Computational modeling and design of mechanical metamaterials: A machine learning approach

Tianju Xue

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Civil and Environmental Engineering
Advisers:
Professor Sigrid Adriaenssens
Professor Ryan P. Adams

April 2022

Abstract

Mechanical metamaterials are a special class of materials, whose mechanical properties are primarily determined by their geometry and topology. Due to their unique properties and wide applications, mechanical metamaterials have gained increasing attention in recent years. To name a few, mechanical metamaterials have been used in designing soft robotics, equipment with local tunable functionalities, etc. For better exploitation of their huge potential, the ability to design mechanical metamaterials with particular desired properties is of key importance. Traditional design methods rely heavily on experimental characterization and are often driven by heuristic rules, which are time consuming and economically inefficient. This dissertation aims to establish a computational design framework that enables rational, efficient, and robust designs for mechanical metamaterials. The challenge is addressed by applying classical numerical methods and leveraging modern machine learning tools.

The research results can be grouped with three major outcomes. First, we propose a multi-scale computational homogenization scheme based on a neural network surrogate energy model to simulate cellular mechanical metamaterials under large deformation. Compared with direct numerical simulation, the proposed scheme reduces the computational cost up to two orders of magnitude. The second part focuses on inverse design problems. Within the framework of topology and shape optimization, we successfully design cellular mechanical metamaterials with several pre-defined goals: achieving negative Poisson's ratio, precise control of instabilities, and arbitrary tuning of band gaps for phononic structures. Besides classical shape optimization methods, we also propose design approaches inspired by generative models in machine learning. Composite mechanical metamaterials with controllable overall elastic moduli are designed, fabricated with additive manufacturing, and experimentally validated. The final part of this dissertation makes a mathematical abstraction of the design/inverse problems and focuses on general partial differential equation (PDE) constrained optimization problems. We propose amortized finite element analysis (AmorFEA), in which a neural network is trained to produce accurate PDE solutions in an unsupervised fashion, while preserving many of the advantages of the traditional finite element method (FEM). In conclusion, this dissertation successfully tackles issues on computational modeling and design of mechanical metamaterials with a machine learning approach.

Acknowledgements

My first day at Princeton in fall 2017 is still vivid like yesterday in my mind. There are many people I would like to thank for making this dissertation possible.

I want thank my first advisor Dr. Maurizio Chiaramonte, whose guidance, encouragement and patience deeply inspired me. Being a rigorous researcher, he taught me the esssentials to open the door of research in the field of computational mechanics. I feel very grateful to have him as my advisor at the beginning of my PhD journey.

In pursuit of an alternative career, Dr. Chiaramonte left Princeton in fall 2018. I was fortunate to meet Prof. Sigrid Adriaenssens and Prof. Ryan P. Adams, who later became my advisors. I would like to express my sincere gratitude to Prof. Adriaenssens for her professional guidance in the research of mechanical metamaterials and her constant support on both my career and personal life. I want to thank Prof. Adams for his sharp thinking and novel insights in the field of machine learning and his broad interest in interdisciplinary research, which greatly influenced the way I think about research questions. I always strive to learn more from him.

I would also like to thank Prof. Branko Glisic who has been in my research committee since my general exam. My sincere thanks go to him for his generous offering of time and careful attention at each step of my progress.

I would like to thank my excellent collaborators Thomas J. Wallin, Tianshu liu, and Yigit Menguc at Facebook, where I had an enjoyable internship in summer 2019. During the coronavirus pandemic, I had the pleasure to work with Chuanqi Liu and Sheng Mao, who were former postdocs at Princeton University. The shared interest and research collaborations led to a fruitful outcome of several publications.

Many thanks go to my colleagues Roberto Porcu, Vivek Kumar, Zeyu Xiong, and Shashank Kumar Anand from the Computational Mechanics Group. In the Form Finding Lab, I would like to thank Olek Nowakowski, Edvard Bruun, Victor Charpentier, Jessica Flores, and Isabel Moreira de Oliveira. I also want to thank my amazing colleagues from the Laboratory for Intelligent Probabilistic Systems: Alex Beatson, Deniz Oktay, Xingyuan Sun, Sulin Liu, Geoffrey Roeder, Jad Rahme, Joshua Aduol, Sam Barnett, Diana Cai, Yaniv Ovadia, Ari Seff, and David Zoltowski.

I am fortunate to have many great friends over the years, making my experience at Princeton so memorable. Among them, I would like to specially thank Hongtao Zhong, Dazhi Xi, Xiaoyu Song, Ruoyao Zhang, Jiarong Wu, and Nathan Li.

I want to thank my parents for their unconditional love. Finally, I would like to thank my wife Lu Shen for being with me since high school. An international long-distance relationship is never easy, but her constant love and support have made the best part of my every day.

To my family.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and objectives

Mechanical metamaterials are artificial structures whose mechanical behavior is determined defined by their geometry rather than their chemical composition. Architected with multi-level periodic structures, mechanical metamaterials demonstrate unique macroscopic properties such as ultra-stiffness [1], negative Poisson's ratio (auxeticity) [2], shape morphing [3], tunable band structures [4], energy absorption [5], etc. These novel properties have created venues for many engineering applications, e.g., soft actuators, materials with *in situ* tunable functionalities, reusable energy-absorbing materials [6, 7, 8, 9]. For a thorough overview of mechanical metamaterials and the current state of the art, we refer the interested readers to [10, 11, 12]. The rapid development of this field has been partly fueled by the advancement of advanced manufacturing technologies such as additive manufacturing [13, 14], which enables fast prototyping and economic small-scale production of materials with specific microstructures [15]. The key step that can bridge the prospect of mechanical metamaterials to real-world engineering applications is the ability to design the desired type of mechanical metamaterials for particular purposes. Traditional design flows focus heavily on experimental characterization and heuristically driven approaches. Optimal structures of mechanical metamaterials are often devised through inefficient trial-and-error loops. This dissertation addresses this problem by utilizing computational tools to build a rational design framework based on efficient numerical modeling and optimization. When establishing the computational framework, we pay particular attention to the application of machine learning techniques.

Before considering the inverse design problems, our first objective is to establish an efficient and robust method for forward numerical simulation of mechanical metamaterials. While adopting the finite element method (FEM) [16, 17] for solid structures are standard practices, direct numerical simulation (DNS) with FEM on the full specimen is only possible for mechanical metamaterials with a relatively small number of repeated cells [18]. The reason is partly due to the high computational cost on super fine mesh required for the resolution of the microstructure. Under extreme deformation conditions, mechanical metamaterials exhibit highly nonlinear responses and instabilities are triggered over certain threshold [19, 20], bringing additional difficulties for accurate simulation. Multi-scale analysis tools such as homogenization [21, 22] give promising solutions to the problem.

Homogenization, also known as “coarse-graining” in the physics community, is often used to predict the macroscopic behavior of composites from their microstructures. Homogenization focuses on calculations of a representative volume element (RVE) at the micro scale by FEM, and then use the obtained average constitutive relation to perform a coarsened simulation the macro scale. Aside from analytical homogenization methods [23, 24, 25, 26], computational homogenization is more suitable for our problem due to the complexity of mechanical metamaterials. There are two major categories of computational homogenization methods: *concurrent* and *off-line* methods. Concurrent methods integrate the macro- and micro-scale problems through proper mathematical formulations and solve both problems simultaneously. One of the representative concurrent methods is the widely adopted FE^2 method [27]. Concurrent methods are powerful, but they can be computationally expensive, due to the nested numerical solvers at both microscopic and macroscopic scales. On the other hand, off-line methods treat the macro- and micro-scale problems separately. Equivalent constitutive laws are first computed based on off-line numerical calculations at micro-scales, and then used to solve problems at macro-scales [28, 29]. Offline methods can be expensive during the data generation phase, but they are typically significantly faster at the deployment phase [30]. Several techniques [31, 32, 33, 34, 30, 29, 35] exist, particularly we emphasize recent developments using neural networks (NNs) [36, 37].

To this point, we propose the neural networks based computational homogenization method to model the nonlinear behavior of mechanical metamaterials under large deformation. Neural networks, as massively-parametric function approximators, have been proven to be effective for solving a variety of traditional engineering problems. More recently, there has been a growing interest in applying neural networks to complicated problems in natural sciences, e.g., physics and chemistry [38, 39, 40, 41], astronomy [42], and material science [43, 44, 45]. In this work, we train a NN-based surrogate model to represent the macroscopic strain energy density function. The training data are generated by performing RVE calculations at the micro scale. At the deployment stage, the trained surrogate model is used for numerical simulation at the macro scale. With the proposed NN-based off-line homogenization scheme, we are able to save the the computational time to several orders of magnitude when compared to direct finite element simulation. This concludes the first major task of this dissertation, i.e., a new NN-based homogenization scheme for accelerating the (traditionally) expensive forward simulation of mechanical metamaterials.

The next part focuses on computational designs of mechanical metamaterials. The objective is to use shape optimization techniques for the optimization of geometric structures of mechanical metamaterials so that desired mechanical properties are obtained. Shape optimization is a rich and classical field of research. Representative techniques, such as density based methods [46, 47], evolutionary structural optimization (ESO) methods [48], and level set methods (LSM) [49, 50, 51] have been developed and broadly applied to typical mechanical design problems such as stiffness optimization of linear/nonlinear elastic structures, optimization of the heat-conductivity of composite structures, etc. However, these methods may not necessarily be best suited for shape optimization of mechanical metamaterials due to the intrinsic geometric and material nonlinearities. Moreover, under large deformation conditions, instabilities at both micro and macro scales pose additional challenges to certain numerical schemes. Despite the complexity, several related works have made positive attempts to metamaterial shape optimization with large deformations [52, 53, 54].

In general accordance with the shape optimization pipeline, we propose a mapped shape optimization method (MSOM), a numerically stable and easy-to-implement method that solves the design problems for mechanical metamaterials efficiently. The MSOM uses gradient-based optimization and the gradients of the objective function to the geometric parameters are computed using the adjoint method [55, 56]. The highlight of MSOM is the use of a pre-defined referential configuration a concept similar to what appears in Arbitrary Lagrangian–Eulerian (ALE) methods [57]. In the ALE description of motion, neither the material configuration nor the spatial configuration is taken as the reference, while a third, referential configuration is used for computations. In MSOM, we also have such a referential configuration where all numerical computations are performed. The choice of this referential configuration is arbitrary, and we fix it. Therefore, the computational mesh can stay the same throughout the optimization loops. Compared with classical shape optimization methods, MSOM has several advantages when applied to the optimization of CMMs. MSOM is not a density-based method, hence to avoid dealing with inherent difficulties like the "checkerboard" problem [58]. MSOM shares certain similarity with the level set method in the sense that they both rely on the concept of "shape derivative" [59]. The dichotomy of Eulerian and Lagrangian specifications divides the level set methods into two classes: the Eulerian shape capturing method with a fixed mesh [49, 50, 60]; the Lagrangian shape tracking method with a moving mesh to represent the shape [61, 62]. MSOM can be considered as a third paradigm that uses a referential description. In contrast to the Eulerian level set method, MSOM avoids the difficulties implied by stable and accurate evolutions of the level set function; compared to the Lagrangian level set method, MSOM saves the numerical complications of modifying the mesh object since it uses a fixed mesh. We apply MSOM to three important design problems for mechanical metamaterials: optimizing negative Poisson's ratio, precise control of the onset of instability, and arbitrary tuning of band gaps for phononic structures. These examples cover a broad range of engineering applications, showing that the proposed method is successful in shape optimization for mechanical metamaterials. The contribution of this part particularly lies in its ability to solve shape optimization problems with large deformation.

Aside from classical shape optimization methods, the recent trends in using machine learning techniques for design and synthesis [41, 63, 64, 65] have also opened up new opportunities for optimal design of mechanical metamaterials. In this next part, we introduce a design method based on machine learning generative models. The objective is to design composite mechanical metamaterials with arbitrary macroscopic elastic moduli. The microscopic RVE is composed of two different base materials, and the homogenization is restricted to linear elasticity. Energy based computational homogenization scheme [66] is employed for the evaluation of the macroscopic elastic moduli. It is noticed that *parametric representation* of RVE structures is key to the success of design. Density based shape optimization method is computationally expensive due to voxel level parametrization [67]. Pre-defined parametric family of microstructures are used in [68] as the template for design. Nevertheless, these traditional methods still follow the pattern of trial-and-error loops that require substantial domain expertise and intensive labor. The proposed method automates the design flow using several tools from machine learning.

The successful applications of machine learning techniques in the area of structure and material optimization are seen in [69, 70, 71]. Particularly, a new paradigm emerges for design optimization with parameters of complicated data structure. The paradigm compress the high dimensional design parameters into a lower dimensional latent space, and designs are performed over this reduced latent space instead. The approach typically involves coupling of a machine learning generative model that maps from the latent space to the parameter space and a gradient-free optimization algorithm that operates over the latent space. This design strategy has been used to a variety of applications: discovery of new molecules with desired properties [41], shape optimization for minimal drag force in fluid dynamics [72], finding optimal optical performance of composite materials [73], improvement in solution efficiency for heat conduction [74], etc. In this work, we propose to combine variational auto-encoders (VAEs) [75] and Bayesian optimization (BayesOpt) [76] so that the procedures of designing composite mechanical metamaterials are automated. VAEs are used to encode the RVE images into the latent space, and then decode the latent vector to an output that resembles the original RVE. BayesOpt perform optimization over the latent space so that the design goal is achieved. We present several design examples where optimal RVEs are found to have desired macroscopic elastic moduli. The RVEs are printed with customized multi-material 3D printing tools [77]. Experimental validations show that the proposed method scheme is effective for the design.

Mathematically, solving the forward simulation problem of mechanical metamaterials is considered as numerically solving the governing partial differential equation (PDE). PDE-constrained optimization (PDE-CO) addresses the situation where an objective function is minimized or maximized, subject to the constraints by PDEs.

In fact, the design problems of mechanical metamaterials can be formulated into PDE-constrained optimization. In the final part of this dissertation, we make this mathematical abstraction and focus on general PDE-CO problems. In addition to designs of mechanical metamaterials, PDE-CO problems also include optimal control, parameter identification, etc [78, 79]. PDE-CO is computationally expensive in general since it usually requires to solve the governing PDE at every iteration step of the outer-loop optimization.

In recent years, there has been a surge of interest in applying deep learning to solving PDEs in replacement of traditional numerical methods [80, 81, 82, 83]. However, limited efforts are spent on the (more difficult) PDE-CO problems. Inspired by *amortized optimization*, widely used in amortized variational inference [75, 84, 85, 86], we propose a deep learning method to efficiently tackle PDE-CO problems, which we refer to as amortized finite element analysis (AmorFEA). AmorFEA introduces a two-stage strategy such that a neural network is first trained to map from parameters to solutions, and then fast PDE-CO can be performed based on the learned model. AmorFEA supports irregular computational domains as an advantage inherited from classical FEM. Compared with the traditional adjoint method for PDE-CO, AmorFEA demonstrates huge potential of saving computational cost.

As a final remark, the major research objectives in this dissertation are two-fold. We aim at establishing new homogenization schemes for accelerating the forward numerical simulations of mechanical metamaterials. The second goal is to tackle the inverse design problems with machine learning approaches.

1.2 Outline of Thesis

The overview of this dissertation is as follows.

Chapter 1 gives the background and objectives of the research, the outline of the dissertation, and some useful notations. Detailed literature review is delayed in the following separate chapters to ease reading and make those chapters readable on their individual basis.

Chapter 2 discusses the forward simulation problem. A neural network based computational homogenization scheme is proposed for efficient numerical simulations of cellular mechanical metamaterials. This part is based on T. Xue, A. Beatson, M. Chiaramonte, G. Roeder, J. T. Ash, Y. Menguc, S. Adriaenssens, R. P. Adams, and S. Mao, "A data-driven computational scheme for the nonlinear mechanical properties of cellular mechanical metamaterials under large deformation," *Soft Matter*, 2020.

Chapter 3 focuses on the inverse design problem of the same cellular mechanical metamaterials. We use the adjoint method for optimal designs with several important engineering applications. The results are in preparation for submission.

Chapter 4 also addresses design problems, but with machine learning aided optimization methods applied to composite mechanical metamaterials. The chapter is based on T. Xue, T. J. Wallin, Y. Menguc, S. Adriaenssens, and M. Chiaramonte, "Machine learning generative models for automatic design of multi-material 3d printed composite solids," *Extreme Mechanics Letters*, vol. 41, p. 100992, 2020.

Chapter 5 abstracts the mathematical nature from design problems and focus on general partial differential equation constrained optimization. We propose amortized finite element analysis for accelerated solutions. The part is based on T. Xue, A. Beatson, S. Adriaenssens, and R. Adams, "Amortized finite element analysis for fast pde-constrained optimization," in *International Conference on Machine Learning*, pp. 10638–10647, PMLR, 2020.

Chapter 6 summarizes the contents with concluding remarks. It also discusses some visions for future work.

1.3 Notations

We conclude the introduction with some remarks on notation. Scalars are generally written in normal fonts, while vectors and second-order tensors are written in boldface fonts. All tensor and vector components are written with respect to a fixed Cartesian coordinate system with orthonormal basis $\{\boldsymbol{e}_i\}$. The summation convention is used for repeated Latin indices, unless otherwise indicated. We denote by $\boldsymbol{I}$ the second-order identity tensor. The prefixes tr and det indicate the trace and the determinant and superscript $\top$ the transpose of a second-order tensor. Let ($\boldsymbol{a}$, $\boldsymbol{b}$) be vectors, ($\boldsymbol{A}$, $\boldsymbol{B}$) be second-order tensors and ∇ the gradient operator; we define the following: $\boldsymbol{a} \cdot \boldsymbol{b} = a_i b_i$, $(\boldsymbol{A} \cdot \boldsymbol{a})_i = A_{il} a_l$, $(\boldsymbol{A} \cdot \boldsymbol{B})_{il} = A_{ip} B_{pl}$, $\boldsymbol{A} : \boldsymbol{B} = A_{il} B_{il}$, $(\nabla \boldsymbol{a})_{il} = \partial_l a_i$, $\nabla \cdot \boldsymbol{a} = \partial_i a_i$ and $(\nabla \cdot \boldsymbol{A})_i = \partial_l A_{il}$. The boldface font is also used for large matrix/vector assembled by the finite element method, e.g. $\boldsymbol{A}$, and $\boldsymbol{A}^*$ means the adjoint of $\boldsymbol{A}$. We denote by $H^k(\Omega)$ the Sobolev space $W^{k,2}(\Omega)$.

Chapter 2

Computational Homogenization with Neural Network Surrogate Energy Model

This chapter focuses on providing effective solutions to forward problems of cellular mechanical metamaterials. Traditional direct numerical simulation methods are computationally prohibitive. The central component is a neural network based surrogate model that approximates the effective strain energy density function. The formulations are within the classical framework of computational homogenization. We show that the neural network surrogate model enables accelerated forward simulations for cellular mechanical metamaterials, compared with a direct numerical simulation approach.

2.1 Introduction

Cellular mechanical metamaterials (CMMs) with repeating cells are commonly observed in both nature and industry: from biological materials such as honeycombs to synthetic structures such as metallic microlattices. These materials can exhibit unique mechanical properties, such as high stiffness- and/or strength-to-density ratio [90], that are primarily determined by geometries. Traditionally, the study of this field is limited to several simple structures and under relatively small deformation [91]. Yet in the past decade, the advent of fabrication technologies such as additive manufacturing [14, 13] have enabled precise but fast realization of sophisticated cellular architectures made of soft materials such as elastomers. These materials, not only can bear large deformation, but also exhibit novel mechanical properties under that condition, including negative Poisson's ratio (auxeticity) [2, 20, 92], shape morphing [3], tunable bandstructures [4] and energy absorption [5]. These novel properties created an avenue for many exciting engineering applications, e.g., soft actuators, materials with *in situ* tunable functionalities, reusable energy-absorbing materials, etc. [6, 7, 10, 8, 12, 9].

The rapid development on the fabrication side demands a fast and predictive computational method to determine the mechanical behaviors of CMMs. For materials with a relatively small number of cells, one can conduct a direct numerical simulation (DNS) on the whole specimen using the finite element method (FEM) [18]. However, the mechanical behaviors of soft CMMs are often highly nonlinear and may involve mechanical instabilities, thus requiring a fine spatiotemporal resolution for DNS calculations. DNS can easily become computationally intractable as CMMs usually consist of a large number of cells. Alternatively, one can conduct a fine-mesh FEM calculation on a representative volume element (RVE), often composed of one or several repeating cells [20], by prescribing appropriate boundary conditions. Such a method can efficiently predict the mechanical response of CMMs under homogeneous deformation. When CMMs are subject to complex loads, such predictions will fail.

When the size of the CMM is much larger than the repeating cells, multiscale analysis such as homogenization can be a useful tool. Homogenization, often called "coarse-graining" in the physics community, is often used to predict the macroscopic behavior of composites from their microstructures [21, 22]. The early development of homogenization primarily focuses on the analytical approach. Following the pioneer work of Eshelby [23], various analytical homogenization methods have been developed for linear materials [24, 93, 94, 95, 96, 97] and later extended to nonlinear materials [25, 26, 98, 99]. While these analytical approaches provide useful bounds, they are less accurate when microstructures become sophisticated [100]. Computational homogenization is more suitable in this scenario. Reviews of the recent progress in this field can be found in Ref's [101, 102, 103] . Among various methods of computational homogenization, two main categories can be distinguished: *concurrent* and *off-line* methods. *Concurrent* methods integrate the macro- and mico-scale problems through proper mathematical formulations and solve both problems concurrently. For example, the FE^2 method is a widely adopted concurrent scheme, and uses FEM to solve problems at both scales [27, 104, 105, 106, 107, 108]. Another example is given by the spectral method based on Fast Fourier Transform (FFT) [109, 110, 111, 112]. *Concurrent* methods are powerful and accurate, yet can still be computationally expensive, mainly due to the nested numerical solvers needed to balance calculations at both scales. On the other hand, for *off-line* methods, the macro- and micro-scale problems are solved separately. Effective constitutive laws are first constructed based on a set of numerical calculations at micoscales, and then used to solve problems at macroscales [28, 29]. Thus, *offline* methods can avoid the huge cost of nested numerical solvers and in some cases be orders of magnitude faster than the *concurrent* methods [30, 29]. Essentially, the constitutive relation can be seen as a multi-dimensional mapping from the microstructural features to the macroscopic response of the material, and the core of *offline* methods is to efficiently and accurately construct such a mapping. To this end, several techniques [31, 32, 33, 34, 30, 29, 35] have been developed, includig recent efforts using neural networks [36, 37].

Neural networks (NNs) are massively-parametric function approximators inspired by biological neural networks [113]. Given a large number of input/output examples, neural networks can often successfully learn functions on high-dimensional spaces. This process of "learning" the function corresponds to identifying an optimal set of parameters, i.e., neural network "weights", to capture the relationship between inputs and outputs reflected in the training data. Neural networks have been demonstrated to be powerful function approximators for solving challenging engineering tasks, such as language modeling [114], image classification [115], and machine translation [116]. More recently, there has been a growing interest in applying neural networks to complex problems in natural sciences, e.g., physics [38, 39, 40], chemistry [41], astronomy [42], biomedicine [117], and material science [43, 44, 45].

In the present work, we extend previous works on NN-based computational homogenization methods [36, 37] to model the nonlinear mechanical behavior of cellular mechanical metamaterials under large deformation. For that purpose, we first generate training data by conducting a large number of RVE calculations at cellular scale. We then train a neural network on those data to approximate the effective strain energy density at macro-scale, which is a function of the macroscopic strain and microscopic structural parameters. Through this NN-based surrogate model for effective strain energy density, the coarse-grained constitutive relations can be easily obtained, which greatly reduces the computational cost of determining the mechanical behavior of the cellular materials at macro-scale.

This chapter is organized as follows. In Section 2.2, the finite deformation theory of hyperelastic cellular mechanical metamaterials and the specific geometries of this study are described. In Section 2.3, we propose a neural network-based multiscale computational scheme to determine the mechanical responses of CMMs and describe the procedures to carry out the scheme in Section 2.4. Numerical examples are shown in Section 2.5, where our NN-based method is compared with DNS in terms of accuracy and efficiency. In Section 2.6, we discuss the applicability and the limitations of our NN-based method. Finally we conclude in Section 2.7.

2.2 Problem Formulation

2.2.1 Finite deformation elasticity

Consider a homogeneous elastic body in a 3D Eucleadian space $\mathbb{R}^3$, where we introduce a fixed Cartesian coordinate system with orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. In an unstressed state, the body occupies a region $\mathbb{B} \subset \mathbb{R}^3$ (reference configuration) with a boundary $\partial\mathbb{B}$ whose outward normal is $\boldsymbol{N}$. Upon mechanical loading, the body deforms and occupies a different region $\mathbb{B}_t \subset \mathbb{R}^3$ (deformed configuration). Deformation can therefore be described as a mapping: $\boldsymbol{\varphi} : \mathbb{B} \to \mathbb{B}_t$, which maps any material point $\boldsymbol{X} \in \mathbb{B}$ to its counterpart $\boldsymbol{x} \in \mathbb{B}_t$, i.e. $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X})$. The corresponding displacement field is defined as $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$ and the deformation gradient $\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}$. We denote vectors and tensors with a bold font in contrast to scalars with a normal font.

The constitutive model of a hyperelastic material can be defined by a strain energy density function (per volume) W, which depends on $\boldsymbol{F}$, through the right Cauchy-Green tensor $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$. The displacement field $\boldsymbol{u}$ can be determined by solving for the stationary point $\delta\Psi = 0$ of the following functional (in the absence of body force):

$$\Psi(\boldsymbol{u}) = \int_{\mathbb{B}} W(\boldsymbol{F})\ \mathrm{d}\boldsymbol{X} - \int_{\partial\mathbb{B}^N} \boldsymbol{u} \cdot \boldsymbol{t}\ \mathrm{d}\boldsymbol{S}, \tag{2.1}$$

with $\boldsymbol{u} = \boldsymbol{u}_b$ and $\delta\boldsymbol{u} = \boldsymbol{0}$ on $\partial\mathbb{B}^D$, where $\boldsymbol{u}_b$ is the prescribed displacement field, $\boldsymbol{t}$ is the prescribed traction (force per unit reference area), $\partial\mathbb{B}^N \cup \partial\mathbb{B}^D = \partial\mathbb{B}$ and $\partial\mathbb{B}^N \cap \partial\mathbb{B}^D = \emptyset$. The measures $d\boldsymbol{X}$ and $d\boldsymbol{S}$ are the infinitesimal volume and surface elements in the reference configuration respectively. Alternatively, the above variational problem can be formulated as a boundary value problem:

$$\begin{aligned} \nabla_{\boldsymbol{X}} \cdot \boldsymbol{P} = \boldsymbol{0} \qquad & \text{in } \mathbb{B}, \\ \boldsymbol{u} = \boldsymbol{u}_b \qquad & \text{on } \partial\mathbb{B}^D, \\ \boldsymbol{P} \cdot \boldsymbol{N} = \boldsymbol{t} \qquad & \text{on } \partial\mathbb{B}^N, \end{aligned} \tag{2.2}$$

where the divergence operator is in the reference configuration, and $\boldsymbol{P} = \frac{\partial W}{\partial \boldsymbol{F}}$ is the first Piola-Kirchoff stress. These problems can be numerically solved by FEM.

The formulation described above is applicable to any hyperelastic material, but in this work we focus on CMMs made of soft elastomers and therefore adopt the following form of W:

$$W = \frac{\mu}{2}(J^{-2/3} I_1 - 3) + \frac{\kappa}{2}(J-1)^2, \tag{2.3}$$

where $J = \det(\boldsymbol{F})$, $I_1 = \mathrm{tr}(\boldsymbol{C})$; $\mu = \frac{E}{2(1+\nu)}$ and $\kappa = \frac{E}{3(1-2\nu)}$ denote the initial shear and bulk moduli, respectively, E and ν being the material's Young's modulus and Poisson's ratio. The above W is commonly used to model isotropic elastomers that are almost incompressible and we assume $\nu = 0.3$ throughout this work.

2.2.2 Problem geometry

In this work, we focus on a special but widely used class of CMM: 2D porous cellular solids with a repeating unit cell. We adopt a plane strain setting here, i.e., $u_i = u_i(X_1, X_2)$, $i = 1, 2$ and $u_3 = 0$. Previous works [20, 92] have shown that the mechanical properties of these CMMs highly depend on the shape of the pore. Following previous treatments, our study focuses on pores with four-fold symmetry

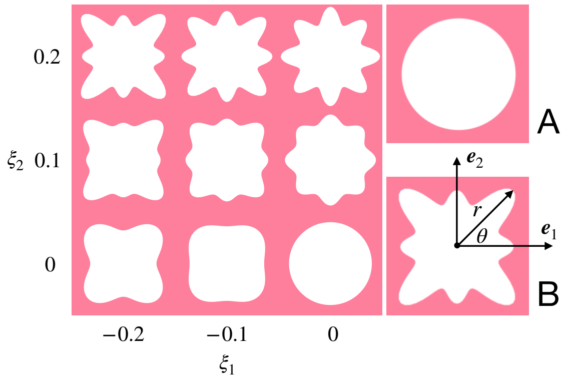


Figure 2.1: Left: A map from the parameters to the pore shapes indicated by Eqn (2.4). For each (ξ_1, ξ_2), we get a corresponding pore shape. Right: The two representative pore shapes studied in this work.

whose contour can be described by the following parametrization:

$$r(\theta) = r_0(1 + \xi_1\cos(4\theta) + \xi_2\cos(8\theta)), \tag{2.4}$$

where r and θ are polar radius and polar angle respectively. By changing the parameters $\boldsymbol{\xi} = (\xi_1, \xi_2)$, we get a family of different pore shapes as illustrated in Fig. 2.1. Specifically in this work, we focus on square arrays of these unit cells and therefore, for material with a unit cell length of L_0, its porosity ϕ_0 is uniquely determined by r_0 and $\boldsymbol{\xi}$ via the relation $\phi_0 = \frac{\pi}{2}(r_0/L_0)^2(\xi_1^2 + \xi_2^2 + 2)$. In this work, we fix $L_0 = 0.5$ and porosity $\phi_0 = 0.5$.

Among all the possible pore shapes, we focus on two types (A and B in Fig. 2.1): *Pore A* is circular, with $\boldsymbol{\xi}_A = (0, 0)$ and *pore B* breaks the angular symmetry, with $\boldsymbol{\xi}_B = (-0.2, 0.2)$. We choose these two shapes as our representative examples for two reasons. First, mechanical instability can be triggered in CMMs made of both unit cells under compression, which leads to an asymmetric mechanical response of the CMMs under tension and compression. Second, even though mechanical instabilities can be triggered in these two CMMs, they still exhibit very distinct mechanical responses under the same mechanical loading [20, 92].

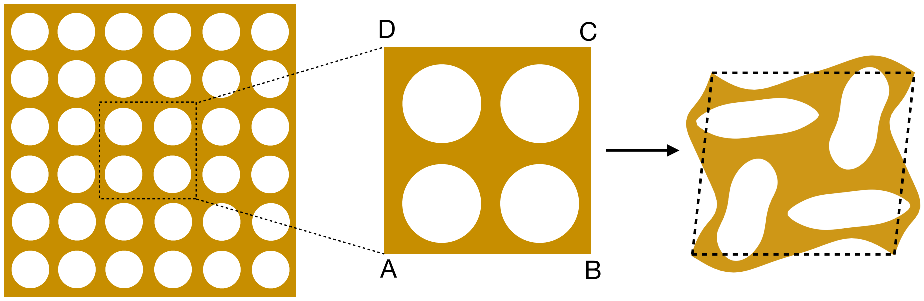


Figure 2.2: A 2x2 RVE taken from a cellular porous structure with repeating units. Followed by the arrow is an example showing the periodic boundary conditions applied to the RVE. Here we specify a macroscopic displacement gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} 0 & 0.2 \\ 0 & -0.2 \end{smallmatrix}\right]$. The deformed configuration (rightmost) shows the resulted displacement field of this RVE: $\boldsymbol{u} = \overline{\boldsymbol{H}} \cdot \boldsymbol{X} + \boldsymbol{u}^{\star}$ with $\boldsymbol{u}^{\star}$ being the periodic fluctuation. The dashed profile shows the deformation corresponding to the appplied mean displacement $\overline{\boldsymbol{u}} = \overline{\boldsymbol{H}} \cdot \boldsymbol{X}$.

2.3 NN-Based Multiscale Approach

Direct numerical simulation of cellular mechanical metamaterials made of large numbers of unit cells is challenging, especially for CMMs with complicated cellular geometries or that undergo large deformation, mainly due to the large computational cost. The computational expense arises from two factors. First, to resolve the detailed cellular geometries, the finite element mesh size must be no larger than the microstructural features. Second, since the mechanical responses can be highly nonlinear under large deformation, small timesteps are required for DNS to converge. To address these issues, we adopt a multiscale approach called (offline) computational homogenization [36, 37], which performs fine-mesh FEM calculations at the RVE level to obtain the coarse-grained constitutive relations, and then uses these to predict the mechanical behavior of the material at larger scale.

Our computational homogenization approach starts with a study of the representative volume element of the CMM. The choice of RVE is not unique, but it has to simultaneously be large enough to capture the influence of cellular geometries on the overall mechanical behavior and small enough to be considered as a material point for the CMM. For example, as shown in Fig.2.2, for a CMM composed of many unit cells with *pore A*, we have chosen $ABCD$, a 2 × 2 array of the unit cells as our RVE. This choice of RVE ensures that we capture the mechanical instabilities that lead to reorganizations of the neighboring unit cells [20, 92]. We emphasize that the above statement relies on the critical assumption of separation of length scales, i.e. the size of the CMM is much larger than that of the unit cell and RVE, without which the homogenization approach is invalid.

To construct the effective strain energy density of this RVE, we conduct a fine-mesh FEM calculation on the RVE under macroscopic deformations $\overline{\boldsymbol{F}}$ by prescribing the following periodic boundary conditions:

$$\boldsymbol{u}(\boldsymbol{X}) = \left(\overline{\boldsymbol{F}} - \boldsymbol{I}\right) \cdot \boldsymbol{X} + \boldsymbol{u}^{\star}(\boldsymbol{X}) = \overline{\boldsymbol{H}} \cdot \boldsymbol{X} + \boldsymbol{u}^{\star}(\boldsymbol{X}), \tag{2.5}$$

where $\boldsymbol{I}$ is identity tensor and $\overline{\boldsymbol{H}}$ denotes the macroscopic displacement gradient. $\overline{\boldsymbol{F}}$ and $\overline{\boldsymbol{H}}$ are uniform on the RVE and periodic boundary conditions are applied to ensure $\int_{\partial V} \boldsymbol{n} \otimes \boldsymbol{u}^{\star} = 0$. For example, for the RVE $ABCD$ in Fig 2.2, we have $\boldsymbol{u}^{\star}_{AD} = \boldsymbol{u}^{\star}_{BC}$ and $\boldsymbol{u}^{\star}_{AB} = \boldsymbol{u}^{\star}_{DC}$. Essentially, eqn (2.5) decomposes the total displacement of the RVE into a macroscopic (overall) part $\overline{\boldsymbol{u}} = \overline{\boldsymbol{H}} \cdot \boldsymbol{X}$ and a microscopic (fluctuating) part $\boldsymbol{u}^{\star}$.

