

# Model order reduction techniques in two-scale solid mechanics

***Citation for published version (APA):***

Guo, T. (2024). *Model order reduction techniques in two-scale solid mechanics*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Eindhoven University of Technology.

***Document status and date:***

Published: 21/02/2024

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

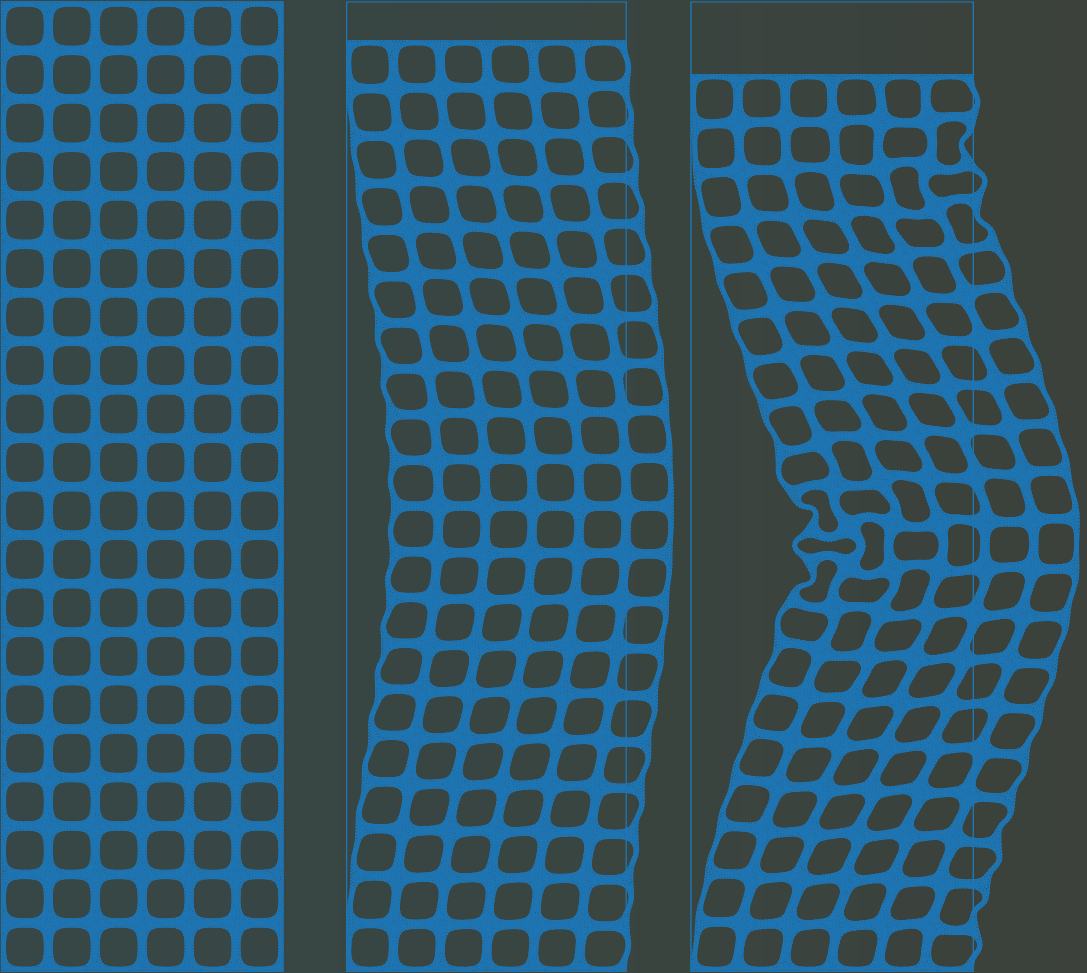
***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Model order reduction techniques in two-scale solid mechanics



**Theron Guo**

Model order reduction techniques in two-scale solid  
mechanics

Theron Guo

Theron Guo

*Model order reduction techniques in two-scale solid mechanics*

Eindhoven University of Technology, 2024

The research described in this thesis was performed at the Centre for Analysis, Scientific Computing and Applications (CASA) within the Department of Mathematics and Computer Science at Eindhoven University of Technology, the Netherlands.

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement No. 818473).

Program website: <https://cordis.europa.eu/project/id/818473>.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-5949-7

Cover design: © Theron Guo.

Printing: Gildeprint – Enschede.

Copyright © 2024 by Theron Guo, The Netherlands.  
All rights are reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of the author.

# Model order reduction techniques in two-scale solid mechanics

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit  
Eindhoven, op gezag van de rector magnificus prof. dr. S. K. Lenaerts,  
voor een commissie aangewezen door het College voor Promoties, in het  
openbaar te verdedigen op woensdag 21 februari 2024 om 13:30 uur

door

Theron Guo

geboren te Berlijn, Duitsland

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof. dr. B. Koren
promotor:	prof. dr. K. Veroy-Grepl
copromotor:	dr. O. Rokoš
leden:	prof. dr. P. Benner (MPI Magdeburg)
	prof. dr. J.T.B. Overvelde (AMOLF)
	prof. dr. W.H.A. Schilders
	prof. dr. E. Cueto (UNIZAR)

*Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.*







# Abstract

In recent years, the design of macrostructures with (quasi-)periodic microstructures, such as composite materials or metamaterials, has attracted significant interest due to their programmable and exotic properties. Resolving these microstructures at the macroscale and directly simulating such a macrostructure is computationally expensive or even infeasible because of the complex geometry of the microstructures, which necessitate extremely fine simulation meshes. To tackle this issue, two-scale methods based on computational homogenization are often employed, in which the microstructure, defined on a representative volume element, and a homogenized macrostructure are separately defined, coupled, and concurrently solved.

Unfortunately, solving such two-scale models remains computationally demanding due to the repeated solution of the microstructure. In this thesis, we have developed several reduced-order models for the microscopic problems that arise in first- and second-order computational homogenization. Both non-intrusive and intrusive methods have been utilized to accelerate nonlinear microstructures. The derived surrogate models have been validated and assessed through several numerical experiments in two scales, and the surrogate results have been compared to the solutions obtained with the direct numerical solution and the full computational homogenization model. For all problems, we have achieved good agreement and significant computational speed-ups.

**Keywords:** Computational homogenization, parameterized partial differential equations, proper orthogonal decomposition, Gaussian process regression, surrogate modeling, hyperreduction, empirical cubature method, geometrical transformations.



