benchmark examples demonstrate that the PTO method obtains less optimal tensile topologies, despite using non-linear FEM with a sophisticated multi-scale material model. Even though the material is in fact stronger in compression, tensile solutions are found as seen in Section 4.2. In this case, this is caused by the tendency to distribute more material to the largest constraint function values of the PTO method. For opposite stress states with equal magnitudes, the distribution algorithm will always locate more material towards tensile regions that are closer to material failure. Because the PTO method does not consider sensitivity information to steer the optimization process, it has no incentive to actually minimize the objective function (i.e. the structural volume) for asymmetric stress limits in tension and compression. It only prevents exceedance of the stress constraints instead. Consequently, the PTO method does provide tensile results for the stress-constrained problem under unequal stress limits as opposed to the expected compression-dominant structures. In this case, it is clear that sensitivity information is required to steer the topology optimization algorithm towards compression-dominant structures. On that score, it is recommended to use TTO as the topology optimization method for stress-constrained volume minimization under asymmetric strength.

In view of the observation that the TTO method was found to be most optimal for materials having asymmetric stress limits in tension and compression and the critical flaw of the PTO method for pressure-dependent materials as described above, an alternative approach to obtain sensitivity information should be considered. Because the derivative of the objective function with respect to the design variable cannot be obtained analytically for the multi-scale material model, numerical sensitivity information could be obtained from the GMA and used in the TTO method. However, the implementation of numerical sensitivity information in combination with the mathematical solvers that this method uses will be left for follow up work with respect to the reserved time for the current research.

5.3 Material Non-Linearities

The effect of including material non-linearities into the structural analysis is also dependent on the topology optimization methodology. In the compliance minimization problem, inclusion of non-linear material behaviour does influence the optimized topologies. In this case, the evolution of the macroscopic tangential stiffness tensor is an important feature for the path analysis. Therefore, the minimum compliance problem does also provide logical and justifiable results for microscopic constitutive relationships that have asymmetric strengths in tension and compression, such as the strength-related inter-granular force-law that has been used in the stress-constrained problem. Since the stiffness decreases when approaching the ultimate strength (i.e. failure) and the tensile strength is lower than the compressive strength, the tensile stiffness will decrease earlier which leads to larger compliance values. However, in the stress-constrained problem inclusion of material non-linearities hardly affects the optimized topologies. In that case, only the proximity to material failure (i.e. the stress constraint) is important in the limit analysis.

At the same time, using non-linear FEM with incorporation of a multi-scale material model may enable utilization and implementation of failure criteria that consider the stress-path in contrast to (classical) failure criteria in terms of the stress tensor where only the stress state is considered. The developed `MATLAB` codes allow for the use of failure criteria that change with respect to the loading path, thus circumventing the deficiencies of failure criteria defined as a function of stress tensor through incorporation of load-path dependency. In view of the aforementioned discussion on failure criteria, yet another topology optimization methodology could be proposed: stiffness-constrained volume minimization. In this case, the failure criterion could be written as $\mathbb{F}\left(C_{ijkl}\right) = 0$, defined as a general function of the tangential stiffness tensor C_{ijkl} of a material. However, this implementation will also be left for follow up work with respect to the reserved time for the current research.

5.4 Further Remarks

As a closing note, it should be mentioned that at the start of this project, entirely new concepts were required to be acquainted by the author. In order to understand the multi-scale material model, the Granular Micromechanics Approach (GMA), it was necessary to get familiar with concepts like continuum mechanics, summation convention, tensorial calculus, particle mechanics and how to relate different scales through energy. Knowledge on these subjects had to be obtained over the course of this research. Furthermore, all the codes and algorithms used and referred to throughout this work are modelled using the platform `MATLAB`. Since this software was entirely new to the author, it had to be learned during this project as well. Starting with small tutorials and help from (study) friends, the workflow of `MATLAB` got internalized. The publicly available 99-line and 88-line `MATLAB` implementations for two-dimensional topology optimization by Sigmund [36] and Andreassen et al. [33], respectively, have also been very helpful in understanding how to use the software. As a result, some of the developed codes could have been more efficient because these have been written in the earlier days of the research, such as the non-linear FEM algorithms. Over the course of the project, the `MATLAB` skills have improved to increase efficiency of the codes, i.e. more efficient data storage and vectorizing `for` loops.

6 Conclusion

The focus of the present research is on topology optimization of cementitious materials (such as 3DCP) to incorporate their specific properties and evolution of microstructure during construction and loading. To this end, the main objective of this graduation thesis has been to see the effects of non-linear and anisotropic behaviour of cement-based materials on their optimized topologies. To implement the GMA to do so, a non-linear Finite Element Method (FEM) was required that has been developed using the Newton-Raphson iterative method. The Granular Micromechanics Approach (GMA) has enabled incorporation of the asymmetric nature of particle interactions into the continuum model in a computationally affordable manner through which the model was capable of capturing the anisotropic evolution of a material's microstructure. Two popular topology optimization methodologies have been considered, which are the compliance minimization problem for a given volume constraint and the stress-constrained volume minimization problem under a certain stress constraint. Both problems were implemented into the Proportional Topology Optimization (PTO) method to perform two-dimensional as well as three-dimensional optimizations. The resulting topology optimization algorithms have been used to study the effect of the sophisticated multi-scale material model on optimized topology.

The results that have been obtained using the developed `MATLAB` codes for the compliance minimization problem clearly demonstrate the effects of asymmetric stiffness on the optimized topologies. The optimized topologies found for asymmetric microscopic stiffness are very different from the classical optimized topologies found for symmetric behaviour in tension and compression that do not account for induced anisotropy. Generally, tensile regions gain a larger size compared to compressive regions, because lower stiffness (tensile) regions require more cross-sectional area to transfer applied loads and satisfy static equilibrium. In the absence of bending mechanisms, the optimized topology will provide compression-only structures for a lower tensile stiffness. In conclusion, the obtained results from the minimum compliance problem are logical and justifiable for asymmetric stiffness.

However, the results for the stress-constrained problem illustrate that the PTO method obtains less optimal tensile topologies, despite using non-linear FEM with a sophisticated multi-scale material model. Namely, this method does provide tensile results for the stress-constrained problem under asymmetric strength (i.e. unequal stress limits in tension and compression) as opposed to the expected compression-dominant structures. Even though the material is in fact stronger in compression, tensile solutions are found. As a result, the tensile topologies do require a larger structural volume in order to prevent exceedance of the material's tensile stress limit, which makes these structures less optimal in terms of volume minimization. To conclude, the results that have been obtained using the developed `MATLAB` codes for the stress-constrained problem clearly demonstrate that sensitivity information is required to steer the topology optimization algorithm towards compression-dominant structures. Therefore, it is recommended to use the Traditional Topology Optimization (TTO) method using numerical sensitivity information for stress-constrained volume minimization under asymmetric strength.

Despite the results from the stress-constrained problem being less optimal, several improvements to topology optimization have been achieved in this graduation thesis. First of all, the implementation of non-linear FEM using a multi-scale material model has enabled incorporation of the asymmetric nature of particle interactions into the continuum model in a computationally affordable manner. As a result, the developed topology optimization algorithms are capable of capturing the anisotropic evolution of a material's microstructure (i.e. induced anisotropy) thus providing a more realistic macroscopic material behaviour. Besides, there is yet another advantage with respect to stress-constrained volume minimization. Instead of using predefined failure envelopes resulting from classical failure criteria that are chosen to match the macroscopic material behaviour from experimental data, the GMA can predict failure points to construct failure envelopes that are an exact match for any given microscopic constitutive relationship. This allows strength-related topology optimization to fully exploit the multi-scale material model, because it is able to both (i) describe the macroscopic material behaviour up to failure in a non-linear framework and to (ii) predict failure points to obtain tailored failure envelopes as a result of natural failure. This means that failure is identified based on the actual material behaviour as opposed to (classical) failure criteria where failure is imposed at predefined stress levels, often using a linear material model. Finally, the model's ability to identify failure based on the stress-path allows for the use of failure criteria that change with respect to the loading path. This makes it possible to perform stiffness-constrained optimization, in which load-path dependency is incorporated into the failure criterion.

7 Recommendations

In the presented work, improvements have been made to topology optimization by means of implementing a multi-scale material model into existing topology optimization algorithms using a non-linear FEM. However, further research can be conducted on many aspects that are covered in this graduation thesis. In this section, these are given by means of recommendations for future research.

Starting with the multi-scale material model, improvements can be made by implementing microscopic constitutive relationships that are calibrated with experimental data. For example, parallel to this work, another graduation research using the GMA performed calibration tests on 3DCP samples. The resulting intergranular force-laws contained, among others, inelastic unloading to describe the material behaviour in a more realistic manner.

Furthermore, the developed non-linear FEM algorithms only consider non-linear material behaviour. However, to adequately describe a structure's response under large displacements (i.e. large strains) inclusion of geometric non-linearities is recommended. Another aspect that requires attention is the computation time of the non-linear FEM codes. While for research purposes the calculation times are acceptable, these are less suitable in view of applicability in practice. Especially with respect to the stress-constrained problem, where the optimized topologies do not differ much from the ones found using a linear material model. To reduce computation times, the developed `MATLAB` codes can be considerably improved in efficiency by loop vectorization, such as vectorizing `for` loops as done in [33].

It has also been illustrated that the non-sensitivity PTO method does provide logical and justifiable results for the minimum compliance problem under asymmetric stiffness while for the stress-constrained problem the method does provide counter-intuitive and less optimal results under asymmetric strength. However, it should be noted that sensitivity information is helpful to accelerate and steer the optimization. Therefore, it is recommended to use TTO as the topology optimization method. In the case of compliance minimization it may provide faster convergence, but for stress-constrained volume minimization it has become clear that sensitivity information is required to steer towards compression-dominant structures. Because the derivative of the objective function with respect to the design variable cannot be obtained analytically for the multi-scale material model, numerical sensitivity information could be obtained from the GMA and used in the TTO method. Initial research has been performed on this specific subject, including a preliminary implementation of numerical sensitivity information, which can be found in Appendix F. Additionally, Appendix F describes the deficiencies of (classical) failure criteria defined as a function of stress tensor. The ability of the GMA to identify failure states that are dependent on the path of loading allows for the use of failure criteria that change with respect to the loading path. This makes it possible to incorporate a material's load-path dependency into the failure criterion by means of stiffness-constrained volume minimization. In this case, the failure criterion can be written as a general function of the material's tangential stiffness tensor: $\mathbb{F}\left(C_{ijkl}\right) = 0$. Appendix F also includes initial research and a preliminary implementation related to this topic.

References

[1] P. Poorsolhjouy and A. Misra, “Grain-size Effects on Mechanical Behavior and Failure of Dense Cohesive Granular Materials,” *KONA Powder and Particle Journal*, no. October, pp. 1–15, 2022.

[2] P. Poorsolhjouy and M. Gonzalez, “Granular micromechanics modeling of beams, plates, and shells,” *Composite Structures*, vol. 278, no. July, p. 114559, 2021. [Online]. Available: https://doi.org/10.1016/j.compstruct.2021.114559

[3] P. Poorsolhjouy, “Operator Calculus for Engineers - Constitutive Equations,” Lecture Notes, Graz University of Technology.

[4] G. K. Rose, “Computational Methods for Nonlinear Systems Analysis With Applications in Mathematics and Engineering,” PhD Dissertation, Old Dominion University, 2017.

[5] I. Bouw, “Stress-Constrained Topology Optimization of Concrete Structures: preliminary study for combining topology optimization and 3D concrete printing,” Master Thesis, Eindhoven University of Technology, 2020.

[6] K. Liu and A. Tovar, “An efficient 3D topology optimization code written in Matlab,” *Structural and Multidisciplinary Optimization*, vol. 50, no. 6, pp. 1175–1196, 2014.

[7] K. J. Willam, “Constitutive Models for Engineering Materials,” *Encyclopedia of Physical Science and Technology*, pp. 603–633, 2003.

[8] G. D. Nguyen, “A thermodynamic approach to constitutive modelling of concrete using damage mechanics and plasticity theory,” *Thesis*, no. June, pp. 1–243, 2005. [Online]. Available: http://www-civil.eng.ox.ac.uk/publications/theses/giang.pdf

[9] H. Jia, A. Misra, P. Poorsolhjouy, and C. Liu, “Optimal structural topology of materials with micro-scale tension-compression asymmetry simulated using granular micromechanics,” *Materials and Design*, vol. 115, pp. 422–432, 2017.

[10] TNO DIANA BV., “DIANA-9.4.4 User’s Manual - Analysis Procedures, section 31.1.5 Incremental Procedures,” 2012. [Online]. Available: https://dianafea.com/manuals/d944/Analys/node396.html#fig:LoDiCtl

[11] A. Becker, *Understanding Non-linear Finite Element analysis Through Illustrative Benchmarks.* Chippenham: NAFEMS, 2001.

[12] N. Vasios, “Nonlinear Analysis of Structures,” Harvard University, Tech. Rep., 2015.

[13] X. Huang and Y. M. Xie, “Topology optimization of nonlinear structures under displacement loading,” *Engineering Structures*, vol. 30, no. 7, pp. 2057–2068, 2008.

[14] P. Poorsolhjouy, "Continuum Modeling Using Granular Micromechanics Approach: Method Development and Applications," PhD Thesis, University of Kansas, 2016.

[15] Y. Luo and Z. Kang, "Topology optimization of continuum structures with Drucker-Prager yield stress constraints," *Computers and Structures*, vol. 90-91, no. 1, pp. 65–75, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.compstruc.2011.10.008

[16] P. Wriggers, *Computational Contact Mechanics.* Chichester: John Wiley & Sons Ltd, 2002.

[17] R. Ibrahim, "Three Dimensional Finite Element Optimization Using the Partial p-Adaptive Method for Stress Analysis of Underground Excavations with Prismatic Cross-sections," Master Thesis, Concordia University, 2014.

[18] G. Dhondt, *The Finite Element Method for Three-Dimensional Thermomechanical Applications.* Munich, Germany: John Wiley & Sons Ltd, 2004.

[19] H. Hofmeyer, M. Schevenels, and S. Boonstra, "The generation of hierarchic structures via robust 3D topology optimisation," *Advanced Engineering Informatics*, vol. 33, pp. 440–455, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.aei.2017.02.002

[20] D. Yang, H. Liu, W. Zhang, and S. Li, "Stress-constrained topology optimization based on maximum stress measures," *Computers and Structures*, vol. 198, pp. 23–39, 2018. [Online]. Available: https://doi.org/10.1016/j.compstruc.2018.01.008

[21] E. Biyikli and A. C. To, "Proportional topology optimization: A new non-sensitivity method for solving stress constrained and minimum compliance problems and its implementation in MATLAB," *PLoS ONE*, vol. 10, no. 12, 2015.

[22] ——, "Proportional topology optimization: A new non-sensitivity method for solving stress constrained and minimum compliance problems and its implementation in MATLAB," *PLoS ONE*, vol. 10, no. 12, pp. 1–23, 2015.

[23] I. van de Werfhorst-Bouw, H. Hofmeyer, S. Boonstra, and R. J. M. Wolfs, "Stress-Constrained Topology Optimization of Concrete Structures: preliminary study for combining topology optimization and 3D concrete printing," *Heron (accepted, in press).*

[24] S. Yang and Y. F. Zhao, "Additive manufacturing-enabled design theory and methodology: a critical review," *International Journal of Advanced Manufacturing Technology*, vol. 80, no. 1-4, pp. 327–342, 2015.

[25] M. Youshi, H. Hofmeyer, and M. Bruurs, "Topology Optimization for Multiple Materials with both Structural and Thermal Objectives to Design 3D Printed

Building Panels," Poster, 24th Engineering Mechanics Symposium, Arnhem, The Netherlands, 2021.

[26] P. Poorsolhjouy and M. Gonzalez, "Granular Micromechanics Modeling of Beams, Plates, and Shells," *Composite Structures*, vol. 278, pp. 1–2, 2021.

[27] A. S. J. Suiker and R. De Borst, "Enhanced continua and discrete lattices for modelling granular assemblies," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 363, no. 1836, pp. 2543–2580, 2005.

[28] D. D. Tjahjanto, S. Turteltaub, A. S. J. Suiker, and S. Van Der Zwaag, "Modelling of the effects of grain orientation on transformation-induced plasticity in multiphase carbon steels," *Modelling and Simulation in Materials Science and Engineering*, vol. 14, no. 4, pp. 617–636, 2006.

[29] H. Jia, Q. Men, A. Misra, and C. Liu, "Topology Optimization of Structure with Asymmetry Cementitious Granular Material," *Advances in Computer Science Research*, vol. 58, pp. 262–266, 2016.

[30] C. L. Liao, T. C. Chan, A. S. J. Suiker, and C. S. Chang, "Pressure-dependent elastic moduli of granular assemblies," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 24, no. 3, pp. 265–279, 2000.

[31] A. S. J. Suiker and C. S. Chang, "Modeling Failure and Deformation of an Assembly of Spheres with Frictional Contacts," *Journal of Engineering Mechanics*, vol. 130, no. 3, pp. 283–293, 2004.

[32] Y. Itin and F. W. Hehl, "The constitutive tensor of linear elasticity: Its decompositions, Cauchy relations, null Lagrangians, and wave propagation," *Journal of Mathematical Physics*, vol. 54, no. 4, 2013.

[33] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, "Efficient topology optimization in MATLAB using 88 lines of code," *Structural and Multidisciplinary Optimization*, vol. 43, no. 1, pp. 1–16, 2011.

[34] S. Chu, L. Gao, M. Xiao, Z. Luo, H. Li, and X. Gui, "A new method based on adaptive volume constraint and stress penalty for stress-constrained topology optimization," *Structural and Multidisciplinary Optimization*, vol. 57, no. 3, pp. 1163–1185, 2018.

[35] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli, "Stress-based topology optimization for continua," *Structural and Multidisciplinary Optimization*, vol. 41, no. 4, pp. 605–620, 2010.

[36] O. Sigmund, "A 99 line topology optimization code written in matlab," *Structural and Multidisciplinary Optimization*, vol. 21, no. 2, pp. 120–127, 2001.

[37] T. Pastore, V. Mercuri, C. Menna, D. Asprone, P. Festa, and F. Auricchio, "Topology optimization of stress-constrained structural elements using risk-factor approach," *Computers and Structures*, vol. 224, 2019.

[38] C. C. Swan and I. Kosaka, "Voigt-Reuss topology optimization for structures with nonlinear material behaviors," *International Journal for Numerical Methods in Engineering*, vol. 40, no. 20, pp. 3785–3814, 1997.

[39] L. Xia and P. Breitkopf, "Recent Advances on Topology Optimization of Multiscale Nonlinear Structures," *Archives of Computational Methods in Engineering*, vol. 24, no. 2, pp. 227–249, 2017.

[40] ——, "Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework," *Computer Methods in Applied Mechanics and Engineering*, vol. 278, pp. 524–542, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.cma.2014.05.022

[41] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method Volume 1: The Basis*, 5th ed. Butterworth-Heinemann, 2000, vol. 1.

[42] S. Klinkel and S. Govindjee, "Using finite strain 3D-material models in beam and shell elements," *Engineering Computations (Swansea, Wales)*, vol. 19, no. 3-4, pp. 254–271, 2002.

[43] M. Jirasek and Z. P. Bazant, *Inelastic Analysis of Structures*, 1st ed. Baffius Lane, Chichester, West Sussex, England: John Wiley & Sons Ltd, 2002.

[44] E. Dvorkin, F. E. Equations, N. Raphson, and T. Broyden-fletcher-goldfarb shanno, "Solution of the Nonlinear Finite Element Equations in Static Analysis Part I," *Analysis*, pp. 871–879, 1985.

[45] N.-h. Kim, "Chapter 2 Nonlinear Finite Element Analysis Procedures Nonlinear Structural Problems What is a linear structural problem ?" University of Florida, Gainesville, Tech. Rep. [Online]. Available: https://mae.ufl.edu/nkim/egm6352/Chap2.pdf

[46] G. P. Nikishkov, "Introduction to the Finite Element Method," Lecture Notes, University of Aizu, 2004. [Online]. Available: niki@u-aizu.ac.jp

[47] E. Biyikli and A. C. To, "Proportional topology optimization: A new non-sensitivity method for solving stress constrained and minimum compliance problems and its implementation in MATLAB," *PLoS ONE*, vol. 10, no. 12, 2015.

[48] ABAQUS, "ABAQUS Online Documentation: Version 6.5-1, section 4.1 Element formulation and integration," 2004. [Online]. Available: https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/-v6.5/books/gss/default.htm?startat=ch04s01.html

[49] S. Kim and G. J. Yun, "Microstructure topology optimization by targeting prescribed nonlinear stress-strain relationships," *International Journal of Plasticity*, vol. 128, no. January, p. 102684, 2020. [Online]. Available: https://doi.org/10.1016/j.ijplas.2020.102684

[50] ABAQUS, “Overview of Compliance.” [Online]. Available: https://abaqus-docs.mit.edu/2017/English/TsoUserMap/tso-c-usr-terms-complianceOvw.htm

[51] M. S. El Naschie, *Stress, Stability, and Chaos in Structural Engineering: An Energy Approach.* McGraw-Hill, 1990.

[52] N. J. Hoff, *Analysis of Structures: Based on the Minimal Principles & the Principle of Virtual Displacement.* John Wiley & Sons Inc, 1956.

[53] F. Jalal and M. Narca, “Module 1 Energy Methods in Structural Analysis,” IIT Kanpur, Kharagpur, Tech. Rep.

[54] Engineering Archives, “Strain Energy Density.” [Online]. Available: http://www.engineeringarchives.com/les_mom_strainenergydensity.html

[55] T. E. Bruns and D. A. Tortorelli, “Topology optimization of geometrically nonlinear structures and compliant mechanisms,” *7th AIAA/USAF/-NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, vol. 190, pp. 1874–1882, 1998.

[56] D. Jung and H. C. Gea, “Compliant mechanism design with non-linear materials using topology optimization,” *Proceedings of the ASME Design Engineering Technical Conference*, vol. 2, pp. 1079–1087, 2002.

[57] Q. Chen, X. Zhang, and B. Zhu, “A 213-line topology optimization code for geometrically nonlinear structures,” *Structural and Multidisciplinary Optimization*, vol. 59, no. 5, pp. 1863–1879, 2019.

[58] A. Verbart, N. Dijk, L. Tin, M. Langelaar, and F. van Keulen, “Effect of design parameterization and relaxation on model responses in topology optimization with stress constraints,” *9th World Congress on Structural and Multidisciplinary Optimization*, no. July, pp. 1–14, 2011.

[59] X. S. Zhang, G. H. Paulino, and A. S. Ramos, “Multi-material topology optimization with multiple volume constraints: a general approach applied to ground structures with material nonlinearity,” *Structural and Multidisciplinary Optimization*, vol. 57, no. 1, pp. 161–182, 2018.

[60] M. P. Bendsøe and O. Sigmund, *Typology Optimization: Theory, Methods, and Applications*, second edi ed. Berlin Heidelberg: Springer-Verlag, 2004. [Online]. Available: https://link.springer.com/book/10.1007/978-3-662-05086-6#toc

[61] T. Zhao, A. S. Ramos, and G. H. Paulino, “Material nonlinear topology optimization considering the von Mises criterion through an asymptotic approach: Max strain energy and max load factor formulations,” *International Journal for Numerical Methods in Engineering*, vol. 118, no. 13, pp. 804–828, 2019.

[62] Nederlands Normalisatie-instituut, “NEN-EN 1992-1-1 - Eurocode 2: Design of concrete structures - Part 1-1: General rules and rules for buildings,” Delft, 2006.

[63] K.-J. B. W. Wunderlich, E. Stein, *Nonlinear Finite Element Analysis in Structural Mechanics*, 1st ed., K.-J. B. W. Wunderlich, E. Stein, Ed. Berlin: Springer-Verlag, 1981.

[64] E. Riks, “An incremental approach to the solution of snapping and buckling problems,” *International Journal of Solids and Structures*, vol. 15, no. 7, pp. 529–551, 1979. [Online]. Available: http://dx.doi.org/10.1016/0020-7683(79)90081-7

[65] B. A. Memon and X. Z. Su, “Arc-length technique for nonlinear finite element analysis,” *Journal of Zhejiang University: Science*, vol. 5, no. 5, pp. 618–628, 2004.

[66] R. Radovitzky, “Module 10: Introduction to Energy Methods,” Ph.D. dissertation, 2013.

[67] Engineering Archives, “Strain Energy.” [Online]. Available: http://www.engineeringarchives.com/les_mom_strainenergy.html

[68] K. Svanberg, “The method of moving asymptotes—a new method for structural optimization,” *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 359–373, 1987.

[69] A. S. J. Suiker and S. Turteltaub, “Computational modelling of plasticity induced by martensitic phase transformations,” *International Journal for Numerical Methods in Engineering*, vol. 63, no. 12, pp. 1655–1693, 2005.

[70] D. G. Cacuci, “Sensitivity theory for nonlinear systems. I. Nonlinear functional analysis approach,” *Journal of Mathematical Physics*, vol. 22, no. 12, pp. 2794–2802, 1980.

[71] J. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

[72] S. Cho and H. S. Jung, “Design sensitivity analysis and topology optimization of displacement-loaded non-linear structures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 22-23, pp. 2539–2553, 2003.

Appendix A Alternative Solution Procedures

In addition to the load control implementation of the Newton-Raphson iterative method, this appendix describes several alternative solution procedures that can be used in non-linear Finite Element Analysis (FEA). Starting with displacement control, including elaborated examples of both variations for comparison. Subsequently, the arc length method will be shortly discussed, followed by a comparison of the two implementations of the Newton-Raphson iterative method for validation purposes.

A.1 Displacement Control

Displacement control is the most often used method to avoid singularity at the critical point by interchanging the dependent and independent variables [63]. In this method, instead of postulating a desired load value certain displacement components are selected as controlling parameters. These displacement components are prescribed and the corresponding load level is unknown, but can be calculated as a reaction on the applied displacements [45]. Since the displacement components are directly prescribed at certain points (by treating them as supports), this technique may also be referred to as *direct displacement control* [43].

Similar to load control, the objective is to find all the displacement components in $\mathbf{u}$ such that Equation (2.33) is satisfied. However, in the case of displacement control the external force vector $\mathbf{F}$ is not increased directly [10]. The prescribed displacements must be incorporated in the external force vector to get a proper first estimate of the displacements. This effective force vector can be computed by rewriting Equation (2.40) and partitioning the displacement increment vector in two groups. The first one consists of unknown displacements at the nodes that are left 'free', referred to as *unconstrained* degrees of freedom (indicated by superscript u), and the second one is composed of prescribed displacements at nodes that are controlled, referred to as *constrained* degrees of freedom (indicated by superscript c). Accordingly, the displacement vector as well as the internal and external force vectors are partitioned as $\{\mathbf{u}^u, \mathbf{u}^c\}^T$, $\{\mathbf{f}^u, \mathbf{f}^c\}^T$ and $\{\mathbf{F}^u, \mathbf{F}^c\}^T$, respectively. The external forces related to the unconstrained degrees of freedom $\mathbf{F}^u$ are prescribed and are assumed to be equal to zero (as in load control). In principle, the only externally applied forces acting on the structure are the reactions at the constrained degrees of freedom $\mathbf{F}^c$. Therefore, Equation (2.33) can be partitioned as:

$$\mathbf{f}^u(\mathbf{u}^u, \mathbf{u}^c) = \mathbf{F}^u = \mathbf{0}\,, \tag{A.1}$$

$$\mathbf{f}^c(\mathbf{u}^u, \mathbf{u}^c) = \mathbf{F}^c\,. \tag{A.2}$$

Contrary to load control, in displacement control the vector containing the prescribed displacements $\mathbf{u}^c$ is gradually increased from zero to a desired, predefined displacement value. The unknown displacements $\mathbf{u}^u$ can be calculated by solving Equation (A.1) for a prescribed $\mathbf{u}^c$. Afterwards, the external forces $\mathbf{F}^c$ can be obtained as a reaction on the prescribed displacements by simply evaluating the left-hand side of Equation (A.2). Continuing according to the procedure as presented in [43], linearization of Equation (A.1) yields:

$$\mathbf{f}^u + \mathbf{K}^{uu}\Delta\mathbf{u}^u + \mathbf{K}^{uc}\Delta\mathbf{u}^c = \mathbf{0}\,, \tag{A.3}$$

where $\mathbf{K}^{uu} \equiv \partial\mathbf{f}^u/\partial\mathbf{u}^u$ and $\mathbf{K}^{uc} \equiv \partial\mathbf{f}^u/\partial\mathbf{u}^c$ are parts of the global tangent stiffness matrix

$$\mathbf{K} \equiv \frac{\partial\mathbf{f}}{\partial\mathbf{u}} = \begin{bmatrix} \frac{\partial\mathbf{f}^u}{\partial\mathbf{u}^u} & \frac{\partial\mathbf{f}^u}{\partial\mathbf{u}^c} \\ \frac{\partial\mathbf{f}^c}{\partial\mathbf{u}^u} & \frac{\partial\mathbf{f}^c}{\partial\mathbf{u}^c} \end{bmatrix} = \begin{bmatrix} \mathbf{K}^{uu} & \mathbf{K}^{uc} \\ \mathbf{K}^{cu} & \mathbf{K}^{cc} \end{bmatrix}\,, \tag{A.4}$$

in the following partitioned linearized set of equations [10]

$$\begin{bmatrix} \mathbf{K}^{uu} & \mathbf{K}^{uc} \\ \mathbf{K}^{cu} & \mathbf{K}^{cc} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u}^u \\ \Delta\mathbf{u}^c \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}^u \\ \mathbf{R}^c \end{Bmatrix}\,, \tag{A.5}$$

where $\mathbf{R}^u = \mathbf{F}^u - \mathbf{f}^u = -\mathbf{f}^u$ is the residual related to the unconstrained degrees of freedom and $\mathbf{R}^c = \mathbf{F}^c - \mathbf{f}^c$ the residual related to the constrained degrees of freedom. The prescribed displacement increment $\Delta\mathbf{u}^c$ is known beforehand, such that Equation (A.3) can be rewritten as

$$\Delta\mathbf{u}^u = (\mathbf{K}^{uu})^{-1}\{-\mathbf{f}^u - \mathbf{K}^{uc}\Delta\mathbf{u}^c\}\,, \tag{A.6}$$

from which the unknown displacement increments $\Delta\mathbf{u}^u$ can be computed. Note that the exact same follows from the first row in Equation (A.5), such that Equation (A.6) can be expressed in terms of the unconstrained residual $\mathbf{R}^u$ as follows:

$$\Delta\mathbf{u}^u = (\mathbf{K}^{uu})^{-1}\{-\mathbf{K}^{uc}\Delta\mathbf{u}^c + \mathbf{R}^u\}\,. \tag{A.7}$$

Comparing Equations (2.40) and (A.7) indicates that $-\mathbf{K}^{uc}\Delta\mathbf{u}^c$ can be regarded as the *effective force vector*, equivalent with the prescribed displacements [10]. Note that, after the first iteration, this correcting term is equal to zero such that the effective force vector vanishes from Equation (A.7). This term is only present in the first iteration.

Figure A.1 below illustrates the two variants of the Newton-Raphson iterative method: (a) load control and (b) displacement control. In Figure A.1(a), the load-displacement curve can only be traced up to ${}^{t_2}\mathbf{f}_{ext}$. After this load increment, ${}^{t_3}\mathbf{f}_{ext}$ exceeds the maximum load that can be resisted by the structure and the equilibrium equations have no solution. Therefore, load control fails around the limit point, where the tangent of the curve becomes horizontal. Figure A.1(b) shows that displacement control can overcome this point.

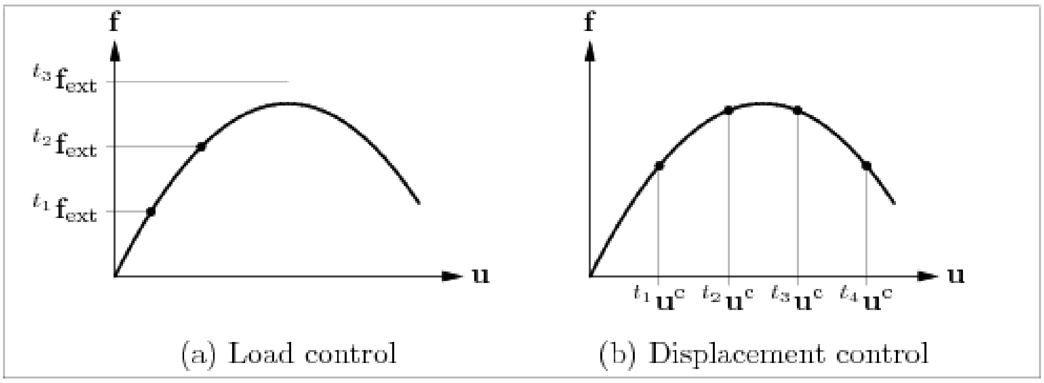

f
${}^{t_3}\mathbf{f}_{\text{ext}}$
${}^{t_2}\mathbf{f}_{\text{ext}}$
${}^{t_1}\mathbf{f}_{\text{ext}}$
u
(a) Load control
f
u
${}^{t_1}\mathbf{u}^c$ ${}^{t_2}\mathbf{u}^c$ ${}^{t_3}\mathbf{u}^c$ ${}^{t_4}\mathbf{u}^c$
(b) Displacement control

Figure A.1: Load and displacement control [10]

As stressed in [43], each iteration in the Newton-Raphson iterative method involves three computationally expensive operations: (i) evaluation of the internal forces, (ii) evaluation of the global tangens stiffness matrix and (iii) solution of a system of linear equations. For large-scale models, evaluating the global tangent stiffness matrix and successively solving a set of linear equations may be excessively expensive. Variations of the Newton-Raphson iterative method exist to save on cost related to computation and inversion of the tangent stiffness matrix [4]. One of these variations is a *modified Newton-Raphson* method which evaluates the stiffness matrix only at the beginning of each increment and keeps it constant in the subsequent iterations. Therefore, step (ii) can be completely skipped. However, this slows down the rate of convergence thus increasing the number of solver iterations required for convergence. Figure A.2 illustrates the two discussed variations of the Newton-Raphson iterative method.

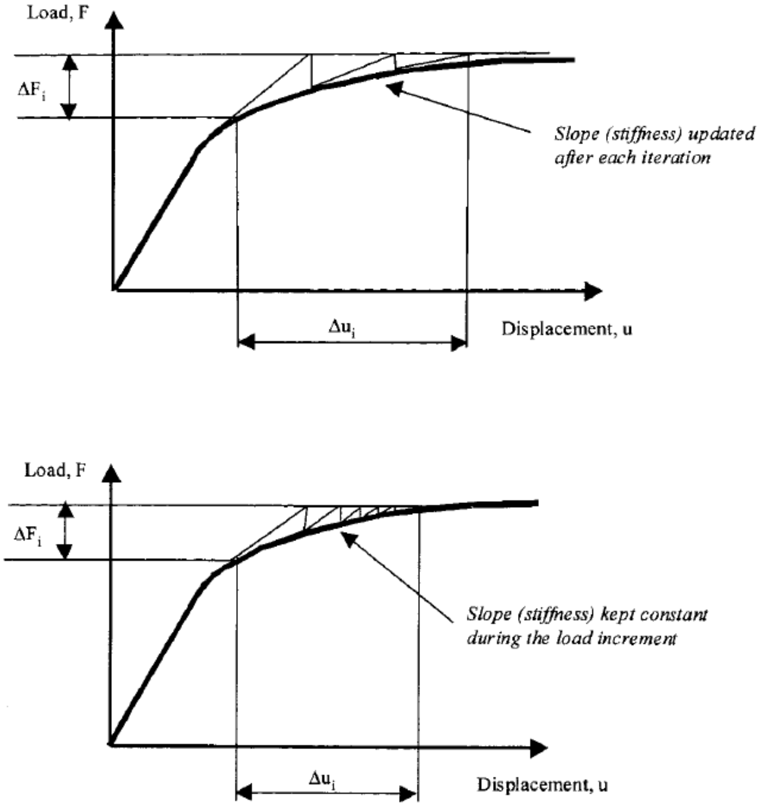

Load, F
ΔF_i
Slope (stiffness) updated after each iteration
Δu_i
Displacement, u
Load, F
ΔF_i
Slope (stiffness) kept constant during the load increment
Δu_i
Displacement, u

Figure A.2: Full & modified Newton-Raphson iterative method [11]

Identical to the implementation for load control, the stiffness matrix is updated for each iteration in the displacement control implementation of the Newton-Raphson iterative method. In this way, the solution procedure exhibits second order or quadratic convergence near the solution [44] which yields guaranteed convergence within few iterations [12] and thus faster convergence [45]. The stepwise procedure for a parameterized, displacement control Newton-Raphson iterative method is provided in Algorithm 5.

Algorithm 5 Newton-Raphson iterative method for displacement control

```
1. Initialization
    (a) Define tolerance
    (b) Define displacement increments ($\Delta\mathbf{u}^c$)
2. Next displacement increment $t \geq 1$
    (a) Read $\Delta\mathbf{u}^c_t$ from displacement incrementation
    (b) Provide an estimate or initial guess: $\mathbf{u}^u_i = \mathbf{u}^u_{t-1}$
3. Next iteration $i \geq 1$
    (a) Find $\mathbf{K}_i$ or the matrix of first-order partial derivatives with respect to state variables in
        vector $\mathbf{u}_i = \{\mathbf{u}^u, \mathbf{u}^c\}^T$ as given by Equation (A.4)
    (b) Find system vector $\mathbf{f}(\mathbf{u}_i) = \{\mathbf{f}^u, \mathbf{f}^c\}^T$        ▷ See Section 2.2.2
    (c) Find the residual related to the unconstrained degrees of freedom or difference between $\mathbf{F}^u_i$
        and $\mathbf{f}^u(\mathbf{u}_i)$ where $\mathbf{R}^u_i = -\mathbf{f}^u(\mathbf{u}_i)$
    (d) Find $\Delta\mathbf{u}^u_i$ via Equation (A.7)
    (e) Update $\mathbf{u}_i$ by adding $\Delta\mathbf{u}_i$ to the current value of $\mathbf{u}_i$ where $\mathbf{u}_i = \mathbf{u}_i + \Delta\mathbf{u}_i$
    (f) Check if $\Delta\mathbf{u}_i$ is small with respect to $\mathbf{u}_i$
        i. if $\|\Delta\mathbf{u}_i\|/\|\mathbf{u}_i\| >$ tolerance then $\implies$ go to next iteration 3
        ii. else if $\|\Delta\mathbf{u}_i\|/\|\mathbf{u}_i\| \leq$ tolerance then store $\mathbf{u}_i \implies$ go to next load increment 2
```

To illustrate the differences between the implementations of load and displacement control, elaborated examples of both the load and displacement control variations of the Newton-Raphson iterative method are reviewed next. These include simple one-dimensional beam element problems and a two-dimensional plane stress problem. Each elaboration starts with displacement control, followed by load control for comparison.

A.1.1 Load & Displacement Control Examples

Consider the one-dimensional problem in Figure A.3 of a beam with an arbitrary number of elements (N_e). Each element has a length equal to L such that the total length of the considered beam equals $N_e \times L$.