The effective strain energy density $\overline{W}$ can obtained via the average W over the RVE: $\overline{W} = V^{-1} \displaystyle\int_V W \, dV$ with V being the total volume of the RVE (for plane strain, $dV = dX_1 dX_2$). Other macroscopic quantities can be obtained in the same fashion. This effective strain energy density $\overline{W}$ is the bridge that connects the microscopic features to macroscopic mechanical responses. In essence, $\overline{W}$ should be a function of $\overline{\boldsymbol{F}}$ as well as the microstructural features $\boldsymbol{\xi}$: $\overline{W} = \overline{W}(\overline{\boldsymbol{F}}, \boldsymbol{\xi})$. Once that relation is established, we then treat the RVE as a material point in the CMM and use FEM to find the approximate solution to the stationary point of the following functional:

$$\overline{\Psi}(\overline{\boldsymbol{u}}) = \int_{\mathbb{B}} \overline{W}(\overline{\boldsymbol{F}}, \boldsymbol{\xi}) \; \mathrm{d}\boldsymbol{X} - \int_{\partial \mathbb{B}^N} \overline{\boldsymbol{u}} \cdot \bar{\boldsymbol{t}} \; \mathrm{d}\boldsymbol{S}, \tag{2.6}$$

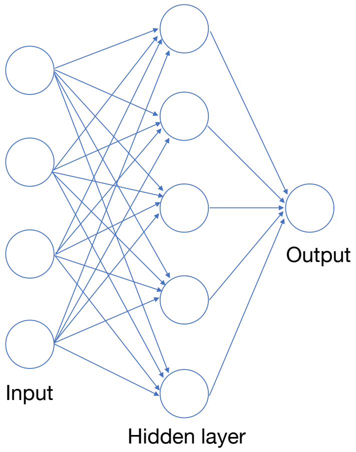


Figure 2.3: A MLP with one hidden layer. The neural network function takes an input vector and performs an affine transformation followed by a non-linear activation function on the input vector, producing an intermediate vector at the hidden layer. A similar transformation on the intermediate vector then yields the output of the neural network function.

with $\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}_b$ and $\delta\overline{\boldsymbol{u}} = 0$ on $\partial\mathbb{B}^D$. Making use of the celebrated Hill-Mandel condition [118]: $\overline{\boldsymbol{P}} = \frac{\partial \overline{W}}{\partial \overline{\boldsymbol{F}}}$, the above minimization fomulation can be shown to be equivalent to the following macroscopic BVP:

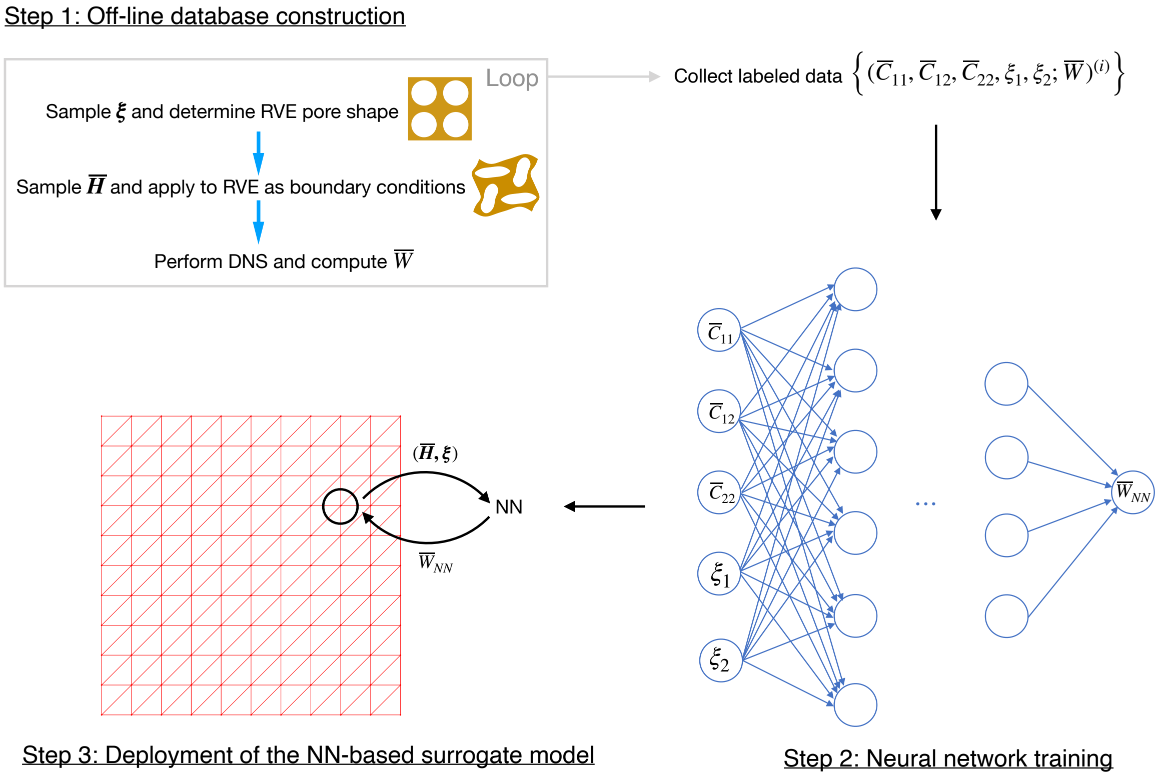
$$\begin{aligned} \nabla_{\boldsymbol{X}} \cdot \overline{\boldsymbol{P}} = \boldsymbol{0} \qquad & \text{in } \mathbb{B}, \\ \overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}_b \qquad & \text{on } \partial\mathbb{B}^D \\ \overline{\boldsymbol{P}} \cdot \boldsymbol{N} = \bar{\boldsymbol{t}} \qquad & \text{on } \partial\mathbb{B}^N. \end{aligned} \tag{2.7}$$

Therefore, the key to this computational homogenization scheme is to construct $\overline{W}$ as a function of $\overline{\boldsymbol{F}}$ (more strictly speaking $\overline{\boldsymbol{C}}$) and $\boldsymbol{\xi}$. In this work, we adopt a data-driven approach based on neural networks.

Neural networks (NNs) are powerful computational structures for constructing massively-parametric mappings [113]. The classic general form of a neural network is the multi-layer perceptron (MLP), made up of fully-connected layers [119]. The input vector $\boldsymbol{x}$ is processed through several hidden layers and finally to an output vector $\boldsymbol{y}$. Suppose that the ith layer has m nodes and the $i+1$ layer has n of them, the computation between these two layers are given by $\boldsymbol{y}^i = f^i(\boldsymbol{x}^i) := \sigma(\boldsymbol{w}^i \cdot \boldsymbol{x}^i + \boldsymbol{b}^i)$, where $\boldsymbol{w}^i \in \mathbb{R}^{n\times m}$, $\boldsymbol{b}^i \in \mathbb{R}^m$ are the weight matrix and bias vector of the ith layer, and σ is an element-wise nonlinear function, such as a logistic function $\sigma(y) = 1/(1+e^{-y})$ or tanh. We can "stack" these transformations as $\boldsymbol{y} = f^k(\boldsymbol{x}^k) = f^k \circ f^{k-1}(\boldsymbol{x}^{k-1}) = ... = f^k \circ ... \circ f^1(\boldsymbol{x})$ for a MLP with $k-1$ hidden layers. We hereby denote this mapping as $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$ to emphasize that this mapping is under the parameter $\boldsymbol{\theta} = (\boldsymbol{w}^1, \boldsymbol{w}^2, ..., \boldsymbol{w}^k; \boldsymbol{b}^1, \boldsymbol{b}^2, ..., \boldsymbol{b}^k)$. A typical MLP with only one hidden layer is shown in Fig. 2.3.

The weight matrices and bias vectors $\boldsymbol{\theta}$ are called the parameters of the neural network and they are optimized through a training procedure that minimizes a loss function reflecting the quality of the fit to data. This training is typically performed via a stochastic optimization procedure such as stochastic gradient descent [120]. At each step, one randomly selects a subset of the training examples $\boldsymbol{x}$ each with corresponding target value $\boldsymbol{y}*$ and computes a loss function $\boldsymbol{\mathcal{L}}(\boldsymbol{y}*, \boldsymbol{y})$ (for example, the squared Euclidean distance between targets and predictions $||\boldsymbol{y}^* - \boldsymbol{y}||_2^2$). Gradients of this loss function with respect to the parameters are then computed using reverse-mode automatic differentiation and the parameters are updated accordingly [121]. This proceeds until a convergence criterion is achieved. After training and other validation and calibration processes, we can then deploy the neural network to predictions for new inputs.

In this work, we use a MLP to approximate the effective strain energy density, namely $\overline{W}_{NN}(\overline{\boldsymbol{C}}, \boldsymbol{\xi}) \approx \overline{W}(\overline{\boldsymbol{C}}, \boldsymbol{\xi})$. The input vector is a 5-dimensional vector: $\boldsymbol{x} = (\overline{C}_{11}, \overline{C}_{12}, \overline{C}_{22}, \xi_1, \xi_2)$ and the output is a scalar $y = \overline{W}_{NN}$. For any given input, the target output is generated by RVE calculations. The goal is to train an MLP which can reproduce the result of these calculations for out-of-sample vectors. After training the MLP, we replace the $\overline{W}$ in Eqn (2.6) with $\overline{W}_{NN}$ and solve the macroscopic problem.


Step 1: Off-line database construction

Step 2: Neural network training

Step 3: Deployment of the NN-based surrogate model

Loop
Sample $\boldsymbol{\xi}$ and determine RVE pore shape
Sample $\overline{\boldsymbol{H}}$ and apply to RVE as boundary conditions
Perform DNS and compute $\overline{W}$
Collect labeled data $\left\{(\overline{C}_{11}, \overline{C}_{12}, \overline{C}_{22}, \xi_1, \xi_2; \overline{W})^{(i)}\right\}$
$\overline{C}_{11}$
$\overline{C}_{12}$
$\overline{C}_{22}$
ξ_1
ξ_2
...
$\overline{W}_{NN}$
$(\overline{\boldsymbol{H}}, \boldsymbol{\xi})$
NN
$\overline{W}_{NN}$

Figure 2.4: The data-driven computational homogenization procedure consists of three steps: 1) building the offline training database by performing RVE calculations at cellular level; 2) training the neural network to obtain a surrogate model for effective strain energy density; 3) deploying the NN-based surrogate model for macroscopic problems.

We use `scikit-learn` [122], an open source machine learning framework, to build, train and later test our neural network. All finite element calculations are carried out using an open-source FEM package `FEniCS` [123].

2.4 Construction of the NN-Based Computational Homogenization Scheme

In the previous section, we have outlined a NN-based multiscale computational homogenization scheme, with the goal of obtaining a mapping from macroscopic deformation $\overline{\boldsymbol{C}}$ and microstructural parameters $\boldsymbol{\xi}$ to the effective strain energy density $\overline{W}$. This scheme, as illustrated in Fig. 2.4, mainly consists of three steps: 1) offline construction of training data, 2) optimization of neural network parameters, and 3) deployment of the NN-based surrogate model for FEM at macro-scale. In this section, we will describe these steps in detail.

2.4.1 Offline generation of training data

The first step of constructing the NN-based multiscale computational scheme is to construct a proper training database composed of labeled data, each in the form of an input vector and its target output: $\{(\overline{C}_{11}, \overline{C}_{12}, \overline{C}_{22}, \xi_1, \xi_2; \overline{W})^{(i)}\}$. The target output $\overline{W}^{(i)}$ is obtained via finite element calculation of the RVE with the corresponding geometric parameters $\boldsymbol{\xi}^{(i)}$ and subject to the corresponding macroscopic field of $\overline{\boldsymbol{C}}^{(i)}$.

In this work, we focus on CMMs composed of two types of unit cells: those with *pore A* and *pore B*, so the choice of $\boldsymbol{\xi}^{(i)}$ is limited: $\boldsymbol{\xi}^{(i)} \in \{\boldsymbol{\xi}_A, \boldsymbol{\xi}_B\}$. However, the choice of the macroscopic field $\overline{\boldsymbol{C}}^{(i)}$ is in principle infinite since it varies continuously. Therefore some care is necessary in constructing an effective sampling method for $\overline{\boldsymbol{C}}^{(i)}$.

First we must identify a domain over which $\overline{\boldsymbol{C}}^{(i)}$ is to be sampled. Such a region should cover the nonlinear mechanical behaviors of primary interest. For example, for CMMs composed of unit cells with *pore A*, beyond certain compression the CMM undergoes mechanical instabilities which lead to auxetic behavior. This behavior can be observed for $\overline{H}_{22} = -0.125$, $\overline{H}_{12} = \overline{H}_{21} = 0$ and $\overline{H}_{11} \in [-0.08,\ 0.08]$, as shown in Fig. 2.5a. The effective Poisson's ratio is calculated to be $\overline{\nu} = -0.26$ (defined as $-\frac{\overline{H}_{11}}{\overline{H}_{22}}$ at the star point, where $\frac{\partial \overline{W}}{\partial \overline{F}_{11}} = \overline{P}_{11} = 0$). As for CMMs made of unit cell with *pore B*, when subject to both shear and compression, mechanical instabilities can lead to the bifurcation of microstructures [92]. This is observed when $\overline{H}_{22} = -0.125$, $\overline{H}_{11} = \overline{H}_{21} = 0$ and $\overline{H}_{12} \in [-0.7,\ 0.7]$. As shown in Fig. 2.5b, a double well shape of the $\overline{W}$ is observed as a function of $\overline{H}_{12}$ with two energy minima. Therefore, when the CMM is subject to an overall compression $\overline{H}_{22} = -0.125$, the CMM will bifurcate into two different microstructures that correspond to these two minima.

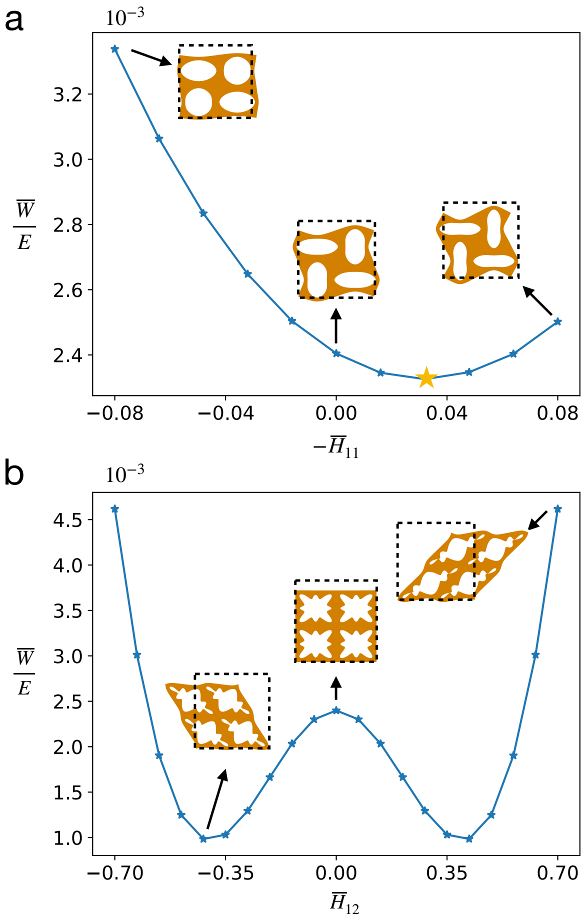


Figure 2.5: Effective strain energy density $\overline{W}$ (normalized by E) under different $\overline{\boldsymbol{H}}$ obtained from RVE calculations for unit cells with *pore A* and *pore B*: (a) $\overline{W}/E$ versus $\overline{H}_{11}$ for *pore A* and (b) $\overline{W}/E$ versus $\overline{H}_{12}$ for pore B.

The inspections above offer insight into a reasonable range of $\overline{\boldsymbol{H}}$ for sampling data. The range should be sufficiently wide so that important mechanics are reflected in the database, while it should also be limited so that undesired behaviors like self-contact are avoided with the current FEM simulation. The sampling region chosen for *pore A* is:

$$\begin{aligned}&\{-0.2 \leq \overline{H}_{11} \leq 0.2, -0.2 \leq \overline{H}_{12} \leq 0.2,\\&-0.2 \leq \overline{H}_{21} \leq 0.2, -0.2 \leq \overline{H}_{22} \leq 0.2\},\end{aligned} \tag{2.8}$$

and that for *pore B*:

$$\begin{aligned}&\{-0.2 \leq \overline{H}_{11} \leq 0.2, -0.8 \leq \overline{H}_{12} \leq 0.8,\\ &-0.8 \leq \overline{H}_{21} \leq 0.8, -0.2 \leq \overline{H}_{22} \leq 0.2\}.\end{aligned} \tag{2.9}$$

We adopted Sobol sequences [124] to generate 5000 differet samples from each of the above regions. Such sequence is often used to generate sparse yet representative samples that are evenly distributed over the given region. To maximize the speed of data collection, we conducted a mesh refinement study to obtain the optimal mesh size under a reasonable tolerance (see SI). On a personal computer with 3.2 GHz Intel Core i7 CPU and 16GB memory, it took about 20 hours to complete the process of data collection.

2.4.2 Neural network training

Once the database is constructed, the next step is to train a neural network to establish a mapping from input vector $\boldsymbol{x} = (\overline{C}_{11}, \overline{C}_{12}, \overline{C}_{22}, \xi_1, \xi_2)$ to its target scalar output $y = \overline{W}$. We randomly split our database into a training set (90% of the data) and a test set (the rest 10%). Our neural networks will be trained on the training set and the test set is used to ensure good generalizability and avoid overfitting.

The neural network used here is a MLP with one hidden layer, which as we will show later is able to provide sufficient accuracy for our problem. The logistic function $\sigma(y) = 1/(1 + e^{-y})$ is chosen as our activation function, and mean squared error (MSE) is chosen as our loss function: MSE $= \sum_{i=1}^{n} (\hat{y}^{(i)} - y^{(i)})^2/n$, where n is the number of data points evaluated and $\hat{y}^{(i)}$ denotes the output of the MLP given input $\boldsymbol{x}^{(i)}$. The neural network is trained using mini-batch stochastic gradient descent with Adam optimizer [125]. Hyperparameters such as number of neurons in the hidden layer are first optimized via k-fold cross-validation [126] (see SI). We

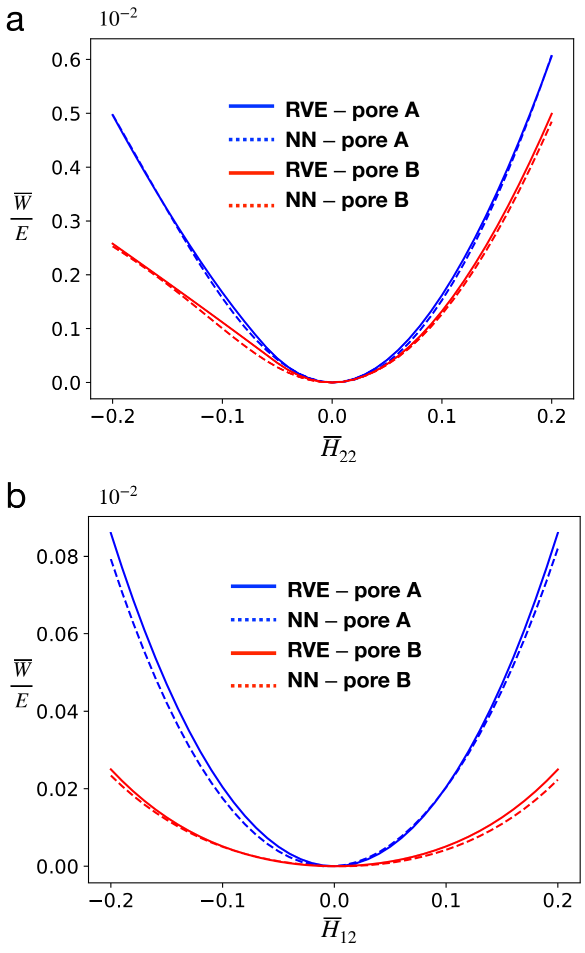


Figure 2.6: Comparisons of the normalized effective strain energy densities obtained via RVE calculation (solid line) and trained NN model (dashed line) for both *pore A* (blue) and *pore B* (red) on the two benchmarks: (a) uniaxial strain, (b) simple shear strain.

obtained the following optimal hyperparameters: learning rate 10^{-2}, batch size 64, and 128 neurons in the hidden layer. We then train our neural networks using these hyperparameters. The training time for the NN is typically within 1 minute on a personal computer. After training of 1000 epochs, our NN model reports a training MSE of 6.11×10^{-5} and a test MSE of 7.67×10^{-5}.

We performed polynomial regression on the same data set as a baseline comparison. As shown in Fig. 2.7, when the maximum degree of polynomial is increased, the MSE tends to decrease. But when the maximum degree exceeds 10, the regression problem start to become ill-conditioned and thus resulting in large MSE [127]. The

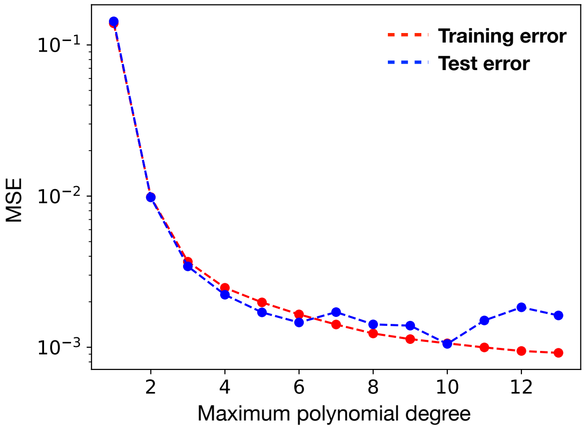


Figure 2.7: Training and test MSE for polynomial regression.

best of polynomial regression reports a test error of 1.05×10^{-3}, which is more than an order of magnitude larger than the neural network model.

We further validate our NN model in two test cases to serve as our benchmarks by prescribing the following $\overline{\boldsymbol{H}}$:

1. uniaxial strain: $\overline{H}_{11}, \overline{H}_{12}, \overline{H}_{21} = 0,\ \overline{H}_{22} \neq 0$
2. simple shear strain: $\overline{H}_{11}, \overline{H}_{21}, \overline{H}_{22} = 0,\ \overline{H}_{12} \neq 0$

We conducted above two test cases on both unit cells with *pore A* and *pore B* and ensure that the $\overline{\boldsymbol{H}}$'s tested here are different from those in the database. The results are shown in Fig. 2.6, where the effective strain energy obtained via RVE calculation and that from the NN model are compared. It is observed that the NN model agrees with the RVE calculation reasonably well. The errors between NN model and RVE calculation are also small among both benchmarks: the respective MSEs are 2.42×10^{-5} and 4.28×10^{-5}.

2.4.3 Deployment of the NN-based surrogate model

Now that we have obtained an optimized NN model of the effective strain energy density $\overline{W}_{NN}$, we replace the $\overline{W}$ in Eqn (2.6) with the obtained $\overline{W}_{NN}$. Then, for any given $\boldsymbol{\xi}$, the solution to the macroscopic problem boils down to finding the stationary point $\delta\overline{\Psi}_{NN} = 0$ of the following functional:

$$\overline{\Psi}_{NN}(\overline{\boldsymbol{u}}) = \int_{\mathbb{B}} \overline{W}_{NN} \left[\, \overline{\boldsymbol{C}}(\overline{\boldsymbol{u}}), \boldsymbol{\xi} \right] \mathrm{d}\boldsymbol{X} - \int_{\partial\mathbb{B}^N} \overline{\boldsymbol{u}} \cdot \bar{\boldsymbol{t}} \; \mathrm{d}\boldsymbol{S}, \tag{2.10}$$

with $\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}_b$ and $\delta\overline{\boldsymbol{u}} = 0$ on $\partial\mathbb{B}^D$.

By extracting out the optimal parameters of the NN model, the explicit functional form of $\overline{W}_{NN}(\overline{\boldsymbol{C}}, \boldsymbol{\xi})$ is readily available, which can be computed iteratively as described in Section 2.3. However, to find the stationary point of the above functional $\overline{\Psi}_{NN}$, one needs to efficiently evaluate the derivatives of $\overline{\Psi}_{NN}$ to compute quantities like first Piola-Kirchhoff stress $\overline{\boldsymbol{P}}_{NN} = \frac{\partial \overline{W}_{NN}}{\partial \overline{\boldsymbol{F}}}$ as well as the tangent stiffness tensor $\mathbb{K} = \frac{\partial \overline{\boldsymbol{P}}_{NN}}{\partial \overline{\boldsymbol{F}}}$. In this work, we made use of the automatic symbolic differentiation feature provided by the open-source package `FEniCS` (specifically the `UFL` component [128] of the package) to compute those derivatives. The standard Newton-Raphson method is used to find the root of the nonlinear equations. We use the automatic differentiation and other features of `FEniCS`, so that the code can be written in a concise and near-math fashion, which provides an easier access to broader audiences. Equivalently, the exact same derivatives could be computed analytically [36] or via finite differences [37].

2.5 Numerical Examples

In this section, we employed the aforementioned NN-based multiscale approach to study the mechanical behavior of CMMs composed of 2D arrays of cells with different pore shapes and compare that with direct numerical simulation. As shown in Fig. 2.8, a square-shape CMM with a length $L = 16L_0$ is studied. The CMM is subject to a uniaxial testing with displacement control. Specifically, a displacement ΔL along the $\mathbf{e}_2$ direction is applied at the top with the bottom kept fixed, while the left and

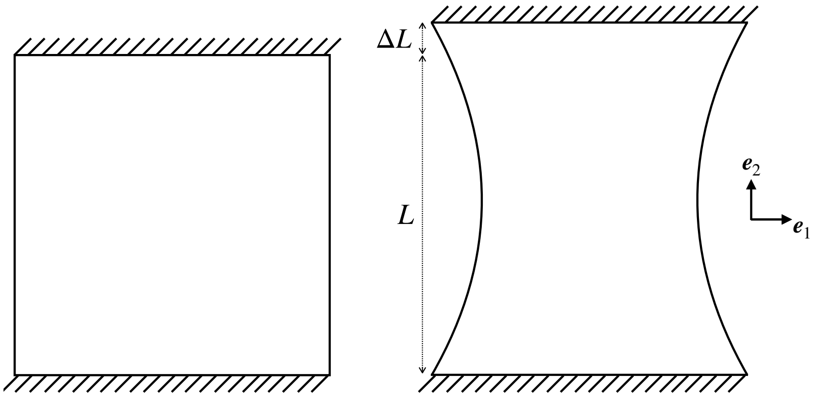


Figure 2.8: Boundary conditions for uniaxial tests. The upper and lower sides of the plate have fixed displacement conditions while the left and right sides are traction free.

right sides of the CMM remain traction free. This uniaxial testing is a standard practice to obtain the stress-strain curve of materials. Here, the engineering strain ϵ and engineering stress σ of the CMM are defined as follows: $\epsilon = \frac{\Delta L}{L}$ and $\sigma = \frac{F}{L}$, where $F = \int_{\text{top}} P_{22} \, \mathrm{d}X_1$ is the total resultant force (per thickness) on the top surface.

We perform this uniaxial test on a CMM with spatially uniform pore shapes of either *pore A* or *pore B*. Previous studies [20] suggested that compression can lead to distinct mechanical behaviors of these two CMMs. This phenomenon is indeed observed in Fig 2.9, where the deformed shapes of the CMMs under an engineering strain of -10% are shown. Both DNS (Fig 2.9a and b) and the NN-based model (Fig 2.9e and f) have shown that even under the same compressive loads, the two CMMs develop very different shapes. We also computed the average displacement field over each RVE element of the CMMs using data from DNS. The results are shown Fig 2.9c and d.

Specifically for CMMs with *pore A*, as mentioned in Section 2.4.2, a negative Poisson's ratio is expected, i.e., the compression in $\mathbf{e}_2$ direction will lead to a contraction also in $\mathbf{e}_1$ direction. This can be easily observed for the DNS results shown in Fig 2.9a, where the CMM starts to contract in the middle region. Although not as visually observable, the NN-based model also produces that result, as shown in

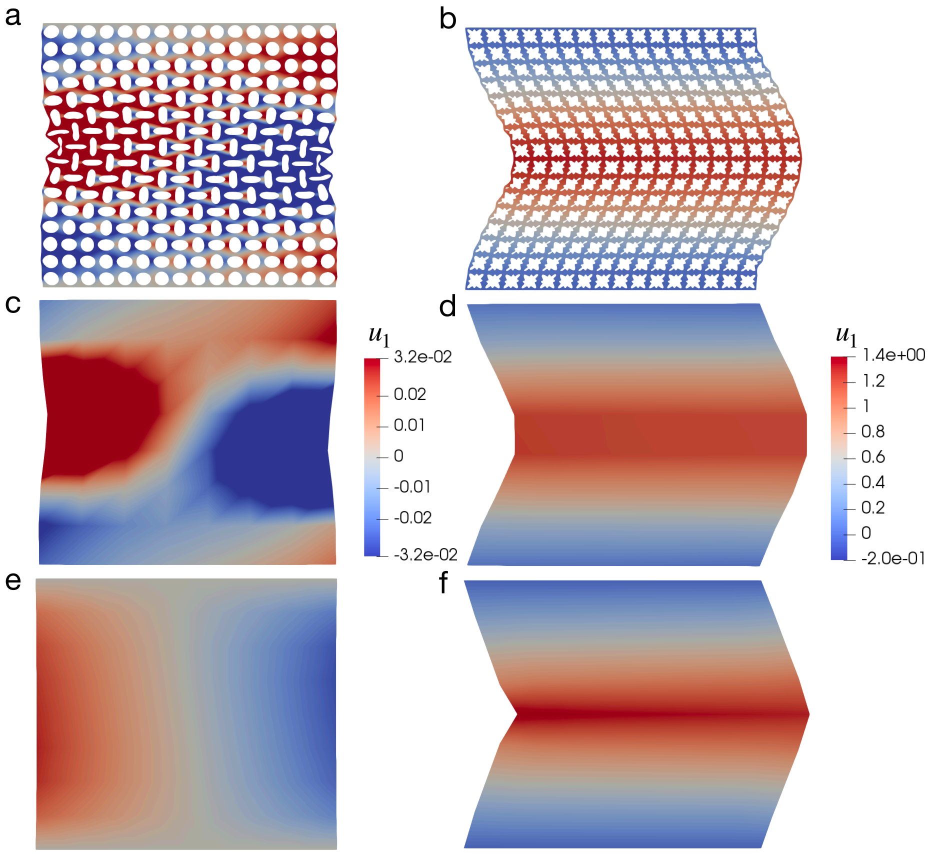


Figure 2.9: Deformed configurations of CMM with *pore A* (left column) and *pore B* (right column) under a 10% uniaxial compressive strain. (a) and (b) are results obtained from DNS, (c) and (d) are the RVE-averaged field of DNS, and (e) and (f) are obtained via NN-based model. Color contours indicate the value of u_1 (displacement component along $\mathbf{e}_1$ direction).

Fig. 2.9e where displacement component u_1 is plotted. We can observe that it is positive on the left and negative on the right, which is a clear sign of contraction. As for CMMs with *pore B*, compression can lead to a bifurcation of local microstructure, as mentioned in Section 2.4.2, which again can be observed from the DNS result in Fig. 2.9b. For NN-based model, since the microstructural information has been averaged out, such bifurcation can only be inferred indirectly. Recall that in Fig. 2.5b, the two minima of $\overline{W}$ correspond to nonzero shear deformation with equal and opposite shear components. In Fig. 2.9f, we observe that under pure compression, the CMM develops a large local shear in the upper and lower part of the material, with opposite

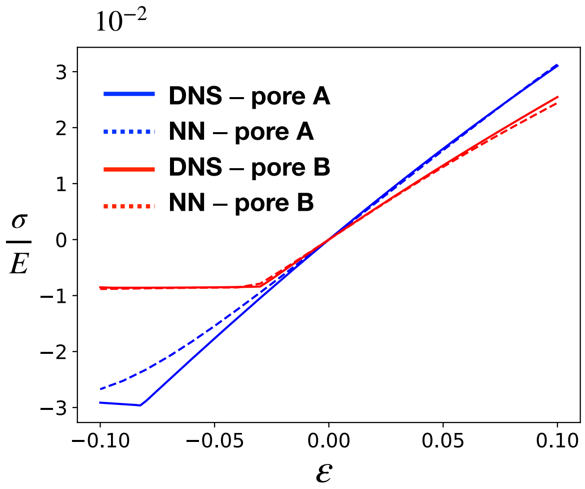


Figure 2.10: Stress-strain curve for CMM under uniaxial test with *pore A* (red) and *pore B* (blue) obtained via DNS (solid line) and NN-based model (dashed line).

shear directions. The results in Fig. 2.9, whether obtained via DNS or the NN-based approach are all in qualitative agreement with previous experimental work [20].

Beyond visual qualitative comparisons, we obtained a stress-strain (σ-ϵ) curve for both CMMs to quantitatively assess the accuracy of our NN-based approach. The curves are obtained by varying the applied engineering strain ϵ from -10% to 10% and the results are shown in Fig. 2.10. The curves obtained from DNS and the NN-based model are almost identical for both *pore A* and *pore B* when under tension, even at large strain up to 10%. The NN-based model can reproduce the result of DNS for *pore B* for compression, but it starts to deviate from DNS results for *pore A* when the compression becomes large. This deviation is observed even with a much finer mesh for our NN-based approach.

One possible reason for such behavior is the boundary effect. For the CMMs with *pore B* we can observe some variation of microstructures near the boundary, but the variation is quite smooth and the main effect – the bifurcation into two shear states – is captured by the NN-based model. However for CMMs with *pore A*, on the left and right boundaries there are strong pore-boundary interactions, resulting in a drastic change in pore shapes (see SI). This boundary effect persists even with the increasing number of composing unit cells. Another possible reason is that there is strong microstructure localization in the case of *pore A*. We can see that the rotation of the holes is drastic, and the distances between holes are small, thus leading to strong local interactions between the RVEs (Fig. 2.9a and SI). This is in contrast to the case of pore B, where pore rotations vary more smoothly yielding no strong localization (Fig. 2.9b and SI). Moreover, the strongly localized deformations in the case of pore A can lead to contacts between the interior boundaries of the pore, at which point our finite element model becomes invalid (without extension to handle contacts). In conclusion, the strong local variations in the microstructural features may impede the predictive power of our computational scheme. Possible ways of resolving this issue include using a separate model to account for the boundary effects [129], or utilizing more advanced homogenization techniques to deal with the aformentioned localization effects [101, 102].

Aside from accuracy, we also compared the efficiency of the NN-based approach with DNS. Recall that one of our main motivations to construct such a surrogate-based approach is to reduce the computational complexity of DNS. This computational expense is mainly due to the fact that a small mesh size is needed to resolve the detailed cellular geometries, which results in a large number of degrees of freedom (DoF). Our NN-based model can use a much coarser mesh since the detailed microstructural features have already been averaged out and factored into the effective strain energy density. This significantly reduces the degrees of freedom when solving for the macroscopic behavior of the CMM–of course at the cost of losing the detailed local information of relevant fields. Table 2.1 summarizes the computational performance of DNS and NN-based approach when they are used to generate the stress-strain curve in Fig. 2.10. NN-based approach needs much fewer DoF to accurately capture the overall mechanics of the CMMs and hence can reduce the computational time by up to two orders of magnitude for both CMM with *pore A*

	DoF	computation time [s]
DNS - *pore A* - compression	158310	5.15×10^2
DNS - *pore A* - tension		7.42×10^2
NN - *pore A* - compression	290	6.18
NN - *pore A* - tension		6.41
DNS - *pore B* - compression	139520	5.33×10^2
DNS - *pore B* - tension		5.19×10^2
NN - *pore B* - compression	290	8.20
NN - *pore B* - tension		6.94

Table 2.1: Comparison of DoF (degrees of freedom) and computation time (in secends) for DNS and NN with different pore shapes and loading conditions.

and that with *pore B*. Note that for our problem, we used the same increment in ϵ to obtain the stress-strain curve. For time-dependent problem, it is also important to examine the numerical stability of both approaches when different time steps are used.

2.6 Discussion

Our NN-based approach is shown to efficiently predict the mechanical behavior of CMMs with homogeneous $\boldsymbol{\xi}$ but an inhomogeneous deformation field. Here we further consider a CMM whose $\boldsymbol{\xi}$ is macroscopically inhomogeneous: $\boldsymbol{\xi}\left(\boldsymbol{X}\right) = \frac{X_2}{L}\boldsymbol{\xi}_B + (1 - \frac{X_2}{L})\boldsymbol{\xi}_A$, a linearly varying $\boldsymbol{\xi}$ along the $\mathbf{e}_2$ direction. The shape of the CMM is still a square, with size $L = 16L_0$ and under the same uniaxial loading as described in Fig. 2.8. A maximum strain of 20% is applied to this CMM. Its deformed shape is shown in Fig. 2.11 when under a tensile strain of 20%. The macroscopic responses for both DNS and the NN-homogenized model agree with each other qualitatively by visual observation. When the applied $\epsilon > 0$, i.e., tensile loading, we have found that the NN-based approach can reproduce the DNS results very well–as in the case of a

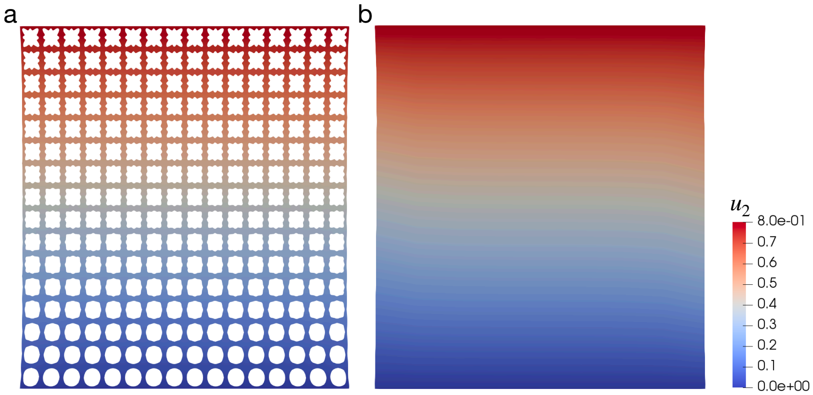


Figure 2.11: Deformed configuration of a CMM with non-uniform pore shapes under a 20% uniaxial tension obtained via (a) DNS and (b) NN-based approach. Color contours indicate the value of u_2.

CMM with homogeneous $\boldsymbol{\xi}$. While when under compression, the NN-based approach starts to diverge and fails to find a solution even at small compression.

We suggest two explanations for this phenomenon. First, since $\overline{W}$ is a strongly nonlinear function of $\boldsymbol{\xi}$, especially when under compression, the ability of our neural network to capture that functional dependence remains uncertain, given that only two different $\boldsymbol{\xi}$'s ($\boldsymbol{\xi}_A$ and $\boldsymbol{\xi}_B$) are included in the training data. Second, since the local mechanical properties of the CMM depend on $\boldsymbol{\xi}$ in a strongly nonlinear fashion, the $\boldsymbol{\xi}$ used in this study can still create a strong local variation. When the characteristic length of that variation is comparable to the RVE size, the assumption of separation of scales can break and in that case our multiscale framework will fail. Particularly in the above example, when the CMM is under uniaxial tension, as shown in Fig. 2.10, the mechanical properties of CMMs vary smoothly between different $\boldsymbol{\xi}$'s. Therefore, the NN-based approach gives us good agreement with the DNS results in tension. However, when under compression, as the properties vary drastically between different $\boldsymbol{\xi}$'s, our NN-based approach starts to fail.