# Contents

<b>Notation</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Numerical methods for multiscale problems . . . . .	2
1.1.1 Large scale separation . . . . .	3
1.1.2 Small scale separation . . . . .	4
1.2 Reduced order models for computational homogenization . . . .	6
1.2.1 Non-intrusive methods . . . . .	6
1.2.2 Intrusive methods . . . . .	8
1.3 Main contributions . . . . .	9
1.4 Outline . . . . .	10
<b>2 First-order computational homogenization</b>	<b>13</b>
2.1 Macroscopic problem . . . . .	14
2.2 Parameterized microscopic problem . . . . .	16
2.2.1 Effective quantities . . . . .	19
2.2.2 Scale coupling and boundary conditions . . . . .	21
2.2.3 Discretization and numerical solution . . . . .	22
2.2.4 Computational costs . . . . .	22
<b>3 A non-intrusive reduced order model for first-order compu-</b>	<b>25</b>
<b>tational homogenization</b>	
3.1 Surrogate modeling . . . . .	26
3.1.1 Transformation of snapshots . . . . .	27
3.1.2 Proper orthogonal decomposition . . . . .	31
3.1.3 Regression model . . . . .	33
3.1.4 Comparison to neural networks . . . . .	37
3.1.5 Offline–online decomposition . . . . .	38
3.2 Numerical examples for fixed geometry . . . . .	39

3.2.1	Porous material . . . . .	40
3.2.2	Fiber reinforced material . . . . .	44
3.2.3	Two-scale Cook's membrane problem . . . . .	48
3.3	Numerical examples for varying geometry . . . . .	52
3.3.1	Composite microstructure with an elliptical fiber . . . . .	53
3.3.2	Composite microstructure with a B-spline controlled inclusion shape . . . . .	57
3.3.3	Geometrically parameterized two-scale Cook's membrane problem . . . . .	59
3.4	Conclusions . . . . .	62
<b>4</b>	<b>An intrusive reduced order model for first-order computational homogenization</b>	<b>65</b>
4.1	Surrogate modelling . . . . .	66
4.1.1	Reduced basis method . . . . .	66
4.1.2	Empirical cubature method . . . . .	67
4.1.3	Effective quantities . . . . .	69
4.1.4	Summary . . . . .	71
4.2	Numerical examples . . . . .	72
4.2.1	Elasto-plastic composite RVE with random inclusions . . . . .	73
4.2.2	Two-scale compression with porous microstructure . . . . .	77
4.3	Conclusions . . . . .	83
<b>5</b>	<b>A reduced order model for second-order computational homogenization</b>	<b>85</b>
5.1	Second-order computational homogenization . . . . .	86
5.1.1	Macroscopic problem . . . . .	86
5.1.2	Parameterized microscopic problem . . . . .	89
5.1.3	Effective quantities . . . . .	92
5.2	Reduced order modeling . . . . .	95
5.2.1	Proper orthogonal decomposition . . . . .	95
5.2.2	Hyperreduction . . . . .	96
5.3	Numerical examples . . . . .	104
5.3.1	Uniaxial compression of perforated plate . . . . .	106
5.3.2	Biaxial compression of graded cruciform . . . . .	112
5.4	Conclusions . . . . .	118
<b>6</b>	<b>Discussion and outlook</b>	<b>119</b>

---

<b>A</b>	<b>Constitutive models</b>	<b>123</b>
A.1	Hyperelastic material models . . . . .	123
A.1.1	Neo-Hookean material . . . . .	123
A.1.2	Bertoldi-Boyce material model . . . . .	124
A.2	Plasticity model . . . . .	124
<b>B</b>	<b>Proper orthogonal decomposition</b>	<b>127</b>
<b>C</b>	<b>Proof of Eq. (3.14)</b>	<b>129</b>
<b>D</b>	<b>Effective sensitivities with corrected coefficients</b>	<b>133</b>
<b>E</b>	<b>Feedforward neural networks</b>	<b>135</b>
<b>F</b>	<b>Empirical cubature method</b>	<b>137</b>
	<b>Bibliography</b>	<b>141</b>
	<b>Summary</b>	<b>157</b>
	<b>List of Publications</b>	<b>159</b>
	<b>Curriculum Vitae</b>	<b>161</b>
	<b>Acknowledgments</b>	<b>163</b>



# Notation

All vectors and tensors are defined in a  $d$ -dimensional Cartesian coordinate frame, with  $\mathbf{e}_i$ ,  $i = 1, \dots, d$ , the unit basis vectors. The Einstein summation convention is assumed on repeated indices.

- scalars  $a$ ,
- vectors  $\mathbf{a} = a_i \mathbf{e}_i$ ,
- position vector  $\mathbf{x} = x_i \mathbf{e}_i$ ,
- second-order tensors  $\mathbf{A} = A_{ij} \mathbf{e}_i \mathbf{e}_j$ ,
- third-order tensors  $\mathcal{A} = \mathcal{A}_{ijk} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k$ ,
- fourth-order tensors  $\mathfrak{A} = \mathfrak{A}_{ijkl} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k \mathbf{e}_l$ ,
- matrices  $\mathbf{A}$  and column matrices  $\mathbf{a}$ ,
- $\mathbf{a} \cdot \mathbf{b} = a_i b_i$ ,
- $\mathbf{a} \otimes \mathbf{b} = a_i b_j \mathbf{e}_i \mathbf{e}_j$ ,
- $\mathbf{A} \mathbf{b} = A_{ij} b_j \mathbf{e}_i$ ,
- $\mathbf{A} \mathbf{B} = A_{ik} B_{kj} \mathbf{e}_i \mathbf{e}_j$ ,
- $\mathbf{A} : \mathbf{B} = A_{ij} B_{ij}$ ,
- $\mathbf{A} \otimes \mathbf{b} = A_{ij} b_k \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k$ ,
- $\mathbf{b} \otimes \mathbf{A} = b_i A_{jk} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k$ ,
- $\mathbf{A} \otimes \mathbf{B} = A_{ij} B_{kl} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k \mathbf{e}_l$ ,
- $\mathbf{a} \cdot \mathcal{A} = a_i \mathcal{A}_{ijk} \mathbf{e}_i \mathbf{e}_j$ ,
- $\mathcal{A} : \mathcal{B} = \mathcal{A}_{ijk} \mathcal{B}_{ijk}$ ,

- $\mathfrak{A} : B = \mathfrak{A}_{ijkl} B_{kl} \mathbf{e}_i \mathbf{e}_j$ ,
- $A : \mathfrak{B} = A_{ij} \mathfrak{B}_{ijkl} \mathbf{e}_k \mathbf{e}_l$ ,
- transpose  $A^T$ ,  $A_{ij}^T = A_{ji}$ ,
- $\text{tr } A = A_{ii}$
- gradient operator with respect to vector  $\mathbf{x}$   
 $\nabla_{\mathbf{x}} \mathbf{a} = \frac{\partial a_i}{\partial x_j} \mathbf{e}_i \mathbf{e}_j$ ,  $\nabla_{\mathbf{x}} A = \frac{\partial A_{ij}}{\partial x_k} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k$ ,
- gradient operator with respect to second-order tensor  $B$   
 $\nabla_B A = \frac{\partial A_{ij}}{\partial B_{kl}} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k \mathbf{e}_l$ ,  $\nabla_B \mathcal{A} = \frac{\partial \mathcal{A}_{ijk}}{\partial B_{lm}} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k \mathbf{e}_l \mathbf{e}_m$ ,
- gradient operator with respect to third-order tensor  $\mathfrak{B}$   
 $\nabla_{\mathfrak{B}} A = \frac{\partial A_{ij}}{\partial \mathfrak{B}_{klm}} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k \mathbf{e}_l \mathbf{e}_m$ ,  $\nabla_{\mathfrak{B}} \mathcal{A} = \frac{\partial \mathcal{A}_{ijk}}{\partial \mathfrak{B}_{lmn}} \mathbf{e}_i \mathbf{e}_j \mathbf{e}_k \mathbf{e}_l \mathbf{e}_m \mathbf{e}_n$ ,
- divergence operator with respect to  $\mathbf{x}$   
 $\nabla_{\mathbf{x}} \cdot A = \frac{\partial A_{ij}}{\partial x_i} \mathbf{e}_j$ ,  $\nabla_{\mathbf{x}} \cdot \mathcal{A} = \frac{\partial A_{ijk}}{\partial x_i} \mathbf{e}_j \mathbf{e}_k$ ,
- Gateaux derivative of a functional  $G$  around state  $\mathbf{a}$  in a direction  $\Delta \mathbf{a}$   
 $DG|_{\mathbf{a}} \cdot (\Delta \mathbf{a}) = \left. \frac{d}{d\tau} G(\mathbf{a} + \tau \Delta \mathbf{a}) \right|_{\tau=0}$ .