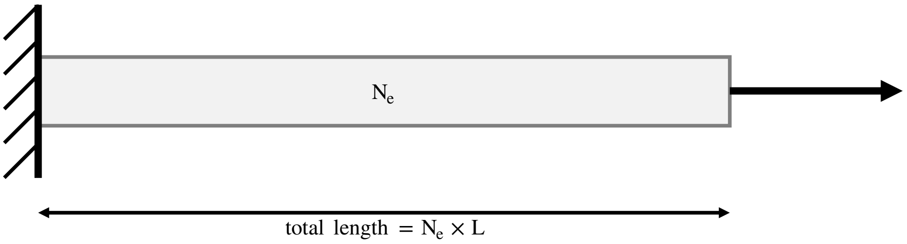


Figure A.3: One-dimensional beam having an arbitrary number of elements

Starting simple by considering a one-dimensional problem for a beam composed of two elements (see Figure A.4) which is solved using displacement control with the following boundary conditions:

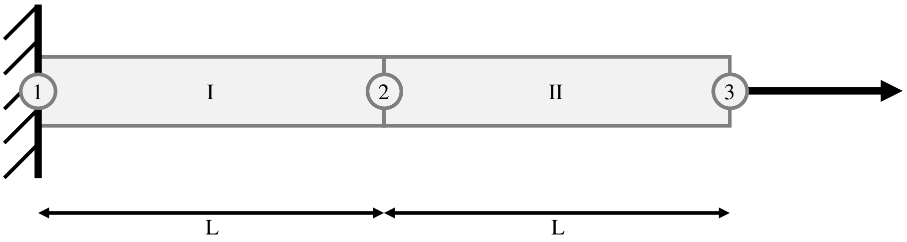


Figure A.4: One-dimensional beam having two elements

$$\mathbf{u} = \left\{ \begin{array}{c} u_1 = 0 \\ u_2 = ? \\ u_3 = \Delta \end{array} \right\} . \tag{A.8}$$

Each element has two Degrees Of Freedom (DOFs). Therefore, the displacement vectors of elements I and II become, respectively

$$\mathbf{u}^I = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} , \qquad \mathbf{u}^{II} = \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} . \tag{A.9}$$

Given that $\boldsymbol{\varepsilon} = \mathbf{Bu}$ with $\mathbf{B} = \begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \end{bmatrix}$:

$$\boldsymbol{\varepsilon}^I = \begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} , \qquad \boldsymbol{\varepsilon}^{II} = \begin{bmatrix} -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} . \tag{A.10}$$

The elemental stresses are a function of their related strains, i.e. $\boldsymbol{\sigma}^i = \boldsymbol{\sigma}^i \left(\varepsilon^i\right)$. Therefore, the stresses of elements I and II become, respectively

$$\boldsymbol{\sigma}^I = \boldsymbol{\sigma}^I \left(\varepsilon^I\right) , \qquad \boldsymbol{\sigma}^{II} = \boldsymbol{\sigma}^{II} \left(\varepsilon^{II}\right) . \tag{A.11}$$

Recall that the internal force vector can be obtained by solving the integral given in Equation (2.50): $\mathbf{f} = \int_V \mathbf{B}^T \boldsymbol{\sigma} \mathrm{d}V$. Please note that for the one-dimensional problem, only the length L of the elements is subjected to change. As a result, the cross-sectional area A remains constant such that it can be taken out of the integral. Insertion of Equation (A.11) into the integral yields the following internal force vectors of elements I and II, respectively

$$\mathbf{f}^I = A \begin{Bmatrix} -\boldsymbol{\sigma}^I \\ \boldsymbol{\sigma}^I \end{Bmatrix} , \qquad \mathbf{f}^{II} = A \begin{Bmatrix} -\boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} \end{Bmatrix} . \tag{A.12}$$

From Equation (A.12) the *global* internal force vector can be assembled, which is given below next to the problem's external force vector:

$$\mathbf{f} = A \begin{Bmatrix} -\boldsymbol{\sigma}^I \\ \boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} \end{Bmatrix}, \qquad \mathbf{F} = \begin{Bmatrix} F_1 = ? \\ F_2 = 0 \\ F_3 = ? \end{Bmatrix}. \tag{A.13}$$

Now that the *global* internal and external force vectors have been determined, the residual can be derived from Equation (A.13) as follows:

$$\mathbf{R} = \mathbf{f} - \mathbf{F} = A \begin{Bmatrix} -\boldsymbol{\sigma}^I - F_1/A \\ \boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} - F_3/A \end{Bmatrix}. \tag{A.14}$$

The unknown displacement $\mathbf{u}^u = u_2$ can be calculated by means of the displacement control implementation of the Newton-Raphson iterative method as described previously. Please note that since $F_2 = 0$, the residual related to the unconstrained degrees of freedom simply reduces to $\mathbf{R}^u = \mathbf{f}^u = \boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}$. To this end, the linearized global tangent stiffness matrix considered for the successive iterations can be derived by taking the derivative of $\mathbf{R}^u$ with respect to $\mathbf{u}^u$ as follows:

$$\mathbf{R}' = \frac{\partial \mathbf{R}^u}{\partial \mathbf{u}^u} = \frac{\mathrm{d}}{\mathrm{d}u_2} A \left(\boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}\right) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\varepsilon}} A \left(\boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}\right) \frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}u_2}. \tag{A.15}$$

By working out Equation (A.15), one obtains

$$\mathbf{R}' = A \left[\frac{\mathrm{d}\boldsymbol{\sigma}^I}{\mathrm{d}\boldsymbol{\varepsilon}^I} \underbrace{\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_2}}_{\frac{1}{L}} - \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}} \underbrace{\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2}}_{-\frac{1}{L}} \right] = \frac{A}{L} \left(\frac{\mathrm{d}\boldsymbol{\sigma}^I}{\mathrm{d}\boldsymbol{\varepsilon}^I} + \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}} \right) = \frac{A}{L} \left[k\left(\boldsymbol{\varepsilon}^I\right) + k\left(\boldsymbol{\varepsilon}^{II}\right) \right], \tag{A.16}$$

where $k = \mathrm{d}\boldsymbol{\sigma}^i / \mathrm{d}\boldsymbol{\varepsilon}^i$ is considered to be a stiffness measure as a function of the elemental strains. In this case, the Newton-Raphson iterative method reads as follows:

$$\mathbf{R}(u_2) + \mathbf{R}'(u_2)\mathrm{d}u_2 = 0, \tag{A.17}$$

where

$$\mathbf{R}(u_2) = A\left[\boldsymbol{\sigma}^I(u_2) - \boldsymbol{\sigma}^{II}(u_2)\right] = A\left[\left(\boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}\right)\right]\big|_{u_2}, \tag{A.18}$$

$$\mathbf{R}'(u_2) = \frac{A}{L}\left[k\left(\boldsymbol{\varepsilon}^I\right) + k\left(\boldsymbol{\varepsilon}^{II}\right)\right]\Bigg|_{u_2}. \tag{A.19}$$

This procedure is repeated until convergence, as discussed in Section 2.2.

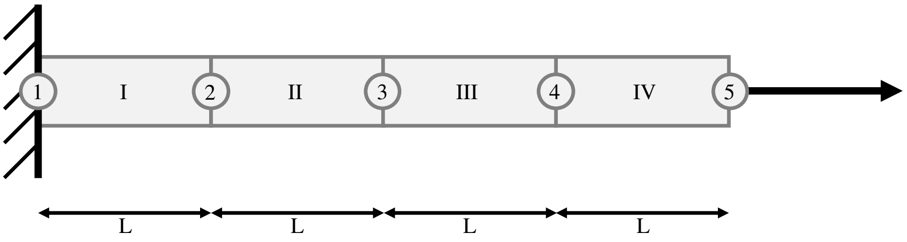


Figure A.5: One-dimensional beam having three elements

Now consider the one-dimensional problem for a beam composed of four elements (see Figure A.5) with the following boundary conditions:

$$\mathbf{u} = \left\{ \begin{array}{l} u_1 = 0 \\ u_2 = ? \\ u_3 = ? \\ u_4 = ? \\ u_5 = \Delta \end{array} \right\} . \tag{A.20}$$

Again, each element has two DOFs. The elemental displacement vectors are provided with the boundary conditions substituted as:

$$\begin{aligned} \mathbf{u}^{I} &= \begin{Bmatrix} 0 \\ u_2 \end{Bmatrix} , & \mathbf{u}^{III} &= \begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix} , \\ \mathbf{u}^{II} &= \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} , & \mathbf{u}^{IV} &= \begin{Bmatrix} u_4 \\ \Delta \end{Bmatrix} . \end{aligned} \tag{A.21}$$

Similar as before, given that $\boldsymbol{\varepsilon} = \mathbf{Bu}$ with $\mathbf{B} = \left[-\frac{1}{L} \quad \frac{1}{L}\right]$ provided the boundary conditions are substituted:

$$\begin{aligned} \boldsymbol{\varepsilon}^{I} &= \frac{u_2}{L} , & \boldsymbol{\varepsilon}^{III} &= \frac{u_3 - u_4}{L} , \\ \boldsymbol{\varepsilon}^{II} &= \frac{u_4 - u_3}{L} , & \boldsymbol{\varepsilon}^{IV} &= \frac{\Delta - u_4}{L} . \end{aligned} \tag{A.22}$$

Similarly, the elemental stresses are a function of their related strains, i.e. $\boldsymbol{\sigma}^{i} = \boldsymbol{\sigma}^{i}\left(\boldsymbol{\varepsilon}^{i}\right)$. Therefore, the elemental stresses become:

$$\begin{aligned} \boldsymbol{\sigma}^{I} &= \boldsymbol{\sigma}^{I}\left(\boldsymbol{\varepsilon}^{I}\right) , & \boldsymbol{\sigma}^{III} &= \boldsymbol{\sigma}^{III}\left(\boldsymbol{\varepsilon}^{III}\right) , \\ \boldsymbol{\sigma}^{II} &= \boldsymbol{\sigma}^{II}\left(\boldsymbol{\varepsilon}^{II}\right) , & \boldsymbol{\sigma}^{IV} &= \boldsymbol{\sigma}^{IV}\left(\boldsymbol{\varepsilon}^{IV}\right) . \end{aligned} \tag{A.23}$$

The internal force vectors are obtained through Equation (2.50) and insertion of Equation (A.23) into the integral yields the following internal force vectors:

$$\mathbf{f}^{I} = A\begin{Bmatrix} -\boldsymbol{\sigma}^{I} \\ \boldsymbol{\sigma}^{I} \end{Bmatrix}, \qquad \mathbf{f}^{III} = A\begin{Bmatrix} -\boldsymbol{\sigma}^{III}, \\ \boldsymbol{\sigma}^{III} \end{Bmatrix}, \tag{A.24}$$

$$\mathbf{f}^{II} = A\begin{Bmatrix} ,-\boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} \end{Bmatrix}, \qquad \mathbf{f}^{IV} = A\begin{Bmatrix} -\boldsymbol{\sigma}^{IV} \\ \boldsymbol{\sigma}^{IV} \end{Bmatrix}.$$

From Equation (A.24) the *global* internal force vector can be assembled, which is given below next to the problem's external force vector once again:

$$\mathbf{f} = A\begin{Bmatrix} -\boldsymbol{\sigma}^{I} \\ \boldsymbol{\sigma}^{I} - \boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} - \boldsymbol{\sigma}^{III} \\ \boldsymbol{\sigma}^{III} - \boldsymbol{\sigma}^{IV} \\ \boldsymbol{\sigma}^{IV} \end{Bmatrix}, \qquad \mathbf{F} = \begin{Bmatrix} F_1 = ? \\ F_2 = 0 \\ F_3 = 0 \\ F_4 = 0 \\ F_5 = ? \end{Bmatrix}. \tag{A.25}$$

Using the *global* internal and external force vectors, the residual can be derived from Equation (A.25) as follows:

$$\mathbf{R} = \mathbf{f} - \mathbf{F} = A\begin{Bmatrix} -\boldsymbol{\sigma}^{I} - F_1/A \\ \boldsymbol{\sigma}^{I} - \boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} - \boldsymbol{\sigma}^{III} \\ \boldsymbol{\sigma}^{III} - \boldsymbol{\sigma}^{IV} \\ \boldsymbol{\sigma}^{IV} - F_5/A \end{Bmatrix}. \tag{A.26}$$

Now the unknown displacements $\mathbf{u}^{u} = \{u_2, u_3, u_4\}^{T}$ can be calculated. Please note again that since $\mathbf{F}^{u} = \{F_2, F_3, F_4\}^{T} = \mathbf{0}$, $\mathbf{R}^{u}$ simply reduces to the internal force vector related to the unconstrained DOFs ($\mathbf{f}^{u}$). Similarly, the stiffness matrix considered for the successive iterations is derived by working out $\partial\mathbf{R}^{u}/\partial\mathbf{u}^{u}$ which leads to the following result:

$$\begin{aligned} \mathbf{R}' &= \frac{\partial \mathbf{R}^{u}}{\partial \mathbf{u}^{u}} \\ &= A\begin{bmatrix} \frac{\mathrm{d}\boldsymbol{\sigma}^{I}}{\mathrm{d}\boldsymbol{\varepsilon}^{I}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{I}}{\mathrm{d}u_2} - \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2} & \frac{\mathrm{d}\boldsymbol{\sigma}^{I}}{\mathrm{d}\boldsymbol{\varepsilon}^{I}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{I}}{\mathrm{d}u_3} - \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_3} & \frac{\mathrm{d}\boldsymbol{\sigma}^{I}}{\mathrm{d}\boldsymbol{\varepsilon}^{I}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{I}}{\mathrm{d}u_4} - \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_4} \\ \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2} - \frac{\mathrm{d}\boldsymbol{\sigma}^{III}}{\mathrm{d}\boldsymbol{\varepsilon}^{III}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_2} & \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_3} - \frac{\mathrm{d}\boldsymbol{\sigma}^{III}}{\mathrm{d}\boldsymbol{\varepsilon}^{III}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_3} & \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_4} - \frac{\mathrm{d}\boldsymbol{\sigma}^{III}}{\mathrm{d}\boldsymbol{\varepsilon}^{III}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_4} \\ \frac{\mathrm{d}\boldsymbol{\sigma}^{III}}{\mathrm{d}\boldsymbol{\varepsilon}^{III}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_2} - \frac{\mathrm{d}\boldsymbol{\sigma}^{IV}}{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}{\mathrm{d}u_2} & \frac{\mathrm{d}\boldsymbol{\sigma}^{III}}{\mathrm{d}\boldsymbol{\varepsilon}^{III}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_3} - \frac{\mathrm{d}\boldsymbol{\sigma}^{IV}}{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}{\mathrm{d}u_3} & \frac{\mathrm{d}\boldsymbol{\sigma}^{III}}{\mathrm{d}\boldsymbol{\varepsilon}^{III}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_4} - \frac{\mathrm{d}\boldsymbol{\sigma}^{IV}}{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}{\mathrm{d}u_4} \end{bmatrix}. \end{aligned} \tag{A.27}$$

Working out the individual terms in Equation (A.27) yields:

$$\begin{aligned}
&\frac{\mathrm{d}\boldsymbol{\sigma}^I}{\mathrm{d}\boldsymbol{\varepsilon}^I} = k\left(\boldsymbol{\varepsilon}^I\right)\,, && \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}} = k\left(\boldsymbol{\varepsilon}^{II}\right)\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_3} = 0\,, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_2} = \frac{1}{L}\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2} = -\frac{1}{L}\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_3} = \frac{1}{L}\,, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_4} = 0\,, && \frac{\mathrm{d}\boldsymbol{\sigma}^{III}}{\mathrm{d}\boldsymbol{\varepsilon}^{III}} = k\left(\boldsymbol{\varepsilon}^{III}\right)\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_3} = -\frac{1}{L}\,, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_4} = 0\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_2} = 0\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_4} = 0\,, \\
&\frac{\mathrm{d}\boldsymbol{\sigma}^{IV}}{\mathrm{d}\boldsymbol{\varepsilon}^{IV}} = k\left(\boldsymbol{\varepsilon}^{IV}\right)\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}{\mathrm{d}u_3} = 0\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}{\mathrm{d}u_4} = -\frac{1}{L}\,, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{IV}}{\mathrm{d}u_2} = 0\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{III}}{\mathrm{d}u_4} = \frac{1}{L}\,.
\end{aligned} \tag{A.28}$$

By substituting Equation (A.28) into Equation (A.27), one obtains

$$\mathbf{R}' = \frac{A}{L}\begin{bmatrix} k\left(\boldsymbol{\varepsilon}^I\right) + k\left(\boldsymbol{\varepsilon}^{II}\right) & -k\left(\boldsymbol{\varepsilon}^{II}\right) & 0 \\ -k\left(\boldsymbol{\varepsilon}^{II}\right) & k\left(\boldsymbol{\varepsilon}^{II}\right) + k\left(\boldsymbol{\varepsilon}^{III}\right) & -k\left(\boldsymbol{\varepsilon}^{III}\right) \\ 0 & -k\left(\boldsymbol{\varepsilon}^{III}\right) & k\left(\boldsymbol{\varepsilon}^{III}\right) + k\left(\boldsymbol{\varepsilon}^{IV}\right) \end{bmatrix}\,. \tag{A.29}$$

Now the Newton-Raphson iterative method reads as follows for the unknown displacements $\mathbf{u}^u = \{u_2, u_3, u_4\}^T$:

$$\mathbf{R}(\mathbf{u}^u) + \mathbf{R}'(\mathbf{u}^u)\mathrm{d}\mathbf{u}^u = 0\,, \tag{A.30}$$

where

$$\mathbf{R}^u(\mathbf{u}^u) = A\left.\begin{Bmatrix} \boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} - \boldsymbol{\sigma}^{III} \\ \boldsymbol{\sigma}^{III} - \boldsymbol{\sigma}^{IV} \end{Bmatrix}\right|_{\mathbf{u}^u}\,, \tag{A.31}$$

$$\mathbf{R}'(\mathbf{u}^u) = \frac{A}{L}\left.\begin{bmatrix} k\left(\boldsymbol{\varepsilon}^I\right) + k\left(\boldsymbol{\varepsilon}^{II}\right) & -k\left(\boldsymbol{\varepsilon}^{II}\right) & 0 \\ -k\left(\boldsymbol{\varepsilon}^{II}\right) & k\left(\boldsymbol{\varepsilon}^{II}\right) + k\left(\boldsymbol{\varepsilon}^{III}\right) & -k\left(\boldsymbol{\varepsilon}^{III}\right) \\ 0 & -k\left(\boldsymbol{\varepsilon}^{III}\right) & k\left(\boldsymbol{\varepsilon}^{III}\right) + k\left(\boldsymbol{\varepsilon}^{IV}\right) \end{bmatrix}\right|_{\mathbf{u}^u}\,. \tag{A.32}$$

Similar as before, this procedure is repeated until convergence. However, by using more elements the structure of the linearized global tangential stiffness matrix becomes visible. It can be clearly seen that Equation (A.29) is composed of several element stiffness matrices of the form

$$\mathbf{K}_e^i = \frac{A}{L}\begin{bmatrix} k\left(\boldsymbol{\varepsilon}^i\right) & -k\left(\boldsymbol{\varepsilon}^i\right) \\ -k\left(\boldsymbol{\varepsilon}^i\right) & k\left(\boldsymbol{\varepsilon}^i\right) \end{bmatrix}\,. \tag{A.33}$$

As a result, the entire linearized global tangential stiffness matrix can be easily constructed from the element stiffness matrices which is beneficial for computational implementations. For comparison, the one-dimensional problem for two beam elements (see Figure A.4) is also solved using load control with the following boundary conditions:

$$\mathbf{F} = \left\{ \begin{array}{c} F_1 = ? \\ F_2 = 0 \\ F_3 = F \end{array} \right\}, \qquad \mathbf{u} = \left\{ \begin{array}{c} u_1 = 0 \\ u_2 = ? \\ u_3 = ? \end{array} \right\}. \tag{A.34}$$

The displacement vectors, elemental strain formulations, stresses as a function of their related strains and internal force vectors are the same as in displacement control. However, the external force vector is different here such that the residual becomes

$$\mathbf{R} = \mathbf{f} - \mathbf{F} = A \left\{ \begin{array}{c} -\boldsymbol{\sigma}^I - F_1/A \\ \boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} - F/A \end{array} \right\}. \tag{A.35}$$

The unknown displacements u_2 and u_3 can be calculated by means of the load control implementation of the Newton-Raphson iterative method as described in Section 2.2.1. The support condition ($u_1 = 0$) is implemented by eliminating this DOF from the set of equations. This is similar to the Finite Element Method (FEM) implementation by Sigmund [36] in which a distinction is made between fixed and free DOFs. In load control, the unknown displacements are related to the free DOFs and the support conditions are related to the fixed DOFs. Identical to displacement control, the linearized global tangent stiffness matrix can be derived by taking the derivative of the reduced residual (without the fixed DOFs) with respect to the free DOFs as follows:

$$\begin{aligned} \mathbf{R}' = \frac{\partial \mathbf{R}}{\partial \mathbf{u}} &= A \begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}u_2}\left(\boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}\right) & \frac{\mathrm{d}}{\mathrm{d}u_3}\left(\boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}\right) \\ \frac{\mathrm{d}}{\mathrm{d}u_2}\left(\boldsymbol{\sigma}^{II} - F/A\right) & \frac{\mathrm{d}}{\mathrm{d}u_3}\left(\boldsymbol{\sigma}^{II} - F/A\right) \end{bmatrix} \\ &= A \begin{bmatrix} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\varepsilon}}\left(\boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}\right)\frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}u_2} & \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\varepsilon}}\left(\boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II}\right)\frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}u_3} \\ \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\varepsilon}}\left(\boldsymbol{\sigma}^{II} - F/A\right)\frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}u_2} & \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\varepsilon}}\left(\boldsymbol{\sigma}^{II} - F/A\right)\frac{\mathrm{d}\boldsymbol{\varepsilon}}{\mathrm{d}u_3} \end{bmatrix}. \end{aligned} \tag{A.36}$$

By working out Equation (A.36), one obtains:

$$\mathbf{R}' = \frac{\partial \mathbf{R}}{\partial \mathbf{u}} = A \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{\sigma}^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_2} - \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2} & \frac{\mathrm{d}\boldsymbol{\sigma}^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_3} - \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_3} \\ \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2} - \frac{1}{A}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2} & \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_3} - \frac{1}{A}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_3} \end{bmatrix}. \tag{A.37}$$

Working out the individual terms in Equation (A.37) yields:

$$\begin{aligned}
&\frac{\mathrm{d}\boldsymbol{\sigma}^I}{\mathrm{d}\boldsymbol{\varepsilon}^I} = k\left(\boldsymbol{\varepsilon}^I\right)\,, && \frac{\mathrm{d}\boldsymbol{\sigma}^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}} = k\left(\boldsymbol{\varepsilon}^{II}\right)\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_3} = 0\,,\\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_2} = \frac{1}{L}\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_2} = -\frac{1}{L}\,, && \frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_3} = \frac{1}{L}\,,\\
&\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\varepsilon}^{II}} = 0\,.
\end{aligned} \tag{A.38}$$

By substituting Equation (A.38) into Equation (A.37), one obtains:

$$\mathbf{R}' = \frac{A}{L}\begin{bmatrix} k\left(\boldsymbol{\varepsilon}^I\right) + k\left(\boldsymbol{\varepsilon}^{II}\right) & -k\left(\boldsymbol{\varepsilon}^{II}\right) \\ -k\left(\boldsymbol{\varepsilon}^{II}\right) & k\left(\boldsymbol{\varepsilon}^{II}\right) \end{bmatrix}\,. \tag{A.39}$$

In this case, the Newton-Raphson iterative method reads as follows:

$$\mathbf{R}\left(\{u_2, u_3\}^T\right) + \mathbf{R}'\left(\{u_2, u_3\}^T\right)\mathrm{d}\{u_2, u_3\}^T = 0\,, \tag{A.40}$$

where

$$\mathbf{R}\left(\{u_2, u_3\}^T\right) = A\left.\begin{Bmatrix} \boldsymbol{\sigma}^I - \boldsymbol{\sigma}^{II} \\ \boldsymbol{\sigma}^{II} - F/A \end{Bmatrix}\right|_{\{u_2,u_3\}^T}\,, \tag{A.41}$$

$$\mathbf{R}'\left(\{u_2, u_3\}^T\right) = \frac{A}{L}\left.\begin{bmatrix} k\left(\boldsymbol{\varepsilon}^I\right) + k\left(\boldsymbol{\varepsilon}^{II}\right) & -k\left(\boldsymbol{\varepsilon}^{II}\right) \\ -k\left(\boldsymbol{\varepsilon}^{II}\right) & k\left(\boldsymbol{\varepsilon}^{II}\right) \end{bmatrix}\right|_{\{u_2,u_3\}^T}\,. \tag{A.42}$$

Similar to displacement control, this procedure is repeated until convergence as discussed in Section 2.2. Please notice the differences in the two implementations of the Newton-Raphson iterative method by comparing the elaborations presented in this appendix.

Increasing complexity by considering the two-dimensional plane stress problem in Figure A.6 of a cantilever beam having two elements. Each square element has side lengths equal to L such that the total length of the considered structure equals $2L$. Only the displacement control implementation of the Newton-Raphson iterative method has been elaborated here for illustrative matters.

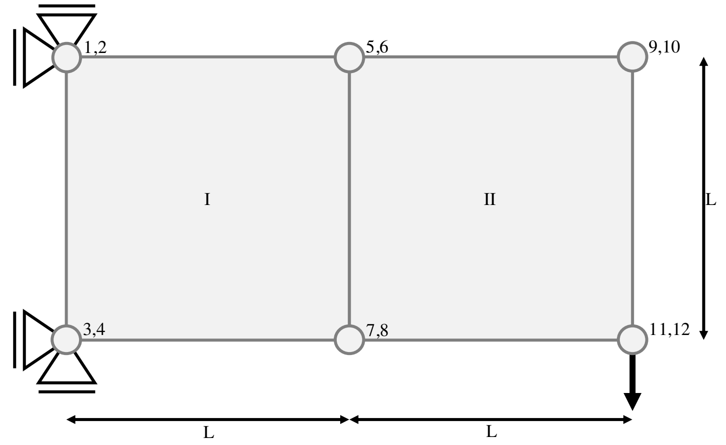


Figure A.6: Two-dimensional cantilever beam having an two elements

Considering the two-dimensional problem for a cantilever beam composed of two elements (see Figure A.6) which is solved using displacement control with the following boundary conditions:

$$\mathbf{u} = \left\{ \begin{array}{l} u_1 = 0 \\ u_2 = 0 \\ u_3 = 0 \\ u_4 = 0 \\ u_5 = ? \\ u_6 = ? \\ u_7 = ? \\ u_8 = ? \\ u_9 = ? \\ u_{10} = ? \\ u_{11} = ? \\ u_{12} = \Delta \end{array} \right\} . \tag{A.43}$$

In this case, the nodes have two DOFs such that each element has eight DOFs.

Given that $\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u}$, the strains elements I and II become, respectively

$$\boldsymbol{\varepsilon}^I = \mathbf{B}\mathbf{u}^I = \mathbf{B}\begin{Bmatrix} u_3 \\ u_4 \\ u_7 \\ u_8 \\ u_5 \\ u_6 \\ u_1 \\ u_2 \end{Bmatrix} , \qquad \boldsymbol{\varepsilon}^{II} = \mathbf{B}\mathbf{u}^{II} = \mathbf{B}\begin{Bmatrix} u_7 \\ u_8 \\ u_{11} \\ u_{12} \\ u_9 \\ u_{10} \\ u_5 \\ u_6 \end{Bmatrix} , \tag{A.44}$$

with

$$\mathbf{B} = \frac{1}{2L}\begin{bmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \end{bmatrix} . \tag{A.45}$$

This yields the following when worked out:

$$\begin{aligned} \boldsymbol{\varepsilon}^I &= \frac{1}{2L}\begin{Bmatrix} -u_3 + u_7 + u_5 - u_1 \\ -u_4 - u_8 + u_6 + u_2 \\ -u_3 - u_4 - u_7 + u_8 + u_5 + u_6 + u_1 - u_2 \end{Bmatrix} , \\ \boldsymbol{\varepsilon}^{II} &= \frac{1}{2L}\begin{Bmatrix} -u_7 + u_{11} + u_9 - u_5 \\ -u_8 - u_{12} + u_{10} + u_6 \\ -u_7 - u_8 - u_{11} + u_{12} + u_9 + u_{10} + u_5 - u_6 \end{Bmatrix} . \end{aligned} \tag{A.46}$$

Similarly, the elemental stresses are a function of their related strains, i.e. $\boldsymbol{\sigma}^i = \boldsymbol{\sigma}^i\left(\boldsymbol{\varepsilon}^i\right)$. Therefore, the stresses of elements I and II become, respectively

$$\boldsymbol{\sigma}^I = \boldsymbol{\sigma}^I\left(\boldsymbol{\varepsilon}^I\right) , \qquad \boldsymbol{\sigma}^{II} = \boldsymbol{\sigma}^{II}\left(\boldsymbol{\varepsilon}^{II}\right) . \tag{A.47}$$

Recall once more that the internal force vector can be obtained by solving the integral given in Equation (2.50): $\mathbf{f} = \int_V \mathbf{B}^T\boldsymbol{\sigma}\mathrm{d}V$. The two-dimensional plane stress elements have a constant thickness t such that it can be taken out of the integral. Insertion of Equation (A.47) into the integral yields the following internal force vectors of elements I and II, respectively

$$\mathbf{f}^I = t \begin{Bmatrix} \boldsymbol{\sigma}_3^I \\ \boldsymbol{\sigma}_4^I \\ \boldsymbol{\sigma}_7^I \\ \boldsymbol{\sigma}_8^I \\ \boldsymbol{\sigma}_5^I \\ \boldsymbol{\sigma}_6^I \\ \boldsymbol{\sigma}_1^I \\ \boldsymbol{\sigma}_2^I \end{Bmatrix}, \qquad \mathbf{f}^{II} = t \begin{Bmatrix} \boldsymbol{\sigma}_7^{II} \\ \boldsymbol{\sigma}_8^{II} \\ \boldsymbol{\sigma}_{11}^{II} \\ \boldsymbol{\sigma}_{12}^{II} \\ \boldsymbol{\sigma}_9^{II} \\ \boldsymbol{\sigma}_{10}^{II} \\ \boldsymbol{\sigma}_5^{II} \\ \boldsymbol{\sigma}_6^{II} \end{Bmatrix}. \tag{A.48}$$

From Equation (A.48) the *global* internal force vector can be assembled, which is given below next to the problem's external force vector:

$$\mathbf{f} = t \begin{Bmatrix} \boldsymbol{\sigma}_1^I \\ \boldsymbol{\sigma}_2^I \\ \boldsymbol{\sigma}_3^I \\ \boldsymbol{\sigma}_4^I \\ \boldsymbol{\sigma}_5^I + \boldsymbol{\sigma}_5^{II} \\ \boldsymbol{\sigma}_6^I + \boldsymbol{\sigma}_6^{II} \\ \boldsymbol{\sigma}_7^I + \boldsymbol{\sigma}_7^{II} \\ \boldsymbol{\sigma}_8^I + \boldsymbol{\sigma}_8^{II} \\ \boldsymbol{\sigma}_9^{II} \\ \boldsymbol{\sigma}_{10}^{II} \\ \boldsymbol{\sigma}_{11}^{II} \\ \boldsymbol{\sigma}_{12}^{II} \end{Bmatrix}, \qquad \mathbf{F} = \begin{Bmatrix} F_1 = ? \\ F_2 = ? \\ F_3 = ? \\ F_4 = ? \\ F_5 = 0 \\ F_6 = 0 \\ F_7 = 0 \\ F_8 = 0 \\ F_9 = 0 \\ F_{10} = 0 \\ F_{11} = 0 \\ F_{12} = ? \end{Bmatrix}. \tag{A.49}$$

Now that the *global* internal and external force vectors have been determined, the residual can be derived from Equation (A.49) as follows:

$$\mathbf{R} = \mathbf{f} - \mathbf{F} = t \begin{Bmatrix} \boldsymbol{\sigma}_1^I - F_1/t \\ \boldsymbol{\sigma}_2^I - F_2/t \\ \boldsymbol{\sigma}_3^I - F_3/t \\ \boldsymbol{\sigma}_4^I - F_4/t \\ \boldsymbol{\sigma}_5^I + \boldsymbol{\sigma}_5^{II} \\ \boldsymbol{\sigma}_6^I + \boldsymbol{\sigma}_6^{II} \\ \boldsymbol{\sigma}_7^I + \boldsymbol{\sigma}_7^{II} \\ \boldsymbol{\sigma}_8^I + \boldsymbol{\sigma}_8^{II} \\ \boldsymbol{\sigma}_9^{II} \\ \boldsymbol{\sigma}_{10}^{II} \\ \boldsymbol{\sigma}_{11}^{II} \\ \boldsymbol{\sigma}_{12}^{II} - F_{12}/t \end{Bmatrix} . \tag{A.50}$$

Identical to the one-dimensional problem, the unknown displacements $\mathbf{u}^u$ can be calculated by means of the displacement control implementation of the Newton-Raphson iterative method as described before. Please note again that since $\mathbf{F}^u = \mathbf{0}$, the residual related to the unconstrained degrees of freedom simply reduces to $\mathbf{R}^u = \mathbf{f}^u$. Similarly, the stiffness matrix considered for the successive iterations is derived by working out $\partial \mathbf{R}^u / \partial \mathbf{u}^u$ which leads to the following result:

$$\begin{aligned} \mathbf{R}' &= \frac{\partial \mathbf{R}^u}{\partial \mathbf{u}^u} \\ &= t \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{\sigma}_5^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_5} + \frac{\mathrm{d}\boldsymbol{\sigma}_5^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_5} & \frac{\mathrm{d}\boldsymbol{\sigma}_5^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_6} + \frac{\mathrm{d}\boldsymbol{\sigma}_5^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_6} & \cdots & \frac{\mathrm{d}\boldsymbol{\sigma}_5^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_{11}} + \frac{\mathrm{d}\boldsymbol{\sigma}_5^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_{11}} \\ \frac{\mathrm{d}\boldsymbol{\sigma}_6^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_5} + \frac{\mathrm{d}\boldsymbol{\sigma}_6^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_5} & \frac{\mathrm{d}\boldsymbol{\sigma}_6^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_6} + \frac{\mathrm{d}\boldsymbol{\sigma}_6^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_6} & \cdots & \frac{\mathrm{d}\boldsymbol{\sigma}_6^I}{\mathrm{d}\boldsymbol{\varepsilon}^I}\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_{11}} + \frac{\mathrm{d}\boldsymbol{\sigma}_6^{II}}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_{11}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathrm{d}\boldsymbol{\sigma}_{11}^I}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_5} & \frac{\mathrm{d}\boldsymbol{\sigma}_{11}^I}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_6} & \cdots & \frac{\mathrm{d}\boldsymbol{\sigma}_{11}^I}{\mathrm{d}\boldsymbol{\varepsilon}^{II}}\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_{11}} \end{bmatrix} . \end{aligned} \tag{A.51}$$

Provided $\mathrm{d}\boldsymbol{\sigma}^i/\mathrm{d}\boldsymbol{\varepsilon}^i = k\left(\boldsymbol{\varepsilon}^i\right)$, working out the terms concerning the derivatives of strain with respect to displacement in Equation (A.51) yields:

$$
\begin{aligned}
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_5} = \frac{1}{2L}\begin{Bmatrix}1\\0\\1\end{Bmatrix}, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_6} = \frac{1}{2L}\begin{Bmatrix}0\\1\\1\end{Bmatrix}, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_7} = \frac{1}{2L}\begin{Bmatrix}1\\0\\-1\end{Bmatrix}, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_8} = \frac{1}{2L}\begin{Bmatrix}0\\-1\\1\end{Bmatrix}, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_5} = \frac{1}{2L}\begin{Bmatrix}-1\\0\\1\end{Bmatrix}, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_6} = \frac{1}{2L}\begin{Bmatrix}0\\1\\-1\end{Bmatrix}, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_7} = \frac{1}{2L}\begin{Bmatrix}-1\\0\\-1\end{Bmatrix}, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_8} = \frac{1}{2L}\begin{Bmatrix}0\\-1\\-1\end{Bmatrix}, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_9} = 0, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_{10}} = 0, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^I}{\mathrm{d}u_{11}} = 0, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_9} = \frac{1}{2L}\begin{Bmatrix}1\\0\\1\end{Bmatrix}, \\
&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_{10}} = \frac{1}{2L}\begin{Bmatrix}0\\1\\1\end{Bmatrix}, \quad
&&\frac{\mathrm{d}\boldsymbol{\varepsilon}^{II}}{\mathrm{d}u_{11}} = \frac{1}{2L}\begin{Bmatrix}1\\0\\-1\end{Bmatrix}.
\end{aligned}
\tag{A.52}
$$

By substituting Equation (A.52) into Equation (A.51), one obtains the linearized global tangent stiffness matrix considered for the successive iterations. Similar as before, the Newton-Raphson iterative method reads as follows for the unknown displacements $\mathbf{u}^u$:

$$
\mathbf{R}(\mathbf{u}^u) + \mathbf{R}'(\mathbf{u}^u)\mathrm{d}\mathbf{u}^u = 0, \tag{A.53}
$$

where

$$
\mathbf{R}^u(\mathbf{u}^u) = t\left.\begin{Bmatrix}\boldsymbol{\sigma}_5^I + \boldsymbol{\sigma}_5^{II}\\ \boldsymbol{\sigma}_6^I + \boldsymbol{\sigma}_6^{II}\\ \boldsymbol{\sigma}_7^I + \boldsymbol{\sigma}_7^{II}\\ \boldsymbol{\sigma}_8^I + \boldsymbol{\sigma}_8^{II}\\ \boldsymbol{\sigma}_9^{II}\\ \boldsymbol{\sigma}_{10}^{II}\\ \boldsymbol{\sigma}_{11}^{II}\end{Bmatrix}\right|_{\mathbf{u}^u}. \tag{A.54}
$$

$\mathbf{R}'(\mathbf{u}^u)$ can be obtained through evaluating Equation (A.51) for the unknown displacements $\mathbf{u}^u$. Once again, this procedure is repeated until convergence as discussed in Section 2.2.

A.2 Arc Length Method

Even though displacement control can overcome limit points in the load-displacement curve and therefore circumvent problems like snap-through behaviours under load control, it should be noted however that a non-linear problem may also exhibit snap-back behaviours under displacement control or even both as illustrated in Figure A.7. An example of a load-displacement diagram with a snap-back is very brittle failure [43]. To model such behaviour, more robust (and more complex) numerical procedures are required [11]. This behaviour has never occurred in the author's experience and is therefore not implemented, but it could be necessary for material models having a relatively low fracture toughness or a relatively high specimen stiffness.

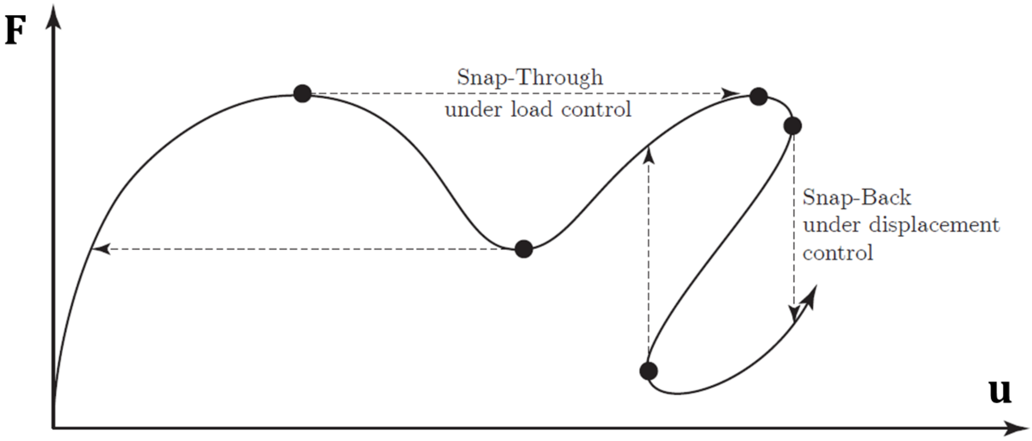


Figure A.7: A system that is unstable under load and displacement control [12]

One very efficient method to model structural behaviour as in Figure A.7 is the arc length method, in which variations in both the displacements $\Delta\mathbf{u}$ and load vector coefficient $\Delta\lambda$ are postulated simultaneously. The main difference is that in the arc-length method both $\Delta\mathbf{u}$ and $\Delta\lambda$ are unknowns in contrast to the Newton-Raphson iterative method where either $\Delta\lambda$ (load control) or $\Delta\mathbf{u}$ (displacement control) were given [12]. However, the arc length method is beyond the scope of the current research. Therefore, readers are referred to [4, 10–12, 43, 63–65] and references therein for further details.

