

Topology optimisation with the discrete element method



Connor Michael O'Shaughnessy

Department of Civil and Geospatial Engineering
Newcastle University

This dissertation is submitted for the degree of
Doctor of Philosophy

Abstract

The drive towards sustainability in engineering system design has renewed interest in topology optimisation as a method to maximise performance while minimizing waste. Typically these methods employ continuum-based analyses using numerical techniques, such as finite elements, to quantify performance. Whilst this approach is efficient for linear elastic systems, non-linearity adds complexity and discontinuous behaviours including rigid body motion cannot be included due to singularity in the stiffness matrix. As a result inherently discontinuous processes such as material fragmentation, powder-based 3D printing, and granular mechanics in general have not benefited from the development of topology optimisation. This thesis proposes an original approach of coupling penalisation-based topology optimisation with computational simulations using the discrete element method. In the penalisation-based approach the stiffness of individual finite elements are scaled based on a penalised element density variable. Here the proposed adaptation is derived from a scaling of interaction forces and potentials between interacting particles. This formulation is developed into a complete topology optimisation framework including analytical and numerical definitions for sensitivity and the formulation of a filtering technique. This new methodology is first implemented in a simple, proof-of-concept, 2D implementation, for validation against well-known cases from the continuum regime, such as simply supported beams, and columnar systems. These systems are discretised as lattices of particles connected by harmonic springs; at this validation stage bond breakage is not allowed, but some cases involve geometric and material non-linearity, which the new method captures already in its basic formulation are shown. The method is then implemented in combination with a state-of-the-art, 3D simulator for particle based mechanics. This generalised implementation provides flexibility to define complex objectives for the optimisation and enables the incorporation of fully discontinuous behaviours and rigid motion. Examples are presented showing the incorporation of discontinuous processes such as the maximisation of fragmentation energy under impact in a beam and the optimal design of granular systems.

To my wife Arailym, for her patience during the process of writing this thesis, and for providing
the inspiration to always keep going.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 265,000 words including appendices, bibliography, footnotes, tables and equations and has than 150 figures.

Connor Michael O'Shaughnessy

submission: March 2023

Acknowledgements

I would like to acknowledge my supervisors, Dr Enrico Maseoro, for sparking my interest in optimal structural design, for developing in me a considered attention to detail and a research skill set that will stay with me in all my future endeavors, and for his countless hours of collaborative work and unwavering patience throughout. Dr Sadegh Nadimi for introducing me to a wide research community, his consistent support, guidance, and for pushing me to make this thesis the best it could be. Professor Peter Gosling for his oversight and belief in me to succeed. My friends and academic colleges with whom I have discussed this research and who contributed by inspiring new ideas or ways of thinking in me.

I would also like to acknowledge the Newcastle University rocket high performance computing team for providing an excellent service to all users, making this work a possibility with their easy to use documentation online and LAMMPS developer and user community for providing high quality software free of charge, as well as brilliant documentation and learning resources that were essential to this thesis.

Table of Contents

List of Figures	xiii
List of Tables	xxi
Nomenclature	xxiii
1 Introduction	1
2 Literature Review	5
2.1 Topology optimisation	5
2.1.1 Historical development	6
2.1.2 Solid Isotropic Material with Penalisation method	8
2.1.3 Development and Applications	12
2.2 The Discrete Element Method	19
2.3 Optimisation algorithms	25
3 A method of Discrete Element Topology Optimisation	29
3.1 Simple DETO formulation to maximise stiffness	29
3.1.1 Software Implementation: DETO_2D	35
3.2 Generalised Discrete Element Topology Optimisation	40
3.2.1 Software Implementation: DETO_3D	44
3.2.2 Worked Example	49
4 Proof of concept results: quasi static beam problems	61
4.1 Validation	61
4.1.1 Effect of the penalisation exponent p	64
4.2 Parametric Study	66

Table of Contents

4.2.1	Volume fraction	66
4.2.2	Filtering length	67
4.2.3	Mesh resolution	69
4.3	Geometric non-linearity under large displacements	70
4.3.1	Full sensitivities computation with finite difference approach	72
4.4	Material non-linearity	77
4.4.1	Interaction potentials	77
4.4.2	Three-support system with imposed displacement from the top	79
4.4.3	Three-support system with mid support settlement	84
4.4.4	Doubly fixed beam	88
5	Extended result: Discontinuous system optimisation	93
5.1	Validation	93
5.2	Parallelisation study	94
5.3	3D optimisation	97
5.4	Multiple load case optimisation	97
5.5	Multiple objective optimisation	101
6	Conclusions and outlook	107
	Appendix A Example input for DETO_3D	111
	References	135

List of Figures

1.1	a) One of the earliest historical approaches to analytically defined optimal truss design in Michell (1904) b) An example of a modern three dimensional optimal design taking advantage of the principles of TO and the numerical technique of the FEM (Gupta et al., 2020)	1
2.1	Example of three categories of structural optimisation from (Sigmund, 2001) a) Sizing of predefined truss elements b) Shape optimisation of cellular beam c) Topology optimisation of structural layout.	6
2.2	a) Example of a 2D simply supported beam optimisation using a 2D MATLAB script (Sigmund, 2001) with a size of 75x25 square elements b) Example of half a 3D simply supported beam optimisation with a periodic boundary condition in python and utilising the FEniCS C++ finite element library (Gupta et al., 2020) with a size of 20x8x8	12
2.3	Example of the catenary forces that may arise due to deformation in a simply supported beam.	14
2.4	Cantilever beam example optimised using geometrically non-linear finite elements from (Buhl et al., 2000). a) shows the boundary conditions used, supports and location of applied force whilst b) shows the resulting optimal topologies from these boundary conditions with varying intensities of force $\mathbf{F}$ applied. . .	15
2.5	a) Initial configuration of plate with square crack from (Challis et al., 2007) b) Optimised result showing the expected result of circular crack leading to the lowest stress concentrations and therefore highest fracture resistance.	17
2.6	a) Optimised structures from (Wu et al., 2022) for dynamic impact and compliance showing final crack configuration. b) Fracture energy curves for both structures.	18

2.7	a) Structures from (Jansen et al., 2014) optimised utilising systematic element removal in each case the size of the damage scenarios is shown by the blue square, damage is simulated by positioning the blue square centered on an element in the mesh and removing all material from that region, this is repeated for each individual element to compute a combined sensitivity b) Structures from (Wu et al., 2018) optimised using a local volume constraint of 0.6 in each case the radius over which the constraint is applied is shown by the gray circle.	20
2.8	Simple illustrative example of gradient descent methods on a quadratic function using the steepest descent (red) and Conjugate gradient (blue) methods.	22
2.9	a) Structural failure of a building modeled with predefined breakable bonds b) Granular particle deposition using force field style interactions.	23
2.10	a) A conventional Rheometer blade featuring a helical structure b) A DEM model of a particle bed set up and Rheometer blade from (Nan et al., 2017b)	24
2.11	An idealised graph showing the evolution of an objective function across a one dimensional fitness landscape, featuring several prominent local minima.	26
3.1	a) Continuum beam structure modeled as a hexagonal packed lattice of discrete elements b) Schematic of the interaction force between two particles emerging from a harmonic interaction potential.	30
3.2	Hexagonal close packing of DE disks with linear elastic interaction potentials between immediate neighbors	36
3.3	Flow chart of the DETO algorithm implementation.	37
3.4	Example of a system that motivate the development of DETO_3D a simply supported beam undergoing fracture due to a soft impact scenario, where an optimal geometry may exhibit maximum resistance to damage.	40
3.5	Schematic of the 6 degrees of freedom between to interacting particles.	41
3.6	Basic domain decomposition scheme for sharing a simulation between processors in LAMMPS, the black lines are the processor grid and particles are handled by the processor assigned to their grid square, the green dotted line represents the skin for communicating ghost atoms see the official LAMMPS documentation for further details (Plimpton, 1995)	45

3.7	Schematic of the main optimisation loop in DETO_3D, specifying the main high level objects in each optimisation	47
3.8	Visual representation of particle type at step 1 and step 150 of an optimisation where the darker black corresponds to a higher type and therefore greater chi value	58
3.9	Benchmark example of the L-Shape problem from Valdez et al. (2017) Problem 4.5. showing the boundary conditions and optimised result	58
3.10	Visual representation of per particle of hydrostatic and deviatoric stress distribution in the optimised structure	59
4.1	Optimal structures for the beam problem under point load. a) Simply supported system with b) result from the FEM-based TO code in (Sigmund, 2001), with inputs: $nelx = 67$, $nely = 39$, $volfrac = 0.6$, $penal = 3$ and $rmin = 2$ (Sigmund, 2001) this example is mirrored across it's central horizontal axis hence the approximately halved $nelx$ value, and c) result from DETO. d) Pin-supported system with results from e) Michell's analysis in (Michell, 1904) and f) DETO which has the dimension inputs $nelx = 109$, $nely = 45$	62
4.2	Simply supported beam case in Fig. 4.1.a: evolution of geometry and objective function c during the first 100 optimisation steps.	63
4.3	Hydrostatic and Von Mises deviatoric stress distributions in the optimized structure from Fig. 4.1.c.	64
4.4	optimisation of a beam domain, made of 75×25 particles, with $D = 1$ and filtering of 1.1 applied, simply supported beam at the bottom left and right ends. Three values of penalization factors p are investigated. The snapshot show the solutions at optimisation step 100.	65
4.5	Convergence of the optimisation problems in Fig. 4.4. The fits are obtained using the exponential function discussed in the main body of the text.	66
4.6	a) Effect of target solid volume fraction f on optimized solutions to the simply supported beam problem in Fig. 4.1.a. b) Effect of volume fraction on objective function	67

4.7	a) Effect of filtering length on optimized geometries. For generality, the values of r_{min} are given in units of particle diameters D . The label <i>off</i> indicates the unfiltered case. b) Effect of filter length on objective function	68
4.8	a) Effect of mesh resolution on optimum geometries: filtering included with length $r_{min} = 1.5$ mm in all cases. b) Effect of mesh resolution on the evolution of c during the optimisation process	70
4.9	Study on mesh effects for the same beam structures as in Fig. 4.9, but here without imposing any filtering. Results from our DE Topology optimisation method are compared to analogous results from the FEM-based optimisation code in (Sigmund, 2001).	71
4.10	Solutions of topology optimisation problems highlighting the impact of geometric nonlinearity. a) Simply-supported and e) pinned systems, with supports and forces applied to the central axis of the beam; (b,f) solutions from linear elastic FEM-based TO using the code in (Sigmund, 2001), with inputs: $nelx = 72$, $nely = 39$, $volfrac = 0.6$, $penal = 3$ and $rmin = 2$ (see (Sigmund, 2001) for details on the meaning of those inputs); (c,g) solutions from the DETO, which naturally accounts for geometric nonlinearity; (d,h) spatial distribution of hydrostatic stresses for the DETO solutions, identifying the elements working in tension (blue) and in compression (red).	72
4.11	Distribution of bond strain in the optimized structures with pinned and roller supports.	73

4.12	optimisation of beam made of 45×15 particles, simply supported and loaded as shown in Fig. 4.10a. A load of 0.2 kN is applied, which leads to small deformations and an overall symmetric solution. Optimisation snapshots at steps 1, 15, and 30 are shown, for full sensitivities computed with the numerical perturbation method in a) and for the approximate sensitivity in Eq. 3.9 in b). Although the difference between a) and b) are too marginal to be represented visually for the same steps, the histograms quantify the difference between per-particle sensitivities obtained with the two methods (Eq. 4.2) are shown next to the diagrams confirming that indeed the percentage difference is indeed very minor.	74
4.13	optimisation of beam made of 45×15 particles, simply supported and loaded as shown in Fig. 4.10a. Here a load of 2 kN is applied, which is sufficient to highlight the effect of large deformations. Sensitivities computed with the numerical perturbation method in a) and with the approximate sensitivity method in Eq. 3.9 in b).	74
4.14	optimisation of 45×15 beam, simply supported and loaded as shown in Fig. 4.10a. Here a high load of 20 kN is applied, which causes visibly large deformations. Sensitivities computed with the numerical perturbation method in a) and with the approximate sensitivity method in Eq. 3.9 in b).	75
4.15	Spatial distribution of normalised difference in sensitivity for the structures in Figs. 4.12 & 4.13 showing the transition from an initially even spread of difference to concentrated error values in the solid-void border as the optimisation progresses.	75
4.16	a) Approximate and full mean sensitivity values during the first 30 steps of the optimisation showing a fast decrease in all three cases. b) Evolution of objective function (complementary work) for the problem in Fig. 4.14 (with 20 kN of applied load) solved using the full perturbation method and the approximate sensitivity in Eq. 3.9.	76

4.17	a) Interaction potentials from Table. 4.3 for some of the parameters used: $k_0 = 100 \text{ kN}$, $D = 1 \text{ mm}$, $a = 400 \text{ mm}^{-1}$; b) Corresponding force-elongation curves from Table. 4.3, which are proportional to the stress-strain behaviors of the materials.	78
4.18	optimisation problem for a beam on three-supports with imposed displacement at point A at midspan. The value of $\mathbf{u}_{imp}$ has been fine-tuned to obtain an appreciable impact of material nonlinearity.	79
4.19	optimisation results for the three-support system in Fig. 4.18 with target solid fraction $f = 0.3$, assuming a) linear elastic and b) symmetrically strain-hardening material, as per <i>Lin</i> and <i>Weak</i> expressions in Table. 4.3. The colors represent the intensity of local von Mises stress; c) Evolution of the objective function U_{tot} , the total strain energy of the systems during the optimisation. The base case with inputs in Table. 4.1 is compared with cases with no filtering and with smaller $\dot{u}_{imp}$; d) Evolution of force–displacement curves during the optimisation.	80
4.20	Comparison between DETO results a),b) for the same set up as Fig. 4.19 (except on a 30x 15 element system with a particle diameter of 1) and the original results for the non-linear FEM structure c), d) published in Maute et al. (1998a).	82
4.21	Force–displacement curves for the configuration in Fig. 4.19.b assuming linear elastic material (<i>Lin from Weak</i>), and for the configuration in Fig. 4.19.a assuming nonlinear material (<i>Weak from Lin</i>). The curves are compared with the base cases for linear and nonlinear materials already shown in Fig. 4.19.d (solid curves). All curves here were obtained using loading rate $\dot{u}_{imp} = 10^{-5} \text{ mm } \mu\text{s}^{-1}$	83
4.22	optimisation problem for a beam on three supports with imposed settlement of the mid support.	84
4.23	Optimum geometries for the problem in Fig. 4.22, with solid fraction $f = 0.25$ and for different material behaviors as per Table. 4.3: a) linear elastic, b) symmetrically strain-hardening, c) strain-hardening in compression and strain-stiffening in tension, and d) strain-hardening in tension and stiffening in compression. The colors represent the intensity of the deviatoric von Mises stress.	85

4.24	Force-displacement curves for the structures in Fig. 4.23, each with their respective material behavior.	86
4.25	Force-displacement curves for different material behaviors and for structures originally optimized assuming the following material types: a) linear elastic <i>Lin</i> , b) symmetrically strain-hardening <i>Weak</i> , c) strain-hardening in compression and stiffening in tension <i>Weak-C</i> , and d) strain-hardening in tension and stiffening in compression <i>Weak-T</i> . The snapshots of the optimized structures are identical to those in Fig. 4.23.	87
4.26	optimisation problem for a beam partly fixed on both ends and with imposed settlement at the mid point.	88
4.27	Optimum geometries for the problem in Fig. 4.26, with solid fraction $f = 0.4$ and for different material behaviors as per Table. 4.3: a) linear elastic, b) symmetrically strain-hardening, c) strain-hardening in compression and strain-stiffening in compression, and d) strain-hardening in tension and stiffening in compression. The colors represent the intensity of the deviatoric von Mises stress.	89
4.28	Preliminary force displacement curves for the doubly fixed beam case.	90
4.29	Force-displacement curves for different material behaviors and for structures originally optimized assuming the following material types: a) linear elastic <i>Lin</i> , b) symmetrically strain-hardening <i>Weak</i> , c) strain-hardening in compression and stiffening in tension <i>Weak-C</i> , and d) strain-hardening in tension and stiffening in compression <i>Weak-T</i> . The snapshots of the optimized structures are identical to those in Fig. 4.23.	91
4.30	Example from Liu and Qiao (2011) showing an optimisation of a bridge deck structure using 3 different values of R the ratio of tensile to compressive modulus.	92
5.1	Validation of DETO_3D code, for a simply supported beam structure of 135x45 hexagonally meshed elements	94
5.2	Step duration across an optimisation of a 45x15 element beam via a finite difference method a) a single sub-communicator allocated a varying number of processors. b) 12 processors distributed across a varying number of sub-communicators.	96

5.3	Example of a 3D optimised cantilever beam made up from a 75x45x25 hexagonal close packed (HCP) lattice of particles. a) boundary conditions of the problem b) the optimal design c) representation of the hydrostatic stress in the optimal structure.	98
5.4	Illustrative example of a multi simulation optimisation on a cantilever beam result in c) is a combination the load cases from a) and b) into a single optimisation process. d) shows the progression of the cost functions in each case over the duration of the optimisation.	100
5.5	Boundary conditions for both simulations making up the combined fracture energy minimisation study sim 1) representing a soft impact of an applied velocity of 2m/s over an area of 6mm x 2mm at the top of the beam and sim 2) an applied force of 0.25N applied as a point load also at the top of the beam. . .	102
5.6	Optimisation for resistance to a dynamic impact scenario, a) shows an example of a truss system optimised for compliance only and the final system configuration after a dynamic impact as well as the system energy over the course of the simulation. b) shows a system optimised instead using a combined fracture-compliance objective as well as its final configuration and energy. c) shows a degenerated optimisation that is optimised only fracture energy minimisation .	104
5.7	Fracture energy plotted across the duration of a impact simulation, for both the fracture and compliance optimised systems for an impact velocity of 2m/s. . . .	105
5.8	Fractured final configurations of the two proposed designs with particles highlighted in red if they have one or more broken bonds which allows for the visualisation of the fracture pattern in the damaged structures.	106

List of Tables

3.1	Example of a typical chimap used in DETO	48
3.2	Example <i>chimap</i> particles are loaded in by assigning them to their closest value in the χ column, then setting type and any other defined parameters to the corresponding values from that row.	48
4.1	Input parameters for the DEM system properties used in Fig. 4.1. and throughout this section unless otherwise specified	61
4.2	Constants used for the fits in Fig. 4.5	66
4.3	Interaction potentials for the case of linear elastic materials (<i>Lin</i>), symmetric strain-hardening material in tension and in compression (<i>Weak</i>), asymmetric material hardening in tension and stiffening in compression (<i>Weak-T</i>), and asymmetric material hardening in compression and stiffening in tension (<i>Weak-C</i>).	77

Nomenclature

Acronyms

BESO Bi-directional Evolutionary Structural Optimisation

CG Conjugate Gradient

DD Direct Derivative

DEM Discrete Element Method

DETO Discrete Element Topology Optimisation

FD Finite Difference

FEM Finite Element Method

FRP Fibre reinforced polymer

GA Genetic Algorithm

LAMMPS Large-scale Atomic/Molecular Massively Parallel Simulator

OC Optimality Criteria

PFM Phase Field Method

SD Steepest Descent

SIMP Solid Isotropic Material with Penalization method

TO Topology Optimisation

Symbols

χ chi vector of a design representing the distribution of χ across the design space

Nomenclature

χ_e	chi design variable in a specific element between [0;1] representing the existence or non-existence of that element
$\Delta\alpha$	angular displacement - rad
σ_{dev}	Von Mises or deviatoric stress - N/m^2
σ_{hyd}	Hydrostatic stress - N/m^2
E	Poisson's ratio
E	Youngs modulus - N/m^2
k_{ij}	Stiffness of an inter-particle connection - N/m
$nelx$	number of elements in the x direction in a DETO simulation
$nely$	number of elements in the y direction in a DETO simulation
p	penalisation factor used in SIMP optimisation
r_{ij}	inter-particle distance - mm
r_{min}	filter radius applied in a DETO simulation
U	Strain energy in a system of bonded particles - Nm
U^*	Complementary energy in a system of bonded particles - Nm

Chapter 1. Introduction

Optimisation is one of the primary and oldest concerns of Engineering. Across all disciplines the function of the engineer has historically been to design better, more efficient and less costly systems. But in recent years renewed interest in the topic has been generated by societal drives to reduce waste and improve the material and energy efficiency of our engineered systems. In Structural Mechanics, this could be a simple truss or beam system optimized for example to maximise stiffness to a given load. In fact the idea of optimising the use of structural material originates with work of Michell (1904), who developed a criteria for optimal truss structures where all members contribute fully to the compliance of the design an example of one such structure is shown in Fig. 1.1a). Computational techniques developed in the 20th century have led to new formulations of the continuum structural optimisation problem, one of these being topology optimisation (TO). The innovative approach of TO introduced in Bendsøe and Kikuchi (1988) is to discretise the design domain of a structural problem into individual units of material each associated with a design variable specifying the existence or non-existence of that element. In this way the optimisation becomes a problem of material allocation across a discretised domain. This means the positioning and connectivity of members is unconstrained and in fact emerges as a result of the optimisation procedure as is shown in the example in Fig. 1.1b).

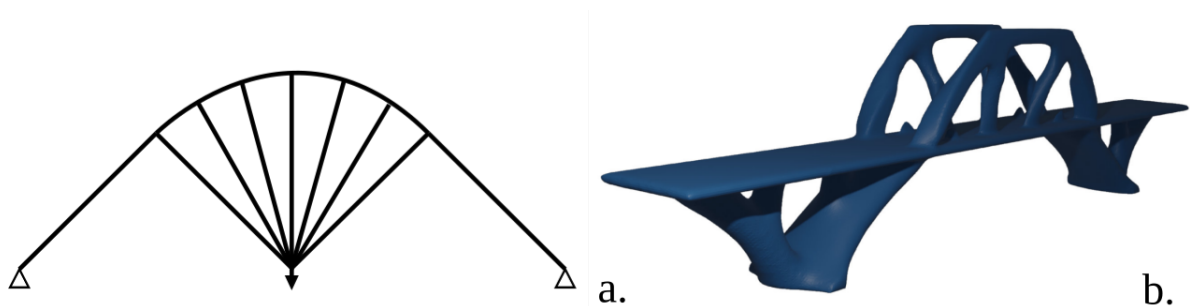


Figure 1.1 a) One of the earliest historical approaches to analytically defined optimal truss design in Michell (1904) b) An example of a modern three dimensional optimal design taking advantage of the principles of TO and the numerical technique of the FEM (Gupta et al., 2020)

TO has by now become a mature method, adopted widely in industry where it is valuable in the design of more sustainable structures, and target performance may include minimising material waste or maximising structural strength. The availability of TO is also being expanded by the fast-paced development of additive manufacturing, which is streamlining the production of complex structures that were too difficult and expensive to fabricate in the past (Plocher and Panesar, 2019). In academia too interest has grown and expansions to the method have been proposed for applications including fiber reinforced materials, compliant mechanisms, and multi-physics convection problems (Sigmund and Maute, 2013). A diverse range of objectives have been explored in the literature targeting properties such as structural vibration or robustness towards uncertainties, linked for example to malicious attacks or climate change. These diverse methods have all relied on the finite element method (FEM) for deriving the mechanical response of the systems in question and have therefore been slow to incorporate discontinuous behaviours and discrete systems, due to the limitations of FEM in describing material separation and the inability to model rigid body motion.

In this thesis the fundamental principles of continuum based TO optimisation are mapped onto a framework of bonded and discontinuous systems interacting via particle to particle interaction potential using the Discrete Element Method (DEM). Aimed to facilitate the use of TO for a range of systems and processes that were previously of limits such as granular materials, powder-based 3D printing, or structural collapse for which the FEM is not well suited.

Aim: To development a method of Topology Optimisation for systems of discrete interacting particles modeled using the discrete element method to optimise performance of granular and discontinuous systems.

Objectives

1. Map the existing principles of continuum TO onto a framework of discrete particles and interaction potentials.
2. Validate the proposed method against important results from the literature.
3. Expand the methodology into a full three dimensional procedure, interfacing with existing DEM software and develop an efficient scalable optimisation procedure.

-
4. Extend the capabilities of the method to include arbitrary complex and history dependant potentials as well as complex objective functions.
 5. Design and implementing a fracture energy based objective function to generate damage resistance oriented topologies.

This Thesis comprises of six chapters. Chapter 1 (this chapter) provides briefly an overview of role of topology optimisation and the popular contemporary approaches. Then presents the need and relevance of the present work and the aims and objectives of the thesis.

Chapter 2 provides the basic formulation of a penalisation-based TO method for continuum structures that will form the basis for the methodological developments detailed later in the thesis. An overview of the state of the art in TO with a particular focus on the inclusion of non-linearities, discontinuity, failure and robustness is presented. The DEM is described in detail highlighting the important methodological elements as applied to the content of this thesis.

In Chapter 3, the fundamental changes to the TO method are described to facilitate the formulation of a proof of concept discrete element topology optimisation (DETO) method applicable to 2D continuum structures discretised as systems of bonded particles, a simple software implementation of this method is put forward. Extensions of the formulation are then described to extend the method to 3D systems incorporating granular and discontinuous behaviours and to handle complex optimisation scenarios. These extensions are implemented in a more advanced software implementation taking advantage of a state-of-the-art particle dynamics simulation package.

In Chapter 4, results are presented to validate the method against well known examples for continuum structures from the literature. The effects of various properties of the implementation are studied and the inclusion of geometric and material non-linearity is presented.

In Chapter 5 examples of optimisation on 3D systems are presented as well as systems incorporating fracture, discontinuity and granular material behaviour. Before methods of defining more complex objectives and optimisation procedures are presented. Finally these techniques are showcased in a fracture resistance optimisation of a beam impact scenario and a proposal for an application to the design of a dynamic granular system.

Chapter 6 discusses the important outcomes of the study undertaken highlighting the achievement of the research aim and objectives before providing some critical examination of the

method. Areas for ongoing and future study are highlighted and some consideration of potential applications of the method is presented.

It is important to note that Chapters 3 Section 3.1 and 4 have been published in peer reviewed journals. The methodology in Chapter 3.1 The results regarding linear elastic systems and geometric non-linearity in Chapter 4 were published in *Meccanica* (O'Shaughnessy et al., 2021) were the manuscript was drafted collaboratively with Dr Enrico Masoero. and the results were produced and visualised by the author. The results regarding material non-linearity in Chapter 4 were published in (Masoero et al., 2021) where the simulations and visualisations created collaboratively by the author and Dr Enrico Masoero.

Chapter 2. Literature Review

This chapter describes the development of TO, focusing particularly on the existing methodology of Solid Isotropic Material with Penalization (SIMP) based TO which is the basis for the methods in this thesis. The SIMP method for linear elastic structures is first explained in detail, before an overview of the important recent developments and applications of the method is given. In these applications the Finite Element Method (FEM) is typically used to quantify the performance of the system and resolve information (e.g. displacement and stress fields) used in the optimisation. This approach has proven effective for linear elastic continuum structures. The application of FEM-based TO becomes more complex and limited when dealing with nonlinear problems; to appreciate this, existing works incorporating geometric and material non-linearity are briefly reviewed. However, when moving beyond non-linearity into failure and collapse, a key limitation of FEM-based TO emerges from its inability to incorporate discontinuous behaviours. This thesis will argue that a Discrete Element Method (DEM) based TO approach can overcome some of these limitations with discontinuous processes. The chapter continues with a brief description of the fundamentals of the DEM an important aspect emerging from the review is that combining DEM with SIMP TO will require a wise choice of optimisation algorithm. The last section in this chapter therefore will introduce some popular gradient and non-gradient based optimisation algorithms, which will be later employed.

2.1. Topology optimisation

Topology Optimisation (TO) refers to a family of computational methods to find structural solutions that optimise a set of target performance indicators under a set of constraints (Bendsoe and Sigmund, 2013; Hassani and Hinton, 2012). The basic concept is to split the space available for design down into small structural units. The task of the optimisation is then to allocate the best distribution of material to these regions to define an optimal structure. Common examples

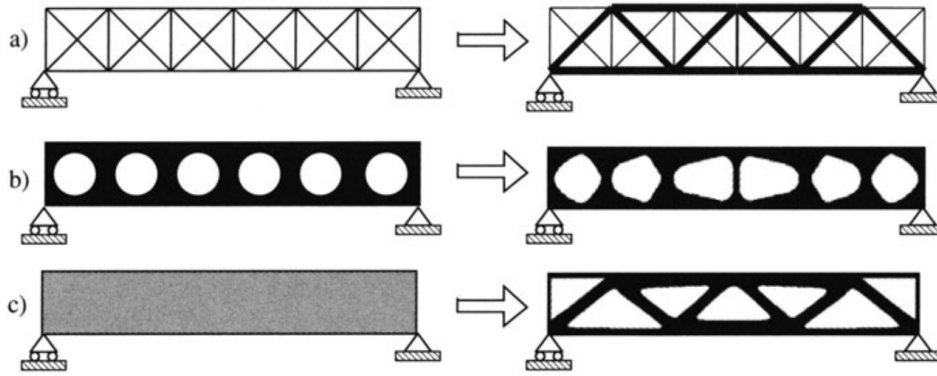


Figure 2.1 Example of three categories of structural optimisation from (Sigmund, 2001) a) Sizing of predefined truss elements b) Shape optimisation of cellular beam c) Topology optimisation of structural layout.

are to minimise compliance to certain loads using a prescribed amount of material, or to minimise mass while obtaining target compliance.

The TO procedure provides an optimal layout for the design including the size, shape, and topology of the system. Methods of size and shape optimisation are shown in Fig. 2.1 a) and b). These examples are limited to optimising parameters associated with predefined layouts, namely in a sizing problem as shown in a), the design variables could represent the thickness or member area of truss beams in a fixed domain, whereas in the shape problem as shown in b), the design variables represent a set of domain boundaries to be optimally arranged.

In contrast TO methods as shown in c) are capable of generating structural layouts with no starting geometry by optimising both member size, shape, and connectivity simultaneously via material reallocation. Since material and void can theoretically be allocated at any point in the design domain this allows for arbitrary connectivity in the design. This extra freedom results in designs that better follow the internal stress distribution of the structure as well as the unique appearance and aesthetic qualities of TO structures. As the topology of any structure is ultimately defined by the position of voids a more apt definition of TO then could be a technique for finding the optimum shape, size and location of “openings” in a design domain.

2.1.1. *Historical development*

The concept of optimising the layout of a structure has been of scientific interest since the seminal work of Michell (1904), who developed analytic examples for a class of uniform member thickness optimal trusses that bare his name. Michell’s criteria for optimality states

that every element in a loaded structure must bare equal strain. He noticed that the members in optimum layouts must follow lines of principal strain and that tension and compression members cross each other orthogonally as shown in Fig. 1.1 a).

In example of Michell structures, nodes could be located anywhere in the design domain; when constraining the possible location of nodes to a finite set of points and allowing for variations in member thickness, this leads instead to the so called Ground Structures Method (GSM) (Dorn et al., 1964; Prager and Rozvany, 1977). The GSM is a method of selecting truss members and sizes from the so called ground structure or set of all possible members and received considerable research interest throughout the 1970s. Recent developments to the GSM involving solution methods based on linear programming have advanced the capabilities of the method to allow it to tackle problems with hundreds of thousands of potential members (Gilbert and Tyas, 2003). As a result, the method has been taken up by some researchers dealing with relatively large and sparse structural problems including large span bridge design (Zegard et al., 2020).

The first computational formulations of TO were based on homogenisation methods (Allaire and Kohn, 1993; Bendsøe and Kikuchi, 1988) that feature a material allocation procedure based on an interpolation of element density by adding small holes. In the 1990s so called Evolutionary Structural Optimisation (ESO) methods (Xie and Steven, 1993) with the later extension to Bi-directional or (BESO) methods (Querín et al., 1998) were developed. These methods are based on a process of sequential element rejection or admission, the idea being to evaluate the Von-Mises stress or other criterion in each element before deciding based on an adapting criterion, whether to include elements in the next iteration of the design. Most of the important discussion including several critiques of these methods are best summarized by Rozvany (2009). Concurrently the development of the Solid Isotropic Material with Penalisation (SIMP) method largely supplanted homogenisation methods of topology optimisation via material distribution as it proved more efficient, especially for complex TO problems requiring discretisation and numerical solution (Rozvany and Zhou, 1991).

2.1.2. *Solid Isotropic Material with Penalisation method*

Here SIMP topology optimisation is first introduced in its simplest formulation, for stiffness maximisation of continuous elastic structures. Several important extensions for the method will be described in later sections. In SIMP-based TO, the objective (e.g. minimum compliance to certain loads) is a function of one design variable only: a density field assigning a value between 0 and 1 to each point in the design domain. In the objective function, the value of the design variable is raised to a power p , which is eventually exploited in the optimisation algorithm to penalise intermediate values and push the optimal solution towards a 0-1, void-solid only configuration.

SIMP-based TO is a well established technique that nowadays encompasses sophisticated algorithms and applications, and can include multiple constraints, and objectives (Lógó and Ismail, 2020; Sigmund and Maute, 2013). This section will first present the basic FEM-SIMP procedure for minimum compliance and show typical results from a simple but influential 2D example by Sigmund (2001). Thanks to its simplicity, this example became the entry point into TO for many researchers. This will provide the necessary background for the developments that will be proposed later in this thesis.

A numerical TO problem starts by defining the boundary conditions, external loads, supports, and spatial domain within which to define the geometric detail of the structure. When the FEM is used the domain is discretized into individual elements and, in the SIMP method, each element is associated with one scalar value of the design variable χ_e continuously distributed between $[0; 1]$ ¹. The design variable represents the density of the material at that point, between void ($\chi_e = 0$) and fully solid ($\chi_e = 1$). A uniform density field is usually chosen as a starting point and an objective function $c(\boldsymbol{\chi})$ is defined, which specifies the performance indicator to be optimised. $\boldsymbol{\chi}$ is the vector collecting the χ_e of all the individual elements. Optimisation is performed via repeated evaluation of the objective function (typically using the FEM) and the application of an update scheme designed to push the design towards optimality.

A typical optimisation problem is that of compliance minimisation, e.g stiffness maximisation, which can be achieved by minimising the complementary energy of the system under imposed

¹It is conventional in the literature to describe the design variable as an element density symbolised with the letter ρ . In this thesis χ , has been used because in the later derivation of TO using discrete element methods, this variable will no longer be directly connected to a density value.

loads or maximising it under imposed displacements (Bendsoe and Sigmund, 2013). For a linear elastic material under small displacements, complementary energy and strain energy coincide, so a problem of stiffness maximisation under imposed loads can also be written as:

$$\min_{\boldsymbol{\chi}} : c(\boldsymbol{\chi}) = \sum_{e=1}^N \mathbf{u}_e^T \mathbf{k}_e \mathbf{u}_e \quad (2.1)$$

$$\text{subject to : } \frac{V(\boldsymbol{\chi})}{V_0} = f \quad (2.2)$$

$$: 0 \leq \chi_{min} \leq \chi_e \leq 1 \quad (2.3)$$

In Eq. 2.1 the objective function is twice the total, linear-elastic strain energy of the system. $\mathbf{u}_e$ is the vector of nodal displacements at equilibrium under a set of imposed external loads. Eqs. 2.2 and 2.3 are constraints that the optimal solution must satisfy. The first constraint fixes the target volume of solid, as $V(\boldsymbol{\chi}) = \sum_e \chi_e$; V_0 is V when the whole domain is solid with $\boldsymbol{\chi} = 1$ everywhere; $f \in (0, 1)$ is a constant. The second constraint sets the bounds for χ_e between a minimum value χ_{min} and the fully solid $\chi_e = 1$. In principle one could use $\chi_{min} = 0$, a small but nonzero χ_{min} , e.g. 10^{-3} , is needed because the element stiffness $\mathbf{k}_e$ will be related to χ_e in such a way that $\chi_e = 0$ would entail $\mathbf{k}_e = 0$, making the structural stiffness matrix singular and thus the FEM problem unsolvable.

In Eq. 2.1, $\mathbf{k}_e$ is the element stiffness matrix. The distinguishing feature of SIMP is that $\mathbf{k}_e$ depends on χ_e as a power law:

$$\mathbf{k}_e = \chi_e^p \mathbf{k}_0 \quad (2.4)$$

$\mathbf{k}_0$ is a constant, base stiffness matrix. This penalisation scheme plays a central role in the solution of the optimisation problem. The dependence of $\mathbf{k}_e$ on χ_e causes the aforementioned dependence of $\mathbf{u}_e$ on χ_e under imposed loads. At a generic step in the optimisation process, the structure features a certain $\boldsymbol{\chi}$ vector and therefore each element has a corresponding $\mathbf{k}_e$; a Finite Element analysis provides the $\mathbf{u}_e$ corresponding to the imposed external loads for the current distribution of $\mathbf{k}_e$, and all this determines the current value of the objective function c .

The optimisation problem in Eqs. 2.1-2.3 can be solved using various methods, including the method of moving asymptotes (Svanberg, 1987) or Successive linear programming (Svanberg and Werme, 2006) The method described here is the Optimality Criteria method (Hassani and Hinton, 1999; Rozvany, 2012; Rozvany and Zhou, 1991; Zhou and Rozvany, 1992, 1993), which provides

the updating scheme. the expressions and algorithms to update χ at the generic optimisation step while respecting the imposed constraints (or tending towards a solution that respects them). Using the Optimality Criteria method (Rozvany, 1989) and imposing the constraints at each optimisation step, the following updating scheme is obtained:

$$\chi_e^{\text{new}} = \chi_e^{\text{old}} \cdot \left(-\frac{dc}{d\chi_e} \lambda \right)^\alpha \quad (2.5)$$

$\frac{dc}{d\chi_e}$ is the gradient of the cost function c with respect to χ , which is called *sensitivity*. $\alpha = \frac{1}{2}$ is a numerical damping coefficient to improve convergence. λ is a parameter that changes at every step of the optimisation and rescales the sensitivity so that χ^{new} respects the constraint on total volume in Eq. 2.2. Additional care must be taken to also guarantee that χ^{new} falls between $\chi_{\min}$ and 1, as per the constraint in Eq. 2.3. This can be achieved by capping the values of χ_e predicted by Eq. 2.5, but this would affect the first constraint. Therefore, an iterative algorithm is usually needed to find a value of λ that respects both the imposed constraints; for example, a bi-sectioning algorithm (Sigmund, 2001).

The analytical expression of the sensitivity can be obtained by combining the definition of c in Eq. 2.1 with the expression of the penalised $\mathbf{k}_e$ in Eq. 2.4 and applying the adjoint method (Bendsoe and Sigmund, 2013):

$$\frac{dc}{d\chi_e} = -p\chi_e^{p-1} \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e \quad (2.6)$$

An alternative numerical method of computing these sensitivities is to take advantage of the finite difference approach.

$$\frac{dc}{d\chi_e} = \frac{U^*(\chi_e + \varepsilon) - U^*(\chi_e)}{\varepsilon} \quad (2.7)$$

By computing the gradient over a sufficiently small perturbation ε the method acts as an approximation of the gradient of the function at χ . This becomes an exact partial derivative at the limit as ε approaches zero, therefore the perturbation must be a sufficiently small finite number to maintain a good approximation. The cost of this numerical method is a single complete solution of the cost function for each χ perturbation, and therefore each element, per step. For a large number of elements this method is orders of magnitude slower than an analytical approach,

but offers the benefit of being applicable to problems where there is no analytical derivative available.

Regardless of the method of computing the sensitivity, the combined effect of Eqs. 2.5 and 2.6 is to push material away from under-utilised areas of lower strain intensity (that contribute less to the overall stiffness of the structure) and move the design towards a solid-void only solution over successive iterations of the optimisation. This approach is only effective if $p > 1$; however, a value of $p \geq 3$ is usually preferred for reasons linked to the fabrication of actual structures using only void or fully solid parts (Bendsøe and Sigmund, 1999). Higher values of p enforce stricter solid-void only solutions and also improve the speed of convergence, but they reduce the ability to escape local minima of $c(\chi)$ and therefore increase the probability of finding sub-optimal solutions.

When constructing sensitivities for a design update it is common to use filtering (Sigmund and Petersson, 1998). The idea is, in order to avoid abrupt boundaries between solid and void in the material field the updating scheme in Eq. 2.5 does not use directly the sensitivity from Eqs. 2.6. Instead, it uses a new coarse-grained sensitivity $\widehat{\frac{\partial c}{\partial \chi}}$ that, for the generic element e , depends also on the sensitivities of neighboring elements:

$$\frac{\widehat{\partial c}}{\partial \chi_e} = \frac{\sum_{k=1}^{n_f} \frac{\partial c}{\partial \chi_k} W_k \chi_k}{\chi_e \sum_{k=1}^{n_f} W_k} \quad (2.8)$$

$W_k = \max\left(1 - \frac{r_k}{r_{min}}, 0\right)$ is a factor that linearly reduces the weight of a neighboring particle k with its distance from element e . The filtering length r_{min} is chosen to ensure a minimum member thickness in the final design. A feature of TO is that the optimal design can usually be split further and further into progressively smaller structural elements with small improvements in performance. However this complicates the manufacturability of the proposed design, and introduces a dependency of the output topology on the resolution of the FEM mesh (the size of the individual units for which χ is defined). In a very fine mesh, a much finer design may be constructed, whereas, a coarser mesh is limited to the width of a single finite element to allocate material. Filtering, e.g. as per Eq. 2.8 is a possible approach to ensure a minimum member

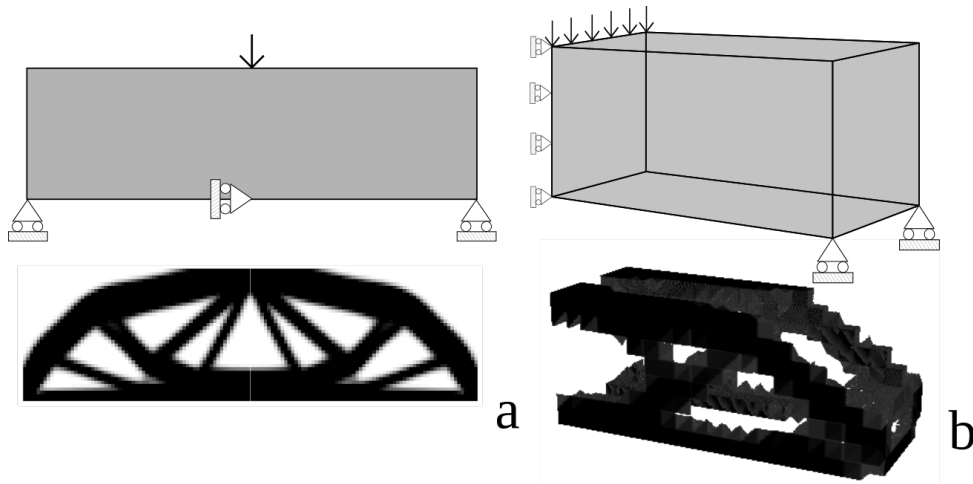


Figure 2.2 a) Example of a 2D simply supported beam optimisation using a 2D MATLAB script (Sigmund, 2001) with a size of 75x25 square elements b) Example of half a 3D simply supported beam optimisation with a periodic boundary condition in python and utilising the FEniCS C++ finite element library (Gupta et al., 2020) with a size of 20x8x8

thickness that is larger than one element, hence effectively removing mesh dependency from TO solutions.

Fig. 2.2 shows solutions of the stiffness optimisation problem for the simply supported beams outlined in this section, both in 2D and 3D. These examples clearly show how TO has indeed allocated material desirably into regions of high stress.

2.1.3. Development and Applications

In recent years, TO has seen an explosion of academic interest leading to considerable developments in many new directions, as well as novel problem applications in various areas, such as compliant mechanism design (Bruns and Tortorelli, 2001), natural convection problems (Alexandersen et al., 2014), fibre reinforced material design (Wu et al., 2017). Large scale applications of TO including civil engineering applications like the design of buildings and bridges have been tackled by some researchers (Kingman et al., 2015) who have utilised TO as a tool to help inspire the design of efficient solutions rather than to generate complete designs, due to the limitations of the length scales involved. In fact, Only recently has the required computational power become available to undertake SIMP based so called giga-voxel topology optimisation (Aage et al., 2017) on structures with length scales in the tens of meters. New approaches have been developed too, notably based on the Level Set method (Challis, 2010; Wang et al., 2003). in these approaches the design of the topology is shape derivatives in the form of so called level-set

functions are used to define the boundaries of the optimal topology rather than element based design variables used in other methods.