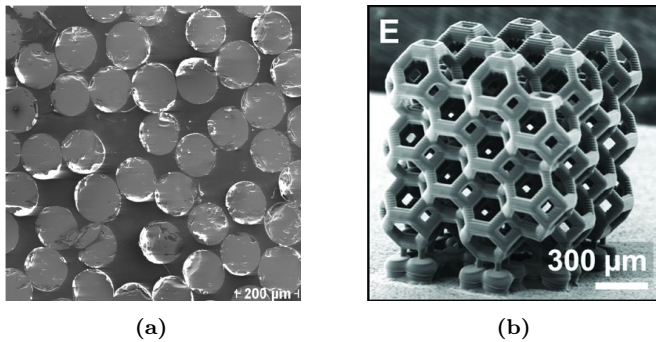
# Chapter 1

## Introduction

Driven by advances in additive manufacturing and design of effective properties for specific applications, there has been a growing interest in understanding the structure-property relationships of complex microstructures. These microstructures can consist of several distinct phases (composite materials, see Fig. 1.1a for an example of a fiber-reinforced composite material) that have attractive effective properties, such as high stiffness, high strength, low weight, etc., due to the interaction of different constituents [60]. Composite materials have found extensive use in diverse sectors, including aerospace [82] and automotive industries [102].

Alternatively, a carefully designed geometry (metamaterials, see Fig. 1.1b for an example lattice metamaterial) can lead to exotic effective properties, such as, for example, negative Poisson's ratio [8], negative compressibility [35], or negative refractive index [108]. Such materials have been used as filters to absorb certain bandwidths of frequencies [72], or to act as acoustic cloaks [138]. Additionally, such materials have been applied in impact mitigation [131] or biomedical applications [96, 43]. A broad overview on their engineering applications can be found in [128].

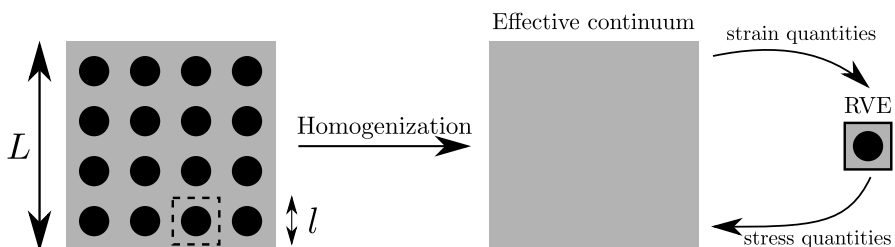
The design space for such microstructures typically involves material and shape parameters, which can result in distinct types of effective behaviors. To study the nonlinear effects of microstructures on effective properties, direct numerical simulations (DNS) are typically utilized. However, as these simulations involve multiple scales, they are generally computationally expensive or even intractable, especially for large-scale engineering applications, since considerably fine meshes are required to capture the complex multiscale geometry. In multi-query contexts, such as the design of materials, where numerous simulations are necessary, the computational costs become infeasible.



**Figure 1.1:** (a) Fiber-reinforced composite material, adapted from [17], and (b) lattice metamaterial, adapted from [152].

## 1.1 Numerical methods for multiscale problems

Over the last few decades, various methodologies have emerged for the efficient treatment of the multiscale partial differential equation (PDE). In general, a split into a macro- and microstructure is performed, which is illustrated schematically in Fig. 1.2 for a composite microstructure with matrix (grey) and inclusions (black). The microstructure is modeled on a representative volume element (RVE) of length scale  $l$ , which captures all fine-scale features, while the macrostructure of size  $L$  is coarse-grained. The size of the RVE is chosen such that its behavior is representative for the microstructure, and effective properties can be computed from it that are utilized for the homogenized macrostructure. Depending on the scale ratio  $\eta := L/l$ , different methods are employed. The most notable methods for addressing the multiscale problem are summarized here, categorized as methods for structures with large or small scale separation.



**Figure 1.2:** Split of multiscale problem into macro- and microscopic problem.

### 1.1.1 Large scale separation

If the length scale of the microstructural features is much smaller than that of the macrostructure (in practice often assumed when  $\eta \geq 10$ ), then the microstructure can be treated as a point on the macroscale. This way, the coarse-grained macrostructure can be modeled as a standard Cauchy continuum and the microstructure essentially establishes a relation between macroscopic strain and stress, see Fig. 1.2.

**Analytical homogenization** For linear elastic materials, several analytical methods were proposed to estimate the effective stiffness of the microstructure. A few notable methods are compiled here.

By assuming uniform strains or stresses in heterogenous materials, upper and lower bounds of the effective stiffness were derived in Voigt [137] and Reuss [112]. The results were later extended for large deformations by Taylor [132] and Sachs [118]. These bounds typically only provide very rough estimates, as they only account for the volume fraction of each component and do not consider the morphology. Sharper bounds on the bulk and shear moduli were derived using variational formulations in Hashin and Shtrikman [47]. Eshelby [29] studied and derived analytical solutions for the stress field of a single ellipsoidal inclusion in an infinite matrix. Later, Mori-Tanaka used Eshelby's result and proposed a method [92] to derive explicit formulas for the effective stiffness tensor, local stresses, and strains for short fiber composites. For all the mentioned methods, there exist numerous extensions and applications, for which a broad overview can be found in [119].

**Asymptotic homogenization** By assuming rapidly oscillating and periodic material coefficients (due to the fast switching between different phases) and employing an asymptotic expansion of the solution field  $u$  of the underlying governing equation [4, 7],

$$u \approx u_\epsilon(x) \approx u_0\left(x, \frac{x}{\epsilon}\right) + \epsilon u_1\left(x, \frac{x}{\epsilon}\right) + \epsilon^2 u_2\left(x, \frac{x}{\epsilon}\right) + \dots,$$

where  $\epsilon := 1/\eta$  is defined and  $y := x/\epsilon$  are fine-scale coordinates assumed to be independent of  $x$ , a hierarchical set of equations on different scales can be derived and solved. These equations define the local cell (microscopic) problem, which is formulated on an RVE and can be solved to find the effective stress and stiffness of the microstructure, and also the averaged macroscopic problem. This approach is very general and does not require any specific assumptions about the microstructure, aside from periodicity. By including more terms in the expansion, higher-order equations can be derived, enabling accurate

solutions for problems without a large scale separation. This has been studied quantitatively for a composite material in Ameen et al. [1], where different numbers of expansion terms were tested.