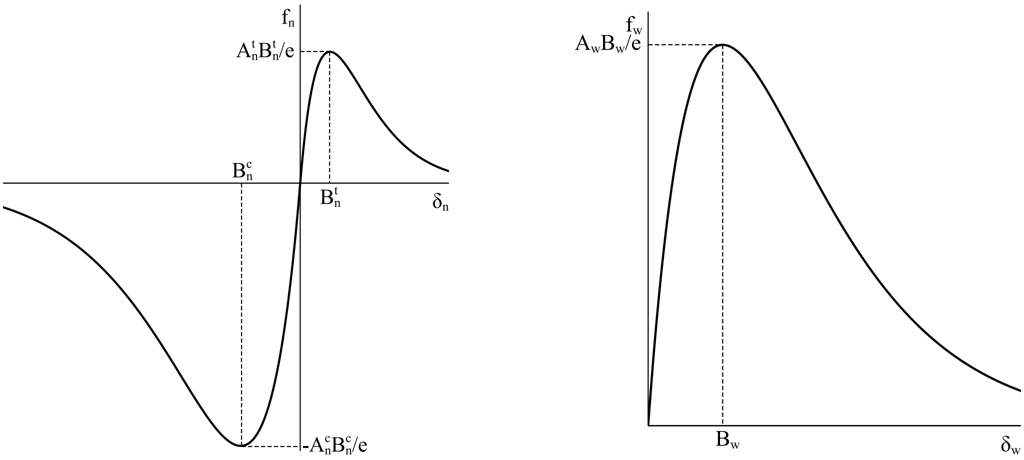
A.3 Newton-Raphson Iterative Methods & Comparison

Besides the previously performed element formulation validations in Section 3.1.2, the two implementations of the Newton-Raphson iterative method have also been compared for validation purposes. Even though load and displacement control solve non-linear systems in an opposite manner, both approaches should yield similar results for equivalent applied loads and displacements. Furthermore, it will be illustrated that load control does indeed fail near the limit point (where the tangent of the load-displacement curve becomes horizontal) and that displacement control can overcome this point. For that matter, the specifics and results of two rather simple three-dimensional non-linear FEA are discussed here. Both structures have identical support conditions and are composed of four elements. Initially the loading conditions are equivalent, meaning that the DOFs in the load controlled FEA are loaded by a force having a magnitude that pushes the system as close as possible to its limit point while the DOFs in the displacement controlled FEA are displaced by the same amount as obtained from the load controlled FEA. In other words: the load controlled FEA is loaded until $\mathbf{F} = \mathbf{F}_{applied}$ from which the displacements $\mathbf{u} = \mathbf{u}_{obtained}$ are obtained and the displacement controlled FEA is displaced until $\mathbf{u} = \mathbf{u}_{applied}$ resulting in the forces $\mathbf{F} = \mathbf{F}_{obtained}$, such that $\mathbf{F}_{applied} = \mathbf{F}_{obtained}$ and $\mathbf{u}_{obtained} = \mathbf{u}_{applied}$. Some specifics of both non-linear FEA are given in Figure A.9 while the resulting *global* force-displacement curves are shown in Figure A.10. For this particular case, the utilized grain-scale constitutive relationships are exponential with softening and are defined as:

$$\begin{aligned} f_{\mathrm{n}}^{\alpha} &= \begin{cases} A_{\mathrm{n}}^{t}\delta_{\mathrm{n}}^{\alpha}e^{-\frac{\delta_{\mathrm{n}}^{\alpha}}{B_{n}^{t}}} & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} \geq 0 \quad (\text{tension})\,, \\ A_{\mathrm{n}}^{c}\delta_{\mathrm{n}}^{\alpha}e^{\frac{\delta_{\mathrm{n}}^{\alpha}}{B_{n}^{c}}} & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} < 0 \quad (\text{compression})\,, \end{cases} \\ f_{\mathrm{w}}^{\alpha} &= A_{\mathrm{w}}\delta_{\mathrm{w}}^{\alpha}e^{-\frac{\delta_{\mathrm{w}}^{\alpha}}{B_{\mathrm{w}}}}\,, \end{aligned} \tag{A.55}$$

where constants A_{n}^{t}, B_{n}^{t}, A_{n}^{c}, B_{n}^{c}, A_{w} and B_{w} are model parameters. The resulting inter-granular force-displacement curves have a peak and subsequent softening as shown in Figure A.8. Parameters A (A_{n}^{t}, A_{n}^{c} and A_{w}) relate to the initial inter-granular stiffness for contacts in tension, compression and shear. The superscripts t and c refer to tension and compression, respectively. Parameters B (B_{n}^{t}, B_{n}^{c} and B_{w}) correspond to the inter-granular displacements at the peaks of the inter-granular force-displacement curves. The inter-granular stiffness can be derived from Equation (A.55) by direct differentiation as:

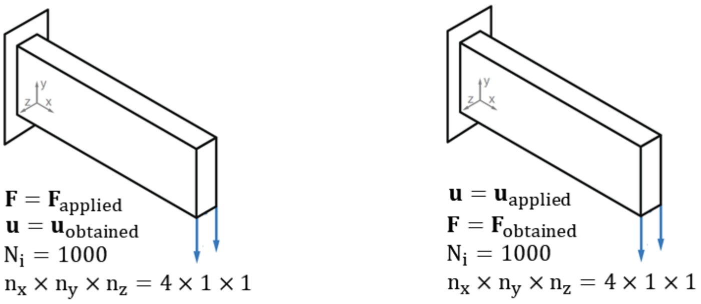
$$\begin{aligned} k_{\mathrm{n}}^{\alpha} &= \begin{cases} A_{\mathrm{n}}^{t}e^{-\frac{\delta_{\mathrm{n}}^{\alpha}}{B_{\mathrm{n}}^{t}}}\left(1-\frac{\delta_{\mathrm{n}}^{\alpha}}{B_{\mathrm{n}}^{t}}\right) & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} \geq 0 \quad (\text{tension})\,, \\ A_{\mathrm{n}}^{c}e^{\frac{\delta_{\mathrm{n}}^{\alpha}}{B_{\mathrm{n}}^{c}}}\left(1+\frac{\delta_{\mathrm{n}}^{\alpha}}{B_{\mathrm{n}}^{c}}\right) & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} < 0 \quad (\text{compression})\,, \end{cases} \\ k_{\mathrm{w}}^{\alpha} &= A_{\mathrm{w}}e^{-\frac{\delta_{\mathrm{w}}^{\alpha}}{B_{\mathrm{w}}}}\left(1-\frac{\delta_{\mathrm{w}}^{\alpha}}{B_{\mathrm{w}}}\right)\,. \end{aligned} \tag{A.56}$$



(a) Normal directions

(b) Tangential directions

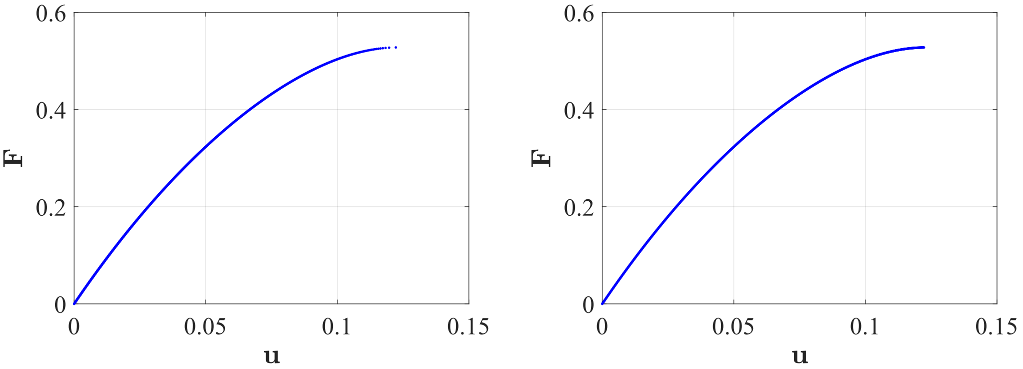
Figure A.8: Exponential inter-granular force-law with softening


y
z
x
$\mathbf{F} = \mathbf{F}_{\text{applied}}$
$\mathbf{u} = \mathbf{u}_{\text{obtained}}$
$N_i = 1000$
$n_x \times n_y \times n_z = 4 \times 1 \times 1$
$\mathbf{u} = \mathbf{u}_{\text{applied}}$
$\mathbf{F} = \mathbf{F}_{\text{obtained}}$
$N_i = 1000$
$n_x \times n_y \times n_z = 4 \times 1 \times 1$

(a) Load control

(b) Displacement control

Figure A.9: Boundary conditions of two identical three-dimensional structures



(a) Load control

(b) Displacement control

Figure A.10: Force-displacement curves for equivalent loading conditions

Note that it is unnecessary in non-linear FEA to use such small incrementation as in Figure A.10. This is merely done to get the load controlled FEA as close as possible to its limit point. However, it can be seen that both load control and displacement control yield similar results for equivalent loading and support conditions. Besides, through close inspection of Figure A.10 near the limit point one can actually see the difference in implementation of the Newton-Raphson iterative method: the converged states of the load controlled FEA (Figure A.10(a)) are further apart, because equilibrium is found for larger steps in the displacements compared to the fixed force increments. This is not the case in the displacement controlled FEA (Figure A.10(b)) due to the fixed displacement incrementation and successively decreasing steps in the forces.

When the applied loading conditions (i.e. the applied displacements) for the displacement controlled FEA are doubled, it can be illustrated that this implementation of the Newton-Raphson iterative method can overcome limit points. As a result, a structure's response can be traced beyond peaks in the force-displacement curve, which is the case for strain softening constitutive relationships. This capability is shown in Figure A.11, which depicts the *global* force-displacement curve for the exact same problem as in Figure A.9(b) whilst the applied displacements are doubled ($\mathbf{u} = 2\mathbf{u}_{applied}$). Please note again that the load incrementation is unnecessarily small in Figure A.11, but this has been chosen to be consistent with Figure A.10.

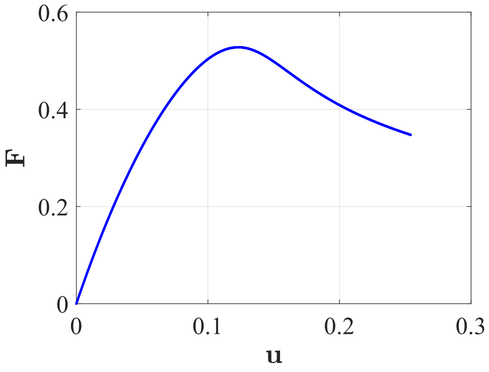


Figure A.11: Force-displacement curve for doubled loading conditions

Appendix B Numerical Integration Implementations

This appendix contains elaborations of several numerical integration implementations that have been used in this research. The respective numerical integration implementations are related to derivations of element stiffness matrices, internal force vectors and two approaches for calculating compliance.

B.1 Element Stiffness Matrices

In two-dimensional topology optimization codes the design domain is often discretized by rectangular finite elements, see for example [20, 33–36]. The rectangular finite elements used in these references are four-node bi-linear elements, as shown in Figure 2.7. These elements are also referred to as bi-linear quadrilateral or Q4 elements. As shown by [5], the element stiffness matrix of such elements can be derived by exact integration solving the integral in Equation (2.46) using the strain-displacement matrix **B**, the constant constitutive matrix **D** for a linear elastic material and constant thickness t.

However, the constitutive matrix **D** becomes anisotropic and a function of strain in the Granular Micromechanics Approach (GMA). Hence, its final form is unknown beforehand and it is not a function of Young's modulus E and Poisson's ratio ν as it is in [5, 47]. Since the constitutive matrix changes with respect to loading and deformation (i.e. strain) for every element, the integral in Equation (2.46) should be evaluated for all elements in every increment of the analysis. For this purpose, it is advantageous to use numerical integration to derive the element stiffness matrix of the four-node bi-linear elements. For that matter, the nodal coordinates of the finite elements have been transformed into natural coordinates as shown in Figure 2.6. For finite elements having width b and height h, the following transformations to this configuration are required:

$$\begin{aligned} \xi &= \frac{2x}{b} \rightarrow x = \frac{\xi b}{2}\,, \\ \eta &= \frac{2y}{h} \rightarrow y = \frac{\eta h}{2}\,. \end{aligned} \tag{B.1}$$

As derived by [5], the strain-displacement matrix **B** is:

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_y}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_y}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}, \end{aligned} \tag{B.2}$$

with the following shape functions:

$$
\begin{aligned}
N_1 &= \frac{1}{4}(1-2\frac{x}{b})(1-2\frac{y}{h})\,, & N_3 &= \frac{1}{4}(1+2\frac{x}{b})(1+2\frac{y}{h})\,, \\
N_2 &= \frac{1}{4}(1+2\frac{x}{b})(1-2\frac{y}{h})\,, & N_4 &= \frac{1}{4}(1-2\frac{x}{b})(1+2\frac{y}{h})\,.
\end{aligned}
\tag{B.3}
$$

The strain-displacement matrix $\mathbf{B}$ can be written in terms of b and h:

$$
\mathbf{B} = \begin{bmatrix}
\frac{-1}{2b} & 0 & \frac{1}{2b} & 0 & \frac{1}{2b} & 0 & \frac{-1}{2b} & 0 \\
0 & \frac{-1}{2h} & 0 & \frac{-1}{2h} & 0 & \frac{1}{2h} & 0 & \frac{1}{2h} \\
\frac{-1}{2h} & \frac{-1}{2b} & \frac{-1}{2h} & \frac{1}{2b} & \frac{1}{2h} & \frac{1}{2b} & \frac{1}{2h} & \frac{-1}{2b}
\end{bmatrix}\,. \tag{B.4}
$$

According to [16] the integral for deriving the element stiffness matrix over the element domain Ω_e can be transformed to the $\xi - \eta$ coordinate system as:

$$
\begin{aligned}
\mathbf{K}_e &= t\int_{(\Omega_e)} \mathbf{B}(x,y)^T\mathbf{D}\mathbf{B}(x,y)\mathrm{d}A \\
&= t\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta)^T\mathbf{D}\mathbf{B}(\xi,\eta)\det\mathbf{J_e}\mathrm{d}\xi\mathrm{d}\eta\,.
\end{aligned}
\tag{B.5}
$$

Thus, we obtain:

$$
\begin{aligned}
\mathbf{K}_e &= t\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta)^T\mathbf{D}\mathbf{B}(\xi,\eta)\det\mathbf{J_e}\mathrm{d}\xi\mathrm{d}\eta \\
&\approx t\sum_{p=1}^{n_p} \mathbf{B}(\xi,\eta)^T\mathbf{D}\mathbf{B}(\xi,\eta)\det\mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p})W_p\,,
\end{aligned}
\tag{B.6}
$$

where the determinant of the *Jacobian* is computed as follows:

$$
\mathbf{J_e}(\xi,\eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{b}{2} & 0 \\ 0 & \frac{h}{2} \end{bmatrix} \rightarrow \det\mathbf{J_e}(\xi,\eta) = \frac{b}{2}\times\frac{h}{2} = \frac{bh}{4}\,. \tag{B.7}
$$

When proceeding with three-dimensional topology optimization, a similar coding structure to the 99-line code by Sigmund [36] and the 88-line code by Andreassen et al. [33] was found in the 169-line program [6]. In this program the design domain is discretized by eight-node hexahedral elements, as shown in Figure 2.8. These elements are also referred to as hex or (eight-node) brick elements. Similar to the two-dimensional derivation, the element stiffness matrix of three-dimensional elements can be derived by means of exact integration as shown by [6]. In general this yields solving the integral in Equation (2.48), as described in [17,46].

As explained before, in GMA the constitutive matrix is a function of strain. Again, since the constitutive matrix changes with respect to the strain in every element, the integral in Equation (2.48) should be evaluated for all elements in every increment of the analysis. In this case the nodal coordinates of the finite elements are already written in natural coordinates, which is advantageous for performing numerical integration to derive the element stiffness matrix of the eight-node hexahedral elements. For finite elements having width b, height h and thickness t, the following transformations are required:

$$\begin{aligned} \xi &= \frac{2x}{b} \rightarrow x = \frac{\xi b}{2}\,, \\ \eta &= \frac{2y}{h} \rightarrow y = \frac{\eta h}{2}\,, \\ \zeta &= \frac{2z}{t} \rightarrow z = \frac{\zeta t}{2}\,. \end{aligned} \tag{B.8}$$

According to [6,46], for the three-dimensional elements the strain-displacement matrix takes the following form for $q = 1, \ldots, 8$:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \ldots & \mathbf{B}_q \end{bmatrix}\,, \tag{B.9}$$

with:

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial z} & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} & 0 & \frac{\partial N_i}{\partial x} \end{bmatrix}\,, \tag{B.10}$$

given the following shape functions:

$$N_q = \frac{1}{8}\left\{\begin{array}{c} (1-2\frac{x}{b})(1-2\frac{y}{h})(1-2\frac{z}{t}) \\ (1+2\frac{x}{b})(1-2\frac{y}{h})(1-2\frac{z}{t}) \\ (1+2\frac{x}{b})(1+2\frac{y}{h})(1-2\frac{z}{t}) \\ (1-2\frac{x}{b})(1+2\frac{y}{h})(1-2\frac{z}{t}) \\ (1-2\frac{x}{b})(1-2\frac{y}{h})(1+2\frac{z}{t}) \\ (1+2\frac{x}{b})(1-2\frac{y}{h})(1+2\frac{z}{t}) \\ (1+2\frac{x}{b})(1+2\frac{y}{h})(1+2\frac{z}{t}) \\ (1-2\frac{x}{b})(1+2\frac{y}{h})(1+2\frac{z}{t}) \end{array}\right\}. \tag{B.11}$$

According to [46], similar to the two-dimensional derivation, the integral for deriving the three-dimensional element stiffness matrix can be transformed to the $\xi - \eta - \zeta$ coordinate system as:

$$\begin{aligned}\mathbf{K}_e &= \int_V \mathbf{B}(x,y,z)^T \mathbf{C}\mathbf{B}(x,y,z)\mathrm{d}V \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta,\zeta)^T \mathbf{C}\mathbf{B}(\xi,\eta,\zeta)\det\mathbf{J_e}\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta\,.\end{aligned} \tag{B.12}$$

Thus, we obtain:

$$\begin{aligned}\mathbf{K}_e &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta,\zeta)^T \mathbf{C}\mathbf{B}(\xi,\eta,\zeta)\det\mathbf{J_e}\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \\ &\approx \sum_{p=1}^{n_p} \mathbf{B}(\xi,\eta,\zeta)^T \mathbf{C}\mathbf{B}(\xi,\eta,\zeta)\det\mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p},\zeta_\mathbf{p})W_p\,,\end{aligned} \tag{B.13}$$

where now the determinant of the *Jacobian* matrix becomes [46]:

$$\mathbf{J_e}(\xi,\eta,\zeta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{b}{2} & 0 & 0 \\ 0 & \frac{h}{2} & 0 \\ 0 & 0 & \frac{t}{2} \end{bmatrix} \rightarrow \det\mathbf{J_e}(\xi,\eta,\zeta) = \frac{b}{2}\times\frac{h}{2}\times\frac{t}{2} = \frac{bht}{8}\,. \tag{B.14}$$

B.2 Internal Force Vectors

In linear-elastic FEM, the internal force vector is simply obtained from $\mathbf{f} = \mathbf{K}\mathbf{u}$. However, with non-linear material behaviour the internal force vector becomes also non-linear. Therefore, in non-linear FEM the elemental internal force vector can be obtained by solving the integral in Equation (2.50) using the strain-displacement matrix $\mathbf{B}$ and the stress vector $\boldsymbol{\sigma}$, as stated in for example [12, 38–40, 43, 49].

Analogous to the element's stiffness matrices being a function of strain in GMA, as a result the stress vector is also a function of strain. Therefore, the stress vector also changes with respect to the strain in every increment causing that the integral in Equation (2.50) should be evaluated for all elements in every increment of the analysis as well. Therefore, numerical integration is used once again to derive the elemental internal force vector of the elements.

Using the same strain-displacement matrices for the two-dimensional and three-dimensional problems, Equations (B.4) and (B.10) respectively, the elemental internal force vectors for both problems can be derived. First, the two-dimensional problem will be treated:

$$\begin{aligned}\mathbf{f} &= t\int_{(\Omega_e)} \mathbf{B}(\mathbf{x},\mathbf{y})^T \sigma \mathrm{d}A \\ &= t\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta)^T \sigma \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \,.\end{aligned} \tag{B.15}$$

Thus, we obtain:

$$\begin{aligned}\mathbf{f} &= t\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta)^T \sigma \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \\ &\approx t\sum_{p=1}^{n_p} \mathbf{B}(\xi,\eta)^T \sigma \det \mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p}) W_p \,.\end{aligned} \tag{B.16}$$

The *Jacobian* matrix is found, including its determinant, in Equation (B.7). Similarly, the three-dimensional elemental internal force vector can be derived:

$$\begin{aligned}\mathbf{f} &= \int_V \mathbf{B}(\mathbf{x},\mathbf{y},\mathbf{z})^T \sigma \mathrm{d}V \\ &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta,\zeta)^T \sigma \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \,.\end{aligned} \tag{B.17}$$

Thus, we obtain:

$$\begin{aligned}\mathbf{f} &= \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} \mathbf{B}(\xi,\eta,\zeta)^T \sigma \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \\ &\approx \sum_{p=1}^{n_p} \mathbf{B}(\xi,\eta,\zeta)^T \sigma \det \mathbf{J_e}(\xi_\mathbf{p},\eta_\mathbf{p},\zeta_\mathbf{p}) W_p \,,\end{aligned} \tag{B.18}$$

where the *Jacobian* is found, including its determinant, in Equation (B.14).

B.3 Compliance From Strain Energy

The total work done by the externally applied force, W in Figure B.1, is given by the area enclosed by the force-displacement curve, which is thus equivalent to strain energy [66]. In mathematical form, this yields solving the following integral [67]:

$$W = U = \int_0^x \mathbf{F} \mathrm{d}x \,, \tag{B.19}$$

where $\mathbf{F}$ is the externally applied load vector and $\mathrm{d}x$ is the width of an element under the force-displacement curve, i.e. the incremental displacement vector.

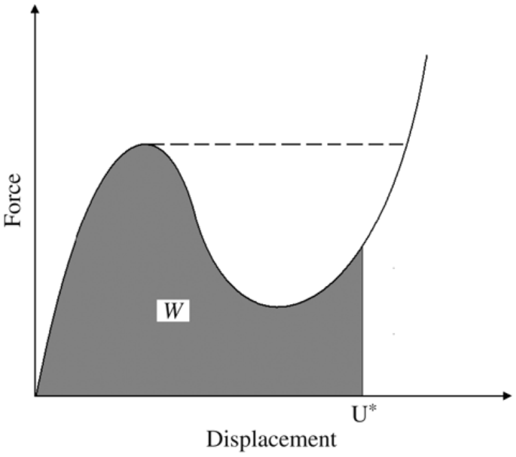


Figure B.1: Non-linear load-displacement curve including work [13]

Since compliance is equal to the overall strain energy [50], it immediately follows from Equation (B.19):

$$c = W = U = \int_0^{\mathbf{u}} \mathbf{F} \mathrm{d}\mathbf{u} \,. \tag{B.20}$$

Similar to the compliance computed using strain energy density, the equation above is also implemented using a Riemann sum (midpoint rule) approximation in the non-linear FEM:

$$c = \sum_{i=1}^{N_i} \left(\mathbf{F}^i - \frac{\mathrm{d}\mathbf{F}}{2} \right) \mathrm{d}\mathbf{u} \,, \tag{B.21}$$

where i is the current increment, N_i the total number of increments, $\mathbf{F}^i$ the external force vector of the current increment, $\mathrm{d}\mathbf{F}$ the incremental external force vector (i.e. $\mathrm{d}\mathbf{F} = \mathbf{F}^i - \mathbf{F}^{i-1}$) and $\mathrm{d}\mathbf{u}$ the incremental displacement vector (i.e. $\mathrm{d}\mathbf{u} = \mathbf{u}^i - \mathbf{u}^{i-1}$).

The advantages of this approach are that the compliance can be calculated directly (without the need for a loop over all the elements) and that the input for Equation (B.21) is readily available as the solution of the Newton-Raphson iterative method. However, a possible disadvantage of this method depends on the type of finite element analysis: load or displacement control. In load controlled FEM, the system of equilibrium equations is initially only solved for the *free degrees of* *freedom*, meaning the degrees of freedom without any boundary conditions (loads and supports). Therefore, the reaction forces (occurring at the supports) are not directly solved for, hence these are not included in the external force vector. Solving for the support reactions in load controlled FEM yields an additional step. In contrast, displacement controlled FEM does solve for all the external forces, including the support reaction forces. Therefore, no additional steps are required when displacement control is used.

B.4 Compliance From Strain Energy Density

Similar to the compliance computed using strain energy (see Equation (B.21)), Equation (2.54) is also implemented using a Riemann sum (midpoint rule) approximation in the non-linear FEM. However, as discussed previously the stress vector also changes with respect to the strain in every increment causing that the integral in that equation should be evaluated for all elements in every increment of the analysis. Again, numerical integration is used to derive the elemental compliance values. Starting with the two-dimensional problem:

$$\begin{aligned} c &= \sum_{i=1}^{N_i} \left[t \int_{(\Omega_e)} \left(\sigma_{ij}^i - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \mathrm{d}A \right] \\ &= \sum_{i=1}^{N_i} \left[t \int_{-1}^{+1} \int_{-1}^{+1} \left(\sigma_{ij}^i - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \right] . \end{aligned} \tag{B.22}$$

Thus, we obtain:

$$\begin{aligned} c &= \sum_{i=1}^{N_i} \left[t \int_{-1}^{+1} \int_{-1}^{+1} \left(\sigma_{ij}^i - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \right] \\ &\approx \sum_{i=1}^{N_i} \left[t \sum_{p=1}^{n_p} \left(\sigma_{ij}^i - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \det \mathbf{J_e}(\xi_p, \eta_p) W_p \right] , \end{aligned} \tag{B.23}$$

where σ_{ij}^i is the stress tensor in the current increment, $d\sigma_{ij}$ the incremental stress tensor (i.e. $d\sigma_{ij} = \sigma_{ij}^i - \sigma_{ij}^{i-1}$), $d\varepsilon_{ij}$ the incremental strain tensor (i.e. $d\varepsilon_{ij} = \varepsilon_{ij}^i - \varepsilon_{ij}^{i-1}$) and $\mathbf{J_e}$ the *Jacobian* matrix which is found including its determinant in Equation (B.7). Similarly, the compliance for the three-dimensional problem can be derived:

$$\begin{aligned} c &= \sum_{i=1}^{N_i} \left[\int_V \left(\sigma_{ij}^i - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \mathrm{d}V \right] \\ &= \sum_{i=1}^{N_i} \left[\int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \left(\sigma_{ij}^i - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \det J_e \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \right] . \end{aligned} \tag{B.24}$$

Thus, we obtain:

$$\begin{aligned} c &= \sum_{i=1}^{N_i} \left[\int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \left(\sigma_{ij}^{i} - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \det \mathbf{J_e} \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \right] \\ &\approx \sum_{i=1}^{N_i} \left[\sum_{p=1}^{n_p} \left(\sigma_{ij}^{i} - \frac{d\sigma_{ij}}{2} \right) d\varepsilon_{ij} \det \mathbf{J_e}(\xi_p, \eta_p, \zeta_p) W_p \right] , \end{aligned} \tag{B.25}$$

where $\mathbf{J_e}$ is the *Jacobian*, found including its determinant in Equation (B.14).

Appendix C Element Formulation Validations & Post Processing

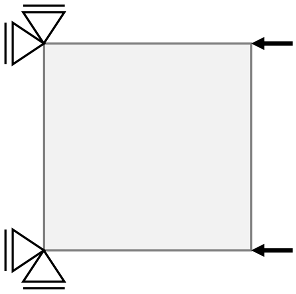
In addition to the element formulation validations as shown in Section 3.1.2, this appendix presents additional ones ranging from very simple (one element) to more sophisticated structures (400 elements). Furthermore, the author's post processing code will be used for illustration purposes and compared to ABAQUS. Finally, some three-dimensional element formulation validations will be performed.

C.1 Additional Element Formulation Validations

Besides the element formulation validations presented in Section 3.1.2, some additional (more sophisticated) structures having different boundary conditions and mesh refinements have been checked with ABAQUS. All structures considered here are variations of a cantilever (beam) that is exclusively loaded in one directional compression, which is applied in the axial (horizontal) direction. Therefore, all the structure's left edge nodes are fully clamped and all right edge nodes are loaded by a force of the same magnitude. Mesh refinements of 1, 4 and 400 elements are checked, see Figures C.1, C.3 and C.5. In addition to the numerical comparisons in Section 3.1.2, the author's post processing `MATLAB` code has been utilized for visual comparison of the deformed structures as discussed in that same section.

Identical to the findings in Section 3.1.2, Tables C.1, C.2 and C.3 show that the developed `MATLAB` FEM code obtains similar results as ABAQUS for the additional (more sophisticated) structures as well. Please note that the displacements, strains and stresses listed in Table C.3 are indicated in Figure C.6 by means of the red dots (DOFs) and squares (elements). Please note again that the omitted results are either zero or below machine precision. In addition, Figures C.2, C.4 and C.6 illustrate that the `MATLAB` code yields similar deformation patterns as ABAQUS for the same scale factors. Figure C.7 is included to emphasize the similarity of the deformation pattern, even in the smallest details, for a scale factor of 100.

C.1.1 Cantilever Structure (1 element)

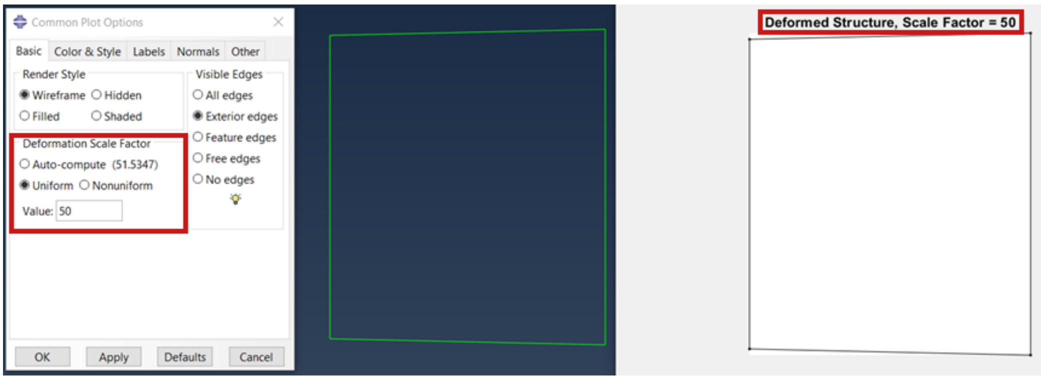


Cantilever structure (1 element)		
Structure size	1×1	mm
Plane thickness	0.01	mm
Fixed DOFs	1,2,3,4	(left edge nodes)
Applied load	1E-05	N (DOFs 5 & 7)
Element type	CP4S	(Q4 plane stress element)
Mesh	1×1	(mesh size = 1×1 mm)
Young's modulus	1	MPa
Poisson's ratio	0.3	-

Figure C.1: Cantilever structure composed of 1 element

Table C.1: Results cantilever structure (1 element)

MATLAB	**Results**	**Units**	**ABAQUS**	**Results**	**Units**
u_5	-0.001940441	mm	N3 U11	-0.00194044	mm
u_6	0.000401471	mm	N3 U22	0.000401471	mm
u_7	-0.001940441	mm	N4 U11	-0.00194044	mm
u_8	-0.000401471	mm	N4 U22	-0.000401471	mm
ε_{11}	-0.001940441	-	E11	-0.00194044	-
ε_{22}	0.000401471	-	E22	0.000401471	-
σ_{11}	-0.002	MPa	S11	-0.002	MPa
σ_{22}	-0.000198529	MPa	S22	-0.000198529	MPa



Common Plot Options

Render Style: ● Wireframe ○ Hidden ○ Filled ○ Shaded

Deformation Scale Factor: ○ Auto-compute (51.5347) ● Uniform ○ Nonuniform

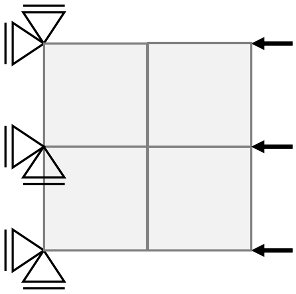
Value: 50

Visible Edges: ○ All edges ● Exterior edges ○ Feature edges ○ Free edges ○ No edges

Deformed Structure, Scale Factor = 50

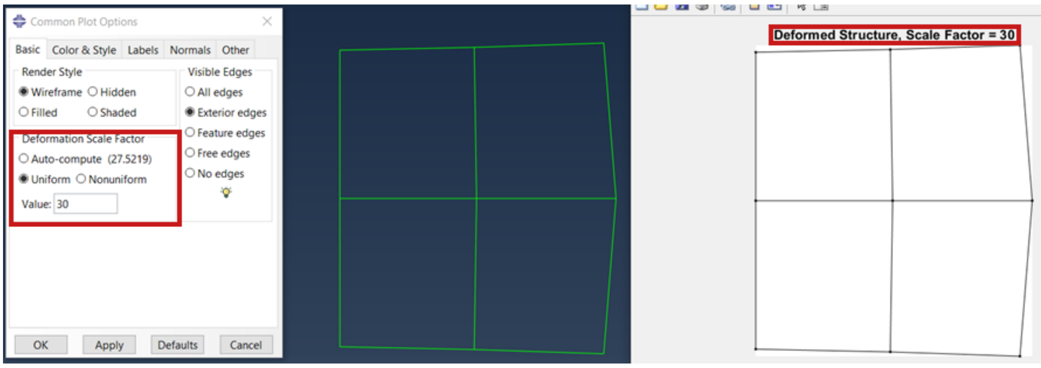
Figure C.2: Scale factor = 5 (1 element; ABAQUS (L), MATLAB (R))

C.1.2 Cantilever structure (4 elements)



Cantilever structure (4 elements)		
Structure size	1×1	mm
Plane thickness	0.01	mm
Fixed DOFs	1 to 6	(left edge nodes)
Applied load	1E-05	N (DOFs 7,9,11)
Element type	CP4S	(Q4 plane stress element)
Mesh	2×2	(mesh size = 0.5×0.5 mm)
Young's modulus	1	MPa
Poisson's ratio	0.3	-

Figure C.3: Cantilever structure composed of 4 elements



Render Style
● Wireframe ○ Hidden
○ Filled ○ Shaded

Deformation Scale Factor
○ Auto-compute (27.5219)
● Uniform ○ Nonuniform
Value: 30

Visible Edges
○ All edges
● Exterior edges
○ Feature edges
○ Free edges
○ No edges

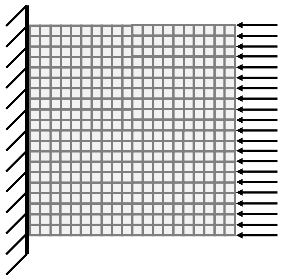
Deformed Structure, Scale Factor = 30

Figure C.4: Scale factor = 30 (4 elements; ABAQUS (L), MATLAB (R))

Table C.2: Results cantilever structure (4 elements)

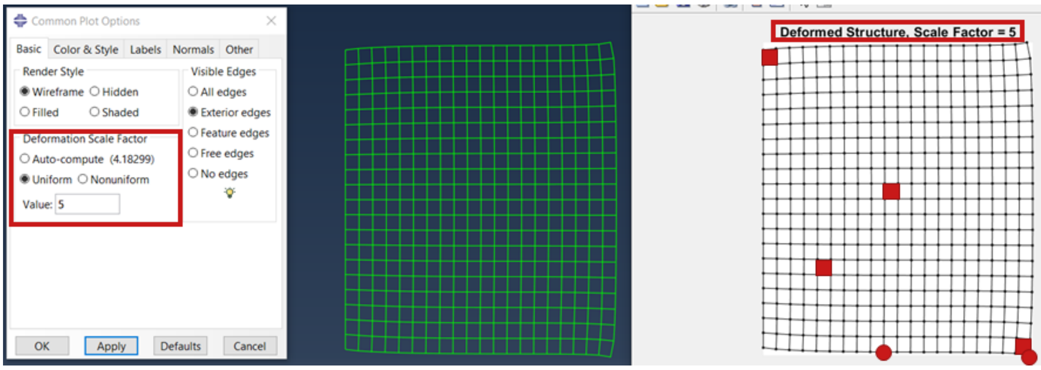
`MATLAB`	**Results**	**Units**	**ABAQUS**	**Results**	**Units**
u_7	-0.001541905	mm	N4 U11	-0.0015419	mm
u_8	0.000312025	mm	N4 U22	0.000312025	mm
u_9	-0.001281703	mm	N5 U11	-0.0012817	mm
u_{11}	-0.001541905	mm	N6 U11	-0.0015419	mm
u_{12}	-0.000312025	mm	N6 U22	-0.000312025	mm
u_{13}	-0.003633465	mm	N7 U11	-0.00363346	mm
u_{14}	0.000794847	mm	N7 U22	0.000794847	mm
u_{15}	-0.002252205	mm	N8 U11	-0.0022522	mm
u_{17}	-0.003633465	mm	N9 U11	-0.00363346	mm
u_{18}	-0.000794847	mm	N9 U22	-0.000794847	mm
Element 1					
ε_{11}	-0.002823608	-	E11	-0.00282361	-
ε_{22}	0.000312025	-	E22	0.000312025	-
ε_{12}	5.18E-05	-	E12	5.18E-05	-
σ_{11}	-0.003	MPa	S11	-0.003	MPa
σ_{22}	-0.000587975	MPa	S22	-0.000587975	MPa
σ_{12}	1.99E-05	MPa	S12	1.99E-05	MPa
Element 2					
ε_{11}	-0.002823608	-	E11	-0.00282361	-
ε_{22}	0.000312025	-	E22	0.000312025	-
ε_{12}	-5.18E-05	-	E12	-5.18E-05	-
σ_{11}	-0.003	MPa	S11	-0.003	MPa
σ_{22}	-0.000587975	MPa	S22	-0.000587975	MPa
σ_{12}	-1.99E-05	MPa	S12	-1.99321E-05	MPa
Element 3					
ε_{11}	-0.003062062	-	E11	-0.00306206	-
ε_{22}	0.001106872	-	E22	0.00110687	-
ε_{12}	-1.16E-03	-	E12	-0.00115864	-
σ_{11}	-0.003	MPa	S11	-0.003	MPa
σ_{22}	0.000206872	MPa	S22	0.000206872	MPa
σ_{12}	-4.46E-04	MPa	S12	-4.46E-04	MPa
Element 4					
ε_{11}	-0.003062062	-	E11	-0.00306206	-
ε_{22}	0.001106872	-	E22	0.00110687	-
ε_{12}	1.16E-03	-	E12	0.00115864	-
σ_{11}	-0.003	MPa	S11	-0.003	MPa
σ_{22}	0.000206872	MPa	S22	0.000206872	MPa
σ_{12}	4.46E-04	MPa	S12	0.000445631	MPa

C.1.3 Cantilever Structure (400 elements)



Cantilever structure (400 elements)		
Structure size	1×1	mm
Plane thickness	0.01	mm
Fixed DOFs	1 to 42	(left edge nodes)
Applied load	1E-05	N (uneven DOFs 841 to 881)
Element type	CP4S	(Q4 plane stress element)
Mesh	20×20	(mesh size = 0.05×0.05 mm)
Young's modulus	1	MPa
Poisson's ratio	0.3	-

Figure C.5: Cantilever structure composed of 400 elements



Render Style: ● Wireframe ○ Hidden ○ Filled ○ Shaded

Deformation Scale Factor: ○ Auto-compute (4.18299) ● Uniform ○ Nonuniform

Value: 5

Visible Edges: ○ All edges ● Exterior edges ○ Feature edges ○ Free edges ○ No edges

Deformed Structure, Scale Factor = 5

Figure C.6: Scale factor = 5 (400 elements; ABAQUS (L), MATLAB (R))

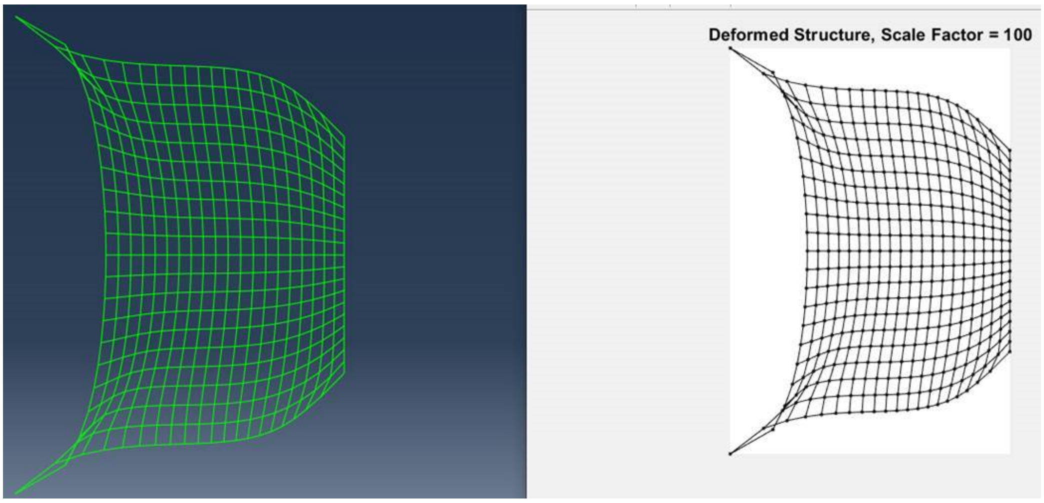


Figure C.7: Scale factor = 100 (400 elements; ABAQUS (L), MATLAB (R))

Table C.3: Results cantilever structure (400 elements)

`MATLAB`	**Results**	**Units**	**ABAQUS**	**Results**	**Units**
u_{419}	-0.009587544	mm	U419	-9.59E-03	mm
u_{420}	-0.002892864	mm	U420	-2.89E-03	mm
u_{881}	-2.39E-02	mm	U881	-2.39E-02	mm
u_{882}	-0.00508218	mm	U882	-0.00508218	mm
Element 1					
ε_{11}	-0.025480671	-	E11	-0.0254807	-
ε_{22}	0.0033924	-	E22	0.0033924	-
ε_{12}	1.21E-02	-	E12	1.21E-02	-
σ_{11}	-0.026882364	MPa	S11	-0.0268824	MPa
σ_{22}	-0.004672309	MPa	S22	-0.00467231	MPa
σ_{12}	4.65E-03	MPa	S12	4.65E-03	MPa
Element 95					
ε_{11}	-0.020568557	-	E11	-0.0205686	-
ε_{22}	0.004343312	-	E22	0.00434331	-
ε_{12}	-1.78E-03	-	E12	-1.78E-03	-
σ_{11}	-0.021170949	MPa	S11	-0.0211709	MPa
σ_{22}	-0.002007972	MPa	S22	-0.00200797	MPa
σ_{12}	-6.86E-04	MPa	S12	-0.000686016	MPa
Element 190					
ε_{11}	-0.020603345	-	E11	-0.0206033	-
ε_{22}	0.005368613	-	E22	0.00536861	-
ε_{12}	-1.46E-04	-	E12	-0.000145586	-
σ_{11}	-0.020871166	MPa	S11	-0.0208712	MPa
σ_{22}	-0.000892736	MPa	S22	-0.000892736	MPa
σ_{12}	-5.60E-05	MPa	S12	-5.60E-05	MPa
Element 400					
ε_{11}	-0.031419481	-	E11	-0.0314195	-
ε_{22}	0.010924058	-	E22	0.0109241	-
ε_{12}	8.48E-03	-	E12	0.00847505	-
σ_{11}	-0.030925565	MPa	S11	-0.0309256	MPa
σ_{22}	0.001646389	MPa	S22	0.00164639	MPa
σ_{12}	3.26E-03	MPa	S12	0.00325964	MPa

C.2 Post Processing

As discussed in Section 3.1.2, the author's post processing code in `MATLAB` can also display any element variable of his choice in a contour plot. The contour plots can be either displayed in the undeformed as well as in the deformed configuration by any scale factor. However, it should be noted that the post processing code is only able to display any element variable through one colour. Therefore, each displayed element variable is the average value of all that element's integration points. In other words, each element variable gets displayed as if it was measured at the element centroid.

For illustration and comparison purposes, some contour plots of the latter considered structure (see Figure C.5) have been obtained by the author's post processing `MATLAB` code and ABAQUS. In this case, the contour plots have been obtained in the deformed configuration with a scale factor of 5 (similar to Figure C.6). Recall that in Section 2.2.2 it was discussed that for two-dimensional elements, ABAQUS utilizes four integration points per element. As a result, the contour plots obtained from ABAQUS display each element variable by means of 75% averaged values for four integration points while the author's code displays the average values at the element centroids. Therefore, the color transitions in ABAQUS are smoother compared to those in `MATLAB`, since the display of element variables in ABAQUS is not restricted to only one color per element. Contour plots of the structure's in-plane normal strains (ε_{11} & ε_{22}) and stresses (σ_{11} & σ_{22}) have been obtained.