To enhance the fidelity of our NN-based approach for CMMs with complex pore shapes, it is useful to include a larger number of sampled $\boldsymbol{\xi}$'s in the database. This way, our neural networks can better capture the nonlinear dependence of $\overline{W}$ on the geometric parameters $\boldsymbol{\xi}$. We can validate our results by conducting similar uniaxial tests on CMMs with a homogeneous $\boldsymbol{\xi}$ that are different from those in the training data and compare with DNS results. Once that dependence is established with good accuracy, we can use the NN-based approach to predict the mechanical behavior of CMMs with an arbitrary uniform $\boldsymbol{\xi}$ or those with a non-uniform $\boldsymbol{\xi}$, which varies "smoothly enough".

However, for those $\boldsymbol{\xi}$'s that lead to large local variations and with characteristic lengths comparable to the size of RVE, our NN-based approach will fail since it breaks the basic assumption of our multiscale framework. For those CMMs, we have to rely on DNS or some other approaches. We must be cautious about the spatial inhomogeneity of the problem that we are solving: be it either in the geometric features or in the deformation field. For the multiscale approach to work, we must ensure that those variations have negligible influence on the macroscopic problems of interest. In principle, our NN-based multiscale approach only works when the size of the CMM is much larger than of the RVE and when there is no strong localization effect.

In Section 2.5 we have demonstrated that our NN-based scheme can capture the mechanics of moderately-sized CMM's with reasonable accuracy, but with significant computational savings over DNS. We anticipate further reduction for even larger CMM size. Nevertheless, it is worth noting that the most time-consuming step of our NN-based scheme is not solving the macroscopic problem using NN-based model, but the generation of the training data. As reported in Section 2.4.1, it takes about 20 hours to construct the database, although it is trivially parallelizable. Even though it is time-consuming, this generation step is only required once, and the resulting NN-based model can be used to solve multiple problems. Therefore, significant time can still be saved compared to other multiscale methods like FE^2 where micro- and marcoscopic problems have to be solved at the same time, as pointed out in Refs [36, 37].

Owing to its great computational efficiency, the NN-based approach proposed in this work can be particularly useful for solving design problems–the reverse engineering of CMMs to achieve specific mechanical properties. Compared with typical multiscale approaches, our approach provides a straightforward way of evaluating the necessary derivatives needed for the sensitivity analysis [130, 131], once the functional dependence of $\overline{W}$ on $\boldsymbol{\xi}$ are well established. Therefore we envision that those powerful tools used for structural topological optimization of composite materials can now be used for the design of CMMs under the NN-based framework. However, since our NN-based multiscale approach only works for CMMs with smoothly varying $\boldsymbol{\xi}$, proper regularization has to be imposed to ensure that the optimal solution that we obtained does not violate that condition. Other structural parameters can also be included for the design of CMM. For example, the porosity of the CMM, the arrangement of the pores (other than the square arrays used in this work), etc.

Differentiability and fast deployment are two important motivators for choosing neural networks for this work. Other than neural networks and polynomial regression in Section 2.4.2, there are many machine learning techniques available, each suitable for different types of problems. For example, decision trees [132] are a popular machine learning technique which offers great interpretability, however they are not typically differentiable. Gaussian process regression can be an alternative to neural networks for small datasets, and can sometimes outperform neural networks in terms of accuracy. Yet, there can be great computational cost for deployment once the dataset becomes large [133]. The choice of machine learning techniques largely depends on the types of problem and applications considered. Sometimes multiple machine learning techniques can be combined together for optimal performance. In this work, our aim is to demonstrate that neural networks are a promising tool for the design of CMMs. There are limitations – the need to generate a large dataset, the need for smoothly varying $\boldsymbol{\xi}$ – but we expect that future work may be able to compose the model we present with other machine learning techniques to relax these restrictions.

2.7 Conclusion

In this work, we proposed a neural network based multiscale computational scheme which can be used to predict the overall mechanical behavior of cellular mechanical metamaterials under large deformation. Our scheme adopts a data-driven approach to estimate the functional dependence of the effective strain energy density on complex cellular geometries and finite overall deformation. We first identify a proper RVE of the cellular solid and build an offline training data set by varying overall deformation as well as cellular geometries. The database is then used to train and validate a neural network model that can best represent the effective strain energy density as a function of cellular geometries and overall deformation. This neural network model is then treated as a coarse-grained constitutive model of the metamaterial and used to predict its overall mechanical behavior. Under certain conditions, our proposed scheme can significantly reduce the computational time compared with direct numerical simulation while achieving reasonable accuracy, especially when the metamaterial consists of a large number of unit cells. We discussed the limitations of the current scheme and emphasized the types of problems for which it is appropriate. Finally, we discussed the potential of using this to enable efficient rational design of metamaterials.

Chapter 3

Mapped Shape Optimization Method for Designing Cellular Mechanical Metamaterials

In this chapter, we propose the mapped shape optimization method that solves the inverse design problem of the same cellular mechanical metamaterials discussed in Chapter 2. The proposed method provides efficient computations of the sensitivity of objective functions to key geometric parameters using the adjoint method. The design framework is of general purpose, and we show several benchmark design problems. Notably, automatic differentiation technique, which is the workhorse for training neural networks in modern deep learning, is used in this work but with a different goal.

3.1 Introduction

Cellular mechanical metamaterials (CMMs) are common observation in nature, such as honeycombs, as well as in industry, such as microlattices. One key feature of these materials is that they can have unique mechanical properties such as high stiffness-

and/or strength-to-density ratio [90] and these properties are largely determined by their geometries . Traditionally, limited by the fabrication technologies, the study of this field tends to focus on several simple structures and under relatively small deformation [91]. Yet in the past decade, due to the advent of new technologies such as additive manufacturing [14, 13], sophisticated cellular architectures made of soft materials such as elastomers can now be easily realized in labs. Such metamaterials not only can go under large deformation without failure, they further open up the possibility of novel mechanical properties, including auxeticity, i.e. negative Poisson's ratio [2, 20, 92], shape morphing [3], tunable bandstructures [4] and energy absorption [5]. These properties have great potential for many exciting engineering applications, e.g., soft actuators, materials with *in situ* tunable functionalities, reusable energy-absorbing materials, etc. [6, 7, 10, 8, 12, 9].

The key step that can bridge the prospect of soft CMMs to real-world engineering application is the ability to design the desired type of CMMs for particular purposes. Since the unique properties of soft CMMs largely rely on their geometries, an efficient, yet reliable computational framework for shape optimization is the core of this ability. Techniques such as the density based method [46, 47], the evolutionary structural optimization (ESO) [48], and the level set method (LSM) [49, 50, 51]. While these methods have been demonstrated to be extremely powerful for traditional problems like stiffness optimization of linear elastic structures, they might not necessarily be best suited for optimization of CMMs. One of the reasons are the geometric and material nonlinearities involved in the forward problem. To achieve novel properties, CMMs tend to undergo large deformation, which is known to be a strongly nonlinear problem. Such large deformation is often accompanied by mechanical instabilities occurring at both the micro and macro scales, which adds extra difficulties to the problem. The forward problem itself is already non-trivial, not to mention solving the inverse problem, namely the design problem, which often involves iterations of solving the forward problems. Despite the complexity, several related works have made positive attempts to metamaterial shape optimization with large deformations [52, 53, 54].

In this work, we propose a mapped shape optimization method (MSOM) to tackle this problem, which is a numerically stable and easy-to-implement method that solves the design problems for CMMs efficiently. MSOM uses gradient-based optimization and the gradient of the objective function to the geometric parameters is computed using the adjoint method [55, 56]. The novelty of MSOM is that it uses a predefined referential configuration, a concept similar to what appears in Arbitrary Lagrangian–Eulerian (ALE) methods [57]. In the ALE description of motion, neither the material configuration nor the spatial configuration is taken as the reference, while a third, referential configuration is used for computations. In MSOM, we also have such a referential configuration where all numerical computations are performed. The choice of this referential configuration is arbitrary, and we fix it. Therefore, the computational mesh can stay the same throughout the optimization loops.

Compared with classical shape optimization methods, MSOM has several advantages when applied to the optimization of CMMs. MSOM is not a density-based method, hence to avoid dealing with inherent difficulties like the "checkerboard" problem [58]. MSOM shares certain similarity with the level set method in the sense that they both rely on the concept of "shape derivative" [59]. The dichotomy of Eulerian and Lagrangian specifications divides the level set methods into two classes: the Eulerian shape capturing method with a fixed mesh [49, 50, 60]; the Lagrangian shape tracking method with a moving mesh to represent the shape [61, 62]. MSOM can be considered as a third paradigm that uses a referential description. In contrast to the Eulerian level set method, MSOM avoids the difficulties implied by stable and accurate evolutions of the level set function; compared to the Lagrangian level set method, MSOM saves the numerical complications of modifying the mesh object since it uses a fixed mesh.

The chapter is organized as follows. Section 3.2 introduces the geometric setting of our problem as well as the fundamentals of finite-deformation elasticity. Section 3.3 focuses on the mathematical foundation and detailed procedures of MSOM. In Section 3.4, we used three important types of numerical examples to demonstrate the advantage of our method. They include the optimization of the negative Poisson's ratio, controlling the instability point, as well as the band-structures. We then conclude in Section 3.5.

3.2 Cellular Mechanical Metamaterials

The main goal of this section is to introduce the subject of this research: cellular mechanical metamaterials (CMMs). We will explain the the geometric parametrization of the structure, the material model, and the boundary value problem involved. The background knowledge about CMMs forms the prerequisite for the discussions of the optimization problems introduced later.

3.2.1 Problem geometry

To connect with previous related work [20, 92, 87], we study a specific class of CMMs that are elastomers containing periodic pores. It was shown that by changing the pore shapes, the CMMs can exhibit a wide range of interesting mechanical behaviors. In polar coordinates, the contours of these pores are parametrized as

$$r(\theta) = r_0(1 + \xi_1 \cos(4\theta) + \xi_2 \cos(8\theta)), \tag{3.1}$$

where r and θ are polar radius and polar angle respectively, ξ_1 and ξ_2 are two parameters that control the shape, and r_0 is given by

$$r_0 = \frac{L_0\sqrt{2\phi_0}}{\sqrt{\pi(2+\xi_1^2+\xi_2^2)}}, \tag{3.2}$$

with L_0 being the unit cell length and ϕ_0 being the initial porosity. In this work, the collection of

$$\boldsymbol{\alpha} = (\xi_1, \xi_2, \phi_0) \tag{3.3}$$

are the design parameters upon which we will perform optimization. For visualization, a family of various pore shapes are illustrated in Fig. 3.1 with ξ_1 ranging from -0.2 to 0, ξ_2 ranging from -0.1 to 0.1, and ϕ_0 fixed to be 0.5.

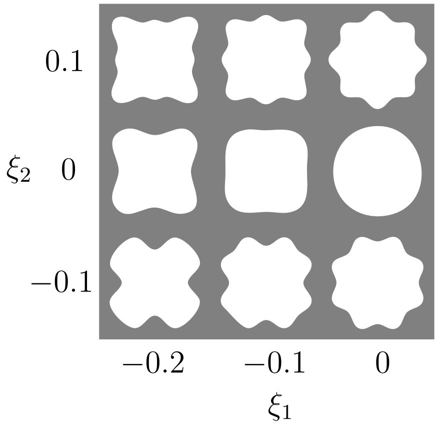


Figure 3.1: A map from the design parameters to the pore shapes indicated by Eq. (3.1). Porosity ϕ_0 is fixed to be 0.5.

3.2.2 Finite deformation elasticity

The mechanical behaviors of the CMMs are assumed to follow finite strain theory. In an unstressed state, the body occupies a region $\mathbb{B}_{\boldsymbol{X}} \subset \mathbb{R}^3$ (material/Lagrangian configuration). Upon mechanical loading, the body deforms and occupies a different region $\mathbb{B}_{\boldsymbol{x}} \subset \mathbb{R}^3$ (spatial/Eulerian configuration). Deformation can therefore be described as a mapping: $\boldsymbol{\varphi} : \mathbb{B}_{\boldsymbol{X}} \to \mathbb{B}_{\boldsymbol{x}}$, which maps any material point $\boldsymbol{X} \in \mathbb{B}_{\boldsymbol{X}}$ to its counterpart $\boldsymbol{x} \in \mathbb{B}_{\boldsymbol{x}}$, i.e. $\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X})$. The corresponding displacement field is defined as $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$ and the deformation gradient $\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}$.

For a hyperelastic material, the constitutive model can be defined by a strain energy density function (per volume) W, which depends on $\boldsymbol{F}$. The first Piola-Kirchhoff stress $\boldsymbol{P}$ is given by $\boldsymbol{P} = \frac{\partial W}{\partial \boldsymbol{F}}$. The displacement field $\boldsymbol{u}$ at static equilibrium can be determined by solving for the boundary value problem (in the absence of body force)

$$\nabla_{\boldsymbol{X}} \cdot \boldsymbol{P} = \boldsymbol{0} \quad \text{in } \mathbb{B}_{\boldsymbol{X}}, \tag{3.4}$$

where the divergence operator is in the material configuration.

In this work, a Neo-Hookean material model is used for the strain energy density function W

$$W(\boldsymbol{F}) = \frac{\mu}{2}(J^{-2/3} I_1 - 3) + \frac{\kappa}{2}(J-1)^2, \tag{3.5}$$

where $J = \det(\boldsymbol{F}),\ I_1 = \operatorname{tr}(\boldsymbol{C});\ \mu = \frac{E}{2(1+\nu)}$ and $\kappa = \frac{E}{3(1-2\nu)}$ denote the initial shear and bulk moduli, respectively, E and ν being the material's Young's modulus and Poisson's ratio. The above W is commonly used to model isotropic elastomers that are almost incompressible. We adopt a plane strain setting here, i.e., $u_i = u_i(X_1, X_2),\ i = 1, 2$ and $u_3 = 0$.

3.2.3 Boundary conditions

The nonlinear problem posed in Eq. (3.4) can be numerically discretized and solved for a finite size structure, e.g., the (leftmost) 6 × 6 elastomeric matrix in Fig. 3.2.

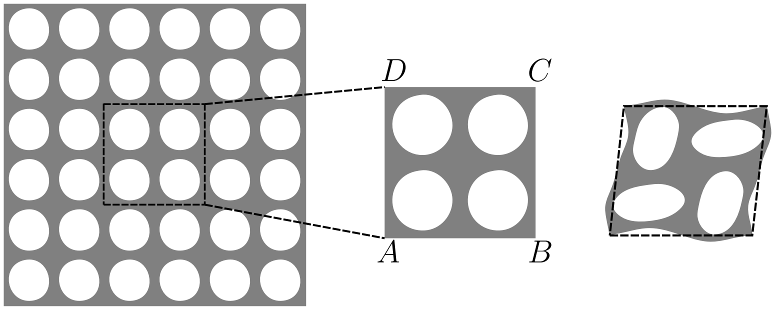


Figure 3.2: A 2x2 RVE taken from a cellular porous structure with repeating units. Periodic boundary conditions are applied to the RVE. Here we specify a macroscopic deformation gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} 0 & 0.1 \\ 0 & -0.1 \end{smallmatrix}\right]$. The deformed configuration (rightmost) shows the resulted displacement field of this RVE: $\boldsymbol{u} = \overline{\boldsymbol{H}} \cdot \boldsymbol{X} + \boldsymbol{u}^{\star}$ with $\boldsymbol{u}^{\star}$ being the periodic fluctuation. The rightmost dashed profile shows the deformation corresponding to the applied mean displacement $\overline{\boldsymbol{u}} = \overline{\boldsymbol{H}} \cdot \boldsymbol{X}$.

However, to reduce computational cost and avoid spurious boundary effects, we focus exclusively on studying a Representative Volume Element (RVE). The choice of RVE must properly reflect local properties in the sense that it can be treated as a material point when considering overall macroscopic mechanical responses. Following previous studies, we choose a 2×2 array of unit cells to be our RVE, as shown of the middle structure $ABCD$ in Fig. 3.2.

The RVE is subject to a macroscopic deformation gradient $\overline{\boldsymbol{F}}$ condition such that the displacement field $\boldsymbol{u}$ is decomposed into a macroscopic (overall) and a microscopic (fluctuation) part $\boldsymbol{u}^{\star}$:

$$\boldsymbol{u}(\boldsymbol{X}) = (\overline{\boldsymbol{F}} - \boldsymbol{I}) \cdot \boldsymbol{X} + \boldsymbol{u}^{\star}(\boldsymbol{X}) = \overline{\boldsymbol{H}} \cdot \boldsymbol{X} + \boldsymbol{u}^{\star}(\boldsymbol{X}), \tag{3.6}$$

where $\overline{\boldsymbol{H}}$ is the macroscopic displacement gradient and $\overline{\boldsymbol{H}} = \overline{\boldsymbol{F}} - \boldsymbol{I}$. The macroscopic deformation gradient $\overline{\boldsymbol{F}}$ (or the macroscopic displacement gradient $\overline{\boldsymbol{H}}$) is uniform all across the RVE. When specifying the macroscopic loading conditions, one can choose to specify either $\overline{\boldsymbol{F}}$ or $\overline{\boldsymbol{H}}$. In fulfilling the celebrated Hill-Mandel condition [95], periodic boundary conditions are applied for fluctuation displacements such that $\boldsymbol{u}^\star_{AD} = \boldsymbol{u}^\star_{BC}$ and $\boldsymbol{u}^\star_{AB} = \boldsymbol{u}^\star_{DC}$; opposite conditions for tractions such that $\boldsymbol{t}_{AD} = -\boldsymbol{t}_{BC}$ and $\boldsymbol{t}_{AB} = -\boldsymbol{t}_{DC}$.

3.3 Mapped Shape Optimization Method

In this section, we propose the mapped shape optimization method (MSOM) and explain it in detail. We first establish a differentiable chain from the pore shape parameter $\boldsymbol{\alpha}$ to a certain objective functional $\mathcal{J}$. The computations from a given pore shape $\boldsymbol{\alpha}$ to the objective $\mathcal{J}$ is referred to as the *forward problem*. We wish to compute the total gradient of $\mathcal{J}$ to $\boldsymbol{\alpha}$ by means of the adjoint method, enabling efficient gradient-based optimization algorithms for the *inverse problem*. The mathematical structure of the inverse problem falls into the category of PDE-constrained optimization. However, the change of the pore shape indicates a change of the computational domain that often needs re-meshing, breaking the differentiability of the forward chain. MSOM is developed to overcome this difficulty by introducing a fixed referential domain and mapping all computations to this referential domain, therefore avoiding an actual change of the computational domain to allow for adjoint optimization.

3.3.1 The referential configuration

We have introduced the finite strain problem in the total Lagrangian formulation (see Eq. (3.4)). The classic deformation map $\boldsymbol{\varphi}$ connects a material point $\boldsymbol{X}$ to a spatial point $\boldsymbol{x}$. For MSOM, we propose a third configuration $\mathbb{B}_{\boldsymbol{\chi}}$, which we refer to as referential configuration. We define the shape map $\boldsymbol{\psi} : \mathbb{B}_{\boldsymbol{\chi}} \to \mathbb{B}_{\boldsymbol{X}}$, which maps a referential point $\boldsymbol{\chi} \in \mathbb{B}_{\boldsymbol{\chi}}$ to a material point $\boldsymbol{X} \in \mathbb{B}_{\boldsymbol{X}}$, i.e. $\boldsymbol{X} = \boldsymbol{\psi}(\boldsymbol{\chi})$. We also define the total map $\boldsymbol{\phi} : \mathbb{B}_{\boldsymbol{\chi}} \to \mathbb{B}_{\boldsymbol{x}}$ that maps a referential point $\boldsymbol{\chi} \in \mathbb{B}_{\boldsymbol{\chi}}$ to a spatial point $\boldsymbol{x} \in \mathbb{B}_{\boldsymbol{x}}$, i.e. $\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{\chi})$. Naturally, we have $\boldsymbol{\phi} = \boldsymbol{\varphi} \circ \boldsymbol{\psi}$. Fig. 3.3 shows the three domains and the one-to-one transformations relating the configurations.

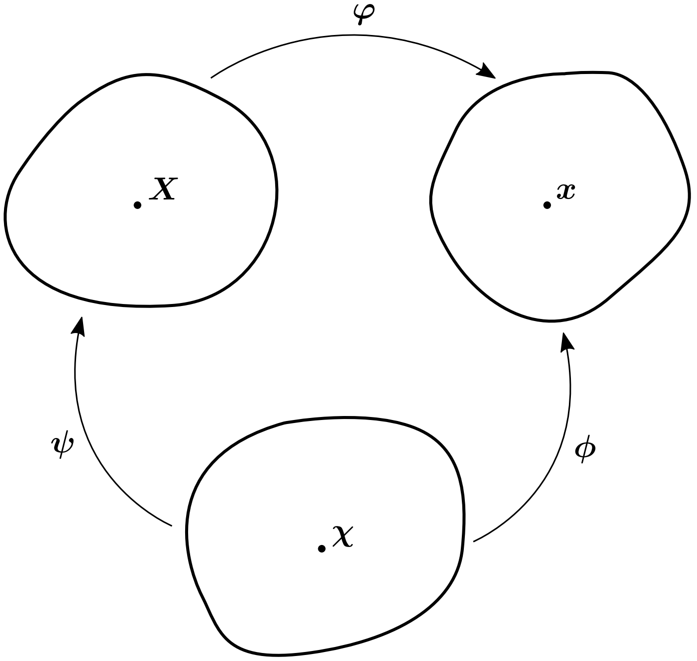


Figure 3.3: The referential configuration, the material configuration, and the spatial configuration with maps between them.

It is clear that for different pore shapes, the material domains $\mathbb{B}_{\boldsymbol{X}}$ must change. However, the referential domain $\mathbb{B}_{\boldsymbol{\chi}}$ can be fixed. This allows us to perform shape optimization without actually changing the shape of computational domains, i.e., we only need to solve the problems in the same, fixed referential configuration. This idea is reflected in Fig. 3.4, where we show the three configurations for a typical RVE. In this case, the referential domain $\mathbb{B}_{\boldsymbol{\chi}}$ is set to be a square with four identical circular pores. The material domain $\mathbb{B}_{\boldsymbol{X}}$ corresponds to a choice of $\boldsymbol{\alpha} = (\xi_1, \xi_2, \phi_0) = (-0.05, -0.05, 0.55)$. The spatial domain $\mathbb{B}_{\boldsymbol{x}}$ shows the deformed body with an imposed macroscopic displacement gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} 0 & 0.1 \\ 0 & -0.1 \end{smallmatrix}\right]$.

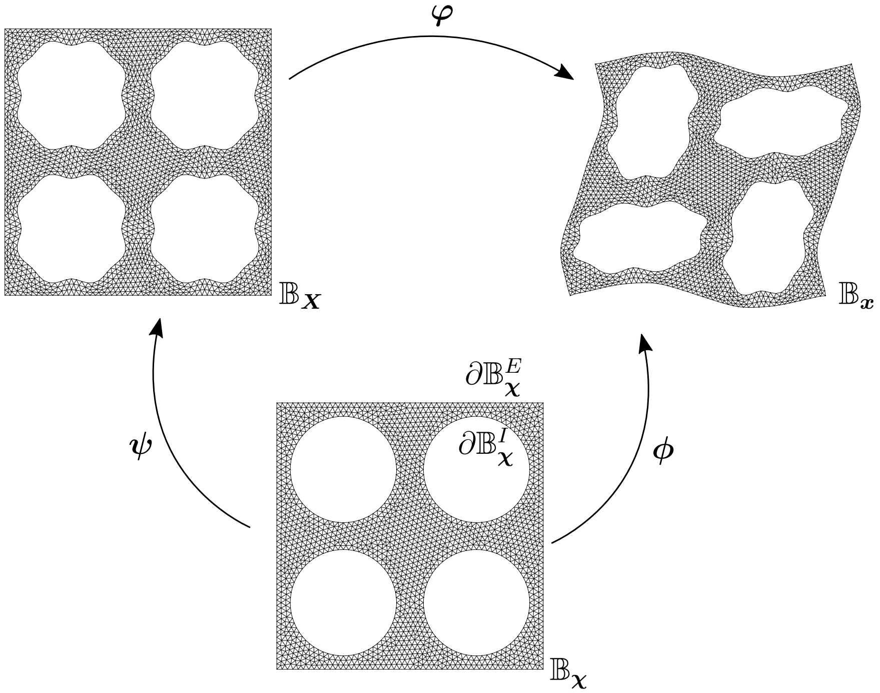

$\boldsymbol{\varphi}$
$\mathbb{B}_{\boldsymbol{X}}$
$\mathbb{B}_{\boldsymbol{x}}$
$\partial\mathbb{B}^E_{\boldsymbol{\chi}}$
$\partial\mathbb{B}^I_{\boldsymbol{\chi}}$
$\boldsymbol{\psi}$
$\boldsymbol{\phi}$
$\mathbb{B}_{\boldsymbol{\chi}}$

Figure 3.4: The referential domain $\mathcal{B}_{\boldsymbol{\chi}}$, the material domain $\mathbb{B}_{\boldsymbol{X}}$, and the spatial domain $\mathbb{B}_{\boldsymbol{x}}$.

3.3.2 The forward problem

Conceptually, there are two major steps in the forward problem. The first step is to compute the shape map $\boldsymbol{\psi}$ given the pore shape parameters $\boldsymbol{\alpha}$. Once $\boldsymbol{\psi}$ is known, the second step is to solve for problems of interest bearing the shape, such as to find the deformation map $\boldsymbol{\varphi}$ subject to certain loading conditions. Throughout the work, the formulation of the first step is always the same, yet the formulation of the second step depends on specific problems.

Step 1 To solve for $\boldsymbol{\psi}$ given $\boldsymbol{\alpha}$, we formulate a pseudo linear elasticity problem, a similar approach being found in [134]. The problem states that find shape displacement $\boldsymbol{s}(\boldsymbol{\chi}) := \boldsymbol{X} - \boldsymbol{\chi} = \boldsymbol{\psi}(\boldsymbol{\chi}) - \boldsymbol{\chi}$ such that

$$\begin{aligned} \nabla_{\boldsymbol{\chi}} \cdot \boldsymbol{\sigma} = \boldsymbol{0} \quad & \text{in } \mathbb{B}_{\boldsymbol{\chi}}, \\ \boldsymbol{s} = \boldsymbol{0} \quad & \text{on } \partial\mathbb{B}^E_{\boldsymbol{\chi}}, \\ \boldsymbol{s} = \boldsymbol{b} \quad & \text{on } \partial\mathbb{B}^I_{\boldsymbol{\chi}}, \end{aligned} \tag{3.7}$$

where

$$\begin{aligned} \boldsymbol{\sigma} &= 2\mu_p \boldsymbol{\varepsilon} + \lambda_p \mathrm{tr}(\boldsymbol{\varepsilon}) \boldsymbol{I} \\ \boldsymbol{\varepsilon} &= \frac{1}{2}(\nabla_{\boldsymbol{\chi}} \boldsymbol{s} + \nabla_{\boldsymbol{\chi}} \boldsymbol{s}^\top) \end{aligned} \tag{3.8}$$

with μ_p and λ_p being the pseudo Lamé parameters. Note that the dependency of $\boldsymbol{\alpha}$ enters the equations in $\boldsymbol{b}$ as part of the Dirichlet boundary conditions. We have fixed $\boldsymbol{s}$ to be zero on the exterior boundaries $\partial\mathbb{B}^E_{\boldsymbol{\chi}}$ (see Fig. 3.4), while on the interior boundaries $\partial\mathbb{B}^E_{\boldsymbol{\chi}}$ the shape displacement $\boldsymbol{s}$ must conform to the pore shape described by $\boldsymbol{\alpha}$. The choice of μ_p and λ_p are not necessarily to be constants. Rather, they can be fields as $\mu_p(\boldsymbol{\chi})$ and $\lambda_p(\boldsymbol{\chi})$. The purpose of tuning these two parameters is to generate a relatively smooth $\boldsymbol{\psi}$, as shown of $\mathbb{B}_{\boldsymbol{X}}$ in Fig. 3.4.

One convenient way to capture the link between $\boldsymbol{b}$ and $\boldsymbol{s}$ is through weakly imposing boundary conditions of $\boldsymbol{s} = \boldsymbol{b}$ on $\partial\mathbb{B}^I_{\boldsymbol{\chi}}$ via Nitsche's method [135, 136]. Consequently, we state the weak form of Eq. (3.7) as follows: find $\boldsymbol{s} \in \mathcal{V}$, such that $\forall \boldsymbol{w} \in \mathcal{V}$,

$$\begin{aligned} \Pi\big([\boldsymbol{s};\boldsymbol{w}],[\boldsymbol{b}]\big) = \int_{\mathbb{B}_{\boldsymbol{\chi}}} \boldsymbol{\sigma}(\boldsymbol{s}) : \nabla_{\boldsymbol{\chi}} \boldsymbol{w} \, \mathrm{d}\boldsymbol{\chi} - \int_{\partial\mathbb{B}^I_{\boldsymbol{\chi}}} \boldsymbol{w} \cdot \boldsymbol{\sigma}(\boldsymbol{s}) \cdot \boldsymbol{n} \, \mathrm{d}\boldsymbol{S} - \int_{\partial\mathbb{B}^I_{\boldsymbol{\chi}}} \boldsymbol{s} \cdot \boldsymbol{\sigma}(\boldsymbol{w}) \cdot \boldsymbol{n} \, \mathrm{d}\boldsymbol{S} + \\ \eta \int_{\partial\mathbb{B}^I_{\boldsymbol{\chi}}} \boldsymbol{s} \cdot \boldsymbol{w} \, \mathrm{d}\boldsymbol{S} + \int_{\partial\mathbb{B}^I_{\boldsymbol{\chi}}} \boldsymbol{b} \cdot \boldsymbol{\sigma}(\boldsymbol{w}) \cdot \boldsymbol{n} \, \mathrm{d}\boldsymbol{S} - \eta \int_{\partial\mathbb{B}^I_{\boldsymbol{\chi}}} \boldsymbol{b} \cdot \boldsymbol{w} \, \mathrm{d}\boldsymbol{S} = 0, \end{aligned} \tag{3.9}$$

where

$$\mathcal{V} = \big\{ \boldsymbol{s} \in H^1(\mathbb{B}_{\boldsymbol{\chi}}) \; \big| \; \boldsymbol{s} = \boldsymbol{0} \text{ on } \partial\mathbb{B}^E_{\boldsymbol{\chi}} \big\}, \tag{3.10}$$

is the set of admissible field, $\boldsymbol{n}$ is the outward normal, and η is a large constant. The weak form is linear in $\boldsymbol{s}$ and is ready for numerical discretization.

We us the finite element method for the solution. Let $\boldsymbol{s}^h(\boldsymbol{\chi}) = \sum_n^N s_n \boldsymbol{n}_n(\boldsymbol{\chi})$ be the discretized solution of $\boldsymbol{s}(\boldsymbol{\chi})$ where s_n is the n^{th} degree of freedom and $\boldsymbol{n}_n(\boldsymbol{\chi})$ is the basis function. Let $\boldsymbol{S} = \{s_n\} \in \mathbb{R}^N$ be the vector of degrees of freedom. Similarly, let $\boldsymbol{b}^h(\boldsymbol{\chi}) = \sum_m^M b_m \boldsymbol{n}_m(\boldsymbol{\chi})$ be the discretization of $\boldsymbol{b}(\boldsymbol{\chi})$, and $\boldsymbol{B} = \{b_m\} \in \mathbb{R}^M$. Eq. (3.9) can be discretized into

$$\begin{aligned} \boldsymbol{G}(\boldsymbol{S}, \boldsymbol{B}) &= \boldsymbol{0}, \\ \boldsymbol{G} : \mathbb{R}^N \times \mathbb{R}^M &\rightarrow \mathbb{R}^M. \end{aligned} \tag{3.11}$$

We leave out the details of assembling $\boldsymbol{G}$ for the sake of brevity.

Step 2 This step is problem dependent. For illustration of the method, we consider a model problem. Let us revisit the problem posed in Eq. (3.4). Besides solving the strong problem, the displacement field $\boldsymbol{u}$ can alternatively be determined by minimization of the following functional:

$$\mathcal{E}([\boldsymbol{u}]) = \int_{\mathbb{B}_{\boldsymbol{X}}} W(\boldsymbol{F}) \; \mathrm{d}\boldsymbol{X}. \tag{3.12}$$

To get the weak form, we take the variation of $\mathcal{E}$ with respect to $\boldsymbol{u}$ in the direction of $\boldsymbol{v}$.

$$\delta\mathcal{E}([\boldsymbol{u}; \boldsymbol{v}]) = \int_{\mathbb{B}_{\boldsymbol{X}}} \boldsymbol{P}(\boldsymbol{F}) : \nabla_{\boldsymbol{X}} \boldsymbol{v} \; \mathrm{d}\boldsymbol{X} = 0. \tag{3.13}$$

To build the link between $\boldsymbol{s}$ and $\boldsymbol{u}$, MSOM requires the integral in Eq. (3.13) to be transformed to the referential domain $\mathbb{B}_{\boldsymbol{\chi}}$. The problem states that find $\boldsymbol{u} \in \mathcal{U}$, such that $\forall \boldsymbol{v} \in \mathcal{U}$,

$$\delta\mathcal{E}([\boldsymbol{u};\boldsymbol{v}],[\boldsymbol{s}]) = \int_{\mathbb{B}_{\boldsymbol{\chi}}} \boldsymbol{P}\Big(\underbrace{\nabla_{\boldsymbol{\chi}}\boldsymbol{x}\cdot\big(\nabla_{\boldsymbol{\chi}}\boldsymbol{X}\big)^{-1}}_{\boldsymbol{F}}\Big) : \underbrace{\nabla_{\boldsymbol{\chi}}\boldsymbol{v}\cdot\big(\nabla_{\boldsymbol{\chi}}\boldsymbol{X}\big)^{-1}}_{\nabla_{\boldsymbol{X}}\boldsymbol{v}}\underbrace{\det\big(\nabla_{\boldsymbol{\chi}}\boldsymbol{X}\big)\ \mathrm{d}\boldsymbol{\chi}}_{\mathrm{d}\boldsymbol{X}} = 0, \tag{3.14}$$

where

$$\mathcal{U} = \big\{\boldsymbol{u} \in H^1(\mathbb{B}_{\boldsymbol{\chi}}) \ \big|\ \boldsymbol{u} \text{ periodic on } \partial\mathbb{B}^E_{\boldsymbol{\chi}}\big\}. \tag{3.15}$$

The transformation is nothing but a change of variable, where we have used

$$\begin{aligned} F_{ij} &= \frac{\partial x_i}{\partial X_j} = \frac{\partial x_i}{\partial \chi_k}\frac{\partial \chi_k}{\partial X_j} = \Big(\nabla_{\boldsymbol{\chi}}\boldsymbol{x}\cdot(\nabla_{\boldsymbol{\chi}}\boldsymbol{X})^{-1}\Big)_{ij}, \\ \mathrm{d}\boldsymbol{X} &= \det\big(\nabla_{\boldsymbol{\chi}}\boldsymbol{X}\big)\ \mathrm{d}\boldsymbol{\chi}. \end{aligned} \tag{3.16}$$

The dependency of $\boldsymbol{s}$ and $\boldsymbol{u}$ in Eq. (3.14) can be seen by replacing $\boldsymbol{X} = \boldsymbol{\chi} + \boldsymbol{s}$ and $\boldsymbol{x} = \boldsymbol{\chi} + \boldsymbol{s} + \boldsymbol{u}$. We emphasize that the shape displacement $\boldsymbol{s}$ has been explicitly incorporated into the weak form of Eq. (3.14), in contrast to Eq. (3.13), a fact that makes optimization possible.

For finite element discretization, let $\boldsymbol{u}^h(\boldsymbol{\chi}) = \sum_l^L s_l \boldsymbol{n}_l(\boldsymbol{\chi})$ be the discretized solution. Let $\boldsymbol{U} = \{u_l\} \in \mathbb{R}^L$ be the vector of degrees of freedom. Eq. (3.14) can be discretized into

$$\begin{aligned} &\boldsymbol{K}(\boldsymbol{U},\boldsymbol{S}) = \boldsymbol{0}, \\ &\boldsymbol{K} : \mathbb{R}^L \times \mathbb{R}^N \to \mathbb{R}^L. \end{aligned} \tag{3.17}$$

In `FEniCS`, codes are written in a concise and near-math fashion. With the energy functional (Eq. (3.12)), `FEniCS` is able to derive its first variation (Eq. (3.13)) automatically. This automatic differentiation feature is enabled due to the `UFL` component [128] of the package. The procedures for linearization of Eq. (3.17) for Newton's method to solve are also automated, whose details we omit. It is noted that the CMMs may undergo extreme deformations, where Newton's method often fails to converge. Therefore alternative methods must be applied such as dynamic relaxation [137] or arc-length method [138]. In this work, we adopt the dynamic relaxation solver originally developed in [139].

3.3.3 Adjoint optimization

To this point, we state the optimization problem that is the main focus of this work, and we use the adjoint method to compute the gradient for efficient optimization algorithms. Formally, we aim to solve the following optimization problem with $\mathscr{J}$ being some objective functional depending on $\boldsymbol{u}$:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^K} \mathscr{J}(\boldsymbol{u}), \tag{3.18}$$

where the design parameter $\boldsymbol{\alpha}$ is implicitly linked to $\boldsymbol{u}$ by the PDEs described in the forward problem. The discretized counterpart of the optimization problem can be stated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^K} J(\boldsymbol{U}). \tag{3.19}$$

Here is a chain of the dependency relationship of the involved variables:

$$\text{Original forward problem: } \boldsymbol{\alpha} \rightarrow \boldsymbol{b} \rightarrow \boldsymbol{s} \rightarrow \boldsymbol{u} \rightarrow \mathcal{J}, \tag{3.20a}$$

$$\text{Discretized forward problem: } \boldsymbol{\alpha} \rightarrow \boldsymbol{B} \rightarrow \boldsymbol{S} \rightarrow \boldsymbol{U} \rightarrow J. \tag{3.20b}$$

The maps from $\boldsymbol{\alpha}$ to $\boldsymbol{b}$ and from $\boldsymbol{u}$ to $\mathcal{J}$ are explicit, hence the derivatives are straightforward to compute. In contrast, the maps from $\boldsymbol{b}$ to $\boldsymbol{s}$ and from $\boldsymbol{s}$ to $\boldsymbol{u}$ are implicitly defined by the forward PDEs mentioned in the two major steps.