On the theoretical side, several influential works have drawn the comparison of TO structures to naturally occurring structures (Aage et al., 2017) such as insect wings, micro structures of wood, and porous bone material in the literature (Daynes and Feih, 2022; Wu et al., 2017). In fact the term computational morphogenesis has been popularized by some researchers (Ohmori, 2011) as a synonym for the broad family of TO methods. In Biology morphogenesis is the process by which organisms develop form in response to the mechanical stresses.

These developments are interesting and promising, however one area at the boundary between what FEM-based TO can achieve, is the incorporation of discontinuous behaviours and discrete systems. To describe the current state of the art, the following sections will return to the simple SIMP method described in Section 2.1.2 and will keep the focus on relatively simple structural geometries, but progressively moving away from linearity into non-linearity, failure, fracture, and collapse.

The vast majority of applications and methods discussed above and in the literature are reliant on the Finite element method for deriving the displacements and therefore solving the cost function in Eq. 2.1². FEM is extremely efficient for linear elastic analyses. Including geometric and mechanical non-linear behaviours is more demanding, but these can be considered too.

Geometric non-linearity

When structures are subjected to high loads and undergo large displacements, geometric non-linearity impacts the optimal topologies (Luo et al., 2015). The effects of geometric non-linearity on structural performance arise when large deflections change the direction of load, relative to the local coordinate system of a structure, such as in the case of the simply supported beam structure show in Fig. 2.3 where deformation of a beam under a central load stretches the beam longitudinally creating catenary action by introducing additional tensile axial forces.

Linear FEM is formulated under the assumption that deformations under any imposed loads will be much smaller than the dimensions of the body and therefore any second order effects caused by them can be ignored in the analysis. The stresses, and displacements are computed

²Two important exceptions are the work of (Gong et al., 2018; Zhang et al., 2020) who have applied an element-free Galerkin method based on moving particles to solve TO problems, and work on the Phase Field methods of TO undertaken in (Bourdin and Chambolle, 2003)

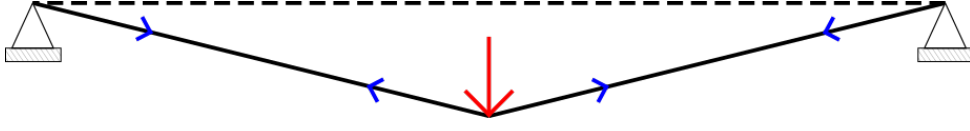


Figure 2.3 Example of the catenary forces that may arise due to deformation in a simply supported beam.

based on a static analysis of the loading conditions in the undeformed state. This type of analysis holds true for many common engineering applications. However, in the case of extreme loading conditions or flexible structures when large displacements occur, a linear analysis would accumulate a residual error R corresponding to the difference between the external loads and the internal forces, leading to an imbalance between internal and external work at equilibrium:

$$\mathbf{R} = \mathbf{P} - \int_v \mathbf{B}^T \mathbf{s} dV \quad (2.9)$$

P is the external force vector, s is the so called Piola-Kirchhoff stress vector (Bonet and Wood, 2008) dependent on the deformed shape of the geometry and the matrix B transforms a change in displacements into Green-Lagrange strain (a non-linear strain measure). To accommodate geometric non-linearity, this residual must be reduced to zero, a common approach is the total Lagrangian finite element method (Zienkiewicz et al., 2000). This method follows an iterative procedure to update the deformation of the structure over a suitable short step using the Newton-Raphson Method and correcting for the new state of the load application, after each incremental displacement, so as to reduce the residual.

In FEM-based TO, accounting for geometric non-linearity requires additional complexity in the formulation of the problem. Buhl et al. (2000) incorporated the above method of geometrically non-linear FEM into the standard TO formulation for stiffness maximisation in Eq. 2.1-2.3 by adding only one additional constraint to the problem formulation $\mathbf{R} = 0$ and solving the cost function iteratively at each step. For a non-linear analysis a residual R can be added to the conventional strain energy minimisation cost function.

$$c = \sum_{e=1}^N \mathbf{u}_e^T \mathbf{k}_e \mathbf{u}_e = \mathbf{P}^T \mathbf{U} + \lambda \mathbf{R} \quad (2.10)$$

U is the displacement vector for the system in its equilibrium position. This new formulation requires a modification of the sensitivity computation in Eq. 2.6 and the adoption of the adjoint method (Cao et al., 2003) which involves introducing an unknown vector of multipliers λ to one

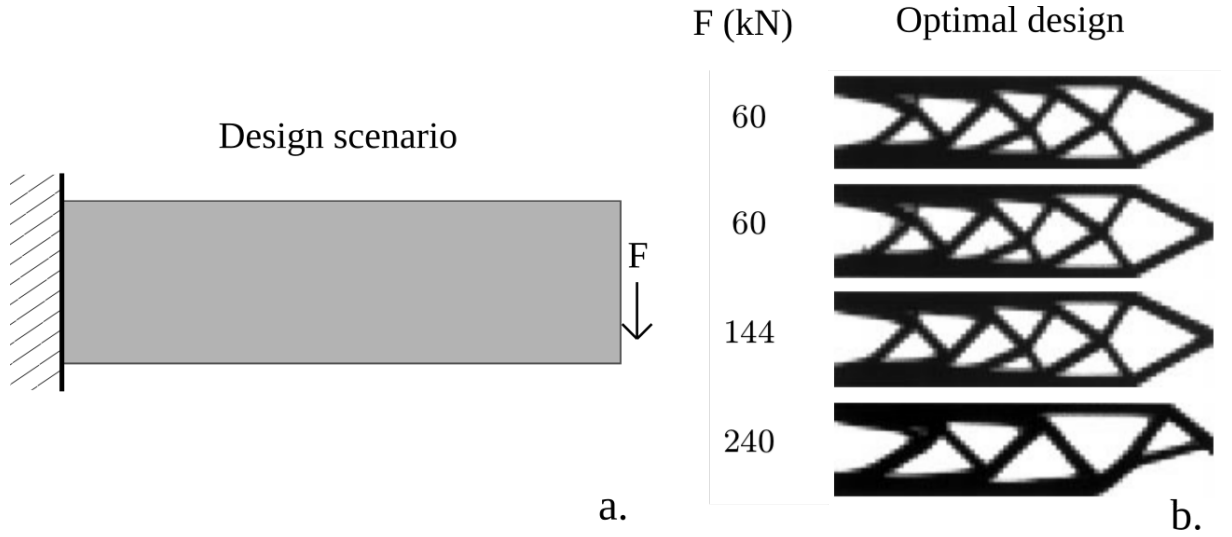


Figure 2.4 Cantilever beam example optimised using geometrically non-linear finite elements from (Buhl et al., 2000). a) shows the boundary conditions used, supports and location of applied force whilst b) shows the resulting optimal topologies from these boundary conditions with varying intensities of force $\mathbf{F}$ applied.

of the constraints, in this case $R = 0$ is incorporated into the objective function. The sensitivity of this derivation is therefore:

$$\frac{dc}{d\chi_e} = \mathbf{P}^T \frac{d\mathbf{U}}{d\chi_e} + \lambda^T \left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \frac{d\mathbf{U}}{d\chi_e} + \frac{\partial \mathbf{R}}{\partial \chi_e} \right) \quad (2.11)$$

where $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}$ is equal to minus the tangent stiffness matrix $\mathbf{K}_T$. λ is selected to eliminate the unknown $\frac{d\mathbf{U}}{d\chi_e}$ which corresponds to solving the system of linear equations:

$$\mathbf{K}_T \lambda = \mathbf{P} \quad (2.12)$$

In this way by inserting λ back into Eq. 2.11 the sensitivity is reduced to simply:

$$\frac{dU}{d\chi_e} = \lambda^T \frac{d\mathbf{R}}{d\chi_e} \quad (2.13)$$

The derivative of the residual here is found by differentiation of Eq. 2.9.

The application of the above method can produce structures such as the ones in Fig. 2.4 where the same cantilever beam example was subjected to progressively higher load intensities. A linear TO routine would generate identical geometries for each case however, since the deformed configuration of the structure is considered in these non-linear examples higher intensity loaded structures show strong signs of non-linearity

Material non-linearity

Incorporation of Material non-linearity into continuum-based TO has been undertaken too (Liu et al., 2018), an important finding has been that structures optimized assuming linear elasticity, may be significantly sub-optimal when the material behaves non-linearly (Maute et al., 1998b; Schwarz et al., 2001). In addition to this since TO structures conventionally exhibit maximum utilisation of available material, often featuring slender elements, material non-linearity is expected in high performance structures making material non-linearity of particular importance.

Existing methods have mostly focusing on elastoplastic material laws (Ryu et al., 1985; Tsay and Arora, 1989). As the material behaviour is no longer reversible in the plastic regime structural performance becomes history-dependent and can only be calculated by an incremental procedure. Consequently, the structural sensitivities are also history-dependent and have to be computed after each incremental step (Schwarz et al., 2001) with respect to all previous structural states. (Jung and Gea, 2004). These approaches add significant computational effort so a number of studies have targeted efficient and accurate computation of non-linear sensitivities in TO

Failure

FEM analyses becomes more problematic as a structure approaches failure, and indeed applications of topology optimisation to problems involving fracture are only very recent. A variety of approaches to incorporating aspects of yielding, crack initiation and failure optimisation have been explored in the literature.

Methods targeting strength in ductile materials are relatively common, and there is a considerable literature on the application of local yield constraints in TO (Duysinx and Bendsøe, 1998). One such popular constraint being the inclusion of a local Von Mises yield criterion such as in (Herfelt et al., 2018; Mirzendehtdel et al., 2018). In this way, a minimum strength requirement is applied to avoid stress concentrations leading to yielding. (Nakshatrala and Tortorelli, 2015) proposed a topology optimisation framework wherein the plastic material response, of materials subject to impact loading, was modeled with Von Mises plasticity. This allowed for the maximisation of energy dissipation through plastic work and was further expanded when (Li et al., 2017) introduced an elasto-plastic damage model and maximum damage constraint. This approach

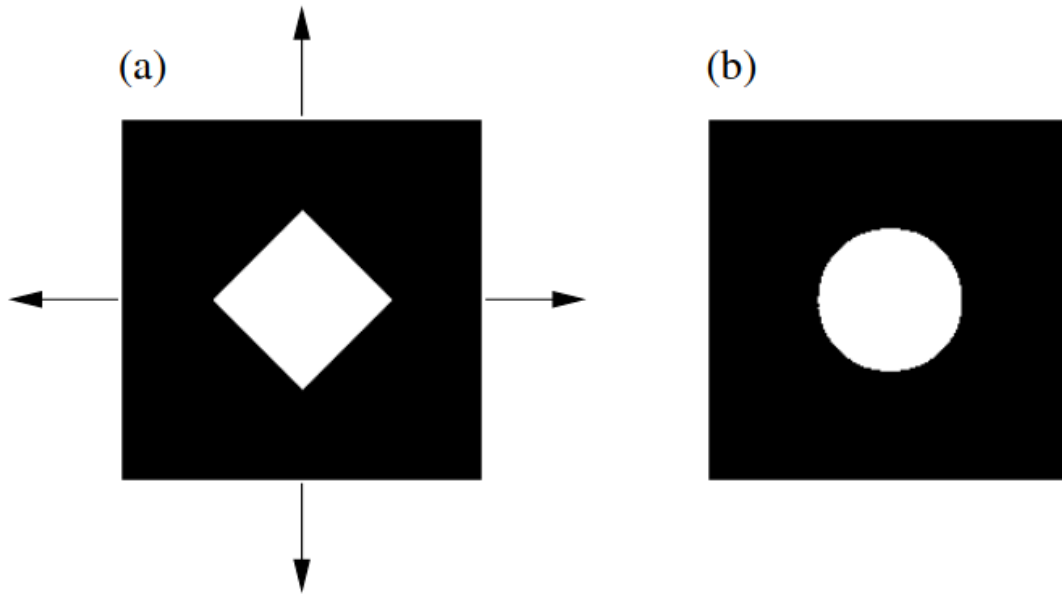


Figure 2.5 a) Initial configuration of plate with square crack from (Challis et al., 2007) b) Optimised result showing the expected result of circular crack leading to the lowest stress concentrations and therefore highest fracture resistance.

allowed the design of structures that maximised energy dissipation whilst avoiding excessive local material failure, that could be utilised to enhance the crashworthiness of structures for example.

One of the firsts approaches to directly apply a brittle fracture criterion in TO (Challis et al., 2007) related fracture resistance to the amount of elastic energy released by crack propagation. This approach utilised the level-set method and assumed cracks could only initiate from the boundaries of the structure. It then used a virtual crack extension technique, whereby small virtual perturbations to the finite element nodes making up the boundary are introduced to simulate crack initiation and a sensitivity is derived based on the virtual energy released. This sensitivity drives the update of the movable boundaries in the design, towards an optimum solution that minimises the energy released upon crack initiation. This method produced the expected result of rounding the edges of a square crack in the plate shown in Fig. 2.5 to a circular one. Whilst this approach is able to optimise against crack initiation it does not include the effects of crack propagation and fracture.

In recent years, the phase field method (PFM) for fracture has allowed for greater inclusion of fracture in TO (Da and Yvonnet, 2020; Xia et al., 2018). PFMs can get around some of the complications of FEM in handling material separation, allowing in fact for the dynamic initiation, extension and propagation of complex arbitrary cracks whilst maintaining a regular mesh. This

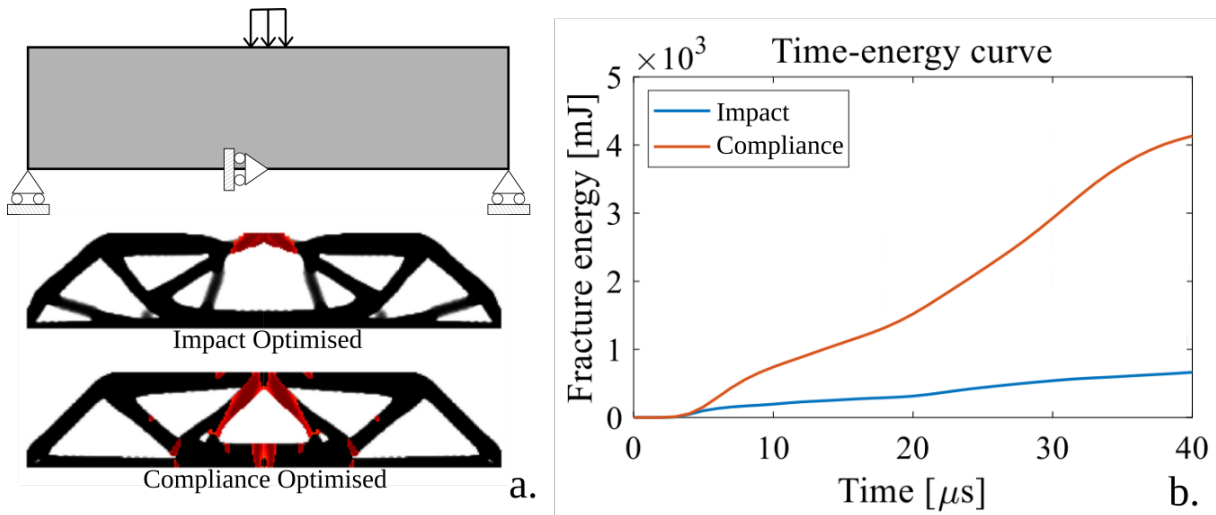


Figure 2.6 a) Optimised structures from (Wu et al., 2022) for dynamic impact and compliance showing final crack configuration. b) Fracture energy curves for both structures.

advancement allowed for (Wu et al., 2022) to incorporate the fracture pattern in beams subjected to dynamic impact loading into a SIMP optimisation aimed at minimising the fracture energy in the system. The results of this work are shown in Fig. 2.6. In particular Fig. 2.6a) shows the final configuration and fracture pattern of a system optimised for minimum fracture energy and another for minimum compliance; both are subjected to a dynamic impact. Fig. 2.6b) shows the fracture energy evolution over the same impact scheme. The formulation of the optimisation towards fracture energy also contains a necessary constraint on the minimum compliance of the structure. This avoids the global optimum associated with a disconnect between the load and the supports, that has been noted in other cases of dynamic TO (Silva et al., 2019). This compliance was computed with a separate static simulation under imposed load.

The result optimised towards impact in Fig. 2.6 shows a tendency for the optimisation to compartmentalise local damage by removing material from around the load creating a structure with two side substructures that will remain intact even after the central member is removed by the impact.

Robustness

Other methods have aimed to utilise only conventional linear elastic TO to emulate the effects of complex loading conditions and even damaged systems to optimise these for structural robustness. The idea being that if localised element failure occurs damage propagation is resisted. This has been done by adding systematic element removal to generate linear substructures representing

all possible individual damage scenarios for a specified size and shape of damage region (Jansen et al., 2014; Zhou and Fleury, 2016) the damage scenarios are usually simulated exhaustively meaning one simulation is run centering the damage on each element in the mesh. By creating a combined sensitivity from the energy minimisation of each of the damaged substructures a damage tolerant design is generated. However the number of possible damage configurations make this a costly method and impractical for larger meshed designs. Another approach has been to utilise a local volume constraints in (Wu et al., 2018) to force the optimisation to spread material more evenly across a domain. In addition to the commonly used global volume constraint it was possible to apply a constraint restricting the volume fraction in each localised region of the structure. The effect was to force the optimisation to build thinner and more numerous elements and, if tuned correctly, to generate cellular structures. Examples of both material removal methods a) and local volume constraints b) are shown in Fig. 2.7. Despite the large methodological difference, these results share distinct similarities of separated slender elements that providing resistance to damage and variable loading conditions, the idea being that if an area of the structure is damaged, the rest of the system can maintain integrity.

These problems are typically analysed with discrete simulations (Frenkel and Smit, 2001; O'Sullivan, 2011). Therefore, arguably, a TO scheme that uses for example the Discrete Element Method (DEM) (Pöschel and Schwager, 2005) is desirable.

2.2. The Discrete Element Method

The DEM is a numerical approach describing the mechanical behavior of assemblies of discrete, interacting particles. The methods earliest formulation was by Cundall and Strack (1979) to model granular media such as sands, soils, and powders (Nan et al., 2018). DEM can also be applied as an approximation of continua, in particular when describing processes that involve fracture (Magnier and Donzé, 1998; Wittel et al., 2003), fragmentation (Carmona et al., 2014; Kun and Herrmann, 1996), or structural collapse (Masoero et al., 2010, 2012; Ye and Xu, 2017). Rigid body motion, impacts, and geometric and mechanical non-linearity (Ghosh and Ananthasuresh, 2020) are naturally captured by individual per particle force and velocity evaluations.

In the DEM, collections of particles are modeled in 2D or 3D, forces are transmitted between particles via mechanical interactions described by predefined constitutive relationships. These

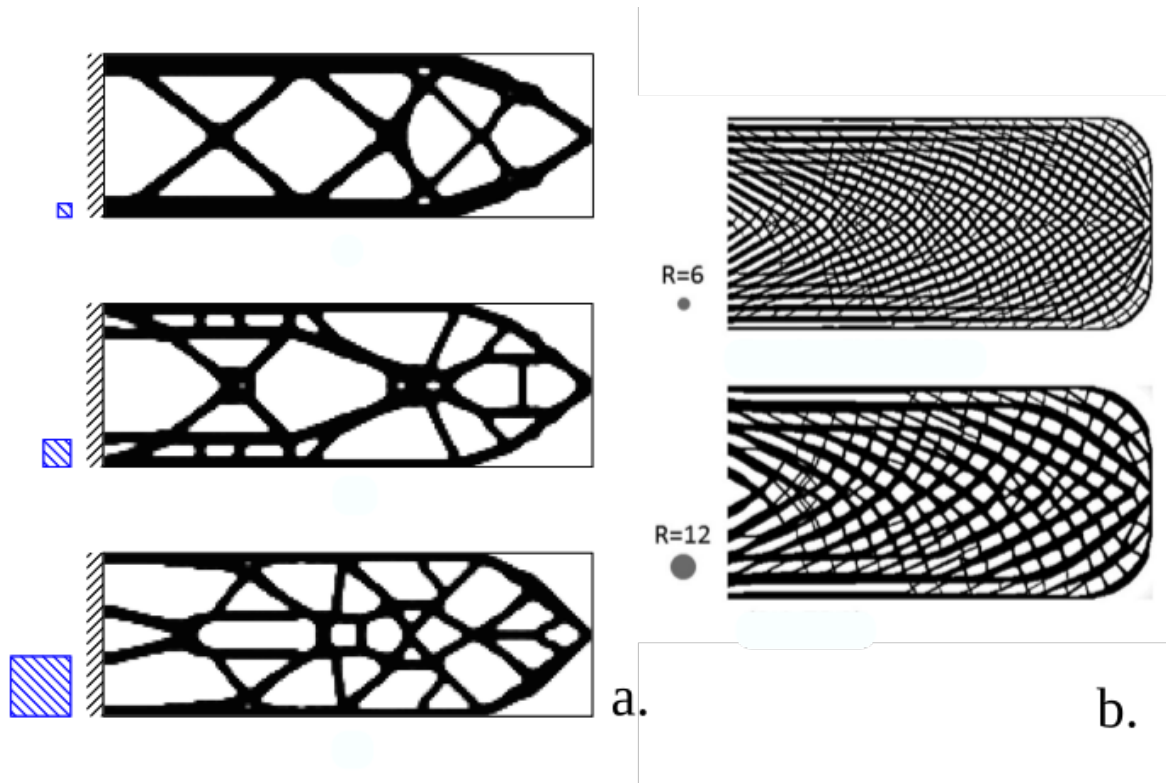


Figure 2.7 a) Structures from (Jansen et al., 2014) optimised utilising systematic element removal in each case the size of the damage scenarios is shown by the blue square, damage is simulated by positioning the blue square centered on an element in the mesh and removing all material from that region, this is repeated for each individual element to compute a combined sensitivity b) Structures from (Wu et al., 2018) optimised using a local volume constraint of 0.6 in each case the radius over which the constraint is applied is shown by the gray circle.

interactions are usually dependent on parameters such as stiffness or Young's modulus, which relates to the properties of the material the particles are made of, as well as the relative positions and orientation of the interacting particles.

Particle trajectories are typically derived by explicitly integrating Newton's second law of motion to determine velocities and therefore displacements over a target time period discretized into suitably short time steps. However DEM can also be used to find static equilibrium through energy minimisation; the former represents the dynamic motion of the system over a given time and can be used to capture history dependent behaviours such as fracture or granular flow. The latter displaces particles via iteratively minimising the total interaction energy of the system $U_{tot} = \sum_{i,j} U_{ij}$, to obtain a static equilibrium configuration similar to what one would obtain with a linear-elastic static FEM solution. This approach can be used to obtain the deformed configuration of a bonded particle system or the initial jammed conditions for a granular packing of elements (Krijgsman and Luding, 2016). Energy minimisation of this kind can be performed via various techniques. The simplest of these is the so called steepest descent (SD) method that involves computing the energy gradient of each element in the system at a given step and displacing them proportionally along the negative gradient, before repeating. The effect is to successively move particles into lower energy configurations until the gradient or change in energy reaches zero at static equilibrium. The SD method is known to be reliable but often less efficient than some more powerful methods such as the conjugate gradient (CG) method (Hestenes and Stiefel, 1952) which utilises the step history to accelerate the convergence. Specifically, at each iteration, the energy gradient is combined with the gradient from the previous iteration to compute a new search direction. This has the effect of reducing the overshooting effect of the SD method and leads to a faster convergence to an energy minimum, which can be seen in Fig. 2.8. Another common approach is to use damped dynamics methods such as the quickmin algorithm described in (Sheppard et al., 2008). In this approach particle velocities and displacements are computed based on interaction forces with strong velocity dependent damping. Individual particles overshooting an energy minimum is handled by freezing them, setting their velocity to zero.

In DEM continuum materials are usually described using persistent interactions between specific pairs of particles. These interactions remain active for the duration of the simulation

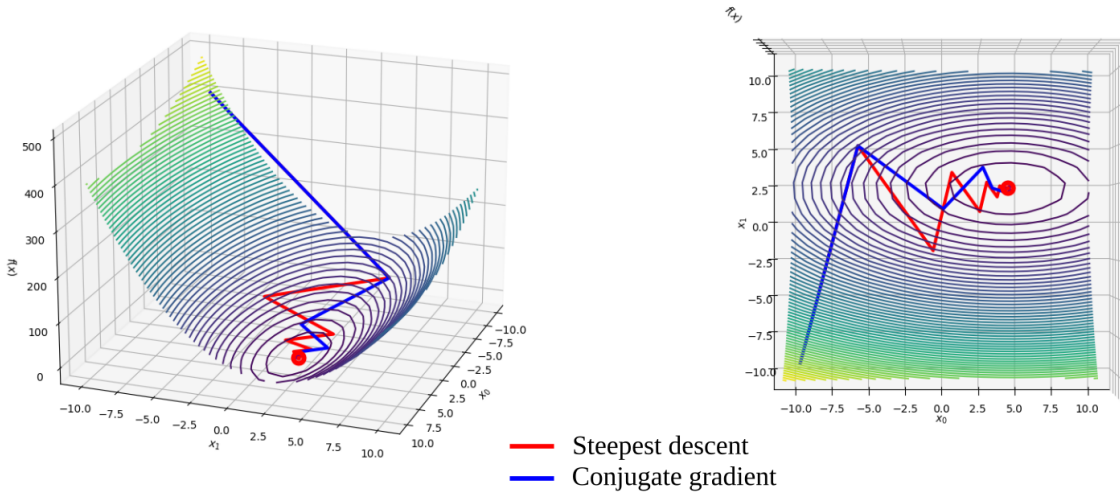


Figure 2.8 Simple illustrative example of gradient descent methods on a quadratic function using the steepest descent (red) and Conjugate gradient (blue) methods.

(unless broken following a breaking criteria). Networks of bonded particles can be built up to represent solids that have been shown to effectively model the structural response of continuous media (Tavarez and Plesha, 2007; Ye and Xu, 2017). The addition of bond breaking criteria have also been shown to accurately model fracture of solid materials (Metzger and Glasser, 2013; Patwa et al., 2016) including for example the crushing response of grains and assemblies of silica sand (Cheng et al., 2003; McDowell and Harireche, 2002). When a bond reaches a critical strain it is removed from the simulation producing fracture, like in the example of structural collapse of a building in Fig. 2.9a). When impact is considered in DEM it can take the form of hard impact scenarios where both bodies remain intact such as particles colliding in a simulation or soft impact scenarios where one body is destroyed in the collision in DEM this is commonly implemented by imparting a velocity in part of a structure, simulating a virtual soft impact for example to a beam or truss structure (Eibl, 1987).

Discontinuous media and granular systems such as Fig. 2.9b) are typically modeled with force field style potentials defined between all pairs of particles within a cutoff distance. The set of active interactions changes as particles move in and out of contact with each other and the relative motion of particles constantly creates and removes contacts (Silbert et al., 2001; Wang et al., 2015). These systems may require interactions that take into consideration rolling, sliding, twisting, and adhesion derived from fundamental contact mechanics of materials (Thornton,

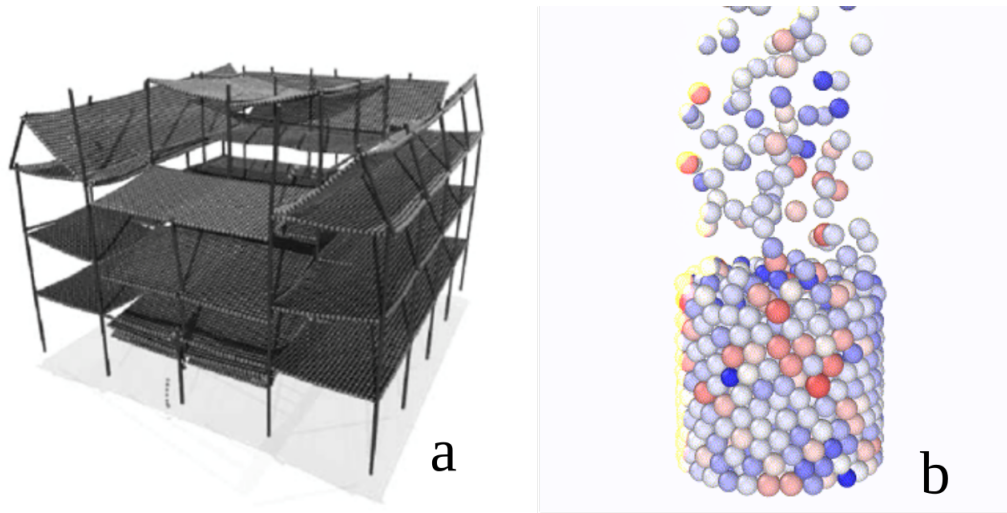


Figure 2.9 a) Structural failure of a building modeled with predefined breakable bonds b) Granular particle deposition using force field style interactions.

2015) applications include pharmaceuticals (Hare et al., 2011), 3D printing (Nan et al., 2018), mining and mineral processing, and geotechnics.

One example of a type of system that motivates the use of DEM for granular materials modeling are dynamic shear Rheometers such as the one shown in Fig. 2.10 which are devices for the characterisation of the fundamental bulk material behaviour of powders.

These devices feature a blade or impeller that is rotated and moved through a powder causing dynamic excitation of the powder bed, with particles interacting in shear as they flow relative to one another over and around the rotating blade (Hare et al., 2015). The resistance experienced by the blade represents the difficulty of this relative particle movement, and is related to the bulk flow properties of the powder via interpretive models such as the flow energy equation:

$$E_{flow} = \int_0^H \left(\frac{T}{R \tan \alpha} + F_{base} \right) dH \quad (2.14)$$

Where H is the blade penetration depth below the free surface of the powder, R is the radius of the impeller, α is the helix angle of the blade as it moves through the powder. This gives a method of quantifying the strain energy of the powder bed as it is deformed, and therefore characterising the flow behaviour of cohesive powders at high strain rates such as in hoppers and mixers (Tardos et al., 2003).

In in bulk particle assemblies flow behaviour is determined both by particle properties such as friction, and mechanical interlocking and by environmental conditions such as stress history, consolidation, and moisture level (Nan et al., 2017a,b). The accurate prediction of the bulk

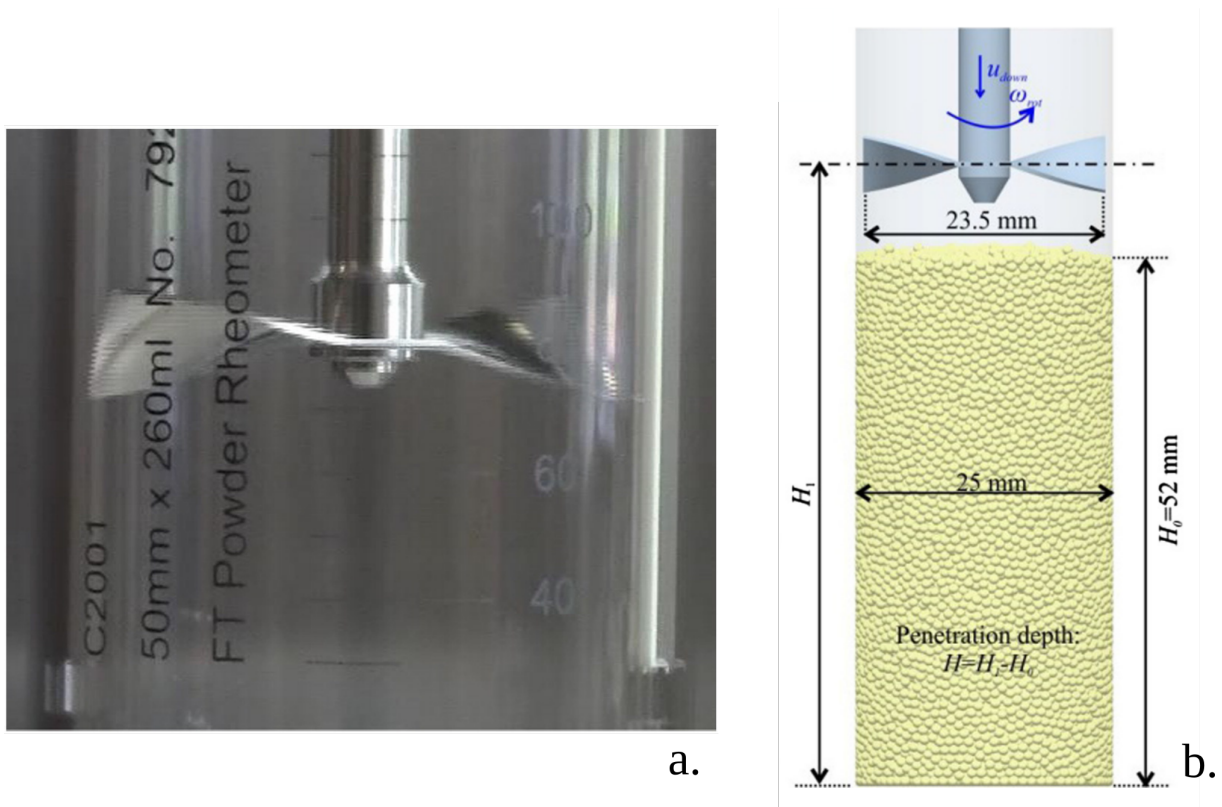


Figure 2.10 a) A conventional Rheometer blade featuring a helical structure b) A DEM model of a particle bed set up and Rheometer blade from (Nan et al., 2017b)

rheological response of these materials relies on the accurate measurement and characterisation of the behaviour of such assemblies in simulated process environments and under specific shear and flow regimes. Rheometers allow for the consistent and repeatable mechanical manipulation of powder materials under controlled environmental conditions to improve the prediction and performance of particle processing applications. Rheometers have applications in many areas of powder processing industries, including Pharmaceuticals, Fine Chemicals, Cosmetics, Metals, Ceramics, Plastics, Powder Coatings, Cements and Additive Manufacturing.

The strength of DEM is that with accurate selection of inter particle interactions the bulk behaviour of assemblies can be obtained from fundamental interaction models derived from the contact mechanics of discrete bodies. Particle methods are inherently meshless and naturally describe nonlinear effects. They can thus be used to effectively model solid and granular systems undergoing large deformations, including crack formation, fracture, and rigid body motion. This makes DEM an attractive method for communities of researchers studying systems and processes characterised by energetic, high strain rate systems such as the ones described above as well as systems where individual granular to granular contact is of specific importance.

2.3. Optimisation algorithms

Selecting an appropriate optimisation algorithm for a given problem can be complex and should usually be done on a case by case basis from an informed understanding of the problem. In general appropriate algorithm selection depends on access to gradient information for the objective function, and knowledge of the design domain. One way to divide potential optimisation algorithms is between gradient descent and non-gradient based approaches.

Gradient descent algorithms uses a deterministic approach to ensure that the process is always moving towards a local optima. The idea is to take incremental steps proportional to the negative gradient of the objective function. The process is the same whether the gradient is computed analytically or numerically by the methods described already in section 2.1.2. In the TO literature these gradients are commonly referred to as sensitivities and this nomenclature is adopted in this thesis too. Non-gradient based algorithms do not compute or directly use any derivative information to find optimal solutions. These are a diverse family of methods that including Genetic algorithms (Mitchell, 1998), Simulated annealing (van Laarhoven and Aarts, 1987), and Particle Swarm optimisation (Kennedy and Eberhart, 1995). They are often inspired by physical or natural processes and usually utilise stochasticity to conduct an efficient explorations of the design space.

Consider the graph in Fig. 2.11 representing a simple fitness landscape with clearly identifiable local and global minima (in fact $\mathbf{x}$ here represents a multi-dimensional design variable vector of length n and the 2D curve here represents an $n + 1$ dimensional hyperplane) each point on this curve represents a unique solution to the optimisation problem, in the case of TO a specific allocation of material. Solving the optimisation is a matter of evaluating solutions from this curve until a satisfactorily optimal solution is found. Since gradient descent methods are able to follow the slope towards optima these methods are extremely efficient at arriving at minima without ensuring they are at a global minimum. A gradient descent approach starting with local optima between it and the overall global optimum (for example at the left hand side of Fig. 2.11) will always end at the local rather than the global optimum because it can never move uphill. Non-gradient based methods on the other hand can only evaluate the value of the cost function and, therefore, rely on sampling techniques to explore a landscape, often with the ability to explore multiple minima without getting stuck at local optima.

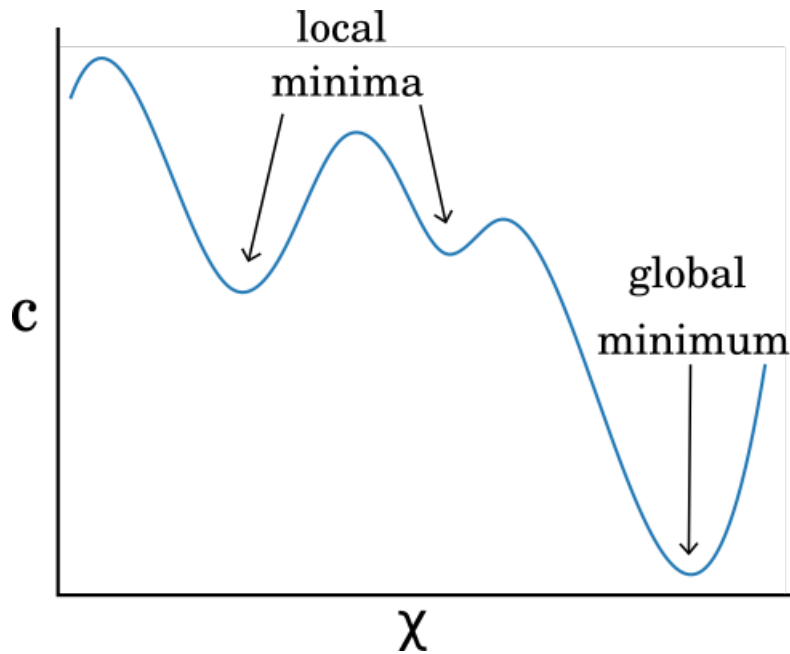


Figure 2.11 An idealised graph showing the evolution of an objective function across a one dimensional fitness landscape, featuring several prominent local minima.

Most influential and mainstream examples of topology optimisation have relied on gradient descent algorithms. In fact some researchers have cast doubt on the efficiency and validity of using non-gradient based methods in TO (Sigmund, 2011). This is because gradient based methods are orders of magnitude more efficient for problems featuring many design variables such as in TO. However some researchers have applied non-gradient methods with some success (Luh and Lin, 2009; Wang et al., 2015).

The efficiency and deterministic nature of gradient descent optimisation makes it an attractive choice. However if gradient information is not available such as when the objective function is defined based of a complicated simulation result, or if the design domain is discontinuous featuring prominent local minima, non-gradient based approaches can be beneficial.

Overall in the literature presented here shows that whilst TO as a technique is by now a sophisticated and diverse field, it has failed to adequately tackle discontinuous problems. A major drawback in the literature is the limitation of FEM-based TO when dealing with nonlinear problems, particularly in capturing discontinuous behaviors associated with failure and collapse. The incapability of FEM to incorporate these aspects undermines its effectiveness in optimizing structures under extreme conditions. In fact in the few examples shown here that have dealt with these systems the limitations of FEM based simulations have been too great to allow for strongly

convincing results. This leaves room for an application of TO within a DEM framework that does not suffer from these limitations and can represent these types of systems.

Chapter 3. A method of Discrete Element Topology Optimisation

This Chapter develops all the necessary methodology for a Discrete Element Topology Optimisation (DETO). In the first section 3.1 some fundamental changes to the SIMP-TO method are detailed to adapt it to systems of discrete interacting particles. Before a proof of concept method is presented that conceptually maps the principles of continuum finite element based TO onto a new framework of continuum systems approximated as unbreakable lattices of bonded particles. This Chapter sets up the framework for both applied force and applied displacement optimisation for the types of systems described above. This method is next incorporated into a simple proof of concept software implementation that is used in Chapter 4 to produce initial results as well as a detailed study of the effects of the input parameters and highlight phenomena like mesh dependency and checkerboard.

In the section 3.2 of this chapter, an extended DETO methodology is elaborated. The existing proof of concept is built upon to including three dimensional systems featuring non-linear and history dependant interactions aimed at the optimisation of dynamic systems involving fracture and discontinuity with extensions to handle multi-objective optimisations drawing information from multiple concurrent simulations. Finally the generalised method is implemented in a more advanced and computationally efficient software implementation taking advantage of a state of the art open source DEM software package and parallel processing. This extended method is later used in Chapter 5 to produce initial results validating the extended approach and showcasing the functionality to incorporate granular, and irreversible interaction potentials.

3.1. Simple DETO formulation to maximise stiffness

This section proposes changes to the SIMP-TO method, described in Section 2.1, to adapt it to systems described using the Discrete Element Method (DEM). The systems considered here all represent continuum structures as lattices of bonded particles, as shown in Fig. 3.1a.

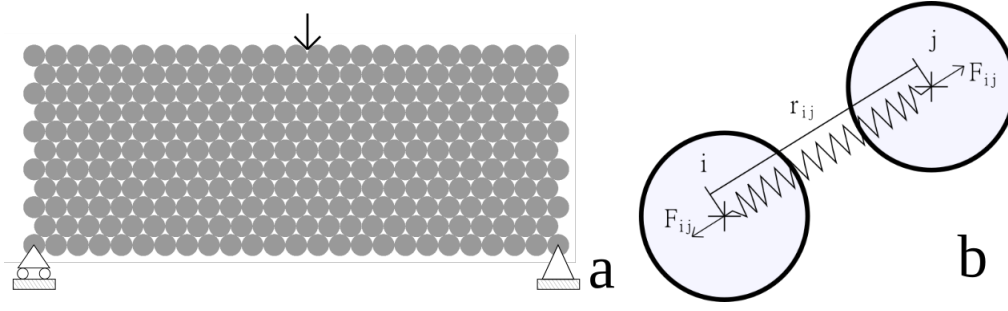


Figure 3.1 a) Continuum beam structure modeled as a hexagonal packed lattice of discrete elements b) Schematic of the interaction force between two particles emerging from a harmonic interaction potential.

These systems allow for simple boundary condition problems to be defined including applied forces, applied displacements and fixed supports. The stiffness of these system is associated with the interactions between particles. Consider a system of N interacting particles under a set of imposed external forces and constraints to motion; these latter may represent structural supports such as pins or rollers. The type of harmonic interaction potential in Fig. 3.1b) connect all nearest neighbour particles in the lattice. The strain energy and force of this potential corresponding to a linear elastic material are given by:

$$U_{ij} = \frac{1}{2}k_{ij}(r_{ij} - r_0)^2 \quad (3.1)$$

$$F_{ij} = -\frac{dU_{ij}}{dr_{ij}} = -k_{ij}(r_{ij} - r_0) \quad (3.2)$$

Where k_{ij} is the stiffness of the connection, r_{ij} is the inter-particle distance, and r_0 is the equilibrium distance. To directly reformulate the problem of strain energy minimisation given in Eq. 2.1 it is possible to take the strain energy described by Eq. 3.2 across the system.

The first change to the method for application to theses systems is straightforward: the design variable χ_e , which in FEM-based TO was specified for each finite element, here becomes a per-particle quantity χ_i whilst each particle i has an associated variable $\chi_i \in [0, 1]$ and all the per-particle χ_i are gathered into a vector $\boldsymbol{\chi}$. Particles with $\chi = 0$ interact with zero intensity with the others, effectively representing voids. Particles with $\chi_i = 1$ interact with full intensity, thus representing full solid. However, since geometric nonlinearities at least are always possible in DEM simulations¹, the problem of maximizing stiffness of the system should be kept as a

¹geometric nonlinearities are captured in DEM because interactions are always computed with reference to the system in its deformed configuration.

problem of minimization of complementary energy U^* , as one cannot simply equate U^* with the strain energy U of the system.

$$\begin{aligned} \min_{\boldsymbol{\chi}} : c(\boldsymbol{\chi}) = U^*(\boldsymbol{\chi}) &= \sum_{i=1}^N \mathbf{F}_i \mathbf{u}_i - U_{tot} = \\ &= \sum_{i=1}^N \mathbf{F}_i \mathbf{u}_i - \frac{1}{2} \sum_{i=1}^N \sum_{j>i}^N k_{ij} (r_{ij} - r_0)^2 \end{aligned} \quad (3.3)$$

$$\text{subject to : } \frac{V(\boldsymbol{\chi})}{V_0} = f \quad (3.4)$$

$$: 0 \leq \chi_{min} \leq \chi_i \leq 1 \quad (3.5)$$

The term $\sum_{i=1}^N \mathbf{F}_i \mathbf{u}_i$ is the external work, the product of the external forces on each particle times their corresponding displacement at equilibrium. If displacements are small, the sum equals $2U_{tot}$ and the FEM-based problem in Eq. 2.1 is recovered, except that the strain energy now features a sum over all *pairs* of particles, instead of over individual particles.