**First-order computational homogenization** In recent years, first-order computational homogenization (CH) has become a popular framework to study composite materials [127, 93, 89, 86, 133, 37, 85]. The fine-scale geometry of the microstructure is defined in full detail on an RVE, while a coarse-grained representation of the macrostructure with a much coarser discretization is assumed at the macroscale, see Fig. 1.2. The macroscopic strain is used to define a microscopic PDE which is then solved to return an effective stress and stiffness. This methodology is similar to the asymptotic homogenization approach. In fact, if a periodic microstructure is assumed, the same two-scale formulation can be derived with asymptotic homogenization [4].

### 1.1.2 Small scale separation

When both micro- and macroscales are of comparable size, i.e.,  $\eta \leq 10$ , it is often insufficient to only communicate the macroscopic strain with the microstructure, as size and boundary effects play a significant role [1]. To obtain a more accurate homogenized approximation, more information, such as, for example, the strain gradient or other quantities, must be considered, resulting in an enriched two-scale problem. If all information must be communicated (for  $\eta \approx 1$ ), the problem can be solved with a domain decomposition: the macroscopic domain is decomposed into subdomains, each subdomain is solved independently on the interior, and coupled at the subdomain interfaces.

**Enriched computational homogenization** Enriched methods, such as second-order CH [69] or micromorphic CH [36], extend the first-order formulation by introducing additional quantities and equations. For micromorphic CH, additional quantities, that are essential to describe the behavior of the underlying microstructure, are introduced at the macroscale and communicated between both scales. To determine the evolution of these quantities, additional equations are included and a coupled system is solved at the macroscale. As an example, the average strain of the inclusions inside a composite RVE was introduced as an additional quantity in Biswas and Poh [14]. In Jänicke et al. [58], microrotations were considered as additional quantities for cellular materials. For buckling elastomeric metamaterials, prior knowledge on the buckling modes was embedded into the micromorphic framework presented in Rokoš et al. [114].

For second-order CH, a strain gradient formulation is considered at the macroscale, i.e., the gradient of the strain (or deformation gradient) is required, giving rise to a length-scale associated with the length-scale of the underlying RVE, thus making it possible to capture size and non-local effects. To ensure a proper scale transition of the kinematical quantities, additional constraints were derived in Kouznetsova et al. [69]. However, their model leads to artificial stress concentrations at the corners of the RVE, and other formulations were proposed that attempt to improve the model [61, 79, 140, 148]. The influence of the microscopic boundary conditions was studied in [61]. In Luscher et al. [79], additional constraints were derived from orthogonality conditions on the different components of the displacement field on the RVE, and in Wu et al. [140] and Yvonnet et al. [148] body forces were included to account for additional effects.

**Domain decomposition and multiscale/generalized FE methods** Domain decomposition (DD) methods [135, 104, 33] are powerful tools, since they do not introduce any approximation of the fully resolved problem in one scale. By dividing the domain into smaller subdomains and coupling them at the interfaces, the problem becomes highly scalable, since each subdomain can be treated independently. Each subdomain can correspond to one RVE, in which case the displacement of the interfaces correspond to the coarse-grained macrostructure. One notable DD method is the so-called FETI-DP [32], where the problem is solved in the corner points of each subdomain and in the Lagrange multipliers that enforce the interface continuities.

DD methods also have close connections to generalized or multiscale FE [55, 3, 81] methods. In these methods, the local subdomain problems are solved for specific boundary conditions, and problem-dependent FE basis functions are obtained, which are then used to discretize the macroscopic domain. These basis functions capture the effective behavior of each subdomain.

DD methods have also been combined with parametric model order reduction methods to increase the computational efficiency. In the reduced basis element (RBE) method [80], each subdomain is accelerated with a reduced basis and the interfaces are coupled weakly in a non-conforming manner with Lagrange multipliers. In [100], the static condensation reduced basis element (SCRBE) was introduced where the internal degrees of freedom of each subdomain are represented by a reduced basis and condensed out, resulting in a conforming approximation space on the interfaces (also referred to as ports in this context). Those ports correspond to multiscale FE basis functions, which are then employed to solve the macroscopic problem, similar as in [55, 3, 81]. Constructing optimal local approximation spaces for these ports

in two-component systems was discussed in [122], and finding them by local solutions of the PDE with random boundary conditions was proposed in [19]. In the context of solid mechanics, recent applications of such methods include, for example, [26, 50, 51]. For a more comprehensive overview of concepts in localized model order reduction, the interested reader is referred to [18].

## 1.2 Reduced order models for computational homogenization

To numerically solve the two-scale problem arising in CH, a microscopic PDE needs to be solved at every macroscopic integration point, see Fig. 1.2, making CH still computationally expensive for realistic engineering applications. To overcome this computational burden, several different reduced order models have been proposed to speed up or to replace the microscopic simulation in first-order CH. These can be largely classified according to how they treat parameters, and according to the level of their intrusiveness. The ability to handle parameterized problems makes a method suitable for inverse tasks such as parameter identification, material design or optimization. Intrusive methods require the modification of the underlying PDE solver, making them less practical to use. In contrast, non-intrusive (data-driven) methods can be wrapped around an already existent PDE solver, allowing for a simpler adoption.

While there is a wide range of works on accelerating the first-order CH model, there have been no attempts at the reduced order modeling of any enriched formulations to the best of our knowledge. A summary on the existing methods for first-order CH is provided below.

### 1.2.1 Non-intrusive methods

Data-driven methods typically involve the following procedure: first, a large dataset is generated from microscopic simulations, and subsequently, an effective constitutive model is trained from the data, i.e., a mapping from strains to stresses is established through a regression model. Afterwards, the macroscopic problem can be solved on one scale.

A popular data-driven framework was introduced in Kirchdoerfer et al. [67] and extended in [68, 28, 63]. These works propose a distance minimizing scheme, with which pairs of strain and stress data can be directly utilized inside a simulation without the need of deriving an empirical constitutive model. This dataset can come either from experiments or (microscopic) simulations. However, due to the nature of this method, it is only able to describe and

reconstruct a given material dataset, and cannot be used to find optimal microstructures.

Recognizing that stress-strain data is generally difficult to obtain from experiments, a slightly different approach was introduced in Huang et al. [57, 143], where the constitutive model is approximated by a neural network, which is then inversely trained from load-displacement data. Unfortunately, the trained parameters inside the neural network (weights and biases) do not represent actual physical parameters, and therefore the method can only describe the given dataset.