We define the reduced functional $\widehat{J}(\boldsymbol{\alpha}) = J(\boldsymbol{U})$ to be the objective function directly represented by $\boldsymbol{\alpha}$. The goal is to compute the total gradient $\frac{\mathrm{d}\widehat{J}}{\mathrm{d}\boldsymbol{\alpha}}$. We first derive the tangent linear equations from Eq. (3.11) and Eq. (3.17):

$$\frac{\partial \boldsymbol{G}(\boldsymbol{S},\boldsymbol{B})}{\partial \boldsymbol{S}}\frac{\mathrm{d}\boldsymbol{S}}{\mathrm{d}\boldsymbol{B}} = -\frac{\partial \boldsymbol{G}(\boldsymbol{S},\boldsymbol{B})}{\partial \boldsymbol{B}}, \tag{3.21a}$$

$$\frac{\partial \boldsymbol{K}(\boldsymbol{U},\boldsymbol{S})}{\partial \boldsymbol{U}}\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}\boldsymbol{S}} = -\frac{\partial \boldsymbol{K}(\boldsymbol{U},\boldsymbol{S})}{\partial \boldsymbol{S}}. \tag{3.21b}$$

Therefore

$$\frac{\mathrm{d}\widehat{J}}{\mathrm{d}\boldsymbol{\alpha}} = \underbrace{\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{U}}}_{1\times L}\underbrace{\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}\boldsymbol{S}}}_{L\times N}\underbrace{\frac{\mathrm{d}\boldsymbol{S}}{\mathrm{d}\boldsymbol{B}}}_{N\times M}\underbrace{\frac{\mathrm{d}\boldsymbol{B}}{\mathrm{d}\boldsymbol{\alpha}}}_{M\times K} = \underbrace{\overbrace{\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{U}}\Big(\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{U}}\Big)^{-1}}^{\boldsymbol{\lambda}_u^*}\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{S}}\Big(\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{S}}\Big)^{-1}}_{\boldsymbol{\lambda}_s^*}\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{B}}\frac{\mathrm{d}\boldsymbol{B}}{\mathrm{d}\boldsymbol{\alpha}} \tag{3.22}$$

can be computed efficiently with the adjoint equations

$$\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{U}}^*\boldsymbol{\lambda}_u = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{U}}^*, \tag{3.23a}$$

$$\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{S}}^*\boldsymbol{\lambda}_s = \frac{\partial \boldsymbol{K}}{\partial \boldsymbol{S}}^*\boldsymbol{\lambda}_u. \tag{3.23b}$$

We perform adjoint optimization based on `dolfin-adjoint` [140], an open-source library that interfaces with `FEniCS` for deriving the discrete adjoint and tangent linear models from a forward model.

3.3.4 Verification of the total gradients

We use Taylor test to verify the correctness of the computed gradient. Given a perturbation $\delta\boldsymbol{\alpha}$, Taylor's theorem states that for residuals to decrease, the rate of convergence is 1 by a zeroth-order expansion

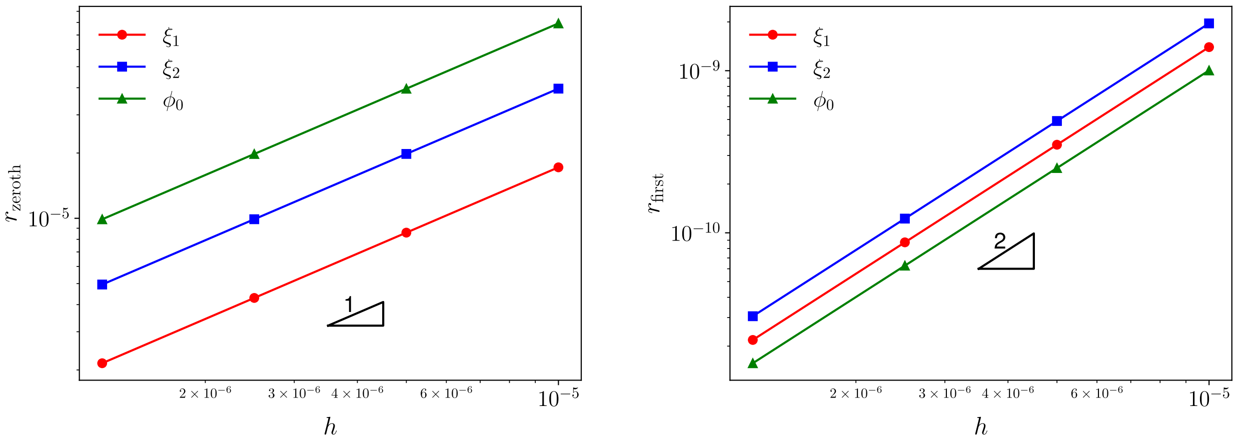
$$r_{\text{zeroth}} = \left|\widehat{J}(\boldsymbol{\alpha} + h\delta\boldsymbol{\alpha}) - \widehat{J}(\boldsymbol{\alpha})\right| \to 0 \text{ at } \mathcal{O}(h), \tag{3.24}$$

and the convergence rate is 2 by a first-order expansion

$$r_{\text{first}} = \left|\widehat{J}(\boldsymbol{\alpha} + h\delta\boldsymbol{\alpha}) - \widehat{J}(\boldsymbol{\alpha}) - h\frac{\mathrm{d}\widehat{J}}{\mathrm{d}\boldsymbol{\alpha}} \cdot \delta\boldsymbol{\alpha}\right| \to 0 \text{ at } \mathcal{O}(h^2). \tag{3.25}$$

We conduct a case study to pass the Taylor test. For the forward problem, we follow the parameters used in Fig. 3.4 and choose the design variable $\boldsymbol{\alpha} = (\xi_1, \xi_2, \phi_0) = (-0.05, -0.05, 0.55)$. Then the shape displacement $\boldsymbol{s}$ can be determined, based on which we impose the macroscopic deformation gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} 0 & 0.1 \\ 0 & -0.1 \end{smallmatrix}\right]$ to compute the material displacement $\boldsymbol{u}$. For adjoint optimization, we define the objective functional to be the total stored strain energy $\mathcal{J}(\boldsymbol{u}) = \int_{\mathbb{B}_{\boldsymbol{X}}} W \ \mathrm{d}\boldsymbol{X}$. The optimization problem is understood as to find an optimal pore shape such that when the RVE is subject to a certain macroscopic deformation gradient condition, the stored elastic energy should be minimized.

We set $h = 10^{-5}, 5 \times 10^{-6}, 2.5 \times 10^{-6}, 1.25 \times 10^{-6}$, and $\boldsymbol{\alpha}$ to be along the directions of ξ_1, ξ_2, ϕ_0 separately. The convergence results are shown in Fig. 3.5. As expected, the residuals of the first order expansion decrease at a rate proportional to 2, which validates the total gradients.



(a) (b)

Figure 3.5: Convergence reports of (a) zeroth- and (b) first-order Taylor expansion.

3.4 Numerical Examples

The CMMs studied in this research have wide applications due to their unique mechanical properties. In this section, we apply MSOM to solve three representative optimization problems: designing metamaterials with negative Poisson's ratio, controlling critical load for buckling, and maximizing band gaps for phononic structures. Compared with traditional approaches that often perform heuristic search for optimal structures, MSOM guides the optimization with gradients and is much more efficient.

The presentations of the three numerical examples are structured to follow the same pattern. We first discuss the background and significance of the application. Then the forward problem is introduced. The final part is to bring up the optimization problem and use MSOM for solutions.

3.4.1 Negative Poisson's ratio

CMMs exhibit negative Poisson's ratio behavior when under compression [2, 20]. The auxetic behavior is triggered by an elastic instability that leads to a pattern transformation. It is uncovered that pore shapes play an important role in the mechanical response of the CMMs [92]. Yet, a sensitivity computation that quantitatively reflects how the auxetic behavior is affected by the change of pore shape is lacking. We address this question by formulating the problem into the framework of MSOM.

Consider imposing a fixed macroscopic displacement gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} -0.04 & 0 \\ 0 & -0.1 \end{smallmatrix}\right]$ condition to a RVE parameterized by $\boldsymbol{\alpha}$ and solving for the equilibrium displacement field $\boldsymbol{u}$. The forward problem follows the formulations of *Step 1* and *Step 2* in Section 3.3.2. It is useful to consider the macroscopic first Piola-Kirchhoff stress

$$\overline{\boldsymbol{P}} = V^{-1} \int_{\mathbb{B}_{\boldsymbol{X}}} \boldsymbol{P} \, \mathrm{d}\boldsymbol{X}, \tag{3.26}$$

where $V = 4L_0^2$ is the total volume for a RVE including the voids. We define the effective Poisson's ratio to be

$$\overline{\nu} = -\frac{\overline{H}_{11}}{\overline{H}_{22}}. \tag{3.27}$$

The effective Poisson's ratio only makes sense when $\overline{P}_{11}$ vanishes. Therefore, the optimization problem is formulated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^K} \overline{P}_{11}^2. \tag{3.28}$$

The optimization problem essentially seeks a solution (an optimal pore shape) that enables the RVE to have a prescribed effective Poisson's ratio $\overline{\nu} = -\frac{\overline{H}_{11}}{\overline{H}_{22}} = 0.4$. By MSOM, we obtain the gradient of the objective function, which is then passed to standard optimization algorithms L-BFGS-B [141, 142] with lower and upper bounds for design variables to avoid nonphysical solutions. The optimization results are demonstrated in Fig. 3.6 (b). As shown, the objective value decreases to almost zero in only several steps, indicating that a pore shape for $\overline{P}_{11} \approx 0$ has been found. The optimization step is counted each time L-BFGS-B queries a gradient computation.

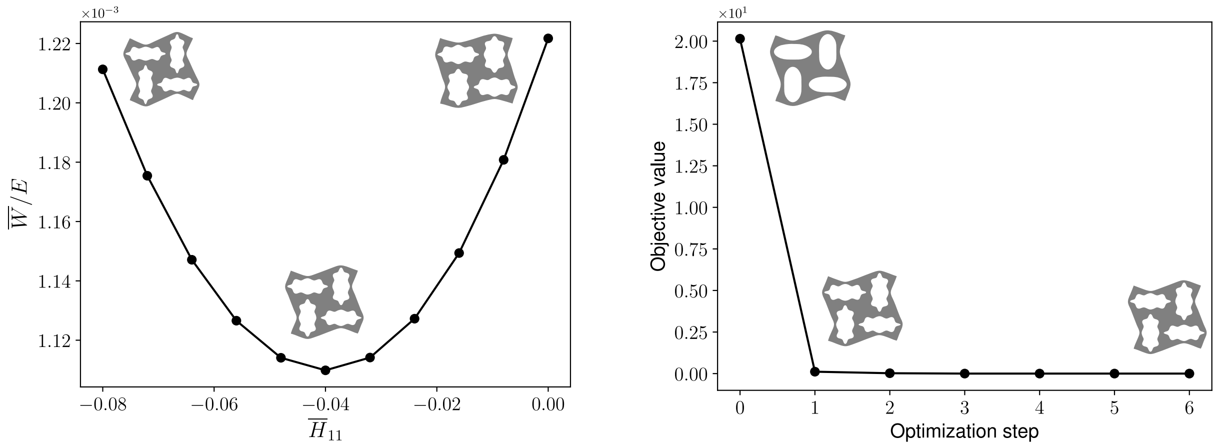
To view the results through a different lens, we compute the macroscopic strain energy density $\overline{W}$ via the average W over the RVE:

$$\overline{W} = V^{-1} \int_{\mathbb{B}_{\boldsymbol{X}}} W \ \mathrm{d}\boldsymbol{X}. \tag{3.29}$$

The macroscopic first Piola-Kirchhoff stress can alternatively be derived as

$$\overline{\boldsymbol{P}} = \frac{\partial \overline{W}}{\partial \overline{\boldsymbol{F}}}. \tag{3.30}$$

In our problem, it is expected that for the optimial RVE we have $\overline{P}_{11} = \frac{\partial \overline{W}}{\partial \overline{F}_{11}} \approx 0$ at $\overline{H}_{11} = -0.04$. Therefore we fix $\overline{H}_{12} = 0$, $\overline{H}_{21} = 0$, and $\overline{H}_{22} = -0.1$ to run the forward problems with varying $\overline{H}_{11}$ and show $\overline{W}$ in Fig. 3.6 (a). We see that $\overline{W}$ achieves a local extreme at $\overline{H}_{11} = -0.04$, hence by first order condition $\overline{P}_{11} \approx 0$, justifying our statement.



(a) (b)

Figure 3.6: Optimization results for a RVE with specified negative Poisson's ratio. (a) The normalized macroscopic strain energy density for the optimal RVE with varying $\overline{H}_{11}$. The three RVEs correspond to $\overline{H}_{11} = -0.08$, $\overline{H}_{11} = -0.04$, and $\overline{H}_{11} = 0$. (b) The optimization iterations. The three RVEs correspond to the first two iterations and the final iteration.

The energy density plot in Fig. 3.6 (a) shows that the optimal RVE achieves a local minimum at $\overline{H}_{11} = -0.04$ among all possible $\overline{H}_{11}$ values, which means that the

RVE is at a stable equilibrium under this macroscopic condition. It is of interest to test if a similar behavior exists when varying along the shear direction, i.e., fixing other components but adjusting $\overline{H}_{12}$. Therefore we consider imposing a macroscopic displacement gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} 0 & 0.3 \\ 0 & -0.125 \end{smallmatrix}\right]$ condition to a RVE. The optimization problem is formulated as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^K} \overline{P}_{12}^2. \tag{3.31}$$

The physical meaning of this problem is to find an optimal pore shape such that the RVE achieves a stable equilibrium at $\overline{H}_{12} = 0.3$. As shown in Fig. 3.7 (b), the objective value decreases to almost zero quickly. To validate the optimization results, we fix $\overline{H}_{11} = 0$, $\overline{H}_{21} = 0$, and $\overline{H}_{22} = -0.125$ to run the forward problems with varying $\overline{H}_{12}$ and show $\overline{W}$ in Fig. 3.7 (a). It is observed that $\overline{W}$ achieves a local minimum at $\overline{H}_{12} = 0.3$ among all possible $\overline{H}_{12}$ values, thereby validating the RVE is at a stable equilibrium.

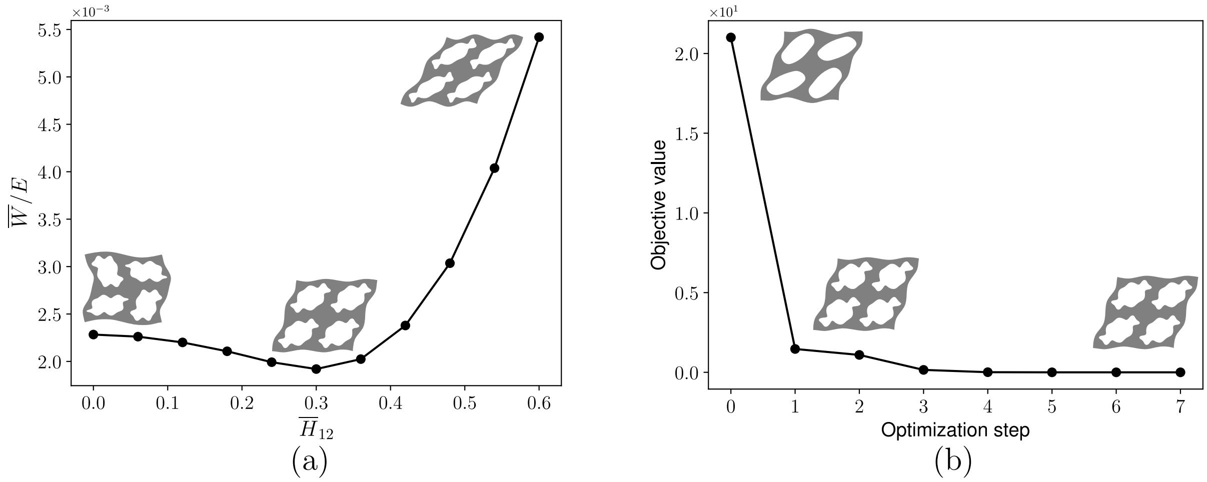


Figure 3.7: Optimization results for the shear test. (a) The normalized macroscopic strain energy density for the optimal RVE with varying $\overline{H}_{12}$. The three RVEs correspond to $\overline{H}_{12} = 0$, $\overline{H}_{12} = 0.3$, and $\overline{H}_{12} = 0.6$. (b) The optimization iterations. The three RVEs correspond to the first two iterations and the final iteration.

3.4.2 Buckling control

To understand the buckling behavior of CMMs under external loading, we investigate the stability of RVEs. Without loss of generality, we consider biaxial compression for RVEs. Specifically, we assume the macroscopic displacement gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} -\lambda & 0 \\ 0 & -\lambda \end{smallmatrix}\right]$, where $\boldsymbol{\lambda} \geq 0$ is the monotonically increasing load parameter. It is observed that when $\boldsymbol{\lambda}$ is small, the RVE achieves a stable equilibrium. With increased $\boldsymbol{\lambda}$, the geometric and material nonlinearities start to affect the stability of the system. Upon reaching a critical value of $\boldsymbol{\lambda}$, the RVE loses stable equilibrium and bifurcates to a certain mode for new stable equilibrium.

The onset of instabilities can be tracked with eigenfrequency analysis. Buckling occurs when an initial negative eigenvalue appears, indicating the system fails to maintain the stable equilibrium. To perform eigenfrequency analysis, we introduce a small time-dependent perturbation $\delta \boldsymbol{u}$ superimposed to a base state displacement $\boldsymbol{u} = \overline{\boldsymbol{H}} \cdot \boldsymbol{X}$. Rigorously speaking, the base displacement should satisfy the equilibrium condition. Here the condition is approximated, but for the sake of simplicity the method is still helpful for the understanding of material behaviors. The incremental displacement $\delta \boldsymbol{u}$ satisfies the governing equations:

$$\nabla_{\boldsymbol{X}} \cdot \delta \boldsymbol{P} = \rho_0 \frac{\partial^2 \delta \boldsymbol{u}}{\partial t^2}\bigg|_{\boldsymbol{X}} \quad \text{in } \mathbb{B}_{\boldsymbol{X}}, \tag{3.32}$$

where ρ_0 is the mass density in the material configuration, and $\delta \boldsymbol{P}$ is the increment of the first Piola-Kirchhoff stress due to $\delta \boldsymbol{u}$,

$$\delta \boldsymbol{P} = \mathbb{L} : \nabla_{\boldsymbol{X}} \delta \boldsymbol{u}, \tag{3.33}$$

where $\mathbb{L}$ is the fourth-order tangent modulus tensor whose components are given by $L_{ijkl} = \frac{\partial^2 W}{\partial F_{ij} \partial F_{kl}}$. We see that $\delta \boldsymbol{u}$ is a harmonic oscillation, i.e.,

$$\delta \boldsymbol{u} = \tilde{\boldsymbol{u}} \exp(-\mathrm{i}\omega t), \tag{3.34}$$

where $(\omega^2, \tilde{\boldsymbol{u}})$ is a pair of eigenvalue and eigenfunction to the eigenvalue problem represented by Eq. (3.32). The weak form states that find $\tilde{\boldsymbol{u}}$ such that $\forall \tilde{\boldsymbol{v}}$

$$\int_{\mathbb{B}_{\boldsymbol{X}}} \nabla_{\boldsymbol{X}} \tilde{\boldsymbol{v}} : \mathbb{L} : \nabla_{\boldsymbol{X}} \tilde{\boldsymbol{u}} \, \mathrm{d}\boldsymbol{X} = \rho_0 \omega^2 \int_{\mathbb{B}_{\boldsymbol{X}}} \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{v}} \, \mathrm{d}\boldsymbol{X}. \tag{3.35}$$

For adjoint optimization, the dependency of the shape displacement $\boldsymbol{s}$ must be explicit. Therefore we transform the weak form in Eq. (3.35) from the material domain $\mathbb{B}_{\boldsymbol{X}}$ to the referential domain $\mathbb{B}_{\boldsymbol{\chi}}$ to obtain

$$\int_{\mathbb{B}_{\boldsymbol{\chi}}} \nabla_{\boldsymbol{\chi}} \tilde{\boldsymbol{v}} \cdot \left(\nabla_{\boldsymbol{\chi}} \boldsymbol{X}\right)^{-1} : \mathbb{L} : \nabla_{\boldsymbol{\chi}} \tilde{\boldsymbol{u}} \cdot \left(\nabla_{\boldsymbol{\chi}} \boldsymbol{X}\right)^{-1} \det\!\left(\nabla_{\boldsymbol{\chi}} \boldsymbol{X}\right) \, \mathrm{d}\boldsymbol{\chi} = \rho_0 \omega^2 \int_{\mathbb{B}_{\boldsymbol{\chi}}} \tilde{\boldsymbol{u}} \cdot \tilde{\boldsymbol{v}} \det\!\left(\nabla_{\boldsymbol{\chi}} \boldsymbol{X}\right) \, \mathrm{d}\boldsymbol{\chi}. \tag{3.36}$$

The finite element discretization yields the following discrete eigenvalue problem:

$$\begin{aligned} \boldsymbol{A}(\boldsymbol{S})\boldsymbol{V} &= \omega^2 \boldsymbol{M}(\boldsymbol{S})\boldsymbol{V}, \\ \boldsymbol{A} &: \mathbb{R}^N \to \mathbb{R}^L \times \mathbb{R}^L, \\ \boldsymbol{M} &: \mathbb{R}^N \to \mathbb{R}^L \times \mathbb{R}^L, \end{aligned} \tag{3.37}$$

where $\boldsymbol{V} \in \mathbb{R}^L$ is the eigenvector paired with the eigenvalue ω^2. The generalized eigenvalue problem is solved with `SLEPc` [143], a software library for the solution of large scale sparse eigenvalue problems. The matrix $\boldsymbol{A}$ and $\boldsymbol{M}$ are real symmetric, hence the eigenvalues are all real numbers.

Consider a representative pore shape $\boldsymbol{\alpha} = (\xi_1, \xi_2, \phi_0) = (0, 0, 0.5)$. We load the RVE from $\lambda = 0$ to $\lambda = 0.05$ with a incremental step of 0.0025. For each loading step, we solve the eigenvalue problem posed in Eq. (3.37). The eigenvalues and eigenmodes are plotted with respect to load parameters in Fig. 3.8. As shown, we record three major eigenvalues with their eigenmodes. The eigenvalue represented by the blue curve is repeated and has two distinct eigenmodes. The eigenvalue by the red curve is the dominant one that first approaches zero at $\lambda_{\mathrm{cr}} \approx 0.0125$. The RVE therefore buckles towards the dominant eigenmode at this critical load.

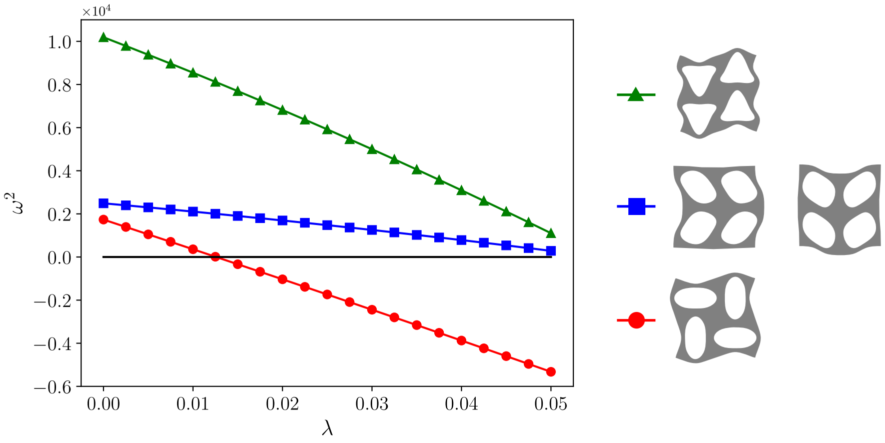


Figure 3.8: Numerical study of the instability of a RVE with a circular pore shape. The left plot shows the evolution of eigenvalues ω^2 at different levels of compression (increasing the load parameter λ) for the RVE with initial porosity. The right legends show the deformation of RVEs with corresponding eigenmodes.

The discussions above are on the forward problem. Namely, given the pore shape and loading conditions, compute the buckling load λ_{cr}. We remind the readers that forward problem actually contains two major steps. The first step solves for the shape displacement $\boldsymbol{s}$ that exactly follows *Step 1* in Section 3.3.2, so the formulation is omitted in this subsection. The second step is different from *Step 2* in Section 3.3.2. We have derived from Eq. (3.32) to Eq. (3.37) to present the formulation of the second step.

As the challenge to address, we aim to solve the following design problem: given a critical load parameter, find the pore shape that exactly buckles precisely at this loading condition. Formally, we propose the optimization problem

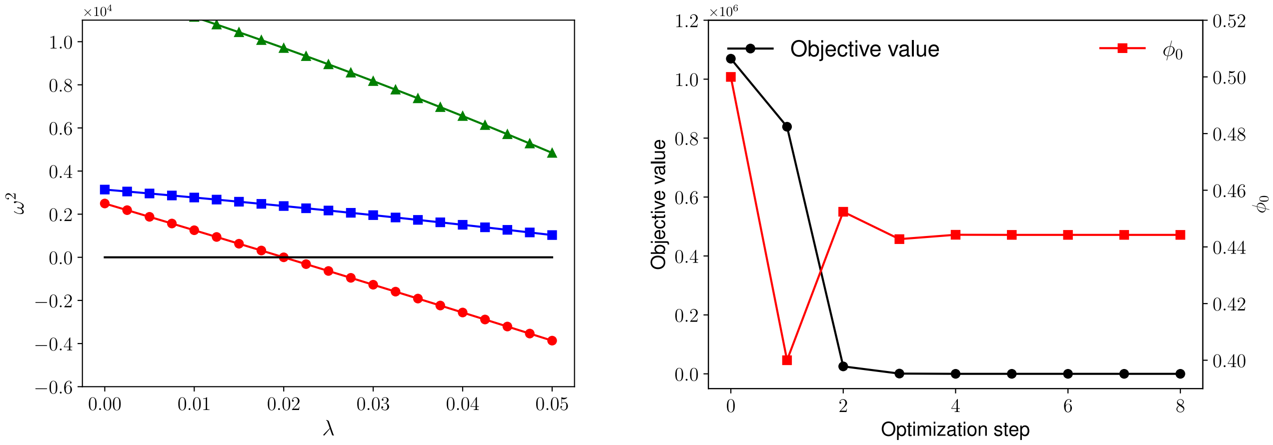
$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^K} \left(\omega^2_{\text{dom}}\right)^2, \tag{3.38}$$

where ω^2_{dom} is the dominant eigenvalue at a prescribed load parameter λ_{cr}. The objective function is basically a penalty term that drives ω^2_{dom} to vanish at λ_{cr}, interpreted in the sense that the RVE buckles at λ_{cr}. To avoid unnecessary difficulties, we fix $\xi_1 = 0$ and $\xi_2 = 0$ (pore shape is a perfect circle), only to optimize over porosity ϕ_0, which suffices to produce satisfactory optimization results. The load parameter is set to be $\lambda_{\text{cr}} = 0.02$. After optimization, we load the RVE again with the optimal porosity ϕ_0. As shown in Fig. 3.9 (a), the dominant eigenvalue (represented by the red curve) intersects the zero axis exactly at $\lambda = 0.02$, indicating that the RVE buckles at this critical load. The objective values and the porosity values during optimization iterations are plotted in Fig. 3.9 (b), showing a relatively quick and steady convergence.

As a remark, when computing the gradient of the objective function in Eq. (3.38), we have adopted a key formula [144] for evaluating the derivative of an eigenvalue

$$\frac{\mathrm{d}\omega^2}{\mathrm{d}\boldsymbol{S}} = \frac{\boldsymbol{V}^*\big(\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}\boldsymbol{S}} - \omega^2\frac{\mathrm{d}\boldsymbol{M}}{\mathrm{d}\boldsymbol{S}}\big)\boldsymbol{V}}{\boldsymbol{V}^*\boldsymbol{M}\boldsymbol{V}}, \tag{3.39}$$

where $(\omega^2, \boldsymbol{V})$ is a pair of eigenvalue and eigenvector.



(a) (b)

Figure 3.9: Optimization results of the circular pore shape RVE for controlling critical buckling load. (a) The evolution of eigenvalues ω^2 at different levels of compression (increasing the load parameter λ) for the RVE with optimal porosity. (b) The optimization iterations showing the objective values and the porosity values.

In the next numerical example, we choose another representative pore shape so that $\xi_1 = -0.2$ and $\xi_2 = 0.1$ (see Fig. 3.1) are fixed. The change of pore shape leads to a distinct buckling behavior. Similarly, we load the RVE with increasing λ from 0 to 0.05 and the results are shown in Fig. 3.10. If we compare the eigenmodes in Fig 3.8 with those in Fig 3.10, we can see that the third eigenmode stays the same, but the first eigenmode(s) and the second eigenmode(s) are exchanged. For the newly selected pore shape, the dominant eigenmodes are under a repeated eigenvalue. By inspecting the intersection point of the dominant eigenvalue (the red curve) with the zero axis, we find that the RVE loses stability at $\lambda \approx 0.0175$.

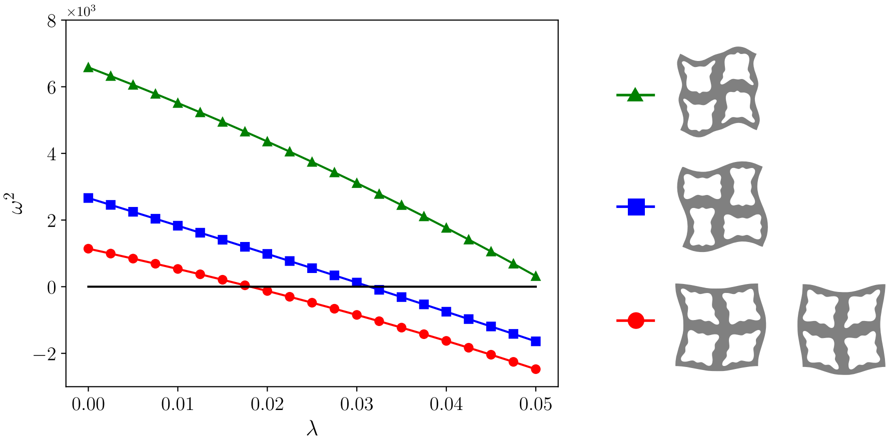


Figure 3.10: Numerical study of the instability of a RVE with a newly selected pore shape. The left plot shows the evolution of eigenvalues ω^2 at different levels of compression (increasing the load parameter λ) for the RVE with initial porosity. The right legends show the deformation of RVEs with corresponding eigenmodes.

The optimization problem is posed using the same objective function as in Eq. (3.38). However, the dominant eigenvalue and eigenvector should be changed accordingly. We set the desired critical load parameter $\lambda_{\mathrm{cr}} = 0.03$ and perform adjoint optimization. We then load the RVE again with the optimized porosity ϕ_0. As shown in Fig. 3.11 (a), the dominant eigenvalue (represented by the red curve) intersects the zero axis exactly at $\lambda = 0.03$, validating the optimization results. The objective values and the porosity values during optimization iterations are plotted in Fig. 3.11 (b), showing a quick convergence.

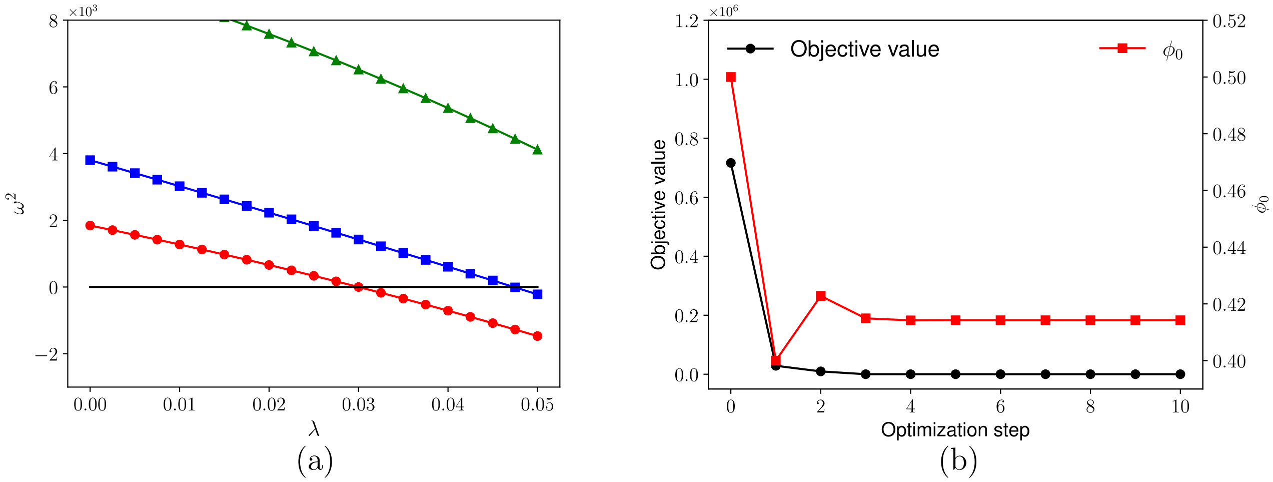


Figure 3.11: Optimization results of the RVE with newly selected pore shape for controlling critical buckling load. (a) The evolution of eigenvalues ω^2 at different levels of compression (increasing the load parameter λ) for the RVE with optimal porosity. (b) The optimization iterations showing the objective values and the porosity values.

3.4.3 Maximizing band gaps

Periodic structures have the potential to manipulate the propagation of waves through band gaps, frequency ranges in which the propagation of elastic waves is significantly attenuated [145, 146, 147]. Band gaps property of periodic structures have been used with wide applications in designing frequency modulators [148], noise controllers [149], and vibration isolators [150]. An example of such periodic structures is given by CMMs, the subject of this study, whose pores are periodically arranged in an elastomer. Previous studies show that the propagation of elastic waves in CMMs are affected by geometric features (e.g., porosity [151]), material nonlinearities [152], and external loadings that cause the structure to undergo large deformations [153]. In discoveries of novel structures with optimal wave-guiding properties though, these approaches are often heuristic and the design process follows the pattern of trial-and-error loops. We use MSOM for gradient-based rational designs of CMMs that achieve maximum band gaps by optimizing pore shapes.

We focus on CMMs with infinite extent characterized by a RVE, which is specified by the lattice vector $\mathbf{a}_1 = (2L_0, 0)$ and $\mathbf{a}_2 = (0, 2L_0)$, as shown in Fig. 3.12 (a). The reciprocal lattice in Fig. 3.12 (b) is identified by the reciprocal lattice vectors

$$\boldsymbol{b}_1 = 2\pi \frac{\mathbf{a}_2 \times \boldsymbol{z}}{\|\boldsymbol{z}\|^2}, \quad \boldsymbol{b}_2 = 2\pi \frac{\boldsymbol{z} \times \mathbf{a}_1}{\|\boldsymbol{z}\|^2}, \tag{3.40}$$

where $\boldsymbol{z} = \mathbf{a}_1 \times \mathbf{a}_2$. It follows that $\mathbf{a}_i \cdot \boldsymbol{b}_j = 2\pi\delta_{ij}$, where δ_{ij} is the Kronecker delta.

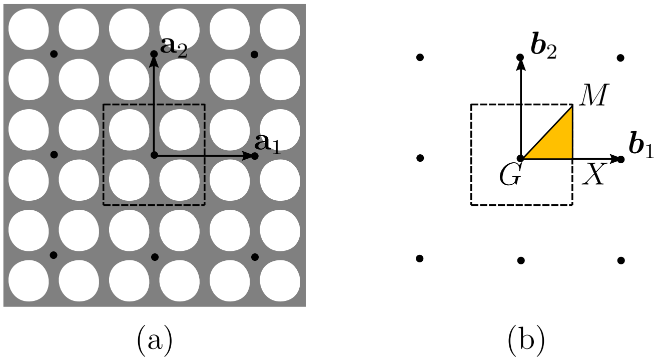


Figure 3.12: (a) Point lattice. (b) Reciprocal lattice and irreducible Brillioun zone (yellow triangle GXM).