The inter-particle distance r_{ij} and the interaction stiffness k_{ij} are now scalar quantities pertaining to pairs of particles, whereas the FEM framework featured a vector of nodal displacements and a stiffness matrix pertaining to individual elements. The constraints in Eqs. 3.4 and 3.5 are the same as for the FEM-based problem. Unlike FEM solvers, DEM algorithms are not compromised if $\chi_i = 0$ causes some interactions to vanish (if some k_{ij} are zero). However, here $\chi_{min} > 0$ is still used because later a type of filtering will be employed which breaks down if a particle's χ equals zero.

As a consequence of the proposed per-particle definition of χ , a key change in DETO concerns the penalisation scheme. In the FEM-based approach, since each finite element contributes individually to c , the χ_e of each element penalises only the stiffness matrix of the element itself (Eq. 2.4). In the DEM context, however, since k_{ij} is associated with pairs of particles rather than individual ones the following penalisation scheme is proposed:

$$k_{ij} = \chi_i^p \chi_j^p k_0 \quad (3.6)$$

where k_0 is a constant base stiffness and χ_i and χ_j are the design variables of two interacting particles. The penalisation exponent p plays an analogous role as discussed in Section 2.1, i.e

pushing the solution towards a 0 – 1 design by penalising intermediary values of χ . If one takes $p = 1$ then k_{ij} scales as the harmonic average of χ_i and χ_j , which correctly ensures $k_{ij} = 0$ when either $\chi_i = 0$ or $\chi_j = 0$. However, preliminary tests with the new approach included in Chapter 4 have shown that $p = 2$ provides a good compromise between optimality of the solution (in the case of c reaching low values), solid-void only result, and convergence speed. Eq. 3.6 imposes that the interaction stiffness k_{ij} is a function of χ_i and χ_j . Therefore, under a given set of external forces, also the interparticle distance r_{ij} at equilibrium (or at a generic step during a simulated dynamic response) will depend on χ due to variable deformation in the structure. r_{ij} as a function of the interparticle force and stiffness is.

$$(r_{ij} - r_0) = \frac{F}{\chi_i^p \chi_j^p k_0} \quad (3.7)$$

The formulation of interaction penalisation in Eq. 3.6 is the crucial generalisable element of the proposed DETO framework and what drives the method towards optimum solutions. Based on this penalisation scheme, the cost function, constraints and boundary conditions of the problem can all be adapted to suit any potential DEM system.

Applying penalization to interactions shares some similarity to the ground structures method discussed in Chapter 2, where a set of nodes are defined and the design variables correspond to the cross sectional areas of bars connecting any pair of nodes. The main difference in Eq. 3.6 is that penalisation is applied through per-particle χ 's rather than directly to each interaction. This may better suit DEM simulations, where often the interactions between particles at or near contact are determined, in reality, by per-particle quantities such as chemical composition or physical and mechanical properties, e.g. the indentation moduli of contacting particles in Hertz potentials (Pöschel and Schwager, 2005) or the Young moduli of connected particles in cohesive nanoparticle models (Masoero et al., 2014).

The solution of the optimisation problem in Eqs. 3.3-3.5 can be obtained with the same updating scheme previously described for the FEM approach:

$$\chi_e^{\text{new}} = \chi_e^{\text{old}} \cdot \left(-\frac{dc}{d\chi_e} \lambda \right)^\alpha \quad (3.8)$$

However, computing the sensitivity $\frac{dc}{d\chi_e}$ now is more difficult than in Eq. 2.6, because there the expression of the sensitivity benefited from simplifications that arise when the adjoint method

is applied in the linear regime (Bendsoe and Sigmund, 2013). In the nonlinear regime, the adjoint method requires the tangent stiffness matrix of the system (Bendsoe and Sigmund, 2013); however, adopting stiffness matrices conflicts with the choice of adopting DEM simulations, in that a key strength of DEM simulations is precisely not to rely on stiffness matrices and thus avoid issues with them becoming singular, for example at mechanical failure. An alternative and general way to compute sensitivities is to use a finite difference approach detailed in Eq. 2.7 by computing directly the change in U^* due to a small but finite perturbation $\Delta_i \chi_i$. This change in energy entails two terms. The first term is the change in U when particles stay fixed at their equilibrium position $\mathbf{r}_{eq}$: this is due to the change of interaction stiffness, $\left. \frac{\partial U^*}{\partial k_{ij}} \frac{\partial k_{ij}}{\partial \chi_i} \right|_{\mathbf{r}=\mathbf{r}_{eq}}$. This is easy to compute, because it does not require any new equilibration of the system and can be obtained analytically from the expression of the strain energy U in Eq. 3.1. The second term is the change in external work and U due to the small change in particle positions, away from $\mathbf{r}_{eq}$ due to the difference in χ_i . Computing this term is what makes the finite difference method in computationally expensive, as one must find a new equilibrium configuration for each of χ_i . However there is no analytic equivalent for the derivative for external work term in Eq. 3.3 as it is the result of a simulation. In practice this method requires a dedicated simulation per χ_i value at each optimisation step effecting performance greatly and meaning that scaling the size of the simulation can rapidly become prohibitive. In Chapter 4 however, optimisation results obtained using the full finite difference approach in Eq. 2.7 are compared with results where the sensitivity is approximated analytically by its first term only:

$$\frac{dc}{d\chi_i} \approx \left. \frac{\partial U^*}{\partial k_{ij}} \frac{\partial k_{ij}}{\partial \chi_i} \right|_{\mathbf{r}=\mathbf{r}_{eq}} = -\frac{1}{2} \sum_{j \neq i} p \chi_i^{p-1} \chi_j^p k_0 (r_{ij} - r_0)^2 \quad (3.9)$$

It turns out that, for the case studies compared there, the approximation in Eq. 3.9 yields almost identical optimisation results as simulations using the full gradient. Also, the values of $\frac{dc}{d\chi_i}$ obtained with the two methods are not very different, meaning that, for these examples, Eq. 3.9 captures indeed the main part of the gradient of U^* . Based on this, the more efficient Eq. 3.9 will be used where ever an analytic gradient derivation is available throughout Chapter 4 unless explicitly mention otherwise. The applicability of the approximation in Eq. 3.9 to other systems should be checked on a case-by-case basis, as the approximation may in principle generate local minima, solutions that differ from those in the original problem. The generality of the

finite difference approach will instead be exploited in Chapter 5 where the complexity of the optimisation objectives make an analytic description of sensitivities impossible.

When the sensitivity is computed, χ can be updated. However, TO algorithms often add an intermediate step of filtering. DETO utilises the same filtering process as previously described to construct a coarse-grained sensitivity $\widehat{\frac{\partial c}{\partial \chi}}$ except now between neighbouring particles in the system

$$\widehat{\frac{\partial c}{\partial \chi_i}} = \frac{\sum_{k=1}^{n_f} \frac{\partial c}{\partial \chi_k} W_k \chi_k}{\chi_i \sum_{k=1}^{n_f} W_k} \quad (3.10)$$

where n_f is the number of particles within a distance r_{min} from the center of particle i , including particle i too. $W_k = r_{min} - r_{ik} \geq 0$ is a weight function ensuring that particles closer to i contribute most to its coarse-grained sensitivity $\widehat{\frac{dc}{d\chi_i}}$. The coarse-graining process in Eq. 3.10, known as *filtering*, is commonly used in Finite Element based TO to avoid the checkerboarding problem (Díaz and Sigmund, 1995; Sigmund and Petersson, 1998). DETO does not suffer from checkerboarding, but filtering can still be used to enforce a minimum member thickness on the results and improves the manufacturability of the optimum solutions.

Eqs. 3.3 to 3.10 complete the formulation of DEM-based TO for the specific case of stiffness maximization using an unbreakable, harmonic, pairwise interaction potential that will be useful for validating the method against key results from linear-elastic continuum finite element based examples.

There is nothing intrinsic to this method requiring that the interaction potential describe an elastic spring, and in fact more general interaction potentials can be employed to incorporate, for example, material non-linearity. The harmonic potentials considered so far have represented linear springs connecting the particles; this is analogous to a linear elastic constitutive law in the FEM. However, one can replace the potentials in Eq. 3.1 or Eq. 3.17 with a more general, non-linear form such as:

$$U_{ij...} = k(\chi_i, \chi_j, ...)g(\mathbf{r}_i, \mathbf{r}_j, ...) \quad (3.11)$$

where k is now a generic function of the design variables of the interacting particles, and g is a function of the position vectors $\mathbf{r}$ of the interacting particles. The ellipsis indicate that the interactions can involve more than pairs of particles, including three-body or four-body terms, as

well as rotational degrees of freedom. The complementary energy objective function then simply becomes:

$$c = U^* = \sum_{i=1}^N \mathbf{F}_i \mathbf{u}_i - \sum_{i,j} U_{ij} \quad (3.12)$$

When the approximation in Eq. 3.9 holds (i.e. the possibility to consider only one term in the derivative of c when computing sensitivities), the resulting sensitivity would then be:

$$\frac{dc}{d\chi_i} \approx \sum_{j, \dots \neq i} -\frac{\partial k}{\partial \chi_i} g(\mathbf{r}_i, \mathbf{r}_j, \dots) \quad (3.13)$$

Indeed, the derivatives apply only to the penalising function. Arbitrarily complex functions can be used for k and g in Eq. 3.11 without adding complexity to the optimisation while allowing for mechanical nonlinearities embedded into $g(\mathbf{r}_i, \mathbf{r}_j, \dots)$ to be seamlessly considered in the DETO framework. The nonlinearity induced in this way is still elastic, meaning that no energy is dissipated upon loading and unloading when using such potentials.

3.1.1. Software Implementation: DETO_2D

This section describes a simple numerical implementation of the proposed DETO method in C++ called DETO_2D. This implementation will serve as a proof of concept for the method, allowing for direct comparison with key results for continuum system produced by the FEM-based approach. The scope of this implementation is continuum systems approximated as unbreakable lattices of bonded particles, where structural responses are simulated via quasi-static analyses.

The initial configuration features approximately $nelx \times nely$ disks² arranged in a close-packed hexagonal lattice that fills a rectangular domain of size $(D \cdot nelx) \times \left(\frac{\sqrt{3}}{2} D \cdot nely\right)$, where D is the disk diameter: see Fig. 3.2. Lattices are simulated with a small finite uniform thickness t_z , into the page, where $t_z < D$ to avoid particle displacement in the constrained z axis. Initially all disks are assigned $\chi_i = f \in (0, 1)$ and a mass m that will be used to compute displacements due to applied forces in a damped dynamic DEM routine (using the quickmin algorithm) to simulate system behaviour. The parameters $nelx$, $nely$, D , and f , are chosen and provided by the

²Actually, the number of particles per row alternate between $nelx$ and $nelx - 1$ to respect horizontal symmetry, so the exact number of disks is $(nelx - \frac{1}{2}) nely$ when $nely$ is even, and $(nelx - \frac{1}{2}) \times (nely - 1) + nelx$ when $nely$ is odd.

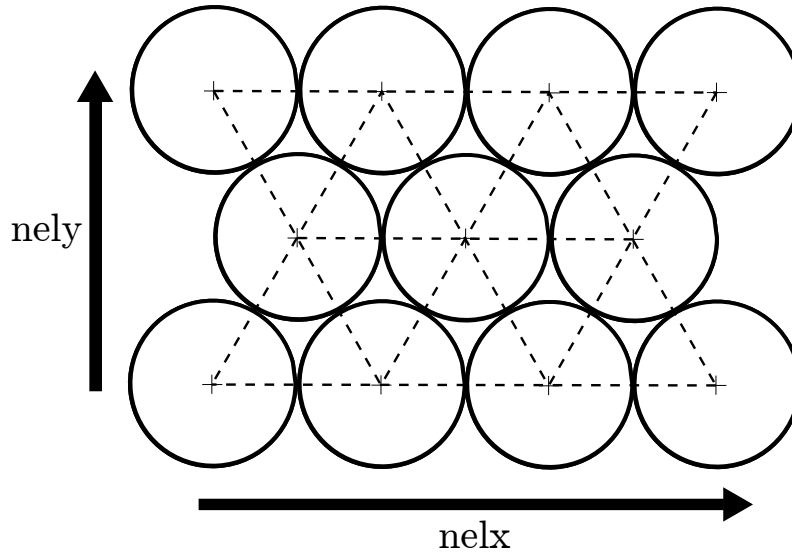


Figure 3.2 Hexagonal close packing of DE disks with linear elastic interaction potentials between immediate neighbors

user. Constraints, external forces and imposed displacements can all be applied to individual particles or groups of particles when setting the conditions for the optimisation.

Each particle interacts only with its immediately adjacent neighbors in the hexagonal lattice. The harmonic potential is the same as in Eq. 3.1, with equilibrium distance $r_0 = D$, stiffness k_{ij} penalised as per Eq. 3.6, and base constant stiffness k_0 chosen by the user. The harmonic bonds are modelled as unbreakable and the particles are not allowed to create new bonds with other particles that initially were not among their first neighbors. This restricts the scope of the DEM, which usually deals with particles that move widely across the system, creating new bonds or colliding with particles that initially might have been far away. In Chapter 4 however, this implementation of the DETO method is validated for the classical example of a simple beam under point load, for which only relatively small deformations are expected. In such applications, the particles will indeed interact only with their initial first neighbors.

Fig. 3.3 shows the flow chart for the program. First the system geometry is generated from the inputs as explained above, adding also the required external forces and constraints (e.g. pinned or roller supports). The initial neighbor list is recorded and stays the same during the whole simulation, for the reason discussed above. This is much faster than a general case in which the neighbor list must be updated dynamically during the DEM simulation.

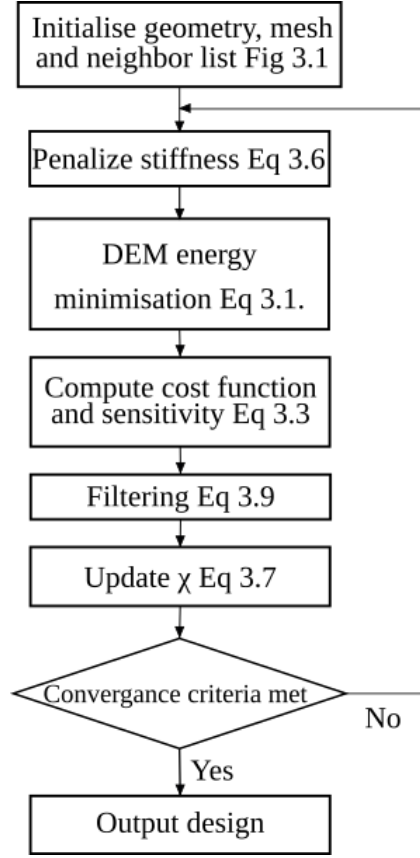


Figure 3.3 Flow chart of the DETO algorithm implementation.

The optimisation loop begins by computing the interaction stiffness k_{ij} for each pair of neighboring particles, following Eq. 3.6. The DEM minimization module computes the particle positions at static equilibrium using a damped dynamics algorithm by Sheppard et al. (Sheppard et al., 2008). The DEM solution is considered as converged when the change in total strain energy between two successive steps is sufficiently small: $\frac{(U_{tot}^{current} - U_{tot}^{previous})}{U_{tot}^{previous}} \leq etol$. The values of *etol* used in this manuscript will be in the $10^{-10} - 10^{-8}$ range. The energy minimization algorithm requires two parameters: a time step dt and a maximum particle displacement allowed at the generic step d_{max} . These should be fine tuned depending on the system that the user wants to analyse. The algorithm also uses the masses of the particles, here all set to the same value m .

After the DEM module converges, the DETO program computes the objective function c (i.e. the total interaction energy U_{tot}) and the sensitivity χ as per Eq. 3.9 or via finite difference method for the full sensitivity Eq. 2.7. The later is significantly more costly as each term of the sensitivity vector is computed by a full simulation. At the generic step of the optimisation process, the structure displays a certain vector of χ_i values, and a complementary work U_{eq}^* at equilibrium under the imposed external forces. The generic term $\frac{dc}{d\chi_i}$ of the sensitivity vector is

obtained by perturbing the value of χ_i by a small quantity $\Delta\chi$ and computing the new value of U^* resulting from a new energy minimization. This is repeated for each χ in the system.

Once the sensitivities are computed with either method filtering can be applied Eq. 3.10 the sums are over the n_f particles at a distance $r_{ik} < r_{min}$ from particle i , including particle i too³. The filtering length r_{min} is chosen and provided by the user. $W_k = \max\left(1 - \frac{r_{ik}}{r_{min}}, 0\right)$ is a factor that linearly reduces the weight of neighboring particle k with its distance from the centre of particle i : its value is 1 for particle i and becomes zero for particles with $r_{ik} > r_{min}$. The presence of χ_i at the denominator in Eq. 3.10 is the reason why one should enforce $\chi_{min} > 0$. In this Chapter a value of $\chi_{min} = 10^{-3}$, is used as is customary in the literature (Sigmund, 2001).

The reasons to include filtering are both practical and numerical. The practical reason is that r_{min} imposes a minimum size of solid and void regions in the final structure; this provides some control over the complexity of the optimal structure, which may help with fabricability. The numerical reason is that optimisation processes not including filtering often converge too rapidly to solid-void solutions getting effectively stuck into sub-optimum local minima. Some filtering (a small $r_{min} \approx D$) usually removes these local minima and leads to a better solution, although one must be careful as a larger r_{min} may also smoothen the global minimum and thus affect the optimality of the solution. When FEM analyses are used, another benefit of filtering is to remove the checkerboarding problem (Díaz and Sigmund, 1995; Sigmund and Petersson, 1998): when a fine FE mesh is used, individual neighboring elements in the optimal solution typically create an alternating pattern of void and solid. The problem arises from a locking effect in certain types of finite elements (Díaz and Sigmund, 1995; Sigmund and Petersson, 1998). DETO does not suffer from checkerboarding.

The last step in the optimisation loop is to update χ following Eq. 2.5, but using the filtered sensitivities instead of the original ones. The optimisation loop is repeated until $\chi_i^{new} - \chi_i^{old} \leq 4 \cdot 10^{-3}$ for every particle. In the final solution, some particles will feature $\chi_i = \chi_{min}$ and others, especially at solid-void interfaces, might be “gray”. feature a χ_i that is intermediate between 0 and 1. The MATLAB implementation of DETO_2D includes an optional post-processing module to reduce the solution to a solid-void only system, where all particles have either $\chi_i = 0$ or 1, while respecting the constraint on the total solid volume fraction f . In the simulations for

³Counter k is used instead of j for the neighboring particles to clarify that the neighbor list for filtering is not in general the same as for the interactions; for example, if $r_{min} > 2D$, also second nearest neighbors in the lattice will be included in the filtering even if they do not contribute to the interaction energy.

this chapter, however, no post-processing is performed to present unaltered optimisation results without further manipulations.

The stress tensor for each particle is also computed; this is not strictly needed for the optimisation process, but knowing the stress field inside the system will support the interpretation of the results. The per-particles stress tensor is based on the virial stress expression (Thompson et al., 2009):

$$\sigma_{ab,i} = \frac{1}{2V_i} \sum_{j \neq i} (r_{a,i} F_{ij,b,i} + r_{a,j} F_{ij,b,j}) \quad (3.14)$$

$\sigma_{ab,i}$ is the ab (xx, xy, or yy) stress component at particle i , $r_{a,i}$ and $r_{a,j}$ are the a -component (x or y) of the positions of particles i and j , $F_{ij,b,i}$ and $F_{ij,b,j}$ are the b -component of the force on particle i due to the interaction with particle j and vice versa. V_i is the averaging volume, here taken equal to the tributary volume of particle i . $\frac{V_{tot}}{N}$, where V_{tot} is the total volume of the rectangular domain (assuming a thickness of one in the third dimension) and N is the number of particles in the system. In particular, it can be beneficial to compute and plot the hydrostatic and Von Mises deviatoric components of the per-particle stress tensors:

$$\sigma_{hyd} = \frac{\sigma_{xx} + \sigma_{yy}}{3} \quad (3.15)$$

$$\sigma_{dev} = \sqrt{\sigma_{xx}^2 - \sigma_{xx}\sigma_{yy} + \sigma_{yy}^2 + 3\sigma_{xy}^2} \quad (3.16)$$

The optimisation problem in Eqs. 3.3-3.5 is quite generic but some notes on its scope and underlying assumptions are due. Discrete Element analyses typically include velocity-dependent dissipative terms (Pöschel and Schwager, 2005); here they aren't considered because the problem refers to static equilibrium conditions. Eq. 3.3 also assumes that U_{tot} is history-independent, with no irreversible processes such as bond breakages. The optimisations in Chapter 4 will target static equilibrium, without irreversibilities, and with expressions of U_{tot} that ensure the applicability and relevance of the problem in Eqs. 3.3-3.5.

Irreversible events can be included in DE analyses and they motivate in part the development of DETO. However, these processes are non-linear and dynamic in nature involving energy dissipation in the form of bond breakage that may invalidate the cost function in Eq. 3.3. In the next section Extensions to both dynamic problems and those considering irreversible processes

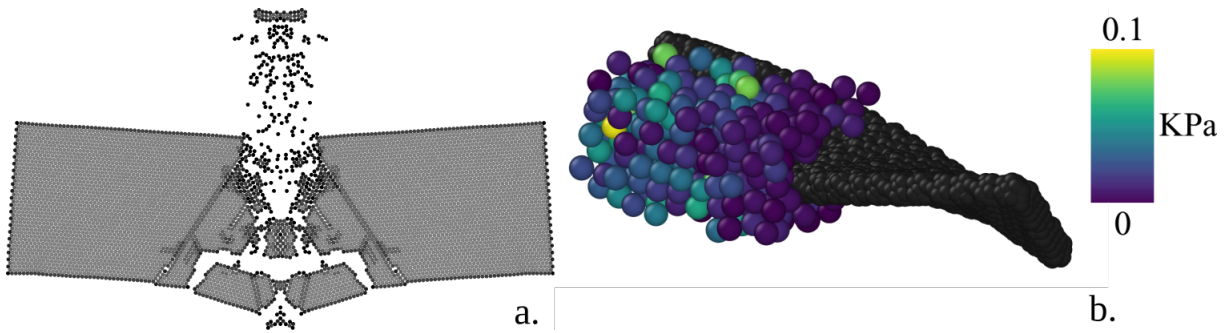


Figure 3.4 Example of a system that motivate the development of DETO_3D a simply supported beam undergoing fracture due to a soft impact scenario, where an optimal geometry may exhibit maximum resistance to damage.

are handled and implemented in a efficient generalisable software implementation that will be the basis for the optimisations described in Chapter 5.

3.2. Generalised Discrete Element Topology Optimisation

So far DETO_2D has dealt with continuum systems approximated as unbreakable lattices of bonded particles for the minimization of structural compliance. This limited scope was chosen to closely approximate typical FEM systems allowing for validation and comparison of the method with the existing literature. This section extends the methodology to encompass a much richer range of systems and processes that can be described using the DEM. Examples of two of the potential systems that motivate DETO_3D are shown in Fig. 3.4. In a) Here the simply supported beam system considered previously is repeated but subjected to a soft impact scenario from above (simulated as an imposed initial velocity) that results in fracture. Optimising this system requires several previously unconsidered extensions. Firstly bonds are breakable, and contact mechanics is considered between particles post fracture. Secondly the behaviour of the system is dynamic with the fracture and breakdown of the system occurring over a specific time interval. In Fig. 3.4b) In this case since the system behaviour is more complex than in previous examples optimality becomes harder to define. A complementary energy optimisation is complicated by the energy dissipation that occurs through fracture and damping. Therefore when generalising for systems of this sort it is often necessary to consider an objective defined on a case by case basis that may in fact be defined as a combination of multiple sub-objectives derived from one or more simulations performed on the same system. This is a challenging task for someone with experience of optimisation and any such objective would need verification to be of value.

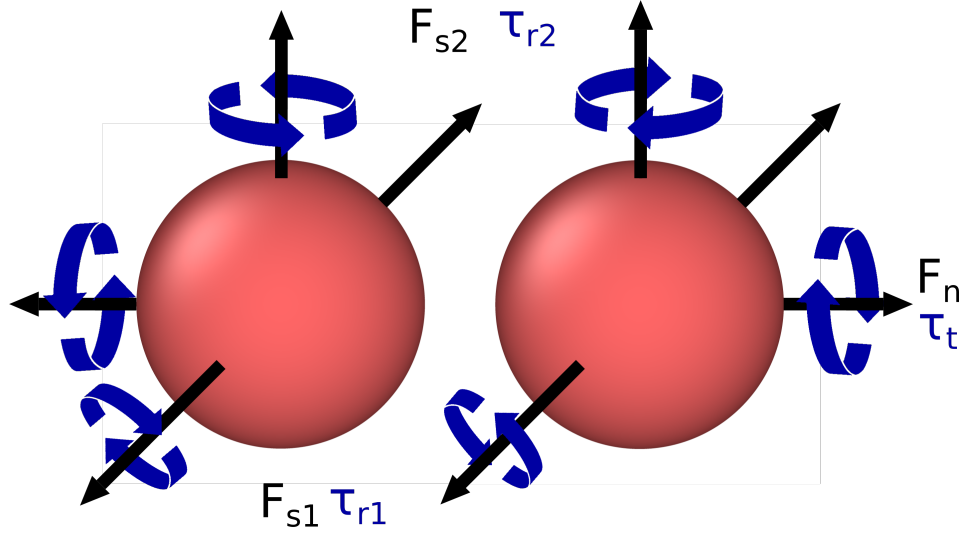


Figure 3.5 Schematic of the 6 degrees of freedom between two interacting particles.

Application of the method to three dimensional problems requires no major methodological additions except that now the DEM simulations are in 3D. Particles are now spherical, they can move along a z coordinate, and forces, stresses, displacements and other vector properties of the system are computed in their three dimensional forms. The simple lattice systems in Fig. 3.2 can be easily extended with a z direction and modeled as hexagonal close packings of 3D particles. The method can also operate on systems not defined by simple lattices, instead incorporating complex base geometries that may or maynot be mechanically stable, depending on the type of interaction potentials employed.

A more significant aspect that can be generalised is the type of interaction potentials that may be employed. The harmonic interaction previously shown in Fig. 3.1b) involves only pairs of particles and features an interaction energy that depends only on the distance between particles. A first extension of this idea is to consider still pairwise interactions but that may also depend on the relative orientations of the particles in 2D this would include rolling whereas in 3D this would account for all the six independent relative movements shown in Fig. 3.5: Normal motion along the radial direction, two shearing motions along the tangential directions, twisting around the particles normal direction, and two types of rolling as shown in Fig. 3.5:

$$\begin{aligned} F_n &= k_n \Delta u_n, F_{s1} = k_{s1} \Delta u_{s1}, F_{s2} = k_{s2} \Delta u_{s2} \\ \tau_t &= k_t \Delta \alpha_t, \tau_{r1} = k_{r1} \Delta \alpha_{r1}, \tau_{r2} = k_{r2} \Delta \alpha_{r2} \end{aligned} \quad (3.17)$$

Δu_r , Δu_{s1} , Δu_{s2} are the relative displacements in the normal and tangential directions and $\Delta \alpha_t$, $\Delta \alpha_{r1}$, $\Delta \alpha_{r2}$ are the relative angular displacements between the particles caused by twisting and rolling. If harmonic springs are associated to each relative motion, then each such motion will have its own associated stiffness to represent a linear force displacement relationship in all six degrees of freedom. In this case the simple penalisation scheme in Eq. 3.6 can be applied to each stiffness for inclusion in DETO. Extension to multi-body interaction potentials, involving more than two particles each, is also possible but it is not treated here because pairwise interactions are more commonly employed in systems of particles at length scales above the micrometre, which are the main focus in this thesis.

Such systems involving granular materials often model particle to particle contact via force field style pair interactions defined between particles that are within a cutoff distance, as opposed to the predefined neighbour lists used in the previous section the set of active interactions typically changes over time as particles move in and out of each others neighbourhood. For example, a popular style of interaction for granular contact is the so-called Hertzian contact (Hertz, 1882) for contact between overlapping elastic particles of homogeneous material. The model predicts an elastic force proportional to the overlapping volume of the spheres. For two equally sized spheres given by:

$$F_n = \frac{4}{3} \frac{E \sqrt{R}}{1 - \nu^2} \Delta_n^{3/2} \quad (3.18)$$

where E is the material Young's modulus, R is the sphere radii, μ is the Poisson's ratio, and Δ_n is the size on the overlap between spheres. This model is shown to agree well with experiments and is therefore often favoured over simple springs for modeling granular flow and rigid body impact. Separate contact models also exist for shearing between particles, most notably, (Mindlin, 2021) and for twisting and rolling spring-dashpot-slider models such as in (Marshall, 2009) are standard. Substituting these more complex interactions into Eq. 3.11-3.13 it can be seen that incorporating them into DETO adds no additional complexity since each interaction receives an independent penalisation using the same principle as Eq. 3.6.

History dependent and irreversible effects can be considered as well, such as in the case of material fracture from Fig. 3.4. This is typically modeled by the addition of a bond breakage criteria to a bonded continuum such as the lattice structures already described. In practice this

means comparing the deformed potentials over the course of a simulation with a critical stress, or strain, depending on the model. Potentials exceeding this critical value are removed from the simulation irreversibly until the end of the simulation. This simple criteria can be used to effectively model the breakage of quasi-brittle materials, that may exhibit crack growth prior to catastrophic failure. If post fracture behaviour is to be meaningfully included it is common to pair these kinds of breakable bonds with a contact model such as the Hertzian described above.

The static analysis performed in DETO_2D can not properly track history dependant behaviours, as only the final equilibrium position of the system and not the trajectory taken in reaching this is considered a valid system state. In the context of crack formation, it cannot be guaranteed that during the course of the energy minimisation the system will deform such that crack initiation and propagation will be captured. The inclusion of any history dependent interactions therefore requires the utilisation of a dynamic DEM routine performed over a finite time period.

In a dynamic analysis, if a complimentary energy cost function is used it should consider the full history of strain energy and extract, for example, the average value if the overall performance is to be considered such as in the example of the Rehometer, or the maximum value reached during the dynamic response if intensity is considered such as in the example of beam impact. If then the beam under goes bond breakages, energy dissipation may have to be considered as well when evaluating the cost function. In this case minimizing the energy dissipated by fracture is more advantages as has been done in the previous works summarized earlier in Fig. 2.6 this idea will be explored in the context of DETO in Chapter 5.

To generalise to the full range of cases that could arise for discrete element system optimisation it is often necessary to specify a bespoke cost function on a case by case basis. Objectives can be any property of the system, or even a linear or non-linear combination of such properties. Such multi-objective optimisations may in fact draw objectives from a number of different simulations on the same base system, for instance incorporating the performance of a structure in response to a number of possible loading scenarios or mixing the dynamic and static response of a system. The associated sensitivities can be derived either analytically via a direct derivative or numerically via the finite difference method.

3.2.1. *Software Implementation: DETO_3D*

This section describes a software implementation called DETO_3D for the generalised DETO method described above. DETO_3D uses the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) software library (Thompson et al., 2022) as its simulation engine. All the DEM simulations to evaluate the objective function are handled by LAMMPS whilst the DETO_3D code acts as a wrapper initialising the simulations and handling the optimisation process. The code is designed in a object oriented framework; The crucial objects that make up the system are described briefly highlighting their relevance to the optimisation process. The code inherits many of its features, input script style and design patterns from LAMMPS and from the MASKE code (Alex and Masoero, 2022) which is itself derived from LAMMPS.

Optimisations in DETO_3D essentially follow the same steps as in the flow chart for the previous code implementation in Fig. 3.3 with a single optimisation being derived from many repeated simulations on different system configurations. The aim is to determine an optimal configuration defined by a vector of per particle χ values. Possible configurations are assessed by executing simulations on these specific configurations. In DETO_3D a simulation is a list of LAMMPS commands to set the boundary conditions and analyse the system with either a dynamic or static DEM routine. Performance is characterised by one or more objectives each associated with a particular simulation. At the end of a simulation these are extracted in order to compute the sensitivities for the update step. This is done in one of two ways depending on the choice of optimisation type between the analytic direct derivative approach or the numerical finite difference approach if no analytic derivative can be obtained. In the former all information necessary for the update is computed directly from the analytic definition of the sensitivity, whereas in the later a significantly more costly round of simulation must take place on a complete population of χ vectors each with a small perturbation of a single value. The contribution of each perturbation to the overall objective function is used to construct a proportional sensitivity value following Eq. 2.7.

Parallelisation is a crucial feature of DETO_3D. The code takes advantage of parallelisation in two separate ways, firstly a LAMMPS instance can be initialised on a group of processors known as a sub-communicator which can perform simulations utilising parallel computations. Specifically a domain decomposition method (Plimpton, 1995) is used to split the computations

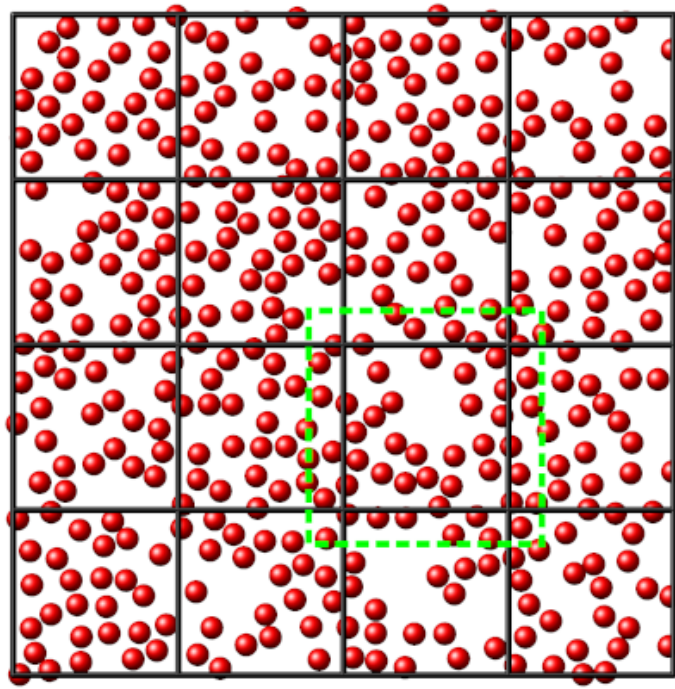


Figure 3.6 Basic domain decomposition scheme for sharing a simulation between processors in LAMMPS, the black lines are the processor grid and particles are handled by the processor assigned to their grid square, the green dotted line represents the skin for communicating ghost atoms see the official LAMMPS documentation for further details (Plimpton, 1995)

involved in a simulation for increased efficiency. Subdivisions of the simulation domain are assigned to individual processors and particles within these subdomains are considered by their respective processor where the computations of force, velocity, and displacement for them are handled. Particles may move across subdomains in which case LAMMPS uses the message passing interface (MPI) parallel computing standard to communicate the particle properties to the processor that then becomes the new owner. Particles near the edges of their respective domain can be close enough to interact with particles owned by separate processors therefore information about the properties of these particles is passed to neighbouring processors where they are stored as so called ghost atoms. Fig. 3.6 shows a typical division of a domain into uniform bricks with an extended communication cutoff for passing ghost atoms.

DETO_3D may manage several LAMMPS instances in a single optimisation. Instances are initialised to separate sub-communicators. Allowing for simulations to be run on separate configurations in parallel. Information regarding objectives is communicated between sub-communicators after these simulations complete. This can dramatically increase efficiency when computing sensitivities numerically via the finite difference method, since a large population of perturbed χ configurations need to be assessed these can be divided equally between available

sub-communicators to run the simulations. These two levels of parallelisation create a trade of between allocating more processors to individual sub-communicators and creating more sub-communicators with fewer processes. Bench marking results are shown in Chapter 5 to explore the efficiency gained from each approach.

The input scripts in DETO_3D follows a similar principle as the input scripts in LAMMPS, where commands are written to initialise a simulation, define systems of particles and interactions, set simulation properties, and run the simulation. Each command is executed in order and causes the code to take some immediate action. A LAMMPS input scripts could contain for example the instructions to initialise a set of particles, define bonds and contact potentials between them, impose boundary conditions such as forces, velocities, and constraints, set a time step and output options, then run a dynamic simulation routine over a specified number of steps.

DETO_3D input scripts allow running any of these LAMMPS-specific operations as well as additional operations that are specific to optimisation problems. This is because in fact DETO_3D in many cases redirects commands to specific LAMMPS instances to execute the necessary simulations and extracts the result to progress the optimisation. As such commands are grouped in several ways; firstly those that set the initial conditions before optimisation, these are only run once and a snapshot is taken of this state that will be returned to before each simulation, secondly commands that define a simulation. These are stored as attributes of a given simulation and run repeatedly each time a simulation result is required by the optimisation. The rest of the commands in a DETO_3D input script deal with defining the properties of the optimisation itself and are not executed by LAMMPS at all. These commands can can define a *Universe* of subcommunicators available to the code, add new simulations to the optimisation and specify a user defined objective function.

The choice of objective function available in DETO_3D is only limited to any arbitrary combination of values ⁴ that can be stored as scalar internal variables of a LAMMPS simulation. The precise details of what can be concluded are contained in the official LAMMPS documentation. Essentially any combination of parameters derived from the positions, velocities, forces,

⁴One important exception and limitation to the freedom of cost function definition involves combining objectives from two simulations that use a direct derivative and finite difference approach respectively. Combining these two methods is possible in DETO_3D but necessitates that the sensitivity of each objectives be computed individually by their respective method first before being combined. This leads to an additional constraint that objectives that use different sensitivity update styles can only be combined via simple addition in a cost function such as in Eq. 5.2 rather than more complex combinations that would lead to a mixture of terms in the derivative.

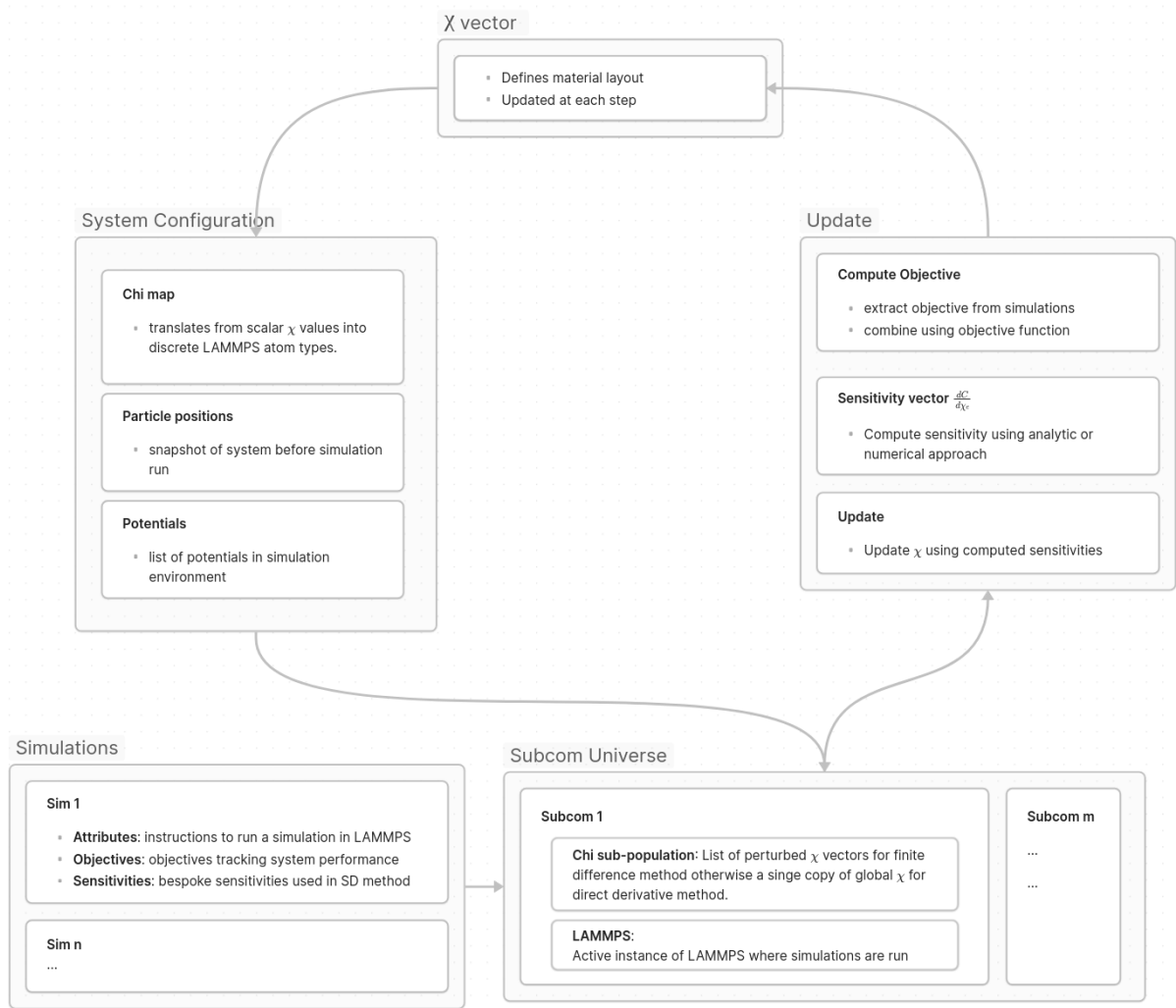


Figure 3.7 Schematic of the main optimisation loop in DETO_3D, specifying the main high level objects in each optimisation

or energy of the particles in the system can be included as well as system wide information, such as number of broken bonds or granular contacts that can be tracked across the course of a simulation run, if a direct derivative approach is used the sensitivity equation must also be user provided, but a finite difference method can compute this from the cost function allown. Individual objectives are extracted after their associated simulations have been executed. After this is complete the objective function is evaluated to combine all sub objectives.

The structure and operation of the DETO_3D code can be encapsulated in a number of high level objects defined laid out in the diagram show in Fig. 3.7.

When the DETO_3D code begins execution it will first divide all available processors as evenly as possible between the specified number of sub-communicators and initialise a fresh LAMMPS instance on each sub communicator. It then executes the initial LAMMPS commands on each instance to ensure that all parallel instances are in the same initial state. This initiation

χ	type	per particle properties
0.05	1	...
0.1	2	...
...	...	...
1.0	20	...

Table 3.2 Example *chimap* particles are loaded in by assigning them to their closest value in the χ column, then setting type and any other defined parameters to the corresponding values from that row.

stage is not limited to constructing initial geometry and boundary conditions but can itself contain dynamic time stepping or energy minimisation to set the conditions for the optimisation. This state will be returned to repeatedly throughout the process.

The code loads a specific system configuration in the form of per particle χ values into the simulation before each simulation. In LAMMPS each particle has an associated numeric type the primary purpose of which is to map the properties of interaction potentials that can be defined to exist between particles to of specified types. Scalar χ values in DETO_3D are mapped their nearest equivalent particle type inside LAMMPS, with the mapping between χ and types being defined in a separate file called *chimap* an example of a typical *chimap* is shown in Table. 3.1 this file can also associate chi values to any other per particle properties in addition to types such as mass, diameter, or even charge, so that these values would also change during an optimisation process whenever the chi value of a particle changes. A complete example of a *chimap* used for optimisation can also be found in AppendixA. Interaction potentials are then defined between particle types; a penalisation scheme such as the one in in Eq. 3.6 must be explicitly respected by the user defined potentials, which allows for an unrestricted use of any of the many styles of interactions available in LAMMPS and for the possibility of different or more complex penalisation schemes.

Once all simulations are complete on a given system configuration the full objective function is constructed as defined in the input script. The flexibility of this approach allows for any linear or non-linear combination of objectives to be defined as the criteria for optimality. Finally a sensitivity vector is constructed by either the direct derivative or Finite difference approach and an update step is undertaken before a tolerance is checked and the code either exits or repeats the optimisation loop.

3.2.2. *Worked Example*

Here it is useful to understand the DETO_3D software through a worked example. A step-by-step description of how to undertake an optimisation problem in DETO_3D, including setting up dimensions, configuring DETO input files, specifying boundary conditions, running optimisations, and analysing output. The aim of this section is to allow the interested reader to be able to repeat the optimisations shown in this thesis by giving a detailed explanation of the operation of the DETO software and how it is used.