Several other works attempted to learn a constitutive model directly from pairs of stress and strain data with different neural network architectures [38, 39, 40, 141, 78, 94, 75, 84]. After the pioneering works in Ghaboussi et al. [38], several works have applied deep neural networks (DNN) to different areas of constitutive modeling. In [40, 141, 78], different recurrent neural network architectures were proposed to learn an inelastic material model from stress and strain loading data that were obtained from RVE simulations. However, the models were trained only with deformation data and did not consider parameters, inhibiting their use for finding new materials and designs. Recent works have also attempted to overcome this challenge. In Le et al. [73], it was proposed to use neural networks to learn the effective potential of a hyperelastic material from which the effective stress and stiffness can be derived, while also including multiple microstructural parameters. In Mozaffar et al. [94] a recurrent neural network was proposed that can simultaneously treat both inelastic material behavior and material parameters, and the stress prediction for a class of composite materials was illustrated. More recently, works [75, 84, 143] have also attempted to construct a neural network that learns constitutive models that fulfill the thermodynamical laws as defined in [97].

Although these neural network-based methods are capable of finding accurate surrogate models, they typically require huge amounts of data, which are often generated from simulations. For complex microstructures, running these simulations repeatedly might be computationally too expensive. Furthermore, these surrogate models generally only consider the effective quantities, which are obtained by averaging the microscopic quantities, and neglect all the field information of the microscopic simulation and the microscopic physics. This also makes it difficult to quantify the errors of the predictions, as the connection between macro- and microstructure is completely lost and local microscopic quantities can no longer be recovered.

### 1.2.2 Intrusive methods

Intrusive methods attempt to accelerate the existing microscopic solver. For instance, if the fast Fourier transform (FFT) [93, 91] is used to simulate the microstructure, its solution can be accelerated by the (nonuniform) transformation field analysis (see, e.g., [27, 87]), or self-consistent clustering analysis [77, 147]. One disadvantage of FFT is that parameterizations of the RVE cannot be directly treated and, hence, sensitivities for material optimization cannot be directly computed. Furthermore, FFT has convergence issues when high stiffness contrasts are considered in the microstructure.

If the microscopic problem is solved via the FE method, the resulting multiscale formulation is referred to as FE2 [34, 37, 88]. By directly solving the microscopic PDE with FE, material or shape parameterizations can be considered in a straightforward manner, making the approach more suitable for inverse problems and optimization. To speed up the microstructural simulation, proper orthogonal decomposition (POD) [103, 52] can be utilized to find a reduced set of basis functions; the method then computes the Galerkin projection of the solution onto the space spanned by the reduced basis. Although POD generally requires several full-order solves for constructing the reduced basis, it typically works well for all input parameters. In the context of first-order CH, POD was first applied in Yvonnet et al. [149] for a hyperelastic RVE, and later explored in Radermacher et al. [106] for an elasto-plastic RVE under small strains.

However, due to the non-linearities of the microscopic problem, the speed-ups were limited since the global force vector and stiffness matrix must be assembled by full integration in every microscopic Newton iteration. To address this issue, a further reduction called hyperreduction is required, which aims at finding an efficient way of assembling microstructural force and stiffness quantities. Notable hyperreduction methods are empirical interpolation method (EIM) (see, e.g., [5]), a variant of EIM called discrete empirical interpolation method (DEIM) (see, e.g., [23]), energy-based mesh sampling and weighting [30], reduced integration domain [117], empirical quadrature procedure [145], and empirical cubature method (ECM) [48]. EIM and DEIM interpolate the non-linear integrand of the global force vector such that the integrals can be pre-computed. In [49, 124], DEIM was used successfully to accelerate the solution of the microscopic PDE. However, these works only discussed the solution of the microscopic PDE and did not derive the effective stress and stiffness quantities required for the macroscopic problem. A possible disadvantage of EIM and DEIM is that they lead to non-symmetric tangent matrices, which might result in convergence issues, observed in, for instance, [124, 107]. The rest of the above-mentioned hyperreduction methods aim at approximating



the integrals by finding a subset of integration points with corresponding positive weights among the set of all integration points used in the formulation of the microscopic PDE. This has the advantage that the stiffness matrix is always symmetric and at least positive semi-definite (in practice usually positive definite unless instabilities occur), ensuring a good convergence of the microscopic problem. Hyperreduction methods have been successfully employed in two-scale simulations in [20], where an elasto-plastic composite RVE under large deformations was considered, and in [109], where a damage model for a composite RVE under small deformations was handled. While both works obtained accurate results and successfully accelerated the forward simulations of a two-scale problem, such formulations were limited to fixed microstructures only, i.e., did not account for possible parameterizations. In order to allow for optimization of microstructures, the surrogate model needs to be extended to a wide range of different design parameters (including geometrical as well as material).

### 1.3 Main contributions

Based on the above discussion, there are several open issues in the reduced order modeling of CH:

1. Although the existing non-intrusive methods can find good approximations of the effective constitutive model, they typically require large amounts of data, neglect the microscopic quantities and physics, their predictions are hard to interpret, or design parameters cannot be treated.
2. The existing intrusive methods have so far only been applied to fixed microstructures and can therefore not be utilized for design or optimization applications.
3. Many reduced order models have been developed for first-order CH. However, there are no works for other CH schemes to the best of our knowledge.

To address these issues, in this work, both non-intrusive and intrusive reduced order models are developed for first- and second-order computational homogenization, with particular focus on parameterized microstructures. Our models could potentially be utilized in applications such as the design of new materials with specified or optimal effective properties, or to study uncertainties of additively manufactured materials due to variability of the manufacturing process. The main contributions for first-order computational homogenization are:

- The microscopic problem is formulated as a parameterized PDE, making it amenable to compute material and geometry sensitivities for material design.
- A data-driven non-intrusive reduced order model is established for hyper-elastic microstructures, which in comparison to other existing methods is more data efficient and allows the user to recover the microscopic stress field.
- An intrusive reduced order model for elasto-plastic microstructures under large deformations is developed that can handle a wide range of material and geometry parameters.
- The results of both ROMs are compared to full FE2 solutions, highlighting the accuracy and speed-up of the developed methodologies.

Building on ideas from first-order computational homogenization, we make the following contributions for second-order computational homogenization:

- Same as for the first-order case, the microscopic problem is formulated as a parameterized PDE.
- An intrusive reduced order model for the second-order parameterized microscopic problem is developed, and a novel hyperreduction method is proposed.
- Comparisons with the direct numerical simulation, where all microstructural features are resolved at the macroscopic scale, and the full second-order CH solution are conducted, and demonstrate that the ROM achieves high accuracy and speed-ups.

## 1.4 Outline

In Chapter 2, the formulation of first-order computational homogenization is presented. Both macroscopic and microscopic problems are detailed, with a particular focus on parameterization of the microstructure and including a discussion on the derivation of effective quantities, numerical solution and computational costs.

To accelerate the parameterized microscopic problem, a non-intrusive reduced order model is proposed in Chapter 3. By employing proper orthogonal decomposition, Gaussian process regression, and geometrical transformations, an effective constitutive model is learned which can fully replace the microscopic

problem for many parameters. Numerical experiments are shown to validate the method, and limitations of the method are discussed.