Even though the same number of steps in the colour bars has been chosen, the spectra are slightly different (especially at their centers where the transition yellow-green-blue takes place). Nevertheless, the contour plots remain (sufficiently) comparable in terms of colours. Despite the ABAQUS contour plots displaying smoother, 75% averaged results and the ones from `MATLAB` displaying average results in the element's centroids, the contour plots are showing significant similarities. For example: both contour plots of ε_{22} (Figure C.9) show a clear dark blue arch spanning from top to bottom at the left edge of the structures that gradually transitions into the cyan outer arch. Besides, both plots display some sort of light green/blue 'X' shape at the centers of the structure. Even the two strain peaks at the top and bottom corners at the right edge of structures are captured in both contour plots, as well as the slightly lower values at the center of the right edge.

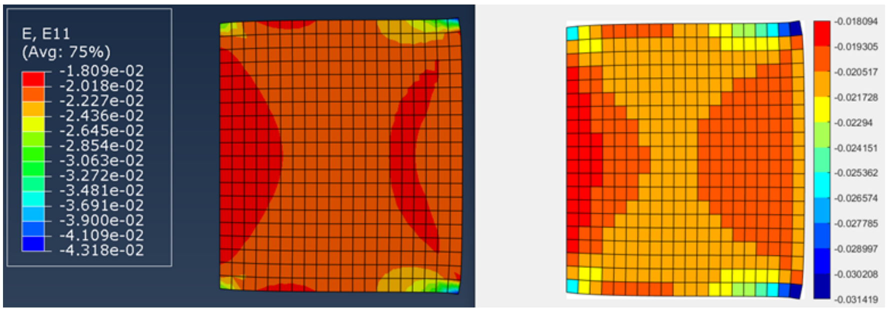


Figure C.8: Contour plot of ε_{11} (400 elements; ABAQUS (L), MATLAB (R))

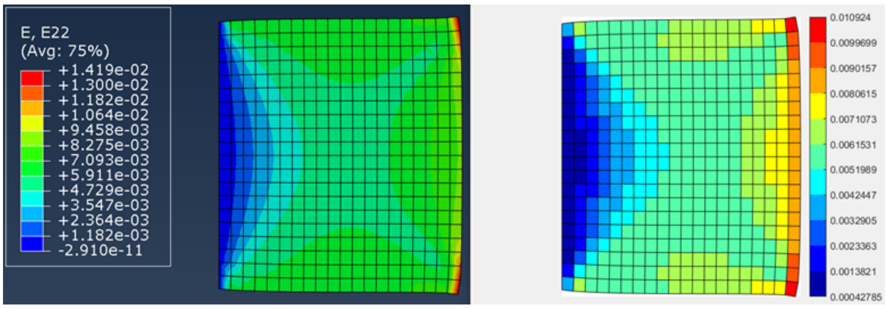


Figure C.9: Contour plot of ε_{22} (400 elements; ABAQUS (L), MATLAB (R))

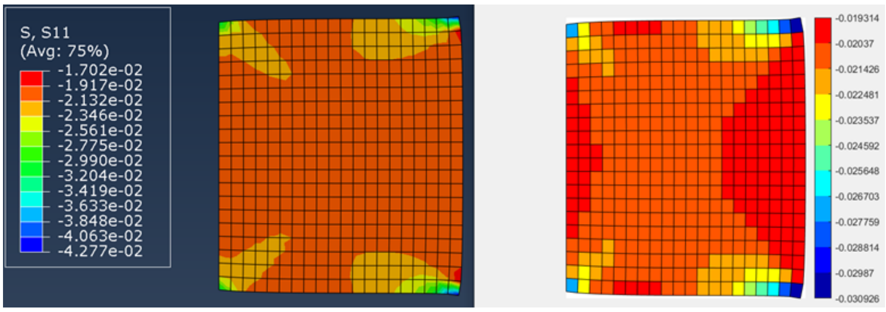


Figure C.10: Contour plot of σ_{11} (400 elements; ABAQUS (L), MATLAB (R))

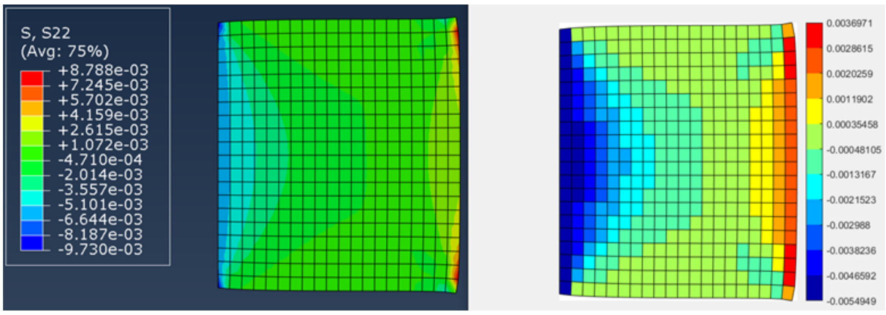
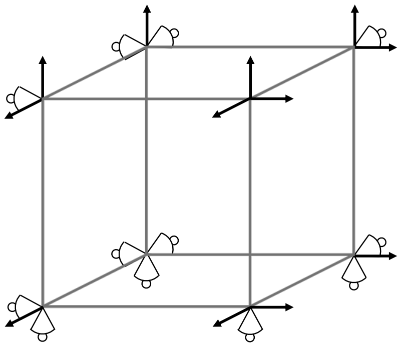


Figure C.11: Contour plot of σ_{22} (400 elements; ABAQUS (L), MATLAB (R))

C.3 Three-dimensional element formulation validations

In addition to the two-dimensional element formulation validations as discussed in Section 3.1.2 and the additional ones in Appendix C.1, the three-dimensional element formulations as presented in Section 2.2.2 by means of of numerical integration are validated here as well. Similar to the two-dimensional FEA, linear-elastic analysis are performed from which the results are compared to the commercial finite element software ABAQUS. To this end, two particularly simple three-dimensional structures are considered. The first structure is composed of one element and has three-dimensional homogeneous boundary conditions as depicted including its specifics in Figure C.12 with its results listed in Table C.4. The second structure has two elements and uniform boundary conditions as shown in Figure C.13 along with its specifics and the results contained in Table C.5.

Again, identical to the two-dimensional findings, Tables C.4 and C.5 show that the developed `MATLAB` FEM code obtains similar results as ABAQUS for the three-dimensional structures. Please note once more that the omitted results are either zero or below machine precision.

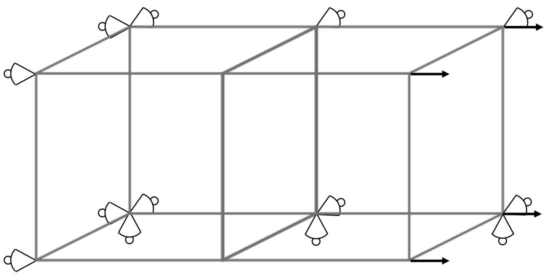


Three-dimensional	**homogeneous**
Structure size	1×1×1 mm
Applied load	1E-03 N
Element type	C3D8
Mesh	1×1×1
Young's modulus	1 MPa
Poisson's ratio	0.3

Figure C.12: Structure with three-dimensional homogeneous boundary conditions

Table C.4: Results three-dimensional homogeneous boundary conditions

`MATLAB`	**Results**	**Units**	**ABAQUS**	**Results**	**Units**
u_2	0.0016	mm	u_2	0.0016	mm
u_7	0.0016	mm	u_7	0.0016	mm
u_8	0.0016	mm	u_8	0.0016	mm
u_{10}	0.0016	mm	u_{10}	0.0016	mm
u_{14}	0.0016	mm	u_{14}	0.0016	mm
u_{15}	0.0016	mm	u_{15}	0.0016	mm
u_{18}	0.0016	mm	u_{18}	0.0016	mm
u_{19}	0.0016	mm	u_{19}	0.0016	mm
u_{20}	0.0016	mm	u_{20}	0.0016	mm
u_{21}	0.0016	mm	u_{21}	0.0016	mm
u_{22}	0.0016	mm	u_{22}	0.0016	mm
u_{24}	0.0016	mm	u_{24}	0.0016	mm
ε_{11}	0.0016	-	E11	0.0016	-
ε_{22}	0.0016	-	E22	0.0016	-
ε_{33}	0.0016	-	E33	0.0016	-
σ_{11}	0.004	MPa	S11	0.004	MPa
σ_{22}	0.004	MPa	S22	0.004	MPa
σ_{33}	0.004	MPa	S33	0.004	MPa



Three-dimensional uniform	
Structure size	2×1×1 mm
Applied load	1E-03 N
Element type	C3D8
Mesh	2×1×1
Young's modulus	1 MPa
Poisson's ratio	0.3

Figure C.13: Structure with three-dimensional uniform boundary conditions

Table C.5: Results three-dimensional uniform boundary conditions

MATLAB	**Results**	**Units**	**ABAQUS**	**Results**	**Units**
u_2	-0.0012	mm	u_2	-0.0012	mm
u_3	0.0012	mm	u_3	0.0012	mm
u_6	0.0012	mm	u_6	0.0012	mm
u_7	0.004	mm	u_7	0.004	mm
u_8	-0.0012	mm	u_8	-0.0012	mm
u_9	0.0012	mm	u_9	0.0012	mm
u_{10}	0.004	mm	u_{10}	0.004	mm
u_{12}	0.0012	mm	u_{12}	0.0012	mm
u_{13}	0.008	mm	u_{13}	0.008	mm
u_{14}	-0.0012	mm	u_{14}	-0.0012	mm
u_{15}	0.0012	mm	u_{15}	0.0012	mm
u_{16}	0.008	mm	u_{16}	0.008	mm
u_{18}	0.0012	mm	u_{18}	0.0012	mm
u_{20}	-0.0012	mm	u_{20}	-0.0012	mm
u_{25}	0.004	mm	u_{25}	0.004	mm
u_{26}	-0.0012	mm	u_{26}	-0.0012	mm
u_{28}	0.004	mm	u_{28}	0.004	mm
u_{31}	0.008	mm	u_{31}	0.008	mm
u_{32}	-0.0012	mm	u_{32}	-0.0012	mm
u_{34}	0.008	mm	u_{34}	0.008	mm
Element 1					
ε_{11}	0.004	-	E11	0.004	-
ε_{22}	-0.0012	-	E22	-0.0012	-
ε_{33}	-0.0012	-	E33	-0.0012	-
σ_{11}	0.004	MPa	S11	0.004	MPa
Element 2					
ε_{11}	0.004	-	E11	0.004	-
ε_{22}	-0.0012	-	E22	-0.0012	-
ε_{33}	-0.0012	-	E33	-0.0012	-
σ_{11}	0.004	MPa	S11	0.004	MPa

Appendix D ABAQUS Validation Studies

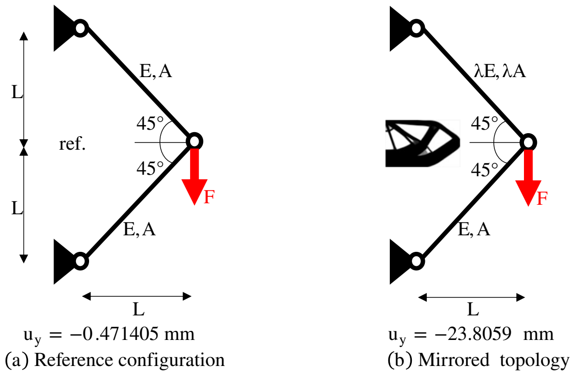
This appendix presents two different validation studies that have been performed using the commercial FEM software ABAQUS. The first study has been used to validate the optimal topology of the cantilever beam problem for asymmetric stiffness. The second study concerns the validation of the observation that the compliance values for symmetric and asymmetric stiffness are equal for the four-bar truss problem.

D.1 Cantilever Beam Problem

In order to validate the optimal topology of the cantilever beam problem for $\lambda = 0.1$ (see Figure 4.1), a simple comparison study with the commercial software ABAQUS has been performed. The reason for doing so are the results presented in the publication by H. Jia et al. [9] on "Optimal structural topology of material with micro-scale tension-compression asymmetry simulated using granular micromechanics", in which the GMA and tension-compression asymmetric intergranular force-law are used in the topology optimization problem for minimization of structural strain energy, identical to the present research. These results show to be different than the results found in Section 4.1 and are in fact opposite in terms of optimal topologies (i.e. having more compressive members than there are tensile members). However, it should be noted that the results in [9] have been obtained though Evolutionary Structure Optimization (ESO) method which is different from the methods used here. The validation study is performed by comparing the vertical displacements of schematic representations of the optimal topologies found from the cantilever beam problem for $\lambda = 0.1$ in Section 4.1 (i.e. Figure 4.1) and its corresponding mirrored version representing the results in [9]. First a statically determinate truss structure is considered, followed by two beam structures of increasing statically indeterminacy. Each structure is first presented for a reference configuration having symmetric input, see Figures (a). Next, the mirrored representations are modelled with lower stiffness and cross-sectional area in the tensile member, see Figures (b). Lastly, the optimized topologies found in this research are represented by having lower stiffness in the tensile member while having a lower cross-sectional area in the compressive member, see Figures (c).

- Statically determinate truss structure

  The first structure is a statically determinate truss having only hinged connections and hinged supports, see Figure D.1. This structure is modelled in ABAQUS by two T2D2 truss elements (one for each truss) using the following input: $L = 1$ m, $E = 30000$ MPa, $\nu = 0.2$, $A = 10000$ mm^2, $F = 100$ kN and $\lambda = 0.1$.



$u_y = -0.471405$ mm
(a) Reference configuration

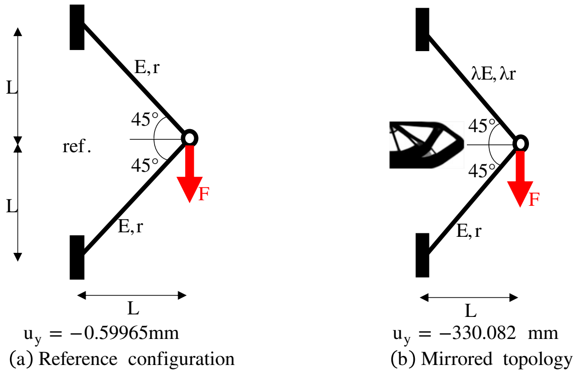
$u_y = -23.8059$ mm
(b) Mirrored topology

$u_y = -4.71405$mm
(c) Optimal topology

Figure D.1: Statically determinate truss structure representation of cantilever

- Statically indeterminate beam structure 1

  This statically indeterminate beam structure has fully fixed supports and a hinged middle connection, see Figure D.2. The structure is modelled in ABAQUS by 20 B21 beam elements (10 for each beam) using the following input: $L = 1$ m, $E = 30000$ MPa, $\nu = 0.2$, $r = 50$ mm, $F = 100$ kN and $\lambda = 0.1$. Assuming a solid circular cross-section, the cross-sectional area follows from $A = \pi r^2$.



$u_y = -0.59965$mm
(a) Reference configuration

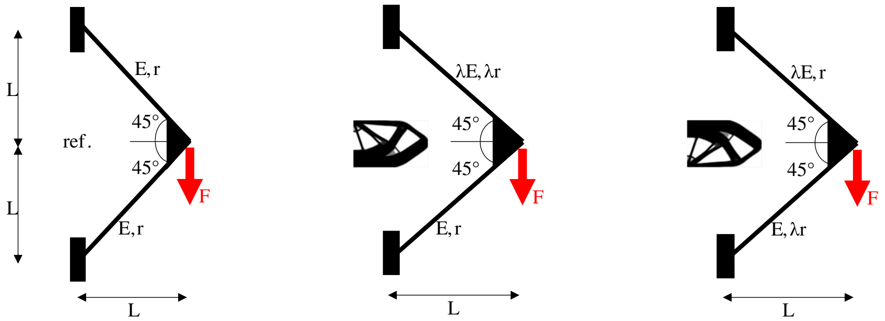
$u_y = -330.082$ mm
(b) Mirrored topology

$u_y = -3.26729$mm
(c) Optimal topology

Figure D.2: Statically indeterminate truss structure representation of cantilever

- Statically indeterminate beam structure 2

  The next statically indeterminate beam structure only has fully fixed connections and supports, see Figure D.3. Similarly, this structure is also modelled in ABAQUS by 20 B21 beam elements (10 for each beam) using the following input: $L = 1$ m, $E = 30000$ MPa, $\nu = 0.2$, $r = 50$ mm, $F = 100$ kN and $\lambda = 0.1$. Assuming a solid circular cross-section, the cross-sectional area follows from $A = \pi r^2$.



$u_y = -0.597984$ mm
(a) Reference configuration

$u_y = -330.003$ mm
(b) Mirroredtopology

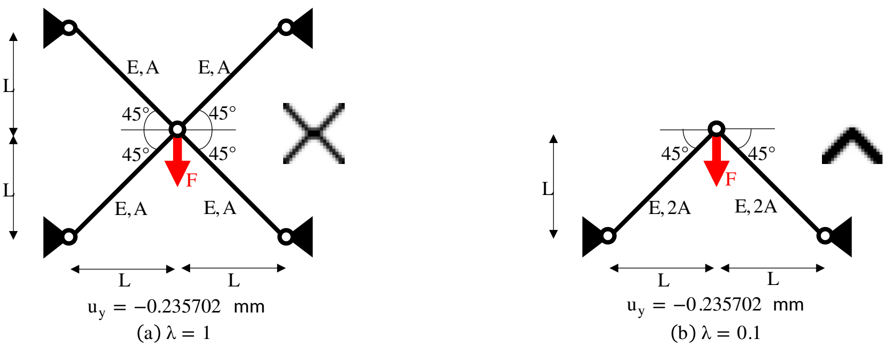
$u_y = -3.19329$ mm
(c) Optimal topology

Figure D.3: Statically indeterminate truss structure representation of cantilever

From Figures D.1 to D.3 it can be concluded that the optimized topologies found in Section 4.1 are indeed more optimal in terms of compliance minimization. The vertical displacements of the loaded nodes are in all cases smaller for the optimal topology representations (Figures (b)) compared to the mirrored topologies (Figures (c)). As a result, smaller displacements yield lower strain energy and thus compliance.

D.2 Four-Bar Truss Problem

The observation of the four-bar truss problem that the compliance values for $\lambda = 1$ and $\lambda = 0.1$ are the same is also validated by means of a simple comparison study using the commercial software ABAQUS. This validation study is also performed by comparing the vertical displacements of schematic representations of the optimal topologies found in Section 4.1 for $\lambda = 1$ and $\lambda = 0.1$, see Figure 4.7. Both structures are represented as statically determinate trusses having only hinged connections and hinged supports, see Figure D.4. Figure D.4(a) shows the typical X-shaped optimal topology for symmetric material behaviour, i.e. $\lambda = 1$, that is modelled in ABAQUS by four T2D2 truss elements (one for each truss). In this case, the total material amount is evenly distributed over the four trusses having cross-sectional area A and the stiffness of both tensile and compressive members is equal: E. On the other hand, Figure D.4(b) depicts the fully compressive optimal topology that is found for $\lambda = 0.1$ which is modelled in ABAQUS by two T2D2 truss elements (one for each truss). Now the total material amount is evenly distributed over only two trusses, each having a doubled cross-sectional area compared to the previous case, hence $2A$. Both trusses still have the same stiffness, E, because λ does not influence the compressive microscopic stiffness. The following input has been used: $L = 1$ m, $E = 30000$ MPa, $\nu = 0.2$, $A = 10000$ mm^2 and $F = 100$ kN.



$u_y = -0.235702$ mm
(a) $\lambda = 1$

$u_y = -0.235702$ mm
(b) $\lambda = 0.1$

Figure D.4: Four-bar truss problem representations for (a) $\lambda = 1$ and (b) $\lambda = 0.1$

Figure D.4 shows that the vertical displacements of the loaded nodes in both cases yield the same amount. As a result, it can be concluded that the compliance values for $\lambda = 1$ and $\lambda = 0.1$ are indeed the same since equal displacements result in equal strain energy and therefore compliance.

Appendix E Nodal Analysis

The results of the two-bar truss problem can be explained and validated through nodal analysis of the loaded node. For symmetric material behaviour, the classical result is known to be symmetric about the vertical axis having equally sized tensile and compressive members that both make angles of 45° with respect to the horizontal. However, for unequal (microscopic) constitutive relationships this symmetry in optimal topology is lost in terms of material distribution and orientation of members (i.e. the angles with respect to the horizontal). For that matter, the sharp angle between the tensile member and the horizontal axis will be referred to as α and the sharp angle between the compressive member and the horizontal axis as β. As a result, static equilibrium at the loaded node in the horizontal and vertical directions therefore yields:

$$\Sigma H = 0 : \quad F - F_t \cos(\alpha) + F_c \cos(\beta) = 0\,, \tag{E.1}$$

$$\Sigma V = 0 : \quad F_t \sin(\alpha) + F_c \sin(\beta) = 0\,, \tag{E.2}$$

where F is the applied horizontal load, F_t the tensile force component and F_c the compressive force component as displayed in Figure E.1 below. From Equation (E.2), F_t can be written in terms of F_c which can then be substituted into Equation (E.1) to work out F_c:

$$F_t = -F_c \left(\frac{\sin(\beta)}{\sin(\alpha)} \right)\,, \tag{E.3}$$

$$F_c = \frac{-F}{\left(\frac{\sin(\beta)}{\sin(\alpha)}\right) \cos(\alpha) + \cos(\beta)}\,. \tag{E.4}$$

From the results using compliance minimization found in this research for the two-bar truss problem, as well as from other sources (e.g. [5,23]), it can be assumed that the internal angle between the tensile and compressive members (at the loaded node) remains approximately 90°. Therefore, β can be expressed in terms of α as $\beta = 180^\circ - 90^\circ - \alpha = 90^\circ - \alpha$. For the considered design domain, having 80×20 elements, the two limit cases in terms of magnitudes for angles α and β are given at the bottom of Figure E.1. These yield $\alpha_{min} = \beta_{min} = \arctan(1/2) \approx 27^\circ$ and $\alpha_{max} = \beta_{max} = \arctan(2) \approx 63^\circ$. The tensile and compressive force components are evaluated using Equations (E.3) and (E.4), respectively, for all possible combinations of α and β given the boundary conditions and assumption that $\beta = 90^\circ - \alpha$. Therefore, in Figure E.1 the resulting normalized absolute force components are displayed against β for the 31 possible horizontal-to-vertical element ratios (ranging from 1/2 to 2).

The results for the two-bar truss problem from compliance minimization (see Figure 4.8) can be explained and validated using the nodal analysis performed here. In Figure E.1 one can see that by increasing α and thereby decreasing β, the compressive force component increases while the tensile force component decreases from a tensile-to-compressive force component ratio of 1/1 for the symmetric configuration (representing the result for $\lambda = 1$) to a ratio of 1/2 for the asymmetric configuration (representing $\lambda = 0.1$). Moreover, as discussed in Section 4.1

for $\lambda = 0.1$, there are more compressive than tensile members with a tensile-to-compressive area ratio of 2/3. As a result, the (weakest) tensile member is both thicker and shorter because α is increased. In conclusion, the width of the design domain allows the topology optimization algorithm to steer towards a more direct compressive load transfer by decreasing the angle of the compressive member and increasing the angle of the tensile member with respect to the horizontal.

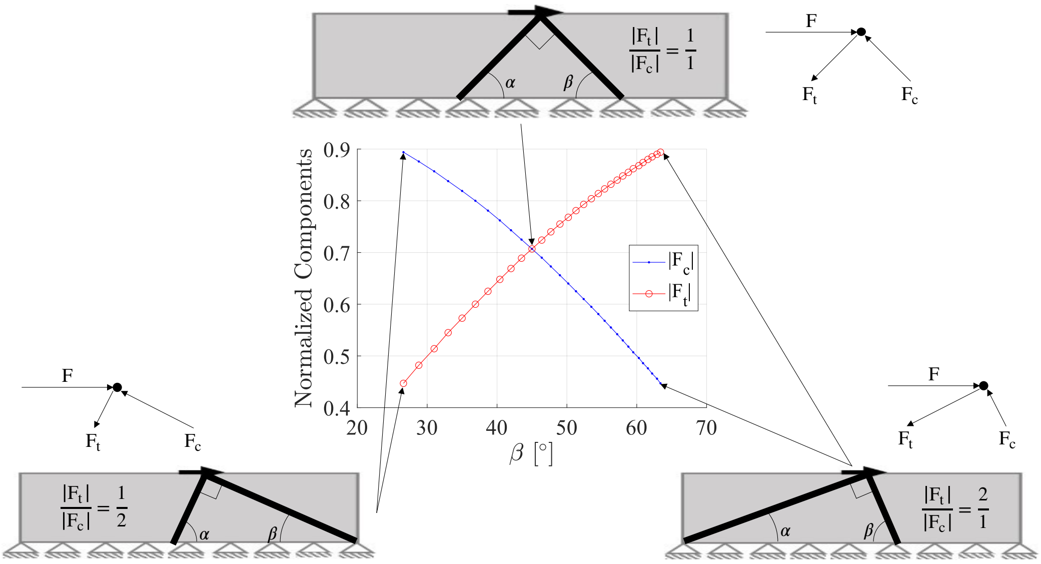


Figure E.1: Normalized absolute nodal force components for different β

The result for the two-bar truss problem from stress-constrained volume minimization (see Figure 4.13) can also be explained using nodal analysis. In this case, the tensile member of the optimized topology has the minimum angle of $\alpha = \alpha_{min} \approx 27°$, while the compressive member has a significantly large angle of about $\beta \approx 81°$. Substitution of the aforementioned angles into Equations (E.3) and (E.4) yields normalized tensile and compressive force components of 1.039 and -0.477, respectively. Therefore, by decreasing α to its minimum value and significantly increasing β, the compressive force component is decreased while the tensile force component is increased which leads to a tensile-to-compressive force component ratio of about 2.2/1 for the asymmetric configuration as opposed to a ratio of 1/1 for the classical symmetric configuration. Moreover, as stressed in Section 4.2, the compressive region is smaller than the tensile region with a tensile-to-compressive area ratio of roughly 9.5/1. As a result, the (weakest) tensile member is thickened to limit the normal stresses occurring in the members because α is decreased. The normal stresses can be estimated via $\sigma = F/A$, which leads to a tensile-to-compressive normal stress ratio of about 0.23/1 given the earlier obtained tensile-to-compressive force component and area ratios. Therefore, it can be concluded that although the found topology may not be the most optimal in terms of material distribution and volume minimization, it still manages to have a lower tensile normal stress than compressive normal stress.

Appendix F Additional Material for Future Research

As an extension of the discussion in Section 7, this appendix provides additional information from literature, approaches and preliminary implementation findings that are useful for future research. Most important are the implementation of numerical sensitivity information into stress-constrained topology optimization under asymmetric strength and the incorporation of a material's load-path dependency into the failure criterion by means of stiffness-constrained volume minimization.

F.1 Additional Literature

This section contains additional information from literature for follow-up work. Starting with a literature review on failure including an example of a (classical) failure criterion that accounts for asymmetric strength, followed by literature on the TTO method and how to obtain numerical sensitivity information.

F.1.1 Failure

According to Poorsolhjouy and Misra in [14,26] and references therein, the essential feature of a material system at failure stage is its inability to bear further external loading. This is often characterized by the material's disability to withstand higher stress (i.e. $\dot{\sigma}_{ij} = 0$). Accordingly, a failure criterion may be defined as in [7]:

$$\det\left(C_{ijkl}\right) \overset{!}{=} 0 \rightarrow \lambda_{min}\left(C_{ijkl}\right) = 0\,, \tag{F.1}$$

where $\det\left(C_{ijkl}\right)$ and λ_{min} are the determinant and minimal eigenvalue of the tangential stiffness tensor C_{ijkl} of the material, respectively. However, failure criteria are rarely defined in this format. Failure criteria are mostly defined as a general function of the stress tensor σ_{ij} that acts on the material. In this case, a failure criterion can be written as $\mathbb{F}\left(\sigma_{ij}\right) = 0$. The common shape of failure criteria defined in this format can be visualized as a surface in the three-dimensional principal stress space where the three axes represent the three principal stress components. Nonetheless, failure often happens along planes for many geomaterials, such as concrete. As a result, some classical failure criteria have been defined as functions of the stress tensor's normal and tangential components acting on that specific failure plane. Whereas these failure criteria were initially defined in the three-dimensional stress space, they are commonly transformed into a two-dimensional function in the plane of normal-tangential stress in which failure is defined as a critical combination of normal and tangential stress components (σ and τ, respectively) on a failure plane: $\mathbb{F}\left(\sigma, \tau\right)$. However, this definition of failure usually results in neglecting the intermediate component of principal because the maximum shear stress only depends on the minimum and maximum principal stress components as illustrated in Figure F.1.

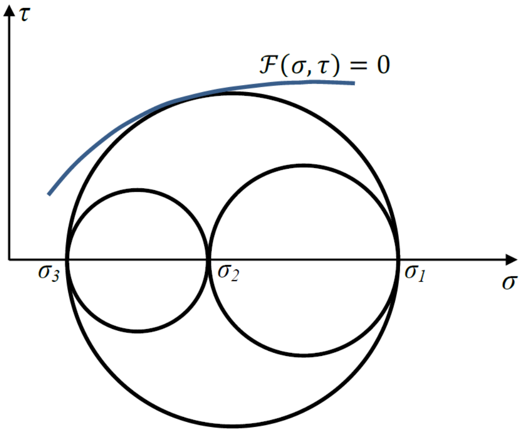


Figure F.1: Failure criteria on normal-tangential stress plane [14]

In fact, true triaxial tests have shown that the failure stress is actually dependent on the intermediate principal stress as opposed to results of failure criteria defined in terms of stress components on the shear failure plane [26]. Thus, failure criteria that ignore the effect of intermediate principal stress are not sufficient to predict failure behaviour. Besides, the results in [14, 26] show that even the most general definition of failure criteria as a function of stress tensor, $\mathbb{F}(\sigma_{ij}) = 0$, are unable to adequately describe a material's failure behaviour. These criteria define failure in terms of the stress state, while it is known that failure is the cumulative effect of loading through which a material transforms from a continuous to an unstable state. As concluded from experiments, a material's response is load-path dependent and therefore a successful failure criterion should consider the stress-path and not only the stress state as done in (classical) failure criteria in terms of the stress tensor. As such, failure criteria need to be devised that change with respect to the loading path.

F.1.2 Drucker-Prager Failure Criterion

One of the simplest available plasticity yield models that is able to manage asymmetric stress limits in tension and compression is the Drucker-Prager criterion, defined in terms of stress invariants [15]. This criterion has been used in a previous master graduation research at the Eindhoven University of Technology (TU/e) on the topic of topology optimization for concrete structures using a linear material model [5, 23]. In what follows, the Drucker-Prager criterion will be briefly reviewed for both two-dimensional as well as three-dimensional applications. This is done to show the form of the yield surface for a failure criterion that is recognized to provide numerical predictions which are in good agreement with experimental results for identifying the failure behaviour of, in particular, concrete. Besides, the criterion will be used for comparison reasons as well as relating this research to the previous master graduation thesis [5]: the topology optimization algorithms used in the present research have emerged from the ones presented in there.

The following description is generally adopted from the publication by Luo & Kang [15], while written down in the format as presented in [5, 23] to prevent inconsistencies in terminology. The Drucker-Prager failure criterion predicts how close a material point is to failure according to its stress state and is the most frequently utilized yield criterion in the field of geotechnical engineering. The failure criterion reads as:

$$\mathbb{F}\left(\sigma_{ij}\right) = \alpha \mathbf{I}_1(\sigma_{ij}) + \sqrt{\mathbf{J}_2(\sigma_{ij})} - \beta \leq 0 \,, \tag{F.2}$$

where α and β are material constants that give the relation between the material's uniaxial tensile- and compressive strength, $\mathbf{I}_1$ is the first invariant of the stress tensor and $\mathbf{J}_2$ the second deviatoric stress invariant. These stress invariants are functions of a material point's stress tensor σ_{ij} and are defined as

$$\begin{aligned} \mathbf{I}_1(\sigma_{ij}) &= \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \,, \\ \mathbf{J}_2(\sigma_{ij}) &= \frac{(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + \sigma_{xy} + \sigma_{yz} + \sigma_{zx}}{6} \,. \end{aligned} \tag{F.3}$$

The failure criterion $\mathbb{F}\left(\sigma_{ij}\right) = 0$ defines the yield surface, which results for the Drucker-Prager criterion into a circular cone in the principal stress space (see Figure F.2). As a matter of fact, if the material constant α is equal to zero, the Drucker-Prager criterion becomes pressure-insensitive for which it reduces to the von Mises criterion. Hence, the criterion can be considered as a modified version of the von Mises criterion by inclusion of an extra term that takes the effect of a hydrostatic stress component on the failure behaviour into account.

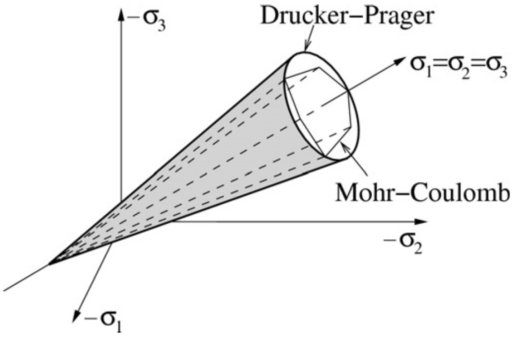


Figure F.2: Drucker-Prager yield surface in the principal stress space [15]

Materials like concrete can be characterized by means of a compressive and tensile strength, f_{cy} and f_{ty}, respectively. For that matter, the material constants α and β can be derived by inserting the two specific principal stress states $(\sigma_1 = \sigma_2 = 0, \sigma_3 = -f_{cy})$ and $(\sigma_1 = f_{ty}, \sigma_2 = \sigma_3 = 0)$ into Equation (F.2) using Equation (F.3), which leads to

$$\alpha = \frac{f_{cy} - f_{ty}}{\sqrt{3}\left(f_{cy} + f_{ty}\right)} \,, \qquad \beta = \frac{2 f_{cy} f_{ty}}{\sqrt{3}\left(f_{cy} + f_{ty}\right)} \,. \tag{F.4}$$

In this circumstance, the two-dimensional Drucker-Prager yield surface can be obtained for a plane stress situation ($\sigma_3 = 0$, i.e. the $\sigma_1\sigma_2$ plane in Figure F.2), which can be seen in Figure F.3. According to [5, 23], a so-called strength ratio $\gamma = f_{cy}/f_{ty}$ can be defined by which the areas of the compressive and tensile zones in the two-dimensional yield surface can be altered. However, as found in that research, the Drucker-Prager criterion may return negative values for bi-axial compression (i.e. $\gamma > 3$) or bi-axial tension (i.e. $\gamma < 1/3$). Therefore, the strength ratios used in that research have been limited by these bounds ($1/3 \leq \gamma \leq 3$) to prevent problems in the topology optimization algorithms.

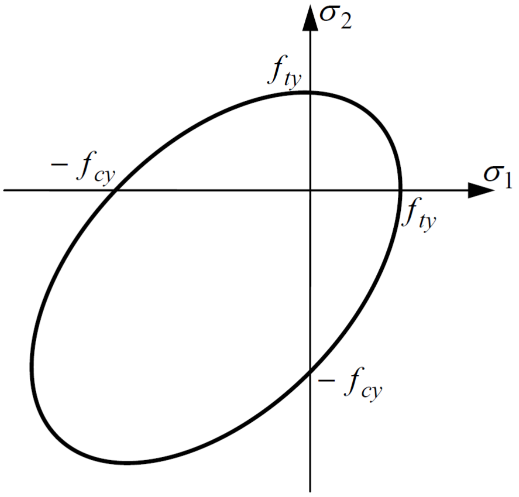


Figure F.3: Drucker-Prager yield surface for a plane stress situation [8]

F.1.3 Traditional Topology Optimization

As described in [5, 23], Traditional Topology Optimization (TTO) solves topology optimization problems by means of mathematical solvers that rely on sensitivity information of the objective- and constraint functions. Good examples of this optimization method are the publicly accessible algorithms by Sigmund [36], Andreassen et al. [33] and Liu & Tovar [6]. In there, the relationship between a linear-elastic material's Young's modulus and the element densities is defined by the Solid Isotropic Material with Penalization (SIMP) method or modified SIMP approach as given in Equations (2.57) and (2.58), respectively. In each iteration of the topology optimization process the material gets redistributed for which a number of methods are available. For example, the compliance minimization problem often utilizes the Optimality Criteria (OC) method as is the case in the aforementioned references. However, this method can only be applied to topology optimization problems having a single constraint, such as the overall compliance. On the other hand, the stress-constrained problem has multiple constraints since all the element stress values may not violate the failure criterion, i.e. $\mathbb{F}(\sigma_{ij}) \leq 0$. For these multiple-constraint topology optimization problems, it is favored to use the Method of Moving Asymptotes (MMA) which is able to deal with multiple constraints. The MMA algorithm utilized in the present research is similar to the one used in a previous research [5, 23] and various other works in the field of topology optimization, which is the `MATLAB` code developed by Svanberg [68]. A detailed description of the MMA algorithm is beyond the scope of this research, but this can be found in the publication by Svanberg himself [68] as well as a brief explanation in [5, 23]. However, it is important to notice that the MMA requires sensitivity information of both the objective and constraint functions.

For the stress-constrained problem (TTOs), this yields the sensitivity of the structural volume with respect to the design variable ($\partial V/\partial x_j$) and the sensitivity of each stress constraint with respect to the element densities ($\partial R_e/\partial x_j$), respectively. The compliance minimization problem (TTOc) requires, besides the sensitivity related to the structural volume ($\partial V/\partial x_j$), the sensitivity of the overall structural compliance with respect to the design variable ($\partial c/\partial x_j$). The sensitivity $\partial V/\partial x_j$ is equal to the elemental volume v_e divided by the total volume V, which holds for all elements because the design domain is divided into a regular FE mesh [5, 23].

After obtaining the *physical densities* through Equation (2.68), the sensitivities of the objective and constraint functions with respect to the *design variables* can be determined by means of the chain rule of differentiation, respectively [5, 23]:

$$\frac{\partial V}{\partial x_j} = \sum_{i=1}^{N_e} \left(\frac{\partial \tilde{x}_i}{\partial x_j} \frac{\partial V}{\partial \tilde{x}_i} \right) = \sum_{i=1}^{N_e} \left(\frac{H_{ji} \frac{\partial V}{\partial \tilde{x}_i}}{\sum_{k=1}^{N_e} H_{ik}} \right) = \sum_{i=1}^{N_e} \left(\frac{H_{ji} \frac{v_i}{V}}{\sum_{k=1}^{N_e} H_{ik}} \right) , \tag{F.5}$$

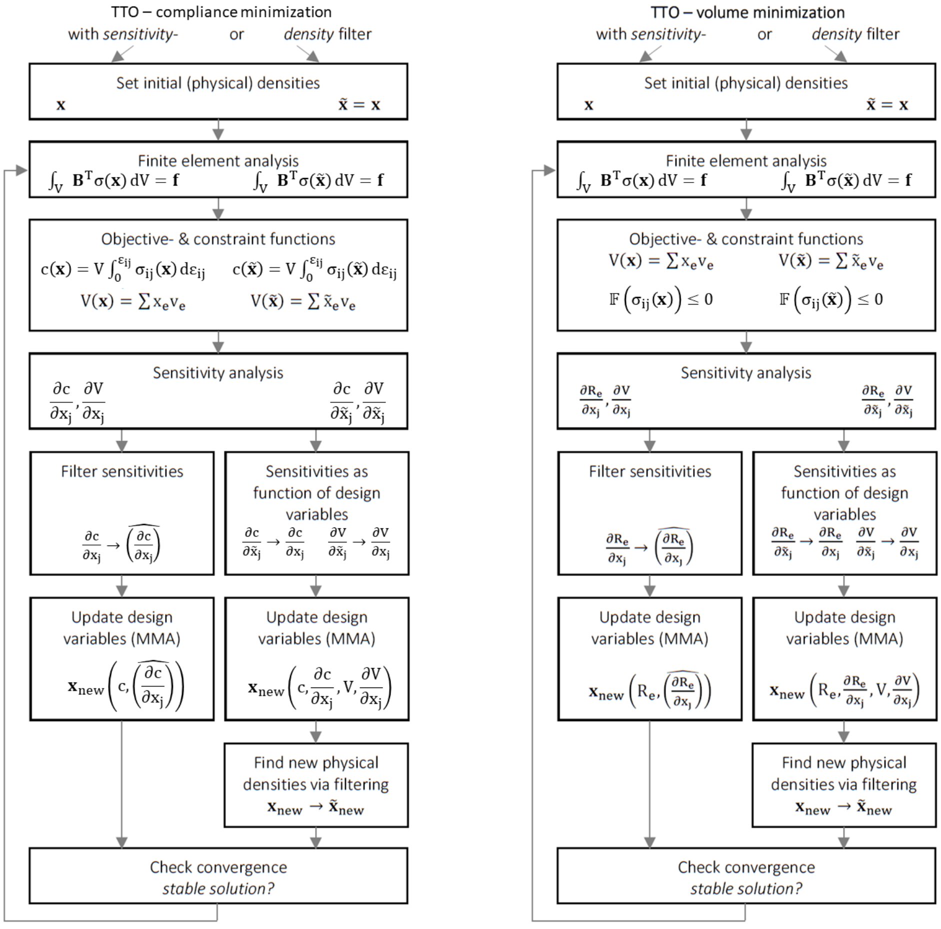
$$\frac{\partial R_e}{\partial x_j} = \sum_{i=1}^{N_e} \left(\frac{\partial \tilde{x}_i}{\partial x_j} \frac{\partial R_e}{\partial \tilde{x}_i} \right) = \sum_{i=1}^{N_e} \left(\frac{H_{ji} \frac{\partial R_e}{\partial \tilde{x}_i}}{\sum_{k=1}^{N_e} H_{ik}} \right) . \tag{F.6}$$

Note that the sensitivities derived in Equations (F.5) and (F.6) describe the sensitivity with respect to the *design variables* instead of the sensitivity with respect to the *physical densities*. As a result, the density filter ensures consistency in the topology optimization algorithm, since the *physical densities* are used for the non-linear FEM algorithm while the *design variables* are utilized for the optimization process. In addition to density filters, there are also sensitivity filters as described in [5, 19, 23, 36]. A sensitivity filter modifies the sensitivity of the objective function with respect to the elemental densities based on the objective function sensitivities of the neighboring elements. For that matter, the sensitivity filter essentially does nothing more than locally averaging the sensitivities within a specified domain, given by the filter radius r_{min}. For the compliance minimization and the stress-constrained problems the sensitivities are modified as follows, respectively:

$$\widehat{\left(\frac{\partial c}{\partial x_j} \right)} = \sum_{i=1}^{N_e} \left(\frac{H_{ji} x_i \frac{\partial c}{\partial x_i}}{x_j \sum_{i=1}^{N_e} H_{ji}} \right) , \tag{F.7}$$

$$\widehat{\left(\frac{\partial R_e}{\partial x_j} \right)} = \sum_{i=1}^{N_e} \left(\frac{H_{ji} x_i \frac{\partial R_e}{\partial x_i}}{x_j \sum_{i=1}^{N_e} H_{ji}} \right) . \tag{F.8}$$

However, as argued in [5,23], sensitivity filtering modifies the sensitivities with respect to the design variables heuristically which are then used in the optimization and material redistribution processes. As a result, not the objective function itself, but a smoothed version is minimized. Therefore, convergence issues may occur for the constraint functions. Accordingly, as argued in [21,22,35], sensitivity filtering is not suitable for stress-constrained topology optimization problems. Therefore, the simulations in the present research are solely using density filters. However, sensitivity filters are included for completeness. In Figure F.4, the solution procedures for (a) the compliance minimization problem (TTOc) as well as (b) the stress-constrained problem (TTOs) using the TTO method are schematically presented, including both filter options.