A secondary aim of this section is to validate the method by comparison to published results, therefore a well known benchmark problem has been chosen for this example. The problem selected is the L Shape Test, a popular benchmark from the literature, as described in (Valdez et al., 2017) Problem 4.5.

LAMMPS integration

DETO_3D is heavily integrated with LAMMPS which is central to constructing and executing simulations. DETO_3D's primary operation links a user-defined LAMMPS model to a list of interaction properties and a set of boundary conditions to apply in a simulation. The main loop of the DETO_3D program iteratively executes the simulation, updating the interaction properties of the model between each iteration towards an optimal solution.

The ground conditions for the optimisation should be defined in a LAMMPS script that

- Defines initialisation parameters.
- Creates and defines the initial position of all particles in the simulation.
- Specifies constraints and the types of any particles that are not included in the optimisation.

The following script is provided for the L-shaped example.

```
# ===== SET INITIALISATION PARAMS =====  
dimension 2  
units si  
boundary s s p  
atom_style hybrid sphere bond
```

```
comm_modify vel yes cutoff 3.2

newton off

special_bonds lj 0 1 1 coul 0 1 1

timestep      10-4

# ===== SYSTEM DEFINITION =====

variable      nelx equal 50
variable      nely equal 50
variable radius equal 0.5

region box block 0 $(v_nelx) 0 $(v_nely) -0.01 0.01
region cutout block 20 $(v_nelx) 20 $(v_nely) -0.01 0.01 side out
region lshape intersect 2 box cutout
create_box 12 box bond/types 78 extra/bond/per/atom 12

region support_reg block INF INF 49 INF INF INF
region force_reg block 49 INF 7.2 8.8 INF INF

lattice      hex $(v_radius*2)
create_atoms 6 region lshape
fix          1 all nve

neighbor 2.2 bin
neigh_modify delay 0

group support_group region support_reg
group force_group region force_reg
fix support support_group setforce 0 0 0
set group support_group type 12
set group force_group type 12
```

```
# ===== DEFINE OUTPUT VARIABLES =====  
  
compute stress all stress/atom NULL bond  
  
variable shyd atom -(c_stress[1]+c_stress[2]+c_stress[3])/3  
  
variable sdev atom sqrt(0.5*((c_stress[1]-c_stress[2])^2+(c_stress[2]-c_stress[3])^2+(c_stress[3]-c_stress[1])^2)/3)  
  
compute nbond all nbond/atom  
  
compute tbond all reduce sum c_nbond
```

The commands above are all LAMMPS commands and in fact this script can be run natively in LAMMPS. The script begins by initializing parameters that must be defined before particles are added. Following this, the script defines all particles that will be involved in the optimisation. This is done here by filling an L-shaped region with a regular lattice of particles but these particle positions could just as easily be read from a file. All particles are initialised as type 6 at the start of the optimisation, but these types will be subject to change as the optimisation progresses. Next boundary conditions are set. Static constraints are applied to the top of the L-shape and the region where force will be applied is also set. Particles in these regions are set to type 12 which they will retain throughout the optimisation as long as this is specified when we describe the DETO inputs in the next section. It's noteworthy that the script refrains from specifying forces or interaction potentials between particles at this point, rendering the system static. The introduction of dynamics is included outside of this script. Only the ground conditions of the optimisation should be included here.⁵ The last section of the script, while not imperative for executing an optimisation, defines a set of output variables. These can be included in dump files, for subsequent analysis.

Configure Chi variable and interactions

The parameter chi can vary throughout an optimisation for each particle. In DETO_3D chi is specified to be selected from a list of discrete possibilities for each particle, this importantly links each value of chi with a specific LAMMPS particle type, that will specify the intensity of

⁵It is possible to add dynamics into this section of an optimisation. This could be useful for example if you want to optimise a system mid process. The commands in this script can be thought of as being used to set the system into the condition it has at the start of the optimisation. Any dynamic behaviour defined here will be run only once and not repeated during each iteration.

interaction potentials. This is done in a so called Chi Map. A chi map must be included in a DETO optimisation and must include at least three columns chi, material, and type but could include more linked to any per particle quantity such as mass or charge for example. For L shaped beam example the a basic chi map is used featuring 1 material and 11 chi values.

```
num_mat 1
```

```
PROPERTIES: chi material type
```

```
0.001 homo 1
```

```
0.1 homo 2
```

```
0.2 homo 3
```

```
0.3 homo 4
```

```
0.4 homo 5
```

```
0.5 homo 6
```

```
0.6 homo 7
```

```
0.7 homo 8
```

```
0.8 homo 9
```

```
0.9 homo 10
```

```
1.0 homo 11
```

Complementary to the Chi Map defined here, is the Potential File which links the defined particle types and therefore chi values to the intensity of particle to particle interactions. This is done by defining individual unique interactions between each potential combination of Chi values this means that the number of potentials to be defined is given by:

$$\frac{n!}{2!(n-2)!} \quad (3.19)$$

Where n is the number of chi values specified plus one. If additional atom types are included that are not associate to a chi and therefore static this number increases again. As can be seen this can become a very long file even with a relatively low number of chi values. In this example since only 11 chi values where used with one static type the number of unique interactions is 78. this file is shown abbreviated below to avoid taking up unnecessary page space. However can be found in full in the appendices of this thesis.

The structure of this file will vary heavily depending on the types of interaction included in the simulation, but here the file follows the pattern of:

- defining a group associated with each type
- defining a bond coefficient for each interaction
- creating bonds of the correct type between each type and using the defined coefficients
- deleting the groups so that they can be re-created once the particle types change in the next iteration.

```
pair_style zero 1.0
pair_coeff * *

group 1 type 1      #chi equal 0.0
group 2 type 2      #chi equal 0.1
group 3 type 3      #chi equal 0.2
group 4 type 4      #chi equal 0.3
group 5 type 5      #chi equal 0.4
group 6 type 6      #chi equal 0.5
group 7 type 7      #chi equal 0.6
group 8 type 8      #chi equal 0.7
group 9 type 9      #chi equal 0.8
group 10 type 10     #chi equal 0.9
group 11 type 11     #chi equal 1.0
group 12 type 12     #non-opt

bond_style harmonic
bond_coeff 1 0.001 1
bond_coeff 2 0.001 1
bond_coeff 3 0.001 1

...
```

```
bond_coeff 76 81 1
bond_coeff 77 100 1
bond_coeff 78 100 1
create_bonds many 1 1 1 0.9 1.1
create_bonds many 1 2 2 0.9 1.1

...

create_bonds many 11 12 77 0.9 1.1
create_bonds many 12 12 78 0.9 1.1

group 1 delete
group 2 delete
group 3 delete
group 4 delete
group 5 delete
group 6 delete
group 7 delete
group 8 delete
group 9 delete
group 10 delete
group 11 delete

#12 particle types
#78 bonds created
```

DETO script

The DETO input script sets up and runs an optimisation in DETO. It combines the LAMMPS model, chi map and potential file from the previous steps together and defines the forces and

dynamic or quasi-static conditions to optimise against in the form of a simulation. The commands used in this script are unique to DETO and so will not run natively in LAMMPS.

```
# ===== SET OPTIMIZATION STYLE AND PARAMS =====
opt_type 0.1 0.5 gradient_descent
objective_function v_c1

# ===== SET UNIVERSE OF SUB-COMMUNICATORS =====
subcomm 1

# ===== INITIAL SETTINGS FOR ALL LAMMPS INSTANCES =====
lammps file ./inputs/in.Lshape_lmp
dump      1 all custom 1 ./dump/dump.Lshape id type x y v_shyd v_sdev c_nbond

# ===== LOADING CHI MAP AND POTENTIALS FILE =====
opt_map_chi ./inputs/chimap.dat
read_potentials ./inputs/potfile.dat

# # ===== SET SIMULATION NUMBER 1 (RUN TYPE, NO REPEAT)=====
simulation Sim1 run      repeat no

add_attribute Sim1 fix      force force_group addforce 0 -0.1 0
add_attribute Sim1 fix_modify force energy yes

add_attribute Sim1 minimize      1.0e-12 1.0e-12 1000000 20000000

#Define variables for cost function
add_attribute Sim1 variable cost_func equal ebond

# #===== objective variables =====
add_objective Sim1 c1 cost_func
```

The function of the commands in the script below in order of execution are:

- First `opt_type` is used to specify a gradient descent optimisation method with a move limit of 0.1 and a total chi fraction of 0.5 meaning that between each step a given particles chi value cannot fluctuate more than 0.1 and the optimisation is constrained to keep the average chi across the system below 0.5.
- A single objective function is specified with `objective_function` this will be linked to a LAMMPS variable with a later command
- One sub communicator is specified meaning the program will run only a single execution thread.
- The `lammmps` commands specified in `in.Lshape_lmp` are read into the system and executed on a LAMMPS instance, to generate the initial configuration of the system. The commands could instead have been written into this script each prepended by the keyword `lammmps`. However it can be useful to store them in a separate file as is done here.
- The `dump` command initialises a optimisation wide dump file which takes the same inputs as the `lammmps dump` command except dumps an output ever N optimisation steps instead of simulated time steps here N is specified as 1 to capture a dump after each optimisation step.
- `opt_map_chi` reads in the previously defined chi map.
- `read_potentials` reads in the previously defined potentials file.
- The simulation command instantiates a simulation called `Sim1` to be run as part of this optimisation.
- The `add_attribute` command is used to specify LAMMPS commands that will be run on each iteration for `Sim1`. These add a downwards force of 0.1 to the `force_group` defined in `in.Lshape_lmp` and then initialise an energy minimisation. Finally the bond energy of the system is captured in the variable `cost_func` after the simulation has been run.
- Finally the `add_objective` command links the LAMMPS variable `cost_func` to the DETO specific variable `c1` to be included in the objective function.

This simple script illustrates a complete DETO optimisation. However it can be straightforwardly expanded to include multiple simulations and more complex composite objective functions as will be described in later chapters.

Running an optimisation and analysis

The script above if stored in a file called `in.Lshape` can be run from the command line using

```
./deto in.Lshape
```

Whilst running DETO will output some useful information to the terminal at each optimisation step such as the objective function and volume fraction. However the output of the optimisation is largely analysed after the fact through dump files that can record much more extensive information on the system between each optimisation step. These outputs can be configured in much the same way that standard LAMMPS dumps can be to track per particle values. In the example above the line:

```
dump      1 all custom 1 ./dump/dump.Lshape id type x y v_shyd v_sdev c_nbond
```

is included. This means that for all particles at each optimisation step the ID, type, x and y coordinates, hydrostatic and deviatoric stress, and number of intact bonds are recorded into a file named `dump.Lshape` for analysis.

A typical way to view this dumped information is via the open source visualisation software Ovito (Stukowski, 2010) where particles can be visualised and colour coded with respect to their properties. By displaying the particle type it is possible to get a straightforward visual representation of the distribution of chi at any stage of the optimisation and on the layout of the optimised structure. For example here the conventional "boot" optimal structure can be seen at step 150 of the optimisation.

Comparison of Fig. 3.8b) and Fig. 3.9b) can be undertaken like for like and by overlaying the two solutions shows a direct co-relation of the optimal topologies.

Other parameters such as stress distribution can be displayed as well giving insight into the location of stress concentrations in the structure such as in this case at the top inner corner of the L.

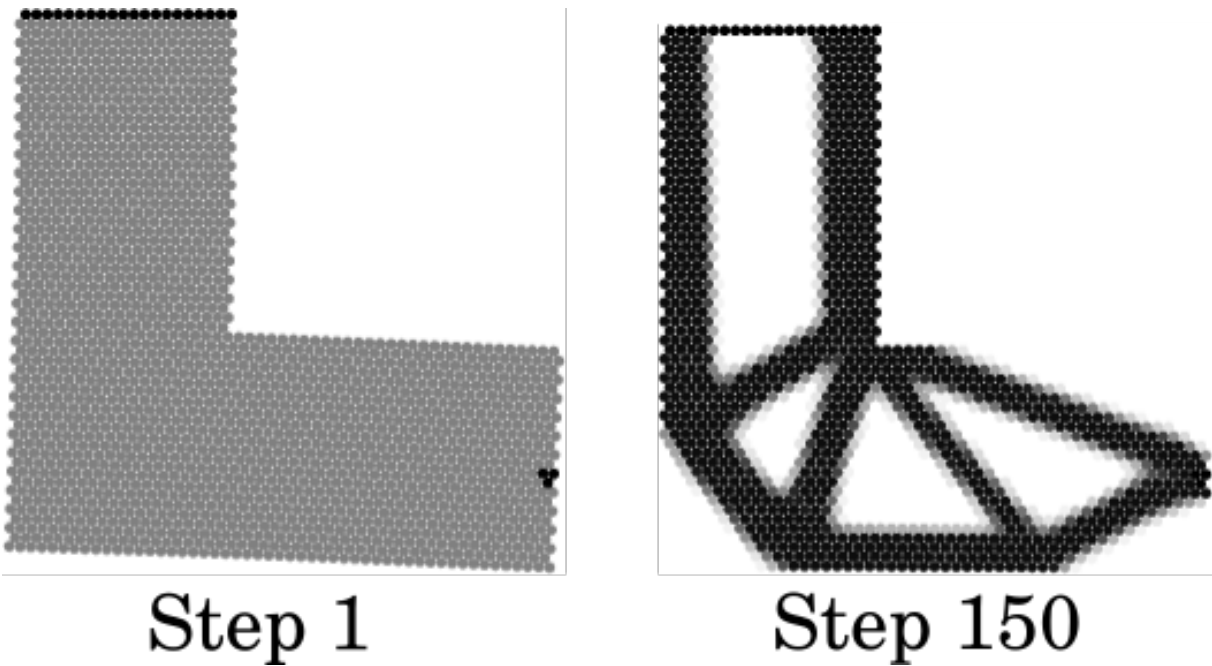


Figure 3.8 Visual representation of particle type at step 1 and step 150 of an optimisation where the darker black corresponds to a higher type and therefore greater chi value

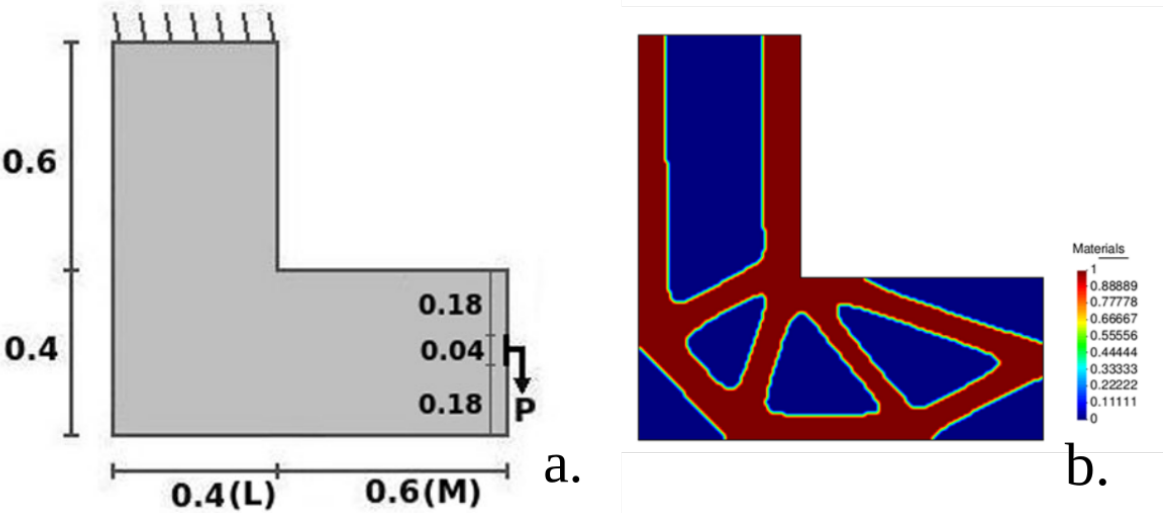


Figure 3.9 Benchmark example of the L-Shape problem from Valdez et al. (2017) Problem 4.5. showing the boundary conditions and optimised result

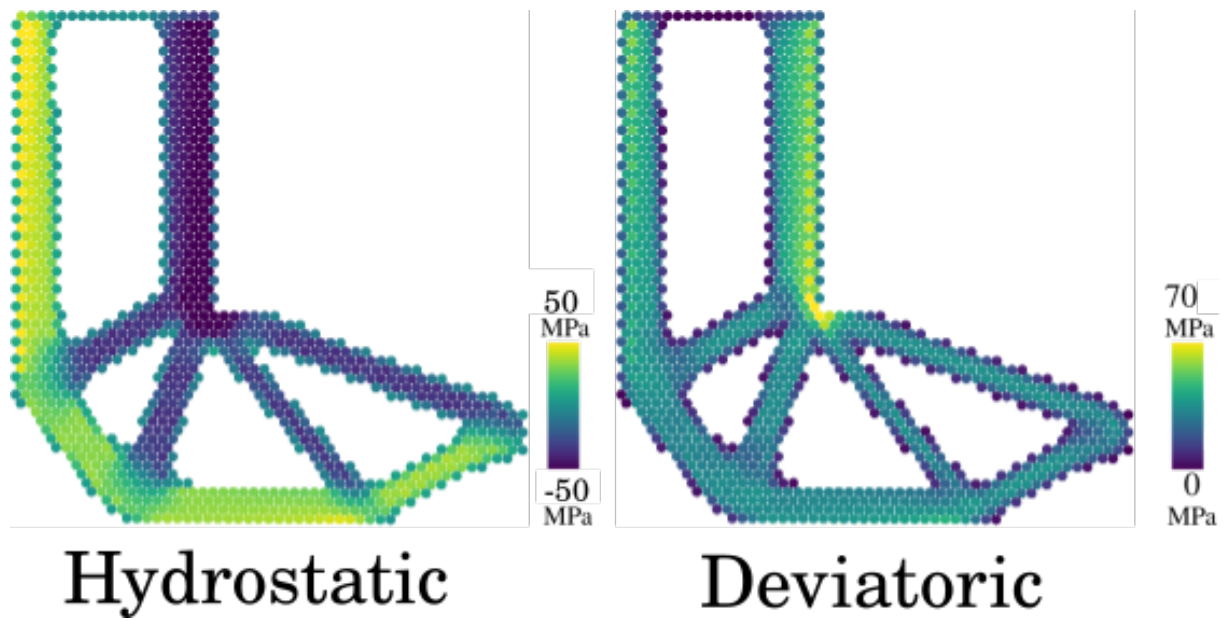


Figure 3.10 Visual representation of per particle of hydrostatic and deviatoric stress distribution in the optimised structure

In addition to per particle dump files DETO will also output a so called thermo file by default called 'thermo.objective' that tracks the progression of the defined objective function over the course of the optimisation.

Chapter 4. Proof of concept results: quasi static beam problems

This chapter presents an initial set of results for quasi-static truss problems following the framework in Chapter 3.1 and generated with the DETO_2D code. First a validation of the method against equivalent methods from the literature for the case of simply supported trusses using linear elastic harmonic bonds is shown. Then a study on the effects of various operational parameters (mesh fineness, solid volume fraction and filtering length) are presented to highlight the effect on the optimisation process. Geometric non-linearities are shown to already be incorporated at this stage. Finally, examples of optimisations utilising four different non-linear interactions are given, which address the applicability of the newly proposed method to systems with material non-linearity. This section serves as a proof of concept for DETO in preparation for the more general results that will be presented in Chapter 5

4.1. Validation

New topology optimisation methods are typically tested on simple structures with known optimal geometries. Two such structures are the simply-supported and pin-supported beam systems shown in Fig. 4.1. The figure shows results from DETO side by side with optimal geometries from established methods. The input parameters for the DETO simulations are shown in Table. 4.1. The intensity of the external force is 1 kN in both cases.

<i>nelx</i>	135	<i>r_{min}</i>	1.5 mm
<i>nely</i>	45	<i>k₀</i>	100 kN/mm
<i>width</i>	135	<i>dt</i>	0.01 μ
<i>height</i>	39	<i>d_{max}</i>	0.01 mm
<i>f</i>	0.6	<i>m</i>	1 mg
<i>D</i>	1 mm	<i>t_z</i>	1 mm

Table 4.1 Input parameters for the DEM system properties used in Fig. 4.1. and throughout this section unless otherwise specified

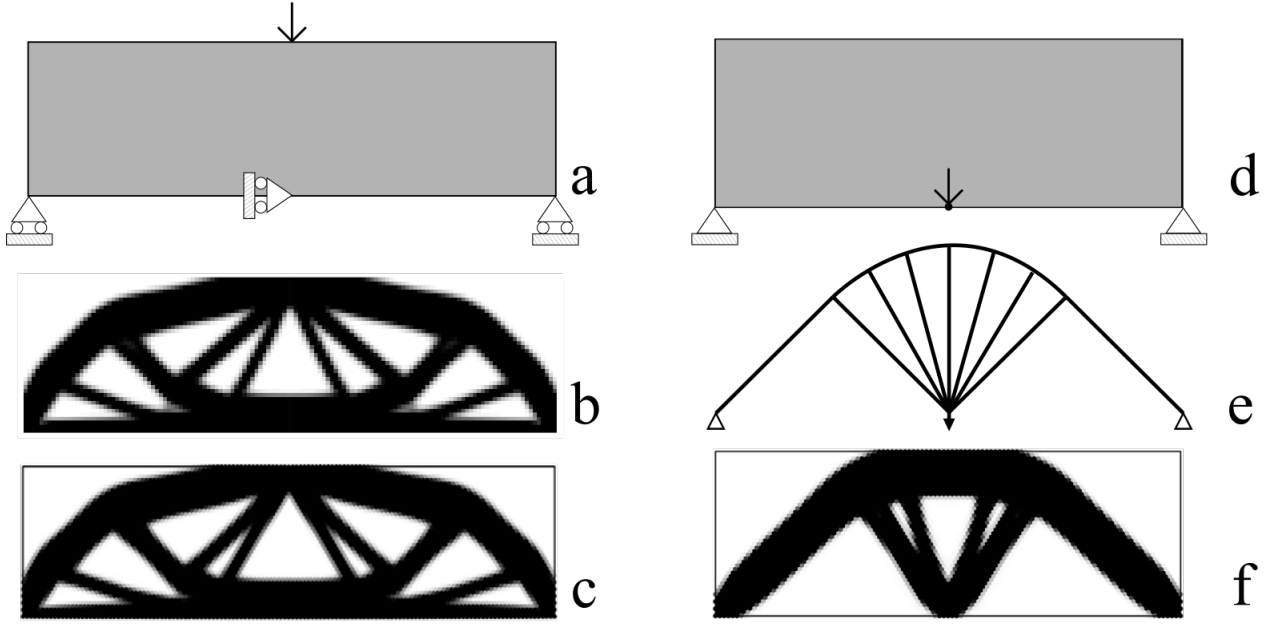


Figure 4.1 Optimal structures for the beam problem under point load. a) Simply supported system with b) result from the FEM-based TO code in (Sigmund, 2001), with inputs: $nelx = 67$, $nely = 39$, $volfrac = 0.6$, $penal = 3$ and $rmin = 2$ (Sigmund, 2001) this example is mirrored across it's central horizontal axis hence the approximately halved $nelx$ value, and c) result from DETO. d) Pin-supported system with results from e) Michell's analysis in (Michell, 1904) and f) DETO which has the dimension inputs $nelx = 109$, $nely = 45$.

1

To improve the physical interpretation of the results, consider that $k_0 \sim \frac{EA}{r_0}$, where E is the Young modulus of the material, t_z is the thickness of the structure in the third direction, A and $r_0 = D$ are the cross-sectional area and the length at rest of the cohesive bridge. the spring connecting neighboring disks. Assuming that the width of the cohesive bridge is proportional to the disk diameter, $A \sim Dt_z$ (Masoero et al., 2014), and rearranging the expression of k_0 we can estimate an equivalent Young modulus:

$$E = \frac{k_0 r_0}{A} = \frac{k_0}{t_z} \quad (4.1)$$

The values in Table. 4.1 return $E = 100$ GPa, thus one can consider the simulated structures as made of sintered metallic powder.

For the simply supported beam in Fig. 4.1.a the optimal geometry from DETO is qualitatively similar to that from the FEM code in (Sigmund, 2001), when similar inputs are provided. For the pinned structure in Fig. 4.1.d, the optimal layout predicted by DETO is analogous to the

¹due to

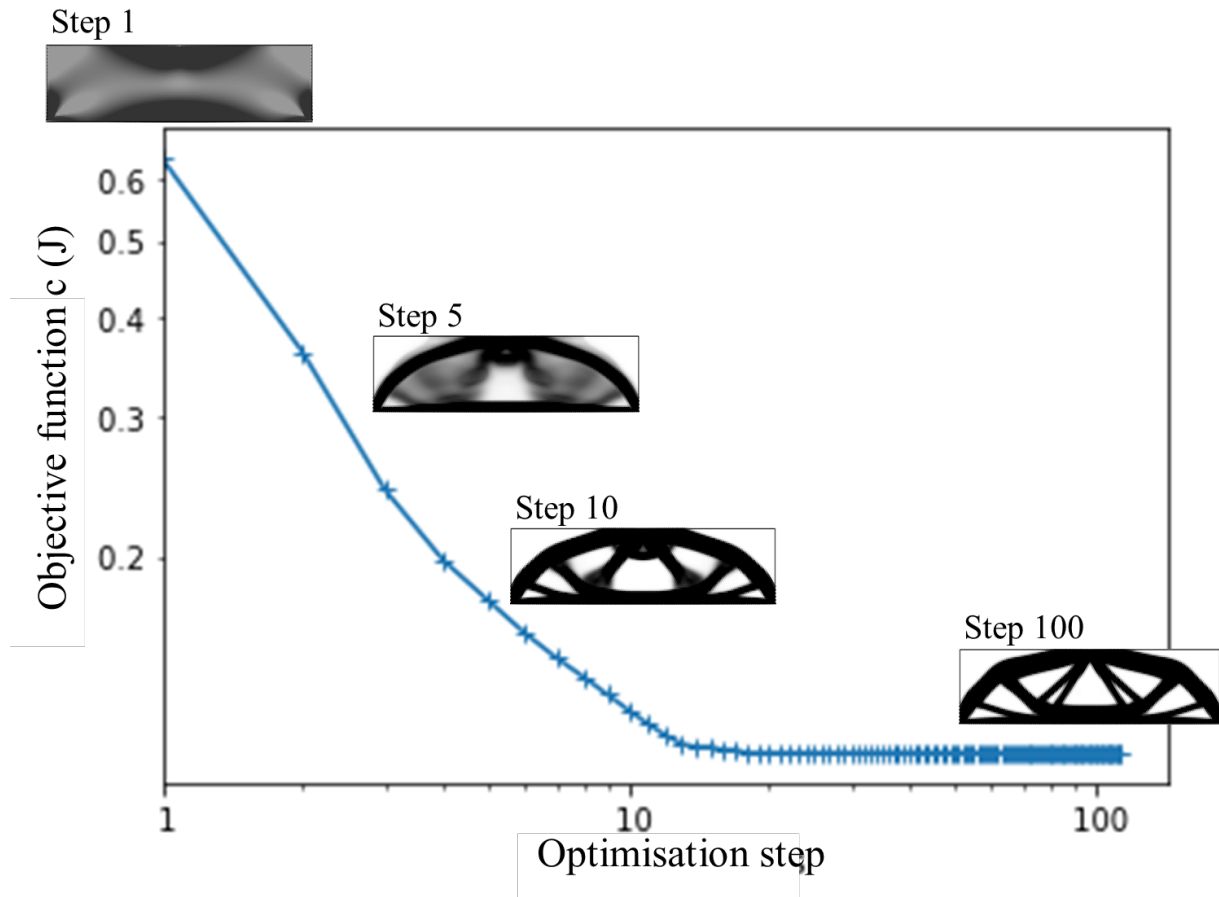


Figure 4.2 Simply supported beam case in Fig. 4.1.a: evolution of geometry and objective function c during the first 100 optimisation steps.

theoretical solution of fully stressed structures of Michell (1904). In this case the similarity is less striking because Michell's solution features one-dimensional members with the additional constraint that all members must have equal cross section. Overall, Fig. 4.1 shows that the optimal solutions obtained using DETO are comparable to those coming from other more established methods in the literature.

Fig. 4.2 shows the evolution of the objective function c (U_{tot}) and of the corresponding geometry during the optimisation process. Significant changes in both c and geometry take place during the first 50 optimisation steps. Between steps 50 and 100, the geometry has practically converged and c remains nearly constant. Fig. 4.3 shows the distribution of hydrostatic and von Mises stresses Eq. 3.16 in the initial structure (left) and the final 0-1 optimized structure (right). The hydrostatic stress distribution in the initial structure shows the expected distribution at midspan, gradually changing from tension to compression. This is lost near the supports, because they are concentrated and placed at the bottom corners of the domain rather than along the central axis of the beam. In the optimised structure, instead, the distribution of hydrostatic

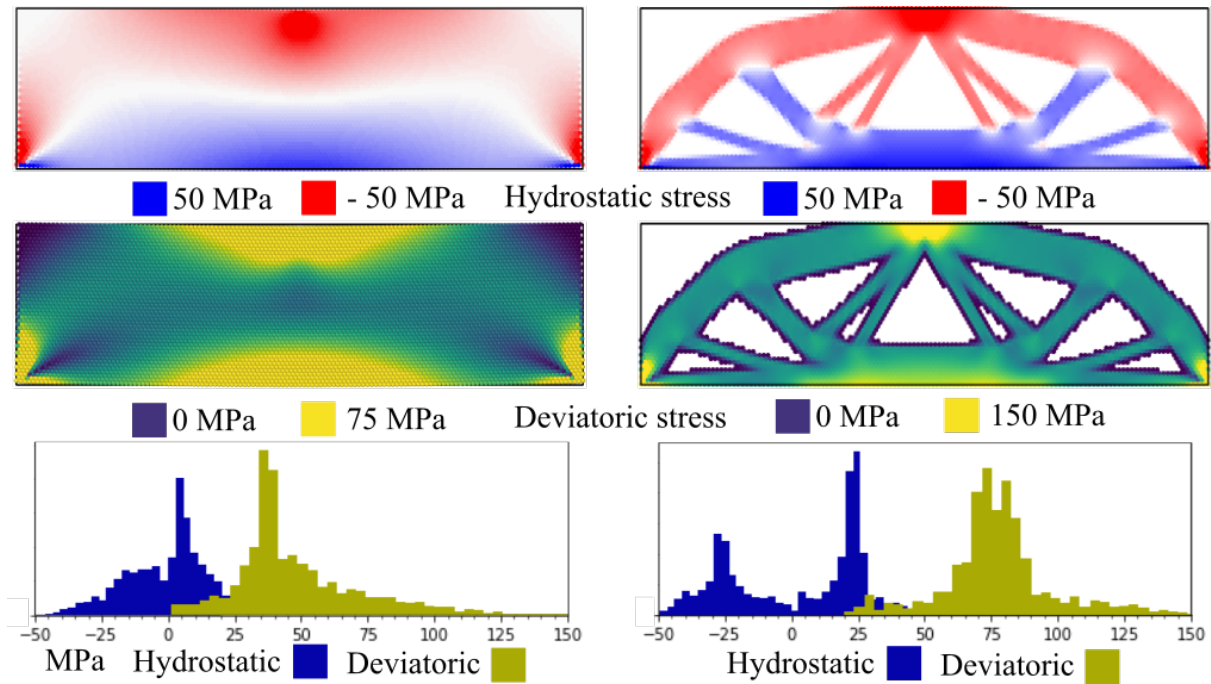


Figure 4.3 Hydrostatic and Von Mises deviatoric stress distributions in the optimized structure from Fig. 4.1.c.

stresses becomes clearly bimodal, which indicates that the elements composing the structure tend to work either fully in tension or in compression. This is desirable for an efficient use of the material. The whereas stress in the initial structure are predictably concentrated at the top and bottom of the beam at midspan, where uniaxial stress from bending is greatest, as well as near the supports, where shear stresses become highly concentrated due to the pointwise nature of the supports themselves. The intensity of the whereas stresses increases during the optimisation process, as a result of the material being used more efficiently, concentrated in fewer elements; a long tail of higher stresses accounts for areas of stress concentration, in particular just under the applied load, at the supports, and at the bottom of the midspan.

4.1.1. Effect of the penalisation exponent p

In this thesis a value of $p = 2$ for the penalisation factor has been used throughout. This choice of p came as a result of test simulations with different values of p . Optimal structures at step 100 for the p values of 1, 2, and 3 are shown in Fig. 4.4.

The $p = 1$ case leads to a structure with a large fraction of “gray” particles, featuring intermediate χ between 0 and 1. This result is similar to what may be obtained in variable thickness sheet problems (Rossow and Taylor, 1973), but it is not a desirable outcome for the

type of optimisation problems studied here, where the final result should be as close as possible to a 0–1, void–solid solution only. $p = 2$ and $p = 3$ both lead to black and white solutions. The structure obtained with $p = 3$ has a more complex topology, featuring more elements, which might make it more difficult to fabricate. This greater complexity may come from the tendency of optimisation problems with high penalization levels, i.e. high p , to get trapped into local minima of the objective function; such tendency is known in the literature on FEM-based TO (Sigmund and Petersson, 1998).

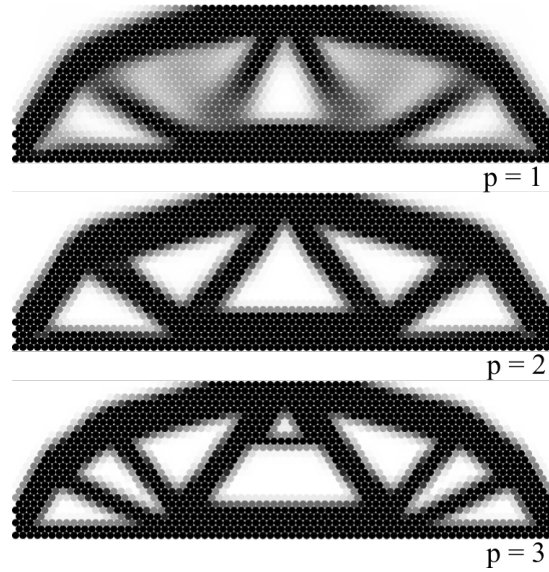


Figure 4.4 optimisation of a beam domain, made of 75×25 particles, with $D = 1$ and filtering of 1.1 applied, simply supported beam at the bottom left and right ends. Three values of penalization factors p are investigated. The snapshot show the solutions at optimisation step 100.

The speed of convergence, for the three cases with different p , is show in Fig. 4.5. All three cases eventually converge to a similar value of the objective function, meaning that the choice of p does not greatly drive the ability to better minimize the objective function. To compare the speed of convergence, the curves in Fig. 4.5 are fitted using an exponential function of the type $f(x) = k_1 + k_2 e^{\frac{n}{k_3}}$, where n is the step number during the optimisation process.

Table. 4.2 shows the fitted values for k_1 , k_2 and k_3 . Among those, k_1 controls the final value of the objective function, hence it is similar for the three cases of p shown here. k_2 , summed to k_1 , gives the starting value of the objective function at step 1, so it correctly increases with p , as the initial structures with all particles featuring χ between 0 and 1 are less stiff when subjected to a high penalization factor. k_3 controls the speed of convergence: the higher k_3 , the slower the convergence, i.e. more optimisation steps are required to attain the final value of the objective function. The values of k_3 in Table. 4.2 are quite similar for the three cases of p considered here,

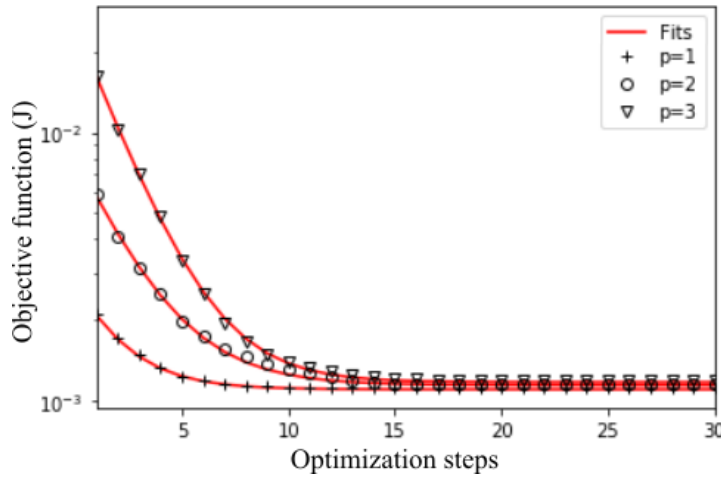


Figure 4.5 Convergence of the optimisation problems in Fig. 4.4. The fits are obtained using the exponential function discussed in the main body of the text.

with $p = 2$ being a bit slower. A common approach to balancing the speed of convergence whilst avoiding local minima is to start an optimisation using a low value of p then gradually increasing it's value as the optimisation progresses. This can be done linearly or in steps, only increasing p when each time a specified level of stability is reached

p	k_1 (J)	k_2 (J)	k_3
1	0.0011	0.0016	2.0
2	0.00114	0.007	2.4
3	0.00117	0.025	2.1

Table 4.2 Constants used for the fits in Fig. 4.5

The results suggest $p = 1$, is likely to generates undesirably gray structures. $p = 2$ is preferred over $p = 3$ as it generates lower topological complexity of the solutions, despite requiring a few more optimisation steps to achieve convergence. Furthermore, $p = 2$ is more recommendable for future studies, knowing from FE-based TO that problems with high p tend to get trapped into local minima.

4.2. Parametric Study

4.2.1. Volume fraction

The base system in Fig. 4.1.c featured a final volume fraction of solid $f = 0.6$, as per Table. 4.1. Here the system is kept the same except f , for which 4 additional values are explored between 0.5 and 0.7. Fig. 4.6a. shows the impact of f on the final 0-1 optimized structure. As expected, small f values force the system to create fewer elements and lead to less optimal solutions, with

higher c compared to more topologically rich solutions at high f . However, these results are not sufficient to determine how much the lower c values at higher f come from topological complexity rather than just having used more material. The next section will add insight to this point.

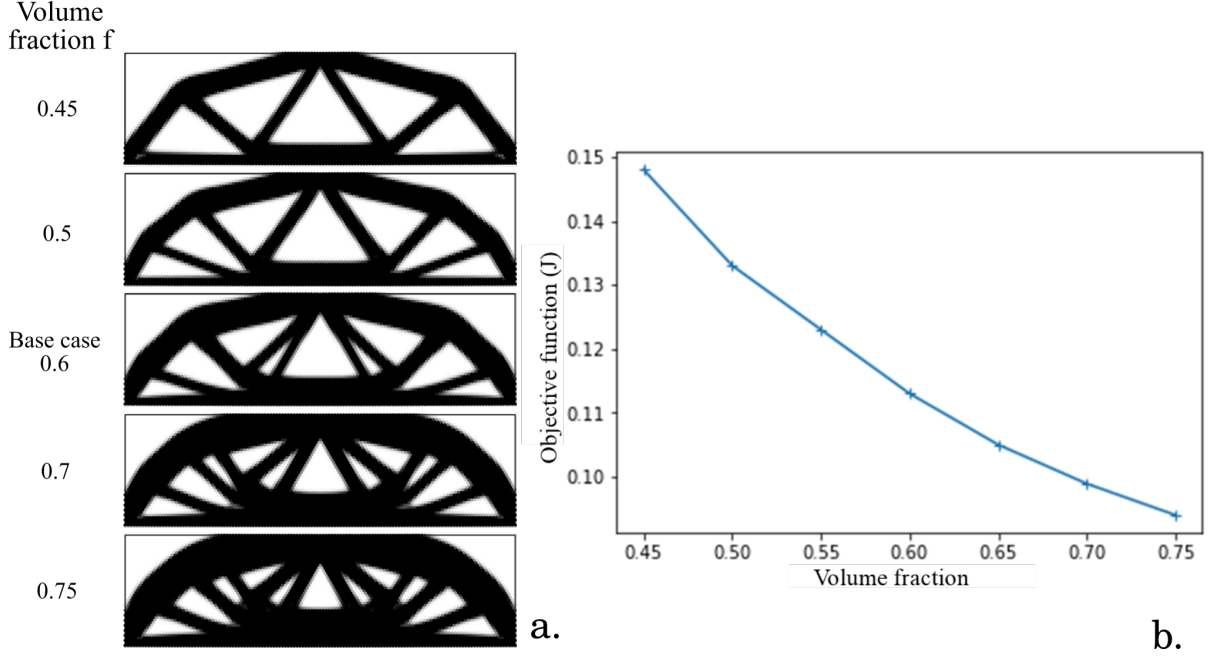


Figure 4.6 a) Effect of target solid volume fraction f on optimized solutions to the simply supported beam problem in Fig. 4.1.a. b) Effect of volume fraction on objective function

4.2.2. Filtering length

Fig. 4.7 shows how the filtering length r_{min} impacts the optimized geometries. The figure also shows a result for the unfiltered case. Predictably, the optimized topologies become simpler at higher r_{min} values, which force the solid to concentrate into fewer, thicker structural elements. Reducing topological complexity by filtering, however, constrains the optimisation problem; as a result, c is expected to increase as the solutions become less optimal at larger r_{min} values. This is confirmed in Fig. 4.7, which shows c growing from 0.111 J to 0.117 J as r_{min} is increased from $1.1D$ to $3D$. This complements the discussion of Fig. 4.6 in the previous section, showing indeed that more optimal solutions can be obtained by increasing topological complexity while keeping f fixed.

As the optimal topologies get simpler with filtering, the structural geometries with $f = 0.6$ for $r_{min} \geq 2D$ in Fig. 4.7 end up resembling those in Fig. 4.6 for smaller $f = 0.5$. Comparing these two examples confirms the expected trend that similar geometries with smaller f lead to

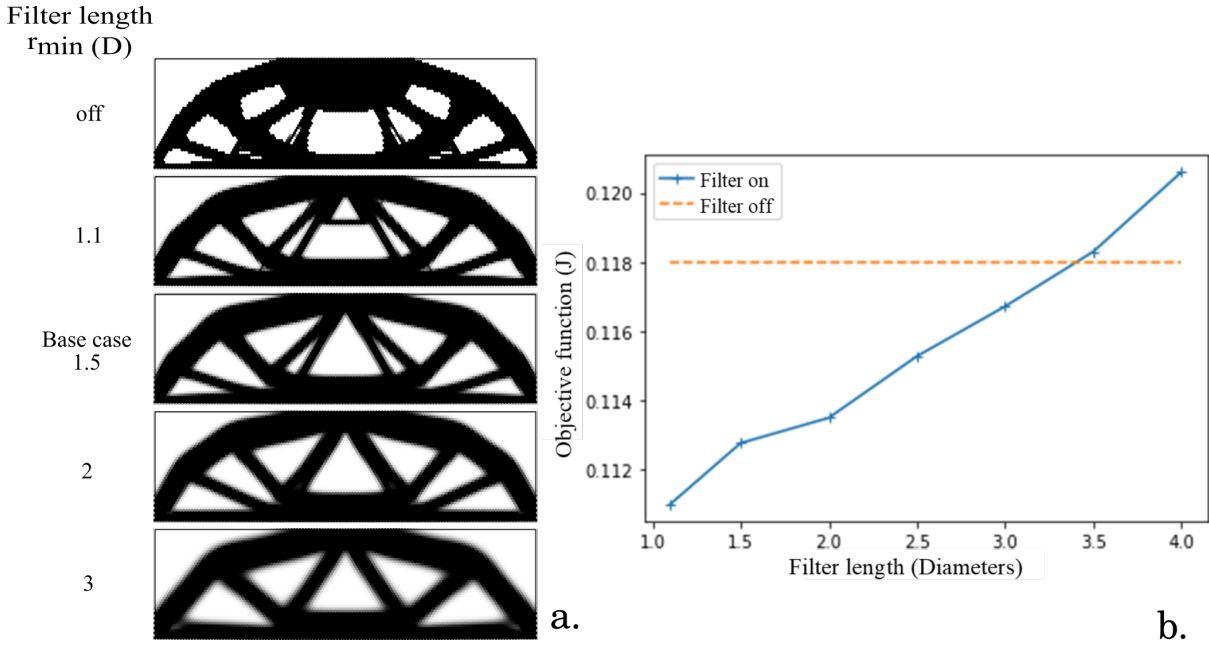


Figure 4.7 a) Effect of filtering length on optimized geometries. For generality, the values of r_{min} are given in units of particle diameters D . The label *off* indicates the unfiltered case. b) Effect of filter length on objective function

higher values of c : cf. $c = 0.132$ J for the structure with $f = 0.5$ in Fig. 4.6 with $c = 0.114$ J for the structure with $r_{min} = 2D$ in Fig. 4.7.