A different approach is followed in Chapter 4. By combining the reduced basis method and the empirical cubature method, an intrusive reduced order model is derived, which resolves the issues of the non-intrusive method. Numerical examples, motivated by energy-absorbing metamaterials, are presented to validate the method.

Chapter 5 extends the method proposed in Chapter 4 to second-order computational homogenization. After briefly reviewing the second-order theory, a reduced order model is proposed. It utilizes the reduced basis method and a novel hyperreduction method, which is inspired by the empirical cubature method and tailored specifically for the second-order problem. Comparing the numerical solutions for several examples, obtained through the surrogate, full second-order and direct numerical simulation models, illustrates the accuracy and efficiency of the proposed model.

Concluding remarks and a summary are given in Chapter 6, together with recommendations for future research.



## Chapter 2

# First-order computational homogenization

The content of this chapter is based on the following publications:

- Guo, T., Rokoš, O., & Veroy, K. (2021). Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 384, 113924.
- Guo, T., Rokoš, O., & Veroy, K. (2024). A reduced order model for geometrically parameterized two-scale simulations of elasto-plastic microstructures under large deformations. *Computer Methods in Applied Mechanics and Engineering*, 418, 116467.

This chapter presents the two-scale problem arising in first-order computational homogenization. In the first part, see Section 2.1, the macroscopic problem is reviewed. Kinematical quantities and the governing partial differential equations are introduced, and details on the weak formulation and the numerical solution are provided. The microscopic problem is presented in the second part, see Section 2.2. Details on the governing equations, boundary conditions and numerical solution are discussed as well. In contrary to the usual definition, the microscopic problem is assumed to be parameterized, i.e., the microstructure can have different material, geometrical or other parameters. This has the advantage that efficient reduced order models can be constructed for a wide range of parameters, which can be used for the design or optimization of materials.

## 2.1 Macroscopic problem

Consider a solid body  $\bar{\Omega} \subset \mathbb{R}^d$ , with  $d = 2, 3$  the space dimension, that is deformed under prescribed boundary conditions. Under the external forces, each material point  $\bar{\mathbf{x}} \in \bar{\Omega}$  of the undeformed body is continuously displaced with a displacement vector  $\bar{\mathbf{u}}(\bar{\mathbf{x}})$ . The deformation gradient, characterizing local deformation in a close vicinity of that material point, can be defined as

$$\bar{\mathbf{F}} := \mathbf{I} + \nabla_{\bar{\mathbf{x}}} \bar{\mathbf{u}}, \quad (2.1)$$

where  $\mathbf{I}$  is the identity tensor. The boundary value problem (BVP) governing the deformation is based on the (quasi-static) linear momentum balance and described by the following set of equations,

$$\nabla_{\bar{\mathbf{x}}} \cdot \bar{\mathbf{P}}^T + \bar{\mathbf{b}} = \mathbf{0} \quad \text{on } \bar{\Omega}, \quad (2.2a)$$

$$\bar{\mathbf{P}} \bar{\mathbf{n}} = \bar{\mathbf{t}}_0 \quad \text{on } \partial \bar{\Omega}^N, \quad \text{and} \quad (2.2b)$$

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_0 \quad \text{on } \partial \bar{\Omega}^D, \quad (2.2c)$$

where  $\bar{\mathbf{P}}$  is the second-order first Piola-Kirchhoff (1PK) stress tensor,  $\bar{\mathbf{b}}$  are the body forces,  $\bar{\mathbf{n}}$  is the outward normal on the surface of the body  $\partial \bar{\Omega}$ ,  $\bar{\mathbf{t}}_0$  and  $\bar{\mathbf{u}}_0$  are the prescribed traction and displacement, and  $\partial \bar{\Omega}^N$ ,  $\partial \bar{\Omega}^D$  represent the Neumann and Dirichlet boundaries with  $\partial \bar{\Omega}^N \cup \partial \bar{\Omega}^D = \partial \bar{\Omega}$  and  $\partial \bar{\Omega}^N \cap \partial \bar{\Omega}^D = \emptyset$ .

The stress tensor  $\bar{\mathbf{P}}$  is typically related to the deformation gradient  $\bar{\mathbf{F}}$  (or its history) with a constitutive model, and thus depends on the displacement  $\bar{\mathbf{u}}$ . For complicated materials or microstructures, finding a closed-form expression relating the stress  $\bar{\mathbf{P}}$  and  $\bar{\mathbf{F}}$ , i.e., phenomenological constitutive law, is often challenging or even impossible. In first-order computational homogenization,

the relation is derived numerically by solving a microscopic problem, which will be explained in depth in Section 2.2, and thus is based on first principles. For the remainder of this section, the relation is assumed to be implicitly defined.

By multiplying Eq. (2.2a) with a test function  $\delta\bar{\mathbf{u}} \in (H_0^1(\bar{\Omega}))^d$  and applying the divergence theorem, the weak form of Eq. (2.2) can be derived,

$$\bar{G}(\bar{\mathbf{u}}) := \int_{\bar{\Omega}} \nabla_{\bar{\mathbf{x}}} \delta\bar{\mathbf{u}} : \bar{\mathbf{P}}(\bar{\mathbf{F}}) d\bar{\mathbf{x}} - \int_{\bar{\Omega}} \bar{\mathbf{b}} \cdot \delta\bar{\mathbf{u}} d\bar{\mathbf{x}} - \int_{\partial\bar{\Omega}^N} \bar{\mathbf{t}}_0 \cdot \delta\bar{\mathbf{u}} d\bar{\mathbf{x}} = 0, \quad (2.3)$$

Here,  $H_0^1(\bar{\Omega}) := \{v \in H^1(\bar{\Omega}) \mid v = 0 \text{ on } \partial\bar{\Omega}^D\}$  and  $H^1(\bar{\Omega})$  is the Hilbert space with square integrable functions and derivatives on  $\bar{\Omega}$ . For prescribed boundary conditions, a solution  $\bar{\mathbf{u}}^* \in \mathcal{V} := \{\mathbf{v} \in (H^1(\bar{\Omega}))^d \mid \mathbf{v} = \mathbf{u}_0 \text{ on } \partial\bar{\Omega}^D\}$  is sought that fulfills Eq. (2.3) for all test functions  $\delta\bar{\mathbf{u}}$ . As the constitutive model  $\bar{\mathbf{P}}(\bar{\mathbf{F}})$  is usually non-linear, the linearization of Eq. (2.3) in the direction  $\Delta\bar{\mathbf{u}}$  around the current deformation  $\bar{\mathbf{u}}$  is required, resulting in,

$$D\bar{G}|_{\bar{\mathbf{u}}} \cdot (\Delta\bar{\mathbf{u}}) = \int_{\bar{\Omega}} \nabla_{\bar{\mathbf{x}}} \delta\bar{\mathbf{u}} : \bar{\mathfrak{A}}(\bar{\mathbf{F}}(\bar{\mathbf{u}})) : \nabla_{\bar{\mathbf{x}}} \Delta\bar{\mathbf{u}} d\bar{\mathbf{x}}, \quad (2.4)$$

where  $\bar{\mathfrak{A}} := \nabla_{\bar{\mathbf{F}}} \bar{\mathbf{P}}$  is the fourth-order material stiffness tensor.