TTO – compliance minimization
with *sensitivity-* or *density* filter
Set initial (physical) densities
$\mathbf{x}$
$\tilde{\mathbf{x}} = \mathbf{x}$
Finite element analysis
$\int_V \mathbf{B}^T\sigma(\mathbf{x})\,dV = \mathbf{f}$
$\int_V \mathbf{B}^T\sigma(\tilde{\mathbf{x}})\,dV = \mathbf{f}$
Objective- & constraint functions
$c(\mathbf{x}) = V\int_0^{\varepsilon_{ij}}\sigma_{ij}(\mathbf{x})\,d\varepsilon_{ij}$
$c(\tilde{\mathbf{x}}) = V\int_0^{\varepsilon_{ij}}\sigma_{ij}(\tilde{\mathbf{x}})\,d\varepsilon_{ij}$
$V(\mathbf{x}) = \sum x_e v_e$
$V(\tilde{\mathbf{x}}) = \sum \tilde{x}_e v_e$
Sensitivity analysis
$\frac{\partial c}{\partial x_j}, \frac{\partial V}{\partial x_j}$
$\frac{\partial c}{\partial \tilde{x}_j}, \frac{\partial V}{\partial \tilde{x}_j}$
Filter sensitivities
$\frac{\partial c}{\partial x_j} \rightarrow \widehat{\left(\frac{\partial c}{\partial x_j}\right)}$
Sensitivities as function of design variables
$\frac{\partial c}{\partial \tilde{x}_j} \rightarrow \frac{\partial c}{\partial x_j} \quad \frac{\partial V}{\partial \tilde{x}_j} \rightarrow \frac{\partial V}{\partial x_j}$
Update design variables (MMA)
$\mathbf{x}_{new}\left(c, \widehat{\left(\frac{\partial c}{\partial x_j}\right)}\right)$
Update design variables (MMA)
$\mathbf{x}_{new}\left(c, \frac{\partial c}{\partial x_j}, V, \frac{\partial V}{\partial x_j}\right)$
Find new physical densities via filtering
$\mathbf{x}_{new} \rightarrow \tilde{\mathbf{x}}_{new}$
Check convergence
stable solution?
TTO – volume minimization
with *sensitivity-* or *density* filter
Set initial (physical) densities
$\mathbf{x}$
$\tilde{\mathbf{x}} = \mathbf{x}$
Finite element analysis
$\int_V \mathbf{B}^T\sigma(\mathbf{x})\,dV = \mathbf{f}$
$\int_V \mathbf{B}^T\sigma(\tilde{\mathbf{x}})\,dV = \mathbf{f}$
Objective- & constraint functions
$V(\mathbf{x}) = \sum x_e v_e$
$V(\tilde{\mathbf{x}}) = \sum \tilde{x}_e v_e$
$\mathbb{F}\left(\sigma_{ij}(\mathbf{x})\right) \leq 0$
$\mathbb{F}\left(\sigma_{ij}(\tilde{\mathbf{x}})\right) \leq 0$
Sensitivity analysis
$\frac{\partial R_e}{\partial x_j}, \frac{\partial V}{\partial x_j}$
$\frac{\partial R_e}{\partial \tilde{x}_j}, \frac{\partial V}{\partial \tilde{x}_j}$
Filter sensitivities
$\frac{\partial R_e}{\partial x_j} \rightarrow \widehat{\left(\frac{\partial R_e}{\partial x_j}\right)}$
Sensitivities as function of design variables
$\frac{\partial R_e}{\partial \tilde{x}_j} \rightarrow \frac{\partial R_e}{\partial x_j} \quad \frac{\partial V}{\partial \tilde{x}_j} \rightarrow \frac{\partial V}{\partial x_j}$
Update design variables (MMA)
$\mathbf{x}_{new}\left(R_e, \widehat{\left(\frac{\partial R_e}{\partial x_j}\right)}\right)$
Update design variables (MMA)
$\mathbf{x}_{new}\left(R_e, \frac{\partial R_e}{\partial x_j}, V, \frac{\partial V}{\partial x_j}\right)$
Find new physical densities via filtering
$\mathbf{x}_{new} \rightarrow \tilde{\mathbf{x}}_{new}$
Check convergence
stable solution?

(a) Schematic overview of TTOc

(b) Schematic overview of TTOs

Figure F.4: Schematic overview of Traditional Topology Optimization methods

F.1.4 Numerical Sensitivity Information

In view of Section 2.3.2, which initiated by stating that the TTO method was found to be most optimal for materials having unequal stress limits in tension and compression (such as concrete) and ended by pointing out a critical flaw of the non-sensitivity method Proportional Topology Optimization (PTO) for pressure-dependent materials, an alternative approach to obtain sensitivity information has been considered. Because the derivative of the objective function with respect to the design variable cannot be obtained analytically for the multi-scale material model, numerical sensitivity information will be obtained from the GMA and used in the MMA that is able to handle multiple constraint problems using the TTO method. The numerical sensitivity information will be computed through direct perturbation [69] of the vector containing all elemental density values $\mathbf{x}$. From this perturbation the consistent tangent operator related to either the overall compliance or stress constraint functions (depending on the topology optimization problem), referred to as $\partial c/\partial x_j$ and $\partial R_e/\partial x_j$, respectively, are calculated. Hence, the sensitivity information obtained from the multi-scale material model (GMA) can be approximated according to [69–71] as:

$$\frac{\mathrm{d}c}{\mathrm{d}x_j}(\mathbf{x}) \approx \frac{c(\mathbf{x}_p^j) - c(\mathbf{x})}{h}\,, \tag{F.9}$$

$$\frac{\mathrm{d}R_e}{\mathrm{d}x_j}(\mathbf{x}) \approx \frac{R_e(\mathbf{x}_p^j) - R_e(\mathbf{x})}{h}\,, \tag{F.10}$$

where $\mathbf{x}_p^j$ is the density distribution in vectorized form, perturbed on the jth component, i.e. $\mathbf{x}_p^j = \mathbf{x} + h\mathbf{e}_j$, with h the perturbation parameter and $\mathbf{e}_j$ the jth unit vector. Trivially, the number of components in $\mathbf{e}_j$ corresponds to that in $\mathbf{x}$. Consequently, sensitivity information is obtained without the determination of any analytical derivatives. The determination of numerical sensitivity information by means of direct perturbation is considered to be computationally feasible for a reasonable number of elements, since the consistent tangent operator ($\partial c/\partial x_j$ or $\partial R_e/\partial x_j$) should only be obtained once for each element.

The perturbation parameter h has a strong influence on the convergence behaviour of the numerical sensitivities in Equations (F.9) and (F.10). According to [69, 71], the optimal value for h is given as $h_{opt} = \sqrt{\eta}$, where the parameter η represents the accuracy at which c and R_e, in Equations (F.9) and (F.10), respectively, can be evaluated. Additionally, it is argued that a perturbation parameter chosen as $h = 10^{-8}$ corresponds to the case when η approaches machine precision ($\eta \approx 10^{-16}$).

F.2 Additional Approaches

This section presents several additional approaches that have been developed in view of future research. The first part describes the strategy for a preliminary implementation of stiffness-constrained volume minimization, together with a more suitable microscopic constitutive relationship for this specific problem. The second part discusses a preliminary implementation of topology optimization using numerical sensitivity information for both the compliance minimization and stress-constrained problems.

F.2.1 Stiffness-Constrained Volume Minimization

As explained in Appendix F.1.1, failure criteria that ignore the effect of intermediate principal stress and even the most general definition of failure criteria as a function of stress tensor, $\mathbb{F}(\sigma_{ij}) = 0$, are not sufficient and unable to adequately describe and predict a material's failure behaviour. Such criteria define failure in terms of the stress state, while failure is known to be the cumulative effect of loading. As pointed out, experiments have shown that a material's response is load-path dependent. Therefore, successful failure criteria should consider the stress-path in contrast to (classical) failure criteria in terms of the stress tensor where only the stress state is considered. It was concluded that failure criteria need to be devised that change with respect to the loading path. In view of the aforementioned discussion on failure criteria, yet another topology optimization methodology has been proposed during the present research: stiffness-constrained volume minimization, referred to as the stiffness-constrained problem. The aim of this new optimization methodology is to circumvent the deficiencies of failure criteria defined as a function of stress tensor through incorporation of load-path dependency.

This is done according to the definition for a failure criterion given in Equation (F.1), that is related to the tangential stiffness tensor C_{ijkl} of a material. The considered multi-scale material model enables to track the evolution of a material's tangential stiffness tensor during non-linear analysis because the macroscopic stiffness tensor is a function of strain in GMA, as established before. Identical to the procedure for constructing GMA-based failure envelopes (see Section 3.3.2), the macroscopic stiffness tensor determinant for all elements $(\det(C_{ij}))$ can be calculated for each converged non-linear FEM solution and compared to its unloaded state $(\det(C_{ij}^0))$. Nonetheless, it should be noted that the GMA-based failure envelopes are also based on the assumption of linear loading- or stress-paths, which is evident from their construction procedure. This is seldom the case in non-linear FEA. The comparison of the stiffness tensor determinants could be performed again by computing the determinant ratio $\det(C_{ij})/\det(C_{ij}^0)$. However, the PTO method distributes the material proportionally to the constraint function values, as discussed in Section 2.3.2. In this case, the constraints are the elemental stiffness (determinant) values. Using the determinant ratio would result in high constraint function values for elements having a high stiffness (i.e. far away from failure) and vice versa, such that the algorithm will distribute more material to the stronger elements. This is not intended, because it is preferable for elements having lower stiffness (i.e. close to failure) to receive a larger portion of the material amount to prevent failure. Therefore, a failure sensitivity regarding stiffness has been defined that is equivalent to that of stress, having a normalized nature. Similarly, the failure sensitivity varies between 0 and 1 where 0 means fully intact or unloaded conditions and 1 tells failure has occurred. Nevertheless, this postulates that elements having high stiffness yield low failure sensitivities and vice versa. The aforementioned has been achieved by defining the failure sensitivity for the stiffness-constrained problem as follows:

$$\frac{\det(C_{ij}) - \det(C^0_{ij})}{\det(C^F_{ij}) - \det(C^0_{ij})} \,,$$

where $\det(C^F_{ij})$ refers to the determinant of the macroscopic stiffness tensor in the failed state. Similar to the failure sensitivity in Section 3.3.2, the above fraction can be regarded as the division in a normalized failure criterion, such as the first term in Equation (2.67). Therefore, this failure sensitivity also indicates how sensible or close a material point is to failure while considering the stress-path and thus incorporating load-path dependency. The term $\det(C^F_{ij})$ is included for strength-related (microscopic) constitutive relationships for which the stiffness tensor determinant cannot reach zero, such as the bi-linear inter-granular force-law with hardening (see Equations (3.7) to (3.9)). For this kind of micro-scale force-displacement relationships the material is considered to be in the failed state when all grain-pair interactions experience hardening, i.e. the micro-scale stiffness of all contacts are equal to $k^{h\alpha}_i$. For ideal plastic (microscopic) constitutive relations, $\det(C^F_{ij})$ yields zero such that the failure sensitivity for the stiffness-constrained problem simplifies to $1 - \det(C_{ij})/\det(C^0_{ij})$. Accordingly, the stiffness-constrained problem can be described in a similar fashion as the stress-constrained problem, provided the failure criterion is now a function of the stiffness tensor, $\mathbb{F}\left(C_{ij}\right) = 0$, as follows:

$$\begin{cases} \min V(\mathbf{x}) = \sum_{e=1}^{N_e} x_e v_e \,, \\ \text{subject to} \quad \begin{cases} \int_V \mathbf{B}^T \boldsymbol{\sigma} \mathrm{d}V = \mathbf{f}, \\ \mathbb{F}\left(C_{ij}(\mathbf{x})\right) \leq 0 \,, \\ 0 < x_{min} \leq x_e \leq 1 \,, \end{cases} \end{cases} \tag{F.11}$$

where in this case $\mathbb{F}\left(C_{ij}(\mathbf{x})\right)$ is a general yield function or failure criterion (i.e. the constraint) as a function of the stiffness tensor C_{ij} that depends on the density distribution $\mathbf{x}$ for which no failure will occur if it is less or equal to zero. Similar to the other PTO implementations (PTOc and PTOs), the micro-scale penalization technique developed in Section 3.2.1 is also used for the stiffness-constrained problem via Equation (3.1). Please note that for the bi-linear microscopic force-displacement relationship (see Section 3.2.4) the stiffness tensor determinant remains constant if all contacts are within its initial branch (i.e. prior to hardening). As a result, the unloaded condition and loads up to hardening cannot be distinguished based on their stiffness tensor determinants. Therefore, an exponential ideal plastic inter-granular force-law was found to be advantageous for its gradual decrease of macroscopic stiffness tensor providing more deviation in its determinant from the very beginning of the loading process. This particular smoothed microscopic constitutive relationship will be discussed later (see Appendix F.2.2). The same numerical input as for the stress-constrained problem is used here for the GMA, mimicking the material behaviour of C20/25. New methodologies are often prone to problems and difficulties, which is also the case for topology optimization using stiffness-constrained volume minimization. These are discussed next.

The first problem encountered is referred to as the void element or penalization problem. As stressed previously, the stiffness-constrained problem also utilizes the micro-scale penalization technique as given by Equation (3.1). This yields that the macroscopic stiffness tensor C_{ij} of void elements ($x_e = x_{min}$) is reduced significantly, because of the "power-law approach" using a penalization power p that is usually set to a value of 3. As a result, the stiffness tensor determinants of void elements may become smaller than those of failed elements which means that void elements could be considered as less optimal than failed elements. Besides, recall that the failure sensitivity (and thus the constraint function) for the stiffness-constrained problem is defined in terms of the stiffness tensor determinant. This means that if the entries of the $n \times n$ macroscopic stiffness tensor C_{ij} are scaled by a scalar α, the determinant scales by a factor α^n which can be regarded as additional penalization. In the end, the penalization problem leads to the issue that the topology optimization algorithm is unable to distinguish between low stiffness tensor determinants due to (i) void elements and (ii) actual material failure.

The described problem was attempted to be solved by decoupling of the penalization effects. This implied the non-linear FEM to use penalized stiffness, according to Equation (3.1), and the optimization algorithm (i.e. the distribution algorithm) to utilize a non-penalized stiffness measure. This decoupling comprises essentially the same as the difference in *physical densities* (used for the non-linear FEM algorithm) and *design variables* (utilized for the optimization process), as discussed in Section 2.3.2. Nevertheless, the non-penalized stiffness measure circumvents the micro-scale penalization technique by excluding the number density of contacts N_p from the equation for the macroscopic stiffness tensor C_{ijkl} (see Equation (2.21)). In fact, it is simply obtained through dividing the macroscopic stiffness tensor by the number density of contacts, C_{ij}/N_p, and computing the determinant from that. Since the density distribution function $\xi(\theta, \phi)$ reduces to a mere normalization constant for isotropic materials (see Equation (2.22)), such as concrete, and that essentially the only remaining term in Equation (2.21) is $K_{ik}n_jn_l$ (besides the l^2 term), the non-penalized stiffness measure is referred to as the determinant of that term, $\det(K_{ik}n_jn_l)$, for convenience. Through using this non-penalized stiffness measure, the determinants of void elements and actual material failure can be distinguished allowing for an improved determination of new density distributions.

Even though $\det(K_{ik}n_jn_l)$ does not receive penalization, it is still possible that very high, non-negligible strains occur in low density elements (due to their lower stiffness in the non-linear FEM algorithm). Thereby $\det(K_{ik}n_jn_l)$ may decrease significantly in low-density elements, meaning that the topology optimization algorithm wants to increase the element's density (preventing complete removal). This second problem was first attempted to be solved by an additional measure on top of the decoupling of penalization effects, namely strain penalization for the optimization process:

$$\tilde{\varepsilon}_{ij} = \tilde{\varepsilon}_{ij}(x_e) = x_e^p \varepsilon_{ij} \quad \text{with} \quad 0 < x_{min} \leq x_e \leq 1\,, \tag{F.12}$$

where x_e is the density of element e, p is again a penalization power, ε_{ij} the macroscopic strain tensor resulting from the non-linear FEM algorithm and x_{min} a small, non-zero value to prevent singularity of the stiffness matrix (typically 10^{-3}). Similar to other penalization techniques, only elements having intermediate densities are affected. However, the penalization power p in Equation (F.12) is set equal to 1 for having a more relaxed (linear) penalization relationship compared to the usual value of 3. This choice for p follows from a trial and error procedure. The penalized strains are then used as a separate input for another, self-contained run of the GMA algorithm to obtain the element's macroscopic stiffness tensors to be used in the distribution algorithm (determining the new density distribution). Obviously, in the second, self-contained run of the GMA algorithm penalization of the stiffness tensor is excluded (penalization already happened during the first, general routine). Using strain penalization for updating the density distribution thereby excludes the effect of (near) void elements having very high, non-negligible strains in the optimization process that caused the (non-penalized) stiffness measure to decrease significantly. By penalizing strain, the distribution algorithm basically perceives that there is hardly any strain in void elements while there may be (significant) strain in more dense elements. Combining strain penalization with the non-penalized stiffness measure, $\det(K_{ik}n_jn_l)$, allows void elements to be present in a structure during the topology optimization process.

Implementation of both the decoupling of penalization effects (i.e. using the non-penalized stiffness measure) and strain penalization significantly improved the stiffness-constrained volume minimization topology optimization procedure. Nevertheless, it remained subject to (i) convergence issues in the non-linear FEM algorithm and (ii) has a tendency to "smear out" densities (meaning no occurrence of full nor void elements). This first point seems to be related to concentrations of local failure that occur during the first iterations of the topology optimization process. Local failure concentrations initially occur near a structure's boundary conditions (i.e. the load and support conditions) causing a significant discontinuity in the failure criterion as a function of stiffness tensor, $\mathbb{F}(C_{ij}(\mathbf{x}))$. For example, consider a cantilever beam as shown in Figure 3.5 having supports on its left edge and a point load at the center of its right edge. When considering a symmetric material model, initially local failure concentrations will start to occur near the support at the top of the left edge and the bottom of the left edge, and near the center of the right edge adjacent to the point load. While these regions start to experience higher failure sensitivities, it might be that most other elements (especially within the cantilever's center region) are having significant lower failure sensitivities. As a result, most of the material (if not all) will be distributed towards the local failure concentrations leaving the other regions more or less void.

This yields a density distribution composed of more dense element clusters, having significantly higher stiffness, and large (near) void regions, having significantly lower stiffness, causing the non-linear FEM algorithm to fail convergence. The second problem prevents the topology optimization algorithm to obtain actual black-and-white solutions in which fully dense ($x_e = 1$) and fully void ($x_e = x_{min}$) elements are present.

It was later considered that the *singularity phenomenon* occurring in stress-constrained topology optimization (see Section 2.3.2) could also occur for the new stiffness-constrained problem. Therefore, the developed strain penalization from Equation (F.12) may be improved upon by replacing it with the epsilon- or ε-relaxation technique as presented in Equation (2.67). This relaxation technique has been proved to be successful in stress-constrained volume minimization. However, the implementation will be left for follow up work with respect to the reserved time for the current research. Additionally, it could very well be that the PTO method for the stiffness-constrained problem exhibits the same issue (i.e. having a tendency to prefer tensile elements over compressive ones) due to its proportional material distribution algorithm (see the end of Section 2.3.2).

F.2.2 Smoothed Microscopic Constitutive Relationship

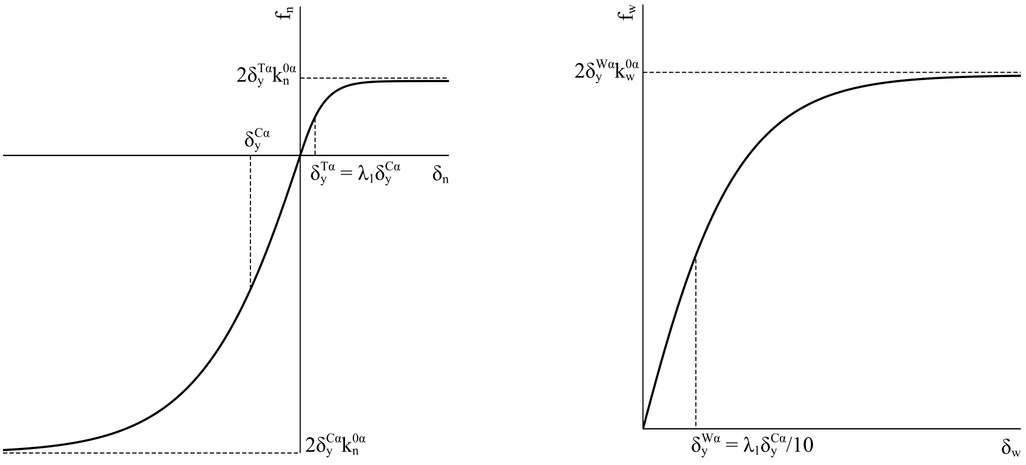
As discussed previously, the bi-linear microscopic force-displacement relationship (see Section 3.2.4) has a stiffness tensor determinant that remains constant if all contacts are within its initial branch (i.e. prior to hardening). As a result, the unloaded condition and loads up to hardening cannot be distinguished based on their stiffness tensor determinants. Therefore, an exponential ideal plastic inter-granular force-law was found to be advantageous for its gradual decrease of macroscopic stiffness tensor providing more deviation in its determinant from the very beginning of the loading process. This particular microscopic constitutive relationship can be regarded as a smoothed version of the bi-linear inter-granular force-law with the hardening ratio set to 0, thus representing ideal plastic material behaviour. Such a microscopic relationship exhibits smoother transitions in the material's macroscopic response (i.e. more deviation and a gradual decrease of its macroscopic stiffness tensor). This may be advantageous for certain reasons that will be explained later. The inter-granular force-law is of an exponential form and is given by:

$$f_{\mathrm{n}}^{\alpha} = \begin{cases} -k_{\mathrm{n}}^{0\alpha}\left(2\delta_y^{T\alpha} + \delta_{\mathrm{n}}^{\alpha}\right) e^{-\frac{\delta_{\mathrm{n}}^{\alpha}}{\delta_y^{T\alpha}}} + 2\delta_y^{T\alpha} k_{\mathrm{n}}^{0\alpha} & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} \geq 0 \quad (\text{tension})\,, \\ -k_{\mathrm{n}}^{0\alpha}\left(2\delta_y^{C\alpha} + \delta_{\mathrm{n}}^{\alpha}\right) e^{-\frac{\delta_{\mathrm{n}}^{\alpha}}{\delta_y^{C\alpha}}} + 2\delta_y^{C\alpha} k_{\mathrm{n}}^{0\alpha} & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} < 0 \quad (\text{compression})\,, \end{cases}$$

$$f_{\mathrm{w}}^{\alpha} = -k_{\mathrm{w}}^{0\alpha}\left(2\delta_y^{W\alpha} + \delta_{\mathrm{w}}^{\alpha}\right) e^{-\frac{\delta_{\mathrm{w}}^{\alpha}}{\delta_y^{W\alpha}}} + 2\delta_y^{W\alpha} k_{\mathrm{w}}^{0\alpha}\,. \tag{F.13}$$

Analogous to the bi-linear inter-granular force-displacement relationship (see Section 3.2.4), the same nomenclature applies as for Equation (3.7) and the different relative displacements at yielding are again given by Equation (3.8). However, in this force-law only the yielding points can be altered through λ_1. The microscopic constitutive relationship is depicted in Figure F.5. Identical to the bi-linear microscopic constitutive relationship, the inter-granular stiffness can be derived through direct differentiation of Equation (F.13):

$$
\begin{gathered}
k_{\mathrm{n}}^{\alpha} = \begin{cases} k_{\mathrm{n}}^{0\alpha}\left(1+\frac{\delta_{\mathrm{n}}^{\alpha}}{\delta_y^{T\alpha}}\right) e^{-\frac{\delta_{\mathrm{n}}^{\alpha}}{\delta_y^{T\alpha}}} & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} \geq 0 \quad (\text{tension})\,, \\ k_{\mathrm{n}}^{0\alpha}\left(1+\frac{\delta_{\mathrm{n}}^{\alpha}}{\delta_y^{C\alpha}}\right) e^{-\frac{\delta_{\mathrm{n}}^{\alpha}}{\delta_y^{C\alpha}}} & \text{if} \quad \delta_{\mathrm{n}}^{\alpha} < 0 \quad (\text{compression})\,, \end{cases} \\
k_{\mathrm{w}}^{\alpha} = k_{\mathrm{w}}^{0\alpha}\left(1+\frac{\delta_{\mathrm{w}}^{\alpha}}{\delta_y^{W\alpha}}\right) e^{-\frac{\delta_{\mathrm{w}}^{\alpha}}{\delta_y^{W\alpha}}}\,.
\end{gathered}
\tag{F.14}
$$



(a) Normal directions

(b) Tangential directions

Figure F.5: Exponential ideal plastic inter-granular force-law

To illustrate the smoother transitions in the material's macroscopic response (i.e. more deviation and a gradual decrease of its macroscopic stiffness tensor), the two-dimensional GMA-based failure envelope corresponding to the exponential ideal plastic (Equations (F.13) and (F.14)), is displayed below. Identical to the bi-linear microscopic constitutive relationship, the evolution of the determinant of the macroscopic stiffness tensor is plotted against the stress increments for five different directions in two-dimensional principal stress space as indicated in the figure. The resulting GMA-based failure envelope has been obtained for $\lambda_1 = 0.01$ (see Figure F.6). Similarly, the failure envelope is discretized into 72 directions (i.e. every 5 degrees) and normalized with respect to their corresponding uniaxial compressive strength, f_c. The corresponding three-dimensional GMA-based failure envelope that is obtained using the same numerical criterion can be found in Figure F.7.

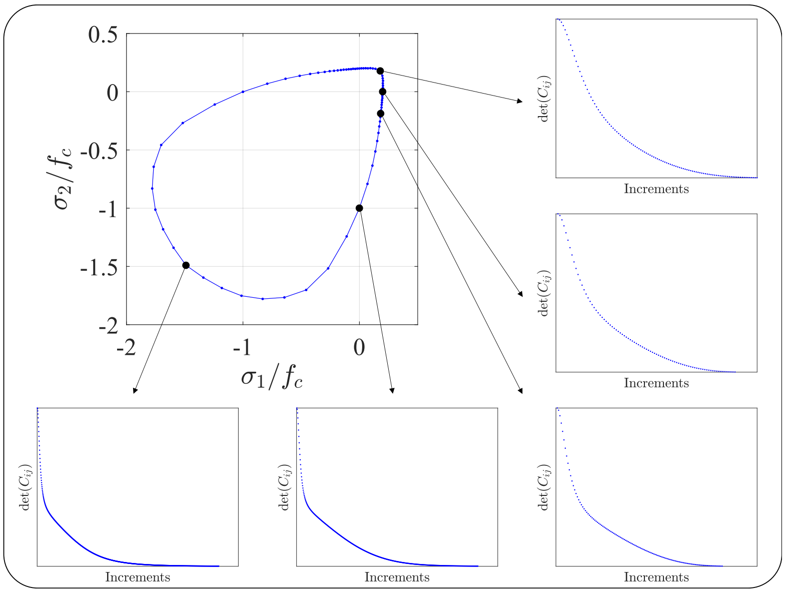
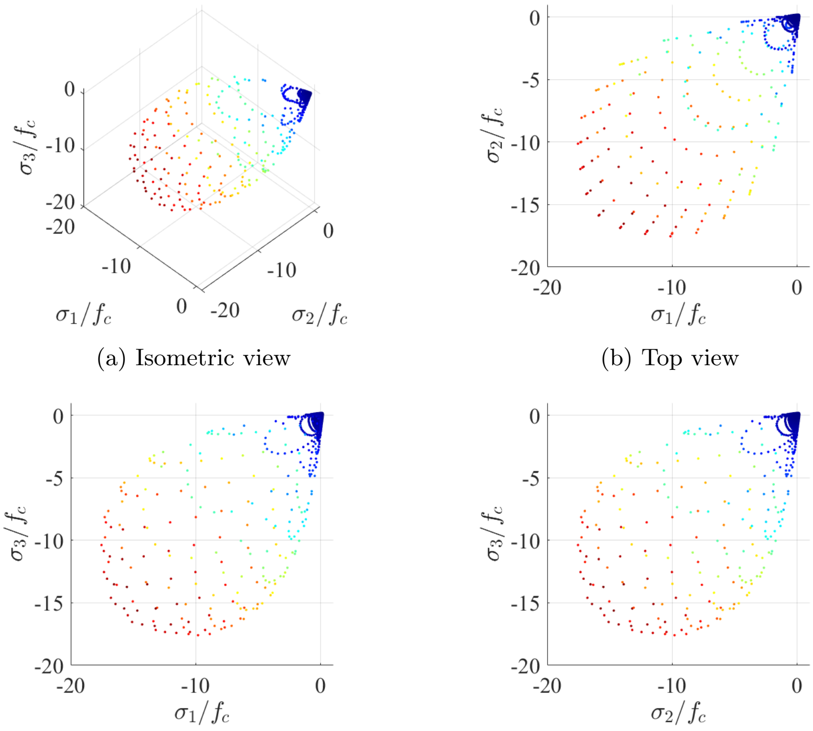


Figure F.6: GMA-based failure envelope from exponential inter-granular force-law



(a) Isometric view

(b) Top view

(c) Front view

(d) Side view

Figure F.7: GMA-based failure envelope from exponential inter-granular force-law

F.2.3 Traditional Topology Optimization & Numerical Sensitivity

In view of Section 2.3.2, in which it was concluded from [5, 23] that the TTO method is recommended for stress-constrained volume minimization of materials with unequal stress limits in tension and compression, this method has been considered here too. However, it has also been argued that TTO utilizes the MMA for multiple-constraint problems which requires sensitivity information of both the objective and constraint functions. Therefore, Appendix F.1.4 introduced the notion of obtaining numerical sensitivity information from the GMA to be used in the MMA, because analytical sensitivity information cannot be obtained for the multi-scale material model. The inclusion of (numerical) sensitivity information significantly broadens the implementation possibilities of the developed non-linear FEM using the multi-scale material model in the field of topology optimization. Besides, utilization of the TTO method eliminates the critical flaw of the aforementioned non-sensitivity PTO method as elaborated at the end of Section 2.3.2.

Identical to Section 3.3, several topology optimization methodologies are treated successively. This time starting with the stress-constrained problem and then continuing with the compliance minimization problem, which is included here for completeness, using the TTO method.

- Traditional topology optimization using stress

  Departing from the PTO methods using compliance and stress in Section 3.3, the TTO implementations started with the stress-constrained problem in view of a previous research's [5, 23] focus on this optimization methodology. Similar to the implementation of PTOs, TTOs aims to minimize the structural volume while satisfying a certain stress constraint (Equation (2.60)) using the micro-scale penalization technique developed in Section 3.2.1 (Equation (3.1)).

  Even though the four benchmark examples presented in Figure 3.7 are already limited in size, the considered benchmarks for the TTO method are restricted even more in view of the additional computational demand of the perturbations for obtaining the numerical sensitivity information on top of the non-linear FEM. To this end, the TTO method only considers the single rod benchmark (Figure 3.7(a)). Similarly, because this benchmark example has been reported for linear-elastic FEA with the stress-constrained problem using the TTO method (TTOs) in [5, 23], it is still possible to compare the result obtained here. Besides, the benchmark example allows for quick visual inspection of the occurrence of either tensile or compressive elements. The required numerical sensitivity information of the constraint functions ($\partial R_e / \partial x_j$) will be obtained via Equation (F.10). However, in order to implement the numerical differentiation method as discussed in Appendix F.1.4, one should be able to determine the constraint function (i.e. the failure criterion) values resulting from direct perturbation of the non-linear FEM. This will be elaborated next.

Inspection of Equation (F.10) shows that it actually requires only two numerical inputs to be determined, namely the stress constraint function based on the perturbed density distribution on the jth component, $R_e(\mathbf{x}_p^j)$, and the stress constraint function resulting from the non-perturbed non-linear FEM solution, $R_e(\mathbf{x})$. The latter value can be obtained and stored directly from the non-linear FEM solution. However, the stress constraint function values based on the perturbed density distributions require more effort. To this end, the last converged non-linear FEM state, $(\mathbf{u}(\mathbf{x}), \mathbf{F})$, density distribution, $\mathbf{x}$, constraint function values, $R_e(\mathbf{x})$, and *global* tangential stiffness matrix, $\mathbf{K}(\mathbf{x})$, are stored. The density distribution is perturbed for each element according to $\mathbf{x}_p^j = \mathbf{x} + h\mathbf{e}_j$, where the jth component corresponds to the element number. For each perturbed density distribution a new displacement correction $\mathrm{d}\mathbf{u}(\mathbf{x}_p^j)$ is determined through Equation (2.40), which is evaluated for non-linear material behaviour as follows:

$$\mathrm{d}\mathbf{u}(\mathbf{x}_p^j) = -\left(\mathbf{K}(\mathbf{x}_p^j)\right)^{-1}\left\{\mathbf{f}(\mathbf{x}_p^j) - \mathbf{F}\right\}, \tag{F.15}$$

where $\mathbf{K}(\mathbf{x}_p^j)$ is the perturbed tangential stiffness matrix and $\mathbf{f}(\mathbf{x}_p^j)$ the perturbed internal force vector, both functions of the the perturbed density distributions, $\mathbf{x}_p^j$. The perturbed terms in Equation (F.15) are given by:

$$\begin{aligned}\mathbf{K}(\mathbf{x}_p^j) &= \int_V \mathbf{B}^T\mathbf{C}(\mathbf{x}_p^j)\mathbf{B}\mathrm{d}V\\ &= \int_V \mathbf{B}^T\left(l^2N_p(\mathbf{x}_p^j)\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi}\left(K_{ik}n_jn_l\right)\xi(\theta,\phi)\sin\theta\mathrm{d}\phi\mathrm{d}\theta\right)\mathbf{B}\mathrm{d}V\,,\end{aligned} \tag{F.16}$$

$$\begin{aligned}\mathbf{f}(\mathbf{x}_p^j) &= \int_V \mathbf{B}^T\boldsymbol{\sigma}(\mathbf{x}_p^j)\mathrm{d}V\\ &= \int_V \mathbf{B}^T\left(lN_p(\mathbf{x}_p^j)\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi}f_i(\theta,\phi)\xi(\theta,\phi)\sin\theta\mathrm{d}\phi\mathrm{d}\theta\right)\mathrm{d}V\,.\end{aligned} \tag{F.17}$$

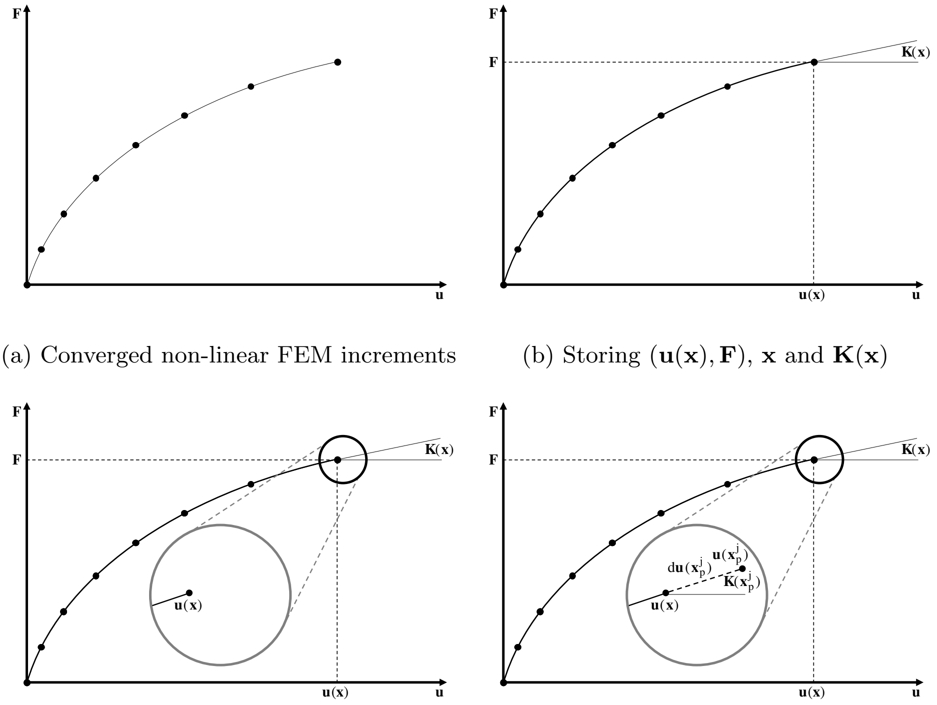
Please note that even though Equations (2.40) and (F.15) are in fact part of the Newton-Raphson iterative method, no iterations are performed for the perturbed displacement correction. The aim of Equations (F.15) to (F.17) is to find the perturbed displacement vectors, i.e. $\mathbf{u}(\mathbf{x}_p^j) = \mathbf{u}(\mathbf{x}) + \mathrm{d}\mathbf{u}(\mathbf{x}_p^j)$, that allow for determination of the stress constraint function values based on the perturbed density distributions, $R_e(\mathbf{x}_p^j)$, for computing numerical sensitivity information via Equation (F.10). In this case, the perturbed stress constraint function values follow from the relaxation of the failure criterion (Equation (2.67)), that on its turn followed from the perturbed stress and strain states which are calculated from the perturbed displacement vectors: $\mathbf{u}(\mathbf{x}_p^j) \to \boldsymbol{\varepsilon}(\mathbf{x}_p^j) \to \boldsymbol{\sigma}(\mathbf{x}_p^j) \to \mathbb{F}(\mathbf{x}_p^j) \to R_e(\mathbf{x}_p^j)$. Please note that it is required to compute an entire new displacement correction for each perturbation, because the MMA requires an $m \times n$ matrix containing derivatives of each stress constraint with respect to all density perturbations. As mentioned in Appendix F.1.4 this is considered to be computationally feasible for a reasonable number of elements, since the displacement correction should only be obtained once for each element. Once all the perturbations have been performed, the derived numerical sensitivity information to be used in the TTO method for stress-constrained volume minimization is presented in the following format:

$$\frac{\partial R_e}{\partial x_j} \approx \begin{bmatrix} \frac{\mathrm{d}R_1}{\mathrm{d}x_1} & \frac{\mathrm{d}R_1}{\mathrm{d}x_2} & \cdots & \frac{\mathrm{d}R_1}{\mathrm{d}x_n} \\ \frac{\mathrm{d}R_2}{\mathrm{d}x_1} & \frac{\mathrm{d}R_2}{\mathrm{d}x_2} & \cdots & \frac{\mathrm{d}R_2}{\mathrm{d}x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathrm{d}R_m}{\mathrm{d}x_1} & \frac{\mathrm{d}R_m}{\mathrm{d}x_2} & \cdots & \frac{\mathrm{d}R_m}{\mathrm{d}x_n} \end{bmatrix}. \tag{F.18}$$

To better understand the above procedure, Figure F.8 will illustrate it step by step. Figure F.8(a) shows a number of converged non-linear FEM increments and its resulting force-displacement curve. Accordingly, the last converged state, $(\mathbf{u}(\mathbf{x}), \mathbf{F})$, including its corresponding density distribution, $\mathbf{x}$, and *global* tangential stiffness matrix, $\mathbf{K}(\mathbf{x})$, are stored as shown in Figure F.8(b). Now let us zoom in on the last converged non-linear FEM state, having a non-perturbed displacement vector $\mathbf{u}(\mathbf{x})$ as depicted in Figure F.8(c). This state will be perturbed by means of $\mathbf{x}_p^j$ in order to find new displacement corrections $\mathrm{d}\mathbf{u}(\mathbf{x}_p^j)$ as a result of the perturbed tangential stiffness matrix $\mathbf{K}(\mathbf{x}_p^j)$. Subsequently, Figure F.8(d) illustrates the perturbed displacement vectors, $\mathbf{u}(\mathbf{x}_p^j)$, that are calculated for determination of the stress constraint function values based on the perturbed density distributions, $R_e(\mathbf{x}_p^j)$. The procedure for the stress-constrained problem is given in Algorithm 6.