The top image in Fig. 4.7 shows a case without filtering. Expectedly, the resulting topology is the most complex compared to the other cases with filtering on. However, less intuitively, the resulting c is higher than in most filtered cases. As mentioned in Chapter 3, a lack of filtering can cause fast convergence to a local minimum of c which can be seen from Fig. 4.7. Filtering tends to smoothen out and remove local minima, thus leading to more optimal solutions. The top structure in Fig. 4.7, obtained without filtering, shows a few very thin elements but no checkerboard effect. an alternating pattern of individual particles with $\chi_i = 0$ and 1. This is because the locking problem leading to the checkerboard effect is specific to FEM-based analyses (Díaz and Sigmund, 1995).

When targeting a specific level of topological complexity for the purpose of fabrication or limiting complexity it may be favourable to select a desired filter length. However since mesh dependency of outputs may have physical meaning when dealing with granular problems it may sometimes be beneficial to run simulations without the application of filtering. Although in this case optimisation is shown to produce significantly less optimal solutions than filtered results. Once a particle mass is assigned zero the sensitivity of that particle becomes zero, as well and no more material can be redistributed into that particle creating a hard-kill scenario. In the

unfiltered case, no corrections are made for this, forcing the procedure into sub-optimal results. Deactivated particles can only gain mass through filtering and proximity to other particles.

4.2.3. Mesh resolution

In FE analyses, the size of the elements discretizing the continuum is in principle arbitrary. Therefore, when performing FEM-based TO one must monitor the impact of mesh resolution on the results, as in some cases the problem might display nonuniqueness and even nonexistence of the solution (Sigmund and Petersson, 1998). By contrast the DEM, in its basic formulation, does not feature a mesh at all, as particles represent physically distinct units. However, in practice, the particles in DE analyses are often coarse grained representations of richer underlying microstructures; for example one particle might summarise a collection of smaller grains. In other cases, like the simple beams in this chapter, the particles actually discretize a continuum. Therefore, also in the DEM there can be some arbitrariness in deciding the number and size of particles, which thus becomes analogous to deciding the mesh resolution in FE analyses.

To mimic the role of mesh resolution in FE analyses, the dimensions of the rectangular design domain from Fig. 4.1a are kept fixed whereas different numbers of particles $nelx$ and $nely$ initially filling the domain are explored. When solving problems with greater $nelx$ and $nely$ than the base case, the particle diameters D are reduced accordingly to always fill the same domain. When changing particle sizes in DE analyses, one should be careful that the intensity of the interaction may depend on D , as opposed to FE analyses where the constitutive parameters describing the material are intrinsically mesh-independent, e.g. the Young modulus E . Specifically for the system here, however, Eq. 4.1 shows that k_0 and E are simply linked by the thickness of the simulation domain in the z direction, t_z ; since the latter is always kept constant and equal to 1 mm, there is no need to change k_0 when changing D here. This is not always the case; for example, in a 3D simulation with spherical discrete elements, $k_0 \sim ED$ and therefore k_0 would be proportional to D .

Fig. 4.8 presents optimisation results for structures with a range of mesh resolutions around the base case. In all cases, a filtering length $r_{min} = 1.5$ mm is applied, as in the base case. The resulting geometries are generally insensitive to the mesh resolution, except for small differences such as an additional horizontal element appearing at low resolution. Fig. 4.8b shows that the

quality of the solution is very similar for all the structures in that they all feature a similar value of c at the end of the optimisation process. This result also proves that, for the 2D examples in this chapter, it is indeed correct to consider k_0 as independent of D .

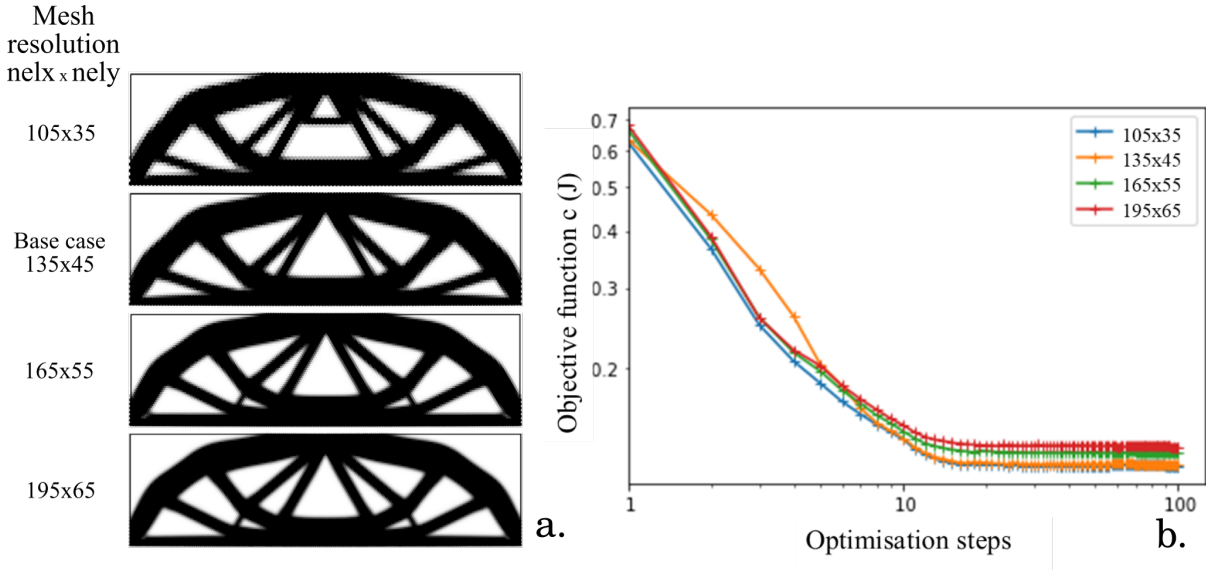


Figure 4.8 a) Effect of mesh resolution on optimum geometries: filtering included with length $r_{min} = 1.5$ mm in all cases. b) Effect of mesh resolution on the evolution of c during the optimisation process

Mesh-independent filtering, with fixed r_{min} irrespective of the mesh resolution, is known to enforce mesh independence also in mesh-sensitive FEM-based TO (Díaz and Sigmund, 1995). Results obtained without filtering for both DEM and FEM are shown in Fig. 4.9 considering the same structures as in Fig. 4.8 but without filtering. For the range of meshes analysed here, the results show an impact of mesh fineness on resulting topologies; this applies to both the FEM and DEM results. From this it is advisable that filtering be used in DETO to control minimum length scale in the same way it is used for FEM applications. A notable difference is that FEM is able to generate structural members at the scale of one element thickness, whereas DETO requires compression members to be triangulated trusses for stability this means the effect of mesh resolution is felt stronger in the FEM case. A more interesting result in Fig. 4.9 concerns instead the checkerboard problem, which clearly affects the FEM-TO solutions at any mesh whereas it is completely absent in all the solutions from DETO.

4.3. Geometric non-linearity under large displacements

Structures subjected to large displacements exhibit geometric nonlinearity which will impact the optimal topology. Accounting for this in FEM-based TO requires the additional complexity of

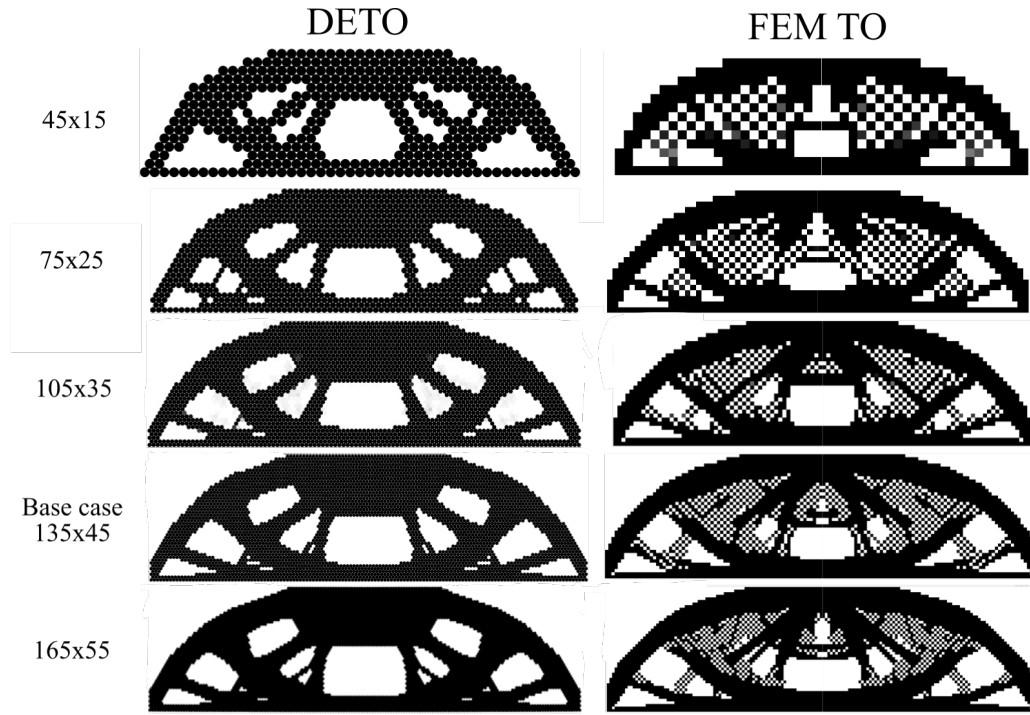


Figure 4.9 Study on mesh effects for the same beam structures as in Fig. 4.9, but here without imposing any filtering. Results from our DE Topology optimisation method are compared to analogous results from the FEM-based optimisation code in (Sigmund, 2001).

the adjoint method. By contrast geometric non-linearities are captured by DETO without any change to the theoretical framework.

Fig. 4.10 highlights the potential impact of geometric nonlinearity by considering two beam systems that are identical to the base case study in Figs. 4.1a,c, except that: (i) the supports are applied to the central axis instead of the bottom corners, and (ii) a larger point load of 10 kN is applied to the center of the beams instead of above or below them; this larger load has been chosen to induce larger displacements and thus better appreciate the effect of geometric nonlinearity (midspan deflection are now approximately 1.3% of the beam length and Fig. 4.11 shows that bond strains are as high as 1%). The only difference between conditions in Fig. 4.10a) and e) is that whilst a) is simply supported with two roller supports at each end, e) is pinned at each end of it's neutral axis. Simulations assuming small displacements should return identical solutions for both systems, because the roller supports act exactly the same as the pinned in this case. Indeed the results from linear elastic FEM-based TO in Fig. 4.10.b,f are identical. By contrast, the results from DETO in Fig. 4.10.c,g feature very different geometries for the two systems. The optimum solution for the simply-supported beam in Fig. 4.10.c is similar to the linear-elastic solution in Fig. 4.10.b. The reason is that the inward motion of the rollers

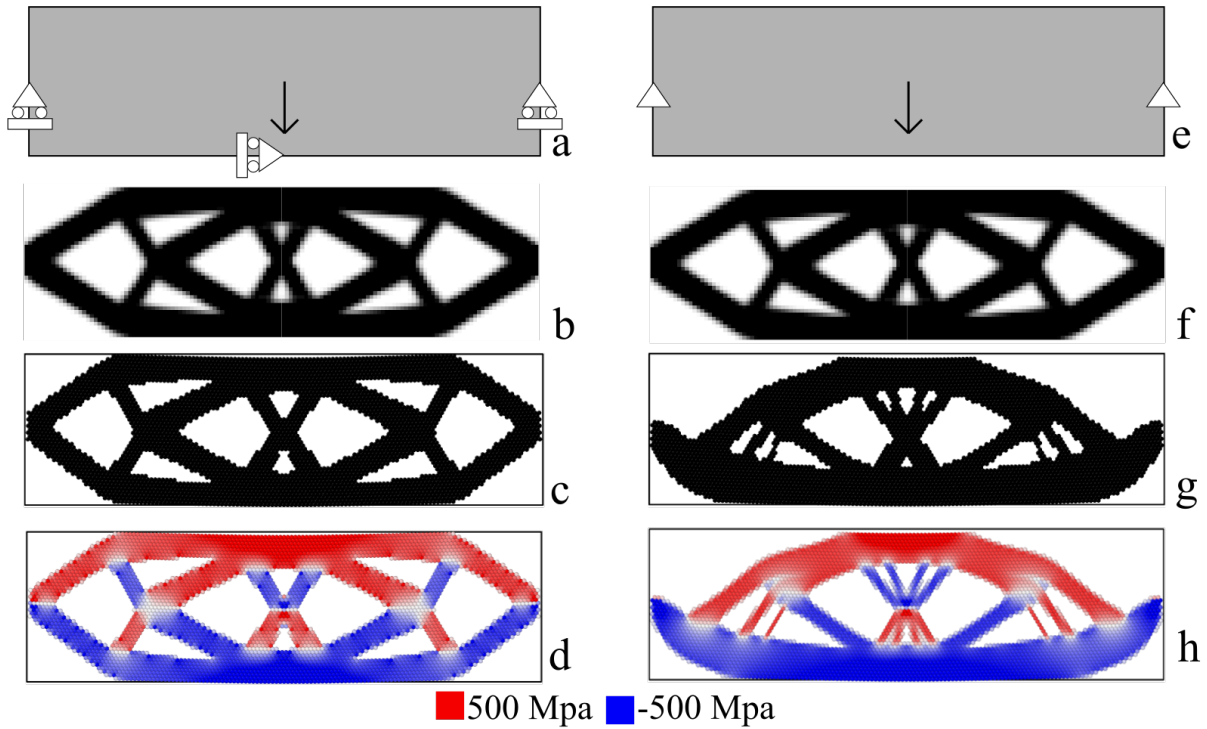


Figure 4.10 Solutions of topology optimisation problems highlighting the impact of geometric nonlinearity. a) Simply-supported and e) pinned systems, with supports and forces applied to the central axis of the beam; (b,f) solutions from linear elastic FEM-based TO using the code in (Sigmund, 2001), with inputs: $nex = 72$, $nely = 39$, $volfrac = 0.6$, $penal = 3$ and $rmin = 2$ (see (Sigmund, 2001) for details on the meaning of those inputs); (c,g) solutions from the DETO, which naturally accounts for geometric nonlinearity; (d,h) spatial distribution of hydrostatic stresses for the DETO solutions, identifying the elements working in tension (blue) and in compression (red).

allows the structure to behave in pure bending also when geometric nonlinearities are included. The distribution of hydrostatic stresses in Fig. 4.10.d shows indeed a symmetric distribution of elements working in tension and in compression. The qualitative difference in Fig. 4.10.g stems from the catenary action induced by the pinned supports. During the deflection, the central axis of the pinned system is stretched and this generates a tensile stress along the beam. During the optimisation process, this additional tensile stress drives material away from the compressed regions and towards the parts under tension. As a result, Fig. 4.10.h displays a thickening of the lower deck, which carries most of the catenary force, whereas the upper arch in compression becomes smaller and migrates towards the centre of the beam.

4.3.1. Full sensitivities computation with finite difference approach

Chapter 3 described how the sensitivity expression in Eq. 3.9 is an approximation of the gradient of the cost function with respect to the design variables under the assumption of small displacements and then offered a numerical approach based on a finite difference method of computing

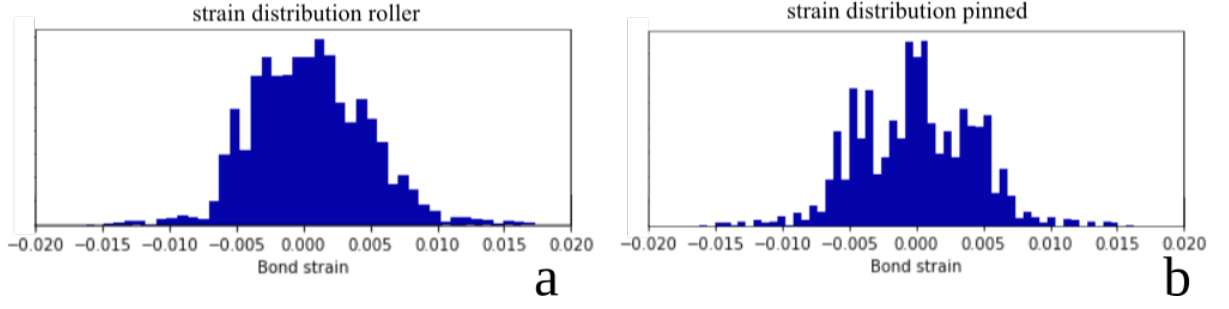


Figure 4.11 Distribution of bond strain in the optimized structures with pinned and roller supports.

the full gradient. Here results are presented using this numerical approach and compared with results obtained using the approximate sensitivity in Eq. 3.9 for examples exhibiting varying degrees of deflection.

The finite difference approach has a high computational cost, in that each term of the sensitivity vector requires one dedicated energy minimization to compute the perturbed U^* . Therefore here only small structures are considered, made of 45×15 particles. The geometry, supports, and loading conditions are the same as for the double-pinned, central force beams in Fig. 4.11e. The systems are loaded with three force intensities, 0.2, 2, and 20 kN, to trigger different levels of geometric nonlinearity. Figs. 4.12, 4.13, and 4.14 show the results for these case studies, including results from both the numerical perturbation method (left) and the approximate sensitivities (right). In these examples, the tolerance for DEM convergence has been set to 10^{-8} , the filtering length to 1.1 diameters, and the perturbation $\Delta\chi$ to 0.01.

All results in Figs. 4.12, 4.13, and 4.14 indicate that the two methods to compute sensitivities give results that are extremely similar, both in small and large deformation regimes (imposed by applying a progressively larger external load from one figure to the next). The figures also feature histograms showing the distribution of normalised per-particle i differences between the sensitivities obtained with the two methods:

$$E_i = \frac{\left| \left(\frac{dc}{d\chi_i} \right)_f - \left(\frac{\partial c}{\partial \chi_i} \right)_p \right|}{\frac{1}{N} \sum_i \left| \frac{dc}{d\chi_i} \right|_f} \quad (4.2)$$

N is the number of particles in the domain. Subscripts f and p indicate *full* sensitivities computed using the numerical perturbation method, and *partial* sensitivities obtained with the approximation in Eq. 3.9. Both the difference in the numerator and the contribution to the average in the denominator are taken as absolute values.

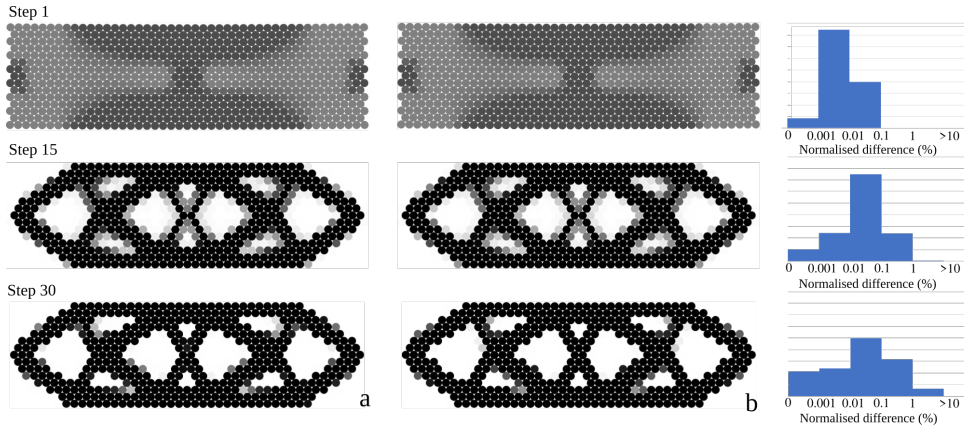


Figure 4.12 optimisation of beam made of 45×15 particles, simply supported and loaded as shown in Fig. 4.10a. A load of 0.2 kN is applied, which leads to small deformations and an overall symmetric solution. Optimisation snapshots at steps 1, 15, and 30 are shown, for full sensitivities computed with the numerical perturbation method in a) and for the approximate sensitivity in Eq. 3.9 in b). Although the difference between a) and b) are too marginal to be represented visually for the same steps, the histograms quantify the difference between per-particle sensitivities obtained with the two methods (Eq. 4.2) are shown next to the diagrams confirming that indeed the percentage difference is indeed very minor.

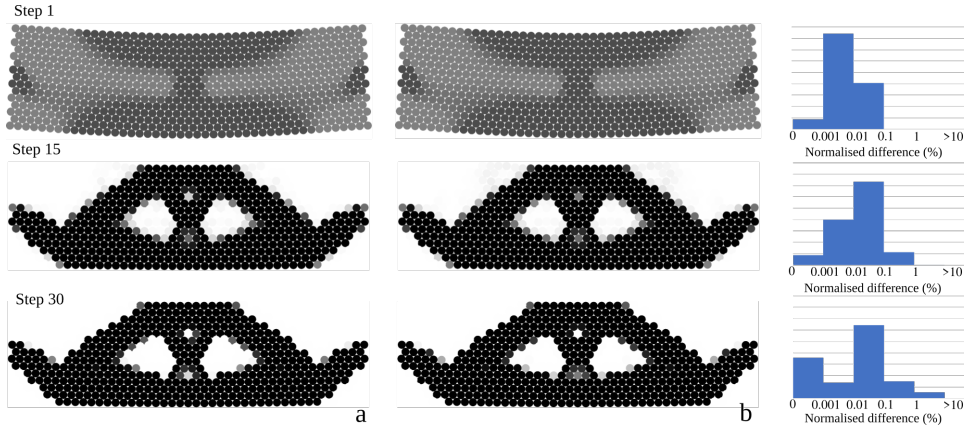


Figure 4.13 optimisation of beam made of 45×15 particles, simply supported and loaded as shown in Fig. 4.10a. Here a load of 2 kN is applied, which is sufficient to highlight the effect of large deformations. Sensitivities computed with the numerical perturbation method in a) and with the approximate sensitivity method in Eq. 3.9 in b).

Fig. 4.15 is a representation of the structure colour coded to show the extent of error between full and partial sensitivities on a per particle basis. This shows that the particles with highest differences in sensitivity are those with intermediate χ between 0 and 1, mostly located at the boundaries between solid. Particles in the solid also display a certain level of difference in sensitivity, whereas void particles tend to zero sensitivity in both approaches as the optimisation process converges. Consistently, the histograms in Figs. 4.12–4.14 starting from a single peak distribution at step 1 move towards bimodal distributions as the optimisation progresses. In particular, a peak for differences tending to zero is formed and grows: this reflects the increase in void-like particles. The other peak in the histograms instead increases by approximately one

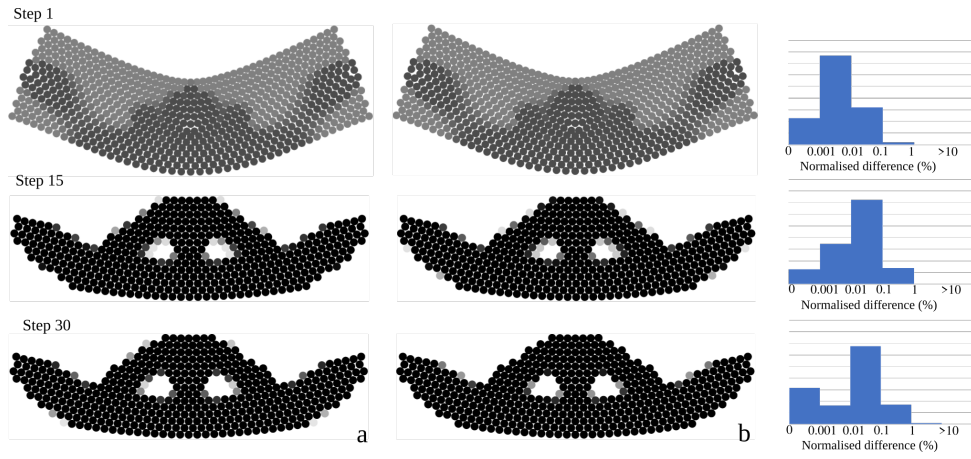


Figure 4.14 optimisation of 45×15 beam, simply supported and loaded as shown in Fig. 4.10a. Here a high load of 20 kN is applied, which causes visibly large deformations. Sensitivities computed with the numerical perturbation method in a) and with the approximate sensitivity method in Eq. 3.9 in b).

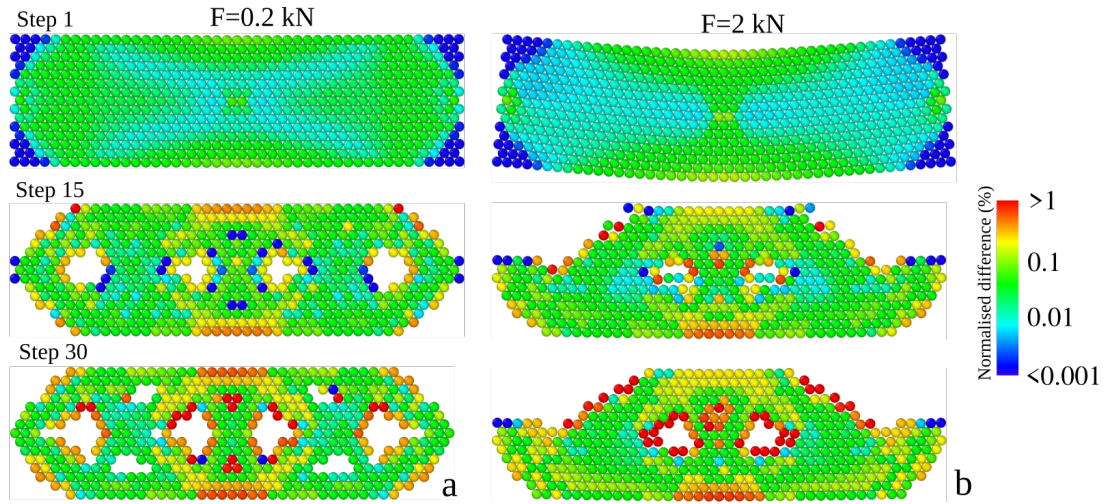


Figure 4.15 Spatial distribution of normalised difference in sensitivity for the structures in Figs. 4.12 & 4.13 showing the transition from an initially even spread of difference to concentrated error values in the solid-void border as the optimisation progresses.

order of magnitude, from differences of 0.001 – 0.01% to 0.01 – 0.1%, during the first 15 steps of the optimisation. This is when the structure gains most of its stiffness, thus reducing significantly the complementary energy and therefore the average magnitude of sensitivities in the structure, as shown in Fig. 4.16a. As a result, the denominator in Eq. 4.2 decreases significantly during the first 15 optimisation steps, and this shifts the normalised values in the histograms up. Comparing the histograms at step 15 with those at step 30, one can notice tail forming, with particles that feature high normalised differences greater than 1%. These are the particles concentrating into a progressively thinner interface between solid and void. In any case, the histograms in the snapshots in Figs. 4.12–4.14 show that the differences between full and partial sensitivities are very small: fractions of percent. This is consistent with the fact that the optimum solutions,

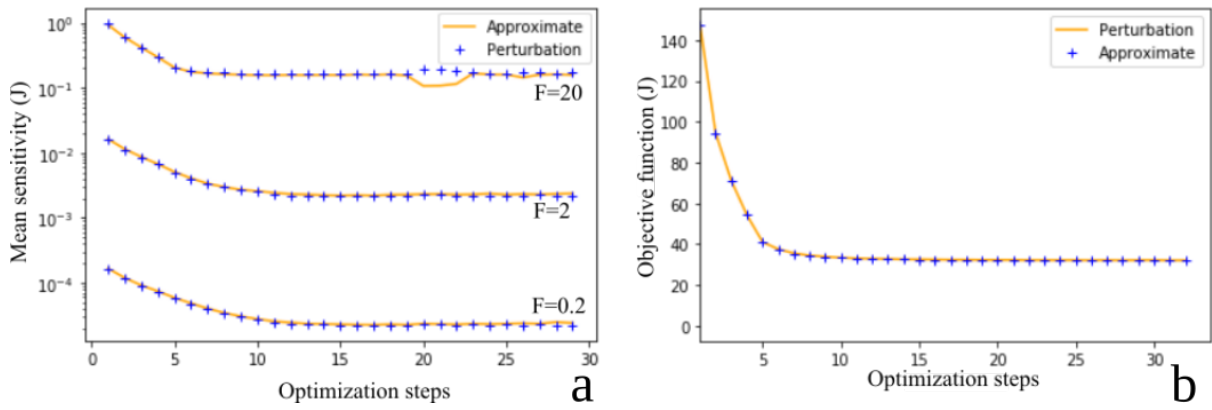


Figure 4.16 a) Approximate and full mean sensitivity values during the first 30 steps of the optimisation showing a fast decrease in all three cases. b) Evolution of objective function (complementary work) for the problem in Fig. 4.14 (with 20 kN of applied load) solved using the full perturbation method and the approximate sensitivity in Eq. 3.9.

shown in the snapshots of the same figures, are nearly identical for the two methods of computing sensitivities.

The approximated sensitivity in Eq. 3.9 is actually exact in the limit of small strain, for linear elastic structures. By contrast, the cases with highest geometric nonlinearity are those where the differences between full and partial sensitivities might be most significant. However, the histograms in Figs. 4.12–4.14 indicate that the levels of nonlinearity explored here, while leading to qualitatively different structures, still imply a similar distribution of differences between full and partial sensitivities, i.e. the approximation in Eq. 3.9 seems quite robust to the geometric nonlinearity sampled here. The good quality of the approximation is indeed confirmed in Fig. 4.16b, which shows the evolution of the objective function, i.e. the complementary work, during the optimisation of the structure with highest load and thus highest geometric nonlinearity, from Fig. 4.14. The figure clearly indicates that, for the problems presented here, the approximate sensitivity from Eq. 3.9 produces an optimisation process that is quantitatively very similar to that obtained using the full sensitivity from the numerical perturbation method presented here. However greater non-linearities do lead to higher risk of accumulating excessive error and therefore use of the partial derivative method should be handled on a cases by cases case basis applying the full method to determine the extent of error present.

4.4. Material non-linearity

This section presents three optimisation problems where material nonlinearity may significant impact the resulting topologies. Results are first obtained for the *Lin* and *Weak* interactions in Table. 4.3, which are analogous to the linear elastic and elastoplastic materials considered in (Maute et al., 1998a). Two of the systems are also tested using the asymmetric potentials *Weak-C* and *Weak-T* in Table. 4.3.

4.4.1. Interaction potentials

Four types of interaction potentials are considered here, each representing a different material behavior. This Chapter uses only first-neighbour interactions, but inclusion of longer-range nonlocal interactions would not require any change to the methodology presented here. Table. 4.3 shows the expressions of each potential, along with the corresponding interaction forces (positive when repulsive).

Potential name	$U_{ij}(r_{ij})$	$F_{ij}(r_{ij}) = -\frac{dU_{ij}}{dr_{ij}}$
<i>Lin</i>	$\frac{1}{2}k_0(r_{ij} - D)^2$	$-k_0(r_{ij} - D)$
<i>Weak</i>	$\frac{k_0}{a^2} \ln\{\cosh[a \cdot (r_{ij} - D)]\}$	$-\frac{k_0}{a} \{\tanh[a \cdot (r_{ij} - D)]\}$
<i>Weak-T</i>	$\frac{k_0}{a} \left\{ \frac{1}{a} \exp[-a \cdot (r_{ij} - D)] + (r_{ij} - D) \right\} - \frac{k_0}{a^2}$	$-\frac{k_0}{a} \{1 - \exp[-a \cdot (r_{ij} - D)]\}$
<i>Weak-C</i>	$\frac{k_0}{a} \left\{ \frac{1}{a} \exp[a \cdot (r_{ij} - D)] - (r_{ij} - D) \right\} - \frac{k_0}{a^2}$	$-\frac{k_0}{a} \{\exp[a \cdot (r_{ij} - D)] - 1\}$

Table 4.3 Interaction potentials for the case of linear elastic materials (*Lin*), symmetric strain-hardening material in tension and in compression (*Weak*), asymmetric material hardening in tension and stiffening in compression (*Weak-T*), and asymmetric material hardening in compression and stiffening in tension (*Weak-C*).

One can immediately notice how the interaction energy and force do not diverge in the $r_{ij} \rightarrow 0^+$ limit; potentials that are commonly used in microstructural simulations, such as the Lennard-Jones potential, feature instead diverging energy and force in such limit. However, the interactions proposed here are meant for macroscopic systems experiencing strain levels limited to few percent. For such systems, typical interactions used in Discrete Element simulations do not diverge in the $r_{ij} \rightarrow 0^+$, *e.g.* Hertz contact forces or Hookean bonds.

Fig. 4.17.a compares the $U_{ij}(r_{ij})$ for the various materials and for a set of k_0 , a , and D parameters. As expected from strain energies, all the U_{ij} curves are zero in the undeformed state

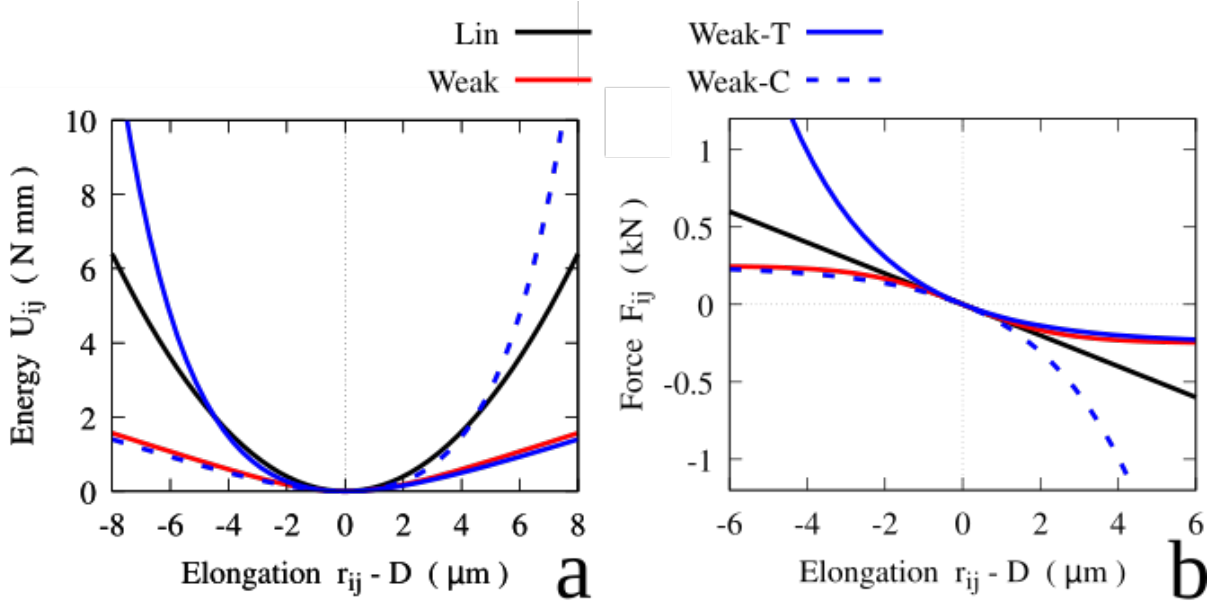


Figure 4.17 a) Interaction potentials from Table. 4.3 for some of the parameters used: $k_0 = 100$ kN, $D = 1$ mm, $a = 400$ mm⁻¹; b) Corresponding force-elongation curves from Table. 4.3, which are proportional to the stress-strain behaviors of the materials.

$r_{ij} = D$, and positive elsewhere. Fig. 4.17.b shows the $F_{ij}(r_{ij})$ curves, from which the strain-hardening and strain-stiffening regimes can be appreciated. The $F_{ij}(r_{ij})$ curves are proportional to the stress-strain behavior of the material, which can be quantitatively estimated assuming that $D = 1$ mm, that the box thickness in the third direction is $t_z = D$, and that the contact area between two particles is one sixth of the lateral surface area of the disk, $\frac{1}{6}\pi Dt_z$. Under these assumptions, the strain between particles in 10^{-3} units is equal to the elongation in μm in Fig. 4.17, whereas the maximum stress between particles in strain-hardening regimes, when $|r_{ij} - D| \gg 0$, is capped to $\frac{k_0}{a} \frac{6}{\pi D^2} = 477$ MPa (assuming $k_0 = 100$ kN/mm and $a = 400$ mm⁻¹ as in Fig. 4.17); simulation results will later confirm this estimation. Fig. 4.17 also shows how, for the materials proposed here, nonlinearity become important at approximately 0.1 – 0.2% strain: this is representative of various metals at the macroscale, for example steel.

The interactions in Fig. 4.17 capture material nonlinearity under strain. However, the potentials are all elastic, with same stress-strain responses upon loading and unloading. Irreversible deformations could be considered in principle, and indeed elastoplasticity and elasto-plastic interactions are within the current capabilities of continuum-based TO and of DE analyses (*e.g.* (Magnier and Donzé, 1998; Masoero et al., 2010)). Such irreversibilities would impact the results if the DE analyses involved dynamic or cyclic loads, or if buckling instabilities or strain localization, *e.g.* due to material softening or fracture, led to stress relaxation in some parts of the

structure. Material softening and fracture are not considered in the chapter (only hardening and stiffening as per Fig. 4.17), buckling will not be considered, and imposed loads or displacements will always induce monotonically increasing strain everywhere in the structure. Under these conditions, the reversible interactions in Fig. 4.17 are as representative of large-strain material behaviors as elastoplastic interactions would be. Therefore the material is not strictly plastic, because its behavior under quasi-statically and monotonically increasing strain is the same.

4.4.2. Three-support system with imposed displacement from the top

The system is shown in Fig. 4.18; it is analogous to one originally analysed in (Maute et al., 1998a) using FEM-based TO. Fig. 4.19.a shows the topology resulting from DETO when the

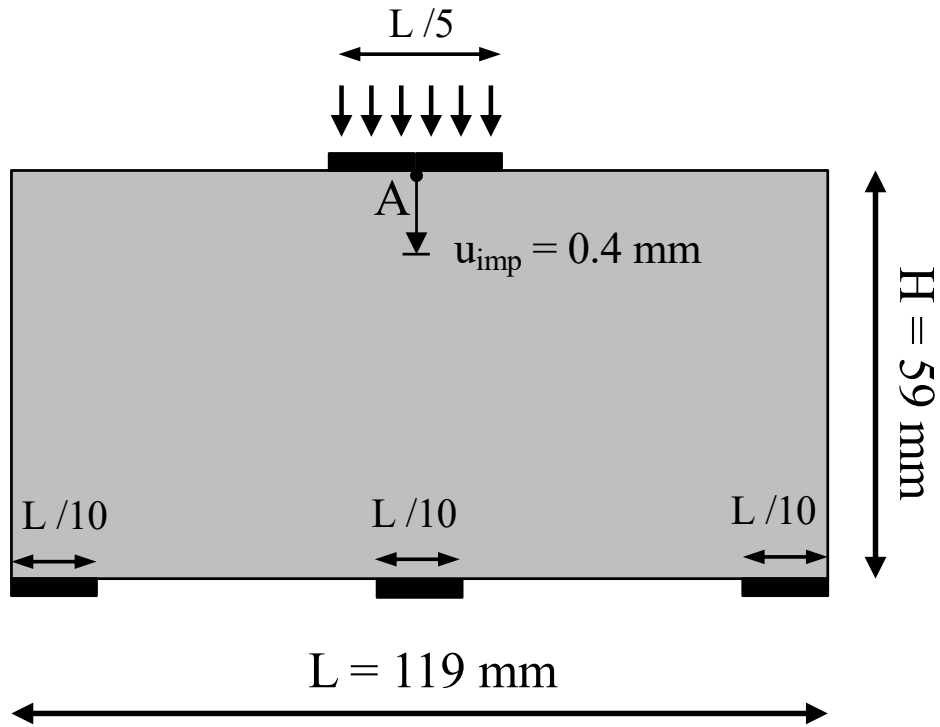


Figure 4.18 optimisation problem for a beam on three-supports with imposed displacement at point A at midspan. The value of u_{imp} has been fine-tuned to obtain an appreciable impact of material nonlinearity.

material is linear elastic. Most of the structure gets concentrated into a central pillar, which provides the shortest and stiffest path to transfer the load from the point A down to the central support. The benefit of increasing the cross section of the pillar is limited by the size of the support, to the extent that for the target solid fraction used here, $f = 0.3$, additional stiffness is gained by creating diagonal branches that reach for the lateral support, despite such branches are longer than the central pillar and thus contribute less efficiently to the overall stiffness. A

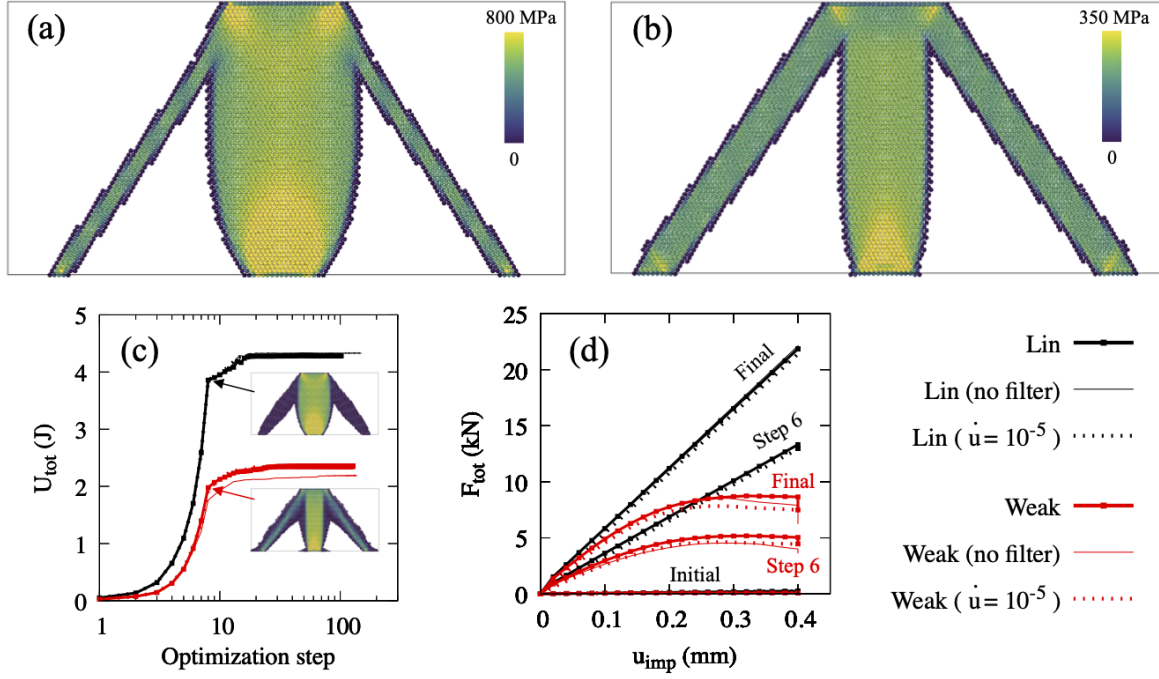


Figure 4.19 optimisation results for the three-support system in Fig. 4.18 with target solid fraction $f = 0.3$, assuming a) linear elastic and b) symmetrically strain-hardening material, as per *Lin* and *Weak* expressions in Table. 4.3. The colors represent the intensity of local von Mises stress; c) Evolution of the objective function U_{tot} , the total strain energy of the systems during the optimisation. The base case with inputs in Table. 4.1 is compared with cases with no filtering and with smaller $\dot{u}_{imp}$; d) Evolution of force–displacement curves during the optimisation.

similar result was obtained in (Maute et al., 1998a) using FEM-based TO; in that work, however, the optimum structure did not feature the diagonal branches. This difference may be due to the difference between an FE-based description and our DE-based one. Another possible explanation lies in different optimisation procedures, *e.g.* the different updating schemes for χ or parameters such as the maximum change of χ_i allowed between subsequent optimisation steps. We found that the lateral branches appear also when imposing much smaller displacements, which excludes that they result from geometric nonlinearity and the fact that DE analyses compute forces in the deformed configuration. In any case, additional simulations not presented here have shown that the overall stiffness changes only very slightly when the mass is all concentrated into the central pillar, rather than being partly distributed to the thin diagonal branches in Fig. 4.19.a.