Bloch wave analysis applied to periodic structures gives the form of small-amplitude elastic waves superimposed on a finite state of deformation [154, 155, 156]

$$\begin{aligned} \delta\boldsymbol{u} &= \tilde{\boldsymbol{u}} \exp(\mathrm{i}\boldsymbol{k} \cdot \boldsymbol{X})\exp(-\mathrm{i}\omega t), \\ \tilde{\boldsymbol{u}} &= \tilde{\boldsymbol{u}}_{\mathrm{re}} + \tilde{\boldsymbol{u}}_{\mathrm{im}}\mathrm{i}, \end{aligned} \tag{3.41}$$

where $\tilde{\boldsymbol{u}}$ is a periodic function of the unit cell, and $\exp(\mathrm{i}\boldsymbol{k} \cdot \boldsymbol{X})$ accounts for the phase change across each unit cell with $\boldsymbol{k}$ being the Bloch-wave vector. Following the linear perturbation analysis by Eq. (3.32), we derive the corresponding weak form by the

Hermitian inner product with a test vector $\boldsymbol{v}$ written as [157]

$$
\begin{aligned}
\boldsymbol{v} &= \tilde{\boldsymbol{v}} \exp(-\mathrm{i}\boldsymbol{k} \cdot \boldsymbol{X}), \\
\tilde{\boldsymbol{v}} &= \tilde{\boldsymbol{v}}_{\mathrm{re}} - \tilde{\boldsymbol{v}}_{\mathrm{im}}\mathrm{i}.
\end{aligned}
\tag{3.42}
$$

Omitting the imaginary part, the weak form states that find $(\tilde{\boldsymbol{u}}_{\mathrm{rm}}, \tilde{\boldsymbol{u}}_{\mathrm{im}})$ such that $\forall(\tilde{\boldsymbol{v}}_{\mathrm{rm}}, \tilde{\boldsymbol{v}}_{\mathrm{im}})$

$$
\begin{aligned}
&\int_{\mathbb{B}_{\boldsymbol{X}}} \Big(\nabla_{\boldsymbol{X}}\tilde{\boldsymbol{v}}_{\mathrm{re}} : \mathbb{L} : \nabla_{\boldsymbol{X}}\tilde{\boldsymbol{u}}_{\mathrm{re}} + \nabla_{\boldsymbol{X}}\tilde{\boldsymbol{v}}_{\mathrm{im}} : \mathbb{L} : \nabla_{\boldsymbol{X}}\tilde{\boldsymbol{u}}_{\mathrm{im}} + \tilde{\boldsymbol{v}}_{\mathrm{re}} \otimes \boldsymbol{k} : \mathbb{L} : \tilde{\boldsymbol{u}}_{\mathrm{re}} \otimes \boldsymbol{k} + \tilde{\boldsymbol{v}}_{\mathrm{im}} \otimes \boldsymbol{k} : \mathbb{L} : \tilde{\boldsymbol{u}}_{\mathrm{im}} \otimes \boldsymbol{k} + \\
&\quad \nabla_{\boldsymbol{X}}\tilde{\boldsymbol{v}}_{\mathrm{im}} : \mathbb{L} : \tilde{\boldsymbol{u}}_{\mathrm{re}} \otimes \boldsymbol{k} - \nabla_{\boldsymbol{X}}\tilde{\boldsymbol{v}}_{\mathrm{re}} : \mathbb{L} : \tilde{\boldsymbol{u}}_{\mathrm{im}} \otimes \boldsymbol{k} - \tilde{\boldsymbol{v}}_{\mathrm{im}} \otimes \boldsymbol{k} : \mathbb{L} : \nabla_{\boldsymbol{X}}\tilde{\boldsymbol{u}}_{\mathrm{re}} + \tilde{\boldsymbol{v}}_{\mathrm{re}} \otimes \boldsymbol{k} : \mathbb{L} : \nabla_{\boldsymbol{X}}\tilde{\boldsymbol{u}}_{\mathrm{im}}\Big) \mathrm{d}\boldsymbol{X} \\
&\qquad = \rho_0 \omega^2 \int_{\mathbb{B}_{\boldsymbol{X}}} \Big(\tilde{\boldsymbol{v}}_{\mathrm{re}} \cdot \tilde{\boldsymbol{u}}_{\mathrm{re}} + \tilde{\boldsymbol{v}}_{\mathrm{im}} \cdot \tilde{\boldsymbol{u}}_{\mathrm{im}}\Big)\, \mathrm{d}\boldsymbol{X}.
\end{aligned}
\tag{3.43}
$$

The weak form of Eq. (3.43) is then transformed to the referential configuration so that the dependency of the shape displacement $\boldsymbol{s}$ is explicit. The final discrete system is an eigenvalue problem as the following

$$
\begin{aligned}
\boldsymbol{N}(\boldsymbol{S})\boldsymbol{W} &= \omega^2 \boldsymbol{M}(\boldsymbol{S})\boldsymbol{W}, \\
\boldsymbol{N} &: \mathbb{R}^N \to \mathbb{R}^{2L} \times \mathbb{R}^{2L}, \\
\boldsymbol{M} &: \mathbb{R}^N \to \mathbb{R}^{2L} \times \mathbb{R}^{2L},
\end{aligned}
\tag{3.44}
$$

where $\boldsymbol{V} \in \mathbb{R}^{2L}$ is the eigenvector paired with the eigenvalue ω^2. Eq. (3.43) is for a fixed $\boldsymbol{k}$, and the angular frequency ω is a function of $\boldsymbol{k}$. Due to symmetries, we only need to consider $\omega(\boldsymbol{k})$ in the irreducible Brillouin zone [158] as shown by the triangle GXM in Fig. 3.12 (b).

In this numerical example, the forward problem involves three major steps. The first two steps exactly follow *Step 1* and *Step 2* in Section 3.3.2, namely, we first solve for the shape displacement $\boldsymbol{s}$ to obtain the pore shape, and then we solve for a base state displacement field $\boldsymbol{u}$ at equilibrium under a specified external loading condition. Specifically for the loading condition, we assume the macroscopic displacement gradient $\overline{\boldsymbol{H}} = \left[\begin{smallmatrix} -\lambda & 0 \\ 0 & -\lambda \end{smallmatrix}\right]$, where $\boldsymbol{\lambda} \geq 0$ is given. The third step as discussed in this subsection is a linear perturbation operation and we solve an eigenvalue problem for the angular frequency ω under a given $\boldsymbol{k}$.

As a first case to study, we set $\boldsymbol{\alpha} = (0, 0, 0.5)$ (circular pore shape with porosity 0.5), and the external loading condition to be free ($\lambda = 0$). We run the forward model for a set of Block-wave vectors in the irreducible Brillouin zone. The normalized frequencies $\overline{\omega} = \frac{\omega L_0}{\pi c_T}$ ($c_T = (\mu/\rho_0)^{\frac{1}{2}}$ is the shear wave speed) are plotted against the reduced wave vectors in the band diagram, shown in Fig. 3.13 (a). As noticed, there appears a band gap at the normalized frequency from 1.39 to 1.58. The band gap width is around 0.19.

The optimization goal is to maximize the band gap. By observation, the band gap is between the 12th and 13th frequencies. Denote the i^{th} normalized frequency by $\overline{w}_i(\boldsymbol{k})$. We state the optimization problem as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^K} \Big(\max(\Omega_{\text{low}}) - \min(\Omega_{\text{high}}) \Big) \tag{3.45}$$

where

$$\Omega_{\text{low}} = \Big\{ \overline{\omega}_i(\boldsymbol{k}) \;\Big|\; i \in \{1, 2, ..., 12\}, \boldsymbol{k} \in [G \to X \to M \to G] \Big\}, \tag{3.46a}$$

$$\Omega_{\text{high}} = \Big\{ \overline{\omega}_i(\boldsymbol{k}) \;\Big|\; i \in \{13, 14, ..., 16\}, \boldsymbol{k} \in [G \to X \to M \to G] \Big\}. \tag{3.46b}$$

In the objective function of Eq. (3.45), the "max" and "min" operators are not differentiable. We apply the LogExpSum (LSE) trick to get a smoothed maximum operator, namely,

$$\max\left(\{x_i\}_{i=1}^{n}\right) \approx \frac{1}{\beta}\log\Big(\sum_{i=1}^{n}\exp(\beta x_i)\Big), \tag{3.47}$$

where β is a large constant that makes the approximation tight. The evaluation of LSE requires numerical hacks to avoid large number overflow due to the exponentials in the function [159]. When computing the total gradient of the objective function, Eq. (3.39) is adopted for computing derivatives of eigenvalues.

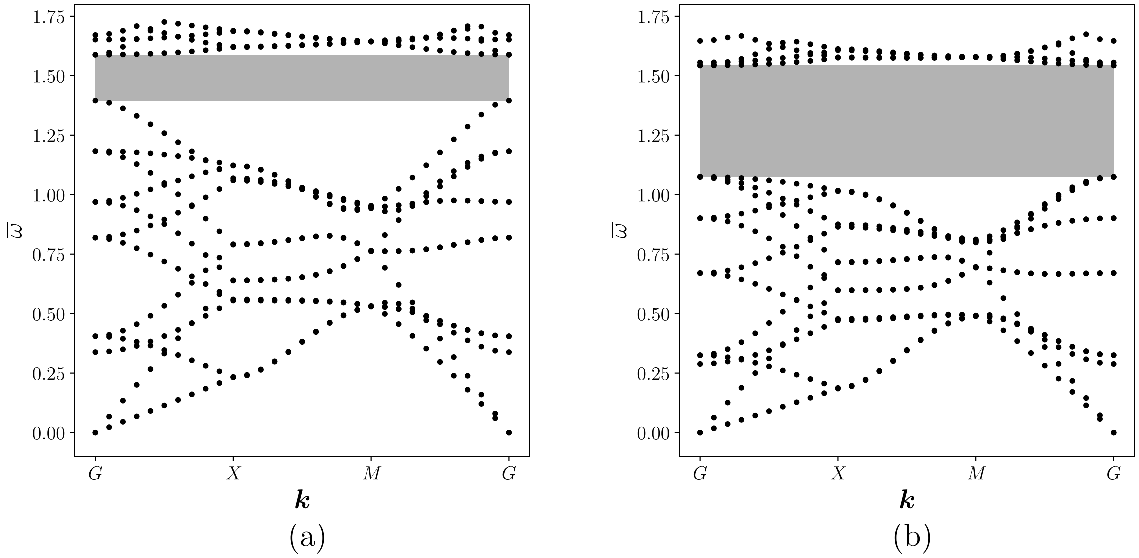


Figure 3.13: Band diagrams for RVEs with no deformation ($\lambda = 0$) (a) before optimization and (b) after optimization. The normalized frequency $\overline{w}$ is plotted against the reduced wave vector $\boldsymbol{k}$. The gray area shows the band gap that forbids wave propagation.

The optimization iterations are shown in Fig. 3.14, where the initial RVE corresponds to the band diagram in Fig. 3.13 (a) and the optimal RVE corresponds to the band diagram in Fig. 3.13 (b). The optimized band gap forbids frequencies from 1.07 to 1.54, forming a band gap of width around 0.47. The width of the band gap is increased by 143%, showing a successful optimization result.

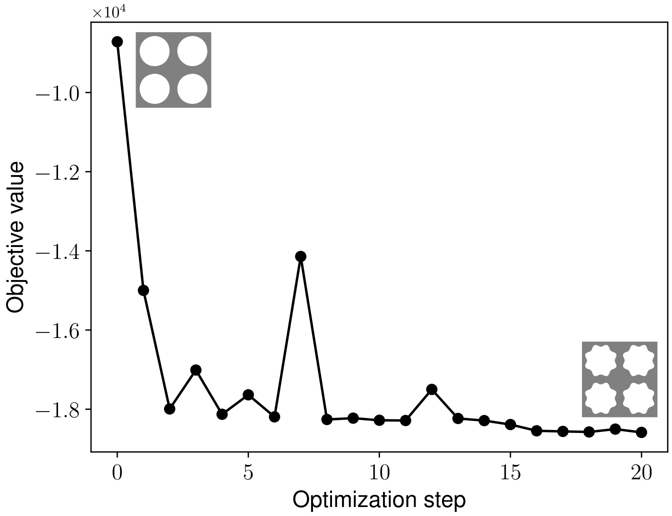


Figure 3.14: The optimization iterations for the case with no RVE deformation ($\lambda = 0$). The left RVE corresponds to the initial pore shape and the right RVE corresponds the optimized pore shape that maximizes the band gap.

It is pointed out by [153] that wave propagation in CMMs is affected when the structure undergoes large deformation. For completeness, we perform a second numerical experiment including deformations of RVE in the sense that we set the loading condition to be $\lambda = 0.1$. Before optimization, the band diagram is shown in Fig. 3.15 (a), where we observe a band gap of width around 0.35. After optimization, the band gap (see Fig. 3.15 (b)) is increased to 0.51, an improvement of around 45%. The corresponding RVEs before and after optimization are found in Fig. 3.16, where optimization iterations are shown. The optimization is still effective, but the improvement is not as significant as when the RVE is undeformed.

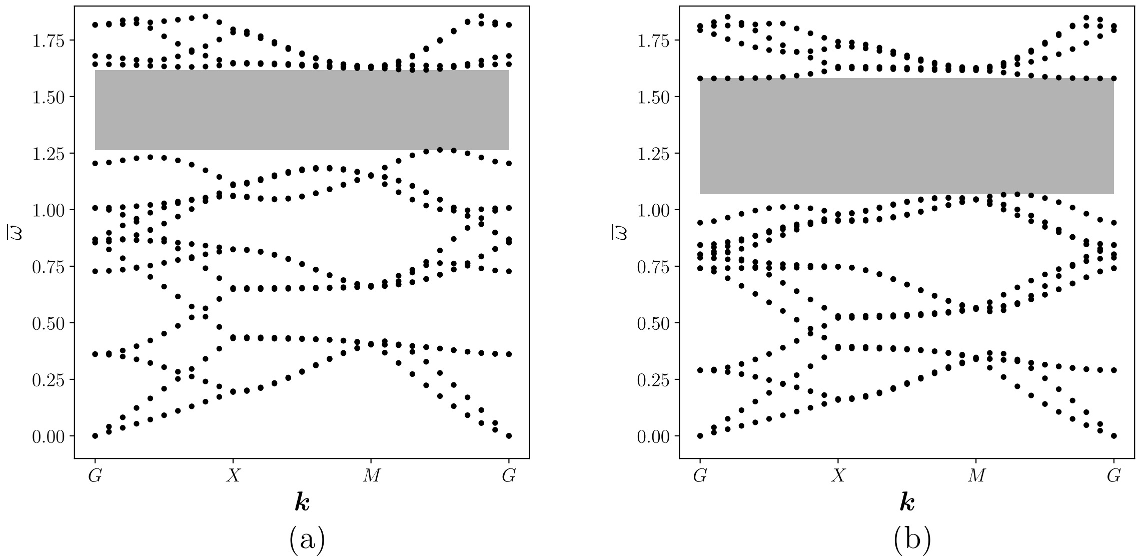


Figure 3.15: Band diagrams for RVEs with deformation ($\lambda = 0.1$) (a) before optimization and (b) after optimization. The normalized frequency $\overline{w}$ is plotted against the reduced wave vector $\boldsymbol{k}$. The gray area shows the band gap that forbids wave propagation.

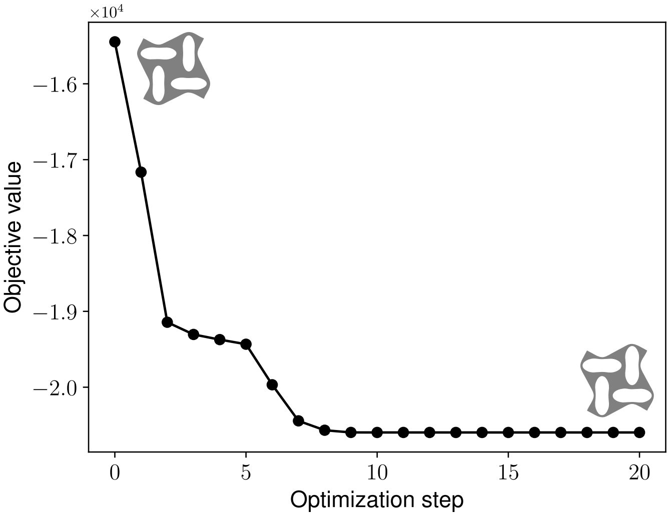


Figure 3.16: The optimization iterations for the case with RVE deformations ($\lambda = 0.1$). The left RVE corresponds to the initial pore shape and the right RVE corresponds the optimized pore shape that maximizes the band gap.

3.5 Conclusions

In this work, we proposed a mapped shape optimization method (MSOM) for the rational design of soft cellular metamaterials under large deformation. The geometry of the metamaterials are parametrized by several key parameters that are used for optimization. Finite element computation is performed over a referential configuration Then, an adjoint-based optimization scheme is developed based on the FEM set-up. Benchmark problems are used to demonstrate the convergence properties of the proposed MSOM, which shows apparent advantages. We used our MSOM to solve several important problems in the design of CMM's, including optimization of the negative Poisson's ratio, precise control over the instability point of the structures, and arbitrary tunability of the band structures of phononic CMM's. In all of these problems, MSOM can not only solve the problem, but solve them efficiently. In conclusion, the proposed MSOM can be a useful tool for advanced optimization purposes in CMM's.

Chapter 4

Machine Learning Generative Models for Designing Composite Mechanical Metamaterials

This chapter studies the inverse design problems of composite mechanical metamaterials. We apply machine learning enabled approaches for optimal design of metamaterials with arbitrary overall elastic moduli. Unlike previous chapters that consider finite strain deformations, we only stick to the assumption of small displacement and linear elasticity in this chapter. The designed samples are fabricated with additive manufacturing techniques, and the mechanical properties are experimentally validated.

4.1 Introduction

Architected with multi-level structures, mechanical metamaterials exhibit unique macroscopic properties such as being ultralight and ultrastiff [1], pentamodal [160], auxetic [2], etc. The rapid growth of this field has been partly fueled by the advancement of manufacturing technologies such as additive manufacturing [13, 14], which enables fast prototyping and economic small-scale production of materials with specific microstructures [15]. Among them, the emerging technique of multi-material 3D printing [161] reveals a promising direction of designing novel composite solids by combining multiple base materials with distinct properties [162, 163].

We focus on elastic moduli control, an important subject in designing mechanical metamaterials [164]. The goal is to design a representative volume element (RVE) composed of different base materials so that when the RVEs are periodically arranged into a lattice structure, the macroscopic elastic moduli achieve a desired set of values. Central to the design problems is the appropriate *parametric representation* of RVE structures. Standard topology optimization methods suffer from being computationally expensive due to voxel level parametrization [67]. To avoid direct parametrization of voxels, [68] pre-defines a set of manually constructed microstructure families and base the design on these templates. Still, traditional approaches like this need to go through trial-and-error design loops which involve domain expertise and intensive labor. Establishing an optimization framework where design parameters are selected automatically without human interference remains challenging. Those parameters must live in a relatively low dimensional space feasible for optimization while not losing their representation power. Our proposed optimization framework addresses the challenge using machine learning techniques.

Over the past few years, groundbreaking advances in machine learning provide new insight into the development of robust design strategies, particularly in mechanics, materials and structures [89, 165, 87]. New paradigms have emerged where complicated data structures are compressed into an effective lower dimensional space and designs are instead performed over such reduced space. This usually requires coupling of a generative model that maps from the reduced space to the original data space and a gradient-free optimization algorithm that operates over the reduced space. Variational autoencoders (VAEs) [75] and generative adversarial networks (GANs) [166] are the two most popular generative models. Bayesian optimization (BayesOpt) [76], a sequential design strategy to seek global optimum, is usually adopted to couple with the generative models for optimization. Representative works using this coherent design strategy include but not limited to discovery of new molecules with desired properties [41], shape optimization for minimal drag force in fluid dynamics [72], finding optimal optical performance of composite materials [73], improvement in solution efficiency for heat conduction [74], etc.

In this work we propose an optimization framework that automates the procedures of designing composite mechanical metamaterials. We adopt VAEs to compress RVE images that describe the spatial distribution of base materials to a reduced latent space. As generative models, VAEs also provide a venue to recover RVE images from the latent space. We then use BayesOpt to find an optimal RVE configuration that fulfills the prescribed design goal. Within each iteration of BayesOpt, we need to compute the macroscopic elastic moduli. We take a systematic approach known as homogenization, a well established technique [93, 94, 95, 97], for the evaluation of the effective macroscopic elastic moduli. After the optimization loop, we use multimaterial 3D printing to fabricate samples and experimentally measure their elastic moduli for validation. Quantitative comparisons show that our scheme is applicable and reliable.

4.2 Computational Homogenization for Linear Elasticity

Before proceeding to the design problem, we introduce the computational homogenization scheme, a key step in each iteration of BayesOpt. We consider a linear elasticity problem over a domain $\Omega \subset \mathbb{R}^2$. The problem (in the absence of body force) reads: Find the displacement field $\boldsymbol{u} : \Omega \rightarrow \mathbb{R}^2$ such that

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} &\qquad \text{in } \Omega, \\ \boldsymbol{u} = \boldsymbol{u}_b &\qquad \text{on } \partial\Omega^D, \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t} &\qquad \text{on } \partial\Omega^N, \end{aligned} \tag{4.1}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{u}_b$ is a known function defined on $\partial\Omega^D$, $\boldsymbol{n}$ is the unit outward normal, $\boldsymbol{t}$ is the traction applied on the boundary $\partial\Omega^N$, $\partial\Omega^N \cup \partial\Omega^D = \partial\Omega$ and $\partial\Omega^N \cap \partial\Omega^D = \emptyset$. The constitutive model can be defined by a strain energy density function

$$W = \frac{1}{2}\boldsymbol{\epsilon} : \mathbb{C} : \boldsymbol{\epsilon}, \tag{4.2}$$

where $\boldsymbol{\epsilon} = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^\top)$ is the strain tensor and $\mathbb{C}$ is the fourth order constitutive tensor.

The above formulation is based on a single, homogeneous material. Our composite solids are composed of multiple materials. The overall mechanical properties may significantly differ from each of the base material. Computational homogenization is a systematic approach to obtain “averaged” macroscopic mechanical properties. At the macroscopic level, we formulate the homogenized constitutive relationship as

$$\overline{W} = \frac{1}{2}\overline{\boldsymbol{\epsilon}} : \overline{\mathbb{C}} : \overline{\boldsymbol{\epsilon}}, \tag{4.3}$$

where $\overline{\square}$ denotes a macroscopic quantity. Given $\mathbb{C}$ for each base material and RVE configuration of how they form the composite solid, our goal is to compute the macroscopic constitutive tensor $\overline{\mathbb{C}}$. We follow the treatment by [66], where the principle for homogenization is that the total strain energy stored in the RVE computed via Eq. 4.2

and Eq. 4.3 must be equal for arbitrarily imposed macroscopic strain conditions. For further details about computational homogenization of linear elasticity, see 4.4.

In this work, the composite solids are made from two base materials: a commercial hard polyurethane (RPU) with measured Young's modulus $E_h = 1300\text{MPa}$ and Poisson's ratio $\nu_h = 0.23$ and a custom soft silicone (SilDN) [167] with $E_s = 0.12\text{MPa}$ and $\nu_s = 0.33$. We make plane stress assumption throughout the work. We further restrict the composite solids to preserving cubic symmetry as shown in Fig. 4.1. It can be shown that under these assumptions the constitutive tensor $\overline{\mathbb{C}}$ are fully described by three independent elastic moduli [168, 169]. For example, it is common to pick Young's modulus $\overline{E}$, shear modulus $\overline{G}$ and Poisson's ratio $\overline{\nu}$ as a complete set of description, in which case the macroscopic stress-strain relationship simplifies to

$$\begin{bmatrix} \overline{\boldsymbol{\sigma}}_{11} \\ \\ \overline{\boldsymbol{\sigma}}_{22} \\ \\ \overline{\boldsymbol{\sigma}}_{12} \end{bmatrix} = \begin{bmatrix} \frac{\overline{E}}{1-\overline{\nu}^2} & \frac{\overline{\nu}\overline{E}}{1-\overline{\nu}^2} & 0 \\ \\ \frac{\overline{\nu}\overline{E}}{1-\overline{\nu}^2} & \frac{\overline{E}}{1-\overline{\nu}^2} & 0 \\ \\ 0 & 0 & \overline{G} \end{bmatrix} \begin{bmatrix} \overline{\boldsymbol{\epsilon}}_{11} \\ \\ \overline{\boldsymbol{\epsilon}}_{22} \\ \\ 2\overline{\boldsymbol{\epsilon}}_{12} \end{bmatrix} . \tag{4.4}$$

As shown in Fig. 4.1, we summarize the computational homogenization problem as to find out the macroscopic elastic moduli $\{\overline{E}, \overline{\nu}, \overline{G}\}$ given an image-like description (binary pixel values indicating base material information) of a RVE. The computational homogenization technique requires the use of the Finite Element Method (FEM). We choose the open source FEM software `FEniCS` [170] for the implementation.

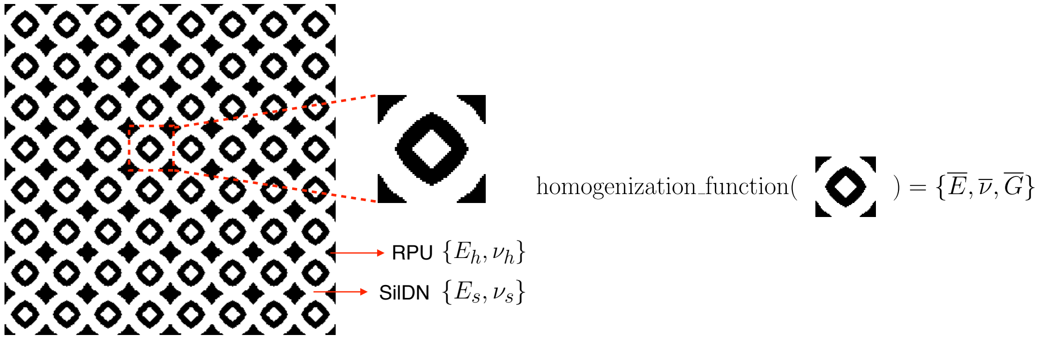

homogenization_function( ) = $\{\overline{E}, \overline{\nu}, \overline{G}\}$
RPU $\{E_h, \nu_h\}$
SilDN $\{E_s, \nu_s\}$

Figure 4.1: Computational homogenization setup. Leftmost is a composite solid composed of two base materials RPU and SilDN, with elastic moduli $\{E_h, \nu_h\}$ and $\{E_s, \nu_s\}$ respectively. A magnified RVE is also shown for clarity. Rightmost is the homogenization function, treated as a black-box function that takes the input of an image representation of a RVE and yields the output of the set of homogenized macroscopic elastic moduli $\{\overline{E}, \overline{\nu}, \overline{G}\}$.

4.3 Composite Mechanical Metamaterials for Elasticity Control

One can view computational homogenization as a "forward problem" in that a description of RVE is given and the aim is to find the macroscopic elastic moduli $\{\overline{E}, \overline{\nu}, \overline{G}\}$. On the contrary, the design problem seeks an optimal RVE in order to achieve a prescribed set of macroscopic elastic moduli, so it is helpful to think about the design problem as an "inverse problem". In our problem, the design space is the set of binary images, immediately leading to an intractable, high dimensional combinatorial optimization problem. We tackle the challenge by learning a reduced representation of the image data, and performing optimization over this low dimensional space instead. Once we obtain the optimal design, we use 3D printing to fabricate the corresponding composite solids and measure the macroscopic mechanical properties experimentally for validation.

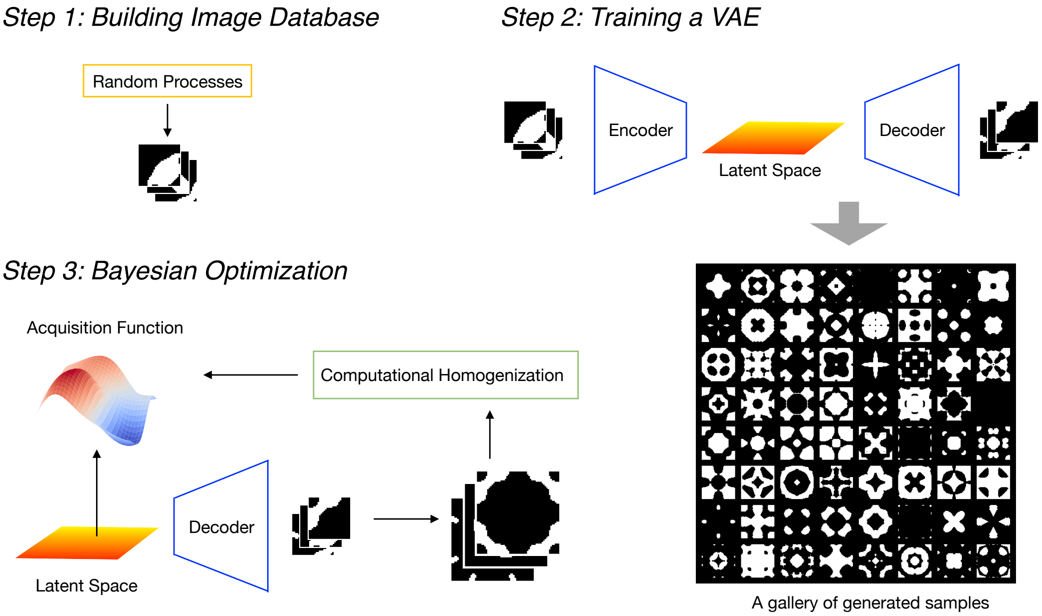

Step 1: Building Image Database
Random Processes
Step 2: Training a VAE
Encoder
Latent Space
Decoder
Step 3: Bayesian Optimization
Acquisition Function
Computational Homogenization
Decoder
Latent Space
A gallery of generated samples

Figure 4.2: The proposed optimization framework. Step 1: Draw samples from a random process to build the artificial database of RVE images (28 × 28 pixels for each image). Step 2: Train a VAE so that it generates realistic output samples. By flipping the 28 × 28 images twice we obtain the 56 × 56 RVE images that preserve symmetry. Step 3: BayesOpt towards optimal design of a RVE that achieves the prescribed macroscopic elastic moduli.

4.3.1 Bayesian optimization over reduced space

The overall design flow is presented in Fig. 4.2. We prepare an artificial image database and train the VAE to generate realistic images. BayesOpt operates over the reduced latent space learned by the VAE and couples with the homogenization procedure introduced in Section 4.2.

VAEs are among the family of latent-variable models [127]. The observable RVE image data can be considered as samples drawn from a probability distribution $p(x)$ where $x \in \mathcal{X}$ lives in a high dimensional space. Under the assumption, the high dimensional image data can be compressed into a low dimensional space without severe loss of information. We denote the associated low dimensional distribution as $p(z)$ where $z \in \mathcal{Z}$ is the latent variable. VAEs provide a principled approach for learning latent-variable models by jointly training over two parameterized neural network models: the encoder or recognition model, and the decoder or generative model. VAEs generalize autoencoders and encourage the latent space to be continuous in producing valid decoded images, which qualifies as the design space for optimization.

We implement the VAE model in `PyTorch` [171], an open source deep learning platform. The input to the VAE are images of size $(batch, 1, 28, 28)$ while the output is of the same size. The input image database is constructed following a stochastic approach adopted in [172, 173]. The method thresholds a Gaussian random field (GRF) to generate realistic binary representations of RVEs. The width for the hidden layer is set to be 10, namely, $\mathscr{Z} = \mathbb{R}^{10}$. The encoder part consists of two convolutional and two dense layers. Similarly, the decoder consists of two dense layers followed by two deconvolutional layers. Right before the output layer we add an additional constraint to force diagonal symmetry on the output tensors. To generate new samples, we only need the decoder model. We draw random variables from a unit Gaussian distribution as the input to the decoder and threshold the decoded images to reconstruct the binary composition. Fig. 4.2 shows a gallery of generated RVE samples. A more detailed description on VAEs can be found in 4.4.

To further demonstrate the feasibility of optimizing over the latent space $\mathscr{Z}$, we draw 200 samples from $p(z)$, obtain decoded RVE images, and perform computational homogenization. For visualization purpose, we use principal component analysis (PCA) [174] to compress the latent space into a 2-dimensional space and show the results in Fig. 4.3. As shown, the latent space demonstrates a continuous variation for the macroscopic Young's modulus $\overline{E}$ and the macroscopic shear modulus $\overline{G}$, while a less clear trend is observed for macroscopic Poisson's ratio $\overline{\nu}$. This is not surprising, though, as adjusting $\overline{E}$ or $\overline{G}$ is straightforward by allocating RPU (hard material) and SilDN (soft material) in different proportions to the RVE, yet to get a desired $\overline{\nu}$ is not immediately obvious from the perspective of a human expert. It is therefore expected that the optimization towards a tailored $\overline{\nu}$ be more difficult than $\overline{E}$ or $\overline{G}$.

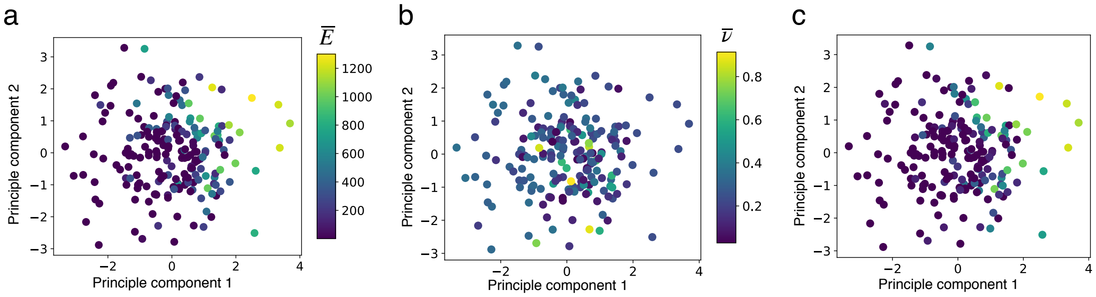


Figure 4.3: PCA plots for latent space carrying information of the macroscopic elastic moduli.

We next show that coupled with BayesOpt, we are able to design a RVE that simultaneously achieves desired $\overline{E}$ and $\overline{\nu}$. Formally, we state our design problem as an optimization problem:

$$z^{\star} = \arg\max_{z \in \mathcal{Z}} f(z), \tag{4.5}$$

where $f(z) =$ `cost_function(homogenization_function(decoder_function(`z`)))` is treated as a black-box function. The `decoder_function` is the decoder part of the VAE model. The `homogenization_function` is discussed in Section 4.2. We define `cost_function` as

$$\texttt{cost\_function}(\{\overline{E}, \overline{\nu}, \overline{G}\}) = \mathrm{e}^{-(\widehat{E}-\widehat{E}_t)^2-(\widehat{\nu}-\widehat{\nu}_t)^2}, \tag{4.6}$$

where

$$\widehat{E} = \frac{\log(\overline{E}) - \log(E_s)}{\log(E_h) - \log(E_s)}, \quad \widehat{\nu} = \frac{\overline{\nu} - \nu_h}{\nu_s - \nu_h}, \quad \widehat{E}_t = \frac{\log(\overline{E}_t) - \log(E_s)}{\log(E_h) - \log(E_s)}, \quad \widehat{\nu}_t = \frac{\overline{\nu}_t - \nu_h}{\nu_s - \nu_h}, \tag{4.7}$$

with $\overline{E}_t = 260\text{MPa}$ and $\overline{\nu}_t = 0.4$ being set as the design goal. These numbers are arbitrarily selected and only serve for the purpose of a concrete demonstration of the proposed methodology. The `cost_function` considers normalized quantities (denoted by $\widehat{\square}$) to alleviate the effect of various length scales of different quantities.

The computational cost to evaluate the objective function in Eq. 4.5 is not cheap, motivating the use of BayesOpt, a class of machine-learning-based optimization methods. BayesOpt builds a probabilistic surrogate model for the objective function and queries the next data point by minimizing an expected loss function. The optimization loop is completely automatic and avoids subjective human decisions in trial-and-error design. In this work, we use noise free Gaussian process regression as the surrogate model for BayesOpt. We apply expected improvement, a classic acquisition function to determine the next data point to sample. The searching process in principle tries to gain a balance between exploitation and exploration. We point to [175, 176] for comprehensive descriptions of BayesOpt.

We adopt an open source library `BoTorch` [177] for BayesOpt. We evaluate the objective function values for 100 randomly generated RVEs for warm-up and obtain the objective value to be around 0.8, considered as a suitable initial point. Then, we start BayesOpt and plot the current best objective values with respect to the iteration number. We terminate the optimization when the current best objective value is close to 1, which is the theoretical limit when the elastic moduli of the optimized RVE perfectly match the targets. As shown in Fig. 4.4, the optimized RVE has the macroscopic elastic moduli:

$$\{\overline{E}^{\star}, \overline{\nu}^{\star}, \cdot\} = \texttt{homogenization\_function}(\texttt{decoder\_function}(z^{\star}), \tag{4.8}$$

which are close to the target values $\{\overline{E}_t, \overline{\nu}_t, \cdot\}$ measured in the normalized sense.