Algorithm 6 Determination of numerical sensitivity information

```
1. Initialization
   (a) Obtain stress constraint function values resulting from the non-perturbed non-linear FEM
       solution using Equation (2.67)
   (b) Store last converged non-linear FEM state, $(\mathbf{u}(\mathbf{x}), \mathbf{F})$, density distribution, $\mathbf{x}$, constraint
       function values, $R_e(\mathbf{x})$, and *global* tangential stiffness matrix, $\mathbf{K}(\mathbf{x})$
   (c) Compute perturbed density distributions according to $\mathbf{x}_p^j = \mathbf{x} + h\mathbf{e}_j$
2. Next perturbation $j \geq 1$
   (a) Compute perturbed tangential stiffness matrix $\mathbf{K}(\mathbf{x}_p^j)$ and perturbed internal force vector
       $\mathbf{f}(\mathbf{x}_p^j)$ based on current perturbation via Equations (F.16) and (F.17), respectively
   (b) Find perturbed displacement vector $\mathbf{u}(\mathbf{x}_p^j) = \mathbf{u}(\mathbf{x}) + \mathrm{d}\mathbf{u}(\mathbf{x}_p^j)$ from perturbed displacement
       correction given by Equation (F.15)
   (c) Determine stress constraint function values using Equation (2.67)
   (d) Compute numerical sensitivity information through Equation (F.10)
   (e) Go to next perturbation 2
3. Construct matrix containing derivatives of each stress constraint with respect to all density per-
   turbations (Equation (F.18))
```


F
u
F
F
K(x)
u(x)
u
F
F
K(x)
u(x)
u(x)
u
F
F
K(x)
$\mathbf{u}(\mathbf{x}_p^j)$
$d\mathbf{u}(\mathbf{x}_p^j)$
$\mathbf{K}(\mathbf{x}_p^j)$
u(x)
u(x)
u

(a) Converged non-linear FEM increments

(b) Storing $(\mathbf{u}(\mathbf{x}), \mathbf{F})$, $\mathbf{x}$ and $\mathbf{K}(\mathbf{x})$

(c) Last converged non-linear FEM state

(d) Obtain perturbed displacement vectors

Figure F.8: Determination of numerical sensitivity information

- Traditional topology optimization using compliance

  Identical to the implementation of PTOc, TTOc aims to minimize structural compliance for a given volume fraction (Equation (2.59)) using the developed micro-scale penalization technique (Equation (3.1)). In this case, the required numerical sensitivity information of the objective function ($\partial c / \partial x_j$) will be obtained via Equation (F.9). However, it should be noted that for the compliance minimization problem it is not required to compute an entire new displacement correction for each perturbation, because for this problem the MMA only requires a column vector containing derivatives of the overall compliance with respect to all density perturbations. Therefore, the previously procedure can be simplified for obtaining numerical sensitivity information related to the minimum compliance problem. Nevertheless, it is also possible to simply adjust four lines in Algorithm 6 with the following: 1(a) Obtain overall compliance resulting from the non-perturbed non-linear FEM solution using Equation (2.54), 2(c) Determine overall compliance based on current perturbation using Equation (2.54), 2(d) Compute numerical sensitivity information through Equation (F.9), and 3. Construct column vector containing derivatives of the overall compliance value with respect to all density perturbations (Equation (F.19)):

$$\frac{\partial c}{\partial x_j} \approx \begin{bmatrix} \frac{\mathrm{d}c}{\mathrm{d}x_1} & \frac{\mathrm{d}c}{\mathrm{d}x_2} & \cdots & \frac{\mathrm{d}c}{\mathrm{d}x_n} \end{bmatrix}^T . \tag{F.19}$$

However, as discussed in Section 2.3 the main challenge of implementing a non-linear material model into topology optimization algorithms lies in the computational efficiency of the sensitivity analysis. This could be done by performing a linear (adjoint) sensitivity analysis, which is quite effective as illustrated in [61]. In this approach, only the tangential stiffness related to the final converged state is used in the sensitivity analysis such that no iterations are necessary for evaluating the sensitivity information [72]. This is similar to the numerical sensitivity approach proposed here.

Furthermore, the inclusion of non-linear and anisotropic behaviour of cement-based materials may cause load path-dependencies. Due to these material path-dependencies, the sensitivity information will also be path-dependent [61]. As discussed in the latter reference, it is in that case possible that a small change in the load may cause a relatively large difference in the displacements, especially for load controlled analysis in the vicinity of limit points. Therefore, path-dependent models may experience difficulty in convergence. This difficulty could be avoided by using a displacement controlled topology optimization formulation as presented in [72].

F.3 Preliminary Results

This section presents two promising results that have been obtained from preliminary implementations of stiffness-constrained volume minimization using the PTO method and numerical sensitivity information into the TTO method. Please note that these preliminary findings are merely included to illustrate their potential for follow-up research and are not intended to serve as validations of the concepts.

F.3.1 Proportional Topology Optimization Using Stiffness

A preliminary implementation of stiffness-constrained volume minimization has been done using the PTO method. The result presented here has been obtained through implementation of both the decoupling of penalization effects (i.e. using a non-penalized stiffness measure) and strain penalization, as discussed Appendix F.2.1. Even though it was previously stated that same numerical input as for the stress-constrained problem has been used for the GMA, this first trial of the stiffness-constrained problem considered a symmetric version of the smoothed microscopic constitutive relationship for matters of simplicity. The considered benchmark example is the cantilever beam problem as depicted in Figure 3.5(a), having 20 × 12 elements. Further settings of the PTO algorithm are identical to the stress-constrained problem.

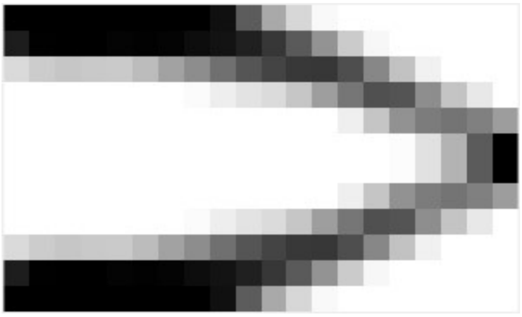
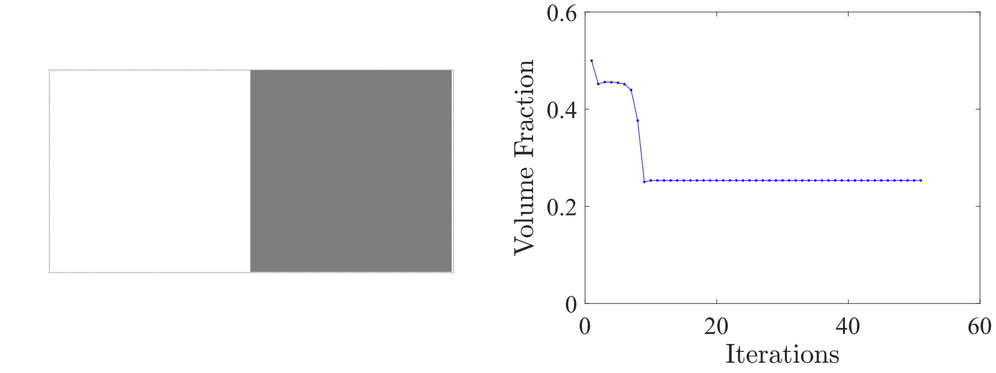


Figure F.9: Optimized topology of cantilever beam problem at iteration 21

The optimized topology shown in Figure F.9 clearly demonstrates the potential of stiffness-constrained volume minimization. However, as addressed in Appendix F.2.1 it remained subject to convergence issues in the non-linear FEM algorithm and has a tendency to ”smear out” densities (i.e. no occurrence of full nor void elements). As a result, the optimized topology in Figure F.9 was found at iteration 21 after which the optimization process diverged and crashed. Nevertheless, it can already be seen that the outlines of the structure have been formed. This is something that has also been seen in other optimization methodologies to be happening in a similar manner. During most optimizations, first a structure’s outlines are roughly defined followed by forming the infill and refining the overall topology. It can be concluded from this that further research is necessary.

F.3.2 Traditional Topology Optimization Using Stress

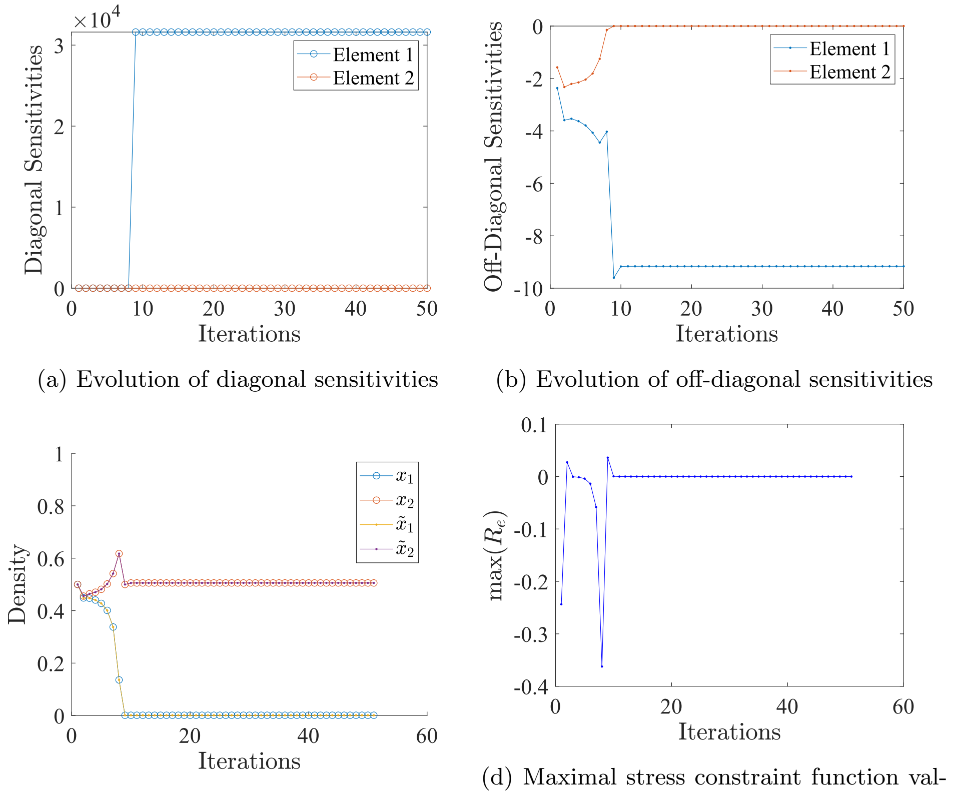
The first step in implementing numerical sensitivity information into the TTO method has been done using the stress-constrained problem. Despite having the framework (i.e. `MATLAB` codes) ready for obtaining numerical sensitivity information from the GMA, it has been chosen to increase the problem's complexity one step at a time. Therefore, instead of immediate incorporation of the multi-scale material model using a non-linear FEM, the Drucker-Prager failure criterion using a linear-elastic FEM has been considered first for matters of simplicity and comparison. However, even though a linear-elastic FEM is used here, numerical sensitivity information is obtained through the exact same procedure as described before. This means that the stress constraint function values are also retrieved using linear interpolation of a failure envelope that is composed of failure data points (FDPs). The aim of doing this has been to analyze and compare the behaviour and evolution of both the analytical and numerical sensitivities. The results presented here have been obtained using modified versions of the `MATLAB` code `PTOs.m` in [5]. In this case the material is modelled to have a strength ratio $\gamma = 3/2$, which means that its compressive stress limit is larger than the tensile stress limit (i.e. asymmetric strength). Recall that only the single rod benchmark is considered here (see Figure 3.7(a)). It should be noted that the only difference between the results presented next is the method of obtaining sensitivity information (i.e. analytical or numerical) and that all other settings are identical. First the results obtained using analytical sensitivity information are shown, followed by the results obtained using numerical sensitivities. For these specific results, it was found by trial and error that a perturbation parameter value of 10^{-2} worked most optimal for steering the optimization.



(a) Optimized topology (density distribution)

(b) Objective function versus iterations

Figure F.10: Optimized topology and evolution of objective function



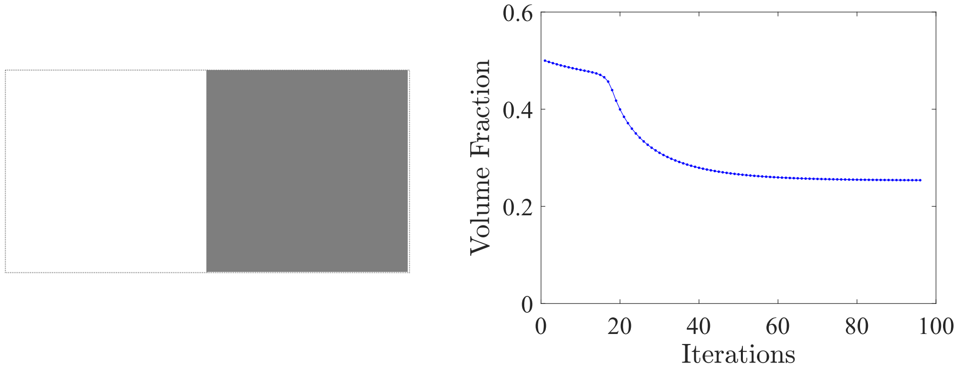
(a) Evolution of diagonal sensitivities

(b) Evolution of off-diagonal sensitivities

(c) Evolution of elemental densities

(d) Maximal stress constraint function values

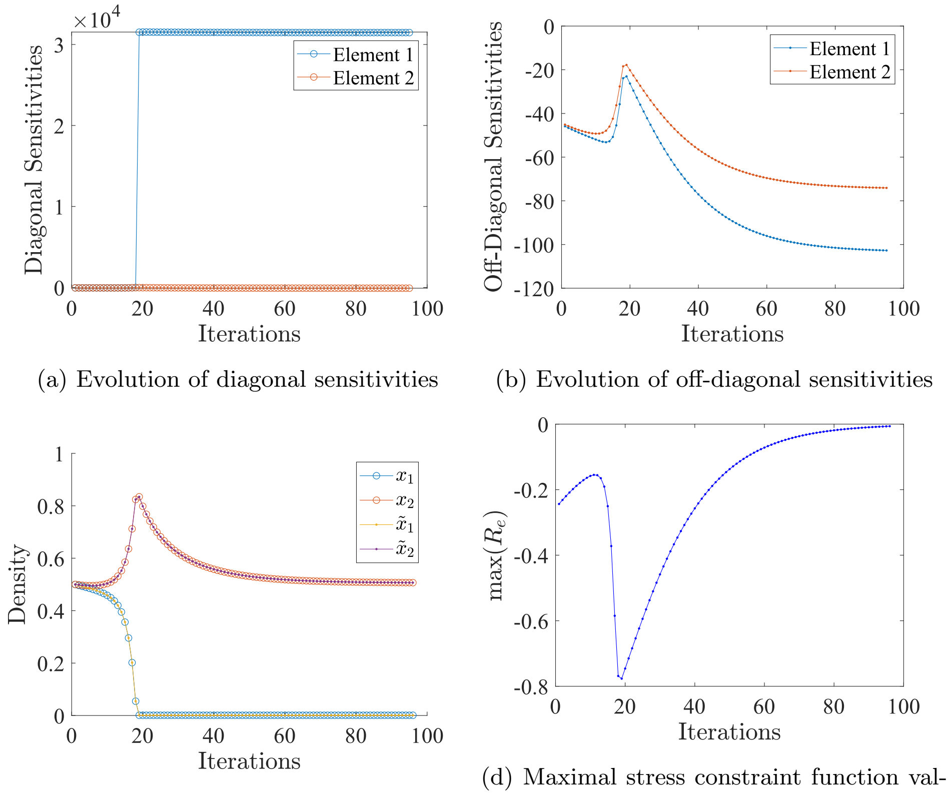
Figure F.11: Evolution of topology optimization process variables



(a) Optimized topology (density distribution)

(b) Objective function versus iterations

Figure F.12: Optimized topology and evolution of objective function



(a) Evolution of diagonal sensitivities

(b) Evolution of off-diagonal sensitivities

(c) Evolution of elemental densities

(d) Maximal stress constraint function values

Figure F.13: Evolution of topology optimization process variables

From Figures F.10 and F.12 it can be concluded that both analysis yield the same compressive optimized topologies, as expected from the TTO method. To be specific, the density distributions obtained using analytical and numerical sensitivity information are identical, having a final volume fraction of 0.25. Furthermore, note the similarities between the evolution of topology optimization process variables of both analysis. First of all, Figures F.10(b) and F.12(b) show that the objective function (i.e. the volume fraction) versus iterations graphs decrease in twofold. In the initial iterations the graphs decrease from 0.5 to about 0.45 and then go down towards the final value of 0.25. Additionally, the evolution of diagonal sensitivities (see Figures F.11(a) and F.13(a)) of the two elements start at a very small value (about zero) after which the sensitivities of elements 1 make a large and identical jump of more than $3 \cdot 10^4$ in both analysis. The same holds for Figures F.11(c) and F.13(c) in which the evolution of elemental densities show similarities. Initially, the densities of elements 1 increase exponentially while the densities of elements 2 decrease towards the minimal value of x_{min}. After the same iteration as the jumps in Figures F.11(a) and F.13(a), the densities of elements 1 decrease towards their final values. Besides, the maximal stress constraint function values (see Figures F.11(d) and F.13(d)) roughly evolve in a similar manner. Namely, both analysis start with an increase, followed by a significant decrease after which the stress constraint function value recovers towards a small value just below zero.

Obviously, there are also (large) differences between the evolution of topology optimization process variables. To start with, Figures F.10(b) and F.12(b) clearly illustrate that the volume fraction of the numerical analysis evolves smoother compared to the analytical analysis. This is probably the reason why the numerical simulation converges significantly slower than the analytical version. This becomes clear when looking at which iteration the peaks and drops appear in all of the evolutionary graphs. For the analysis using analytical sensitivities these occur close to iteration 10, while for the one using numerical sensitivity information these occur only at about iteration 20. Besides, the numerical analysis requires almost 90 iterations to reach convergence while the analytical analysis only needs about 15 iterations (the standard TTOs codes are set to have a minimum number of iterations of 50, hence the larger number of iterations). The most significant differences occur for the off-diagonal sensitivities (see Figures F.11(b) and F.13(b)). Even though the off-diagonal sensitivity values of element 2 are at all times larger than those of element 1 in both analysis, their evolution and order of magnitudes are very different. In the analytical simulation the off-diagonal sensitivity of the void element (i.e. element 2) becomes zero, which is something that does not happen in the numerical simulation. Furthermore, in the numerical analysis the magnitudes of these sensitivities are about 10 times larger than those in the analytical version.

The same strategy has also been applied to larger problems having more elements. However, in these cases the optimization process already diverged from the beginning causing several issues. This may be due to the significant differences that have been found in the aforementioned comparison. Especially the differences in evolution and order of magnitude in the off-diagonal sensitivities may incorrectly steer the optimization algorithm. In conclusion, more research is required to better understand the relation between numerical sensitivity information and the mathematical solvers used in the TTO method.

Appendix G Developed Codes

The MATLAB codes that have been developed and used throughout this research are provided in this appendix.

G.1 Topology Optimization Codes

The MATALB codes presented here are related to the topology optimization methods as described in Section 2.3.

G.1.1 Two-Dimensional Optimization Codes

The next two codes have been used for the optimization of two-dimensional structures while considering the minimum compliance and stress-constrained problems, respectively, using the PTO method. Both codes substantially inherit from the implementations by Biyikli & To [21, 22]. Please note that the dimensions of BeamL and BeamH are in m while MaxLoad is in N.

```
% PROPORTIONAL TOPOLOGY OPTIMIZATION FOR THE MINIMUM COMPLIANCE PROBLEM
% WITH INCORPORATION OF THE DEVELOPED NON-LINEAR FEM USING THE GRANULAR
% MICROMECHANICS APPROACH ---- DEVELOPED AND CODED BY R.J.M. Bol (2022)
function [x,c] = PTOc_LC_FEM_GMA_Plane_Stress(BeamL,BeamH,nelx,nely, ...
MaxLoad,rmin,StrucTyp,lambda)
%% MATERIAL AND PARAMETER PROPERTIES
volfrac = 0.5;            % volume fraction
penal = 3;                % penalization power
cd = 0.1;                 % change in density tolerance
cc = 0.1;                 % change in compliance tolerance
MatTyp = 1;               % material type: tension-compression asymmetry
GMAinput = 1;             % GMA input parameters (H. Jia et al. (2017))
Ninc = 1;                 % number of load increments
tolerance = 10^(-2);      % non-linear FEM tolerance
xMin = 1e-3;              % minimum density value
alpha = 0.5;              % history coefficient
b = BeamL/nelx;           % [m]
h = BeamH/nely;           % [m]
t = min(BeamL,BeamH)/100; % [m]
%% PREPARE FINITE ELEMENT ANALYSIS
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = ...
repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
%% DEFINE BOUNDARY CONDITIONS
[Fdofs,freedofs] = BCs(nely,nelx,StrucTyp);
%% PREPARE FILTER
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
        e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
    end
    end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
%% INITIALIZE ITERATION
x = repmat(volfrac,nely,nelx);
loop = 0;
changedens = 1; changecomp = 1;
% START ITERATION
while (true)
loop = loop + 1;
[EnergySE_mat,principal_stresses] = LC_FEM_GMA_Plane_Stress(nelx, ...
nely,penal,Fdofs,freedofs,MaxLoad,x,b,h,t,MatTyp,edofMat,iK,jK, ...
GMAinput,Ninc,tolerance,lambda);
C = EnergySE_mat;
c(loop) = sum(C(:));
% PRINT RESULTS
if loop > 10; changecomp = ...
abs(sum(c(loop-9:loop-5))-sum(c(loop-4:loop)))/sum(c(loop-4:loop));
end
fprintf('It.:%5i C.:%8.2f Vol.:%5.2f Chd:%6.3f Chc:%6.3f\n',loop, ...
sum(C(:)),mean(x(:)),changedens,changecomp);
% PLOT DENSITIES
colormap(gray); imagesc(1-x); caxis([0 1]); axis equal off; drawnow;
% CHECK STOP CRITERIA
if(changedens < cd && changecomp < cc && loop > 50); break; end
% UPDATE DESIGN VARIABLES
xTarget = nelx*nely*volfrac;
xRemaining = xTarget;
xold = x; x(:) = 0;
C_proportion = C/sum(C(:));
C_proportion(:) = (H*C_proportion(:))./Hs;
while (xRemaining > 0.001)
    xDist = xRemaining.*C_proportion;
    x = x+xDist;
    x = max(min(x,1),xMin);
    xRemaining = xTarget-sum(x(:));
end
x = alpha*xold+(1-alpha)*x;
changedens = max(abs((1/alpha-1)*(x(:)-xold(:))));
end
%% PLOTTING
figure;
plot(c,'-b.');
xlabel('Iteration','FontName','Times New Roman','Interpreter','latex')
```

```
ylabel('Compliance','FontName','Times New Roman','Interpreter','latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
end
%% DEFINE BOUNDARY CONDITIONS
function [Fdofs,freedofs] = BCs(nely,nelx,StrucTyp)
alldofs = [(1:2*(nely+1)*(nelx+1))];
if StrucTyp == 1 %% cantilever ------------------- H. Jia et al. (2017)
    fdofs = [1:2*(nely+1)];
    Topx = (nelx+1)*2*(nely+1)-1-2*nely;
    Topy = Topx+1;
    Fdofs = Topy+2*floor(nely/2);
elseif StrucTyp == 2 %% corner loading cantilever  H. Jia et al. (2017)
    fdofs = [1:2*(nely+1)];
    Fdofs = 2*(nely+1)*(nelx+1);
elseif StrucTyp == 3 %% Michell structure -------- H. Jia et al. (2017)
    fdofs = [2*(nely+1)-1,2*(nely+1),2*(nelx+1)*(nely+1)];
    Fdofs = 2*(nely+1)*(floor(nelx/2)+1);
elseif StrucTyp == 4 %% MBB structure ------------ H. Jia et al. (2017)
    fdofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(nely+1)]);
    Fdofs = 2;
elseif StrucTyp == 5 %% one-bar truss example ------------- Bouw (2020)
    fdofs = [1 2 union(3:2:((nely+1)*2), ...
    (((nelx+1)*(nely+1)*2)-(nely*2)-1):2:(((nelx+1)*(nely+1)*2)))];
    Fdofs = (((nelx+1)*(nely+1)+nely)-(0:2:2*nely+1));
elseif StrucTyp == 6 %% four-bar truss example (even) ----- Bouw (2020)
    fdofs = [1 2 nely*2+1 nely*2+2 ((nelx+1)*(nely+1)*2)-(nely*2)-1 ...
    ((nelx+1)*(nely+1)*2)-(nely*2) (nelx+1)*(nely+1)*2-1 ...
    (nelx+1)*(nely+1)*2];
    Fdofs = (nelx+1)*(nely+1)+1;
elseif StrucTyp == 7 %% two-bar truss example ------------- Bouw (2020)
    yrem = rem(nelx,2); % Load Distribution (nr of elements)
    Topx = (nelx+1)*2*(nely+1)-1-2*nely;
    Topy = Topx+1;
    if yrem == 0
        Fdofs = Topy/2;
    else; ld = 2;
        Mid_y_left_bot = 2*(nely+1)*(nelx+1)/2;
        Mid_y_left_top = Mid_y_left_bot-2*nely;
        Mid_x_left_top = Mid_y_left_top-1;
        Mid_x_right_top = Mid_x_left_top+2*(nely+1);
        Fdofs = union(Mid_x_left_top,Mid_x_right_top);
    end
    fdofs_hor = 2*(nely+1)-1:2*(nely+1):2*(nely+1)*(nelx+1)-1;
    fdofs_ver = 2*(nely+1):2*(nely+1):2*(nely+1)*(nelx+1);
    fdofs = union(fdofs_hor,fdofs_ver);
elseif StrucTyp == 8 %% (half) MBB-beam ------------------- Bouw (2020)
    fdofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(nely+1)]);
    Fdofs = 2;
end
freedofs = setdiff(alldofs,fdofs);
end
```

```
% PROPORTIONAL TOPOLOGY OPTIMIZATION FOR THE STRESS-CONSTRAINED PROBLEM
% WITH INCORPORATION OF THE DEVELOPED NON-LINEAR FEM USING THE GRANULAR
% MICROMECHANICS APPROACH ---- DEVELOPED AND CODED BY R.J.M. Bol (2022)
function [x,v] = PTOs_LC_FEM_GMA_Plane_Stress(BeamL,BeamH,nelx,nely, ...
MaxLoad,rmin,StrucTyp)
%% MATERIAL AND PARAMETER PROPERTIES
volfrac = 0.5;              % (initial) volume fraction
penal = 3;                  % penalizatoion power
cv = 0.1;                   % change in volume tolerance
MatTyp = 2;                 % material type: bi-linear force-law
GMAinput = 2;               % GMA input parameters (R.J.M. Bol (2022))
Ninc = 1;                   % number of load increments
tolerance = 10^(-8);        % non-linear FEM tolerance
xMin = 1e-3;                % minimum density value
q = 2;                      % proportion exponent
b = BeamL/nelx;             % [m]
h = BeamH/nely;             % [m]
t = min(BeamL,BeamH)/100;   % [m]
if MatTyp == 2; lambda = 1; end
Eps = repmat(sqrt(xMin),1,nelx*nely); % vector relaxation parameters
%% PREPARE FINITE ELEMENT ANALYSIS
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = ...
repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
%% DEFINE BOUNDARY CONDITIONS
[Fdofs,freedofs] = BCs(nely,nelx,StrucTyp);
%% PREPARE FILTER
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
        e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
    end
    end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
%% INITIALIZE ITERATION
x = repmat(volfrac,nely,nelx);
loop = 0;
changedens = 1; changevol = 1;
% START ITERATION
```

```
while (true)
loop = loop + 1;
[EnergySE_mat,principal_stresses] = LC_FEM_GMA_Plane_Stress(nelx, ...
nely,penal,Fdofs,freedofs,MaxLoad,x,b,h,t,MatTyp,edofMat,iK,jK, ...
GMAinput,Ninc,tolerance,lambda);
v(loop) = sum(x(:))/(nelx*nely);
C = EnergySE_mat;
[FS] = FailSens_Plane_Stress(principal_stresses);
% relaxation of failure criterion (<=1)
FS = FS-Eps.*((1-reshape(x,1,nelx*nely))./reshape(x,1,nelx*nely));
FS = max(FS,xMin); %%% remove negative values
FS = reshape(FS,nely,nelx);
% PRINT RESULTS
if loop > 10; changevol = ...
abs(sum(v(loop-9:loop-5))-sum(v(loop-4:loop)))/sum(v(loop-4:loop));
end
fprintf('It.:%5i C.:%8.2f Vol.:%5.2f Chd:%6.3f Chv:%6.3f FS:%6.3f\n',...
loop,sum(C(:)),mean(x(:)),changedens,changevol,max(FS(:)));
% PLOT DENSITIES
colormap(gray); imagesc(1-x); caxis([0 1]); axis equal off; drawnow;
% CHECK STOP CRITERIA
if (changevol < cv && max(FS(:)) < 1); break; end
% UPDATE DESIGN VARIABLES
if (max(FS(:)) > 1)
xTarget = sum(x(:))+0.01*numel(x);
else
xTarget = sum(x(:))-0.01*numel(x);
end
xRemaining = xTarget;
xold = x; x(:) = 0;
FS_proportion = FS.^q/sum(sum(FS.^q));
FS_proportion(:) = (H*FS_proportion(:))./Hs;
while (xRemaining > 0.001)
xDist = xRemaining.*FS_proportion;
x = x+xDist;
x = max(min(x,1),xMin);
xRemaining = xTarget-sum(x(:));
end
changedens = max(abs(x(:)-xold(:)));
end
%% PLOTTING
figure;
plot(v,'-b.');
xlabel('Iterations','FontName','Times New Roman','Interpreter','latex')
ylabel('Volume Fraction','FontName','Times New Roman','Interpreter', ...
'latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
ylim([0 0.6])
end
%% DEFINE BOUNDARY CONDITIONS
function [Fdofs,freedofs] = BCs(nely,nelx,StrucTyp)
alldofs = [(1:2*(nely+1)*(nelx+1))];
if StrucTyp == 1 %% cantilever ------------------- H. Jia et al. (2017)
    fdofs = [1:2*(nely+1)];
    Topx = (nelx+1)*2*(nely+1)-1-2*nely;
    Topy = Topx+1;
    Fdofs = Topy+2*floor(nely/2);
elseif StrucTyp == 2 %% corner loading cantilever  H. Jia et al. (2017)
    fdofs = [1:2*(nely+1)];
    Fdofs = 2*(nely+1)*(nelx+1);
elseif StrucTyp == 3 %% Michell structure -------- H. Jia et al. (2017)
    fdofs = [2*(nely+1)-1,2*(nely+1),2*(nelx+1)*(nely+1)];
    Fdofs = 2*(nely+1)*(floor(nelx/2)+1);
elseif StrucTyp == 4 %% MBB structure ------------ H. Jia et al. (2017)
    fdofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(nely+1)]);
    Fdofs = 2;
elseif StrucTyp == 5 %% one-bar truss example ------------- Bouw (2020)
    fdofs = [1 2 union(3:2:((nely+1)*2), ...
    (((nelx+1)*(nely+1)*2)-(nely*2)-1):2:(((nelx+1)*(nely+1)*2)))];
    Fdofs = (((nelx+1)*(nely+1)+nely)-(0:2:2*nely+1));
elseif StrucTyp == 6 %% four-bar truss example (even) ----- Bouw (2020)
    fdofs = [1 2 nely*2+1 nely*2+2 ((nelx+1)*(nely+1)*2)-(nely*2)-1 ...
    ((nelx+1)*(nely+1)*2)-(nely*2) (nelx+1)*(nely+1)*2-1 ...
    (nelx+1)*(nely+1)*2];
    Fdofs = (nelx+1)*(nely+1)+1;
elseif StrucTyp == 7 %% two-bar truss example ------------- Bouw (2020)
    yrem = rem(nelx,2); % Load Distribution (nr of elements)
    Topx = (nelx+1)*2*(nely+1)-1-2*nely;
    Topy = Topx+1;
    if yrem == 0
        Fdofs = Topy/2;
    else; ld = 2;
        Mid_y_left_bot = 2*(nely+1)*(nelx+1)/2;
        Mid_y_left_top = Mid_y_left_bot-2*nely;
        Mid_x_left_top = Mid_y_left_top-1;
        Mid_x_right_top = Mid_x_left_top+2*(nely+1);
        Fdofs = union(Mid_x_left_top,Mid_x_right_top);
    end
    fdofs_hor = 2*(nely+1)-1:2*(nely+1):2*(nely+1)*(nelx+1)-1;
    fdofs_ver = 2*(nely+1):2*(nely+1):2*(nely+1)*(nelx+1);
    fdofs = union(fdofs_hor,fdofs_ver);
elseif StrucTyp == 8 %% (half) MBB-beam ------------------- Bouw (2020)
    fdofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(nely+1)]);
    Fdofs = 2;
end
freedofs = setdiff(alldofs,fdofs);
end
```

G.1.2 Three-Dimensional Optimization Codes

The following codes have been used for the optimization of three-dimensional structures while considering the minimum compliance and stress-constrained problems, respectively, using the PTO method. Please note once more that the dimensions of `BeamL`, `BeamH` and `BeamW` are in m while `MaxLoad` is in N.

```
% PROPORTIONAL TOPOLOGY OPTIMIZATION FOR THE MINIMUM COMPLIANCE PROBLEM
% WITH INCORPORATION OF THE DEVELOPED NON-LINEAR FEM USING THE GRANULAR
% MICROMECHANICS APPROACH ---- DEVELOPED AND CODED BY R.J.M. Bol (2022)
function [x,C] = PTOc_LC_FEM_GMA_3D(BeamL,BeamH,BeamW,nelx,nely, ...
nelz,MaxLoad,rmin,lambda)
%% MATERIAL AND PARAMETER PROPERTIES
volfrac = 0.5;            % volume fraction
nele = nelx*nely*nelz;    % total number of elements
penal = 3;                % penlization power
MatTyp = 1;               % material type: tension-compression asymmetry
GMAinput = 1;             % GMA input parameters (H. Jia et al. (2017))
Ninc = 1;                 % number of load increments
tolerance = 10^(-2);      % non-linear FEM tolerance
maxloop = 50;             % Maximum number of iterations
tolx = 0.001;             % Terminarion criterion
displayflag = 0;          % Display structure flag
xMin = 1e-3;              % minimal density value
alpha = 0.5;              % history coefficient
b = BeamL/nelx;           % [m]
h = BeamH/nely;           % [m]
t = BeamW/nelz;           % [m]
%% PREPARE FINITE ELEMENT ANALYSIS
nodegrd = reshape(1:(nely+1)*(nelx+1),nely+1,nelx+1);
nodeids = reshape(nodegrd(1:end-1,1:end-1),nely*nelx,1);
nodeidz = 0:(nely+1)*(nelx+1):(nelz-1)*(nely+1)*(nelx+1);
nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
edofVec = 3*nodeids(:)+1;
edofMat = repmat(edofVec,1,24)+ ...
   repmat([0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1 ...
   3*(nely+1)*(nelx+1)+[0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1]],nele,1);
iK = reshape(kron(edofMat,ones(24,1))',24*24*nele,1);
jK = reshape(kron(edofMat,ones(1,24))',24*24*nele,1);
%% DEFINE BOUNDARY CONDITIONS (one-bar truss example ------ Bouw (2020))
ndof = 3*(nelx+1)*(nely+1)*(nelz+1);
F = zeros(ndof,1);
U = zeros(ndof,1);
Fdofs = [7,10,25,28];
fdofs = [1,3,4,5,6,19,22,13,16,31,34];
freedofs = setdiff(1:ndof,fdofs);
%% PREPARE FILTER
iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for k1 = 1:nelz
    for i1 = 1:nelx
        for j1 = 1:nely
```

```
            e1 = (k1-1)*nelx*nely + (i1-1)*nely+j1;
            for k2 = ...
                max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-1),nelz)
                for i2 = ...
                    max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
                    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+ ...
                        (ceil(rmin)-1),nely)
                        e2 = (k2-1)*nelx*nely + (i2-1)*nely+j2;
                        k = k+1;
                        iH(k) = e1;
                        jH(k) = e2;
                        sH(k) = ...
                        max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-k2)^2));
                    end
                end
            end
        end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
%% INITIALIZE ITERATION
x = repmat(volfrac,[nely,nelx,nelz]);
loop = 0;
change = 1;
% START ITERATION
while change > tolx && loop < maxloop
    loop = loop+1;
    [EnergySE_mat,principal_stresses] = LC_FEM_GMA_3D(nelx,nely, ...
    nelz,penal,Fdofs,freedofs,MaxLoad,x,b,h,t,MatTyp,edofMat,iK,jK, ...
    GMAinput,Ninc,tolerance,lambda);
    c = EnergySE_mat;
    C(loop) = sum(c(:));
    % PRINT RESULTS
    fprintf(' It.:%5i C.:%7.3f Vol.:%7.3f ch.:%7.3f\n',loop, ...
    sum(c(:)),mean(x(:)),change);
    hold on;
    % UPDATE DESIGN VARIABLES
    xTarget = nelx*nely*nelz*volfrac;
    xRemaining = xTarget;
    xold = x; x(:) = 0;
    C_proportion = c/sum(c(:));
    C_proportion(:) = (H*C_proportion(:))./Hs;
    while (xRemaining > 0.001)
        xDist = xRemaining.*C_proportion;
        x = x+xDist;
        x = max(min(x,1),xMin);
        xRemaining = xTarget-sum(x(:));
    end
    x = alpha*xold+(1-alpha)*x;
    change = max(abs((1/alpha-1)*(x(:)-xold(:))));
    % PLOT DENSITIES
    if displayflag, clf; display_3D(x); end
end
clf; display_3D(x);
hold on;
```

```
%% PLOTTING
figure;
plot(C,'-b.');
xlabel('Iteration','FontName','Times New Roman','Interpreter','latex')
ylabel('Compliance','FontName','Times New Roman','Interpreter','latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
end
%% === DISPLAY 3D TOPOLOGY (ISO-VIEW) ===
function display_3D(rho)
[nely,nelx,nelz] = size(rho);
hx = 1; hy = 1; hz = 1;            % unit element size
face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
set(gcf,'Name','ISO display','NumberTitle','off');
for k = 1:nelz
    z = (k-1)*hz;
    for i = 1:nelx
        x = (i-1)*hx;
        for j = 1:nely
            y = nely*hy - (j-1)*hy;
            if (rho(j,i,k) > 0.01 )% display density threshold
                vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; ...
                x y z+hx;x y-hx z+hx; x+hx y-hx z+hx;x+hx y z+hx];
                vert(:,[2 3]) = vert(:,[3 2]); vert(:,2,:) = ...
                -vert(:,2,:);
                patch('Faces',face,'Vertices',vert,'FaceColor', ...
                [0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k)),0.2+ ...
                0.8*(1-rho(j,i,k))]);
                hold on;
            end
        end
    end
end
axis equal; axis tight; axis off; box on; view([30,30]); pause(1e-6);
end
```

```
% PROPORTIONAL TOPOLOGY OPTIMIZATION FOR THE STRESS-CONSTRAINED PROBLEM
% WITH INCORPORATION OF THE DEVELOPED NON-LINEAR FEM USING THE GRANULAR
% MICROMECHANICS APPROACH ---- DEVELOPED AND CODED BY R.J.M. Bol (2022)
function [x,v] = PTOs_LC_FEM_GMA_3D(BeamL,BeamH,BeamW,nelx,nely, ...
nelz,MaxLoad,rmin)
%% MATERIAL AND PARAMETER PROPERTIES
volfrac = 0.5;              % (initial) volume fraction
nele = nelx*nely*nelz;      % total number of elements
penal = 3;                  % penlization power
MatTyp = 2;                 % material type: bi-linear force-law
GMAinput = 2;               % GMA input parameters (R.J.M. Bol (2022))
Ninc = 1;                   % number of load increments
tolerance = 10^(-8);        % non-linear FEM tolerance
tolx = 0.01;                % Terminarion criterion
displayflag = 0;            % Display structure flag
xMin = 1e-3;                % minimal density value
q = 2;                      % proportion exponent
b = BeamL/nelx;             % [m]
h = BeamH/nely;             % [m]
t = BeamW/nelz;             % [m]
if MatTyp == 2; lambda = 1; end
Eps = repmat(sqrt(xMin),1,nelx*nely*nelz); % vector relaxation
%% PREPARE FINITE ELEMENT ANALYSIS
nodegrd = reshape(1:(nely+1)*(nelx+1),nely+1,nelx+1);
nodeids = reshape(nodegrd(1:end-1,1:end-1),nely*nelx,1);
nodeidz = 0:(nely+1)*(nelx+1):(nelz-1)*(nely+1)*(nelx+1);
nodeids = repmat(nodeids,size(nodeidz))+repmat(nodeidz,size(nodeids));
edofVec = 3*nodeids(:)+1;
edofMat = repmat(edofVec,1,24)+ ...
   repmat([0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1 ...
   3*(nely+1)*(nelx+1)+[0 1 2 3*nely + [3 4 5 0 1 2] -3 -2 -1]],nele,1);
iK = reshape(kron(edofMat,ones(24,1))',24*24*nele,1);
jK = reshape(kron(edofMat,ones(1,24))',24*24*nele,1);
%% DEFINE BOUNDARY CONDITIONS (one-bar truss example ------ Bouw (2020))
ndof = 3*(nelx+1)*(nely+1)*(nelz+1);
F = zeros(ndof,1);
U = zeros(ndof,1);
Fdofs = [7,10,25,28];
fdofs = [1,3,4,5,6,19,22,13,16,31,34];
freedofs = setdiff(1:ndof,fdofs);
%% PREPARE FILTER
iH = ones(nele*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for k1 = 1:nelz
    for i1 = 1:nelx
        for j1 = 1:nely
            e1 = (k1-1)*nelx*nely + (i1-1)*nely+j1;
            for k2 = ...
                max(k1-(ceil(rmin)-1),1):min(k1+(ceil(rmin)-1),nelz)
                for i2 = ...
                    max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
                    for j2 = max(j1-(ceil(rmin)-1),1):min(j1+ ...
                        (ceil(rmin)-1),nely)
                        e2 = (k2-1)*nelx*nely + (i2-1)*nely+j2;
                        k = k+1;
                        iH(k) = e1;
                        jH(k) = e2;
                        sH(k) = ...
                        max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2+(k1-k2)^2));
                    end
                end
            end
        end
    end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
%% INITIALIZE ITERATION
x = repmat(volfrac,[nely,nelx,nelz]);
loop = 0;
change = 1;
% START ITERATION
while (true)
    loop = loop+1;
    [EnergySE_mat,principal_stresses] = LC_FEM_GMA_3D(nelx,nely, ...
    nelz,penal,Fdofs,freedofs,MaxLoad,x,b,h,t,MatTyp,edofMat,iK,jK, ...
    GMAinput,Ninc,tolerance,lambda);
    v(loop) = sum(x(:))/(nelx*nely*nelz);
    c = EnergySE_mat;
    C(loop) = sum(c(:));
    [FS] = FailSens_3D(principal_stresses);
    % relaxation of failure criterion (<=1)
    FS = FS-Eps.*((1-reshape(x,1,nelx*nely*nelz))./reshape(x,1, ...
    nelx*nely*nelz));
    FS = max(FS,xMin); %%% remove negative values
    FS = reshape(FS,[nely,nelx,nelz]);
    % PRINT RESULTS
    if loop > 10; change = ...
    abs(sum(v(loop-9:loop-5))-sum(v(loop-4:loop)))/sum(v(loop-4:loop));
    end
    fprintf(' It.:%5i FS:%7.3f Vol.:%7.3f ch.:%7.3f\n',loop, ...
    max(FS(:)),mean(x(:)),change);
    hold on;
    if (change < tolx && max(FS(:)) < 1); break; end
    % UPDATE DESIGN VARIABLES
    if (max(FS(:)) > 1)
    xTarget = sum(x(:))+0.01*numel(x);
    else
    xTarget = sum(x(:))-0.01*numel(x);
    end
    xRemaining = xTarget;
    xold = x; x(:) = 0;
    FS_proportion = FS.^q/sum(sum(FS.^q));
    FS_proportion(:) = (H*FS_proportion(:))./Hs;
    while (xRemaining > 0.001)
        xDist = xRemaining.*FS_proportion;
        x = x+xDist;
        x = max(min(x,1),xMin);
        xRemaining = xTarget-sum(x(:));
    end
    % PLOT DENSITIES
    if displayflag, clf; display_3D(x); end
end
clf; display_3D(x);
hold on;
%% PLOTTING
figure;
plot(v,'-b.');
xlabel('Iterations','FontName','Times New Roman','Interpreter','latex')
ylabel('Volume Fraction','FontName','Times New Roman','Interpreter', ...
'latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
ylim([0 0.6])
end
%% === DISPLAY 3D TOPOLOGY (ISO-VIEW) ===
function display_3D(rho)
[nely,nelx,nelz] = size(rho);
hx = 1; hy = 1; hz = 1;            % unit element size
face = [1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
set(gcf,'Name','ISO display','NumberTitle','off');
for k = 1:nelz
    z = (k-1)*hz;
    for i = 1:nelx
        x = (i-1)*hx;
        for j = 1:nely
            y = nely*hy - (j-1)*hy;
            if (rho(j,i,k) > 0.01 )% display density threshold
                vert = [x y z; x y-hx z; x+hx y-hx z; x+hx y z; ...
                x y z+hx;x y-hx z+hx; x+hx y-hx z+hx;x+hx y z+hx];
                vert(:,[2 3]) = vert(:,[3 2]); vert(:,2,:) = ...
                -vert(:,2,:);
                patch('Faces',face,'Vertices',vert,'FaceColor', ...
                [0.2+0.8*(1-rho(j,i,k)),0.2+0.8*(1-rho(j,i,k)),0.2+ ...
                0.8*(1-rho(j,i,k))]);
                hold on;
            end
        end
    end
end
axis equal; axis tight; axis off; box on; view([30,30]); pause(1e-6);
end
```