Fig. 4.19.b shows the optimisation result for the symmetrically nonlinear material. The limiting factor for U_{tot} in this case is that some pairs of particles may reach the maximum asymptotic value of their interaction force (see Fig. 4.17), thus entering into the analogous of a plastic flow regime. This happens near the supports and under the plate applying the imposed displacement, as shown by the sharp diagonal fronts of large whereas stress in Fig. 4.19b.

Because of these mechanisms, the response of the system is controlled by the thinnest cross section across which the load is transferred. Therefore, if the thick central pillar in Fig. 4.19.a was retained, all its mass in excess to its smallest cross section, the size of the support below it, would not contribute to the maximum U_{tot} . Therefore, when the material nonlinearity is considered in the optimisation process, the excess mass is removed from the central pillar and used to thicken the diagonal branches, exploiting as much additional area from the lateral supports as possible. The result is analogous to that obtained in (Maute et al., 1998a).

Fig. 4.19.c shows the evolution of the objective function, U_{tot} , during the optimisation process. As expected, the weaker nonlinear material ends up with significantly lower U_{tot} . The snapshots within the figure show how the systems in Figs. 4.19.a and b appear after 8 optimisation steps only. Both systems then feature thick diagonal branches, but with the key difference that the *Weak* system is already clearly utilizing the branches (light color meaning intense von Mises stresses in them), whereas the *Lin* system is not utilizing them significantly (dark color meaning little stress). As a result, at this step during the optimisation mass tends to move away from the branches in the *Lin* case, whereas it tends to move towards the branches in the *Weak* case.

Fig. 4.19.d shows the force–displacement curves for the *Lin* and *Weak* materials, evolving during the optimisation process. Clearly the final solutions are much stronger than the initial ones, where all particles had $\chi_i = f = 0.3$. At small displacements the two systems feature similar stiffness, whereas the nonlinearity caused by the material in the *Weak* system becomes evident at larger u_{imp} .

For both types of material, Fig. 4.19.c and d compare results for three different cases: the base case with input data in Table. 4.1, the base case but without filtering, and the base case but with a smaller loading rate $\dot{u}_{imp} = 10^{-5}$ (instead of $10^{-4} \text{ mm } \mu\text{s}^{-1}$ in the base case). For the linear material all cases give identical result. For the nonlinear material, instead, the case without filtering reaches a less optimal solution with lower U_{tot} , whereas the other two cases returns the same evolution of U_{tot} . A close scrutiny of the force–displacement curves for the *Weak* system indicates that the curves for the base case are the highest, suggesting a more optimum outcome. However, when reaching the target u_{imp} , the base systems continues to minimize its strain energy which causes a drop of force while u_{imp} remains fixed at 0.4 mm. The case with lower loading rate features a lower curve but with no further relaxation at $u_{imp} = 0.4 \text{ mm}$. As a result, both the

base case and the slower one attain the same final value of force, and thus of U_{tot} , at $u_{imp} = 0.4$ mm; this explains why their U_{tot} are identical in Fig. 4.19.c. By contrast, the force–displacement curves for the case without filtering are intermediate between the base and slower cases but, when $u_{imp} = 0.4$ mm is reached, a significant further relaxation sees the force dropping below those of the other cases (see the thin vertical lines at $u_{imp} = 0.4$ in Fig. 4.19.d). This explains why U_{tot} in Fig. 4.19.c is smaller in this case than for the others. In terms of geometry evolution, what limits the unfiltered case is that the system rapidly converges to a configuration with all $\chi_i \approx 0$ or 1, getting effectively stuck into a local energy minimum. This hinders a full transfer of mass towards the lateral branches, hence a full exploitation of the supports.

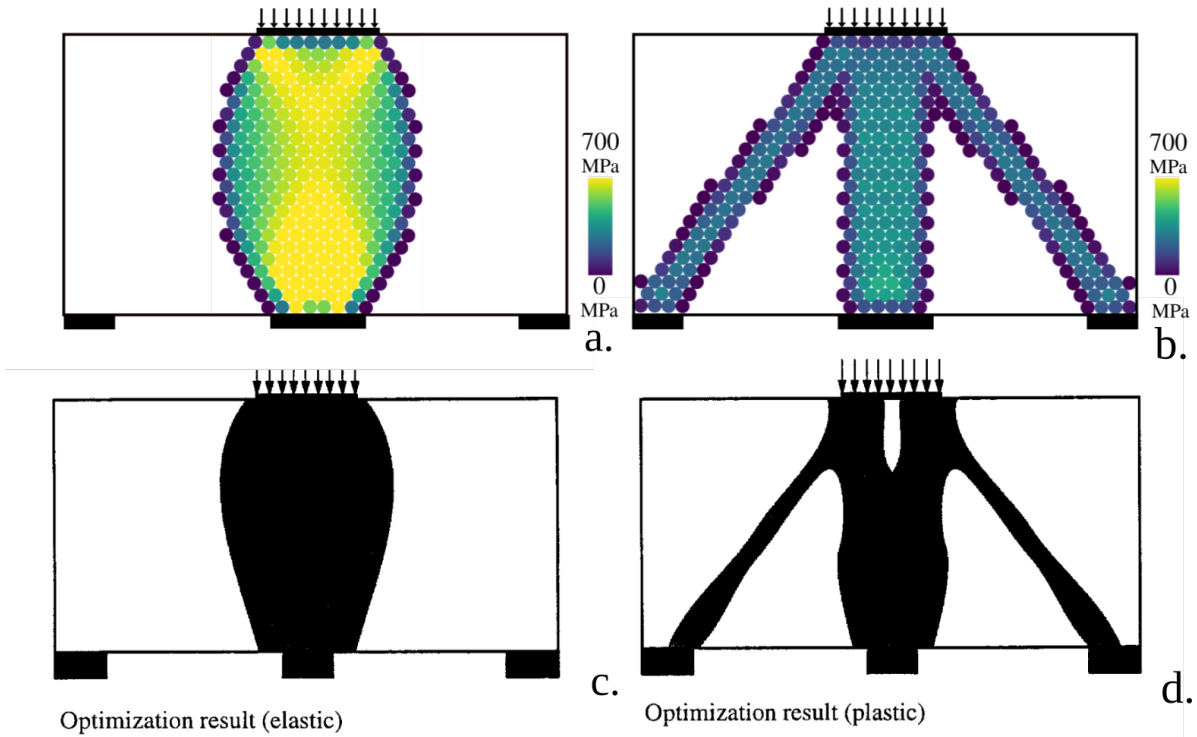


Figure 4.20 Comparison between DETO results a),b) for the same set up as Fig. 4.19 (except on a 30×15 element system with a particle diameter of 1) and the original results for the non-linear FEM structure c), d) published in Maute et al. (1998a).

The direct comparison of the results shown in Fig. 4.20 show the same formation of structure in both cases with the example from Maute et al. (1998a) also showing the tendency for plastic material behaviour to lead to the construction of lateral supports. These similarities support and validate the inclusion of material non-linearity in DETO. The slight differences in form are likely here due to the difference in material discretisation between DEM and FEM approximations.

Fig. 4.21 shows results that are particularly relevant for structural design. The *Lin from Weak* series explores how the structure in Fig. 4.19.b, optimized for a nonlinear material (for best

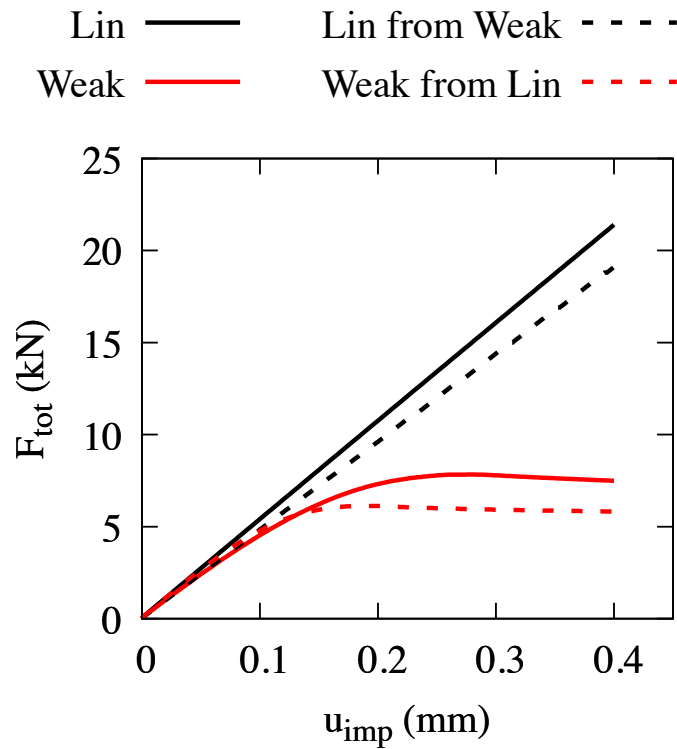


Figure 4.21 Force–displacement curves for the configuration in Fig. 4.19.b assuming linear elastic material (*Lin from Weak*), and for the configuration in Fig. 4.19.a assuming nonlinear material (*Weak from Lin*). The curves are compared with the base cases for linear and nonlinear materials already shown in Fig. 4.19.d (solid curves). All curves here were obtained using loading rate $\dot{u}_{imp} = 10^{-5} \text{ mm } \mu\text{s}^{-1}$.

performance approaching failure), behaves in the linear elastic range. The results show that the stiffness of the structure is lower than that in the *Lin* structure, which was originally optimized assuming a linear elastic material. The loss in stiffness is 11%, from a gradient of 53.5 kN/mm in the *Lin* case to 47.75 kN/mm in the *Lin from Weak* case. In the same figure, the *Weak from Lin* series explores how the structure in Fig. 4.19.a, optimized for a linear material (for maximum stiffness in service conditions) behaves when approaching failure. The results show that the maximum force and the strain energy at $u_{imp} = 0.4 \text{ mm}$ are both substantially smaller than in the *Weak* structure, which was originally optimized assuming nonlinear material. The maximum force and strain energy go from 7.83 kN and 2.34 kN mm for the *Weak* case, to 6.12 kN and 2.11 kN mm for the *Weak from Lin* case, decreasing by 22% and 10% respectively. A 15% loss in maximum force was obtained in (Maute et al., 1998a) for a system with same geometry, but using FEM-based TO and elastoplastic material. An 11% loss of stiffness in service conditions is likely to be less problematic than a 22% loss of strength approaching failure. Therefore, the designer should use TO with linear elastic materials carefully and favour optimisation using realistic material behaviors when addressing strength and structural failure.

4.4.3. Three-support system with mid support settlement

The system in Fig. 4.22 has very similar geometry as the previous one in Fig. 4.18. The differences are that the displacement is imposed at the mid support instead of above the beam, and that the lateral supports are only half as wide as before. This problem was also originally addressed in (Maute et al., 1998a), there using FEM-based TO with elastoplastic material.

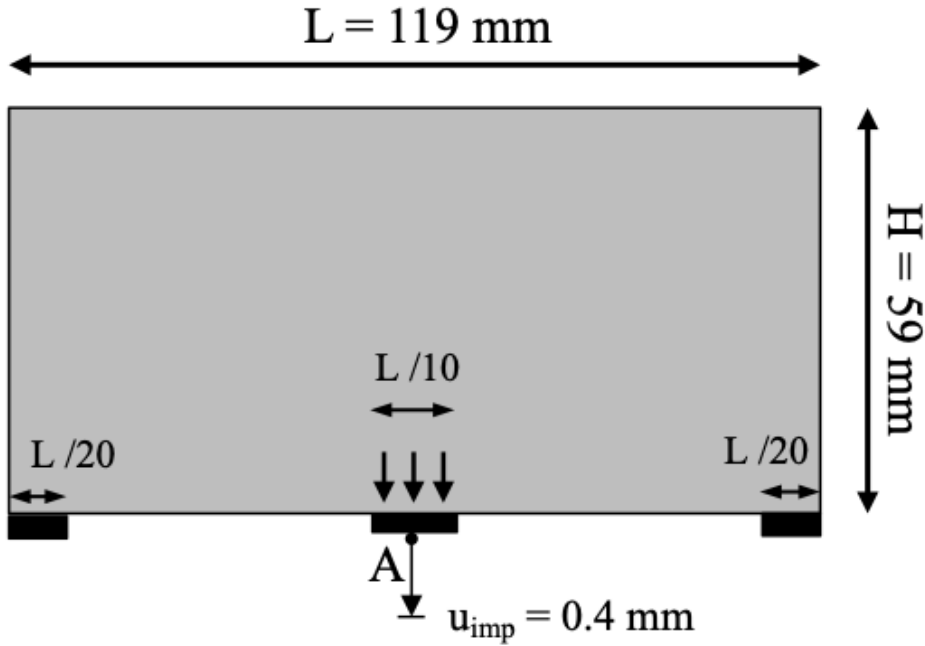


Figure 4.22 optimisation problem for a beam on three supports with imposed settlement of the mid support.

Fig. 4.23 shows the optimum geometries obtained from DETO. In all cases, the resisting mechanism is akin to that in the seminal work of Michell (Michell, 1904), where the central ties connect the settling plate to the compressed arch above, which transfers the load to the stable lateral supports. The linear elastic *Lin* case produces a structure that is very similar to that in (Maute et al., 1998a), despite the already mentioned methodological differences. A material that is strain-hardening both in tension and in compression produces the *Weak* structure in Fig. 4.23.b, with a flattening of the arch at its top and with fewer thicker ties linking the settling mid support with the compressed arch. Another important detail is that the *Weak* structure concentrates more mass near the lateral supports, which are instead not fully utilized in the *Lin* case. An analogous tendency to fully exploit the supports has been already discussed in the previous section, and was also observed in (Maute et al., 1998a) for this case study.

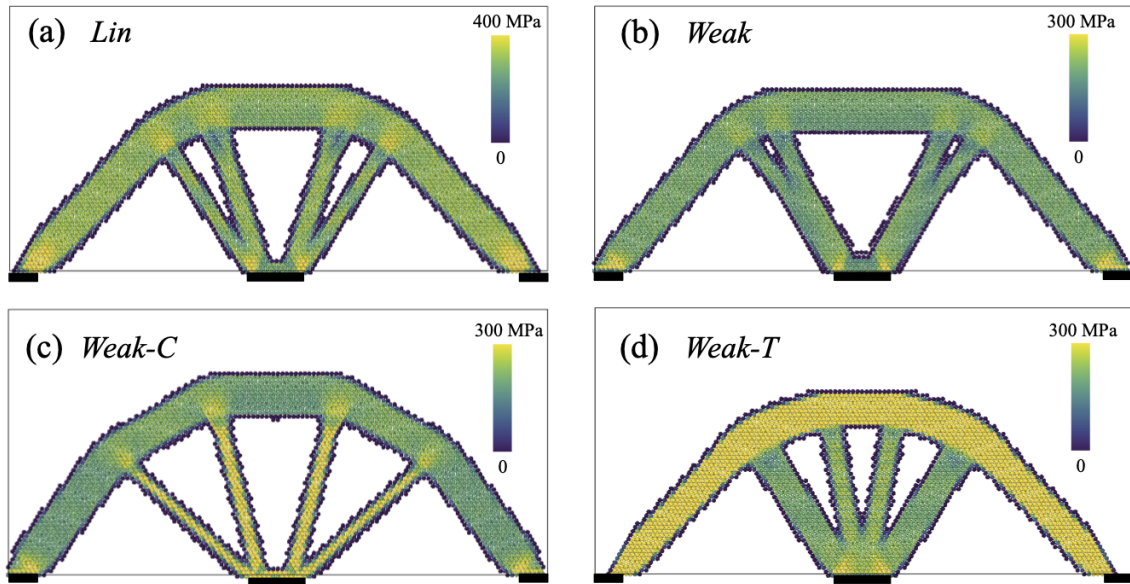


Figure 4.23 Optimum geometries for the problem in Fig. 4.22, with solid fraction $f = 0.25$ and for different material behaviors as per Table. 4.3: a) linear elastic, b) symmetrically strain-hardening, c) strain-hardening in compression and strain-stiffening in tension, and d) strain-hardening in tension and stiffening in compression. The colors represent the intensity of the deviatoric von Mises stress.

The structures in Fig. 4.19, in the previous section, were fully under compression when loaded, hence considering asymmetric materials in tension and compression was not useful then. Here instead, Fig. 4.23 shows how asymmetric material behaviors lead to different optimum structures. In particular, Fig. 4.23.c shows that a material that is weak, strain-hardening, in compression and strong, strain-stiffening, in tension produces a structure with thin central ties under high stress, and a thicker compressed arch that fully utilizes the lateral supports. By contrast, in Fig. 4.23.d, a material that is weak in tension and strong in compression creates thick central ties and a shallower and thinner compressed arch which utilizes the lateral supports only in part. In this latter case, the limiting factor is the size of the settling central support, which controls the maximum cross section in tension and thus the maximum force that the structure can carry.

Fig. 4.24 shows the force-displacement curves for the four systems in Fig. 4.23. As expected, all curves start with the same gradient in the initial linear regime. The *Weak-T* system displays an initially increasing gradient, due to strain-stiffening in the compressed arch. At displacements over 0.5 mm, however, the strain-hardening behavior of the central ties takes over and plastic flow caps the maximum force. The *Weak* system features the smallest strength, but the *Weak-C* is only marginally better, as opposed to the significantly stronger *Weak-T* system. This happens because the strength-controlling element in the *Weak* and *Weak-C* system is the compressed arch.

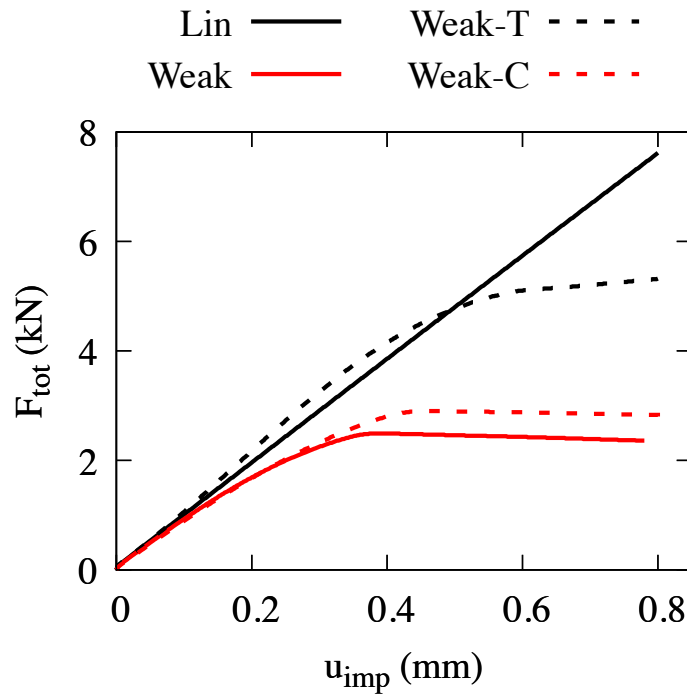


Figure 4.24 Force-displacement curves for the structures in Fig. 4.23, each with their respective material behavior.

The *Weak-C* system can transfer a bit more mass from the central ties into the arch, but eventually the minimum cross sectional area of the arch is limited by the size of the lateral supports, which both the *Weak* and *Weak-C* systems utilize in full or almost. In the *Weak-T* system, instead, strength is controlled by the central ties and therefore the system has more freedom to move mass away from the compressed arch and alter the overall geometry to maximize its strain energy.

The different optimum solutions in Fig. 4.23 raise the question of how much an incorrect assumption of material behavior in the TO process may affect the structural performance. As an example, consider a structure where the elements under compression are confined using fiber reinforced polymer (FRP) to induce strain-stiffening in a material that would otherwise be symmetrically strain-hardening. In our model, this means turning a *Weak* system into a *Weak-T* one. If optimized assuming *Weak-T* behavior, the geometry in Fig. 4.23.d would be obtained. However, if the FRP system failed in the actual structure, the material behavior would go back to *Weak*, for which the optimum geometry would be that in Fig. 4.23.b instead. This raises two questions: how much strength loss may be caused by an incorrect assumption of material behavior? Which of the four material behaviors considered here would produce the most robust structure, in case the material ends up behaving differently? The results in Fig. 4.25 address these questions.

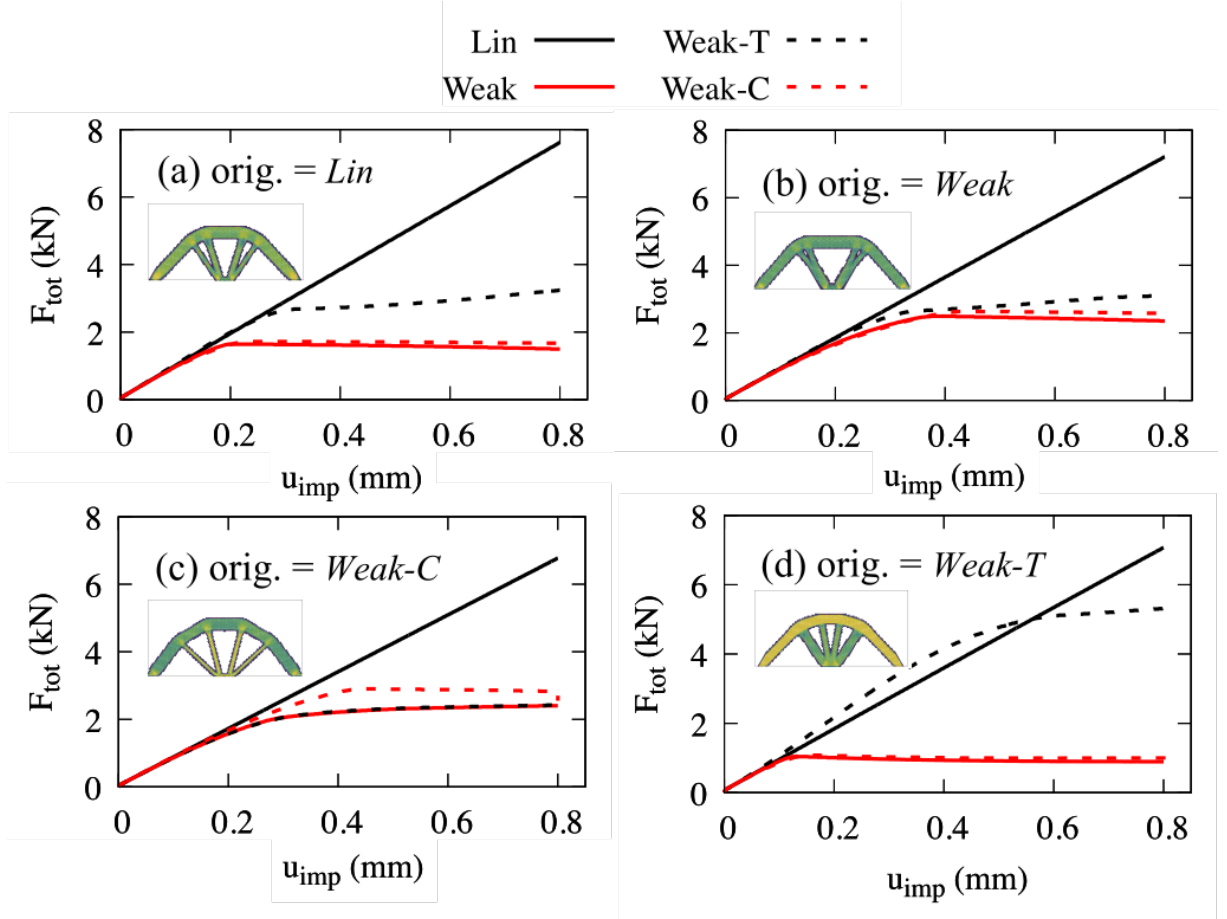


Figure 4.25 Force-displacement curves for different material behaviors and for structures originally optimized assuming the following material types: a) linear elastic *Lin*, b) symmetrically strain-hardening *Weak*, c) strain-hardening in compression and stiffening in tension *Weak-C*, and d) strain-hardening in tension and stiffening in compression *Weak-T*. The snapshots of the optimized structures are identical to those in Fig. 4.23.

Each subfigure in Fig. 4.25 shows how one of the optimized structures in Fig. 4.23 would behave for any of the four material types. A first take is that all four structures, irrespective of the material assumption underlying them, feature a similar force-displacement curve when the material behaves linearly (compare the black solid *Lin* curves across the four subfigures in Fig. 4.25). This means that, for the structural system considered here, stiffness is not sensitive to the geometric details and the risk of losing service performance due to an incorrect material assumption is low. A second take is that, for all the material behaviors considered here, the structure that has been optimized assuming the correct type of material is the one featuring highest strength. For example, consider the *Weak-T* curves in all the subfigures in Fig. 4.25: the one reaching the highest force is that in Fig. 4.25.d, where the structure was indeed optimized assuming a *Weak-T* material. The same applies to the other three material types, confirming and extending the result in Fig. 4.21 in the previous section.

To address the question on strength loss from unexpected material behavior, consider Fig. 4.25.d. Going back to our example with the FRP, a structure optimized assuming *Weak-T* material should resist a force of *ca.* 5 kN, if the material behaves as predicted. However, if the FRP system fails and the material ends up behaving as *Weak*, the maximum force drops to 1 kN, with an 80% strength loss that would likely entail collapse. An analogous loss of strength would occur for structures optimized assuming *Lin* or *Weak-C* materials, in Fig. 4.25.a and c, albeit less pronounced in the latter case due to the aforementioned, similar resisting mechanisms in the *Weak* and *Weak-C* cases. The only case not involving strength loss is that of a structure optimized assuming *Weak* material, in Fig. 4.25.b. At first sight, this may be simply reduced to a “design for the worst-case scenario” message. However, designing for the worst case is a way to define suitably large cross sections for the various structural elements. Here the problem is different, as optimisation with fixed f implies that any increase in cross section at one place requires a reduction of cross section elsewhere. Under this constraint, it is a nontrivial finding that the geometry optimized assuming *Weak* material gives the most robust structure with respect to other possible material behaviors.

4.4.4. Doubly fixed beam

In the previous section, the load was transferred to the lateral supports *via* a serial arrangement of ties working in tension, followed by the arch working in compression. In this section, a problem is devised to obtain elements in tension working in parallel with elements in compression. The system in Fig. 4.26 is proposed to this end; it is similar, but not identical, to the system in Fig. 4.10 to highlight the impact of geometric nonlinearity.

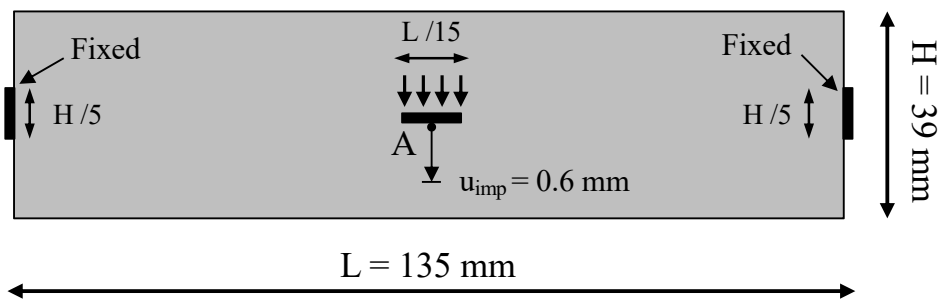


Figure 4.26 optimisation problem for a beam partly fixed on both ends and with imposed settlement at the mid point.

Fig. 4.27 shows the optimum geometries obtained with different assumptions on material behavior. Despite the symmetry of the system in Fig. 4.26, the structure optimized assuming

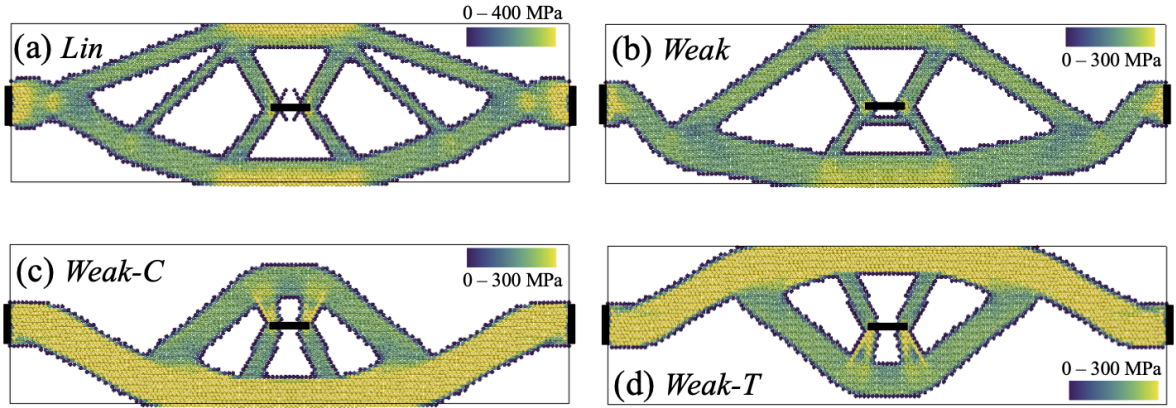


Figure 4.27 Optimum geometries for the problem in Fig. 4.26, with solid fraction $f = 0.4$ and for different material behaviors as per Table. 4.3: a) linear elastic, b) symmetrically strain-hardening, c) strain-hardening in compression and strain-stiffening in tension, and d) strain-hardening in tension and stiffening in compression. The colors represent the intensity of the deviatoric von Mises stress.

Lin material is asymmetric with respect to the horizontal axis: see Fig. 4.27.a. The asymmetry stems from geometric nonlinearity, which generates additional tensile stresses and thus favors concentration of material in the lower half of the structure. Fig. 4.27b shows the optimum structure for a symmetrically strain-hardening material, *Weak*. The material nonlinearity enhances the asymmetry caused by the geometric nonlinearity, while mass is more concentrated in the main compressed arch and lower deck in tension, removing some of the diagonal struts that were present in Fig. 4.27.a. Fig. 4.27.c shows the optimum structure for a material that is weak in compression only, *Weak-C*. This case features further concentrates mass in the lower deck, which is now fully exploited in tension, whereas the weaker compressed arch is significantly reduced in size. An almost specular geometry, except for a slight asymmetry due to geometric nonlinearity, is obtained for the *Weak-T* material, as shown in Fig. 4.27.d.

Fig. 4.28 shows the force-displacement curves for the four structures in Fig. 4.27. The curve for the *Weak* case shows that the imposed displacement of 0.6 mm is triggering significant nonlinearity. Indeed, an upper bound for the strain in the structure can be estimated in $\frac{u_{imp}}{\sqrt{2}H/2}$, considering the diagonal struts and ties (assumed at 45°) in the optimum structures immediately below and above the center of the beam, and assuming that the very top and bottom rows of particles do not move vertically at all. This leads to an upper bound strain of 2.1%, which is

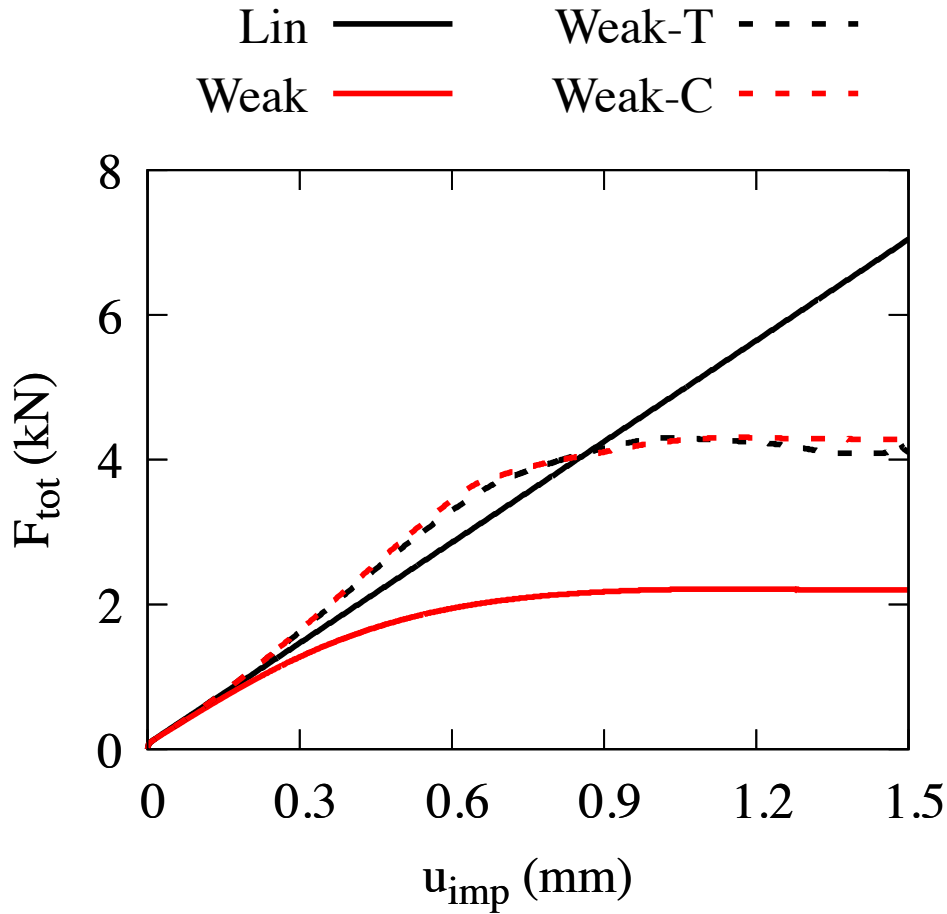


Figure 4.28 Preliminary force displacement curves for the doubly fixed beam case.

indeed well in the nonlinear regime as per Fig. 4.17, while still far from strain levels that would require consideration of diverging energy and force upon strong compression.

The results in Fig. 4.28 agree conceptually with those in the previous section, with all materials providing similar stiffness at small deformations, and with significant differences emerging at larger u_{imp} . As expected, the *Weak* material results in the lowest strength. The *Weak-C* and *Weak-T* materials lead to very similar force-displacement curves, which well reflect their almost specular geometries, combined with their specular material behaviors Fig. 4.17. Both structures with *Weak-C* and *Weak-T* materials overshoot the *Lin* curve at $u_{imp} < 0.9$ mm; this is due to the strain-stiffening behavior of the *Weak-C* and *Weak-T* materials respectively in tension and in compression, which is eventually overtaken by strain-hardening in compression and tension.

Fig. 4.29 explores how robust the structure in Fig. 4.27 are with respect to wrong assumptions of material behavior. The results in Fig. 4.29 corroborate those in Fig. 4.25 in the previous section. Namely, all structures feature a similar stiffness, meaning comparable performance

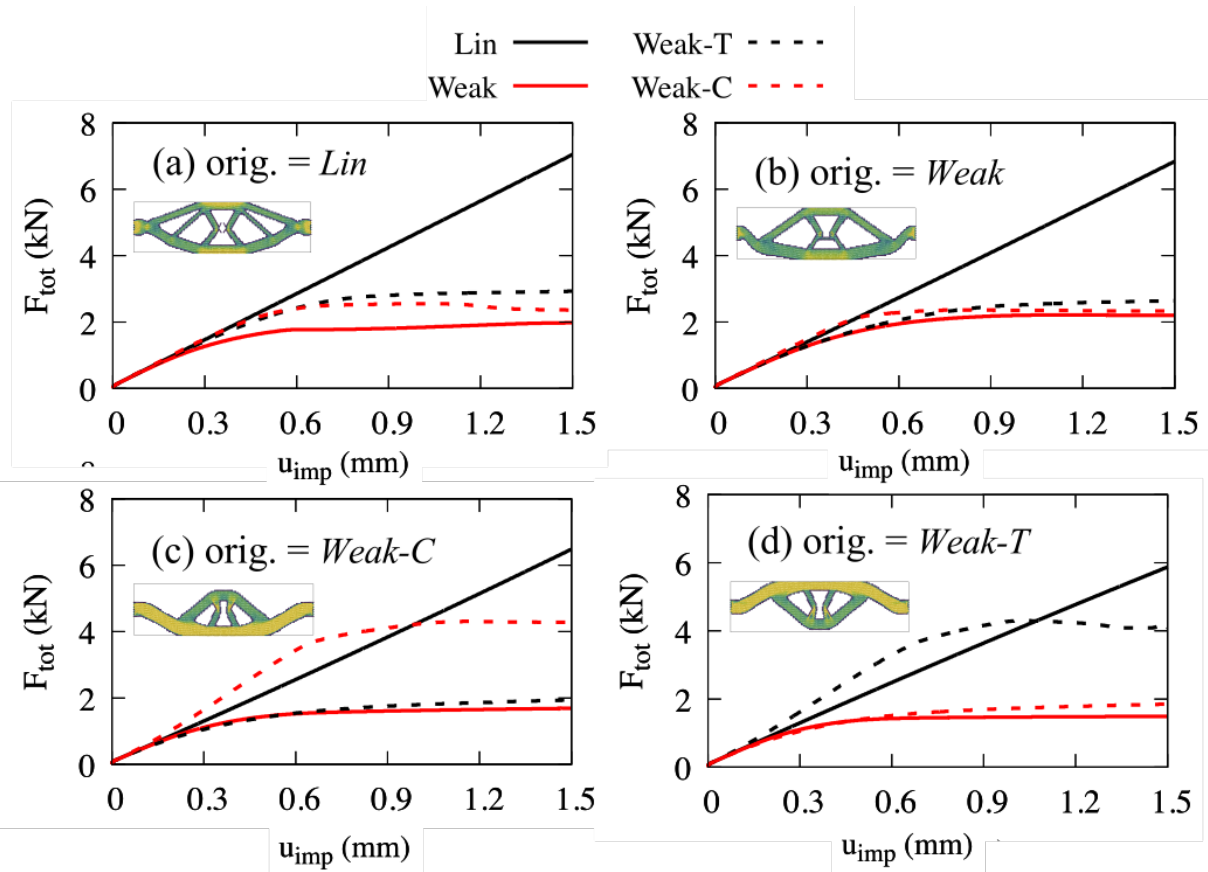


Figure 4.29 Force-displacement curves for different material behaviors and for structures originally optimized assuming the following material types: a) linear elastic *Lin*, b) symmetrically strain-hardening *Weak*, c) strain-hardening in compression and stiffening in tension *Weak-C*, and d) strain-hardening in tension and stiffening in compression *Weak-T*. The snapshots of the optimized structures are identical to those in Fig. 4.23.

in service conditions. By contrast, strength is sensitive to material behavior and geometry. Out of the structures considered here, only the structure assuming *Weak* material preserves a similar strength if the material ends up behaving differently: see Fig. 4.29.b. Instead, structures optimized assuming *Weak-C* or *Weak-T* materials, in Figs. 4.29.c and 4.29.d, would end up with as little as half their design strength if the material turns out to feature a different type of nonlinearity. This means that, also for the parallel tension-compression system considered here, assuming the weakest material behavior for the optimisation leads to the structure that is most robust against other unexpected material behaviors approaching failure.

A similar example from the literature is the work on bridge deck design in Liu and Qiao (2011) where a similar optimisation is performed with a varying ration between tensile and compressive modulus shown in Fig. 4.30. In this case similarly the structure inverts in order to utilise material either in tension or compression when it is strongest. It can be seen that when $R = \frac{1}{3.5}$ equivalent to *Weak-T* in 4.29 the optimisation generates a predominantly compressing

arch structure and $R = \frac{3.5}{1}$ equivalent to *Weak-C* the structure is instead mirrored and mostly resists load in tension. With the $R = 1$ example being a combination of the two strategies.

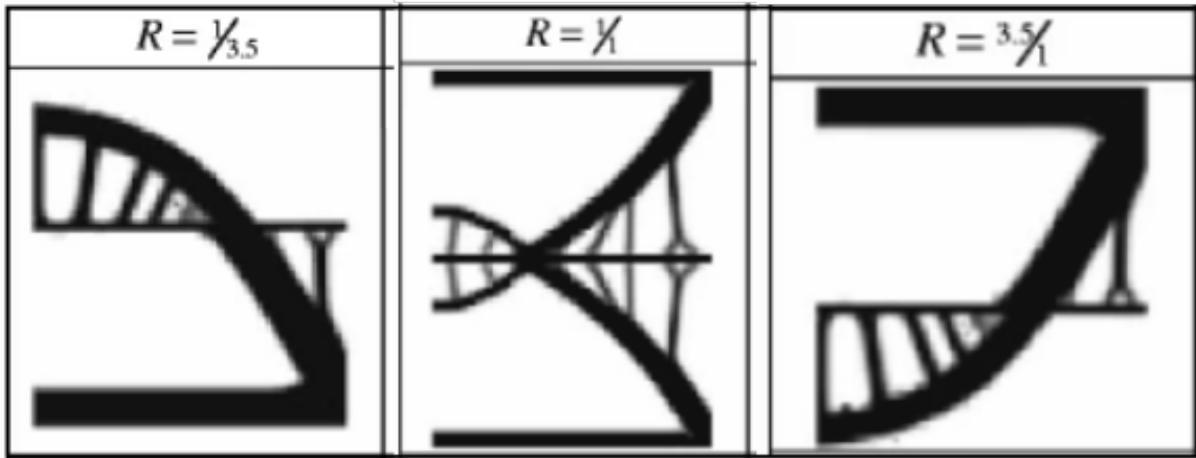


Figure 4.30 Example from Liu and Qiao (2011) showing an optimisation of a bridge deck structure using 3 different values of R the ratio of tensile to compressive modulus.

So far results have been presented for the application of DETO to quasi-static continuum beam problems following the framework in Section 3.1.1. The method has been validated against important results from the literature featuring continuum beam design and methodological constants such as the penalisation factor have been explored and set at appropriate levels to tune the method. A study on the effects of various operational parameters (mesh fineness, solid volume fraction and filtering length) have further increased confidence in the method. Geometric non-linearity is found to already be incorporated here. Systems incorporating material non-linearity are also included by swapping out the interaction potentials between particles allowing the method to capture nonlinear behaviours of systems under large deformation.

So far results have been limited to quasi-static analysis of systems with unbreakable potentials and simple complementary energy cost functions reproducing the functionality of conventional FEM-based TO. Example cases have been shown under large deflections and approaching failure, however fracture and therefore discontinuity have not been featured. Neither have dynamic granular systems which feature inherent discontinuity, despite this being one of the main attractions of the DEM method. The next Chapter will show results incorporate post-failure behaviour and granular dynamics into topology optimisation using the full DETO implementation described in Section 3.2.1

Chapter 5. Extended result: Discontinuous system optimisation

Chapter 3.2 put forward a method to extend the initial DETO methodology and incorporate three dimensional systems, include more complex potentials, and provide the ability to define arbitrarily complex objective functions, e.g for combining multiple static or dynamic simulation results. The method is implemented in a code called DETO_3D that leverages the DEM capabilities of LAMMPS to perform general and efficient optimisations; results from this code are presented here. Firstly the extended code is validated against results from the simpler 2D code which provides a measure of the efficiency gain from the use of parallel processing. Then new optimisation results are shown, which take advantage of the additional capabilities of the extended code; the results cover in particular 3D systems, optimisations combining objectives extracted from multiple static simulations and a dynamic impact and material fracture example on a simply supported beam system using bond breakage and granular contact potentials.

5.1. Validation

The simply supported beam used in the previous chapter (Fig. 4.1a) is again considered here to validate the results from the extended code DETO_3D. The system and optimisation parameters match those listed in Table. 4.1. The original code utilised an energy minimisation procedure based on the quickmin algorithm, however the new code could leverage the more efficient conjugate gradient (CG) method available in LAMMPS here. The methodology section 3.2 explained how the extended code only allows for a discrete set of χ values, associated to particle types via a user-provided *chimap* file. Here, a *chimap* featuring 20 subdivisions of χ corresponding to 20 particle types was found to be sufficient. This subdivision scheme was used throughout the rest of this section.