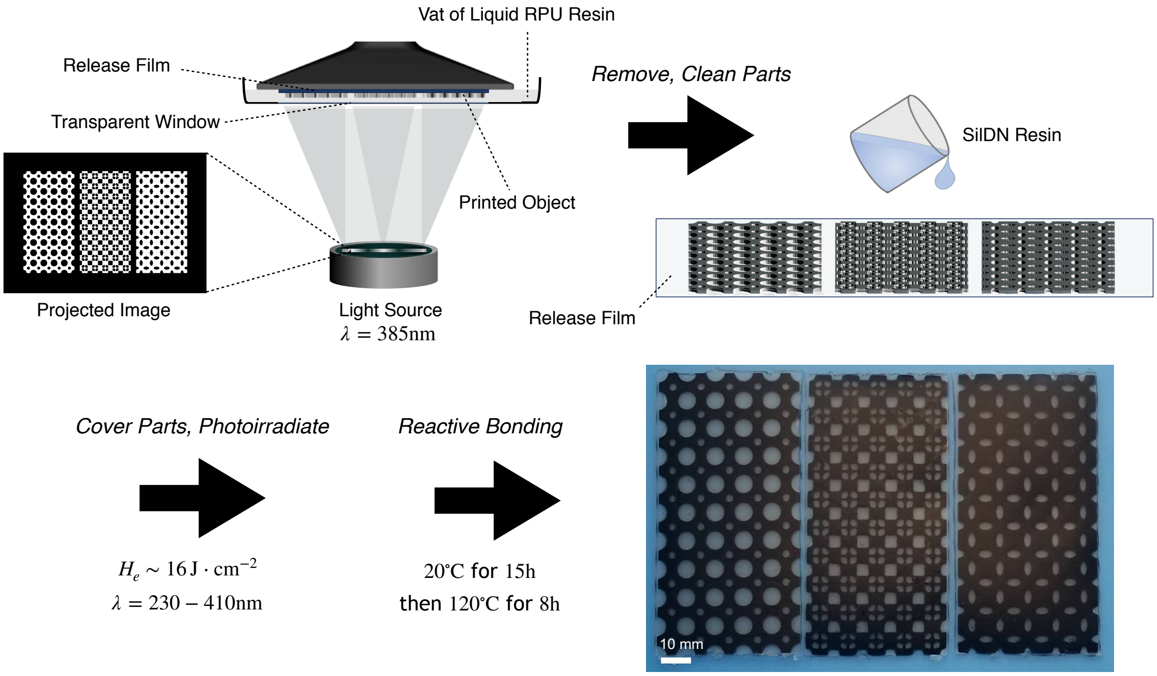


Figure 4.5: Detailed procedures of multi-material 3D printing. We fabricate the three highlighted samples matching the ones presented in Fig. 4.4.

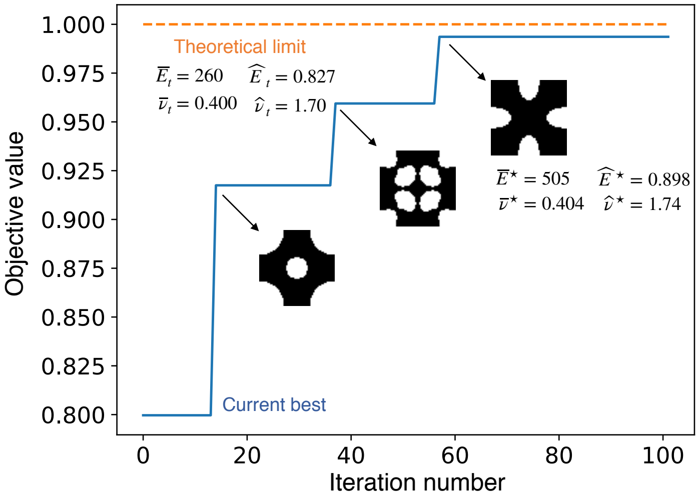


Figure 4.4: Iterations of BayesOpt. The objective function is defined using Eq. 4.5. The theoretical limit is achieved only if the macroscopic elastic moduli exactly match the desired values. Three RVEs are plotted on the path to optimum. The third RVE is the optimal one with $\widehat{E}^\star$ and $\widehat{\nu}^\star$ close to the targets $\widehat{E}_t$ and $\widehat{\nu}_t$.

4.3.2 Experimental validation

It is acknowledged here that Thomas J. Wallin led the physical prototyping of the composite solids. With his help, we fabricate the three highlighted samples in Fig. 4.4 and measure their mechanical properties for experimental validation. Both RPU and SilDN photopolymerize from low viscosity precursors which enables high resolution Stereolithography (SLA, see Fig. 4.5) 3D printing of arbitrary structures like those generated by the VAE. Strong interfacial bonding between material layers remains a significant challenge to 3D printing multimaterial combined structures. Our chosen materials are "dual-stage" polymers; photo-crosslinking during 3D printing sets the object's shape followed by a second, condensation based crosslinking reaction that improves the mechanical robustness. [77] recently employed this latent condensation reaction to cohesively bond 3D printed materials to substrates spanning seven orders of magnitude in Young's Modulus. Though possible to SLA print directly using resin-exchange methods, manufacturing ease and rapid iteration led us to combine SLA printing and UV-assisted replica molding (see Fig. 4.5) when fabricating our multimaterial structures. Briefly, we first print (Carbon$^{\mathrm{TM}}$ M2 printer) the RPU structure onto a release film and then cast the SilDN precursor into the voids. Temporarily clamping a flat, transparent substrate coated with Teflon on top of the printed structure removes excess resin and the subsequent irradiation solidifies the SilDN material. Interfacial crosslinks between RPU and SilDN form during the second stage of curing at room temperature ($t \sim 15$ hours) followed by thermal treatment at 120°C ($t \sim 8$ hours).

With samples fabricated, we experimentally validated our design strategy via tensile tests. As shown in Fig. 4.6, our experimentally measured values for macroscopic elastic moduli generally agree with the computational homogenization predicted trends. The predicted Poisson's ratios fall within the error of the measured values. However, the actual Young's modulus values are consistently higher than predictions. We postulate that the manufacturing process of the composite materials could have consistently favoured RPU over SilDN, leading to a higher macroscopic Young's modulus. Also, the samples (see Fig. 4.5) contain only a finite number RVEs and the measurement could have been affected by the boundary effects, whereas the computational homogenization essentially assumes infinite size of samples. It is important to note that while the composite solid yields an intermediate modulus ($E_s < \overline{E} < E_h$), this strategy can obtain Poisson's ratios beyond the range found between the base materials ($\nu_h < \nu_s < \overline{\nu}$). Thus confirming the ability to obtain novel material properties via mechanical design of metamaterials. For completeness, Table 4.1 gives a summary of the target values, optimization results, and experiment outcomes.

True	$\overline{E}_t$ (Target)	$\overline{E}^\star$ (Optimized)	$\overline{E}_e$ (Exp)	$\overline{\nu}_t$ (Target)	$\overline{\nu}^\star$ (Optimized)	$\overline{\nu}_e$ (Exp)
	260MPa	505MPa	562MPa	0.400	0.404	0.43
Normalized	$\widehat{E}_t$ (Target)	$\widehat{E}^\star$(Optimized)	$\widehat{E}_e$(Exp)	$\widehat{\nu}_t$ (Target)	$\widehat{\nu}^\star$ (Optimized)	$\widehat{\nu}_e$ (Exp)
	0.827	0.898	0.910	1.70	1.74	2.0

Table 4.1: Comparisons between target values, optimization results and experiment outcomes.

In Table 4.1, we notice that the optimization result of $\overline{E}^\star$ (505MPa) is still much larger than the target value $\overline{E}_t$ (260MPa), while the optimization result of $\overline{\nu}^\star$ (0.404) is relatively close to the specified $\overline{\nu}_t$ (0.4). However, the discrepancy is smaller under the normalized metric, with $\widehat{E}^\star$ (0.898) close to $\widehat{E}_t$ (0.827) and $\widehat{\nu}^\star$ (1.74) close to $\widehat{\nu}_t$ (1.70). The actual optimization is implemented using the normalized metric. There is no obvious evidence showing either optimizing towards a tailored Young's modulus or Poisson's ratio is easier, in contrast to our original speculation. For more flexible control of the optimization outcomes, we propose to use a weighted cost function

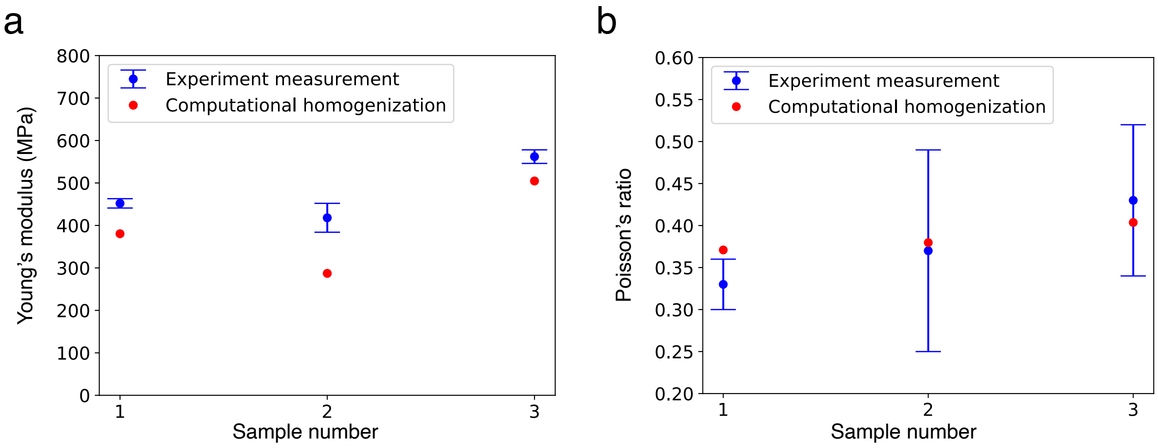


Figure 4.6: Experimental validation. Left: macroscopic Young's modulus by experiment measurement and computational homogenization for the three RVEs in Fig. 4.4. Right: macroscopic Poisson's ratio by experiment measurement and computational homogenization for the same RVEs.

instead of the original Eq. 4.6

$$\texttt{cost\_function}(\{\overline{E}, \overline{\nu}, \overline{G}\}) = \mathrm{e}^{-w_E(\widehat{E}-\widehat{E}_t)^2 - w_\nu(\widehat{\nu}-\widehat{\nu}_t)^2}, \tag{4.9}$$

where w_E and w_ν are two constants reflecting a bias on the optimization targets. We leave the explorations of a more adequately designed cost function as our future work.

Though we have already simplified the optimization problem to a relatively low dimensional parameter space, it is important to be aware that the problem is still highly non-convex. The searching path may finally lead to a local optimum.

4.4 Conclusions

We propose an optimization framework based on machine learning techniques to design composite mechanical metamaterials. We focus on controls of macroscopic elastic moduli and design optimal RVEs to achieve the goal. Results are verified by experimental measurements on multi-material 3D printed samples.

One limitation of this work is that the database of RVE samples are constructed via drawing samples from a artificially defined random process. Since we know precisely about the data generating process, it is possible to use more effective data compression methods than to use a VAE. For example, directly applying PCA to the database used in this work could have yielded a comparable performance to a nonlinear autoencoder model. However, the proposed scheme with VAEs are generally applicable to empirical databases whose structure may not be transparent. Promising future improvements on this work include adopting variants of VAEs like conditional variational autoencoder (CVAE) [178] for more regularized mappings from the latent space to the decoded samples, a more systematic way of choosing BayesOpt hyperparameters such as Gaussian process kernels and acquisition functions [175], etc. Finally, we wish to extend the work to 3D mechanical metamaterial designs.

Appendix A Computational Homogenization

We use a homogenization method to compute the macroscopic material parameters. For an arbitrary RVE, we impose macroscopic deformations by prescribing the following periodic boundary conditions:

$$\boldsymbol{u}(\boldsymbol{x}) = \overline{\boldsymbol{h}} \cdot \boldsymbol{x} + \boldsymbol{u}^{\star}(\boldsymbol{x}), \tag{4.10}$$

where $\overline{\boldsymbol{h}} = \nabla \overline{\boldsymbol{u}}$ is the macroscopic displacement gradient and is uniform on the RVE. Essentially, Eq. (4.10) decomposes the total displacement of the RVE into a macroscopic (overall) part $\overline{\boldsymbol{u}} = \overline{\boldsymbol{h}} \cdot \boldsymbol{x}$ and a microscopic (fluctuating) part $\boldsymbol{u}^{\star}$. Periodic boundary conditions are applied such that for RVE $ABCD$ in Fig. 4.1, we have $\boldsymbol{u}^{\star}_{AD} = \boldsymbol{u}^{\star}_{BC}$ and $\boldsymbol{u}^{\star}_{AB} = \boldsymbol{u}^{\star}_{DC}$. A fine-mesh FEM calculation is used to solve for $\boldsymbol{u}$ with such boundary conditions described above. Then the macroscopic strain energy density $\overline{W}$ can be obtained via the average of W over the RVE:

$$\overline{W} = V^{-1} \int_V W dV, \tag{4.11}$$

with V being the total volume of the RVE. Note that $\overline{W}$ is also computed using Eq. 4.3. Therefore we are able to obtain an equation for the unknown macroscopic constitutive tensor $\overline{\mathbb{C}}$.

The choice of $\overline{\boldsymbol{h}}$ is not unique, though. In this work, since $\overline{\mathbb{C}}$ has three independent components, we pick three distinct $\overline{\boldsymbol{h}}$:

$$\begin{aligned}
\overline{\boldsymbol{h}}_1 &= \epsilon \mathbf{e}_1 \otimes \mathbf{e}_1, \\
\overline{\boldsymbol{h}}_2 &= \epsilon \mathbf{e}_2 \otimes \mathbf{e}_2, \\
\overline{\boldsymbol{h}}_3 &= \frac{\epsilon}{2}(\mathbf{e}_1 \otimes \mathbf{e}_2 + \mathbf{e}_2 \otimes \mathbf{e}_1),
\end{aligned} \tag{4.12}$$

where ϵ is a small constant. The resulting three independent equations are used to solve for $\overline{\mathbb{C}}$, thus completing the homogenization procedures for such a RVE.

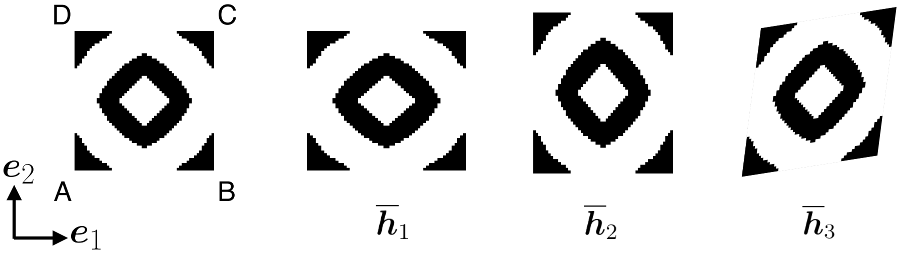

D
C
A
B
$\boldsymbol{e}_2$
$\boldsymbol{e}_1$
$\overline{\boldsymbol{h}}_1$
$\overline{\boldsymbol{h}}_2$
$\overline{\boldsymbol{h}}_3$

Figure 4.1: A sample RVE in its reference configuration (leftmost), and illustration of the deformed configurations with imposed macroscopic displacement gradient conditions (right three).

Appendix B Formulation of a VAE

We present a more detailed description of the formulation of a VAE, particularly the one used in this work. As a generative model, a VAE aims at simulating the underlying data generating process. The observable RVE image data are assumed to be generated from a high dimensional probability distribution $p(x)$ where $x \in \mathbb{R}^{28\times 28}$. Approximating $p(x)$ with a parametrized distribution $p_\theta(x)$ directly is considered not feasible [179]. To see this, the maximum likelihood estimate given dataset $\{x^{(i)}\}$ yields

$$\max_\theta \sum_i \ln\, p_\theta(x^{(i)}) = \max_\theta \sum_i \ln\Big(\int_z p_\theta(x^{(i)}, z)dz\Big), \tag{4.13}$$

where marginalization over the latent variable z is often intractable. Note that z typically lives in a low dimensional space. In our work, $z \in \mathbb{R}^{10}$ serves as the design parameter. Based on the variational principle, an auxiliary function variable $q \in \mathcal{Q}$ (Q is the variational family) is introduced to reformulate Eq. 4.13 as:

$$\max_\theta \sum_i \Big(\ln\, p_\theta(x^{(i)}) - \min_{q\in\mathcal{Q}} \mathbb{KL}\big(q(z)\,||\,p_\theta(z\,|\,x^{(i)})\big)\Big) = \max_\theta \sum_i \max_{q\in\mathcal{Q}} \mathbb{E}_{q(z)}\Big[\ln \frac{p_\theta(x^{(i)}, z)}{q(z)}\Big], \tag{4.14}$$

where $\mathbb{KL}(\,\cdot\,||\,\cdot\,)$ is the Kullback-Leibler divergence [180]. The optimization problem is still rather complicated, though, since it is a bi-level optimization problem and involves a function variable q.

To overcome these difficulties, VAEs adopt an approximate approach. By amortizing the computational cost, VAEs propose to solve the following optimization problem:

$$\max_{\theta,\phi} \sum_i \mathbb{E}_{q_\phi(z\,|\,x^{(i)})}\Big[\ln \frac{p(z)p_\theta(x^{(i)}\,|\,z)}{q_\phi(z\,|\,x^{(i)})}\Big], \tag{4.15}$$

where $q_\phi(z\,|\,x^{(i)})$ is a neural network function parametrized by ϕ (known as the encoder function), and $p_\theta(x^{(i)}\,|\,z)$ is another neural network function parametrized by θ (known as the decoder function). Eq. 4.15 is the loss function of a VAE and can be rewritten into a more familiar format:

$$\min_{\theta,\phi} \sum_i \Big(-\mathbb{E}_{q_\phi(z\,|\,x^{(i)})}\Big[\ln\, p_\theta(x^{(i)}\,|\,z)\Big] + \mathbb{KL}\big(q_\phi(z\,|\,x^{(i)})\,||\,p(z)\big)\Big). \tag{4.16}$$

In Eq. 4.16, the first term is the reconstruction loss of the i^{th} data point $x^{(i)}$ while the second term is the KL-divergence between two multivariate Gaussian distributions.

We randomly split our database (containing 3000 data points) into a training set (80% of the data) and a test set (the rest 20%). We use mini-batch stochastic gradient descent with Adam optimizer [125] to train the VAE with the loss function defined by Eq. 4.16. Learning rate is set to be 10^{-3} while batch size is set to be 64. The training time is typically within 1 minute on a personal computer to achieve a satisfactory model performance.

Chapter 5

Amortized Finite Element Analysis for Fast PDE-Constrained Optimization

The content in this chapter is beyond the scope of only considering mechanical metamaterials. General partial differential equation constrained optimization problems are studied with neural networks. We propose the so-called amortize finite element analysis so that both the forward numerical problems and the inverse design problems are solved with acceleration compared with traditional methods.

5.1 Introduction

Partial differential equations (PDEs) are widely used to describe the properties of physical systems, including heat transfer, electromagnetics, and elasticity. PDE-constrained optimization (PDE-CO) addresses the situation in which an objective function must be minimized or maximized, subject to the constraints of real-world physics as expressed by PDEs. Common examples include optimal design, optimal control, and the identification of parameters to relate a simulation to observed data

[78, 79]. PDE-CO can be computationally challenging, however, for even moderately sized problems, as it is usually necessary to solve the associated PDE at every iteration of the outer-loop optimization.

As a motivating example, consider a heat conduction problem on a unit disk. The physical system is governed by a Poisson's equation:

$$\begin{aligned} -\Delta u &= \lambda \quad \text{in } \Omega, \\ u &= u_b \quad \text{on } \Gamma, \end{aligned} \tag{5.1}$$

where $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$ is the Laplace differential operator, $u(x)$ is the temperature field and $\lambda(x)$ is the heat source field (see Fig. 5.1). Finite element analysis (FEA) is arguably the most powerful method known for computing numerical solutions to this kind of PDE problems. With a given source field $\lambda(x)$, FEA identifies the best approximate solution field $u(x)$ in a piece-wise polynomial function space [16].

PDE-CO poses a higher level problem, seeking to optimize an objective functional jointly over $u(x)$ and $\lambda(x)$, under the constraint imposed by the governing PDE. In the example problem above, one may reasonably ask: how can we design a source field $\lambda(x)$ so that a desired temperature field $u_d(x)$ is fulfilled while the

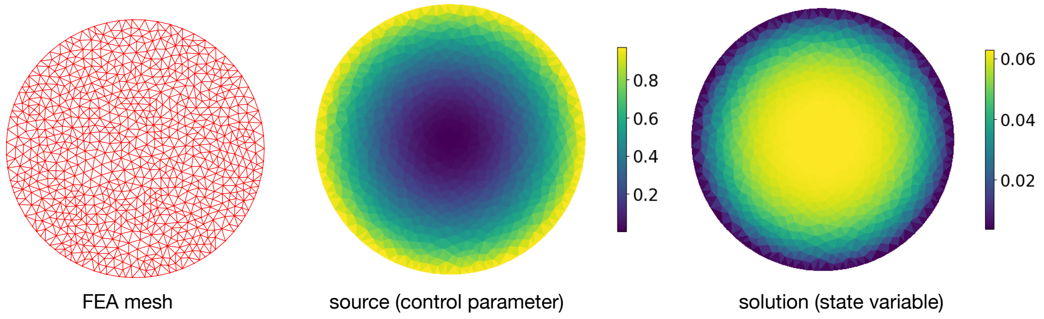


Figure 5.1: The heat equation on a disk of unit radius. Left: the finite element mesh. Middle: an example source field $\lambda(x) = x_1^2 + x_2^2$. Right: the solution field $u(x)$ associated with the source field solved by FEA. $x = (x_1, x_2) \in \mathbb{R}^2$ denote the spatial coordinates.

cost of $\lambda(x)$ is minimized? Such PDE-CO problems are typically high-dimensional (e.g., # of input parameters = 811 in the model problem) and often require iterative procedures that must solve the governing PDE repeatedly using FEA.

We propose a two-stage optimization framework to efficiently tackle sequences of related PDE-constrained optimization problems. At the first stage, we introduce amortized finite element analysis (AmorFEA) to efficiently learn the physics governed by the PDE without requiring supervised data provided by expensive PDE solvers. We borrow the idea of *amortized optimization*, widely used in amortized variational inference [75, 84, 85, 86]. By learning to jump directly to the FEA solution with a neural network, we obtain a surrogate model that is able to predict the solutions directly from the control parameters. At the second stage, we perform gradient-based PDE-CO using the AmorFEA-enabled neural surrogate model. During each optimization iteration, the gradient is efficiently computed via one forward and one backward pass through the neural network, instead of querying an expensive PDE solver as in the traditional adjoint method requires [56].

5.1.1 Related work

Deep Learning for Solving PDEs The idea of approximating the solution to the PDE by a neural network instead of a piece-wise polynomial function (as in FEA) goes back decades [181] and has been continuously explored since then [182, 80, 81]. However, neural networks have only generally shown advantages on high-dimensional PDEs when the finite element mesh is infeasible, as shown by [183]. Recently, [83] proposed to predict PDE solutions directly from parameter fields by training neural networks with a physically-constrained loss. Since their method essentially integrates with the finite *difference* method [184], it only handles structured data in regular domains. Our proposed method, AmorFEA, integrates with the more powerful finite *element* method and benefits from many of the advantages of FEA, such as native support of irregular domains and approximation guarantees [16].

PDE-constrained Optimization The two main workhorses for PDE-constrained optimization problems are the *all-at-once* approach and the *reduced* approach [185]. The all-at-once approach simultaneously optimizes over both the control parameters and the state variables with a strategy such as sequential quadratic programming [186]. Though attractive from an optimization perspective, they are infeasible for large-scale problems with a huge number of state variables to store [187]. The reduced approaches treat the state variables as implicit functions of the control parameters and optimize only over the control parameters. When the control parameters significantly outnumber the objectives (which is often the case), adjoint sensitivity analysis becomes the dominant method under the reduced approaches [55, 56]. In spite of its relative success, the adjoint method requires solutions to the original PDE and the adjoint PDE during each optimization iteration, which is still expensive.

Optimization Using Surrogate Models AmorFEA-based PDE-CO is a surrogate model optimization approach. In optimization problems where the objective function is expensive to evaluate, it is popular to build a surrogate model and perform optimization on the surrogate model instead [188]. There are various approaches for building the surrogate model such as random forests [189], Gaussian processes [175] and Student-t processes [190]. AmorFEA builds a neural network surrogate model for optimization. Although some previous work has used neural network surrogates (e.g., [191]), AmorFEA takes a further step to integrate tightly with finite element analysis.

5.1.2 Contributions

This chapter proposes AmorFEA, a framework for training surrogate PDE solvers for finite element analysis, without supervised training data. As AmorFEA is based on an amortized approach to FEA, it inherits the benefits of naturally handling irregular domains and unstructured meshes. As FEA is a variational formalism, we are able to draw useful connections to variational inference tools developed for probabilistic reasoning—AmorFEA inherits FEA's expressive basis, analagous to performing amortized inference with a rich variational family. The speed and differentiability of AmorFEA make it particularly well-suited to PDE-constrained optimization problems and we show that it can outperform the traditional adjoint method in terms of computation time, while still providing solutions of comparable quality.

5.2 Amortized Finite Element Analysis

Many physical systems governed by PDEs obey variational principles. For a given control field λ (e.g., heat source), the true solution field u (e.g., temperature) is the one that minimizes the total potential energy of the system. Finite element analysis is an approach in which the domain is discretized into a finite set of elements, and the solution u is approximated by a piece-wise polynomial function. This approximation allows us to use a vector $\boldsymbol{u} \in \mathbb{R}^n$ to represent the FEA solution, since the piece-wise polynomial function space is finite-dimensional and forms an isomorphism with $\mathbb{R}^n$. For a full description of FEA, see Appendix A.

In a FEA problem, given a fixed control vector $\boldsymbol{\lambda} \in \mathbb{R}^m$, we find the solution vector (or the state vector) $\boldsymbol{u} \in \mathbb{R}^n$ by solving an optimization problem:

$$\min_{\boldsymbol{u} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda}), \tag{5.2}$$

where $\mathcal{L}(\boldsymbol{u}, \boldsymbol{\lambda})$ denotes the total potential energy.

Our amortized finite element analysis (AmorFEA) approach reframes the per-control-vector optimization process into a shared regression problem. We use a neural network to build a deterministic mapping $g_\psi : \mathbb{R}^m \to \mathbb{R}^n$, whose weights ψ are learned by minimizing the expected potential energy:

$$\min_{\psi} \mathbb{E}_{p(\boldsymbol{\lambda})}[\mathcal{L}(g_\psi(\boldsymbol{\lambda}), \boldsymbol{\lambda})], \tag{5.3}$$

where $p(\boldsymbol{\lambda})$ is the distribution of typical control parameters associated with specific problems and we hope that $\boldsymbol{u} \approx \widehat{\boldsymbol{u}} = g_\psi(\boldsymbol{\lambda})$. Traditional FEA forces the trial solution to be correct by committing to a class of test functions. In a similar sense, AmorFEA forces the correctness of the model by considering a distribution over control parameters.

Since FEA is an approximate variational procedure (in the general sense of minimizing a functional), it resembles variational inference in latent variable models in that the finite element basis functions are analogous to the variational family of distributions and the FEA solution vector mirrors the variational parameters. It can therefore be helpful to think of AmorFEA as a direct analogue to popular amortized variational inference approaches [75, 192] in which the solution to the variational problem is produced via function approximation rather than per-example optimization.

5.2.1 Amortization suboptimality

For amortized variational inference, [193] introduced the notions of *approximation*, *amortization*, and *inference gap*. The approximation gap arises in variational inference due to the inadequacy of the variational family to approximate the true posterior distribution. The amortization gap reflects the difference between the approximate posterior produced by function approximation and the optimal one *within the variational family*. The inference gap is the sum of the approximation and amortization gaps, reflecting the total error in the amortized variational inference scheme.

Similar to gaps for amortized inference, we propose the approximation, amortization and "inference" (total error) gaps for AmorFEA:

$$\Delta_{\text{ap}} = \mathbb{E}_{p(\boldsymbol{\lambda})}\Big[\mathcal{L}(\boldsymbol{u}^{\star}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) - \mathcal{L}(u^e, \lambda)\Big] \tag{5.4}$$

$$\Delta_{\text{am}} = \mathbb{E}_{p(\boldsymbol{\lambda})}\Big[\mathcal{L}(g_{\psi^{\star}}(\boldsymbol{\lambda}), \boldsymbol{\lambda}) - \mathcal{L}(\boldsymbol{u}^{\star}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda})\Big] \tag{5.5}$$

$$\Delta_{\text{inf}} = \Delta_{\text{ap}} + \Delta_{\text{am}}, \tag{5.6}$$

where $\psi^{\star}$ is the optimal solution to Eq. 5.3, $\boldsymbol{u}^{\star}_{\boldsymbol{\lambda}}$ is the optimal solution to Eq. 5.2 for a given $\boldsymbol{\lambda}$, and u^e is the exact solution to the governing PDE which is generally impossible to obtain. Note that we are abusing notation somewhat here in allowing $\mathcal{L}(\cdot, \cdot)$ to take inputs of either functions (e.g., normal font u) or FEA vectors (e.g., bold font $\boldsymbol{u}$).

5.2.2 Computational complexity

Solving a PDE for $\boldsymbol{u} \in \mathbb{R}^n$ by a numerical method such as FEA usually involves solving a large system of equations. A crude estimate of the cost can be described as $\mathcal{O}(dn^r)$, where $d = 1$ if the system is linear and $d > 1$ if it is nonlinear. For the nonlinear system, d can be viewed as the number of iterations required for a nonlinear solver such as Newton-Raphson to converge. The exponent r depends on the sparsity structure of the linear system (or the linearized system for a nonlinear problem) and the algorithm used by the solver. A naïve solver yields $r = 3$, while more efficient solvers usually require case-by-case analysis to take advantage of problem structure.

By amortizing the FEA solving process, we obtain a neural network surrogate function that predicts the state vector $\boldsymbol{u}$ directly from the control vector $\boldsymbol{\lambda}$. Excluding the cost of training time, for a standard multilayer perceptron (MLP) with l equally wide layers of n hidden units, the computational cost is simply the forward pass, which is $\mathcal{O}(ln^2)$. However, AmorFEA requires an up-front training time that we informally think of as $\mathcal{O}(ksln^2)$ where s is the number of training examples and k is the number of training epochs, which tends to be fixed. AmorFEA is able to produce a relatively efficient feed-forward solver, which could be used to save computational resources when an expensive PDE must be solved many times, or when many PDEs must be solved which lie within a given class. In the next subsection, we introduce PDE-CO as such a scenario where AmorFEA can be advantageous.

5.2.3 PDE-constrained optimization

The discretized PDE-constrained optimization is formulated as

$$\begin{aligned} &\min_{\boldsymbol{u}\in\mathbb{R}^n,\boldsymbol{\lambda}\in\mathbb{R}^m} \mathcal{J}(\boldsymbol{u},\boldsymbol{\lambda}) \\ &\text{s.t.} \quad \boldsymbol{c}(\boldsymbol{u},\boldsymbol{\lambda}) = 0, \end{aligned} \tag{5.7}$$

where $\mathcal{J}(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the objective function and $\boldsymbol{c}(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the constraint function imposed by the governing PDE. A reduced formulation is often used to embed the PDE constraint as

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}^m} \widehat{\mathcal{J}}(\boldsymbol{\lambda}), \tag{5.8}$$

where $\widehat{\mathcal{J}}(\boldsymbol{\lambda}) := \mathcal{J}(\boldsymbol{u}(\boldsymbol{\lambda}),\boldsymbol{\lambda})$ and $\boldsymbol{u}(\boldsymbol{\lambda})$ is the implicit function arising from the solution to Eq. 5.2. Gradient-based optimization algorithms require the evaluation of the derivative of the objective function with respect to the control vector:

$$\frac{d\widehat{\mathcal{J}}}{d\boldsymbol{\lambda}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\lambda}} + \frac{\partial \mathcal{J}}{\partial \boldsymbol{\lambda}}. \tag{5.9}$$

A common way to compute this gradient is to use the adjoint method [56]. Taking the derivative of Eq. 5.7 with respect to $\boldsymbol{\lambda}$ yields chains of Jacobian matrices:

$$\frac{d\boldsymbol{c}}{d\boldsymbol{\lambda}} = \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\frac{d\boldsymbol{u}}{d\boldsymbol{\lambda}} + \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\lambda}} = 0. \tag{5.10}$$

Hence,

$$\frac{d\boldsymbol{u}}{d\boldsymbol{\lambda}} = -\Big(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\Big)^{-1}\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\lambda}}. \tag{5.11}$$

Substitute Eq. 5.11 to Eq. 5.9, we obtain

$$\frac{d\widehat{\mathcal{J}}}{d\boldsymbol{\lambda}} = -\overbrace{\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}}\underbrace{\Big(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\Big)^{-1}}_{}}^{\text{adjoint PDE}}\!\!\!\!\!\!\!\!\!\!\underbrace{\phantom{\Big(\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}\Big)^{-1}}\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{\lambda}}}_{\text{tangent linear PDE}} + \frac{\partial \mathcal{J}}{\partial \boldsymbol{\lambda}}. \tag{5.12}$$

Resembling the two different modes of automatic differentiation, we could choose to either solve the adjoint PDE first (reverse-mode) or the tangent linear PDE first (forward-mode). When the size of the control vector is larger than that of the objective (e.g., $m \gg 1$ in our case), it is more efficient to solve the adjoint PDE first, giving the name adjoint method. The adjoint PDE is a linear PDE to solve, but it also relies on the Jacobian matrix $\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{u}}$, which requires the solution vector $\boldsymbol{u}$. As shown, the cost for solving the governing PDE is $\mathcal{O}(dn^r)$, which dominates the total cost of using the adjoint method in one iteration of PDE-CO.

We propose accelerating PDE-CO with AmorFEA. AmorFEA yields a differentiable map from the control vector $\boldsymbol{\lambda}$ to the state vector $\boldsymbol{u}$, which does not require solving the governing PDE via traditional FEA, and which can be used to approximate the costly term $\frac{d\boldsymbol{u}}{d\boldsymbol{\lambda}}$ in Eq. 5.9. With the differentiable neural surrogate model, we can formulate the PDE-constrained optimization problem as

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}^m} \widetilde{\mathcal{J}}(\boldsymbol{\lambda}), \tag{5.13}$$

where $\widetilde{\mathcal{J}}(\boldsymbol{\lambda}) := \mathcal{J}(g_{\psi^\star}(\boldsymbol{\lambda}), \boldsymbol{\lambda})$ and $\psi^\star$ are the learned weights of the neural network. The derivative is given by

$$\frac{d\widetilde{\mathcal{J}}}{d\boldsymbol{\lambda}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}} \frac{dg_{\psi^\star}}{d\boldsymbol{\lambda}} + \frac{\partial \mathcal{J}}{\partial \boldsymbol{\lambda}}. \tag{5.14}$$

We employ reverse-mode automatic differentiation, since $m \gg 1$. By using a standard MLP, the total cost within each iteration of the PDE-CO is $\mathcal{O}(ln^2)$, and so guaranteed to be relatively efficient.

A summary for the procedures of AmorFEA along with AmorFEA based PDE-CO can be found in Fig. 5.2 by visualizing the computation graph.

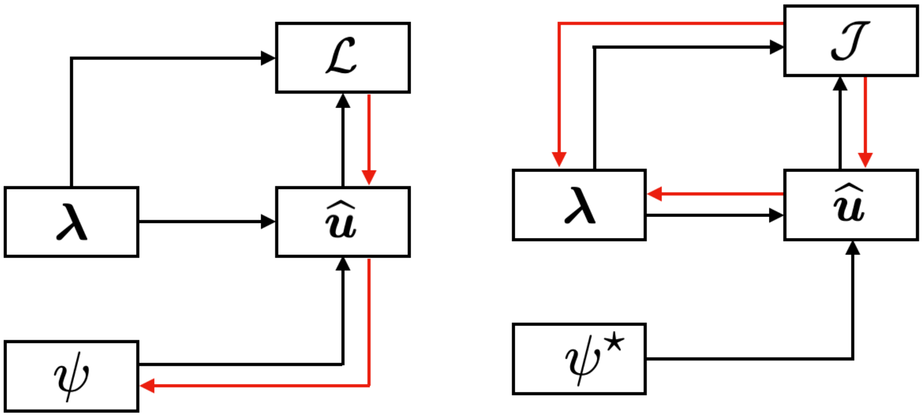


Figure 5.2: Computation graph for AmorFEA based PDE-CO. Red arrows denote automatic differentiation. Left: AmorFEA training for the surrogate model. Right: PDE-CO with the learned model.

5.3 Linear Models

An important and general class of PDE-CO problems impose a linear relationship between the control parameter and the state variable. These linear problems provide a useful testbed for examining the AmorFEA approach. We focus on the model problem in Section 5.1, where the source vector $\boldsymbol{\lambda}$ is linearly related to the solution vector $\boldsymbol{u}$. We show that by employing a linear regression model and performing AmorFEA, we are able to fully recover the underlying physics and achieve an amortization gap of zero; this result is unsurprising due to the assumption of a linear relationship between $\boldsymbol{u}$ and $\boldsymbol{\lambda}$. Though simple to analyze, the linear case gives intuition about the proposed scheme. We also compare AmorFEA with supervised learning, where we run FEA simulations to obtain labeled data and train the linear model in a traditional fashion. FEA simulations are carried out using an open source Python package `FEniCS` [123]. Neural network training is performed in `PyTorch` [194]. We show that both AmorFEA and supervised training have the same global optimality condition.