To find a solution  $\bar{\mathbf{u}}^*$  that fulfills Eq. (2.3), the Newton method is usually employed. Starting from an initial guess  $\bar{\mathbf{u}}^0$  and Newton iteration  $m = 0$ , the following linear system needs to be solved for  $\Delta\bar{\mathbf{u}}$  (for a fixed current iterate  $\bar{\mathbf{u}}^m$ ),

$$\int_{\bar{\Omega}} \nabla_{\bar{\mathbf{x}}} \delta\bar{\mathbf{u}} : \bar{\mathfrak{A}}(\bar{\mathbf{F}}(\bar{\mathbf{u}}^m)) : \nabla_{\bar{\mathbf{x}}} \Delta\bar{\mathbf{u}} d\bar{\mathbf{x}} = -\bar{G}(\bar{\mathbf{u}}^m, \delta\bar{\mathbf{u}}). \quad (2.5)$$

After solving Eq. (2.5) for  $\Delta\bar{\mathbf{u}}$ , the solution  $\bar{\mathbf{u}}^m$  and iteration number  $m$  are incremented with  $\bar{\mathbf{u}}^{m+1} = \bar{\mathbf{u}}^m + \Delta\bar{\mathbf{u}}$  and  $m = m + 1$ , and the previous steps are repeated until Eq. (2.3) is fulfilled (up to some user-defined tolerance). If the starting value  $\bar{\mathbf{u}}^0$  is close to the true solution  $\bar{\mathbf{u}}^*$ , a quadratic convergence behavior can be expected. However, if the initial guess is not good, the Newton method might not converge. In that case, the boundary conditions are usually applied gradually in multiple load steps or other techniques to stabilize the iterative procedure are used.

**Remark 2.1.1** *Sometimes, it is helpful to compute a polar decomposition of the deformation gradient  $\bar{\mathbf{F}} = \bar{\mathbf{R}}\bar{\mathbf{U}}$  with orthogonal rotation  $\bar{\mathbf{R}}$  and symmetric positive-definite stretch tensor  $\bar{\mathbf{U}}$ . Due to the principle of material objectivity, a constitutive model always fulfills the following property:*

$$\bar{\mathbf{P}}(\bar{\mathbf{F}}) = \bar{\mathbf{R}}(\bar{\mathbf{F}}) \bar{\mathbf{P}}(\bar{\mathbf{U}}(\bar{\mathbf{F}})). \quad (2.6)$$

Since  $\bar{\mathbf{U}}$  is symmetric, it has fewer independent components as compared to  $\bar{\mathbf{F}}$ . In subsequent chapters, where surrogate models for  $\bar{\mathbf{P}}$  are constructed, this implies that the input space  $\bar{\mathbf{U}}$  for the surrogate model is reduced.

**Remark 2.1.2** *To solve the macroscopic problem, the problem in continuous form of Eqs. (2.3)–(2.5) is typically discretized and solved with the standard Galerkin finite element method (FEM). More details on the finite element method and discretization of weak forms, can be found in [6].*

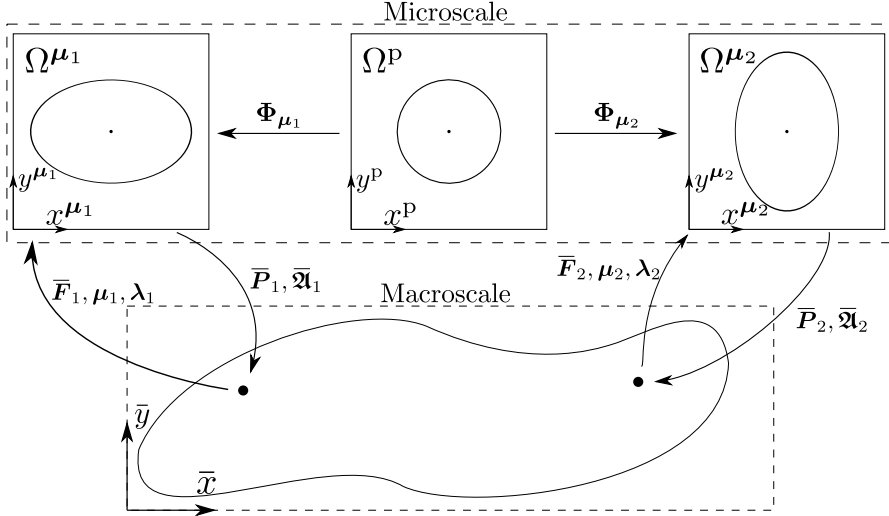
## 2.2 Parameterized microscopic problem

In first-order computational homogenization, while the macroscopic problem is governed by the standard linear momentum balance, the macroscopic constitutive model (relating strains to stresses and stiffness) is replaced by a microscopic partial differential equation (PDE) which is defined on a representative volume element (RVE) that captures the representative behavior of the microstructure. By prescribing the macroscopic deformation gradient  $\bar{\mathbf{F}}$  (or stretch tensor  $\bar{\mathbf{U}}$ ) at the microscale, the PDE can be solved and effective stress and stiffness returned to the macroscopic solver, see Fig. 2.1. For applications such as microstructure optimization, it is reasonable to additionally introduce a parameterization of the RVE with design variables  $\boldsymbol{\mu}$  (geometry) and  $\boldsymbol{\lambda}$  (material) in order to compute relevant sensitivities. The microscopic PDE is formulated below on a parameterized domain, as is usually the case in shape optimization. For brevity, the dependence on the macroscopic coordinates is omitted and a fixed macroscopic material point is assumed unless otherwise specified.

Consider a family of domains  $\Omega^\mu \subset \mathbb{R}^d$ , parameterized by geometrical parameters  $\boldsymbol{\mu} \in \mathbb{P}_\mu$ , with  $\mathbb{P}_\mu$  the parameter space of geometrical parameters, and spanned by position vectors  $\mathbf{x}^\mu \in \Omega^\mu$ . For all  $\boldsymbol{\mu}$ , the outer boundaries and topology of  $\Omega^\mu$  are assumed to remain fixed (the outer boundaries of the RVE domain are fixed while the shape of the interior geometry can change). As a consequence, the volume  $|\Omega^\mu|$  remains constant for all  $\boldsymbol{\mu}$ . Additionally, it is assumed that there exists a parent domain  $\Omega^p := \Omega^{\mu^p}$  with  $\boldsymbol{\mu}^p \in \mathbb{P}_\mu$ , which can be transformed into any  $\Omega^\mu$  with a transformation map  $\Phi_\mu : \Omega^p \rightarrow \Omega^\mu$ ,  $\mathbf{x}^p \mapsto \mathbf{x}^\mu$ , transformation gradient  $\mathbf{F}_\mu := \nabla_{\mathbf{x}^p} \Phi_\mu$  and  $d\mathbf{x}^\mu = |\det \mathbf{F}_\mu| d\mathbf{x}^p$ . In Fig. 2.1, an example parent domain with a circular inclusion  $\Omega^p$  is geometrically parameterized and mapped to two distinct parameterized domains with elliptical inclusions,  $\Omega^{\mu^1}$  and  $\Omega^{\mu^2}$ .