The structural topologies in Fig. 5.1a) match up well, when computed using the simpler 2D code and the extended DETO_3D code. Furthermore Fig. 5.1b) provides a quantitative

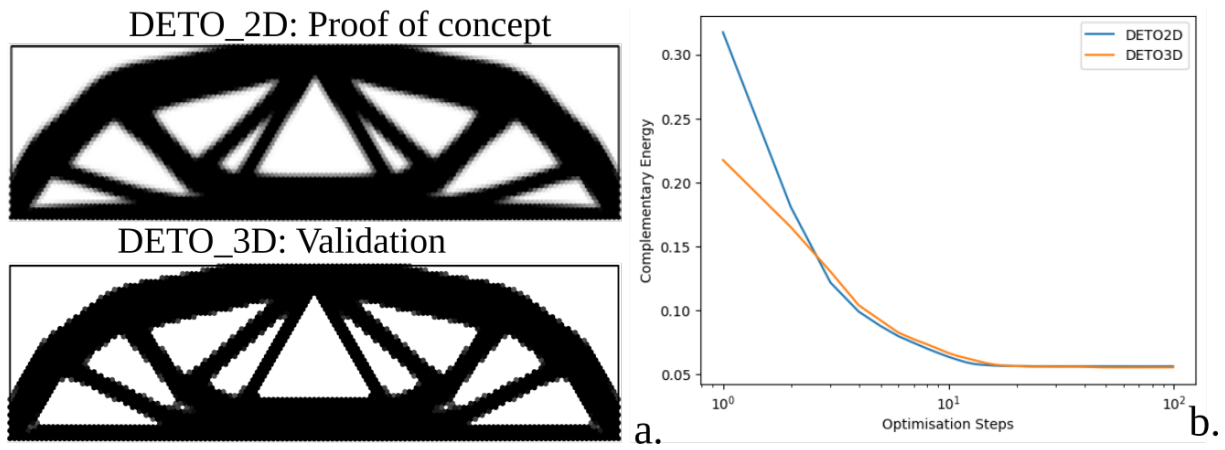


Figure 5.1 Validation of DETO_3D code, for a simply supported beam structure of 135x45 hexagonally meshed elements

validation through the evolution of CE during the optimization process.. The lower initial energy value in the DETO_3D result is likely because the CG energy minimisation algorithm is able to find a better minimum, however this does not change the final result as the effect reduces as the optimisation continues. The DETO_3D result took only 25 minutes to run distributed on 4 processors whereas the same result previously took approximately 8 hours for the same number of time steps in DETO_2D. This significant efficiency gain allows for a dramatic increase of the scale of systems DETO_3D can reasonably handle.

5.2. Parallelisation study

A detailed study of the efficiency gained from parallelisation was run in two parts corresponding to the two levels of parallelisation available in the DETO_3D code i.e multiple instances of LAMMPS can be initialised on subcommunicators that themselves can contain multiple processors. Firstly a set of optimisations were run each using a single subcommunicator with a varying the number of processors allocated. Then a separate set of optimizations were run each using exactly 12 processors but allocated to a varying number of available subcommunicators so that each subcommunicator had access to more or less computation resources. All the optimisations were a compliance minimisation of identical 45x15 elements in x and y simply supported beams and using the same parameters in Table. 4.1, run on the same computer architecture. The optimisations used the finite difference update method which requires a separate simulation for each particle perturbation per update step which can be divided between separate subcommunicators and run in parallel.

For the study in Fig. 5.2a) and b) a single subcommunicator was used and processors were added, this means only one LAMMPS instance handled all the particle perturbation simulations, each run on multiple processors via domain decomposition. Fig. 5.2a) shows the duration of each optimisation step over the first 100 steps whilst b) shows the average for each optimisation. The quadratic fit curve in Fig. 5.2b) shows there is a minimum where adding additional processors to the same subcommunicator is no longer beneficial and in fact slows down the process. This type of parallel slowdown results because, as more processors are added time spent on inter-processor communication outweighs the benefits of parallel computation. Since processors are frequently reliant on ghost atom information from their neighbours they are constantly communicating this information between each other. With additional processors the proportion of time spent on communication grows until it dominates the step duration and creates the slow down evident in Fig. 5.2b) for the system studied here seven processors produced the highest efficiency with step duration approximately halved over a single processor. However the benefit of a greater number of processors is related to the system scale and the relatively small system used here is likely to reach a maximum efficiency with a relatively low number of processors.

In contrast to the study described above the results in Fig. 5.2c) and d) do not take up any additional computational resources, they were all run utilising 12 processors. Instead these resources are distributed differently to a varying number of subcommunicators meaning that multiple LAMMPS instances are created and the many individual perturbation simulations required per update step can be distributed across these threads. The efficiency gained is substantial as additional subcommunicators are added. Since the inter subcommunicator communication is relatively simple, only occurring once per optimisation step, after the simulations to gather the sensitivity information into a single vector for a centralised update. the results in this study were not effected by any noticeable parallel slow down and strongly suggest the benefits of a greater number of independent LAMMPS instances over providing more processors to each instance for the example of this simple system. Other optimisations are likely to respond differently to different resource allocation, therefore if optimal efficiency is desired benchmarking should be carried out on a case by case basis.

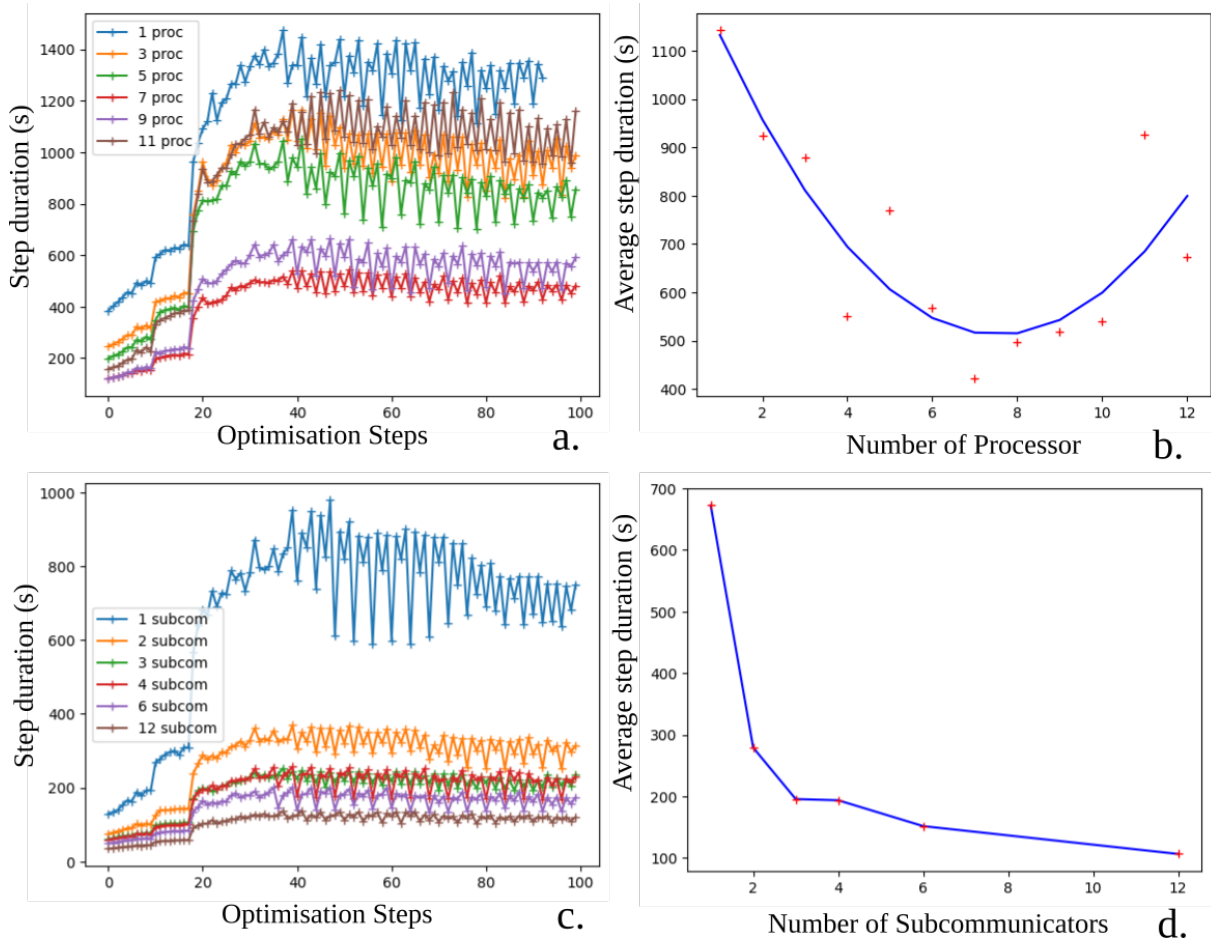


Figure 5.2 Step duration across an optimisation of a 45x15 element beam via a finite difference method a) a single sub-communicator allocated a varying number of processors. b) 12 processors distributed across a varying number of sub-communicators.

5.3. 3D optimisation

3D capabilities are an essential component of most real world optimisation problems and are therefore an important addition to the method. The capabilities of a 3D optimisation are showcased by the cantilever beam system in Fig. 5.3. This example was generated from a design domain containing approximately 82,000 particles and the loading conditions are shown in 5.3 a). The system was optimised using the same compliance based objective function that has already been used extensively with a volume fraction of 0.3. Taking advantage of the CG algorithm in LAMMPS and distributing the simulations, via domain decomposition, over 8 processors allowed this optimisation to be completed in approximately 5 hours.

The final topology in 5.3 b) shows that the design takes advantage of the additional dimension by splitting internal struts saving material and creating an overall stiffer structure. Recreating these boundary conditions in 2D would ultimately lead to a worse performing structure. The internal hydrostatic stress distribution is shown in the structure in Fig. 5.3 similar to previous 2D example this shows a strong utilisation of material in the optimised structure.

This result is taken as a bench mark optimisation for the capabilities of DETO_3D for large 3D simulations. The input files required to run this optimisation are therefore provide in Appendix A including the input script, chimap and potentials file used. When compared to 3D cantilever topology optimisation results from the literature such as those in (Yago et al., 2022) qualitatively similar structures can be recognised as well as similar material performance.

5.4. Multiple load case optimisation

An important advancement made by the DETO_3D code is the capability to consider multiple load cases in an optimisation. to do so unique loading senarios can be defined for the same structural boundary conditions. Multiple simulations are then performed per optimisation step, extracting separate objectives which are combined via a weighted sum method Li et al. (2020). Here these capabilities of the method are explored, via a simple optimisation combining two static compliance based objectives.

When multiple simulations are defined in a DETO_3D input script each simulation is associate with its own objectives. During each optimisation step all simulations are completed on the same system configuration and then objectives are combined via a user defined cost function

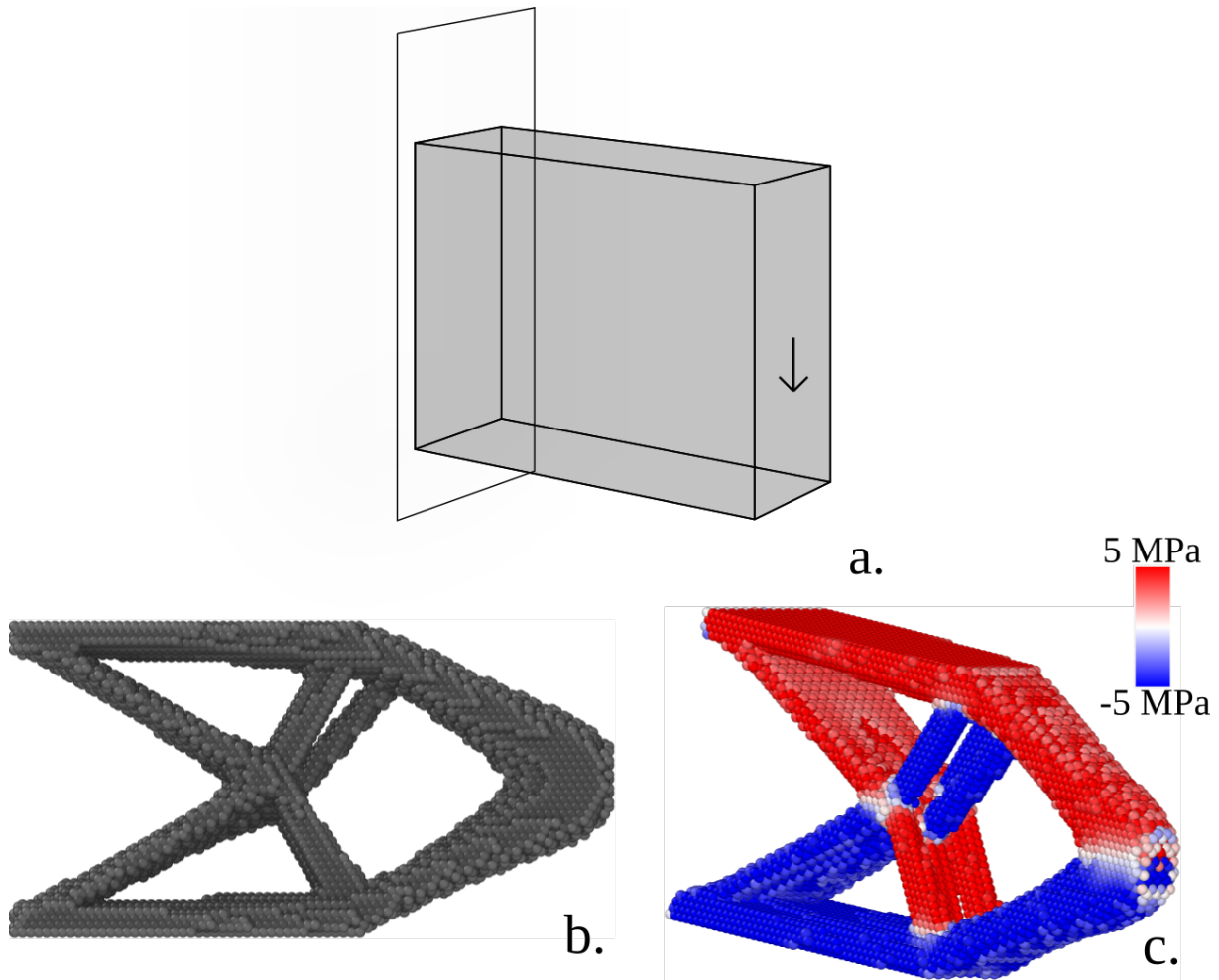


Figure 5.3 Example of a 3D optimised cantilever beam made up from a $75 \times 45 \times 25$ hexagonal close packed (HCP) lattice of particles. a) boundary conditions of the problem b) the optimal design c) representation of the hydrostatic stress in the optimal structure.

as described in section 3.2 before an update is derived. One possible use of this feature within the context of static compliance optimisations is to optimise a structure towards multiple competing load cases. This has the effect of weighing the priorities of each load case for the optimisation to hopefully account for variability or uncertainty in the loading conditions of the system and make the output less specialised for a particular case.

Here a simple but illustrative example is shown in the form of the cantilever beam design in Fig. 5.4. Two symmetrically opposing loading conditions are considered in a) and b) with the output design for each case applied separately. The relatively strong applied load at the end of the cantilever produces asymmetrical structures due to the effect of geometric non-linearity. In each case the single objective is a simple complementary energy minimization. However in c) both load cases are applied and the averaged multi-objective function in Eq. 5.1 is applied

$$C = 0.5C_1 + 0.5C_2 \quad (5.1)$$

where C_1 and C_2 are the objectives of the two previous simulations, i.e the complementary strain energies from the individual load cases. The result is an optimisation balancing the priorities of the two objectives ultimately resulting in a symmetrical design, as expected for the loading conditions and equal weight factors employed. d) shows the progression of the objective function in each case (for a and b this is complementary energy and for c this is the combined objective in Eq. 5.1). Overall c) performs the worst here as its objective is a combination of the two scenarios for a) and b) for which it is not an optimal design. A benefit of the design in c) however is that its response to either of the load cases in a) and b) would be the same and it does not have a weakness to either. The result in c) shows some significant areas of grey material near the base of the cantilever where the optimisation has been slow to allocate material. This could be as a result of the competing load cases tending towards a significantly different result for this portion of the structure, making the optimum ultimately harder to find.

This simple example is illustrative of the general principle of multi-objective optimisation. The procedure here could be straightforwardly extended to incorporate more complex geometries and more numerous boundary conditions that could be derived for example from the loading conditions of a particular structure with weights used to prioritise certain critical loading conditions.

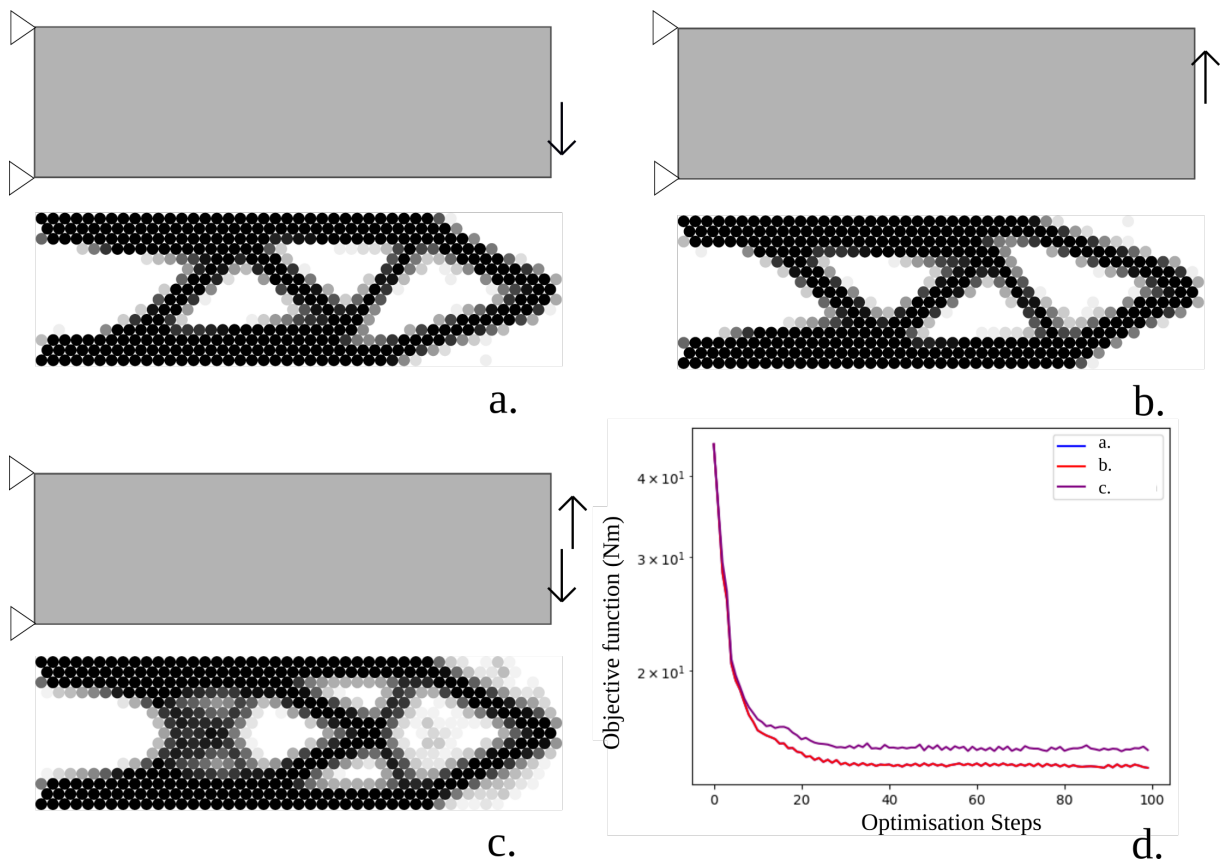


Figure 5.4 Illustrative example of a multi simulation optimisation on a cantilever beam result in c) is a combination the load cases from a) and b) into a single optimisation process. d) shows the progression of the cost functions in each case over the duration of the optimisation.

5.5. Multiple objective optimisation

Resistance to material fracture and crack propagation in structures is an important topic in TO as a way to help design structural systems with robustness to the effects of damage and to avoid catastrophic impacts. FEM techniques have struggled to capture these processes meaningfully whereas DEM is well suited to modeling the discontinuous processes of material fracture. The aim in this section is to develop a procedure, objective and set of boundary conditions to enhance the mechanical resistance to damage or cracks in structures and materials.

Taking lead from a key result from the Literature review in Fig. 2.6 The example described here has utilised a minimum fracture energy objective function the idea being to attempt to minimise the extent of failure if only small local damage takes place and improve the compartmentalisation of the structure should member failure occur i.e maintaining the integrity of the rest of the structure given a missing member.

In the case here a multi-simulation procedure is used. This procedure utilises a fracture energy minimisation for a soft impact scenario on a simply supported beam, measured as the difference between the initial kinetic energy added to the system and the final total energy once it has settled to a steady value indicating that no more bond breakage will occur. This is combined with a compliance minimisation of the system in a conventional static analysis. Optimizing these two objectives together ultimately makes it possible to circumvent a well know insure with fracture energy minimisation; the optimiser targeting a global minimum by disconnecting the applied velocity from all supports creating an unstable but optimal system without any fracture. The effect of the compliance objective is to ensure a connected structure with at least some load bearing capacity.

The system is made up of a 45x15 particle lattice of bonded particles sharing the properties from Table. 4.1 except now bonds are irreversibly removed when reaching a 10% strain in tension or in compression and a granular Hertzian contact force field is applied to model post failure impacts between particles. The optimisation incorporates the two simulations with the boundary conditions shown in Fig. 5.5 labeled sim 1 and sim 2. In sim 1, a soft impact scenario is applied, at the start of the simulation a small section of the beam is initialised with a velocity of 2 m/s. This simulation features a dynamic Discrete Element run (i.e. explicit integration of Newton's second law of motion as per Section 2.2) of 20,000 time steps of duration $10e^{-4}$ s for

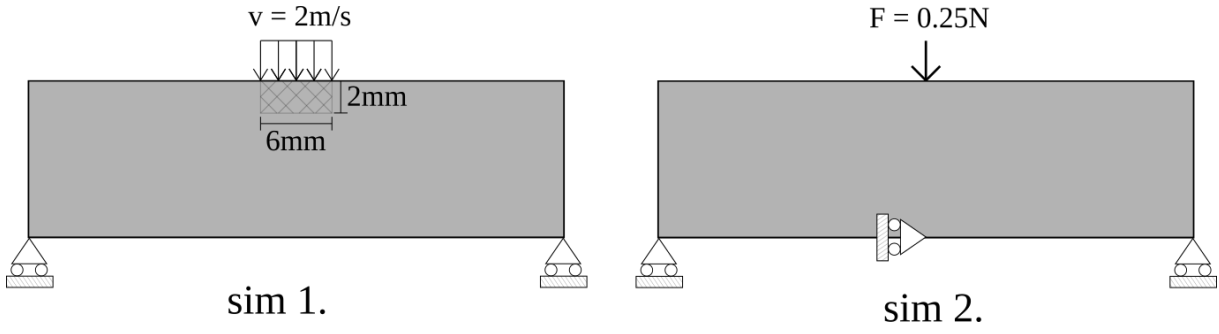


Figure 5.5 Boundary conditions for both simulations making up the combined fracture energy minimisation study sim 1) representing a soft impact of an applied velocity of 2m/s over an area of 6mm x 2mm at the top of the beam and sim 2) an applied force of 0.25N applied as a point load also at the top of the beam.

a total duration of 20 seconds simulation time. The applied velocity is great enough to cause significant bond strain triggering breakage and fracture propagation and a resultant dissipation of energy from the removed bonds. The rest of the systems energy after impact is made up from the residual kinetic, bond, and contact energy. Since no viscose damping was utilised in this simulation this difference in energy is equivalent to the energy dissipated by fracture.

The second simulation, sim 2, is a conventional static compliance minimisation featuring an applied point load again at the top of the system. System behaviour is determined via a static energy minimisation and the relatively low load intensity of 0.25N does not produce any bond strains higher than 10%, meaning no breakage occurs. The cost function is a straightforward linear combination of fracture energy, from the first simulation and complementary energy, from the second.

$$C = \alpha U_{frac} + \beta U^* \quad (5.2)$$

where α and β are weighting parameters that can be adjusted to weight the priorities of the individual objectives. Each objective is taken from their respective simulation. U^* is relatively straight froward to compute as it can be extracted simply from the final equilibrium configuration of the system after sim 2 is run, since an analytic derivative is available this objective can take advantage of the relatively efficient direct derivative method to construct its sensitivity. However since the U_{frac} is the result of dynamic irreversible processes during sim 1 a derivative cannot be formulated straightforwardly and so a finite difference approach should be undertaken to determine the sensitivity of this objective.

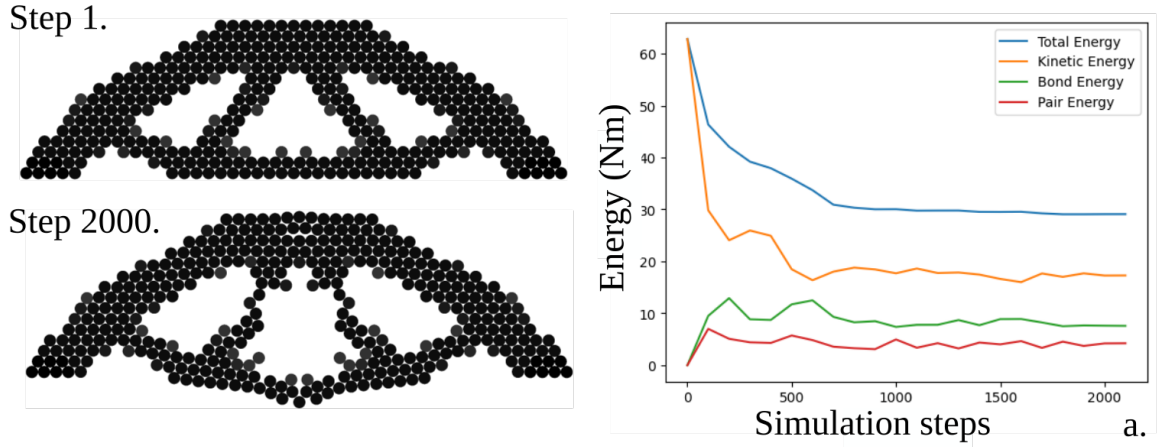
The optimisation described above was run with the resulting topologies presented in Fig. 5.6. Firstly in Fig. 5.6a) a bench marking optimisation taking into account only the complementary energy from sim 2 was run using values of $\alpha = 0$ and $\beta = 1$ in Eq. 5.2. Then in in Fig. 5.6b) an optimisation taking both sim 1) and 2) was run using uniform weights of α and $\beta = 0.5$. Finally in in Fig. 5.6c) a simulation taking into account only the fracture energy from sim 1 was undertaken, in this case the result is clearly not useful as it is a completely separated structure, however from the standpoint of minimum fracture energy this turns out to be a highly optimal structure, since there is no material beneath the point of impact no fracture can occur there. All three output topologies are shown at the top left of Fig. 5.6 a), b) and c) below each is shown the same topology after undergoing the soft impact scenario from sim 1)

Fig. 5.6 a and b show that the two objectives prioritise different topologies. Firstly the fracture optimised result generally produces a less black and white solution. Instead this result produces more, less stiff, grey areas. The compliance minimization result in a) predictably props up the centrally applied load in a straight forward fashion taking advantage of two central compression members. Whereas the topology in b) taking into account impact and subsequent fracture, tends towards removing material from the center of the structure, under the load, with the compression members migrated outward. This creates a fracture region under the load. Additionally where the bottom of the truss is connects to the main arch is significantly weakened and remains grey, creating a fracture here under impact that separates the base of the truss from the main structure. A similar topology is shown in Fig. 2.6 of the literature review for an impact problem that also forms a void in the center of the beam under the load, in that case this helped the system with compartmentalisation of local damage, to avoid damage propagation from the impact, this effect can be seen here too although less noticeably because of the small size of the results in Fig. 5.6

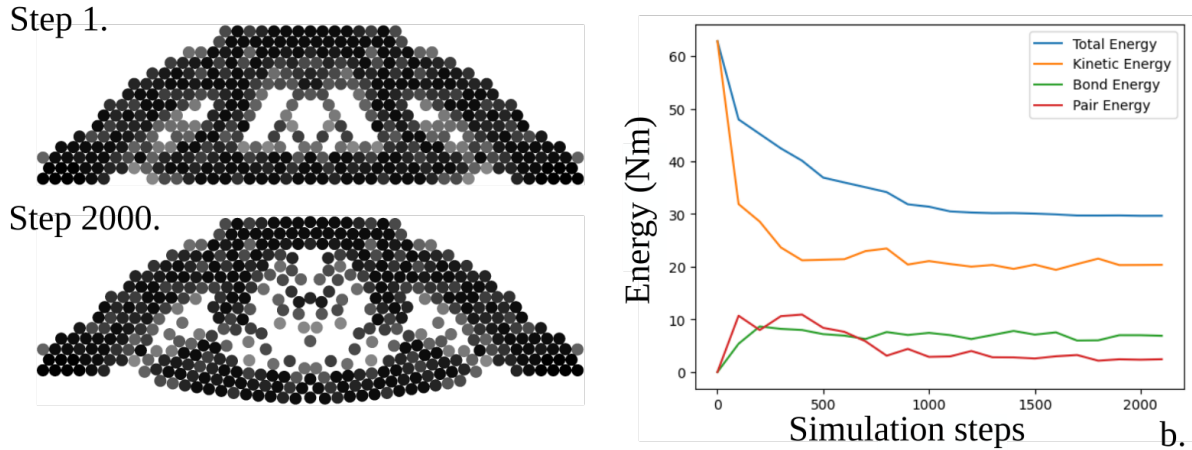
Fig. 5.7 shows a lower fracture energy throughout the process for the structure optimised towards fracture compared to the one only optimised for compliance when subjected to a 2m/s impact.

The fracture pattern can be seen in greater detail in Fig. 5.8 where the final system configuration is shown again, but with particles featuring at least one broken bond highlighted in red. In the fracture optimised case a larger more diffuse fracture is propagated throughout the center of the beam. Because the fracture in b) mostly propagates through the softer grey elements despite

Solely compliance optimisation function



Combined optimisation function



Solely fracture optimisation function

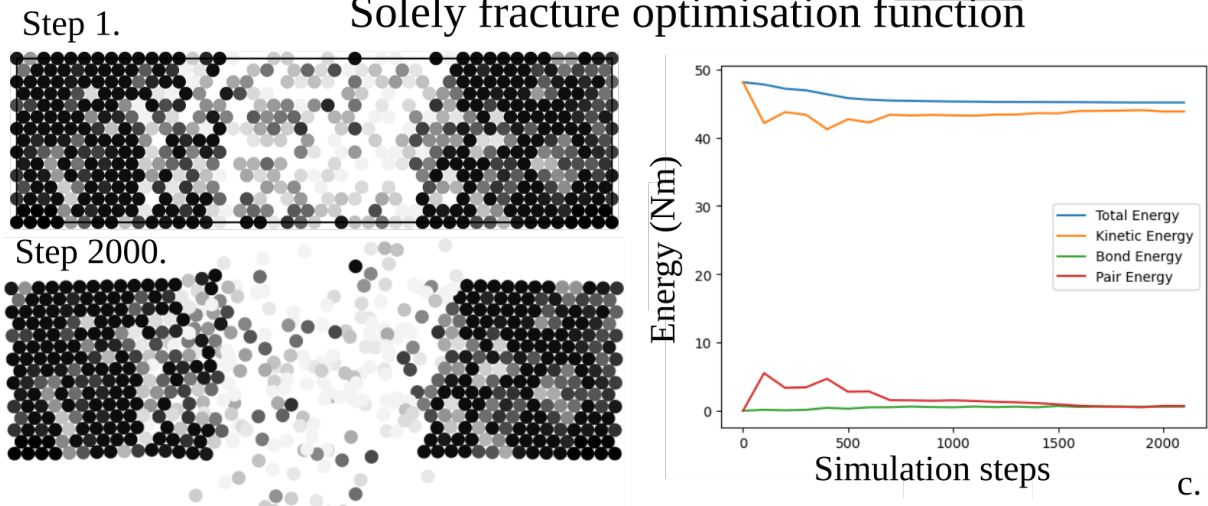


Figure 5.6 Optimisation for resistance to a dynamic impact scenario, a) shows an example of a truss system optimised for compliance only and the final system configuration after a dynamic impact as well as the system energy over the course of the simulation. b) shows a system optimised instead using a combined fracture-compliance objective as well as its final configuration and energy. c) shows a degenerated optimisation that is optimised only fracture energy minimisation

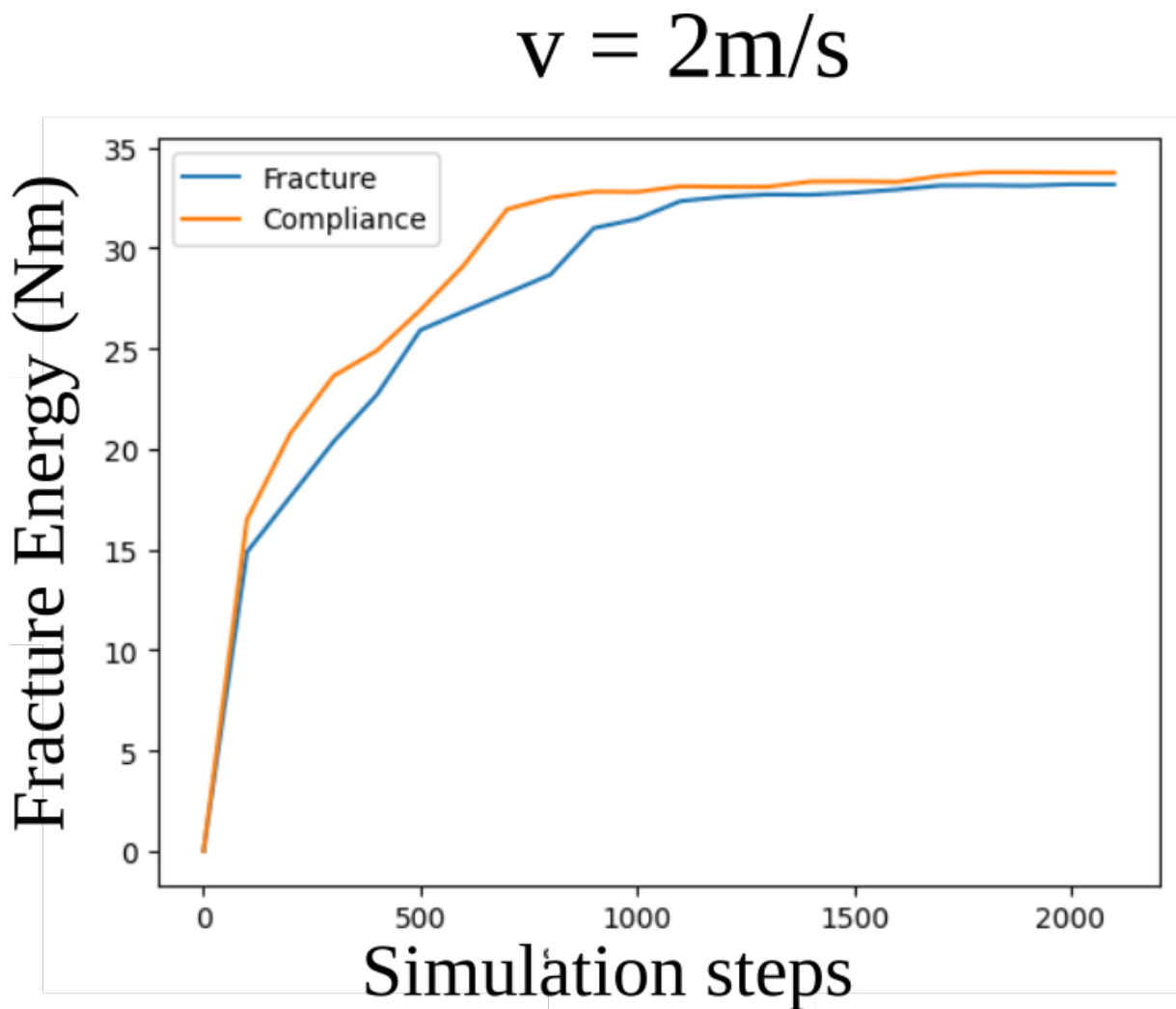


Figure 5.7 Fracture energy plotted across the duration of a impact simulation, for both the fracture and compliance optimised systems for an impact velocity of 2m/s.

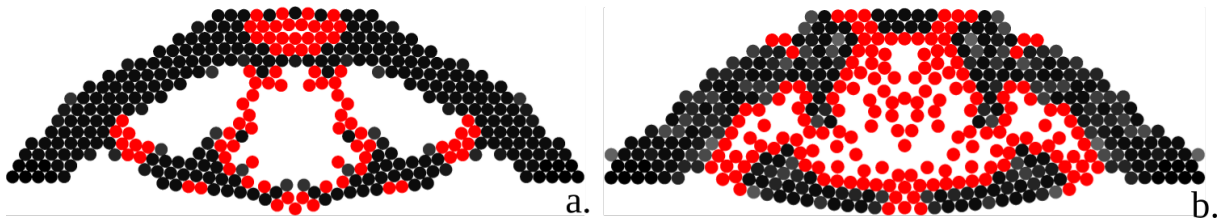


Figure 5.8 Fractured final configurations of the two proposed designs with particles highlighted in red if they have one or more broken bonds which allows for the visualisation of the fracture pattern in the damaged structures.

a larger fracture in b) this still represents a lower overall fracture energy in the system. These grey elements have less stiff bonds connecting them so breakage in these region corresponds to a lower energy dissipation.

Chapter 6. Conclusions and outlook

A comprehensive method for the topology optimisation of systems of discrete interacting particles has been presented in this thesis. The fundamental principles of continuum based TO optimisation have been mapped successfully onto a new framework of bonded and discontinuous systems interacting via particle to particle interaction potentials using the Discrete Element Method (DEM). This thesis has argued that this DEM based TO approach can overcome many of the limitations with discontinuous processes in FEM and has aimed to facilitate the use of TO for a range of systems and processes that were previously of limits including material fracture, fragmentation, and even granular systems.

The proposed changes to the SIMP-TO method elaborated in Chapter 3.1 revolve around relocating the central approach to stiffness penalisation from individual elements to instead act between elements on the particle to particle interaction stiffness' that define the behaviour of DEM systems. From this proposed change the rest of a complete formulation for the basic discrete element topology optimisation is derived. This includes restating a conventional compliance minimisation problem to incorporating an objective function, and sensitivity derivative that respect this new penalisation between particles. The cost function is related here to the complementary energy of the system rather than the directly to the strain energy since geometric non-linearity is theoretically always possible in DEM systems, however it is proposed to use a strain energy formulation as a strong approximation. This makes the computation of a sensitivity derivative significantly easier. Two methods are presented for computing sensitivities, the first is the analytic approach to taking a direct derivative of the strain energy to find the partial derivative, and the second is a numerical approach using the finite difference method to obtain the full derivative at the cost of one complete solution to the cost function per particle at each update step.

A proof of concept method has been presented that operates on a framework of continuum systems approximated as unbreakable lattices of bonded particles where system performance

is derived from a damped dynamic energy minimisation. These systems closely resemble the conditions for a conventional static FEM based TO and in Chapter 4 allow for the validation of the method against well known examples from the literature including a simply supported truss from one of the most influential TO codes (Sigmund, 2001). In fact the approximation of the full complementary energy derivative from Chapter 3.1 is shown to hold for systems studied here. The effect of the penalisation exponent is examined and a value of $p = 2$ is selected to improve the performance and quality of the solutions from the method. Then a study on the effects of various operational parameters (mesh fineness, solid volume fraction and filtering length) are presented highlighting their effect on the optimisation process and showcasing the versatility of this method. The proposed methods of analytic and numerical sensitivity computation are extensively tested. Both geometric and non-linearities are shown to already be incorporated at this stage and examples utilising non-linear interactions showing that the method permits the use of arbitrary interaction potentials straight forwardly as well. Both forms of non-linearity are shown to have a significant impact on output topologies.

An extended DETO methodology is elaborated in Chapter 3.2 to encompass a wider range of systems and processes that might conventionally be modeled by the DEM highlighting the examples of continuum beam fracture and of granular systems such as the design of rheometer blades that have motivated in part the development of a DETO method. The approach is extended to include three dimensional systems, complex interaction potentials that may for example feature history dependant and irreversible interaction effects such as breakage criteria, and to handle complex multi-objective optimisations potentially drawing information from multiple concurrent simulations including dynamic or static system behaviour. With these additions the method is theoretically extended to a much richer range of systems and processes captured by conventional DEM. In order to accomplish these the code implementation of this extended method takes advantage of the state-of-the-art particle dynamics simulator LAMMPS. The code gains access to the rich range of DEM functionality available in LAMMPS including complex interaction potentials and parallelisation. This software implementation takes uses parallelisation in two separate ways, firstly individual simulations can be divided across several processors using the domain decomposition capabilities of LAMMPS. Secondly the method can distribute separate systems to be solved in parallel to separate groups of processors called subcommunicators.

Results in Chapter 5 showcase the new functionality of the extended method. Parallelisation is shown to have a significant impact on the duration of optimisations. The domain decomposition approach is shown to have a maximum number of processors before efficiency is harmed by the large amount of inter processor communications actually slowing the optimisation down. Whereas initialising separate parallel simulation threads does not exhibit this behaviour. Complex systems of a discontinuous nature are handled in the form of two case studies. Firstly an impact scenario on a simply supported beam system involving dynamic DEM analysis and history dependant breakable interactions is explored. This system utilises a multi-simulation optimisation that combines the fracture energy minimisation of a dynamic impact scenario with a conventional static compliance minimisation to avoid a well know issue of disconnected topologies, this hybrid cost function is shown to create effective topologies, reducing some of the common problems arising from dynamic TO. The output topology is then shown to have a lower total fracture energy and therefore a higher damage tolerance in comparison with a separate system optimised for compliance alone.