We use AmorFEA to train a single-layer network that predicts $\boldsymbol{u} \in \mathbb{R}^n$ from $\boldsymbol{\lambda} \in \mathbb{R}^m$, where $m = 811$ and $n = 721$ in this case. We assume the distribution over source terms $\boldsymbol{\lambda}$ is a uniform distribution on the hypercube $[-1, 1]^m$. $10,000$ samples were drawn from this distribution to form the training data. FEA converts the minimization problem in Eq. 5.2 into a linear system to solve:

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{B}\boldsymbol{\lambda}, \tag{5.15}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ arise from the assembly procedure of the weak form in FEA (see details in Appendix A). Let us denote $\boldsymbol{f} = \boldsymbol{B}\boldsymbol{\lambda}$. The linear model we employ has a weight matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ and no bias vector so that $\widehat{\boldsymbol{u}} = \boldsymbol{W}\boldsymbol{f}$. The potential energy in this case is therefore equivalent to the least-squares loss for the solution to the linear system, preconditioned by the stiffness matrix $\boldsymbol{A}$:

$$\mathcal{L}_a(\boldsymbol{W}; \boldsymbol{f}) = \frac{1}{2}(\boldsymbol{W}\boldsymbol{f})^\top \boldsymbol{A}(\boldsymbol{W}\boldsymbol{f}) - \boldsymbol{f}^\top(\boldsymbol{W}\boldsymbol{f}). \tag{5.16}$$

As an instantiation of Eq. 5.3 using the Monte Carlo estimate of the expectation, AmorFEA leads to the following optimization problem:

$$\min_{\boldsymbol{W}\in\mathbb{R}^{n\times n}} \overline{\mathcal{L}}_a(\boldsymbol{W}), \tag{5.17}$$

where

$$\overline{\mathcal{L}}_a(\boldsymbol{W}) = \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{1}{2}(\boldsymbol{W}\boldsymbol{f}_k)^\top\boldsymbol{A}(\boldsymbol{W}\boldsymbol{f}_k) - \boldsymbol{f}_k^\top(\boldsymbol{W}\boldsymbol{f}_k)\Big). \tag{5.18}$$

Next, we study supervised learning in the same environment. We expand the empirical data set $\mathcal{D} = \{\boldsymbol{\lambda}^{(i)}\}$ to $\mathcal{D}' = \{(\boldsymbol{\lambda}, \boldsymbol{u})^{(i)}\}$ by running FEA simulations. The supervised loss function for each data point is defined as

$$\mathcal{L}_s(\boldsymbol{W};\boldsymbol{f}) = \frac{1}{2}\|\widehat{\boldsymbol{u}} - \boldsymbol{u}\|_2^2 = \frac{1}{2}\|\boldsymbol{W}\boldsymbol{f} - \boldsymbol{A}^{-1}\boldsymbol{f}\|_2^2. \tag{5.19}$$

Supervised training solves the following minimization problem:

$$\min_{\boldsymbol{W}\in\mathbb{R}^{n\times n}} \overline{\mathcal{L}}_s(\boldsymbol{W}), \tag{5.20}$$

where

$$\overline{\mathcal{L}}_s(\boldsymbol{W}) = \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{1}{2}(\boldsymbol{W}\boldsymbol{f}_k - \boldsymbol{A}^{-1}\boldsymbol{f}_k)^\top(\boldsymbol{W}\boldsymbol{f}_k - \boldsymbol{A}^{-1}\boldsymbol{f}_k)\Big). \tag{5.21}$$

We show that both $\overline{\mathcal{L}}_a(\boldsymbol{W})$ and $\overline{\mathcal{L}}_s(\boldsymbol{W})$ are convex (see proofs in Appendix B).

Proposition 1. *The empirical loss function $\overline{\mathcal{L}}_a(\boldsymbol{W})$ of AmorFEA in the linear model is convex.*

Proposition 2. *The empirical loss function $\overline{\mathcal{L}}_s(\boldsymbol{W})$ of supervised learning in the linear model is convex.*

We also see that the first order condition (FOC) for AmorFEA gives

$$\frac{\partial \overline{\mathcal{L}}_a}{\partial \boldsymbol{W}} = (\boldsymbol{A}\boldsymbol{W} - \boldsymbol{I})\Big(\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{f}_k \boldsymbol{f}_k^\top\Big) = \boldsymbol{0}. \tag{5.22}$$

Similarly, the FOC for supervised training gives

$$\frac{\partial \overline{\mathcal{L}}_s}{\partial \boldsymbol{W}} = (\boldsymbol{W} - \boldsymbol{A}^{-1})\Big(\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{f}_k \boldsymbol{f}_k^\top\Big) = \boldsymbol{0}. \tag{5.23}$$

By Proposition 1 and Proposition 2 along with Eq. 5.22 and Eq. 5.23, we conclude that both AmorFEA and supervised learning achieve the global minimum with same condition $\boldsymbol{W} = \boldsymbol{A}^{-1}$. The only difference between these problems is the “physical” preconditioning from $\boldsymbol{A}$. As a supporting experiment, we show in Fig. 5.3 that both AmorFEA and supervised training have similar capabilities to recover the underlying model.

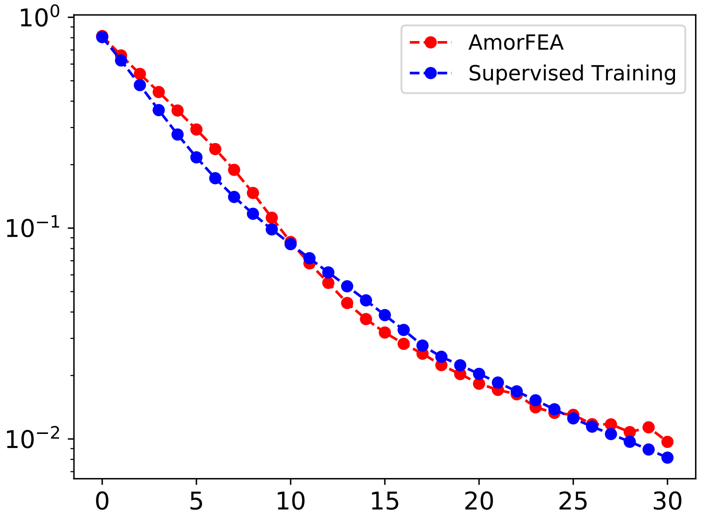


Figure 5.3: Normalized error versus training epochs. We define the normalized error $\epsilon = \frac{\|\boldsymbol{W}-\boldsymbol{A}^{-1}\|_\infty}{\|\boldsymbol{A}^{-1}\|_\infty}$.

Since the fully trained linear model is able to produce $\widehat{\boldsymbol{u}} = \boldsymbol{u}$, we have a zero amortization gap as computed by Eq. 5.5. We do not study PDE-CO in the linear case.

5.4 Nonlinear Models

Method	α	Optimized objective [10^{-3}]	Wall time [ms]
Adjoint Method	10^{-6}	0.338 ± 0.002	728.2 ± 5.5
	10^{-3}	2.599 ± 0.075	511.6 ± 223.2
	1	2.587 ± 0.012	15370.0 ± 1804.3
AmorFEA	10^{-6}	0.340 ± 0.003	39.7 ± 5.0
	10^{-3}	2.613 ± 0.104	71.5 ± 63.1
	1	2.593 ± 0.008	838.5 ± 208.4

Table 5.1: PDE-CO results for the adjoint method and AmorFEA. For different regularity coefficient α, AmorFEA achieves similar optimal objectives compared with the adjoint method, but with less wall time.

We examine two more realistic nonlinear settings and perform PDE-CO in this section. Here the solution vector $\boldsymbol{u}$ and the control vector $\boldsymbol{\lambda}$ have a nonlinear relationship, which naturally motivates the use of a neural network as the predictive mapping.

5.4.1 Source field finding

We consider a two-dimensional optimal source control problem simplified from superconductivity theory [195]. Intuitively, we have a known, target field $u_d(x)$ that we want our physical field $u(x)$ to achieve. We accomplish this by imposing an appropriate source field $\lambda(x)$. To save cost, we prefer the magnitude of $\lambda(x)$ to be small. The solution field $u(x)$ and the source field $\lambda(x)$ are related by the governing PDE (Eq. 5.25).

Mathematically, the problem is to minimize the functional

$$\mathcal{J}(u, \lambda) = \frac{1}{2}\int_{\Omega}(u - u_d)^2 dx + \frac{\alpha}{2}\int_{\Omega}\lambda^2 dx \tag{5.24}$$

subject to a nonlinear Poisson's equation with Dirichlet boundary conditions:

$$\begin{aligned} -\Delta u + 10(u + u^3) &= \lambda \quad \text{in } \Omega, \\ u &= u_b \quad \text{on } \Gamma, \end{aligned} \tag{5.25}$$

where we set a constant boundary condition $u_b = 1$. We use Fig. 5.4 to further demonstrate the setup of this problem. Note that we employ an irregular domain with a dolphin-shaped hole in the middle. The irregular domain cannot be discretized using a structured mesh (a lattice). Many FEA benchmark cases are performed in this domain with `dolfin` [196], the finite element computing component of `FEniCS`.

The potential energy $\mathcal{L}(u, \lambda)$ from which the governing PDE (Eq. 5.25) can be derived is described in Appendix C. We use AmorFEA to train a neural network that predicts $\boldsymbol{u} \in \mathbb{R}^n$ from $\boldsymbol{\lambda} \in \mathbb{R}^m$, where $m = n = 759$ in this case. The assumed distribution over $\boldsymbol{\lambda}$ is constructed from a zero-mean Gaussian process given by

$$\begin{aligned} f(x) &\sim \mathcal{GP}\big(\mu(\cdot), k(\cdot, \cdot)\big) \\ \mu(x) &= 0 \\ k(x^{(i)}, x^{(j)}) &= \sigma^2 \exp\Big(- \frac{\|x^{(i)} - x^{(j)}\|_2^2}{l^2} \Big), \end{aligned} \tag{5.26}$$

where we set the output variance $\sigma = 10^2$ and the lengthscale $l = 0.1$. To construct the training and testing data from this distribution, 30,000 source terms were generated. Compared with supervised data generated by expensive FEA simulations, our data are almost free to obtain.

For the neural network model, we use a MLP with "scaled exponential linear units" (SELUs) for the activation functions [197]. We perform a 90/10 train-test split for our data. For the 10% test data, we run FEA simulation so that we can report test error by comparing the AmorFEA solutions with the FEA solutions. Since our data

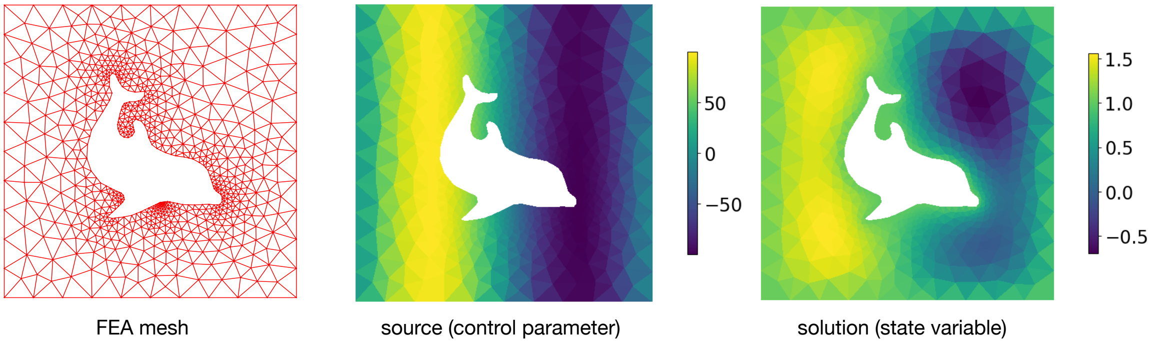


Figure 5.4: Setup of the problem. Left: the finite element mesh for a irregular domain with a dolphin-shaped hole. Middle: an example source field $\lambda(x) = 100\text{sin}(2\pi x_1)$. Right: the solution field $u(x)$ associated with the source field.

	MLP-0	MLP-1	MLP-2
Δ_{am}	0.1593 ± 0.0032	0.0227 ± 0.0003	0.0156 ± 0.0002
ϵ	0.0468 ± 0.0004	0.0110 ± 0.0002	0.0086 ± 0.0001

Table 5.2: Test performance. The amortization gap Δ_{am} is computed according Eq. 5.5. The relative error ϵ is computed as Eq. 5.27. The number suffix to "MLP" refers to the number of hidden layers.

represent functions, it is more meaningful to define an error quantification metric for functions than to use simple mean squared error (MSE). We define the relative error ϵ using norms on the $\mathcal{L}^2(\Omega)$ space:

$$\epsilon = \mathbb{E}_{p(\boldsymbol{\lambda})}\Big[\frac{\|u^\star - u^a\|_{\mathcal{L}^2}}{\|u^\star\|_{\mathcal{L}^2}}\Big], \tag{5.27}$$

where $u^\star$ is the FEA solution and u^a is the AmorFEA solution. With FEA simulations available in the test set, we can also compute the amortization gap introduced in Eq. 5.5. Since analytical solutions are generally not possible, we cannot explicitly evaluate the approximation gap or the inference gap. However, the approximation gap is a well-studied subject in the literature on finite element analysis [16].

We run experiments using MLPs with different layers and report the amortization gap and relative error in Table 5.2. The layers in all MLPs have equal widths.

As shown in the table, a single-layer network cannot perfectly solve this nonlinear problem, in contrast to the linear problem in Section 5.3. We observe that both the amortization gap and the relative error decrease with deeper neural networks. Using bigger neural networks may give us smaller amortization gaps, but there is a trade-off between training time and performance. We show next that a MLP with 2 hidden layers provides adequate accuracy for the PDE-CO problem while being trained cheaply.

We solve the aforementioned PDE-constrained optimization problem using both the adjoint method (see Eq. 5.12) and the AmorFEA based method (see Eq. 5.14). We use `dolfin-adjoint` [196], an open source package written in the Python interface to `FEniCS` for implementing the adjoint method. We vary the values of the regularity coefficient α in Eq. 5.24 and run PDE-CO experiments using the conjugate gradient method [198] in `SciPy` [199]. We record the wall-clock time for different cases and show the results in Table 5.1. Note that since AmorFEA-based PDE-CO solves a surrogate model optimization problem, the objective function is defined differently (see Eq.5.8 and Eq.5.13). We have considered this difference and reported the reconstructed true objective function for AmorFEA based PDE-CO in Table 5.1. As shown, AmorFEA based PDE-CO achieves similar optimized objective values compared with the adjoint method, but with significantly less computation time, excluding time to train the network. However, it took approximately 5 minutes to train the neural network; thus AmorFEA is most interesting when such problems need to be solved repeatedly for, e.g., optimal control of a source.

5.4.2 Inverse kinematics of a soft robot

In the second nonlinear example, we consider the control of a snake-like soft robot made of elastic material. Such robots represent promising solutions to minimally invasive surgery [200]. We control the static position of the robot by expanding or contracting "muscles" on the left and right sides of the robot, quantitatively described by an actuation field $\lambda(x)$ that controls the stretch ratio on the two sides. As shown in Fig. 5.5, the bottom side of the robot is fixed while the top side is free to deform. While a forward PDE problem solves for the displacement field $u(x)$ with given $\lambda(x)$, we here consider the PDE-constrained optimization problem of inverse kinematics, i.e., determining the appropriate actuation field to achieve an end-effector location. We focus on the middle point x_0 at the top of the robot and specify an arbitrary two-dimensional displacement u_0 that we hope this tip point can achieve by optimizing the actuation field $\lambda(x)$.

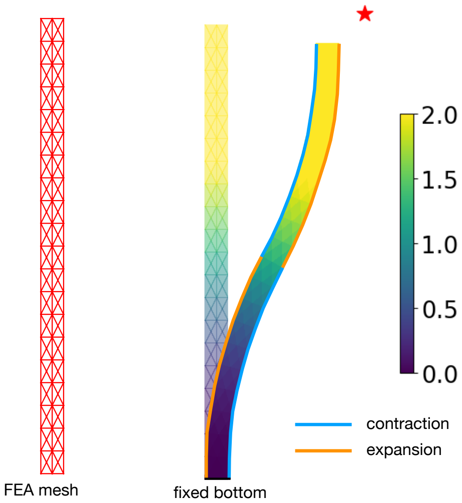


Figure 5.5: Setup of the problem. Left: the finite element mesh for the soft robot in an undeformed configuration. Right: with the actuation field $\lambda(x)$ set to be half-and-half for contraction and expansion, but upside down for the two sides, the robot can deform to a specific configuration. Colors indicate the displacement magnitude $\|u\|_2$.

Mathematically, the problem is to minimize the functional

$$\mathcal{J}(u, \lambda) = \|u(x_0) - u_0)\|_2^2 \tag{5.28}$$

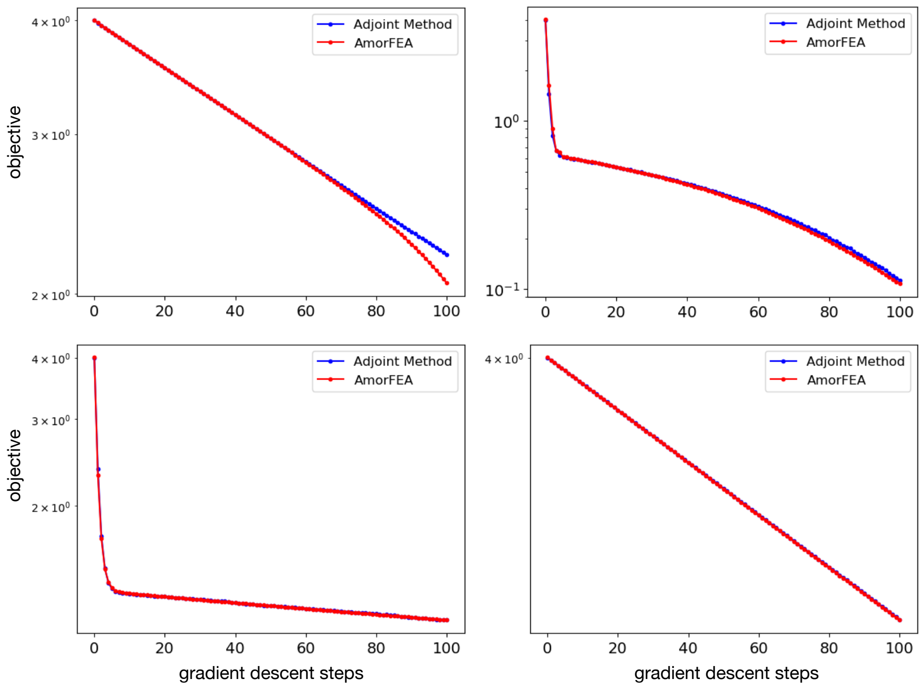


Figure 5.6: Objective function versus gradient descent steps for both AmorFEA and the adjoint method. The four cases correspond to the same scenarios in Fig. 5.7.

subject to an equilibrium equation for hyperelastic material with appropriate boundary conditions:

$$\begin{aligned} \text{Div } P(u) &= 0 \quad \text{in } \Omega, \\ r(u, \lambda) &= 0 \quad \text{on } \Gamma, \end{aligned} \tag{5.29}$$

where "Div" is the divergence operator, P is the a second-order stress tensor and r is the boundary constraint. For more precise definitions of these terms, refer to Appendix C. Since the boundary constraint in Eq. 5.29 is unusual, there is no direct implementation in `FEniCS`. We implement both the FEA algorithms and the adjoint method with our custom code for this problem.

The potential energy $\mathcal{L}(u, \lambda)$ from which the governing PDE (Eq. 5.29) can be derived is known as the strain energy, fully described in Appendix C. We use AmorFEA to train a neural network to predict $\boldsymbol{u} \in \mathbb{R}^n$ from $\boldsymbol{\lambda} \in \mathbb{R}^m$, where $m = 40$ and $n = 206$

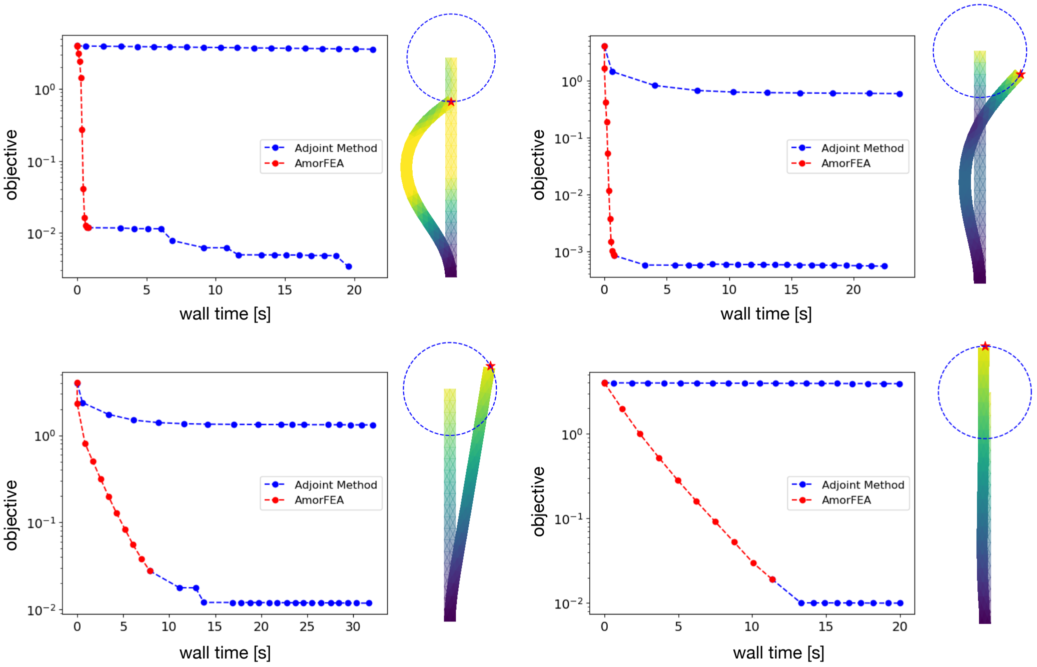


Figure 5.7: Wall time measurement for both the adjoint method and AmorFEA. There are four target displacements for the tip point displacement $u(x_0)$ to achieve. In these plots, one blue dot indicate one gradient descent step of the adjoint method, while one red dot indicate multiple gradient steps for AmorFEA. Following the row-first order, the numbers of steps per red dot are $(40, 40, 400, 600)$. The gradient descent step size is set to be consistent within each case, but different across the four cases: $(10^{-2}, 10^{-2}, 2 \times 10^{-3}, 2 \times 10^{-3})$.

in this case. The assumed distribution over actuation fields is constructed from a uniform distribution over the hypercube $[-a, a]^m$ with $a > 0$. $30,000$ training data were generated from this distribution. Training is similar to the source-finding case in Section 5.4.1. However, the potential energy $\mathcal{L}(u, \lambda)$ in this case is sensitive to the displacement u. For a displacement field that causes any overlap of the deformed robot, the energy will be infinite. We take a warm-start strategy to train the network by varying a from 0.1, 0.2, 0.4 until 0.8, and train the network successively. Training took $5 - 10$ minutes.

To demonstrate the ability of the resulting neural network to enable solutions to the inverse kinematics problem via PDE-CO of the actuation field, we pick four equally-spaced target points on a circle centered at the robot tip and perform PDE-CO to find actuations. As shown in Fig. 5.7, we use gradient descent for both the adjoint method and AmorFEA. AmorFEA consumes significantly less time than the adjoint method. However, since AmorFEA-based PDE-CO uses an approximate surrogate model, we may expect the objective to plateau before reaching the true minimum. One may view this as an asymptotic bias, which can be remedied by "fine tuning" the AmorFEA result with the adjoint method. From this perspective, AmorFEA provides a fast and accurate initial guess for the non-convex PDE-CO problem.

We additionally demonstrate that AmorFEA follows a similar optimization path to the adjoint method, by showing the objective function versus the number of gradient descent steps in Fig. 5.6 for each of the four cases. Wall time measurements in [s] of these four cases are $(119, 218, 133, 113)$ for the adjoint method and $(0.215, 0.210, 0.226, 0.249)$ for AmorFEA. For completeness, we also explored gradient-free techniques such as the Nelder–Mead method [201], but they typically converged to an unsatisfactory local minimum.

5.5 Limitations and Future Work

Compared to traditional FEA, AmorFEA consumes additional computational resources up front for training the model. To accommodate the training time, AmorFEA only starts to show its edge when a sequence of related problems need to be solved. Additionally, we note that the current formulation requires the governing PDE to be derived from minimization of a potential energy, but this is not applicable for all PDEs, which limits the scope of this work.

Promising directions for future work include a more systematic study of the amortization gap in nonlinear problems, using higher order finite element basis functions for AmorFEA, and parameterization of the domain for topology optimization in PDE-CO.

5.6 Conclusions

In this work, we proposed a novel formulation (AmorFEA) that amortizes the solving process of classic finite element analysis. AmorFEA enables fast, differentiable prediction of PDE solutions, which accelerates PDE-constrained optimization. We quantitatively studied the amortization gap for both linear problems and nonlinear problems. Numerical experiments show that our method outperforms the traditional adjoint method on a per-iteration basis.

Appendix A Finite Element Analysis

The Finite Element Analysis (FEA) is arguably the most powerful approach known for the numerical solutions of problems characterized by partial differential equations (PDEs). We demonstrate the basic idea by considering the linear Poisson's equation as a model problem. As shown in Fig. 5.8, we start with the *strong formulation* of the problem and then introduce the *weak formulation* upon which the finite element approximation is built, namely, the *Galerkin weak formulation.* We further introduce the *minimization formulation* and the corresponding *Galerkin minimization formulation*, which is the cornerstone for the proposed amortized finite element analysis (AmorFEA).

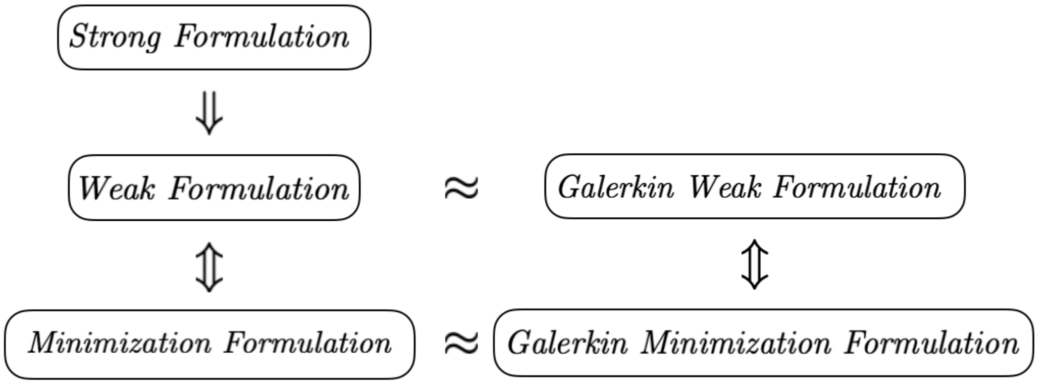


Figure 5.8: FEA roadmap.

The strong formulation of the Poisson's equation reads as

$$\begin{aligned} -\Delta u &= \lambda \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Gamma. \end{aligned} \tag{5.30}$$

For simplicity, we have assumed homogenous Dirichlet boundary conditions. Instead of heuristically approximating the differential operator like the Finite Difference Method (FDM), FEA employs a more systematic approach by searching the best possible solution over a constructed finite-dimensional function space.

First, let us multiply a test function v for both sides of the PDE in Eq. 5.30, integrate over Ω, and use integration by parts to obtain the weak formulation: find $u \in \mathbb{V} := \{v \text{ sufficiently smooth} \mid v|_\Gamma = 0\}$ such that

$$a(u, v) = l(\lambda, v), \text{ for all } v \in \mathbb{V}, \tag{5.31}$$

where

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx, \quad l(\lambda, v) = \int_\Omega \lambda v \, dx. \tag{5.32}$$

Second, we construct a finite-dimensional subspace $\mathbb{V}_h \subset \mathbb{V}$ and $\mathbb{V}_h = \text{span}\{\phi_1, ..., \phi_n\}$ is the piece-wise polynomial function space. Note that for any function $v \in \mathbb{V}_h$, there is a unique representation $v = \sum_{i=1}^n v_i \phi_i$. We thus can define an isomorphism $\mathbb{V}_h \cong \mathbb{R}^n$ by $v = \sum_{i=1}^n v_i \phi_i \leftrightarrow \boldsymbol{b} = (v_1, ..., v_n)^\top$.

The Galerkin weak formulation yields an approximation to Eq. 5.31 by solving the following system of linear equations:

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}, \tag{5.33}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the stiffness matrix with $A_{ij} = a(\phi_i, \phi_j)$, $\boldsymbol{u} \in \mathbb{R}^n$ is the solution vector and $\boldsymbol{f} \in \mathbb{R}^n$ is the source vector with $f_i = l(\phi_i, \lambda)$. As a remark, if λ is also represented by piece-wise polynomial basis functions, we could further write $\boldsymbol{f} = \boldsymbol{B}\boldsymbol{\lambda}$ with $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ and $B_{ij} = l(\phi_i, \phi_j)$. It is not difficult to verify that $\boldsymbol{A}$ is symmetric and positive definite. By solving Eq. 5.33, the FEM gives the best possible solution we can find in $\mathbb{V}_h$.

The minimization formulation states that

$$\min_{u\in\mathbb{V}} \mathcal{L}(u), \tag{5.34}$$

where

$$\mathcal{L}(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \, dx - \int_\Omega \lambda u \, dx. \tag{5.35}$$

By setting the functional derivative to be zero, we recover the weak formulation.

Finally, we introduce the Galerkin minimization formulation, which is the discretized version of the minimization formulation above. Replace $u \in \mathbb{V}$ in Eq. 5.34 with $u = \sum_{i=1}^n u_i\phi_i \in \mathbb{V}_h$, we get the following finite-dimensional optimization problem:

$$\min_{\boldsymbol{u}\in\mathbb{R}^n} \mathcal{L}(\boldsymbol{u}), \tag{5.36}$$

where

$$\mathcal{L}(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}^\top \boldsymbol{u}. \tag{5.37}$$

The quadratic programming problem yields the same solution as Eq. 5.33. AmorFEA is based on the Galerkin minimization formulation. Note that for a nonlinear problem, we may have a more complicated nonlinear optimization problem rather than the quadratic programming.

For further details regarding FEA and its variational formulation, we refer readers to [202].

Appendix B Proofs

Lemma 1. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if the function $g : \mathbb{R} \to \mathbb{R}$ given by $g(t) = f(\boldsymbol{x} + t\boldsymbol{y})$ is convex (as a univariate function) for all $\boldsymbol{x}$ in domain of f and all $\boldsymbol{y} \in \mathbb{R}^n$. (The domain of g here is all t for which $\boldsymbol{x} + t\boldsymbol{y}$ is in the domain of f.)*

Proof. This is straightforward from the definition of convexity of multivariable functions. □

Proposition 3. *The empirical loss function $\overline{\mathcal{L}}_a(\boldsymbol{W})$ of AmorFEA in the linear model is convex.*

Proof. We use Lemma 1 to show the function $\overline{\mathcal{L}}_a : \mathbb{R}^{n\times n} \to \mathbb{R}$ is convex. For any fixed $\boldsymbol{W}_1 \in \mathbb{R}^{n\times n}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{n\times n}$, let

$$
\begin{aligned}
g(t) =& \overline{\mathcal{L}}_a(\boldsymbol{W}_1 + t\boldsymbol{W}_2) \\
=& \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{1}{2}\big((\boldsymbol{W}_1 + t\boldsymbol{W}_2)\boldsymbol{f}_k\big)^\top \boldsymbol{A}\big((\boldsymbol{W}_1 + t\boldsymbol{W}_2)\boldsymbol{f}_k\big) - \boldsymbol{f}_k^\top\big((\boldsymbol{W}_1 + t\boldsymbol{W}_2)\boldsymbol{f}_k\big)\Big) \\
=& \Big(\frac{1}{2K}\sum_{k=1}^{K}\boldsymbol{f}_k^\top\boldsymbol{W}_2^\top\boldsymbol{A}\boldsymbol{W}_2\boldsymbol{f}_k\Big)t^2 - \Big(\frac{1}{2K}\sum_{k=1}^{K}\boldsymbol{f}_k^\top\boldsymbol{W}_1\boldsymbol{f}_k)\Big) \\
&+ \Big(\frac{1}{2K}\sum_{k=1}^{K}\big(\boldsymbol{f}_k^\top\boldsymbol{W}_2^\top\boldsymbol{A}\boldsymbol{W}_1\boldsymbol{f}_k + \boldsymbol{f}_k^\top\boldsymbol{W}_1^\top\boldsymbol{A}\boldsymbol{W}_2\boldsymbol{f}_k - 2\boldsymbol{f}_k^\top\boldsymbol{W}_2\boldsymbol{f}_k\big)\Big)t
\end{aligned}
\tag{5.38}
$$

Since $\boldsymbol{A}$ is symmetric and positive definite, the coefficient of the quadratic term is always positive:

$$\frac{1}{2K}\sum_{k=1}^{K}\boldsymbol{f}_k^\top \boldsymbol{W}_2^\top \boldsymbol{A}\boldsymbol{W}_2\boldsymbol{f}_k = \frac{1}{2K}\sum_{k=1}^{K}\|\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{W}_2\boldsymbol{f}_k\|_2^2 \geq 0. \tag{5.39}$$

This shows that $g(t)$ is always convex. □

Proposition 4. *The empirical loss function $\overline{\mathcal{L}}_s(\boldsymbol{W})$ of supervised learning in the linear model is convex.*

Proof. We use Lemma 1 to show the function $\overline{\mathcal{L}}_s : \mathbb{R}^{n\times n} \to \mathbb{R}$ is convex. For any fixed $\boldsymbol{W}_1 \in \mathbb{R}^{n\times n}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{n\times n}$, let

$$\begin{aligned}
g(t) =& \overline{\mathcal{L}}_s(\boldsymbol{W}_1 + t\boldsymbol{W}_2)\\
=& \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{1}{2}\big((\boldsymbol{W}_1 + t\boldsymbol{W}_2)\boldsymbol{f}_k - \boldsymbol{A}^{-1}\boldsymbol{f}_k\big)^\top\big((\boldsymbol{W}_1 + t\boldsymbol{W}_2)\boldsymbol{f}_k - \boldsymbol{A}^{-1}\boldsymbol{f}_k\big)\Big)\\
=& \Big(\frac{1}{2K}\sum_{k=1}^{K}\boldsymbol{f}_k^\top \boldsymbol{W}_2^\top \boldsymbol{W}_2\boldsymbol{f}_k\Big)t^2 + \Big(\frac{1}{2K}\sum_{k=1}^{K}\boldsymbol{f}_k^\top(\boldsymbol{W}_1 - \boldsymbol{A}^{-1})^\top(\boldsymbol{W}_1 - \boldsymbol{A}^{-1})\boldsymbol{f}_k\Big)\Big)\\
&- \Big(\frac{1}{2K}\sum_{k=1}^{K}\big(\boldsymbol{f}_k^\top(\boldsymbol{W}_1 - \boldsymbol{A}^{-1})^\top\boldsymbol{W}_2\boldsymbol{f}_k + \boldsymbol{f}_k^\top\boldsymbol{W}_2^\top(\boldsymbol{W}_1 - \boldsymbol{A}^{-1})\boldsymbol{f}_k\big)\Big)t
\end{aligned} \tag{5.40}$$

The coefficient of the quadratic term is always positive:

$$\frac{1}{2K}\sum_{k=1}^{K}\boldsymbol{f}_k^\top \boldsymbol{W}_2^\top \boldsymbol{W}_2\boldsymbol{f}_k = \frac{1}{2K}\sum_{k=1}^{K}\|\boldsymbol{W}_2\boldsymbol{f}_k\|_2^2 \geq 0. \tag{5.41}$$

This shows that $g(t)$ is always convex. □

Appendix C Detailed Formulations

Source Field Finding The potential energy is

$$\mathcal{L}(u,\lambda) = \frac{1}{2}\int_\Omega \|\nabla u\|_2^2 dx + \frac{1}{2}\int_\Omega (10u^2 + 5u^4) dx - \int_\Omega \lambda u dx. \tag{5.42}$$

The governing PDE (Eq. 5.25) can be derived by minimizing Eq. 5.42.

We set $\lambda(x) = 100\exp(\frac{\|x-(0.1,0.1)\|_2^2}{0.02})$, solve the governing PDE by FEA and set the solution to be the desired field $u_d(x)$ for Eq. 5.24.

Inverse Kinematics of a Soft Robot We model the soft robot with a neo-Hookean hyperelastic solid [203]. The total potential energy is

$$\mathcal{L}(u,\lambda) = \int_\Omega W dx, \tag{5.43}$$

where the energy density W is defined for material bulk and shear moduli μ and κ as:

$$W = \frac{\mu}{2}\Big((\det F)^{-2/3}\text{tr}(FF^T) - 3\Big) + \frac{\kappa}{2}(\det F - 1)^2, \tag{5.44}$$

and F is the deformation gradient,

$$F = \nabla u + I. \tag{5.45}$$

Since our problem is in 2d, we have assumed plain strain condition. The governing PDE (Eq. 5.29) can be derived by minimization Eq. 5.43. The stress tensor P in Eq. 5.29 is known as the first Piola-Kirchoff stress and can be obtained by

$$P = \frac{\partial W}{\partial F}. \tag{5.46}$$

The boundary constraint $r(u,\lambda)$ in Eq. 5.29 is expressed over the bottom side boundary Γ_b, top side boundary Γ_t and the side boundaries Γ_s respectively as

$$u = 0 \qquad \text{on} \;\; \Gamma_b, \tag{5.47}$$

$$P \cdot N = 0 \qquad \text{on} \;\; \Gamma_t, \tag{5.48}$$

$$\Lambda = h(\lambda) \quad \text{on} \;\; \Gamma_s, \tag{5.49}$$

where N is the normal vector to the boundary, Λ is the stretch ratio (see definition in [204]) and $h(\lambda) = \frac{1}{1+e^{-\lambda}} + \frac{1}{2}$ forces the range of Λ to $(0.5, 1.5)$ in order to avoid extreme and unrealistic deformations.

Chapter 6

Concluding Remarks

The dissertation studied several different kinds of mechanical metamaterials with a computational approach. We focused on both modeling and design aspects of these metamaterials, and successfully used machine learning techniques to achieve the goals of both forward simulations and inverse designs.