G.1.3 Determination of Failure Sensitivity Codes

The considered failure envelopes are read by a separate algorithm to determine the failure sensitivities as described in Section 3.3.2. The next MATLAB implementations perform linear interpolation of the FDPs for two-dimensional as well as three-dimensional optimizations, respectively.

```
function [FS] = FailSens(principal_stresses)
%% Read data from text file
fileID = fopen('FDPs_Plane_Stress','r');
formatSpec = '%f %e %e'; %% alphas,s11s,s22s
sizeINPUT = [3 Inf];
INPUT = fscanf(fileID,formatSpec,sizeINPUT);
fclose(fileID);
INPUT = INPUT';
alphas = INPUT(:,1);
s11_int_failure = INPUT(:,2);
s22_int_failure = INPUT(:,3);
%% Determination of failure sensitivity
thetas = atan2(principal_stresses(2,:), ...
principal_stresses(1,:)).*(180/pi);
difs = abs(alphas-thetas);
[~,ind1] = min(difs);
[dif2,~] = max(difs);
cols = [0:width(ind1)-1];
difs_size = size(difs);
colvals = cols*difs_size(1);
dif_ind = ind1+colvals;
difs2=reshape(difs,length(difs(:)),1);
difs2(dif_ind) = difs2(dif_ind) + dif2(:);
difs2=reshape(difs2,difs_size(1),difs_size(2));
[~,ind3] = min(difs2);
near_dir = sort([ind3; ind1]);
%%% linearly interpolate failure data points (FDPs) from nearest FDPs
FDP11s = s11_int_failure(near_dir(1,:))+ ...
(s11_int_failure(near_dir(2,:))- ...
s11_int_failure(near_dir(1,:))).*((thetas'- ...
alphas(near_dir(1,:)))./(alphas(near_dir(2,:))-alphas(near_dir(1,:))));
FDP22s = s22_int_failure(near_dir(1,:))+ ...
(s22_int_failure(near_dir(2,:))- ...
s22_int_failure(near_dir(1,:))).*((thetas'- ...
alphas(near_dir(1,:)))./(alphas(near_dir(2,:))-alphas(near_dir(1,:))));
%%% determine failure sensitivities
FS = (sqrt(principal_stresses(1,:).^2+ ...
principal_stresses(2,:).^2))./(sqrt(FDP11s'.^2+FDP22s'.^2));
end
```

```
function [FS] = FailSens_3D(principal_stresses)
%% Read data from text file
fileID = fopen('FDPs_3D.txt','r');
formatSpec = '%f %f %e %e %e'; %% thetas,phis,s11s,s22s,s33s
sizeINPUT = [5 Inf];
INPUT = fscanf(fileID,formatSpec,sizeINPUT);
fclose(fileID);
INPUT = INPUT';
LTs = INPUT(:,1);
LPs = INPUT(:,2);
s11_int_failure = INPUT(:,3);
s22_int_failure = INPUT(:,4);
s33_int_failure = INPUT(:,5);
%% Determination of failure sensitivity
thetas = acos(principal_stresses(3,:)./sqrt(principal_stresses(1, ...
:).^2+principal_stresses(2,:).^2+principal_stresses(3,:).^2)).*(180/pi);
phis = atan2(principal_stresses(2,:),principal_stresses(1,:)).*(180/pi);
%%% transform phis to unique coordinates (+360 degrees)
negs = phis < 0;
phis(negs) = phis(negs) + 360;
%%% isolate unique discretized angles
LTs_unique = uniquetol(LTs);
LPs_unique = uniquetol(LPs);
%%% find two closest unique discretized angles
difsLT = abs(thetas-LTs_unique);
difsLP = abs(phis-LPs_unique);
[~,indLT1] = min(difsLT);
[difLT2,~] = max(difsLT);
[~,indLP1] = min(difsLP);
[difLP2,~] = max(difsLP);
colsLT = [0:width(indLT1)-1];
colsLP = [0:width(indLP1)-1];
difs_sizeLT = size(difsLT);
difs_sizeLP = size(difsLP);
colvalsLT = colsLT*difs_sizeLT(1);
colvalsLP = colsLP*difs_sizeLP(1);
dif_indLT = indLT1+colvalsLT;
dif_indLP = indLP1+colvalsLP;
difsLT2 = reshape(difsLT,length(difsLT(:)),1);
difsLP2 = reshape(difsLP,length(difsLP(:)),1);
difsLT2(dif_indLT) = difsLT2(dif_indLT) + difLT2(:);
difsLP2(dif_indLP) = difsLP2(dif_indLP) + difLP2(:);
difsLT2 = reshape(difsLT2,difs_sizeLT(1),difs_sizeLT(2));
difsLP2 = reshape(difsLP2,difs_sizeLP(1),difs_sizeLP(2));
[~,indLT3] = min(difsLT2);
[~,indLP3] = min(difsLP2);
near_dirLT = sort([indLT3; indLT1]);
near_dirLP = sort([indLP3; indLP1]);
near_dirLT_val1 = LTs_unique(near_dirLT(1,:)); % finds theta values
near_dirLT_val2 = LTs_unique(near_dirLT(2,:)); % finds theta values
near_dirLP_val1 = LPs_unique(near_dirLP(1,:)); % finds phi values
near_dirLP_val2 = LPs_unique(near_dirLP(2,:)); % finds phi values
%%% find nearest failure data point directions
near_dir = zeros(4,width(principal_stresses));
for i = 1:width(principal_stresses)
```

```
% find FDP for theta_j & phi_i
indLTi1 = find(LTs==near_dirLT_val2(i));
indLPi1 = find(round(LPs)==round(near_dirLP_val1(i)));
[val1,~] = intersect(indLTi1,indLPi1); % 1st common value & position
near_dir(1,i) = val1;
% find FDP for theta_i & phi_i
indLTi2 = find(LTs==near_dirLT_val1(i));
indLPi2 = find(round(LPs)==round(near_dirLP_val1(i)));
[val2,~] = intersect(indLTi2,indLPi2); % 2nd common value & position
near_dir(2,i) = val2;
% find FDP for theta_j & phi_j
indLTi3 = find(LTs==near_dirLT_val2(i));
indLPi3 = find(round(LPs)==round(near_dirLP_val2(i)));
[val3,~] = intersect(indLTi3,indLPi3); % 3rd common value & position
near_dir(3,i) = val3;
% find FDP for theta_i & phi_j
indLTi4 = find(LTs==near_dirLT_val1(i));
indLPi4 = find(round(LPs)==round(near_dirLP_val2(i)));
[val4,~] = intersect(indLTi4,indLPi4); % 4th common value & position
near_dir(4,i) = val4;
end
%%% linearly interpolate failure data points (FDPs) from nearest FDPs
% 2 x-coordinates for phi
FDP11s1 = s11_int_failure(near_dir(2,:))+ ...
(s11_int_failure(near_dir(4,:))- ...
s11_int_failure(near_dir(2,:))).*((phis'- ...
LPs(near_dir(2,:)))./(LPs(near_dir(4,:))-LPs(near_dir(2,:))));
FDP11s2 = s11_int_failure(near_dir(1,:))+ ...
(s11_int_failure(near_dir(3,:))- ...
s11_int_failure(near_dir(1,:))).*((phis'- ...
LPs(near_dir(1,:)))./(LPs(near_dir(3,:))-LPs(near_dir(1,:))));
% 2 y-coordinates for theta
FDP22s1 = s22_int_failure(near_dir(2,:))+ ...
(s22_int_failure(near_dir(1,:))- ...
s22_int_failure(near_dir(2,:))).*((thetas'- ...
LTs(near_dir(2,:)))./(LTs(near_dir(1,:))-LTs(near_dir(2,:))));
FDP22s2 = s22_int_failure(near_dir(4,:))+ ...
(s22_int_failure(near_dir(3,:))- ...
s22_int_failure(near_dir(4,:))).*((thetas'- ...
LTs(near_dir(4,:)))./(LTs(near_dir(3,:))-LTs(near_dir(4,:))));
% interpolate x for theta & y for phi
FDP11s = FDP11s1+(FDP11s2-FDP11s1).*((thetas'- ...
LTs(near_dir(2,:)))./(LTs(near_dir(1,:))-LTs(near_dir(2,:))));
FDP22s = FDP22s1+(FDP22s2-FDP22s1).*((phis'- ...
LPs(near_dir(2,:)))./(LPs(near_dir(4,:))-LPs(near_dir(2,:))));
% 2 z-coordinates for theta
FDP33s1 = s33_int_failure(near_dir(2,:))+ ...
(s33_int_failure(near_dir(1,:))- ...
s33_int_failure(near_dir(2,:))).*((thetas'- ...
LTs(near_dir(2,:)))./(LTs(near_dir(1,:))-LTs(near_dir(2,:))));
FDP33s2 = s33_int_failure(near_dir(4,:))+ ...
(s33_int_failure(near_dir(3,:))- ...
s33_int_failure(near_dir(4,:))).*((thetas'- ...
LTs(near_dir(4,:)))./(LTs(near_dir(3,:))-LTs(near_dir(4,:))));
% interpolate z for phi
FDP33s = FDP33s1+(FDP33s2-FDP33s1).*((phis'- ...
LPs(near_dir(2,:)))./(LPs(near_dir(4,:))-LPs(near_dir(2,:))));
%%% determine failure sensitivities
FS = (sqrt(principal_stresses(1,:).^2+principal_stresses(2,:).^2+ ...
principal_stresses(3,:).^2))./(sqrt(FDP11s'.^2+FDP22s'.^2+FDP33s'.^2));
end
```

G.2 Non-Linear Finite Element Method Codes

The implementations provided in what follows consider the developed non-linear FEM as discussed in Section 2.2.

G.2.1 Load Control Codes

The following two non-linear FEM `function` codes use the load control variation of the Newton-Raphson iterative method to trace a structural model's response that exhibits non-linear material behaviour. Similarly, first the two-dimensional implementation will be provided, followed by the three-dimensional one.

```
function [EnergySE_mat,principal_stresses] = ...
LC_FEM_GMA_Plane_Stress(nelx,nely,penal,Fdofs,freedofs,MaxLoad,x,b, ...
h,t,MatTyp,edofMat,iK,jK,GMAinput,Ninc,tolerance,lambda)
%% INPUT PARAMETERS
nele = nelx*nely;                   % number of elements
nIP = 4;                            % number of integration points (1,4,9)
LoadInc = MaxLoad/Ninc;             % load increment [N]
LoadVals = [0:LoadInc:MaxLoad]';    % load values for each increment
NumInc = length(LoadVals);          % number of load increments
%% MATERIAL AND PARAMETER PROPERTIES
nLBD = 5810; % number of Lebedev points (50,110,590,974,1202,5810)
Densities = reshape(x,nele,1);
%% Calculate [B]
[xi,eta,Wp] = CalcIP(nIP);
Bs = zeros(3*nIP,8);
for n = 1:nIP
    x = xi(n)*b/2;
    y = eta(n)*h/2;
    B = [-(1-2*y/h)/(2*b) 0 (1-2*y/h)/(2*b) 0 (1+2*y/h)/(2*b) 0 ...
        -(1+2*y/h)/(2*b) 0; 0 -(1-2*x/b)/(2*h) 0 -(1+2*x/b)/(2*h) 0 ...
        (1+2*x/b)/(2*h) 0 (1-2*x/b)/(2*h); -(1-2*x/b)/(2*h) ...
        -(1-2*y/h)/(2*b) -(1+2*x/b)/(2*h) (1-2*y/h)/(2*b) ...
        (1+2*x/b)/(2*h) (1+2*y/h)/(2*b) (1-2*x/b)/(2*h) ...
        -(1+2*y/h)/(2*b)];
    Bs(3*(n-1)+1:3*n,:) = B;
end
%% Initialize F & U
F = sparse(2*(nely+1)*(nelx+1),1);
U = zeros(2*(nely+1)*(nelx+1),1);
%% PERFORM NON-LINEAR FEA
max_iter = 100;
AllKs = zeros(64,nele);
EnergySE = 0;
Eps_inc112212 = zeros(3*nIP,nele);
Sigmas_inc112212 = zeros(3*nIP,nele);
sigmas = zeros(3*nIP,1);
sigmasOP = zeros(3*nIP,1);
dE332313 = zeros(3*nIP,nele);
Eps_prev332313 = zeros(3*nIP,nele);
EnergySE_ele = zeros(nele,1);
principal_stresses_average = zeros(2,nele);
Allepsprev332313 = zeros(3*nIP,nele);
```

```
nbytes = fprintf('FEM progress: %2.0f%%\n',0);
for inc = 2:NumInc
    k = 0; % reset iteration counter
    lv = LoadVals(inc);
    Eps_prev112212 = Eps_inc112212;
    Eps_prev332313 = Eps_prev332313+dE332313;
    sigmas_prev112212 = Sigmas_inc112212;
    normR = 1;
    F(Fdofs) = lv; %%% input lv
    while normR > tolerance && k < max_iter
    k = k + 1;
    if k == max_iter
    fprintf(...
    '\n ERROR: solution does not converge within %d iterations\n',...
    max_iter)
    nbytes = fprintf('FEM progress: %2.0f%%\n',(inc/NumInc)*100);
    end
        Fi = zeros(2*(nely+1)*(nelx+1),1);
        for ele = 1:nele
            density=Densities(ele,1);
            for n = 1:nIP
                U_el = U(edofMat(ele,:));
                eps112212 = Bs(3*(n-1)+1:3*n,:)*U_el;
                strains112212(3*(n-1)+1:3*n,:) = eps112212;
                epsprev112212 = Eps_prev112212(3*(n-1)+1:3*n,ele);
                epsprev332313 = Eps_prev332313(3*(n-1)+1:3*n,ele)+ ...
                dE332313(3*(n-1)+1:3*n,ele);
                [s_int,Cij_ps,deps332313] = ...
                GMA_Plane_Stress(eps112212,nLBD,epsprev112212, ...
                epsprev332313,density,penal,MatTyp,GMAinput,lambda);
                dE332313(3*(n-1)+1:3*n,ele) = deps332313;
                sigmas(3*(n-1)+1:3*n,:) = s_int([1,2,6]);
                Ds(3*(n-1)+1:3*n,:) = Cij_ps;
            end
            [Fi_el] = CalcFi_nIP(Bs,sigmas,b,h,Wp,nIP,t);
            Fi(edofMat(ele,:)) = Fi(edofMat(ele,:)) + Fi_el;
            [Kij] = CalcKij_nIP(Bs,Ds,b,h,Wp,nIP,t);
            for i = 1:8
                AllKs(8*(i-1)+1:8*i,ele) = Kij(:,i);
            end
        end
        sK = reshape(AllKs,64*nelx*nely,1);
        K = reshape(AllKs,64*nelx*nely,1); K = sparse(iK,jK,sK);
        K = (K+K')/2; % to be sure that it is symmetric
        R = Fi(freedofs) - F(freedofs);
        dU = -K(freedofs,freedofs)\R;
        U(freedofs) = U(freedofs) + dU;
        normR = norm(dU)/norm(U);
    end
    for ele = 1:nele
        density = Densities(ele,1);
        sigmas_average = zeros(3,1);
        for n = 1:nIP
            U_el = U(edofMat(ele,:));
            eps112212 = Bs(3*(n-1)+1:3*n,:)*U_el;
            strains112212(3*(n-1)+1:3*n,:) = eps112212;
            epsprev112212 = Eps_prev112212(3*(n-1)+1:3*n,ele);
            epsprev332313 = Eps_prev332313(3*(n-1)+1:3*n,ele)+ ...
            dE332313(3*(n-1)+1:3*n,ele);
            [s_int,Cij_ps,deps332313] = ...
            GMA_Plane_Stress(eps112212,nLBD,epsprev112212, ...
            epsprev332313,density,penal,MatTyp,GMAinput,lambda);
            dE332313(3*(n-1)+1:3*n,ele) = deps332313;
            sigmas(3*(n-1)+1:3*n,:) = s_int([1,2,6]);
            sigmasOP(3*(n-1)+1:3*n,:) = s_int([3,4,5]);
            sigmas_average = s_int([1,2,6])*Wp(n)/sum(Wp)+ ...
            sigmas_average;
            strains332313(3*(n-1)+1:3*n,:) = epsprev332313;
        end
        Eps_inc112212(:,ele) = strains112212;
        Sigmas_inc112212(:,ele) = sigmas;
        Dsigmas = sigmas-sigmas_prev112212(:,ele);
        DEps_inc = Eps_inc112212(:,ele)-Eps_prev112212(:,ele);
        [EnergySE_nIP] = ...
        CalcEnergySE_nIP(DEps_inc,sigmas,b,h,t,Wp,nIP,Dsigmas);
        EnergySE = EnergySE + sum(EnergySE_nIP);
        EnergySE_ele(ele) = EnergySE_ele(ele) + sum(EnergySE_nIP);
        %%% local stress homogenization:
        sigmas_average = sigmas_average./(density);
        principal_stresses_average(1,ele) = ...
        (sigmas_average(1)+sigmas_average(2))/2 + ...
        sqrt(((sigmas_average(1)-sigmas_average(2))/2)^2+ ...
        sigmas_average(3)^2);
        principal_stresses_average(2,ele) = ...
        (sigmas_average(1)+sigmas_average(2))/2 - ...
        sqrt(((sigmas_average(1)-sigmas_average(2))/2)^2+ ...
        sigmas_average(3)^2);
        Allepsprev332313(:,ele) = strains332313;
    end
    EnergySE_mat = reshape(EnergySE_ele,nely,nelx);
    principal_stresses = sort(principal_stresses_average,'descend');
    fprintf(repmat('\b',1,nbytes))
    nbytes = fprintf('FEM progress: %2.0f%%',(inc/NumInc)*100);
end
fprintf('\n');
end
```

```
function [EnergySE_mat,principal_stresses] = ...
LC_FEM_GMA_3D(nelx,nely,nelz,penal,Fdofs,freedofs,MaxLoad,x,b,h,t, ...
MatTyp,edofMat,iK,jK,GMAinput,Ninc,tolerance,lambda)
%% INPUT PARAMETERS
nele = nelx*nely*nelz;            % number of elements
nIP = 8;                          % number of integration points (1,8,27)
LoadInc = MaxLoad/Ninc;           % load increment [N]
LoadVals = [0:LoadInc:MaxLoad]'; % load values for each increment
NumInc = length(LoadVals);        % number of load increments
%% MATERIAL  AND PARAMETER PROPERTIES
nLBD = 5810; % number of Lebedev points (50,110,590,974,1202,5810)
Densities=reshape(x,nele,1);
%% CALCULATE [B]
[xi,eta,zeta,Wp] = CalcIP_3D(nIP);
[Bs] = B_3D(nIP,xi,eta,zeta,b,h,t);
%% Initialize F & U
ndof = 3*(nelx+1)*(nely+1)*(nelz+1);
F = zeros(ndof,1);
U = zeros(ndof,1);
%% PERFORM NON-LINEAR FEA
max_iter = 100;
AllKs = zeros(576,nele);
EnergySE = 0;
Eps_inc = zeros(6*nIP,nele);
Sigmas_inc = zeros(6*nIP,nele);
sigmas = zeros(6*nIP,1);
EnergySE_ele = zeros(nele,1);
principal_stresses_average = zeros(3,nele);
nbytes = fprintf('FEM progress: %2.0f%%\n',0);
for inc=2:NumInc
    k = 0; % reset iteration counter
    lv = LoadVals(inc);
    Eps_prev = Eps_inc; % epsilons 11,22,12 from previous increment
    sigmas_prev = Sigmas_inc;
    normR = 1;
    F(Fdofs) = lv; %%% input lv
    while normR > tolerance && k < max_iter
    k = k + 1;
    if k == max_iter
    fprintf(...
    '\n ERROR: solution does not converge within %d iterations\n',...
    max_iter)
    nbytes = fprintf('FEM progress: %2.0f%%\n',(inc/NumInc)*100);
    end
        Fi = zeros(ndof,1);
        for ele = 1:nele
            density=Densities(ele,1);
            for n = 1:nIP
                U_el = U(edofMat(ele,:));
                eps = Bs(6*(n-1)+1:6*n,:)*U_el;
                strains(6*(n-1)+1:6*n,:) = eps;
                [s_int,Cij] = ...
                GMA_3D(eps,nLBD,density,penal,MatTyp,GMAinput,lambda);
                sigmas(6*(n-1)+1:6*n,:) = s_int;
                Cs(6*(n-1)+1:6*n,:) = Cij;
            end
            [Fi_el] = CalcFi_nIP_3D(Bs,sigmas,b,h,Wp,nIP,t);
            Fi(edofMat(ele,:)) = Fi(edofMat(ele,:)) + Fi_el;
            [Kij] = CalcKij_nIP_3D(Bs,Cs,b,h,Wp,nIP,t);
            for i = 1:24
                AllKs(24*(i-1)+1:24*i,ele) = Kij(:,i);
            end
        end
        sK = reshape(AllKs,576*nele,1);
        K = reshape(AllKs,576*nele,1); K = sparse(iK,jK,sK);
        K = (K+K')/2; % to be sure that it is symmetric
        R = Fi(freedofs) - F(freedofs);
        dU = -K(freedofs,freedofs)\R;
        U(freedofs) = U(freedofs) + dU;
        normR = norm(dU)/norm(U);
    end
    for ele = 1:nele
        density = Densities(ele,1);
        sigmas_average = zeros(6,1);
        for n = 1:nIP
            U_el = U(edofMat(ele,:));
            eps = Bs(6*(n-1)+1:6*n,:)*U_el;
            strains(6*(n-1)+1:6*n,:) = eps;
            [s_int,Cij] = ...
            GMA_3D(eps,nLBD,density,penal,MatTyp,GMAinput,lambda);
            sigmas(6*(n-1)+1:6*n,:) = s_int;
            sigmas_average = s_int*Wp(n)/sum(Wp) + sigmas_average;
        end
        Eps_inc(:,ele) = strains;
        Sigmas_inc(:,ele) = sigmas;
        Dsigmas = sigmas-sigmas_prev(:,ele);
        DEps_inc = Eps_inc(:,ele)-Eps_prev(:,ele);
        [EnergySE_nIP] = ...
        CalcEnergySE_nIP_3D(DEps_inc,sigmas,b,h,t,Wp,nIP,Dsigmas);
        EnergySE = EnergySE + sum(EnergySE_nIP);
        EnergySE_ele(ele) = EnergySE_ele(ele) + sum(EnergySE_nIP);
        %%% local stress homogenization:
        T3 = [sigmas_average(1) sigmas_average(6) sigmas_average(5);
              sigmas_average(6) sigmas_average(2) sigmas_average(4);
              sigmas_average(5) sigmas_average(4) ...
              sigmas_average(3)]./(density);
        PS = sort(eig(T3));
        principal_stresses_average(1,ele) = PS(3);
        principal_stresses_average(2,ele) = PS(2);
        principal_stresses_average(3,ele) = PS(1);
    end
    EnergySE_mat = reshape(EnergySE_ele,[nely,nelx,nelz]);
    principal_stresses = sort(principal_stresses_average,'descend');
    fprintf(repmat('\b',1,nbytes))
    nbytes = fprintf('FEM progress: %2.0f%%',(inc/NumInc)*100);
end
fprintf('\n');
end
```

G.2.2 Numerical Integration Implementation Codes

The aforementioned algorithms utilize several numerical integration implementations, as discussed in Section 2.2.2. Respectively, the following `function` codes have been developed for finding integration points and weights, internal force vectors, stiffness matrices and compliance using strain energy density.

The following `MATLAB` codes find the location and weights for various numbers of integration points in two-dimensional and three-dimensional FEA, respectively.

```
function [xi,eta,Wp] = CalcIP(nIP)
if nIP == 1
    xi = 0;
    eta = 0;
    Wp = 4;
elseif nIP == 4
    xi = [-1/sqrt(3); 1/sqrt(3); -1/sqrt(3); 1/sqrt(3)];
    eta = [-1/sqrt(3); -1/sqrt(3); 1/sqrt(3); 1/sqrt(3)];
    Wp = [1; 1; 1; 1];
elseif nIP == 9
    xi = [-sqrt(3/5); 0; sqrt(3/5); -sqrt(3/5); 0; sqrt(3/5); ...
    -sqrt(3/5); 0; sqrt(3/5)];
    eta = [-sqrt(3/5); -sqrt(3/5); -sqrt(3/5); 0; 0; 0; sqrt(3/5); ...
    sqrt(3/5); sqrt(3/5)];
    Wp = [25/81; 40/81; 25/81; 40/81; 64/81; 40/81; 25/81; 40/81; ...
    25/81];
end
end
```

```
function [xi,eta,zeta,Wp] = CalcIP_3D(nIP)
if nIP == 1
    xi = 0;
    eta = 0;
    zeta = 0;
    Wp = 8;
elseif nIP == 8
    xi = [-1/sqrt(3); 1/sqrt(3); 1/sqrt(3); -1/sqrt(3); -1/sqrt(3); ...
    1/sqrt(3); 1/sqrt(3); -1/sqrt(3)];
    eta = [-1/sqrt(3); -1/sqrt(3); -1/sqrt(3); -1/sqrt(3); ...
    1/sqrt(3); 1/sqrt(3); 1/sqrt(3); 1/sqrt(3)];
    zeta = [-1/sqrt(3); -1/sqrt(3); 1/sqrt(3); 1/sqrt(3); ...
    -1/sqrt(3); -1/sqrt(3); 1/sqrt(3); 1/sqrt(3)];
    Wp = [1; 1; 1; 1; 1; 1; 1; 1];
elseif nIP == 27
    xi = [-sqrt(3/5); 0; sqrt(3/5); -sqrt(3/5); 0; sqrt(3/5); ...
    -sqrt(3/5); 0; sqrt(3/5); -sqrt(3/5); 0; sqrt(3/5); -sqrt(3/5); ...
    0; sqrt(3/5); -sqrt(3/5); 0; sqrt(3/5); -sqrt(3/5); 0; ...
    sqrt(3/5); -sqrt(3/5); 0; sqrt(3/5); -sqrt(3/5); 0; sqrt(3/5)];
    eta = [-sqrt(3/5); -sqrt(3/5); -sqrt(3/5); -sqrt(3/5); ...
    -sqrt(3/5); -sqrt(3/5); -sqrt(3/5); -sqrt(3/5); -sqrt(3/5); 0; ...
    0; 0; 0; 0; 0; 0; 0; 0; sqrt(3/5); sqrt(3/5); sqrt(3/5); ...
    sqrt(3/5); sqrt(3/5); sqrt(3/5); sqrt(3/5); sqrt(3/5); sqrt(3/5)];
    zeta = [sqrt(3/5); sqrt(3/5); sqrt(3/5); 0; 0; 0; -sqrt(3/5); ...
    -sqrt(3/5); -sqrt(3/5); sqrt(3/5); sqrt(3/5); sqrt(3/5); 0; 0; ...
    0; -sqrt(3/5); -sqrt(3/5); -sqrt(3/5); sqrt(3/5); sqrt(3/5); ...
    sqrt(3/5); 0; 0; 0; -sqrt(3/5); -sqrt(3/5); -sqrt(3/5)];
    Wp = [(5/9)^3; (8/9)*(5/9)^2; (5/9)^3; (8/9)*(5/9)^2; ...
    ((8/9)^2)*(5/9); (8/9)*(5/9)^2; (5/9)^3; (8/9)*(5/9)^2; (5/9)^3; ...
    (8/9)*(5/9)^2; ((8/9)^2)*(5/9); (8/9)*(5/9)^2; ((8/9)^2)*(5/9); ...
    (8/9)^3; ((8/9)^2)*(5/9); (8/9)*(5/9)^2; ((8/9)^2)*(5/9); ...
    (8/9)*(5/9)^2; (5/9)^3; (8/9)*(5/9)^2; (5/9)^3; (8/9)*(5/9)^2; ...
    ((8/9)^2)*(5/9); (8/9)*(5/9)^2; (5/9)^3; (8/9)*(5/9)^2; (5/9)^3];
end
end
```

The next implementations determine the elemental internal force vectors according to Equations (2.51) and (2.52), respectively.

```
function [Fi_el] = CalcFi_nIP(Bs,sigmas,b,h,Wp,nIP,t)
Jacobian = b*h/4;
Bsigma = 0;
for n = 1:nIP
    B = Bs(3*(n-1)+1:3*n,:);
    sigma = sigmas(3*(n-1)+1:3*n,:);
    Bsigma = Bsigma+Wp(n)*B'*sigma;
end
Fi_el = t*Jacobian*Bsigma;
end
```

```
function [Fi_el] = CalcFi_nIP_3D(Bs,sigmas,b,h,Wp,nIP,t)
Jacobian = b*h*t/8;
Bsigma = 0;
for n = 1:nIP
    B = Bs(6*(n-1)+1:6*n,:);
    sigma = sigmas(6*(n-1)+1:6*n,:);
    Bsigma = Bsigma+Wp(n)*B'*sigma;
end
Fi_el = Jacobian*Bsigma;
end
```

The `MATLAB` algorithms presented next construct the element stiffness matrices through Equations (2.47) and (2.49), respectively.

```
function [Kij] = CalcKij_nIP(Bs,Ds,b,h,Wp,nIP,t)
Jacobian = b*h/4;
BDB = 0;
for n = 1:nIP
    B = Bs(3*(n-1)+1:3*n,:);
    D = Ds(3*(n-1)+1:3*n,:);
    BDB = BDB+Wp(n)*B'*D*B;
end
Kij = t*Jacobian*BDB;
end
```

```
function [Kij] = CalcKij_nIP_3D(Bs,Cs,b,h,Wp,nIP,t)
Jacobian = b*h*t/8;
BCB = 0;
for n = 1:nIP
    B = Bs(6*(n-1)+1:6*n,:);
    C = Cs(6*(n-1)+1:6*n,:);
    BCB = BCB+Wp(n)*B'*C*B;
end
Kij = Jacobian*BCB;
end
```

Finally, the following two codes compute the elemental compliance values using strain energy density as provided in Equations (2.55) and (2.56), respectively.

```
function [EnergySE_nIP] = ...
CalcEnergySE_nIP(DEps_inc,sigmas,b,h,t,Wp,nIP,Dsigmas)
Jacobian = b*h/4;
sigmadeps = 0;
for n = 1:nIP
    sigma = sigmas(3*(n-1)+1:3*n,:)-Dsigmas(3*(n-1)+1:3*n,:)/2;
    deps = DEps_inc(3*(n-1)+1:3*n,:);
    sigmadeps = sigmadeps+Wp(n)*sigma.*deps;
end
EnergySE_nIP = t*Jacobian*sigmadeps;
end
```

```
function [EnergySE_nIP] = ...
CalcEnergySE_nIP_3D(DEps_inc,sigmas,b,h,t,Wp,nIP,Dsigmas)
Jacobian = b*h*t/8;
sigmadeps = 0;
for n = 1:nIP
    sigma = sigmas(6*(n-1)+1:6*n,:)-Dsigmas(6*(n-1)+1:6*n,:)/2;
    deps = DEps_inc(6*(n-1)+1:6*n,:);
    sigmadeps = sigmadeps+Wp(n)*sigma.*deps;
end
EnergySE_nIP = Jacobian*sigmadeps;
end
```

Additionally, the three-dimensional implementations find their strain-displacement matrices in a separate `function` which is given below.

```
function [Bs] = B_3D(nIP,xi,eta,zeta,b,h,t)
Bs = zeros(6*nIP,24);
for n = 1:nIP
    x = xi(n)*b/2;
    y = eta(n)*h/2;
    z = zeta(n)*t/2;
    B1 = [-((1-2*y/h)*(1-2*z/t))/(4*b) 0 0;
        0 -((1-2*x/b)*(1-2*z/t))/(4*h) 0;
        0 0 -((1-2*x/b)*(1-2*y/h))/(4*t);
        -((1-2*x/b)*(1-2*z/t))/(4*h) -((1-2*y/h)*(1-2*z/t))/(4*b) 0;
        0 -((1-2*x/b)*(1-2*y/h))/(4*t) -((1-2*x/b)*(1-2*z/t))/(4*h);
        -((1-2*x/b)*(1-2*y/h))/(4*t) 0 -((1-2*y/h)*(1-2*z/t))/(4*b)];
    B2 = [((1-2*y/h)*(1-2*z/t))/(4*b) 0 0;
        0 -((1+2*x/b)*(1-2*z/t))/(4*h) 0;
        0 0 -((1+2*x/b)*(1-2*y/h))/(4*t);
        -((1+2*x/b)*(1-2*z/t))/(4*h) ((1-2*y/h)*(1-2*z/t))/(4*b) 0;
        0 -((1+2*x/b)*(1-2*y/h))/(4*t) -((1+2*x/b)*(1-2*z/t))/(4*h);
        -((1+2*x/b)*(1-2*y/h))/(4*t) 0 ((1-2*y/h)*(1-2*z/t))/(4*b)];
    B3 = [((1+2*y/h)*(1-2*z/t))/(4*b) 0 0;
        0 ((1+2*x/b)*(1-2*z/t))/(4*h) 0;
        0 0 -((1+2*x/b)*(1+2*y/h))/(4*t);
        ((1+2*x/b)*(1-2*z/t))/(4*h) ((1+2*y/h)*(1-2*z/t))/(4*b) 0;
        0 -((1+2*x/b)*(1+2*y/h))/(4*t) ((1+2*x/b)*(1-2*z/t))/(4*h);
        -((1+2*x/b)*(1+2*y/h))/(4*t) 0 ((1+2*y/h)*(1-2*z/t))/(4*b)];
    B4 = [-((1+2*y/h)*(1-2*z/t))/(4*b) 0 0;
        0 ((1-2*x/b)*(1-2*z/t))/(4*h) 0;
        0 0 -((1-2*x/b)*(1+2*y/h))/(4*t);
        ((1-2*x/b)*(1-2*z/t))/(4*h) -((1+2*y/h)*(1-2*z/t))/(4*b) 0;
        0 -((1-2*x/b)*(1+2*y/h))/(4*t) ((1-2*x/b)*(1-2*z/t))/(4*h);
        -((1-2*x/b)*(1+2*y/h))/(4*t) 0 -((1+2*y/h)*(1-2*z/t))/(4*b)];
    B5 = [-((1-2*y/h)*(1+2*z/t))/(4*b) 0 0;
        0 -((1-2*x/b)*(1+2*z/t))/(4*h) 0;
        0 0 ((1-2*x/b)*(1-2*y/h))/(4*t);
        -((1-2*x/b)*(1+2*z/t))/(4*h) -((1-2*y/h)*(1+2*z/t))/(4*b) 0;
        0 ((1-2*x/b)*(1-2*y/h))/(4*t) -((1-2*x/b)*(1+2*z/t))/(4*h);
        ((1-2*x/b)*(1-2*y/h))/(4*t) 0 -((1-2*y/h)*(1+2*z/t))/(4*b)];
    B6 = [((1-2*y/h)*(1+2*z/t))/(4*b) 0 0;
        0 -((1+2*x/b)*(1+2*z/t))/(4*h) 0;
        0 0 ((1+2*x/b)*(1-2*y/h))/(4*t);
        -((1+2*x/b)*(1+2*z/t))/(4*h) ((1-2*y/h)*(1+2*z/t))/(4*b) 0;
        0 ((1+2*x/b)*(1-2*y/h))/(4*t) -((1+2*x/b)*(1+2*z/t))/(4*h);
        ((1+2*x/b)*(1-2*y/h))/(4*t) 0 ((1-2*y/h)*(1+2*z/t))/(4*b)];
    B7 = [((1+2*y/h)*(1+2*z/t))/(4*b) 0 0;
        0 ((1+2*x/b)*(1+2*z/t))/(4*h) 0;
        0 0 ((1+2*x/b)*(1+2*y/h))/(4*t);
        ((1+2*x/b)*(1+2*z/t))/(4*h) ((1+2*y/h)*(1+2*z/t))/(4*b) 0;
        0 ((1+2*x/b)*(1+2*y/h))/(4*t) ((1+2*x/b)*(1+2*z/t))/(4*h);
        ((1+2*x/b)*(1+2*y/h))/(4*t) 0 ((1+2*y/h)*(1+2*z/t))/(4*b)];
    B8 = [-((1+2*y/h)*(1+2*z/t))/(4*b) 0 0;
        0 ((1-2*x/b)*(1+2*z/t))/(4*h) 0;
        0 0 ((1-2*x/b)*(1+2*y/h))/(4*t);
        ((1-2*x/b)*(1+2*z/t))/(4*h) -((1+2*y/h)*(1+2*z/t))/(4*b) 0;
        0 ((1-2*x/b)*(1+2*y/h))/(4*t) ((1-2*x/b)*(1+2*z/t))/(4*h);
        ((1-2*x/b)*(1+2*y/h))/(4*t) 0 -((1+2*y/h)*(1+2*z/t))/(4*b)];
    B = [B1 B2 B3 B4 B5 B6 B7 B8];
    Bs(6*(n-1)+1:6*n,:) = B;
end
end
```

G.3 Granular Micromechanics Approach Codes

The final codes regarding the GMA as described in Section 2.1 are given here.