Appendix A. Example input for DETO_3D

Sample input files

This appendix contains all the necessary input files to run the 3D cantilever simulation from Fig. 5.3 including the input script, chimap and potentials file to act as an example optimisation. Further details on all the commands used here can be found at the DETO_3D official documentation online at: <https://connor-os.github.io/DETO/>

Input Script

```
# ===== SET OPTIMIZATION STYLE AND PARAMS =====
opt_type 0.1 0.3 gradient_descent
objective_function v_c1

# ===== SET UNIVERSE OF SUB-COMMUNICATORS =====
subcomm 1

# ===== INITIAL SETTINGS FOR ALL LAMMPS INSTANCES =====
lammps dimension 3
lammps units si
lammps boundary s s s
lammps atom_style hybrid sphere bond
lammps comm_modify vel yes cutoff 3.2
lammps newton off
lammps special_bonds lj 0 1 1 coul 0 1 1
lammps timestep 10-4
```

Example input for DETO_3D

```
lammps variable      nelx equal 75
lammps variable      nely equal 45
lammps variable      nelz equal 25
lammps variable      radius equal 0.5

#
lammps lattice        hcp $(v_radius*2)
lammps region         box block 0 $(v_nelx-1) 0 $((v_nely-1)/2) 0 $((v_nelz-1)/2) \\\
units lattice
lammps create_box     22 box bond/types 253 extra/bond/per/atom 12
lammps create_atoms   9 box
lammps fix            sens_fix all property/atom d_sens
lammps fix            1 all nve

lammps neighbor       2.2 bin #include ghost atoms within cutoff
lammps neigh_modify   delay 0

lammps region support_reg block INF 1.1 INF INF INF INF units lattice
lammps region fource_reg block $(((v_nelx-1))-1) INF \\\
$((sqrt(3)/2*v_radius)*(v_nely/2+1))      \\\
$((sqrt(3)/2*v_radius)*(v_nely/2+5))      \\\
$((sqrt(3)/2*v_radius)*(v_nelz/2)) $((sqrt(3)/2*v_radius)*(v_nelz/2+3)) \\\
units lattice
lammps group support_group region support_reg
lammps group force_group region fource_reg
lammps fix support support_group setforce 0 0 0

lammps compute stress all stress/atom NULL bond
lammps variable shyd atom -(c_stress[1]+c_stress[2]+c_stress[3])/3
```

```

lammps variable sdev atom sqrt(0.5*((c_stress[1]-c_stress[2])^2+ \\
(c_stress[2]-c_stress[3])^2+(c_stress[3]-c_stress[1])^2)+ \\
3*(c_stress[4]^2+c_stress[5]^2+c_stress[6]^2))

lammps compute          sens_comp all property/atom d_sens

lammps compute          nbond all nbond/atom
lammps compute          tbond all reduce sum c_nbond

#write_plog yes #use this to create processor specific logs for de-bugging
write_lmp_log no #use this to turn of lammps logs
write_restart ./dump/data.validation # save a restart file after each sucessful \\
optimise step so that you can restart the optimisation
dump      1 all custom 1 ./dump/dump.3D_cant id type x y z v_shyd v_sdev d_sens \\
c_nbond

# ===== LOADING CHI MAP AND POTENTIALS FILE =====
opt_map_chi ./inputs/chimap.dat
read_potentials ./inputs/potfile.dat

# # ===== SET SIMULATION NUMBER 1 (RUN TYPE, NO REPEAT)=====
simulation Sim1 run repeat no

add_attribute Sim1 fix          force force_group addforce 0 0.1 0
add_attribute Sim1 fix_modify  force energy yes

add_attribute Sim1 minimize      1.0e-12 1.0e-12 1000000 20000000

add_attribute Sim1 python        compute_sens input 1 SELF format p file \\
./inputs/py_utils.py

```

Example input for DETO_3D

```
add_attribute Sim1 python          compute_sens invoke
```

```
#Define variables for cost function
```

```
add_attribute Sim1 variable cost_func equal ebond
```

```
# #===== objective variables =====
```

```
add_objective Sim1 c1 cost_func
```

```
add_sensitivity Sim1 s1 d_sens
```

Chimpap

```
num_mat 1
```

```
PROPERTIES: chi material type mass
```

```
0.05 homo 2 0.05e-7
```

```
0.1 homo 3 0.1e-7
```

```
0.15 homo 4 0.15e-7
```

```
0.2 homo 5 0.2e-7
```

```
0.25 homo 6 0.25e-7
```

```
0.3 homo 7 0.3e-7
```

```
0.35 homo 8 0.35e-7
```

```
0.4 homo 9 0.4e-7
```

```
0.45 homo 10 0.45e-7
```

```
0.5 homo 11 0.5e-7
```

```
0.55 homo 12 0.55e-7
```

```
0.6 homo 13 0.6e-7
```

```
0.65 homo 14 0.65e-7
```

```
0.7 homo 15 0.7e-7
```

```
0.75 homo 16 0.75e-7
```

```
0.8 homo 17 0.8e-7
```

0.85 homo 18 0.85e-7
0.9 homo 19 0.9e-7
0.95 homo 20 0.95e-7
1 homo 21 1e-7

Potentials file

#potfile generated with input_gen.py

pair_style zero 1.0

pair_coeff * *

group 1 type 1 #chi equal 0.0
group 2 type 2 #chi equal 0.05
group 3 type 3 #chi equal 0.1
group 4 type 4 #chi equal 0.15
group 5 type 5 #chi equal 0.2
group 6 type 6 #chi equal 0.25
group 7 type 7 #chi equal 0.3
group 8 type 8 #chi equal 0.35
group 9 type 9 #chi equal 0.4
group 10 type 10 #chi equal 0.45
group 11 type 11 #chi equal 0.5
group 12 type 12 #chi equal 0.55
group 13 type 13 #chi equal 0.6
group 14 type 14 #chi equal 0.65
group 15 type 15 #chi equal 0.7
group 16 type 16 #chi equal 0.75
group 17 type 17 #chi equal 0.8
group 18 type 18 #chi equal 0.85
group 19 type 19 #chi equal 0.9

Example input for DETO_3D

```
group 20 type 20    #chi equal 0.95
group 21 type 21    #chi equal 1.0
group 22 type 22    #non-opt
```

```
bond_style harmonic
```

```
bond_coeff 1 0.001 1
```

```
bond_coeff 2 0.001 1
```

```
bond_coeff 3 0.001 1
```

```
bond_coeff 4 0.001 1
```

```
bond_coeff 5 0.001 1
```

```
bond_coeff 6 0.001 1
```

```
bond_coeff 7 0.001 1
```

```
bond_coeff 8 0.001 1
```

```
bond_coeff 9 0.001 1
```

```
bond_coeff 10 0.001 1
```

```
bond_coeff 11 0.001 1
```

```
bond_coeff 12 0.001 1
```

```
bond_coeff 13 0.001 1
```

```
bond_coeff 14 0.001 1
```

```
bond_coeff 15 0.001 1
```

```
bond_coeff 16 0.001 1
```

```
bond_coeff 17 0.001 1
```

```
bond_coeff 18 0.001 1
```

```
bond_coeff 19 0.001 1
```

```
bond_coeff 20 0.001 1
```

```
bond_coeff 21 0.001 1
```

```
bond_coeff 22 0.001 1
```

```
bond_coeff 23 0.0025 1
```

```
bond_coeff 24 0.005625 1
```

```
bond_coeff 25 0.01 1
```

bond_coeff 26 0.01563 1
bond_coeff 27 0.0225 1
bond_coeff 28 0.03063 1
bond_coeff 29 0.04 1
bond_coeff 30 0.05063 1
bond_coeff 31 0.0625 1
bond_coeff 32 0.07563 1
bond_coeff 33 0.09 1
bond_coeff 34 0.1056 1
bond_coeff 35 0.1225 1
bond_coeff 36 0.1406 1
bond_coeff 37 0.16 1
bond_coeff 38 0.1806 1
bond_coeff 39 0.2025 1
bond_coeff 40 0.2256 1
bond_coeff 41 0.25 1
bond_coeff 42 0.01 1
bond_coeff 43 0.0225 1
bond_coeff 44 0.04 1
bond_coeff 45 0.0625 1
bond_coeff 46 0.09 1
bond_coeff 47 0.1225 1
bond_coeff 48 0.16 1
bond_coeff 49 0.2025 1
bond_coeff 50 0.25 1
bond_coeff 51 0.3025 1
bond_coeff 52 0.36 1
bond_coeff 53 0.4225 1
bond_coeff 54 0.49 1
bond_coeff 55 0.5625 1

Example input for DETO_3D

bond_coeff 56 0.64 1
bond_coeff 57 0.7225 1
bond_coeff 58 0.81 1
bond_coeff 59 0.9025 1
bond_coeff 60 1 1
bond_coeff 61 0.05062 1
bond_coeff 62 0.09 1
bond_coeff 63 0.1406 1
bond_coeff 64 0.2025 1
bond_coeff 65 0.2756 1
bond_coeff 66 0.36 1
bond_coeff 67 0.4556 1
bond_coeff 68 0.5625 1
bond_coeff 69 0.6806 1
bond_coeff 70 0.81 1
bond_coeff 71 0.9506 1
bond_coeff 72 1.102 1
bond_coeff 73 1.266 1
bond_coeff 74 1.44 1
bond_coeff 75 1.626 1
bond_coeff 76 1.823 1
bond_coeff 77 2.031 1
bond_coeff 78 2.25 1
bond_coeff 79 0.16 1
bond_coeff 80 0.25 1
bond_coeff 81 0.36 1
bond_coeff 82 0.49 1
bond_coeff 83 0.64 1
bond_coeff 84 0.81 1
bond_coeff 85 1 1

bond_coeff 86 1.21 1
bond_coeff 87 1.44 1
bond_coeff 88 1.69 1
bond_coeff 89 1.96 1
bond_coeff 90 2.25 1
bond_coeff 91 2.56 1
bond_coeff 92 2.89 1
bond_coeff 93 3.24 1
bond_coeff 94 3.61 1
bond_coeff 95 4 1
bond_coeff 96 0.3906 1
bond_coeff 97 0.5625 1
bond_coeff 98 0.7656 1
bond_coeff 99 1 1
bond_coeff 100 1.266 1
bond_coeff 101 1.562 1
bond_coeff 102 1.891 1
bond_coeff 103 2.25 1
bond_coeff 104 2.641 1
bond_coeff 105 3.062 1
bond_coeff 106 3.516 1
bond_coeff 107 4 1
bond_coeff 108 4.516 1
bond_coeff 109 5.062 1
bond_coeff 110 5.641 1
bond_coeff 111 6.25 1
bond_coeff 112 0.81 1
bond_coeff 113 1.102 1
bond_coeff 114 1.44 1
bond_coeff 115 1.823 1

Example input for DETO_3D

bond_coeff 116 2.25 1
bond_coeff 117 2.723 1
bond_coeff 118 3.24 1
bond_coeff 119 3.803 1
bond_coeff 120 4.41 1
bond_coeff 121 5.062 1
bond_coeff 122 5.76 1
bond_coeff 123 6.502 1
bond_coeff 124 7.29 1
bond_coeff 125 8.123 1
bond_coeff 126 9 1
bond_coeff 127 1.501 1
bond_coeff 128 1.96 1
bond_coeff 129 2.481 1
bond_coeff 130 3.062 1
bond_coeff 131 3.706 1
bond_coeff 132 4.41 1
bond_coeff 133 5.176 1
bond_coeff 134 6.002 1
bond_coeff 135 6.891 1
bond_coeff 136 7.84 1
bond_coeff 137 8.851 1
bond_coeff 138 9.922 1
bond_coeff 139 11.06 1
bond_coeff 140 12.25 1
bond_coeff 141 2.56 1
bond_coeff 142 3.24 1
bond_coeff 143 4 1
bond_coeff 144 4.84 1
bond_coeff 145 5.76 1

bond_coeff 146 6.76 1
bond_coeff 147 7.84 1
bond_coeff 148 9 1
bond_coeff 149 10.24 1
bond_coeff 150 11.56 1
bond_coeff 151 12.96 1
bond_coeff 152 14.44 1
bond_coeff 153 16 1
bond_coeff 154 4.101 1
bond_coeff 155 5.062 1
bond_coeff 156 6.126 1
bond_coeff 157 7.29 1
bond_coeff 158 8.556 1
bond_coeff 159 9.922 1
bond_coeff 160 11.39 1
bond_coeff 161 12.96 1
bond_coeff 162 14.63 1
bond_coeff 163 16.4 1
bond_coeff 164 18.28 1
bond_coeff 165 20.25 1
bond_coeff 166 6.25 1
bond_coeff 167 7.563 1
bond_coeff 168 9 1
bond_coeff 169 10.56 1
bond_coeff 170 12.25 1
bond_coeff 171 14.06 1
bond_coeff 172 16 1
bond_coeff 173 18.06 1
bond_coeff 174 20.25 1
bond_coeff 175 22.56 1

Example input for DETO_3D

bond_coeff 176 25 1
bond_coeff 177 9.151 1
bond_coeff 178 10.89 1
bond_coeff 179 12.78 1
bond_coeff 180 14.82 1
bond_coeff 181 17.02 1
bond_coeff 182 19.36 1
bond_coeff 183 21.86 1
bond_coeff 184 24.5 1
bond_coeff 185 27.3 1
bond_coeff 186 30.25 1
bond_coeff 187 12.96 1
bond_coeff 188 15.21 1
bond_coeff 189 17.64 1
bond_coeff 190 20.25 1
bond_coeff 191 23.04 1
bond_coeff 192 26.01 1
bond_coeff 193 29.16 1
bond_coeff 194 32.49 1
bond_coeff 195 36 1
bond_coeff 196 17.85 1
bond_coeff 197 20.7 1
bond_coeff 198 23.77 1
bond_coeff 199 27.04 1
bond_coeff 200 30.53 1
bond_coeff 201 34.22 1
bond_coeff 202 38.13 1
bond_coeff 203 42.25 1
bond_coeff 204 24.01 1
bond_coeff 205 27.56 1

bond_coeff 206 31.36 1
bond_coeff 207 35.4 1
bond_coeff 208 39.69 1
bond_coeff 209 44.22 1
bond_coeff 210 49 1
bond_coeff 211 31.64 1
bond_coeff 212 36 1
bond_coeff 213 40.64 1
bond_coeff 214 45.56 1
bond_coeff 215 50.77 1
bond_coeff 216 56.25 1
bond_coeff 217 40.96 1
bond_coeff 218 46.24 1
bond_coeff 219 51.84 1
bond_coeff 220 57.76 1
bond_coeff 221 64 1
bond_coeff 222 52.2 1
bond_coeff 223 58.52 1
bond_coeff 224 65.21 1
bond_coeff 225 72.25 1
bond_coeff 226 65.61 1
bond_coeff 227 73.1 1
bond_coeff 228 81 1
bond_coeff 229 81.45 1
bond_coeff 230 90.25 1
bond_coeff 231 100 1
bond_coeff 232 0.001 1
bond_coeff 233 0.25 1
bond_coeff 234 1 1
bond_coeff 235 2.25 1

Example input for DETO_3D

```
bond_coeff 236 4 1
bond_coeff 237 6.25 1
bond_coeff 238 9 1
bond_coeff 239 12.25 1
bond_coeff 240 16 1
bond_coeff 241 20.25 1
bond_coeff 242 25 1
bond_coeff 243 30.25 1
bond_coeff 244 36 1
bond_coeff 245 42.25 1
bond_coeff 246 49 1
bond_coeff 247 56.25 1
bond_coeff 248 64 1
bond_coeff 249 72.25 1
bond_coeff 250 81 1
bond_coeff 251 90.25 1
bond_coeff 252 100 1
bond_coeff 253 100 1

create_bonds many 1 1 1 0.9 1.1
create_bonds many 1 2 2 0.9 1.1
create_bonds many 1 3 3 0.9 1.1
create_bonds many 1 4 4 0.9 1.1
create_bonds many 1 5 5 0.9 1.1
create_bonds many 1 6 6 0.9 1.1
create_bonds many 1 7 7 0.9 1.1
create_bonds many 1 8 8 0.9 1.1
create_bonds many 1 9 9 0.9 1.1
create_bonds many 1 10 10 0.9 1.1
create_bonds many 1 11 11 0.9 1.1
create_bonds many 1 12 12 0.9 1.1
```

create_bonds many 1 13 13 0.9 1.1
create_bonds many 1 14 14 0.9 1.1
create_bonds many 1 15 15 0.9 1.1
create_bonds many 1 16 16 0.9 1.1
create_bonds many 1 17 17 0.9 1.1
create_bonds many 1 18 18 0.9 1.1
create_bonds many 1 19 19 0.9 1.1
create_bonds many 1 20 20 0.9 1.1
create_bonds many 1 21 21 0.9 1.1
create_bonds many 2 2 22 0.9 1.1
create_bonds many 2 3 23 0.9 1.1
create_bonds many 2 4 24 0.9 1.1
create_bonds many 2 5 25 0.9 1.1
create_bonds many 2 6 26 0.9 1.1
create_bonds many 2 7 27 0.9 1.1
create_bonds many 2 8 28 0.9 1.1
create_bonds many 2 9 29 0.9 1.1
create_bonds many 2 10 30 0.9 1.1
create_bonds many 2 11 31 0.9 1.1
create_bonds many 2 12 32 0.9 1.1
create_bonds many 2 13 33 0.9 1.1
create_bonds many 2 14 34 0.9 1.1
create_bonds many 2 15 35 0.9 1.1
create_bonds many 2 16 36 0.9 1.1
create_bonds many 2 17 37 0.9 1.1
create_bonds many 2 18 38 0.9 1.1
create_bonds many 2 19 39 0.9 1.1
create_bonds many 2 20 40 0.9 1.1
create_bonds many 2 21 41 0.9 1.1
create_bonds many 3 3 42 0.9 1.1

Example input for DETO_3D

```
create_bonds many 3 4 43 0.9 1.1
create_bonds many 3 5 44 0.9 1.1
create_bonds many 3 6 45 0.9 1.1
create_bonds many 3 7 46 0.9 1.1
create_bonds many 3 8 47 0.9 1.1
create_bonds many 3 9 48 0.9 1.1
create_bonds many 3 10 49 0.9 1.1
create_bonds many 3 11 50 0.9 1.1
create_bonds many 3 12 51 0.9 1.1
create_bonds many 3 13 52 0.9 1.1
create_bonds many 3 14 53 0.9 1.1
create_bonds many 3 15 54 0.9 1.1
create_bonds many 3 16 55 0.9 1.1
create_bonds many 3 17 56 0.9 1.1
create_bonds many 3 18 57 0.9 1.1
create_bonds many 3 19 58 0.9 1.1
create_bonds many 3 20 59 0.9 1.1
create_bonds many 3 21 60 0.9 1.1
create_bonds many 4 4 61 0.9 1.1
create_bonds many 4 5 62 0.9 1.1
create_bonds many 4 6 63 0.9 1.1
create_bonds many 4 7 64 0.9 1.1
create_bonds many 4 8 65 0.9 1.1
create_bonds many 4 9 66 0.9 1.1
create_bonds many 4 10 67 0.9 1.1
create_bonds many 4 11 68 0.9 1.1
create_bonds many 4 12 69 0.9 1.1
create_bonds many 4 13 70 0.9 1.1
create_bonds many 4 14 71 0.9 1.1
create_bonds many 4 15 72 0.9 1.1
```

create_bonds many 4 16 73 0.9 1.1
create_bonds many 4 17 74 0.9 1.1
create_bonds many 4 18 75 0.9 1.1
create_bonds many 4 19 76 0.9 1.1
create_bonds many 4 20 77 0.9 1.1
create_bonds many 4 21 78 0.9 1.1
create_bonds many 5 5 79 0.9 1.1
create_bonds many 5 6 80 0.9 1.1
create_bonds many 5 7 81 0.9 1.1
create_bonds many 5 8 82 0.9 1.1
create_bonds many 5 9 83 0.9 1.1
create_bonds many 5 10 84 0.9 1.1
create_bonds many 5 11 85 0.9 1.1
create_bonds many 5 12 86 0.9 1.1
create_bonds many 5 13 87 0.9 1.1
create_bonds many 5 14 88 0.9 1.1
create_bonds many 5 15 89 0.9 1.1
create_bonds many 5 16 90 0.9 1.1
create_bonds many 5 17 91 0.9 1.1
create_bonds many 5 18 92 0.9 1.1
create_bonds many 5 19 93 0.9 1.1
create_bonds many 5 20 94 0.9 1.1
create_bonds many 5 21 95 0.9 1.1
create_bonds many 6 6 96 0.9 1.1
create_bonds many 6 7 97 0.9 1.1
create_bonds many 6 8 98 0.9 1.1
create_bonds many 6 9 99 0.9 1.1
create_bonds many 6 10 100 0.9 1.1
create_bonds many 6 11 101 0.9 1.1
create_bonds many 6 12 102 0.9 1.1

Example input for DETO_3D

create_bonds many 6 13 103 0.9 1.1
create_bonds many 6 14 104 0.9 1.1
create_bonds many 6 15 105 0.9 1.1
create_bonds many 6 16 106 0.9 1.1
create_bonds many 6 17 107 0.9 1.1
create_bonds many 6 18 108 0.9 1.1
create_bonds many 6 19 109 0.9 1.1
create_bonds many 6 20 110 0.9 1.1
create_bonds many 6 21 111 0.9 1.1
create_bonds many 7 7 112 0.9 1.1
create_bonds many 7 8 113 0.9 1.1
create_bonds many 7 9 114 0.9 1.1
create_bonds many 7 10 115 0.9 1.1
create_bonds many 7 11 116 0.9 1.1
create_bonds many 7 12 117 0.9 1.1
create_bonds many 7 13 118 0.9 1.1
create_bonds many 7 14 119 0.9 1.1
create_bonds many 7 15 120 0.9 1.1
create_bonds many 7 16 121 0.9 1.1
create_bonds many 7 17 122 0.9 1.1
create_bonds many 7 18 123 0.9 1.1
create_bonds many 7 19 124 0.9 1.1
create_bonds many 7 20 125 0.9 1.1
create_bonds many 7 21 126 0.9 1.1
create_bonds many 8 8 127 0.9 1.1
create_bonds many 8 9 128 0.9 1.1
create_bonds many 8 10 129 0.9 1.1
create_bonds many 8 11 130 0.9 1.1
create_bonds many 8 12 131 0.9 1.1
create_bonds many 8 13 132 0.9 1.1

create_bonds many 8 14 133 0.9 1.1
create_bonds many 8 15 134 0.9 1.1
create_bonds many 8 16 135 0.9 1.1
create_bonds many 8 17 136 0.9 1.1
create_bonds many 8 18 137 0.9 1.1
create_bonds many 8 19 138 0.9 1.1
create_bonds many 8 20 139 0.9 1.1
create_bonds many 8 21 140 0.9 1.1
create_bonds many 9 9 141 0.9 1.1
create_bonds many 9 10 142 0.9 1.1
create_bonds many 9 11 143 0.9 1.1
create_bonds many 9 12 144 0.9 1.1
create_bonds many 9 13 145 0.9 1.1
create_bonds many 9 14 146 0.9 1.1
create_bonds many 9 15 147 0.9 1.1
create_bonds many 9 16 148 0.9 1.1
create_bonds many 9 17 149 0.9 1.1
create_bonds many 9 18 150 0.9 1.1
create_bonds many 9 19 151 0.9 1.1
create_bonds many 9 20 152 0.9 1.1
create_bonds many 9 21 153 0.9 1.1
create_bonds many 10 10 154 0.9 1.1
create_bonds many 10 11 155 0.9 1.1
create_bonds many 10 12 156 0.9 1.1
create_bonds many 10 13 157 0.9 1.1
create_bonds many 10 14 158 0.9 1.1
create_bonds many 10 15 159 0.9 1.1
create_bonds many 10 16 160 0.9 1.1
create_bonds many 10 17 161 0.9 1.1
create_bonds many 10 18 162 0.9 1.1

Example input for DETO_3D

create_bonds many 10 19 163 0.9 1.1
create_bonds many 10 20 164 0.9 1.1
create_bonds many 10 21 165 0.9 1.1
create_bonds many 11 11 166 0.9 1.1
create_bonds many 11 12 167 0.9 1.1
create_bonds many 11 13 168 0.9 1.1
create_bonds many 11 14 169 0.9 1.1
create_bonds many 11 15 170 0.9 1.1
create_bonds many 11 16 171 0.9 1.1
create_bonds many 11 17 172 0.9 1.1
create_bonds many 11 18 173 0.9 1.1
create_bonds many 11 19 174 0.9 1.1
create_bonds many 11 20 175 0.9 1.1
create_bonds many 11 21 176 0.9 1.1
create_bonds many 12 12 177 0.9 1.1
create_bonds many 12 13 178 0.9 1.1
create_bonds many 12 14 179 0.9 1.1
create_bonds many 12 15 180 0.9 1.1
create_bonds many 12 16 181 0.9 1.1
create_bonds many 12 17 182 0.9 1.1
create_bonds many 12 18 183 0.9 1.1
create_bonds many 12 19 184 0.9 1.1
create_bonds many 12 20 185 0.9 1.1
create_bonds many 12 21 186 0.9 1.1
create_bonds many 13 13 187 0.9 1.1
create_bonds many 13 14 188 0.9 1.1
create_bonds many 13 15 189 0.9 1.1
create_bonds many 13 16 190 0.9 1.1
create_bonds many 13 17 191 0.9 1.1
create_bonds many 13 18 192 0.9 1.1

create_bonds	many	13	19	193	0.9	1.1
create_bonds	many	13	20	194	0.9	1.1
create_bonds	many	13	21	195	0.9	1.1
create_bonds	many	14	14	196	0.9	1.1
create_bonds	many	14	15	197	0.9	1.1
create_bonds	many	14	16	198	0.9	1.1
create_bonds	many	14	17	199	0.9	1.1
create_bonds	many	14	18	200	0.9	1.1
create_bonds	many	14	19	201	0.9	1.1
create_bonds	many	14	20	202	0.9	1.1
create_bonds	many	14	21	203	0.9	1.1
create_bonds	many	15	15	204	0.9	1.1
create_bonds	many	15	16	205	0.9	1.1
create_bonds	many	15	17	206	0.9	1.1
create_bonds	many	15	18	207	0.9	1.1
create_bonds	many	15	19	208	0.9	1.1
create_bonds	many	15	20	209	0.9	1.1
create_bonds	many	15	21	210	0.9	1.1
create_bonds	many	16	16	211	0.9	1.1
create_bonds	many	16	17	212	0.9	1.1
create_bonds	many	16	18	213	0.9	1.1
create_bonds	many	16	19	214	0.9	1.1
create_bonds	many	16	20	215	0.9	1.1
create_bonds	many	16	21	216	0.9	1.1
create_bonds	many	17	17	217	0.9	1.1
create_bonds	many	17	18	218	0.9	1.1
create_bonds	many	17	19	219	0.9	1.1
create_bonds	many	17	20	220	0.9	1.1
create_bonds	many	17	21	221	0.9	1.1
create_bonds	many	18	18	222	0.9	1.1

Example input for DETO_3D

```
create_bonds many 18 19 223 0.9 1.1
create_bonds many 18 20 224 0.9 1.1
create_bonds many 18 21 225 0.9 1.1
create_bonds many 19 19 226 0.9 1.1
create_bonds many 19 20 227 0.9 1.1
create_bonds many 19 21 228 0.9 1.1
create_bonds many 20 20 229 0.9 1.1
create_bonds many 20 21 230 0.9 1.1
create_bonds many 21 21 231 0.9 1.1
create_bonds many 1 22 232 0.9 1.1
create_bonds many 2 22 233 0.9 1.1
create_bonds many 3 22 234 0.9 1.1
create_bonds many 4 22 235 0.9 1.1
create_bonds many 5 22 236 0.9 1.1
create_bonds many 6 22 237 0.9 1.1
create_bonds many 7 22 238 0.9 1.1
create_bonds many 8 22 239 0.9 1.1
create_bonds many 9 22 240 0.9 1.1
create_bonds many 10 22 241 0.9 1.1
create_bonds many 11 22 242 0.9 1.1
create_bonds many 12 22 243 0.9 1.1
create_bonds many 13 22 244 0.9 1.1
create_bonds many 14 22 245 0.9 1.1
create_bonds many 15 22 246 0.9 1.1
create_bonds many 16 22 247 0.9 1.1
create_bonds many 17 22 248 0.9 1.1
create_bonds many 18 22 249 0.9 1.1
create_bonds many 19 22 250 0.9 1.1
create_bonds many 20 22 251 0.9 1.1
create_bonds many 21 22 252 0.9 1.1
```

create_bonds many 22 22 253 0.9 1.1

group 1 delete

group 2 delete

group 3 delete

group 4 delete

group 5 delete

group 6 delete

group 7 delete

group 8 delete

group 9 delete

group 10 delete

group 11 delete

group 12 delete

group 13 delete

group 14 delete

group 15 delete

group 16 delete

group 17 delete

group 18 delete

group 19 delete

group 20 delete

group 21 delete

#22 particle types

#253 bonds created

References

- Aage, N., Andreassen, E., Lazarov, B., and Sigmund, O. (2017). Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674):84–86.
- Alex, A. and Masoero, E. (2022). Autogenous healing in cement: A kinetic monte carlo simulation of CaCO_3 precipitation. In *Computational Modelling of Concrete and Concrete Structures*, pages 102–106. CRC Press.
- Alexandersen, J., Aage, N., Schousboe, C., and Sigmund, O. (2014). Topology optimisation for natural convection problems. *International Journal for Numerical Methods in Fluids*, 76(September 17 2014):699–721.
- Allaire, G. and Kohn, R. (1993). Topology optimization and optimal shape design using homogenization. In *Topology design of structures*, pages 207–218. Springer.
- Bendsøe, M. and Sigmund, O. (1999). Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69:635–654.
- Bendsoe, M. P. and Sigmund, O. (2013). *Topology optimization: theory, methods, and applications*. Springer Science & Business Media.
- Bendsøe, M. P. and Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197 – 224.
- Bonet, J. and Wood, R. D. (2008). *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press.
- Bourdin, B. and Chambolle, A. (2003). Design-dependent loads in topology optimization. *ESAIM: Control, Optimisation and Calculus of Variations*, 9:19–48.
- Bruns, T. E. and Tortorelli, D. A. (2001). Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26):3443–3459.
- Buhl, T., Pedersen, C. B., and Sigmund, O. (2000). Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization*, 19(2):93–104.
- Cao, Y., Li, S., Petzold, L., and Serban, R. (2003). Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089.
- Carmona, H. A., Wittel, F. K., and Kun, F. (2014). From fracture to fragmentation: Discrete element modeling. *The European Physical Journal Special Topics*, 223(11):2369–2382.
- Challis, V. J. (2010). A discrete level-set topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 41(3):453–464.
- Challis, V. J., Roberts, A. P., and Wilkins, A. H. (2007). Fracture resistance via topology optimization. *Structural and Multidisciplinary Optimization*, 36(3):263–271.

References

- Cheng, Y. P., Nakata, Y., and Bolton, M. D. (2003). Discrete element simulation of crushable soil. *Géotechnique*, 53(7):633–641.
- Cundall, P. A. and Strack, O. D. L. (1979). A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47–65.
- Da, D. and Yvonnet, J. (2020). Topology optimization for maximizing the fracture resistance of periodic quasi-brittle composites structures. *Materials*, 13(15).
- Daynes, S. and Feih, S. (2022). Bio-inspired lattice structure optimisation with strain trajectory aligned trusses. *Materials & Design*, 213:110320.
- Díaz, A. and Sigmund, O. (1995). Checkerboard patterns in layout optimization. *Structural Optimization*, 10(1):40–45.
- Dorn, W. S., Gomory, R. E., and Greenberg, H. J. (1964). Automatic design of optimal structures. In *Automatic design of optimal structures*.
- Duysinx, P. and Bendsøe, M. P. (1998). Topology optimization of continuum structures with local stress constraints. *International Journal for Numerical Methods in Engineering*, 43(8):1453–1478.
- Eibl, J. (1987). Soft and hard impact. In *Proceedings of the First International Conference on Concrete for Hazard Protection, Edinburgh*.
- Frenkel, D. and Smit, B. (2001). *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier.
- Ghosh, P. and Ananthasuresh, G. K. (2020). Discrete element modeling of cantilever beams subjected to geometric nonlinearity and particle–structure interaction. *Computational Particle Mechanics*.
- Gilbert, M. and Tyas, A. (2003). Layout optimization of large-scale pin-jointed frames. *Engineering Computations*, 20(8):1044.
- Gong, S.-g., Wei, Y.-b., Xie, G.-l., and Zhang, J.-p. (2018). Study on Topology Optimization Method of Particle Moving Based on Element-Free Galerkin Method. *International Journal of Computational Methods in Engineering Science and Mechanics*, 19(5):305–313.
- Gupta, A., Chowdhury, R., Chakrabarti, A., and Rabczuk, T. (2020). A 55-line code for large-scale parallel topology optimization in 2D and 3D. *arXiv preprint arXiv:2012.08208*.
- Hare, C., Ghadiri, M., and Dennehy, R. (2011). Prediction of attrition in agitated particle beds. *Chemical Engineering Science*, 66(20):4757–4770.
- Hare, C., Zafar, U., Ghadiri, M., Freeman, T., Clayton, J., and Murtagh, M. (2015). Analysis of the dynamics of the ft4 powder rheometer. *Powder Technology*, 285:123–127. Pharmaceutical Particle Technology.
- Hassani, B. and Hinton, E. (1999). Structural topology optimization using optimality criteria methods. In *Homogenization and Structural Topology Optimization*, pages 71–101. Springer.
- Hassani, B. and Hinton, E. (2012). *Homogenization and structural topology optimization: theory, practice and software*. Springer Science & Business Media.
- Herfelt, M. A., Poulsen, P. N., and Hoang, L. C. (2018). Strength-based topology optimisation of plastic isotropic von mises materials. *Structural and Multidisciplinary Optimization*, 59(3):893–906.
- Hertz, H. (1882). On the contact of rigid elastic solids and on hardness. *Ch 6: Assorted Papers*.

- Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–435.
- Jansen, M., Lombaert, G., Schevenels, M., and Sigmund, O. (2014). Topology optimization of fail-safe structures using a simplified local damage model. *Structural and Multidisciplinary Optimization*, 49(4):657–666.
- Jung, D. and Gea, H. C. (2004). Topology optimization of nonlinear structures. *Finite Elements in Analysis and Design*, 40(11):1417–1427.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4.
- Kingman, J., Tsavdaridis, K., and Toropov, V. (2015). Applications of topology optimisation in structural engineering: High-rise buildings & steel components. *Jordan Journal of Civil Engineering*, 9.
- Krijgsman, D. and Luding, S. (2016). Simulating granular materials by energy minimization. *Computational Particle Mechanics*, 3(4):463–475.
- Kun, F. and Herrmann, H. J. (1996). A study of fragmentation processes using a discrete element method. *Computer Methods in Applied Mechanics and Engineering*, 138(1-4):3–18.
- Li, L., Zhang, G., and Khandelwal, K. (2017). Topology optimization of energy absorbing structures with maximum damage constraint. *International Journal for Numerical Methods in Engineering*, 112(7):737–775.
- Li, Y., Yang, Q., Chang, T., Qin, T., and Wu, F. (2020). Multi-load cases topological optimization by weighted sum method based on load case severity degree and ideality. *Advances in Mechanical Engineering*, 12(8):1687814020947510.
- Liu, J., Gaynor, A. T., Chen, S., Kang, Z., Suresh, K., Takezawa, A., Li, L., Kato, J., Tang, J., Wang, C. C., et al. (2018). Current and future trends in topology optimization for additive manufacturing. *Structural and Multidisciplinary Optimization*, 57(6):2457–2483.
- Liu, S. and Qiao, H. (2011). Topology optimization of continuum structures with different tensile and compressive properties in bridge layout design. *Structural and Multidisciplinary Optimization*, 43:369–380.
- Luh, G.-C. and Lin, C.-Y. (2009). Structural topology optimization using ant colony optimization algorithm. *Applied Soft Computing*, 9(4):1343 – 1353.
- Luo, Y., Wang, M. Y., and Kang, Z. (2015). Topology optimization of geometrically nonlinear structures based on an additive hyperelasticity technique. *Computer Methods in Applied Mechanics and Engineering*, 286:422 – 441.
- Lógó, J. and Ismail, H. (2020). Milestones in the 150-year history of topology optimization: a review. *Computer Assisted Methods in Engineering and Science*, 27(2–3):97–132.
- Magnier, S. A. and Donzé, F.-V. (1998). Numerical simulations of impacts using a discrete element method. *Mechanics of Cohesive-frictional Materials: An International Journal on Experiments, Modelling and Computation of Materials and Structures*, 3(3):257–276.
- Marshall, J. (2009). Discrete-element modeling of particulate aerosol flows. *Journal of Computational Physics*, 228(5):1541–1561.
- Masoero, E., Jennings, H., Ulm, F., Del Gado, E., Manzano, H., Pellenq, R., and Yip, S. (2014). Modelling cement at fundamental scales: From atoms to engineering strength and durability. *Comput. Model. Concr. Struct*, 1:139–148.

References

- Masoero, E., O'Shaughnessy, C., Gosling, P. D., and Chiaia, B. (2021). Topology optimization using the discrete element method. part 2: Material nonlinearity. *Meccanica*.
- Masoero, E., Wittel, F., Herrmann, H., and Chiaia, B. (2010). Progressive collapse mechanisms of brittle and ductile framed structures. *Journal of Engineering Mechanics-asce - J ENG MECH-ASCE*, 136.
- Masoero, E., Wittel, F. K., Herrmann, H. J., and Chiaia, B. M. (2012). Hierarchical structures for a robustness-oriented capacity design. *Journal of Engineering Mechanics*, 138(11):1339–1347.
- Maute, K., Schwarz, S., and Ramm, E. (1998a). Adaptive topology optimization of elastoplastic structures. *Structural optimization*, 15(2):81–91.
- Maute, K., Schwarz, S., and Ramm, E. (1998b). Adaptive topology optimization of elastoplastic structures. *Structural optimization*, 15(2):81–91.
- McDowell, G. R. and Harireche, O. (2002). Discrete element modelling of soil particle fracture. *Géotechnique*, 52(2):131–135.
- Metzger, M. J. and Glasser, B. J. (2013). Simulation of the breakage of bonded agglomerates in a ball mill. *Powder Technology*, 237:286–302.
- Michell, A. (1904). Lviii. the limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(47):589–597.
- Mindlin, R. D. (2021). Compliance of Elastic Bodies in Contact. *Journal of Applied Mechanics*, 16(3):259–268.
- Mirzendehtel, A. M., Rankouhi, B., and Suresh, K. (2018). Strength-based topology optimization for anisotropic parts. *Additive manufacturing*, 19:104–113.
- Mitchell, M. (1998). *An introduction to genetic algorithms*. Complex adaptive systems. MIT, London.
- Nakshatrala, P. and Tortorelli, D. (2015). Topology optimization for effective energy propagation in rate-independent elastoplastic material systems. *Computer Methods in Applied Mechanics and Engineering*, 295:305–326.
- Nan, W., Ghadiri, M., and Wang, Y. (2017a). Analysis of powder rheometry of ft4: Effect of air flow. *Chemical Engineering Science*, 162:141–151.
- Nan, W., Ghadiri, M., and Wang, Y. (2017b). Analysis of powder rheometry of ft4: Effect of particle shape. *Chemical Engineering Science*, 173:374–383.
- Nan, W., Pasha, M., Bonakdar, T., Lopez, A., Zafar, U., Nadimi, S., and Ghadiri, M. (2018). Jamming during particle spreading in additive manufacturing. *Powder Technology*, 338:253 – 262.
- Ohmori, H. (2011). Computational morphogenesis: Its current state and possibility for the future. *International Journal of Space Structures*, 26(3):269–276.
- O'Sullivan, C. (2011). *Particulate discrete element modelling: a geomechanics perspective*. CRC Press.
- O'Shaughnessy, C., Masoero, E., and Gosling, P. D. (2021). Topology optimization using the discrete element method. part 1: Methodology, validation, and geometric nonlinearity. *Meccanica*.
- Patwa, A., Ambrose, R. K., and Casada, M. (2016). Discrete element method as an approach to model the wheat milling process. *Powder Technology*, 302:350–356.

- Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1):1–19.
- Plocher, J. and Panesar, A. (2019). Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures. *Materials & Design*, 183:108164.
- Pöschel, T. and Schwager, T. (2005). *Computational granular dynamics: models and algorithms*. Springer Science & Business Media.
- Prager, W. and Rozvany, G. (1977). Optimal layout of grillages. *Journal of Structural Mechanics*, 5(1):1–18.
- Querin, O., Steven, G., and Xie, Y. (1998). Evolutionary structural optimisation (eso) using a bidirectional algorithm. *Engineering Computations*, 15:1031–1048.
- Rossow, M. P. and Taylor, J. E. (1973). A finite element method for the optimal design of variable thickness sheets. *AIAA Journal*, 11(11):1566–1569.
- Rozvany, G. (1989). *Structural design via optimality criteria*, volume 8. Springer Science & Business Media.
- Rozvany, G. (2009). A critical review of established methods of structural topology optimization. structural and multidisciplinary optimization 37, 217-237. *Structural and Multidisciplinary Optimization*, 37:217–237.
- Rozvany, G. (2012). *Structural design via optimality criteria: the Prager approach to structural optimization*, volume 8. Springer Science & Business Media.
- Rozvany, G. and Zhou, M. (1991). Applications of the coc algorithm in layout optimization. In *Engineering optimization in design processes*, pages 59–70. Springer.
- Ryu, Y., Haririan, M., Wu, C., and Arora, J. (1985). Structural design sensitivity analysis of nonlinear response. *Computers & structures*, 21(1-2):245–255.
- Schwarz, S., Maute, K., and Ramm, E. (2001). Topology and shape optimization for elastoplastic structural response. *Computer methods in applied mechanics and engineering*, 190(15-17):2135–2155.
- Sheppard, D., Terrell, R., and Henkelman, G. (2008). Optimization methods for finding minimum energy paths. *Journal of Chemical Physics*, 128(13).
- Sigmund, O. (2001). A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127.
- Sigmund, O. (2011). On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization*, 43:589–596.
- Sigmund, O. and Maute, K. (2013). Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055.
- Sigmund, O. and Petersson, J. (1998). Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, 16(1):68–75.
- Silbert, L. E., Ertaş, D., Grest, G. S., Halsey, T. C., Levine, D., and Plimpton, S. J. (2001). Granular flow down an inclined plane: Bagnold scaling and rheology. *Phys. Rev. E*, 64:051302.
- Silva, O. M., Neves, M. M., and Lenzi, A. (2019). A critical analysis of using the dynamic compliance as objective function in topology optimization of one-material structures considering steady-state forced vibration problems. *Journal of Sound and Vibration*, 444:1–20.

- Stukowski, A. (2010). Visualization and analysis of atomistic simulation data with OVITO—the Open Visualization Tool. *MODELLING AND SIMULATION IN MATERIALS SCIENCE AND ENGINEERING*, 18(1).
- Svanberg, K. (1987). The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373.
- Svanberg, K. and Werme, M. (2006). Topology optimization by sequential integer linear programming. In Bendsøe, M. P., Olhoff, N., and Sigmund, O., editors, *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, pages 425–436, Dordrecht. Springer Netherlands.
- Tardos, G. I., McNamara, S., and Talu, I. (2003). Slow and intermediate flow of a frictional bulk powder in the couette geometry. *Powder Technology*, 131(1):23–39.
- Tavarez, F. A. and Plesha, M. E. (2007). Discrete element method for modelling solid and particulate materials. *International Journal for Numerical Methods in Engineering*, 70(4):379–404.
- Thompson, A. P., Aktulga, H. M., Berger, R., Bolintineanu, D. S., Brown, W. M., Crozier, P. S., in 't Veld, P. J., Kohlmeyer, A., Moore, S. G., Nguyen, T. D., Shan, R., Stevens, M. J., Tranchida, J., Trott, C., and Plimpton, S. J. (2022). LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271:108171.
- Thompson, A. P., Plimpton, S. J., and Mattson, W. (2009). General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions. *The Journal of Chemical Physics*, 131(15):154107.
- Thornton, C. (2015). *Granular Dynamics, Contact Mechanics and Particle System Simulations*. Springer.
- Tsay, J. and Arora, J. (1989). Optimum design of nonlinear structures with path dependent response. *Structural optimization*, 1(4):203–213.
- Valdez, S. I., Botello, S., Ochoa, M. A., Marroquin, J. L., and Cardoso, V. (2017). Topology optimization benchmarks in 2d: Results for minimum compliance and minimum volume in planar stress problems. *Archives of Computational Methods in Engineering*, 24(4):803–839.
- van Laarhoven, P. J. M. and Aarts, E. H. L. (1987). *Simulated annealing*, pages 7–15. Springer Netherlands, Dordrecht.
- Wang, M. Y., Wang, X., and Guo, D. (2003). A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1):227 – 246.
- Wang, Y., Alonso-Marroquin, F., and Guo, W. W. (2015). Rolling and sliding in 3-d discrete element models. *Particuology*, 23:49–55.
- Wittel, F. K., Kun, F., Kröplin, B.-H., and Herrmann, H. J. (2003). A study of transverse ply cracking using a discrete element method. *Computational materials science*, 28(3-4):608–619.
- Wu, C., Gao, Y., Fang, J., Lund, E., and Li, Q. (2017). Discrete topology optimization of ply orientation for a carbon fiber reinforced plastic (cfrp) laminate vehicle door. *Materials & Design*, 128:9–19.
- Wu, J., Aage, N., Westermann, R., and Sigmund, O. (2018). Infill Optimization for Additive Manufacturing—Approaching Bone-Like Porous Structures. *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1127–1140.

- Wu, Y., Yvonnet, J., Li, P., and He, Z.-C. (2022). Topology optimization for enhanced dynamic fracture resistance of structures. *Computer Methods in Applied Mechanics and Engineering*, 394:114846.
- Xia, L., Da, D., and Yvonnet, J. (2018). Topology optimization for maximizing the fracture resistance of quasi-brittle composites. *Computer Methods in Applied Mechanics and Engineering*, 332:234–254.
- Xie, Y. and Steven, G. (1993). A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49(5):885–896.
- Yago, D., Cante, J., and Lloberas-Valls, O. (2022). Topology optimization methods for 3d structural problems: A comparative study. *Arch Computat Methods Eng*, 12:1525–1567.
- Ye, J. and Xu, L. (2017). Member Discrete Element Method for static and dynamic responses analysis of steel frames with semi-rigid joints. *Applied Sciences (Switzerland)*, 7(7).
- Zegard, T., Hartz, C., Mazurek, A., and Baker, W. (2020). Advancing building engineering through structural and topology optimization. *Structural and Multidisciplinary Optimization*, 62:1–21.
- Zhang, J., Wang, S., Zhou, G., Gong, S., and Yin, S. (2020). Topology optimization of thermal structure for isotropic and anisotropic materials using the element-free galerkin method. *Engineering Optimization*, 52(7):1097–1118.
- Zhou, M. and Fleury, R. (2016). Fail-safe topology optimization. *Structural and Multidisciplinary Optimization*, 54(5):1225–1243.
- Zhou, M. and Rozvany, G. (1992). Dcoc: an optimality criteria method for large systems part i: theory. *Structural optimization*, 5(1):12–25.
- Zhou, M. and Rozvany, G. (1993). Dcoc: an optimality criteria method for large systems part ii: algorithm. *Structural optimization*, 6(4):250–262.
- Zienkiewicz, O., Taylor, R., and Taylor, R. (2000). *The Finite Element Method: The basis*. Fluid Dynamics. Butterworth-Heinemann.