To provide efficient solutions to forward simulation problems, we first introduced a neural network based computational homogenization scheme to accelerate the forward simulation of cellular mechanical metamaterials. We used a neural network to approximate the effective strain energy density as a function of cellular geometry and overall deformation. The network was constructed by “learning” from the data generated by finite element calculation of a set of representative volume elements at cellular scales. This effective strain energy density was then used to predict the mechanical responses of cellular materials at larger scales. Compared with direct numerical simulation, the proposed scheme could greatly reduce the computational cost.

We studied the same cellular mechanical metamaterials, concentrating on the design aspects. An efficient and robust shape optimization method was introduced to compute optimal geometric pore structures. The proposed method adopted a fixed referential configuration, so that shape optimization was accomplished without actually modifying the shape of computational domain. Three important numerical examples were used to demonstrate the advantages of the method, namely, optimizing negative Poisson's ratio, precise control of the onset of instability, and arbitrary tuning of band gaps for phononic structures. These examples cover a broad range of important engineering applications of cellular mechanical metamaterials.

Besides cellular mechanical metamaterials, we proposed an optimization framework that automates the design flow of composite mechanical metamaterials. Using generative models in machine learning, we successfully designed metamaterials with arbitrarily tunable macroscopic elastic moduli. The optimal samples were fabricated with advanced multi-material additive manufacturing techniques, and were tested by standard mechanical experiments for validation of the method.

Thinking beyond the design problems, we focused our attention on general PDE-constrained optimization problems. We proposed amortized finite element analysis (AmorFEA), in which a neural network learns to produce accurate PDE solutions. This network was trained to directly minimize the potential energy from which the PDE and finite element method are derived, avoiding the need to generate costly supervised training data by solving PDEs with traditional numerical methods. As FEA is a variational procedure, AmorFEA is a direct analogue to popular amortized inference approaches in latent variable models, with the finite element basis acting as the variational family. AmorFEA could perform PDE-CO without the need to repeatedly solve the associated PDE, accelerating optimization when compared to a traditional workflow using FEA and the adjoint method.

Finally, we point out some promising future directions for considerations. From the forward modeling perspective, other than simple feed forward neural networks used in this dissertation, we may adjust to more advanced network architectures for specific purposes of mechanical metamaterials. For example, the porous structure of cellular mechanical materials naturally forms a graph, which prompts the use of a graph-based neural network architecture for modeling. As for the inverse design part, we anticipate the role of automatic differentiation to be more significant in the optimization problems of mechanical metamaterials.

Bibliography

[1] X. Zheng, H. Lee, T. H. Weisgraber, M. Shusteff, J. DeOtte, E. B. Duoss, J. D. Kuntz, M. M. Biener, Q. Ge, J. A. Jackson, *et al.*, "Ultralight, ultrastiff mechanical metamaterials," *Science*, vol. 344, no. 6190, pp. 1373–1377, 2014.

[2] K. Bertoldi, P. M. Reis, S. Willshaw, and T. Mullin, "Negative poisson's ratio behavior induced by an elastic instability," *Advanced materials*, vol. 22, no. 3, pp. 361–366, 2010.

[3] M. Mirzaali, S. Janbaz, M. Strano, L. Vergani, and A. A. Zadpoor, "Shape-matching soft mechanical metamaterials," *Scientific reports*, vol. 8, no. 1, p. 965, 2018.

[4] D. Krishnan and H. T. Johnson, "Optical properties of two-dimensional polymer photonic crystals after deformation-induced pattern transformations," *Journal of the Mechanics and Physics of Solids*, vol. 57, no. 9, pp. 1500–1513, 2009.

[5] L. R. Meza, S. Das, and J. R. Greer, "Strong, lightweight, and recoverable three-dimensional ceramic nanolattices," *Science*, vol. 345, no. 6202, pp. 1322–1326, 2014.

[6] X. Li and H. Gao, "Mechanical metamaterials: Smaller and stronger," *Nature Materials*, vol. 15, no. 4, p. 373, 2016.

[7] B. Florijn, C. Coulais, and M. van Hecke, "Programmable mechanical metamaterials: the role of geometry," *Soft Matter*, vol. 12, no. 42, pp. 8736–8743, 2016.

[8] K. Bertoldi, "Harnessing instabilities to design tunable architected cellular materials," *Annual Review of Materials Research*, vol. 47, pp. 51–61, 2017.

[9] J. U. Surjadi, L. Gao, H. Du, X. Li, X. Xiong, N. X. Fang, and Y. Lu, "Mechanical metamaterials and their engineering applications," *Advanced Engineering Materials*, vol. 21, no. 3, p. 1800864, 2019.

[10] K. Bertoldi, V. Vitelli, J. Christensen, and M. Van Hecke, "Flexible mechanical metamaterials," *Nature Reviews Materials*, vol. 2, no. 11, pp. 1–11, 2017.

[11] X. Yu, J. Zhou, H. Liang, Z. Jiang, and L. Wu, “Mechanical metamaterials associated with stiffness, rigidity and compressibility: A brief review,” *Progress in Materials Science*, vol. 94, pp. 114–173, 2018.

[12] E. Barchiesi, M. Spagnuolo, and L. Placidi, “Mechanical metamaterials: a state of the art,” *Mathematics and Mechanics of Solids*, vol. 24, no. 1, pp. 212–234, 2019.

[13] S. Yuan, C. K. Chua, and K. Zhou, “3d-printed mechanical metamaterials with high energy absorption,” *Advanced Materials Technologies*, vol. 4, no. 3, p. 1800419, 2019.

[14] D. Chen and X. Zheng, “Multi-material additive manufacturing of metamaterials with giant, tailorable negative poisson’s ratios,” *Scientific reports*, vol. 8, no. 1, p. 9139, 2018.

[15] X. Ren, J. Shen, P. Tran, T. D. Ngo, and Y. M. Xie, “Design and characterisation of a tuneable 3d buckling-induced auxetic metamaterial,” *Materials & Design*, vol. 139, pp. 336–342, 2018.

[16] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis.* Courier Corporation, 2012.

[17] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary, *Nonlinear finite elements for continua and structures.* John wiley & sons, 2013.

[18] M. M. Ameen, O. Rokoš, R. Peerlings, and M. Geers, “Size effects in nonlinear periodic materials exhibiting reversible pattern transformations,” *Mechanics of Materials*, vol. 124, pp. 55–70, 2018.

[19] T. Mullin, S. Deschanel, K. Bertoldi, and M. C. Boyce, “Pattern transformation triggered by deformation,” *Physical review letters*, vol. 99, no. 8, p. 084301, 2007.

[20] J. T. Overvelde, S. Shan, and K. Bertoldi, “Compaction through buckling in 2d periodic, soft and porous structures: effect of pore shape,” *Advanced Materials*, vol. 24, no. 17, pp. 2337–2342, 2012.

[21] W. E, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, “Heterogeneous multiscale methods: a review,” *Communications in computational physics*, vol. 2, no. 3, pp. 367–450, 2007.

[22] S. Ahuja, V. Yakhot, and I. G. Kevrekidis, “Computational coarse graining of a randomly forced one-dimensional burgers equation,” *Physics of Fluids*, vol. 20, no. 3, p. 035111, 2008.

[23] J. D. Eshelby, “The determination of the elastic field of an ellipsoidal inclusion, and related problems,” *Proceedings of the royal society of London A*, vol. 241, no. 1226, pp. 376–396, 1957.

[24] E. Kröner, “Berechnung der elastischen konstanten des vielkristalls aus den konstanten des einkristalls,” *Zeitschrift für Physik*, vol. 151, no. 4, pp. 504–518, 1958.

[25] R. Hill, “Continuum micro-mechanics of elastoplastic polycrystals,” *Journal of the Mechanics and Physics of Solids*, vol. 13, no. 2, pp. 89–101, 1965.

[26] M. Berveiller and A. Zaoui, “An extension of the self-consistent scheme to plastically-flowing polycrystals,” *Journal of the Mechanics and Physics of Solids*, vol. 26, no. 5-6, pp. 325–344, 1978.

[27] F. Feyel, “Multiscale fe2 elastoviscoplastic analysis of composite structures,” *Computational Materials Science*, vol. 16, no. 1-4, pp. 344–354, 1999.

[28] A. Gendy and A. Saleeb, “Nonlinear material parameter estimation for characterizing hyper elastic large strain models,” *Computational mechanics*, vol. 25, no. 1, pp. 66–77, 2000.

[29] J. Yvonnet, E. Monteiro, and Q.-C. He, “Computational homogenization method and reduced database model for hyperelastic heterogeneous structures,” *International Journal for Multiscale Computational Engineering*, vol. 11, no. 3, pp. 201–225, 2013.

[30] J. Yvonnet, D. Gonzalez, and Q.-C. He, “Numerically explicit potentials for the homogenization of nonlinear elastic heterogeneous materials,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 33-36, pp. 2723–2737, 2009.

[31] N. Takano, M. Zako, and Y. Ohnishi, “Macro-micro uncoupled homogenization procedure for microscopic nonlinear behavior analysis of composites,” *Journal of the Society of Materials Science, Japan*, vol. 45, no. 6Appendix, pp. 81–86, 1996.

[32] D. Ryckelynck, “A priori hyperreduction method: an adaptive approach,” *Journal of Computational Physics*, vol. 202, no. 1, pp. 346–366, 2005.

[33] I. Temizer and T. Zohdi, “A numerical method for homogenization in non-linear elasticity,” *Computational Mechanics*, vol. 40, no. 2, pp. 281–298, 2007.

[34] R. Chowdhury, B. Rao, and A. M. Prasad, “High-dimensional model representation for structural reliability analysis,” *Communications in Numerical Methods in Engineering*, vol. 25, no. 4, pp. 301–337, 2009.

[35] F. Fritzen and O. Kunc, “Two-stage data-driven homogenization for nonlinear solids using a reduced order model,” *European Journal of Mechanics-A/Solids*, vol. 69, pp. 201–220, 2018.

[36] B. Le, J. Yvonnet, and Q.-C. He, “Computational homogenization of nonlinear elastic materials using neural networks,” *International Journal for Numerical Methods in Engineering*, vol. 104, no. 12, pp. 1061–1084, 2015.

[37] X. Lu, D. G. Giovanis, J. Yvonnet, V. Papadopoulos, F. Detrez, and J. Bai, “A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites,” *Computational Mechanics*, vol. 64, no. 2, pp. 307–321, 2019.

[38] J. Carrasquilla and R. G. Melko, “Machine learning phases of matter,” *Nature Physics*, vol. 13, no. 5, pp. 431–434, 2017.

[39] G. Carleo and M. Troyer, “Solving the quantum many-body problem with artificial neural networks,” *Science*, vol. 355, no. 6325, pp. 602–606, 2017.

[40] D.-L. Deng, “Machine learning detection of bell nonlocality in quantum many-body systems,” *Physical review letters*, vol. 120, no. 24, p. 240402, 2018.

[41] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, “Automatic chemical design using a data-driven continuous representation of molecules,” *ACS central science*, vol. 4, no. 2, pp. 268–276, 2018.

[42] M. C. Chan and J. P. Stott, “Deep-cee i: fishing for galaxy clusters with deep neural nets,” *Monthly Notices of the Royal Astronomical Society*, vol. 490, no. 4, pp. 5770–5787, 2019.

[43] Z. Nie, H. Jiang, and L. B. Kara, “Deep learning for stress field prediction using convolutional neural networks,” *arxiv: 1808.08914*, 2018.

[44] K. Sagiyama and K. Garikipati, “Machine learning materials physics: Deep neural networks trained on elastic free energy data from martensitic microstructures predict homogenized stress fields with high accuracy,” *arxiv: 1901.00524*, 2019.

[45] H. Zhao, B. D. Storey, R. D. Braatz, and M. Z. Bazant, “Learning the physics of pattern formation from images,” *Physical Review Letters*, vol. 124, no. 6, p. 060201, 2020.

[46] M. P. Bendsøe and N. Kikuchi, “Generating optimal topologies in structural design using a homogenization method,” *Computer methods in applied mechanics and engineering*, vol. 71, no. 2, pp. 197–224, 1988.

[47] M. P. Bendsøe and O. Sigmund, “Material interpolation schemes in topology optimization,” *Archive of applied mechanics*, vol. 69, no. 9, pp. 635–654, 1999.

[48] Y. M. Xie and G. P. Steven, “A simple evolutionary procedure for structural optimization,” *Computers & structures*, vol. 49, no. 5, pp. 885–896, 1993.

[49] J. A. Sethian and A. Wiegmann, "Structural boundary design via level set and immersed interface methods," *Journal of computational physics*, vol. 163, no. 2, pp. 489–528, 2000.

[50] S. J. Osher and F. Santosa, "Level set methods for optimization problems involving geometry and constraints: I. frequencies of a two-density inhomogeneous drum," *Journal of Computational Physics*, vol. 171, no. 1, pp. 272–288, 2001.

[51] M. Y. Wang, X. Wang, and D. Guo, "A level set method for structural topology optimization," *Computer methods in applied mechanics and engineering*, vol. 192, no. 1-2, pp. 227–246, 2003.

[52] F. Wang, O. Sigmund, and J. S. Jensen, "Design of materials with prescribed nonlinear properties," *Journal of the Mechanics and Physics of Solids*, vol. 69, pp. 156–174, 2014.

[53] A. Clausen, F. Wang, J. S. Jensen, O. Sigmund, and J. A. Lewis, "Topology optimized architectures with programmable poisson's ratio over large deformations," *Adv. Mater*, vol. 27, no. 37, pp. 5523–5527, 2015.

[54] G. Zhang and K. Khandelwal, "Computational design of finite strain auxetic metamaterials via topology optimization and nonlinear homogenization," *Computer Methods in Applied Mechanics and Engineering*, vol. 356, pp. 490–527, 2019.

[55] R. M. Errico, "What is an adjoint model?," *Bulletin of the American Meteorological Society*, vol. 78, no. 11, pp. 2577–2592, 1997.

[56] Y. Cao, S. Li, L. Petzold, and R. Serban, "Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution," *SIAM Journal on Scientific Computing*, vol. 24, no. 3, pp. 1076–1089, 2003.

[57] J. Donea, S. Giuliani, and J.-P. Halleux, "An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions," *Computer methods in applied mechanics and engineering*, vol. 33, no. 1-3, pp. 689–723, 1982.

[58] A. Diaz and O. Sigmund, "Checkerboard patterns in layout optimization," *Structural optimization*, vol. 10, no. 1, pp. 40–45, 1995.

[59] G. Allaire, C. Dapogny, and F. Jouve, "Shape and topology optimization," *Geometric Partial Differential Equations, part II, A. Bonito and R. Nochetto eds., Handbook of Numerical Analysis*, vol. 22, pp. 1–132, 2020.

[60] G. Allaire, F. Jouve, and A.-M. Toader, "A level-set method for shape optimization," *Comptes Rendus Mathematique*, vol. 334, no. 12, pp. 1125–1130, 2002.

[61] G. Allaire, C. Dapogny, and P. Frey, "Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh," *Comptes Rendus Mathematique*, vol. 349, no. 17-18, pp. 999–1003, 2011.

[62] G. Allaire, C. Dapogny, and P. Frey, "Shape optimization with a level set based mesh evolution method," *Computer Methods in Applied Mechanics and Engineering*, vol. 282, pp. 22–53, 2014.

[63] H. Tao, T. Wu, M. Aldeghi, T. C. Wu, A. Aspuru-Guzik, and E. Kumacheva, "Nanoparticle synthesis assisted by machine learning," *Nature Reviews Materials*, pp. 1–16, 2021.

[64] B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. M. Alvarado, J. M. Janey, R. P. Adams, and A. G. Doyle, "Bayesian reaction optimization as a tool for chemical synthesis," *Nature*, vol. 590, no. 7844, pp. 89–96, 2021.

[65] X. Sun, T. Xue, S. M. Rusinkiewicz, and R. P. Adams, "Amortized synthesis of constrained configurations using a differentiable surrogate," *arXiv preprint arXiv:2106.09019*, 2021.

[66] L. Kharevych, P. Mullen, H. Owhadi, and M. Desbrun, "Numerical coarsening of inhomogeneous elastic materials," *ACM Transactions on graphics (TOG)*, vol. 28, no. 3, pp. 1–8, 2009.

[67] B. Zhu, M. Skouras, D. Chen, and W. Matusik, "Two-scale topology optimization with microstructures," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 1, 2017.

[68] C. Schumacher, B. Bickel, J. Rys, S. Marschner, C. Daraio, and M. Gross, "Microstructures to control elasticity in 3d printing," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–13, 2015.

[69] S. Hoyer, J. Sohl-Dickstein, and S. Greydanus, "Neural reparameterization improves structural optimization," *arXiv preprint arXiv:1909.04240*, 2019.

[70] K.-H. Chang and C.-Y. Cheng, "Learning to simulate and design for structural engineering," in *International Conference on Machine Learning*, pp. 1426–1436, PMLR, 2020.

[71] H. Sun, H. V. Burton, and H. Huang, "Machine learning applications for building structural design and performance assessment: state-of-the-art review," *Journal of Building Engineering*, p. 101816, 2020.

[72] S. Eismann, S. Bartzsch, and S. Ermon, "Shape optimization in laminar flow with a label-guided variational autoencoder," *arXiv preprint arXiv:1712.03599*, 2017.

[73] Z. Yang, X. Li, L. Catherine Brinson, A. N. Choudhary, W. Chen, and A. Agrawal, "Microstructural materials design via deep adversarial learning methodology," *Journal of Mechanical Design*, vol. 140, no. 11, 2018.

[74] T. Guo, D. J. Lohan, R. Cang, M. Y. Ren, and J. T. Allison, "An indirect design representation for topology optimization using variational autoencoder and style transfer," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 0804, 2018.

[75] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[76] J. Močkus, "On bayesian methods for seeking the extremum," in *Optimization techniques IFIP technical conference*, pp. 400–404, Springer, 1975.

[77] T. J. Wallin, L. Simonsen, W. Pan, K. Wang, E. Giannelis, R. F. Shepherd, and Y. Menguc, "3d printable tough silicone double networks," *Nature Communications*, 2020.

[78] T. Rees, H. S. Dollar, and A. J. Wathen, "Optimal solvers for PDE-constrained optimization," *SIAM Journal on Scientific Computing*, vol. 32, no. 1, pp. 271–298, 2010.

[79] M. Ulbrich and B. van Bloemen Waanders, "An introduction to partial differential equations constrained optimization," *Optimization and Engineering*, vol. 19, no. 3, pp. 515–520, 2018.

[80] E. Weinan and B. Yu, "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems," *Communications in Mathematics and Statistics*, vol. 6, no. 1, pp. 1–12, 2018.

[81] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *arXiv preprint arXiv:1907.04502*, 2019.

[82] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[83] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris, "Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data," *Journal of Computational Physics*, vol. 394, pp. 56–81, 2019.

[84] S. Gershman and N. Goodman, "Amortized inference in probabilistic reasoning," in *Proceedings of the annual meeting of the cognitive science society*, vol. 36, 2014.

[85] S. Ravi and A. Beatson, "Amortized Bayesian meta-learning," 2018.

[86] K. Choi, M. Wu, N. Goodman, and S. Ermon, "Meta-amortized variational inference and learning," *arXiv preprint arXiv:1902.01950*, 2019.

[87] T. Xue, A. Beatson, M. Chiaramonte, G. Roeder, J. T. Ash, Y. Menguc, S. Adriaenssens, R. P. Adams, and S. Mao, "A data-driven computational scheme for the nonlinear mechanical properties of cellular mechanical metamaterials under large deformation," *Soft Matter*, 2020.

[88] T. Xue, T. J. Wallin, Y. Menguc, S. Adriaenssens, and M. Chiaramonte, "Machine learning generative models for automatic design of multi-material 3d printed composite solids," *Extreme Mechanics Letters*, vol. 41, p. 100992, 2020.

[89] T. Xue, A. Beatson, S. Adriaenssens, and R. Adams, "Amortized finite element analysis for fast pde-constrained optimization," in *International Conference on Machine Learning*, pp. 10638–10647, PMLR, 2020.

[90] L. J. Gibson and M. F. Ashby, *Cellular solids: structure and properties.* Cambridge university press, 1999.

[91] T. A. Schaedler and W. B. Carter, "Architected cellular materials," *Annual Review of Materials Research*, vol. 46, pp. 187–210, 2016.

[92] J. T. Overvelde and K. Bertoldi, "Relating pore shape to the non-linear response of periodic elastomeric structures," *Journal of the Mechanics and Physics of Solids*, vol. 64, pp. 351–366, 2014.

[93] Z. Hashin and S. Shtrikman, "A variational approach to the theory of the elastic behaviour of multiphase materials," *Journal of the Mechanics and Physics of Solids*, vol. 11, no. 2, pp. 127–140, 1963.

[94] B. Budiansky, "On the elastic moduli of some heterogeneous materials," *Journal of the Mechanics and Physics of Solids*, vol. 13, no. 4, pp. 223–227, 1965.

[95] R. Hill, "A self-consistent mechanics of composite materials," *Journal of the Mechanics and Physics of Solids*, vol. 13, no. 4, pp. 213–222, 1965.

[96] T. Mori and K. Tanaka, "Average stress in matrix and average elastic energy of materials with misfitting inclusions," *Acta metallurgica*, vol. 21, no. 5, pp. 571–574, 1973.

[97] J. Willis, "Bounds and self-consistent estimates for the overall properties of anisotropic composites," *Journal of the Mechanics and Physics of Solids*, vol. 25, no. 3, pp. 185–202, 1977.

[98] P. P. Castañeda, "The effective mechanical properties of nonlinear isotropic composites," *Journal of the Mechanics and Physics of Solids*, vol. 39, no. 1, pp. 45–71, 1991.

[99] P. P. Castañeda, "Exact second-order estimates for the effective mechanical properties of nonlinear composite materials," *Journal of the Mechanics and Physics of Solids*, vol. 44, no. 6, pp. 827–862, 1996.

[100] J. Segurado, R. A. Lebensohn, and J. LLorca, "Computational homogenization of polycrystals," in *Advances in Applied Mechanics*, vol. 51, pp. 1–114, Elsevier, 2018.

[101] M. G. Geers, V. G. Kouznetsova, and W. Brekelmans, "Multi-scale computational homogenization: Trends and challenges," *Journal of computational and applied mathematics*, vol. 234, no. 7, pp. 2175–2182, 2010.

[102] V. P. Nguyen, M. Stroeven, and L. J. Sluys, "Multiscale continuous and discontinuous modeling of heterogeneous materials: a review on recent developments," *Journal of Multiscale Modelling*, vol. 3, no. 04, pp. 229–270, 2011.

[103] S. Saeb, P. Steinmann, and A. Javili, "Aspects of computational homogenization at finite deformations: a unifying review from reuss' to voigt's bound," *Applied Mechanics Reviews*, vol. 68, no. 5, p. 050801, 2016.

[104] F. Feyel and J.-L. Chaboche, "Fe2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 183, no. 3-4, pp. 309–330, 2000.

[105] S. Ghosh, K. Lee, and P. Raghavan, "A multi-level computational model for multi-scale damage analysis in composite and porous materials," *International Journal of Solids and Structures*, vol. 38, no. 14, pp. 2335–2385, 2001.

[106] K. Terada and N. Kikuchi, "A class of general algorithms for multi-scale analyses of heterogeneous media," *Computer methods in applied mechanics and engineering*, vol. 190, no. 40-41, pp. 5427–5464, 2001.

[107] V. Kouznetsova, M. G. Geers, and W. Brekelmans, "Multi-scale second-order computational homogenization of multi-phase materials: a nested finite element solution strategy," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 48-51, pp. 5525–5550, 2004.

[108] J. Yvonnet and Q.-C. He, "The reduced model multiscale method (r3m) for the non-linear homogenization of hyperelastic media at finite strains," *Journal of Computational Physics*, vol. 223, no. 1, pp. 341–368, 2007.

[109] H. Moulinec and P. Suquet, "A fast numerical method for computing the linear and nonlinear mechanical properties of composites," *Comptes rendus de l'Académie des sciences. Série II, Mécanique, physique, chimie, astronomie*, vol. 318, no. 11, pp. 1417–1423, 1994.

[110] H. Moulinec and P. Suquet, "A numerical method for computing the overall response of nonlinear composites with complex microstructure," *Computer methods in applied mechanics and engineering*, vol. 157, no. 1-2, pp. 69–94, 1998.

[111] D. J. Eyre and G. W. Milton, "A fast numerical scheme for computing the response of composites using grid refinement," *The European Physical Journal-Applied Physics*, vol. 6, no. 1, pp. 41–47, 1999.

[112] J. Michel, H. Moulinec, and P. Suquet, "A computational method based on augmented lagrangians and fast fourier transforms for composites with high contrast," *CMES(Computer Modelling in Engineering & Sciences)*, vol. 1, no. 2, pp. 79–88, 2000.

[113] S. Haykin, "Neural networks: A comprehensive foundation, prentice hall ptr," *Upper Saddle River, NJ, USA*, 1998.

[114] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *ICML*, 2011.

[115] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[116] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[117] M. Wainberg, D. Merico, A. Delong, and B. J. Frey, "Deep learning in biomedicine," *Nature biotechnology*, vol. 36, no. 9, pp. 829–838, 2018.

[118] R. Hill, "On constitutive macro-variables for heterogeneous solids at finite strain," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 326, no. 1565, pp. 131–147, 1972.

[119] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

[120] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[121] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[122] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[123] A. Logg, K.-A. Mardal, and G. Wells, *Automated solution of differential equations by the finite element method: The FEniCS book*, vol. 84. Springer Science & Business Media, 2012.

[124] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[125] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arxiv: 1412.6980*, 2014.

[126] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974.

[127] C. M. Bishop, *Pattern recognition and machine learning.* springer, 2006.

[128] M. S. Alnæs, "Ufl: a finite element form language," in *Automated Solution of Differential Equations by the Finite Element Method*, pp. 303–338, Springer, 2012.

[129] S. Sarkar, M. Cebron, M. Brojan, and A. Kosmrlj, "Image charges for deformed 2d solid structures with circular holes and inclusions," *arXiv: 2004.01044*, 2020.

[130] G. Allaire, F. De Gournay, F. Jouve, and A.-M. Toader, "Structural optimization using topological and shape sensitivity via a level set method," *Control and cybernetics*, vol. 34, no. 1, p. 59, 2005.

[131] H. A. Eschenauer and N. Olhoff, "Topology optimization of continuum structures: a review," *Applied Mechanics Reviews*, vol. 54, no. 4, pp. 331–390, 2001.

[132] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees.* CRC press, 1984.

[133] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63–71, Springer, 2003.

[134] V. Schulz and M. Siebenborn, "Computational comparison of surface metrics for pde constrained shape optimization," *Computational Methods in Applied Mathematics*, vol. 16, no. 3, pp. 485–496, 2016.

[135] J. Nitsche, "Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind," in *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, vol. 36, pp. 9–15, Springer, 1971.

[136] P. Hansbo, "Nitsche's method for interface problems in computational mechanics," *GAMM-Mitteilungen*, vol. 28, no. 2, pp. 183–206, 2005.

[137] T. Belytschko and T. J. Hughes, “Computational methods for transient analysis,” *Amsterdam, North-Holland(Computational Methods in Mechanics.*, vol. 1, 1983.

[138] E. Riks, “An incremental approach to the solution of snapping and buckling problems,” *International journal of solids and structures*, vol. 15, no. 7, pp. 529–551, 1979.

[139] D. J. Luet, *Bounding volume hierarchy and non-uniform rational B-splines for contact enforcement in large deformation finite element analysis of sheet metal forming.* PhD thesis, Princeton University, 2016.

[140] S. K. Mitusch, S. W. Funke, and J. S. Dokken, “dolfin-adjoint 2018.1: automated adjoints for fenics and firedrake,” *Journal of Open Source Software*, vol. 4, no. 38, p. 1292, 2019.

[141] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal on scientific computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[142] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on mathematical software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[143] V. Hernandez, J. E. Roman, and V. Vidal, “Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 351–362, 2005.

[144] P. Lancaster, “On eigenvalues of matrices dependent on a parameter,” *Numerische Mathematik*, vol. 6, no. 1, pp. 377–387, 1964.

[145] M. S. Kushwaha, P. Halevi, L. Dobrzynski, and B. Djafari-Rouhani, “Acoustic band structure of periodic elastic composites,” *Physical review letters*, vol. 71, no. 13, p. 2022, 1993.

[146] B. Merheb, P. A. Deymier, M. Jain, M. Aloshyna-Lesuffleur, S. Mohanty, A. Berker, and R. Greger, “Elastic and viscoelastic effects in rubber/air acoustic band gap structures: A theoretical and experimental study,” *Journal of Applied Physics*, vol. 104, no. 6, p. 064913, 2008.

[147] M. Ruzzene and F. Scarpa, “Directional and band-gap behavior of periodic auxetic lattices,” *physica status solidi (b)*, vol. 242, no. 3, pp. 665–680, 2005.

[148] W. Cheng, J. Wang, U. Jonas, G. Fytas, and N. Stefanou, “Observation and tuning of hypersonic bandgaps in colloidal crystals,” *Nature materials*, vol. 5, no. 10, pp. 830–836, 2006.

[149] F. Casadei, L. Dozio, M. Ruzzene, and K. A. Cunefare, "Periodic shunted arrays for the control of noise radiation in an enclosure," *Journal of sound and vibration*, vol. 329, no. 18, pp. 3632–3646, 2010.

[150] F. Casadei, B. S. Beck, K. A. Cunefare, and M. Ruzzene, "Vibration control of plates through hybrid configurations of periodic piezoelectric shunts," *Journal of Intelligent Material Systems and Structures*, vol. 23, no. 10, pp. 1169–1177, 2012.

[151] F. Javid, P. Wang, A. Shanian, and K. Bertoldi, "Architected materials with ultra-low porosity for vibration control," *Advanced materials*, vol. 28, no. 28, pp. 5943–5948, 2016.

[152] P. Wang, J. Shim, and K. Bertoldi, "Effects of geometric and material nonlinearities on tunable band gaps and low-frequency directionality of phononic crystals," *Physical Review B*, vol. 88, no. 1, p. 014304, 2013.

[153] K. Bertoldi and M. C. Boyce, "Wave propagation and instabilities in monolithic and periodically structured elastomeric materials undergoing large deformations," *Physical Review B*, vol. 78, no. 18, p. 184107, 2008.

[154] G. Geymonat, S. Müller, and N. Triantafyllidis, "Homogenization of nonlinearly elastic materials, microscopic bifurcation and macroscopic loss of rank-one convexity," *Archive for rational mechanics and analysis*, vol. 122, no. 3, pp. 231–290, 1993.

[155] N. Triantafyllidis, M. Nestorović, and M. Schraad, "Failure surfaces for finitely strained two-phase periodic solids under general in-plane loading," 2006.

[156] K. Bertoldi, M. C. Boyce, S. Deschanel, S. Prange, and T. Mullin, "Mechanics of deformation-triggered pattern transformations and superelastic behavior in periodic elastomeric structures," *Journal of the Mechanics and Physics of Solids*, vol. 56, no. 8, pp. 2642–2668, 2008.

[157] B. M. Goldsberry and M. R. Haberman, "Negative stiffness honeycombs as tunable elastic metamaterials," *Journal of applied physics*, vol. 123, no. 9, p. 091711, 2018.

[158] L. Brillouin, *Wave propagation in periodic structures: electric filters and crystal lattices*, vol. 2. Dover publications, 1953.

[159] P. Blanchard, D. J. Higham, and N. J. Higham, "Accurately computing the log-sum-exp and softmax functions," 2019.

[160] M. Kadic, T. Bückmann, N. Stenger, M. Thiel, and M. Wegener, "On the practicability of pentamode mechanical metamaterials," *Applied Physics Letters*, vol. 100, no. 19, p. 191901, 2012.

[161] P. Sitthi-Amorn, J. E. Ramos, Y. Wangy, J. Kwan, J. Lan, W. Wang, and W. Matusik, "Multifab: a machine vision assisted platform for multi-material 3d printing," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–11, 2015.

[162] P. Vogiatzis, S. Chen, X. Wang, T. Li, and L. Wang, "Topology optimization of multi-material negative poisson's ratio metamaterials using a reconciled level set method," *Computer-Aided Design*, vol. 83, pp. 15–32, 2017.

[163] M. Mirzaali, A. Caracciolo, H. Pahlavani, S. Janbaz, L. Vergani, and A. Zadpoor, "Multi-material 3d printed mechanical metamaterials: Rational design of elastic properties through spatial distribution of hard and soft phases," *Applied Physics Letters*, vol. 113, no. 24, p. 241903, 2018.

[164] M. Mirzaali, R. Hedayati, P. Vena, L. Vergani, M. Strano, and A. Zadpoor, "Rational design of soft mechanical metamaterials: Independent tailoring of elastic properties with randomness," *Applied Physics Letters*, vol. 111, no. 5, p. 051903, 2017.

[165] A. Beatson, J. T. Ash, G. Roeder, T. Xie, and R. P. Adams, "Learning composable energy surrogates for pde order reduction," *arXiv preprint arXiv:2005.06549*, 2020.

[166] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[167] T. Wallin, J. Pikul, S. Bodkhe, B. Peele, B. Mac Murray, D. Therriault, B. McEnerney, R. Dillon, E. Giannelis, and R. Shepherd, "Click chemistry stereolithography for soft robots that self-heal," *Journal of Materials Chemistry B*, vol. 5, no. 31, pp. 6249–6255, 2017.

[168] J. N. Reddy, *An introduction to continuum mechanics.* Cambridge university press, 2007.

[169] M. E. Gurtin, *An introduction to continuum mechanics.* Academic press, 1982.

[170] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, "The fenics project version 1.5," *Archive of Numerical Software*, vol. 3, no. 100, 2015.

[171] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[172] J. D. Hyman and C. L. Winter, "Stochastic generation of explicit pore structures by thresholding gaussian random fields," *Journal of Computational Physics*, vol. 277, pp. 16–31, 2014.

[173] Z. Jiang, W. Chen, and C. Burkhart, "Efficient 3d porous microstructure reconstruction via gaussian random field and hybrid optimization," *Journal of microscopy*, vol. 252, no. 2, pp. 135–148, 2013.

[174] I. T. Jolliffe, "Principal components in regression analysis," in *Principal component analysis*, pp. 129–155, Springer, 1986.

[175] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[176] D. Drieß, P. Englert, and M. Toussaint, "Constrained bayesian optimization of combined interaction force/task space controllers for manipulations," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 902–907, IEEE, 2017.

[177] E. Bakshy, L. Dworkin, B. Karrer, K. Kashin, B. Letham, A. Murthy, and S. Singh, "Ae: A domain-agnostic platform for adaptive experimentation," 2018.

[178] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in neural information processing systems*, pp. 3483–3491, 2015.

[179] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[180] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[181] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE transactions on neural networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[182] R. Lopez, E. Balsa-Canto, and E. Oñate, "Neural networks for variational problems in engineering," *International Journal for Numerical Methods in Engineering*, vol. 75, no. 11, pp. 1341–1360, 2008.

[183] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, pp. 1339–1364, 2018.

[184] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, vol. 98. SIAM, 2007.

[185] R. Herzog and K. Kunisch, "Algorithms for PDE-constrained optimization," *GAMM-Mitteilungen*, vol. 33, no. 2, pp. 163–176, 2010.

[186] J. Dennis, M. Heinkenschloss, and L. N. Vicente, "Trust-region interior-point SQP algorithms for a class of nonlinear programming problems," *SIAM Journal on Control and Optimization*, vol. 36, no. 5, pp. 1750–1794, 1998.

[187] T. van Leeuwen and F. J. Herrmann, "A penalty method for PDE-constrained optimization in inverse problems," *Inverse Problems*, vol. 32, no. 1, p. 015007, 2015.

[188] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for optimization*. MIT Press, 2019.

[189] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning," *Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114*, vol. 5, no. 6, p. 12, 2011.

[190] A. Shah, A. Wilson, and Z. Ghahramani, "Student-t processes as alternatives to Gaussian processes," in *International Conference on Artificial Intelligence and Statistics*, pp. 877–885, 2014.

[191] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, "Scalable Bayesian optimization using deep neural networks," in *International Conference on Machine Learning*, pp. 2171–2180, 2015.

[192] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[193] C. Cremer, X. Li, and D. Duvenaud, "Inference suboptimality in variational autoencoders," *arXiv preprint arXiv:1801.03558*, 2018.

[194] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

[195] C. S. De Melo, M. Randeria, and J. R. Engelbrecht, "Crossover from BCS to Bose superconductivity: Transition temperature and time-dependent Ginzburg-Landau theory," *Physical Review Letters*, vol. 71, no. 19, p. 3202, 1993.

[196] A. Logg and G. N. Wells, "DOLFIN: Automated finite element computing," *ACM Transactions on Mathematical Software (TOMS)*, vol. 37, no. 2, pp. 1–28, 2010.

[197] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in neural information processing systems*, pp. 971–980, 2017.

[198] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," 1994.

[199] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, "SciPy 1.0–fundamental algorithms for scientific computing in Python," *arXiv preprint arXiv:1907.10121*, 2019.

[200] M. Runciman, A. Darzi, and G. P. Mylonas, "Soft robotics in minimally invasive surgery," *Soft robotics*, vol. 6, no. 4, pp. 423–443, 2019.

[201] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[202] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, vol. 15. Springer Science & Business Media, 2012.

[203] R. W. Ogden, *Non-linear elastic deformations.* Courier Corporation, 1997.

[204] A. G. Holzapfel, "Nonlinear solid mechanics ii," 2000.