G.3.1 Determination of Stiffness Tensor Codes

The following two MATLAB codes have been used to determine each material point's stiffness tensor through the GMA as described in Section 2.1. The implementations presented here consider two-dimensional FEA under plane stress conditions and three-dimensional FEA, respectively.

```matlab
function [s_int,Cij_ps,deps332313] = ...
GMA_Plane_Stress(eps112212,nLBD,epsprev112212,epsprev332313,density, ...
penal,MatTyp,GMAinput,lambda)
%% GMA input parameters
if GMAinput == 1 %% H. Jia et al. (2017)
    L0 = 10*10^(-6);  %% [m]
    NumDens_Particles = (10^18)/3; %% [1/m3]
elseif GMAinput == 2 %% R.J.M. Bol (2022)
    L0 = 10^(-4); %% [m]
    NumDens_Particles = 1*10^12; %% [1/m3]
end
MeanCoord = 6; % average number of contacts a particle has
Np0 = NumDens_Particles*MeanCoord/2; % number density of contacts [1/m3]
Np = (density^penal)*Np0; % penalized number density of contacts [1/m3]
L2Np = L0^2*Np; % (shows up in the calculation of stiffness tensor)
%% Finding integration points
if nLBD == 50
    [ xLEB, yLEB, zLEB, wLEB ] = ld0050 ( );
elseif nLBD == 110
    [ xLEB, yLEB, zLEB, wLEB ] = ld0110 ( );
elseif nLBD == 590
    [ xLEB, yLEB, zLEB, wLEB ] = ld0590 ( );
elseif nLBD == 974
    [ xLEB, yLEB, zLEB, wLEB ] = ld0974 ( );
elseif nLBD == 1202
    [ xLEB, yLEB, zLEB, wLEB ] = ld1202 ( );
elseif nLBD == 5810
    [ xLEB, yLEB, zLEB, wLEB ] = ld5810 ( );
end
theta = acos(zLEB);
phi = atan2(yLEB,xLEB);
a20 = 0;
a40 = 0;
xi_distribution = (1+(1/4)*a20*(3*cos(2*theta.*1)+1)+ ...
a40/8*(35*(cos(theta)).^4-30*(cos(theta)).^2+3));
xi = xi_distribution.*wLEB;
ntheta = length(theta);
%% Direction cosines
n1 = cos(theta.*1);
n2 = sin(theta.*1).*cos(phi.*1);
n3 = sin(theta.*1).*sin(phi.*1);
s1 = -sin(theta.*1);
s2 = cos(theta.*1).*cos(phi.*1);
```

```
s3 = cos(theta.*1).*sin(phi.*1);
t1 = zeros(length(theta),1);
t2 = -sin(phi.*1);
t3 = cos(phi.*1);
%% Define solved strains (FEM)
e11 = eps112212(1);
e22 = eps112212(2);
gamma12 = eps112212(3); e12 = gamma12/2;
deps112212 = eps112212-epsprev112212;
%% Define unknown strains (previous state)
e33 = epsprev332313(1);
gamma23 = epsprev332313(2); e23 = gamma23/2;
gamma13 = epsprev332313(3); e13 = gamma13/2;
%% Enforce dsigma332313 = 0
delta1 = L0*(e11*n1+e12*n2+e13*n3);
delta2 = L0*(e12*n1+e22*n2+e23*n3);
delta3 = L0*(e13*n1+e23*n2+e33*n3);
deltan = delta1.*n1+delta2.*n2+delta3.*n3;
deltas = delta1.*s1+delta2.*s2+delta3.*s3;
deltat = delta1.*t1+delta2.*t2+delta3.*t3;
deltaw = (deltas.^2+deltat.^2).^.5;
[kn,kw,fn,fw] = Micro_const_rel(L0,deltan,deltaw,MatTyp,ntheta,lambda);
k11 = kn.*n1.*n1+kw.*(s1.*s1+t1.*t1);
k12 = kn.*n1.*n2+kw.*(s1.*s2+t1.*t2);
k13 = kn.*n1.*n3+kw.*(s1.*s3+t1.*t3);
k21 = kn.*n2.*n1+kw.*(s2.*s1+t2.*t1);
k22 = kn.*n2.*n2+kw.*(s2.*s2+t2.*t2);
k23 = kn.*n2.*n3+kw.*(s2.*s3+t2.*t3);
k31 = kn.*n3.*n1+kw.*(s3.*s1+t3.*t1);
k32 = kn.*n3.*n2+kw.*(s3.*s2+t3.*t2);
k33 = kn.*n3.*n3+kw.*(s3.*s3+t3.*t3);
% calling Stifness_Matrix function
[Cij] = Stifness_Matrix(L2Np, k11, k12, k13, k21, k22, k23, k31, ...
k32, k33, n1, n2, n3, xi);
A11 = Cij(1:2,1:2);
B1 = Cij(1:2,3:5);
A12 = Cij(1:2,6);
D1 = Cij(3:5,1:2);
C = Cij(3:5,3:5);
D2 = Cij(3:5,6);
A21 = Cij(6,1:2);
B2 = Cij(6,3:5);
A22 = Cij(6,6);
D = [D1 D2];
A = [A11 A12; A21 A22];
B = [B1; B2];
deps332313 = -C\D*deps112212; %% deps332313 for dsigma332313 = 0
Cij_ps = A-B*(C\D);
ndir = length(theta);
fs = zeros(ndir,1); ft = zeros(ndir,1);
for check = 1:ndir
    if deltaw(check) ~= 0
        fs(check) = fw(check)*deltas(check)/deltaw(check);
        ft(check) = fw(check)*deltat(check)/deltaw(check);
    end
end
```

```
f1 = fn.*n1+fs.*s1+ft.*t1;
f2 = fn.*n2+fs.*s2+ft.*t2;
f3 = fn.*n3+fs.*s3+ft.*t3;
s11int = L0*Np*(sum(xi.*f1.*n1));
s22int = L0*Np*(sum(xi.*f2.*n2));
s33int = L0*Np*(sum(xi.*f3.*n3));
s23int = L0*Np*((sum(xi.*f2.*n3)+sum(xi.*f3.*n2))/2);
s13int = L0*Np*((sum(xi.*f1.*n3)+sum(xi.*f3.*n1))/2);
s12int = L0*Np*((sum(xi.*f1.*n2)+sum(xi.*f2.*n1))/2);
s_int = [s11int;s22int;s33int;s23int;s13int;s12int];
%% Calculate forces
function [kn,kw,fn,fw] = ...
Micro_const_rel(L0,deltan,deltaw,MatTyp,ntheta,lambda)
    if MatTyp == 1 %%% tension-compression
        knC = 2000; knT = lambda*knC; %% H. Jia et al. (2017)
        kwC = 1000; kwT = lambda*kwC; %% H. Jia et al. (2017)
        fn = zeros(ntheta,1); kn = zeros(ntheta,1);
        fw = zeros(ntheta,1); kw = zeros(ntheta,1);
        for contact = 1:ntheta
            dn = deltan(contact);
            dw = deltaw(contact);
            if dn > 0 %% tension
                fn(contact) = knT*dn; kn(contact) = knT;
                fw(contact) = kwT*dw; kw(contact) = kwT;
            else %% compression
                fn(contact) = knC*dn; kn(contact) = knC;
                fw(contact) = kwC*dw; kw(contact) = kwC;
            end
        end
    elseif MatTyp == 2 %%% bi-linear
    lambda1 = 0.1; % d_y_t/d_y_c-ratio
    lambda2 = 0.1; % kh/k0-ratio
    kn0 = 5*10^6; knh = lambda2*kn0;
    kw0 = kn0*(1/6); kwh = knh*(1/6);
    d_y_c = -0.01*L0; d_y_t = -lambda1*d_y_c; d_y_w = -lambda1*d_y_c/10;
        fn = zeros(ntheta,1); kn = zeros(ntheta,1);
        fw = zeros(ntheta,1); kw = zeros(ntheta,1);
        for contact = 1:ntheta
            dn = deltan(contact);
            if dn >= 0 %% tension
                if dn <= d_y_t %% initial tensile branch
                    fn(contact) = kn0*dn; kn(contact) = kn0;
                else %% hardened tensile branch
                    fn(contact) = kn0*d_y_t+knh*(dn-d_y_t);
                    kn(contact) = knh;
                end
            else %% compression
                if dn > d_y_c %% initial compresssive branch
                    fn(contact) = kn0*dn; kn(contact) = kn0;
                else %% hardened compressive branch
                    fn(contact) = kn0*d_y_c+knh*(dn-d_y_c);
                    kn(contact) = knh;
                end
            end
            dw = deltaw(contact); %% shear
            if dw <= d_y_w %% initial shear branch
```

```
                fw(contact) = kw0*dw; kw(contact) = kw0;
            else %% hardened shear branch
                fw(contact) = kw0*d_y_w+kwh*(dw-d_y_w);
                kw(contact) = kwh;
            end
        end
    end
end
end
```

```
function [s_int,Cij] = ...
GMA_3D(eps,nLBD,density,penal,MatTyp,GMAinput,lambda)
%% GMA input parameters
if GMAinput == 1 %% H. Jia et al. (2017)
    L0 = 10*10^(-6);  %% [m]
    NumDens_Particles = (10^18)/3; %% [1/m3]
elseif GMAinput == 2 %% R.J.M. Bol (2022)
    L0 = 10^(-4); %% [m]
    NumDens_Particles = 1*10^12; %% [1/m3]
end
MeanCoord = 6; % average number of contacts a particle has
Np0 = NumDens_Particles*MeanCoord/2; % number density of contacts [1/m3]
Np = (density^penal)*Np0; % penalized number density of contacts [1/m3]
L2Np = L0^2*Np; % (shows up in the calculation of stiffness tensor)
%% Finding integration points
if nLBD == 50
    [ xLEB, yLEB, zLEB, wLEB ] = ld0050 ( );
elseif nLBD == 110
    [ xLEB, yLEB, zLEB, wLEB ] = ld0110 ( );
elseif nLBD == 590
    [ xLEB, yLEB, zLEB, wLEB ] = ld0590 ( );
elseif nLBD == 974
    [ xLEB, yLEB, zLEB, wLEB ] = ld0974 ( );
elseif nLBD == 1202
    [ xLEB, yLEB, zLEB, wLEB ] = ld1202 ( );
elseif nLBD == 5810
    [ xLEB, yLEB, zLEB, wLEB ] = ld5810 ( );
end
theta = acos(zLEB);
phi = atan2(yLEB,xLEB);
a20 = 0;
a40 = 0;
xi_distribution = (1+(1/4)*a20*(3*cos(2*theta.*1)+1)+ ...
a40/8*(35*(cos(theta)).^4-30*(cos(theta)).^2+3));
xi = xi_distribution.*wLEB;
ntheta = length(theta);
%% Direction cosines
n1 = cos(theta.*1);
n2 = sin(theta.*1).*cos(phi.*1);
n3 = sin(theta.*1).*sin(phi.*1);
s1 = -sin(theta.*1);
s2 = cos(theta.*1).*cos(phi.*1);
s3 = cos(theta.*1).*sin(phi.*1);
t1 = zeros(length(theta),1);
t2 = -sin(phi.*1);
t3 = cos(phi.*1);
%% Define strains (FEM)
e11 = eps(1);
e22 = eps(2);
e33 = eps(3);
gamma23 = eps(4); e23 = gamma23/2;
gamma13 = eps(5); e13 = gamma13/2;
gamma12 = eps(6); e12 = gamma12/2;
%% Find stiffness tensor
delta1 = L0*(e11*n1+e12*n2+e13*n3);
```

```
delta2 = L0*(e12*n1+e22*n2+e23*n3);
delta3 = L0*(e13*n1+e23*n2+e33*n3);
deltan = delta1.*n1+delta2.*n2+delta3.*n3;
deltas = delta1.*s1+delta2.*s2+delta3.*s3;
deltat = delta1.*t1+delta2.*t2+delta3.*t3;
deltaw = (deltas.^2+deltat.^2).^.5;
[kn,kw,fn,fw] = Micro_const_rel(L0,deltan,deltaw,MatTyp,ntheta,lambda);
ndir = length(theta);
fs = zeros(ndir,1); ft = zeros(ndir,1);
for check = 1:ndir
    if deltaw(check) ~= 0
        fs(check) = fw(check)*deltas(check)/deltaw(check);
        ft(check) = fw(check)*deltat(check)/deltaw(check);
    end
end
f1 = fn.*n1+fs.*s1+ft.*t1;
f2 = fn.*n2+fs.*s2+ft.*t2;
f3 = fn.*n3+fs.*s3+ft.*t3;
s11int = L0*Np*(sum(xi.*f1.*n1));
s22int = L0*Np*(sum(xi.*f2.*n2));
s33int = L0*Np*(sum(xi.*f3.*n3));
s23int = L0*Np*((sum(xi.*f2.*n3)+sum(xi.*f3.*n2))/2);
s13int = L0*Np*((sum(xi.*f1.*n3)+sum(xi.*f3.*n1))/2);
s12int = L0*Np*((sum(xi.*f1.*n2)+sum(xi.*f2.*n1))/2);
s_int = [s11int;s22int;s33int;s23int;s13int;s12int];
k11 = kn.*n1.*n1+kw.*(s1.*s1+t1.*t1);
k12 = kn.*n1.*n2+kw.*(s1.*s2+t1.*t2);
k13 = kn.*n1.*n3+kw.*(s1.*s3+t1.*t3);
k21 = kn.*n2.*n1+kw.*(s2.*s1+t2.*t1);
k22 = kn.*n2.*n2+kw.*(s2.*s2+t2.*t2);
k23 = kn.*n2.*n3+kw.*(s2.*s3+t2.*t3);
k31 = kn.*n3.*n1+kw.*(s3.*s1+t3.*t1);
k32 = kn.*n3.*n2+kw.*(s3.*s2+t3.*t2);
k33 = kn.*n3.*n3+kw.*(s3.*s3+t3.*t3);
% calling Stifness_Matrix function
[Cij] = Stifness_Matrix(L2Np, k11, k12, k13, k21, k22, k23, k31, ...
k32, k33, n1, n2, n3, xi);
%% Calculate forces
function [kn,kw,fn,fw] = ...
Micro_const_rel(L0,deltan,deltaw,MatTyp,ntheta,lambda)
    if MatTyp == 1 %%% tension-compression
        knC = 2000; knT = lambda*knC; %% H. Jia et al. (2017)
        kwC = 1000; kwT = lambda*kwC; %% H. Jia et al. (2017)
        fn = zeros(ntheta,1); kn = zeros(ntheta,1);
        fw = zeros(ntheta,1); kw = zeros(ntheta,1);
        for contact = 1:ntheta
            dn = deltan(contact);
            dw = deltaw(contact);
            if dn > 0 %% tension
                fn(contact) = knT*dn; kn(contact) = knT;
                fw(contact) = kwT*dw; kw(contact) = kwT;
            else %% compression
                fn(contact) = knC*dn; kn(contact) = knC;
                fw(contact) = kwC*dw; kw(contact) = kwC;
            end
        end
    elseif MatTyp == 2 %%% bi-linear
    lambda1 = 0.1; % d_y_t/d_y_c-ratio
    lambda2 = 0.1; % kh/k0-ratio
    kn0 = 5*10^6; knh = lambda2*kn0;
    kw0 = kn0*(1/6); kwh = knh*(1/6);
    d_y_c = -0.01*L0; d_y_t = -lambda1*d_y_c; d_y_w = -lambda1*d_y_c/10;
        fn = zeros(ntheta,1); kn = zeros(ntheta,1);
        fw = zeros(ntheta,1); kw = zeros(ntheta,1);
        for contact = 1:ntheta
            dn = deltan(contact);
            if dn >= 0 %% tension
                if dn <= d_y_t %% initial tensile branch
                    fn(contact) = kn0*dn; kn(contact) = kn0;
                else %% hardened tensile branch
                    fn(contact) = kn0*d_y_t+knh*(dn-d_y_t);
                    kn(contact) = knh;
                end
            else %% compression
                if dn>d_y_c %% initial compresssive branch
                    fn(contact) = kn0*dn; kn(contact) = kn0;
                else %% hardened compressive branch
                    fn(contact) = kn0*d_y_c+knh*(dn-d_y_c);
                    kn(contact) = knh;
                end
            end
            dw = deltaw(contact); %% shear
            if dw <= d_y_w %% initial shear branch
                fw(contact) = kw0*dw; kw(contact) = kw0;
            else %% hardened shear branch
                fw(contact) = kw0*d_y_w+kwh*(dw-d_y_w);
                kw(contact) = kwh;
            end
        end
    end
end
end
```

G.3.2 Determination of Failure Data Point Codes

Please note that the presented MATLAB codes in Appendix G.1 for the stress-constrained problem require definition of GMA-based failure envelopes to determine the stress constraint function values. These GMA-based failure envelopes can be constructed by running the following codes first. The first one defines Failure Data Points (FDPs) for two-dimensional optimization under plane stress conditions and the second defines FDPs for three-dimensional optimization.

```
clc; clear all; close all;
%% GMA input parameters (R.J.M. Bol (2022))
L0 = 10^(-4); %% [m]
NumDens_Particles = 1*10^12; %% [1/m3]
MeanCoord = 6; %% average number of contacts a particle has
Np = NumDens_Particles*MeanCoord/2; %% number density of contacts [1/m3]
L2Np = L0^2*Np; %% (shows up in the calculation of stiffness tensor)
%% Finding integration points
[ xLEB, yLEB, zLEB, wLEB ] = ld5810 ( );
theta = acos(zLEB);
phi = atan2(yLEB,xLEB);
a20 = 0;
a40 = 0;
xi_distribution = (1+(1/4)*a20*(3*cos(2*theta.*1)+1)+ ...
a40/8*(35*(cos(theta)).^4-30*(cos(theta)).^2+3));
xi = xi_distribution.*wLEB;
ntheta = length(theta);
%% Direction cosines
n1 = cos(theta.*1);
n2 = sin(theta.*1).*cos(phi.*1);
n3 = sin(theta.*1).*sin(phi.*1);
n = [n1,n2,n3];
s1 = -sin(theta.*1);
s2 = cos(theta.*1).*cos(phi.*1);
s3 = cos(theta.*1).*sin(phi.*1);
s = [s1,s2,s3];
t1 = zeros(length(theta),1);
t2 = -sin(phi.*1);
t3 = cos(phi.*1);
t = [t1,t2,t3];
%% Calculation of initial stiffness tensor determinant
kn = 5*10^6; kw = kn*(1/6); %% initial micro stiffnesses
k11 = kn.*n1.*n1+kw.*(s1.*s1+t1.*t1);
k12 = kn.*n1.*n2+kw.*(s1.*s2+t1.*t2);
k13 = kn.*n1.*n3+kw.*(s1.*s3+t1.*t3);
k21 = kn.*n2.*n1+kw.*(s2.*s1+t2.*t1);
k22 = kn.*n2.*n2+kw.*(s2.*s2+t2.*t2);
k23 = kn.*n2.*n3+kw.*(s2.*s3+t2.*t3);
k31 = kn.*n3.*n1+kw.*(s3.*s1+t3.*t1);
k32 = kn.*n3.*n2+kw.*(s3.*s2+t3.*t2);
k33 = kn.*n3.*n3+kw.*(s3.*s3+t3.*t3);
% calling Stifness_Matrix function
[Cij] = Stifness_Matrix(L2Np, k11, k12, k13, k21, k22, k23, k31, ...
k32,k33, n1, n2, n3, xi);
DetCintact = det(Cij);
```

```
%% Apply stress
sBImax = 1E9; %% maximal bi-axial stress
sBIinc = sBImax/10000; %% bi-axial stress increment size
sBI = [0:sBIinc:sBImax]; %% bi-axial stress increments
NumSteps = length(sBI);
alpha_min = -pi; %% minimal angle (-180 degrees)
alpha_max = pi; %% maximal angle (+180 degrees)
alpha_inc = 1/180*pi; %% angle increment size
alphas = [alpha_min:alpha_inc:alpha_max]'; %% angle increments
NumDir = length(alphas);
alldetC = zeros(NumSteps,NumDir);
s11_int_failure = zeros(NumDir,1);
s22_int_failure = zeros(NumDir,1);
detC_threshold = 0.01; %% loss of original stiffness = 1-detC_threshold
for ink = 1:length(alphas)
    alpha = alphas(ink);
    x = cos(alpha);
    y = sin(alpha);
    % make stress vectors
    s11s = sBI*x;
    s22s = sBI*y;
    s33s = zeros(size(s11s));
    s23s = zeros(size(s11s));
    s13s = zeros(size(s11s));
    s12s = zeros(size(s11s));
    e11s = zeros(size(s11s));
    e22s = zeros(size(s11s));
    e33s = zeros(size(s11s));
    e23s = zeros(size(s11s));
    e13s = zeros(size(s11s));
    e12s = zeros(size(s11s));
    % solve for all steps
    s11_int = zeros(size(e11s));
    s22_int = zeros(size(e11s));
    s33_int = zeros(size(e11s));
    s23_int = zeros(size(s11s));
    s13_int = zeros(size(s11s));
    s12_int = zeros(size(s11s));
    all_fn = zeros(ntheta,length(s11s));
    all_fw = zeros(ntheta,length(s11s));
    all_dn = zeros(ntheta,length(s11s));
    all_dw = zeros(ntheta,length(s11s));
    detC_ratio = 1;
    inc = 2;
    while detC_ratio > detC_threshold
        e11 = e11s(inc-1); e22 = e22s(inc-1); e33 = e33s(inc-1);
        e23 = e23s(inc-1); e13 = e13s(inc-1); e12 = e12s(inc-1);
        delta1 = L0*(e11*n1+e12*n2+e13*n3);
        delta2 = L0*(e12*n1+e22*n2+e23*n3);
        delta3 = L0*(e13*n1+e23*n2+e33*n3);
        deltan = delta1.*n1+delta2.*n2+delta3.*n3;
        deltas = delta1.*s1+delta2.*s2+delta3.*s3;
        deltat = delta1.*t1+delta2.*t2+delta3.*t3;
        deltaw = (deltas.^2+deltat.^2).^.5;
        all_dn(:,inc) = deltan; all_dw(:,inc) = deltaw;
        [kn,kw,fn,fw] = Micro_const_rel(L0,ntheta,deltan,deltaw);
        all_fn(:,inc) = fn; all_fw(:,inc) = fw;
        fs = zeros(length(theta),1);
        ft = zeros(length(theta),1);
        for dw_check=1:length(theta)
            if abs(deltaw(dw_check)) > 10^(-14)
                fs(dw_check) = ...
                fw(dw_check)*deltas(dw_check)/deltaw(dw_check);
                ft(dw_check) = ...
                fw(dw_check)*deltat(dw_check)/deltaw(dw_check);
            end
        end
        f1 = fn.*n1+fs.*s1+ft.*t1;
        f2 = fn.*n2+fs.*s2+ft.*t2;
        f3 = fn.*n3+fs.*s3+ft.*t3;
        s11_int(inc) = sum(f1.*n1.*xi)*L0*Np;
        s22_int(inc) = sum(f2.*n2.*xi)*L0*Np;
        s33_int(inc) = sum(f3.*n3.*xi)*L0*Np;
        s23_int(inc) = sum(f2.*n3.*xi)*L0*Np;
        s13_int(inc) = sum(f1.*n3.*xi)*L0*Np;
        s12_int(inc) = sum(f1.*n2.*xi)*L0*Np;
        k11 = kn.*n1.*n1+kw.*(s1.*s1+t1.*t1);
        k12 = kn.*n1.*n2+kw.*(s1.*s2+t1.*t2);
        k13 = kn.*n1.*n3+kw.*(s1.*s3+t1.*t3);
        k21 = kn.*n2.*n1+kw.*(s2.*s1+t2.*t1);
        k22 = kn.*n2.*n2+kw.*(s2.*s2+t2.*t2);
        k23 = kn.*n2.*n3+kw.*(s2.*s3+t2.*t3);
        k31 = kn.*n3.*n1+kw.*(s3.*s1+t3.*t1);
        k32 = kn.*n3.*n2+kw.*(s3.*s2+t3.*t2);
        k33 = kn.*n3.*n3+kw.*(s3.*s3+t3.*t3);
        % calling Stifness_Matrix function
        [Cij] = Stifness_Matrix(L2Np, k11, k12, k13, k21, k22, k23, ...
        k31, k32, k33, n1, n2, n3, xi);
        detC = det(Cij);
        alldetC(inc,ink) = detC;
        ds11 = s11s(inc)-s11s(inc-1);
        ds22 = s22s(inc)-s22s(inc-1);
        ds33 = s33s(inc)-s33s(inc-1);
        de = Cij\[ds11;ds22;ds33;0;0;0];
        e11s(inc) = e11+de(1);
        e22s(inc) = e22+de(2);
        e33s(inc) = e33+de(3);
        detC_ratio = detC/DetCintact;
        inc = inc+1;
    end
    %%% stresses calculated with filj
    s11_int_failure(ink) = s11_int(inc-1);
    s22_int_failure(ink) = s22_int(inc-1);
    ink
end
%% PLOTTING
figure;
plot(s11_int_failure/-s11_int_failure(1), ...
s22_int_failure./-s11_int_failure(1),'-b.');
xlabel(...
'$\sigma_{1}/f_c$','FontName','Times New Roman','Interpreter','latex')
ylabel(...
'$\sigma_{2}/f_c$','FontName','Times New Roman','Interpreter','latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
daspect([1 1 1])
xlim([-2 0.5])
ylim([-2 0.5])
grid on
figure; %% plot evolution of detC for alpha = -135 degrees
plot(nonzeros(alldetC(:,round((length(alphas)-1)/8))),'b.');
xlabel('Increments','FontName','Times New Roman','Interpreter','latex')
ylabel('$\det(C_{ij})$','FontName','Times New Roman','Interpreter', ...
'latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
set(gca,'XTick',[]);
set(gca,'YTick',[]);
figure; %% plot evolution of detC for alpha = -90 degrees
plot(nonzeros(alldetC(:,round((length(alphas)-1)/4))),'b.');
xlabel('Increments','FontName','Times New Roman','Interpreter','latex')
ylabel('$\det(C_{ij})$','FontName','Times New Roman','Interpreter', ...
'latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
set(gca,'XTick',[]);
set(gca,'YTick',[]);
figure; %% plot evolution of detC for alpha = -45 degrees
plot(nonzeros(alldetC(:,round(3*(length(alphas)-1)/8))),'b.');
xlabel('Increments','FontName','Times New Roman','Interpreter','latex')
ylabel('$\det(C_{ij})$','FontName','Times New Roman','Interpreter', ...
'latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
set(gca,'XTick',[]);
set(gca,'YTick',[]);
figure; %% plot evolution of detC for alpha = 0 degrees
plot(nonzeros(alldetC(:,round((length(alphas)-1)/2))),'b.');
xlabel('Increments','FontName','Times New Roman','Interpreter','latex')
ylabel('$\det(C_{ij})$','FontName','Times New Roman','Interpreter', ...
'latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
set(gca,'XTick',[]);
set(gca,'YTick',[]);
figure; %% plot evolution of detC for alpha = +45 degrees
plot(nonzeros(alldetC(:,round(5*(length(alphas)-1)/8))),'b.');
xlabel('Increments','FontName','Times New Roman','Interpreter','latex')
ylabel('$\det(C_{ij})$','FontName','Times New Roman','Interpreter', ...
'latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
set(gca,'XTick',[]);
set(gca,'YTick',[]);
%% Write data to text file
OUTPUT = [alphas*(180/pi) s11_int_failure s22_int_failure]';
fileID = fopen('FDPs_Plane_Stress','w');
fprintf(fileID,'%6.0f %24.16e %24.16e\n',OUTPUT);
fclose(fileID);
%% Calculate forces (bi-linear force-law)
function [kn,kw,fn,fw]=Micro_const_rel(L0,ntheta,deltan,deltaw)
lambda1 = 0.1; % d_y_t/d_y_c-ratio
lambda2 = 0.1; % kh/k0-ratio
kn0 = 5*10^6; knh=lambda2*kn0;
```

```
kw0 = kn0*(1/6); kwh=knh*(1/6);
d_y_c = -0.01*L0; d_y_t = -lambda1*d_y_c; d_y_w = -lambda1*d_y_c/10;
fn = zeros(ntheta,1); kn = zeros(ntheta,1);
fw = zeros(ntheta,1); kw = zeros(ntheta,1);
for contact = 1:ntheta
    dn = deltan(contact);
    if dn >= 0 %% tension
        if dn <= d_y_t %% initial tensile branch
            fn(contact) = kn0*dn; kn(contact) = kn0;
        else %% hardened tensile branch
            fn(contact) = kn0*d_y_t+knh*(dn-d_y_t); kn(contact) = knh;
        end
    else %% compression
        if dn > d_y_c %% initial compresssive branch
            fn(contact) = kn0*dn; kn(contact) = kn0;
        else %% hardened compressive branch
            fn(contact) = kn0*d_y_c+knh*(dn-d_y_c); kn(contact) = knh;
        end
    end
    dw = deltaw(contact); %% shear
    if dw <= d_y_w %% initial shear branch
        fw(contact) = kw0*dw; kw(contact) = kw0;
    else %% hardened shear branch
        fw(contact) = kw0*d_y_w+kwh*(dw-d_y_w); kw(contact) = kwh;
    end
end
end
```

```
clc; clear all; close all;
%% GMA parameters (R.J.M. Bol (2022))
L0 = 10^(-4); %% [m]
NumDens_Particles = 1*10^12; %% [1/m3]
MeanCoord = 6; %% average number of contacts a particle has
Np = NumDens_Particles*MeanCoord/2; %% number density of contacts [1/m3]
L2Np = L0^2*Np; %% (shows up in the calculation of stiffness tensor)
%% Finding integration points
[ xLEB, yLEB, zLEB, wLEB ] = ld5810 ( );
theta = acos(zLEB);
phi = atan2(yLEB,xLEB);
a20 = 0;
a40 = 0;
xi_distribution = (1+(1/4)*a20*(3*cos(2*theta.*1)+1)+ ...
a40/8*(35*(cos(theta)).^4-30*(cos(theta)).^2+3));
xi = xi_distribution.*wLEB;
ntheta = length(theta);
%% Direction cosines
n1 = cos(theta.*1);
n2 = sin(theta.*1).*cos(phi.*1);
n3 = sin(theta.*1).*sin(phi.*1);
n = [n1,n2,n3];
s1 = -sin(theta.*1);
s2 = cos(theta.*1).*cos(phi.*1);
s3 = cos(theta.*1).*sin(phi.*1);
s = [s1,s2,s3];
t1 = zeros(length(theta),1);
t2 = -sin(phi.*1);
t3 = cos(phi.*1);
t = [t1,t2,t3];
%% Calculation of initial stiffness tensor determinant
kn = 5*10^6; kw = kn*(1/6);
k11 = kn.*n1.*n1+kw.*(s1.*s1+t1.*t1);
k12 = kn.*n1.*n2+kw.*(s1.*s2+t1.*t2);
k13 = kn.*n1.*n3+kw.*(s1.*s3+t1.*t3);
k21 = kn.*n2.*n1+kw.*(s2.*s1+t2.*t1);
k22 = kn.*n2.*n2+kw.*(s2.*s2+t2.*t2);
k23 = kn.*n2.*n3+kw.*(s2.*s3+t2.*t3);
k31 = kn.*n3.*n1+kw.*(s3.*s1+t3.*t1);
k32 = kn.*n3.*n2+kw.*(s3.*s2+t3.*t2);
k33 = kn.*n3.*n3+kw.*(s3.*s3+t3.*t3);
% calling Stifness_Matrix function
[Cij] = Stifness_Matrix(L2Np, k11, k12, k13, k21, k22, k23, k31, ...
k32, k33, n1, n2, n3, xi);
DetCintact = det(Cij);
%% Apply stress
sBImax = 1E10; %% maximal bi-axial stress
sBIinc = sBImax/10000; %% bi-axial stress increment size
sBI = [0:sBIinc:sBImax]; %% bi-axial stress increments
NumSteps = length(sBI);
ntheta_dir = 36;
nphi_dir = 72; %% 360/nphi_dir should yield a whole number
[Rho1,theta1,phi1,W1] = spherequad(1,ntheta_dir,nphi_dir,1);
%%% store uniaxial compression ink
f_yc = ntheta_dir*nphi_dir/2+ceil(ntheta_dir/2);
```

```
LTs = theta1;
LPs = phi1;
NumDir = length(LTs);
s11_int_failure = zeros(NumDir,1);
s22_int_failure = zeros(NumDir,1);
s33_int_failure = zeros(NumDir,1);
detC_threshold = 0.01; %% loss of original stiffness = 1-detC_threshold
for ink = 1:NumDir
    LT = LTs(ink); LP = LPs(ink);
    x = sin(LT)*cos(LP);
    y = sin(LT)*sin(LP);
    z = cos(LT);
    % make stress vectors
    s11s = sBI*x;
    s22s = sBI*y;
    s33s = sBI*z;
    s23s = zeros(size(s11s));
    s13s = zeros(size(s11s));
    s12s = zeros(size(s11s));
    e11s = zeros(size(s11s));
    e22s = zeros(size(s11s));
    e33s = zeros(size(s11s));
    e23s = zeros(size(s11s));
    e13s = zeros(size(s11s));
    e12s = zeros(size(s11s));
    % solve for all steps
    s11_int = zeros(size(e11s));
    s22_int = zeros(size(e11s));
    s33_int = zeros(size(e11s));
    s23_int = zeros(size(s11s));
    s13_int = zeros(size(s11s));
    s12_int = zeros(size(s11s));
    all_fn = zeros(ntheta,length(s11s));
    all_fw = zeros(ntheta,length(s11s));
    all_dn = zeros(ntheta,length(s11s));
    all_dw = zeros(ntheta,length(s11s));
    detC_ratio = 1;
    inc = 2;
    while detC_ratio > detC_threshold
        e11 = e11s(inc-1); e22 = e22s(inc-1); e33 = e33s(inc-1);
        e23 = e23s(inc-1); e13 = e13s(inc-1); e12 = e12s(inc-1);
        delta1 = L0*(e11*n1+e12*n2+e13*n3);
        delta2 = L0*(e12*n1+e22*n2+e23*n3);
        delta3 = L0*(e13*n1+e23*n2+e33*n3);
        deltan = delta1.*n1+delta2.*n2+delta3.*n3;
        deltas = delta1.*s1+delta2.*s2+delta3.*s3;
        deltat = delta1.*t1+delta2.*t2+delta3.*t3;
        deltaw = (deltas.^2+deltat.^2).^.5;
        all_dn(:,inc) = deltan; all_dw(:,inc) = deltaw;
        [kn,kw,fn,fw] = Micro_const_rel(L0,ntheta,deltan,deltaw);
        all_fn(:,inc) = fn; all_fw(:,inc) = fw;
        fs = zeros(length(theta),1);
        ft = zeros(length(theta),1);
        for dw_check = 1:length(theta)
            if abs(deltaw(dw_check)) > 10^(-14)
                fs(dw_check) = ...
                fw(dw_check)*deltas(dw_check)/deltaw(dw_check);
                ft(dw_check) = ...
                fw(dw_check)*deltat(dw_check)/deltaw(dw_check);
            end
        end
        f1 = fn.*n1+fs.*s1+ft.*t1;
        f2 = fn.*n2+fs.*s2+ft.*t2;
        f3 = fn.*n3+fs.*s3+ft.*t3;
        s11_int(inc) = sum(f1.*n1.*xi)*L0*Np;
        s22_int(inc) = sum(f2.*n2.*xi)*L0*Np;
        s33_int(inc) = sum(f3.*n3.*xi)*L0*Np;
        s23_int(inc) = sum(f2.*n3.*xi)*L0*Np;
        s13_int(inc) = sum(f1.*n3.*xi)*L0*Np;
        s12_int(inc) = sum(f1.*n2.*xi)*L0*Np;
        k11 = kn.*n1.*n1+kw.*(s1.*s1+t1.*t1);
        k12 = kn.*n1.*n2+kw.*(s1.*s2+t1.*t2);
        k13 = kn.*n1.*n3+kw.*(s1.*s3+t1.*t3);
        k21 = kn.*n2.*n1+kw.*(s2.*s1+t2.*t1);
        k22 = kn.*n2.*n2+kw.*(s2.*s2+t2.*t2);
        k23 = kn.*n2.*n3+kw.*(s2.*s3+t2.*t3);
        k31 = kn.*n3.*n1+kw.*(s3.*s1+t3.*t1);
        k32 = kn.*n3.*n2+kw.*(s3.*s2+t3.*t2);
        k33 = kn.*n3.*n3+kw.*(s3.*s3+t3.*t3);
        % calling Stifness_Matrix function
        [Cij] = Stifness_Matrix(L2Np, k11, k12, k13, k21, k22, k23, ...
        k31, k32, k33, n1, n2, n3, xi);
        detC = det(Cij);
        ds11 = s11s(inc)-s11s(inc-1);
        ds22 = s22s(inc)-s22s(inc-1);
        ds33 = s33s(inc)-s33s(inc-1);
        de = Cij\[ds11;ds22;ds33;0;0;0];
        e11s(inc) = e11+de(1);
        e22s(inc) = e22+de(2);
        e33s(inc) = e33+de(3);
        detC_ratio = detC/DetCintact;
        inc = inc+1;
    end
    %%% stresses calculated with filj
    s11_int_failure(ink) = s11_int(inc-1);
    s22_int_failure(ink) = s22_int(inc-1);
    s33_int_failure(ink) = s33_int(inc-1);
    ink
end
%% PLOTTING
ds = sqrt(s11_int_failure.^2+s22_int_failure.^2+s33_int_failure.^2);
figure; %% 3D
scatter3(s11_int_failure/-s11_int_failure(f_yc), ...
s22_int_failure/-s11_int_failure(f_yc), ...
s33_int_failure/-s11_int_failure(f_yc),5,ds,'filled')
colormap(jet);
xlabel(...
'$\sigma_{1}/f_c$','FontName','Times New Roman','Interpreter','latex')
ylabel(...
'$\sigma_{2}/f_c$','FontName','Times New Roman','Interpreter','latex')
zlabel(...
'$\sigma_{3}/f_c$','FontName','Times New Roman','Interpreter','latex')
```

```
set(gca,'FontSize',20,'FontName','Times New Roman')
daspect([1 1 1])
xlim([-2.5 0.5])
ylim([-2.5 0.5])
zlim([-2.5 0.5])
grid on
view(45,45)
figure; %% Y-X
scatter3(s11_int_failure/-s11_int_failure(f_yc), ...
s22_int_failure/-s11_int_failure(f_yc), ...
s33_int_failure/-s11_int_failure(f_yc),5,ds,'filled')
colormap(jet);
xlabel(...
'$\sigma_{1}/f_c$','FontName','Times New Roman','Interpreter','latex')
ylabel(...
'$\sigma_{2}/f_c$','FontName','Times New Roman','Interpreter','latex')
zlabel(...
'$\sigma_{3}/f_c$','FontName','Times New Roman','Interpreter','latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
daspect([1 1 1])
xlim([-2.5 0.5])
ylim([-2.5 0.5])
zlim([-2.5 0.5])
grid on
view(2)
figure; %% Z-X
scatter3(s11_int_failure/-s11_int_failure(f_yc), ...
s22_int_failure/-s11_int_failure(f_yc), ...
s33_int_failure/-s11_int_failure(f_yc),5,ds,'filled')
colormap(jet);
xlabel(...
'$\sigma_{1}/f_c$','FontName','Times New Roman','Interpreter','latex')
ylabel(...
'$\sigma_{2}/f_c$','FontName','Times New Roman','Interpreter','latex')
zlabel(...
'$\sigma_{3}/f_c$','FontName','Times New Roman','Interpreter','latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
daspect([1 1 1])
xlim([-2.5 0.5])
ylim([-2.5 0.5])
zlim([-2.5 0.5])
grid on
view(0,0);
figure; %% Z-Y
scatter3(s11_int_failure/-s11_int_failure(f_yc), ...
s22_int_failure/-s11_int_failure(f_yc), ...
s33_int_failure/-s11_int_failure(f_yc),5,ds,'filled')
colormap(jet);
xlabel(...
'$\sigma_{1}/f_c$','FontName','Times New Roman','Interpreter','latex')
ylabel(...
'$\sigma_{2}/f_c$','FontName','Times New Roman','Interpreter','latex')
zlabel(...
'$\sigma_{3}/f_c$','FontName','Times New Roman','Interpreter','latex')
set(gca,'FontSize',20,'FontName','Times New Roman')
daspect([1 1 1])
```

```
xlim([-2.5 0.5])
ylim([-2.5 0.5])
zlim([-2.5 0.5])
grid on
view(90,0)
%% Write data to text file
OUTPUT = [LTs*(180/pi) LPs*(180/pi) s11_int_failure ...
s22_int_failure s33_int_failure]';
fileID = fopen('FDPs_3D.txt','w');
fprintf(fileID,'%6.0f %24.16e %24.16e\n',OUTPUT);
fclose(fileID);
%% Calculate forces (bi-linear force-law)
function [kn,kw,fn,fw]=Micro_const_rel(L0,ntheta,deltan,deltaw)
lambda1 = 0.1; % d_y_t/d_y_c-ratio
lambda2 = 0.1; % kh/k0-ratio
kn0 = 5*10^6; knh = lambda2*kn0;
kw0 = kn0*(1/6); kwh = knh*(1/6);
d_y_c = -0.01*L0; d_y_t = -lambda1*d_y_c; d_y_w = -lambda1*d_y_c/10;
fn = zeros(ntheta,1); kn = zeros(ntheta,1);
fw = zeros(ntheta,1); kw = zeros(ntheta,1);
for contact = 1:ntheta
    dn = deltan(contact);
    if dn >= 0 %% tension
        if dn <= d_y_t %% initial tensile branch
            fn(contact) = kn0*dn; kn(contact) = kn0;
        else %% hardened tensile branch
            fn(contact) = kn0*d_y_t+knh*(dn-d_y_t); kn(contact) = knh;
        end
    else %% compression
        if dn > d_y_c %% initial compresssive branch
            fn(contact) = kn0*dn; kn(contact) = kn0;
        else %% hardened compressive branch
            fn(contact) = kn0*d_y_c+knh*(dn-d_y_c); kn(contact) = knh;
        end
    end
    dw = deltaw(contact); %% shear
    if dw <= d_y_w %% initial shear branch
        fw(contact) = kw0*dw; kw(contact) = kw0;
    else %% hardened shear branch
        fw(contact) = kw0*d_y_w+kwh*(dw-d_y_w); kw(contact) = kwh;
    end
end
end
```