By assuming scale separation between macro- and microscale, the microscopic displacement field on the parameterized domain  $\mathbf{u}(\mathbf{x}^\mu)$  can be writ-





**Figure 2.1:** Two-scale problem based on first-order homogenization. At every macroscopic point, a microscopic simulation is defined through deformation gradient  $\bar{\mathbf{F}}$ , shape parameters  $\boldsymbol{\mu}$  and material parameters  $\boldsymbol{\lambda}$ , and solved to obtain an effective stress  $\bar{\mathbf{P}}$  and stiffness  $\bar{\mathbf{A}}$ . For different macroscopic points, different parameterized microstructures can be considered through  $\boldsymbol{\mu}$  and  $\boldsymbol{\lambda}$ . As an example of a family of geometrically parameterized microstructures, a parent domain with a circular inclusion  $\Omega^P$  (center), can be mapped onto parameterized domains  $\Omega^{\mu_1}$  (top left) and  $\Omega^{\mu_2}$  (top right) with mappings  $\Phi_{\mu_1}$  and  $\Phi_{\mu_2}$ .

ten as the summation of a mean field  $\bar{\mathbf{u}}(\mathbf{x}^\mu)$  and a fluctuation field  $\mathbf{w}(\mathbf{x}^\mu)$ , i.e.,  $\mathbf{u}(\mathbf{x}^\mu) = \bar{\mathbf{u}}(\mathbf{x}^\mu) + \mathbf{w}(\mathbf{x}^\mu)$ . The mean field is fully specified through  $\bar{\mathbf{u}}(\mathbf{x}^\mu) := (\bar{\mathbf{F}} - \mathbf{I})\mathbf{x}^\mu$ , where  $\bar{\mathbf{F}} \in \mathbb{P}_{\bar{\mathbf{F}}}$  is the macroscopic deformation gradient tensor and  $\mathbb{P}_{\bar{\mathbf{F}}}$  the parameter space of macroscopic loads. The total deformation gradient tensor  $\mathbf{F}$  is defined as

$$\mathbf{F}(\mathbf{w}) := \mathbf{I} + \nabla_{\mathbf{x}^\mu} \mathbf{u} = \bar{\mathbf{F}} + \nabla_{\mathbf{x}^\mu} \mathbf{w}. \quad (2.7)$$

The governing microscopic PDE is given as

$$\nabla_{\mathbf{x}^\mu} \cdot \mathbf{P}^T(\mathbf{F}(\mathbf{w})) = \mathbf{0} \quad \text{on } \Omega^\mu, \quad (2.8)$$

where  $\mathbf{P}$  denotes the microscopic first Piola-Kirchhoff (1PK) stress tensor. No constitutive model is specified at this point, although we assume that the stress  $\mathbf{P}$  is a known non-linear function of the deformation gradient  $\mathbf{F}$  (or its history) and contains material parameters  $\boldsymbol{\lambda} \in \mathbb{P}_{\boldsymbol{\lambda}}$  with  $\mathbb{P}_{\boldsymbol{\lambda}}$  the material parameter space. A list of constitutive models considered in this work is provided in Appendix A. The weak form of the problem is then: given  $(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{P} := \mathbb{P}_{\bar{\mathbf{F}}} \otimes \mathbb{P}_{\boldsymbol{\lambda}} \otimes \mathbb{P}_{\boldsymbol{\mu}}$ , find the fluctuation field  $\mathbf{w}^* \in \mathcal{V}$  that fulfills

$$G(\mathbf{w}) := \int_{\Omega^\mu} \nabla_{\mathbf{x}^\mu} \delta \mathbf{w} : \mathbf{P}(\bar{\mathbf{F}} + \nabla_{\mathbf{x}^\mu} \mathbf{w}) \, d\mathbf{x}^\mu \stackrel{!}{=} 0, \quad \forall \delta \mathbf{w} \in \mathcal{V}, \quad (2.9)$$

where the integral bounds depend on the parameters  $\boldsymbol{\mu}$  and  $\delta\boldsymbol{w}$  denotes a test function. From Eq. (2.9), it is apparent that the macroscopic deformation gradient  $\bar{\mathbf{F}}$  represents the external loading, while the fluctuation displacement field  $\boldsymbol{w}$  balances the system. The definition of the function space  $\mathcal{V}$  depends on the boundary conditions, which will be discussed in Section 2.2.2. However, it is assumed that  $\mathcal{V} \subset (H^1(\Omega^\mu))^d$  and equipped with the inner product,

$$(\boldsymbol{u}, \boldsymbol{v})_{\mathcal{V}} := \int_{\Omega^\mu} (\boldsymbol{u} \cdot \boldsymbol{v} + \nabla_{\boldsymbol{x}^\mu} \boldsymbol{u} : \nabla_{\boldsymbol{x}^\mu} \boldsymbol{v}) d\boldsymbol{x}^\mu, \quad (2.10)$$

and induced norm  $\|\boldsymbol{u}\|_{\mathcal{V}} = \sqrt{(\boldsymbol{u}, \boldsymbol{u})_{\mathcal{V}}}$ .

With the aid of the parametric transformation map  $\Phi_\mu$ , see Fig. 2.1, and using integration by substitution, the problem of Eq. (2.9) can be simplified and restated on the parent domain  $\Omega^p$  as follows: given  $(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{P}$ , find  $\boldsymbol{w}^{*p} \in \mathcal{V}^p \subset (H^1(\Omega^p))^d$  that fulfills for all  $\delta\boldsymbol{w}^p \in \mathcal{V}^p$ ,

$$\begin{aligned} G^p(\boldsymbol{w}^p) &:= \int_{\Omega^p} ((\nabla_{\boldsymbol{x}^p} \delta\boldsymbol{w}^p) \mathbf{F}_\mu^{-1}) : \mathbf{P}^p(\mathbf{F}^p) |\det \mathbf{F}_\mu| d\boldsymbol{x}^p = 0, \\ \mathbf{F}^p(\boldsymbol{w}^p) &:= \bar{\mathbf{F}} + (\nabla_{\boldsymbol{x}^p} \boldsymbol{w}^p) \mathbf{F}_\mu^{-1}, \end{aligned} \quad (2.11)$$

with the transformation gradient  $\mathbf{F}_\mu := \nabla_{\boldsymbol{x}^p} \Phi_\mu$  and  $d\boldsymbol{x}^\mu = |\det \mathbf{F}_\mu| d\boldsymbol{x}^p$ . The superscript p is used to denote quantities pertinent to the parent domain, e.g.,  $\boldsymbol{w}(\boldsymbol{x}^\mu) = (\boldsymbol{w} \circ \Phi_\mu)(\boldsymbol{x}^p) = \boldsymbol{w}^p(\boldsymbol{x}^p)$ ,  $\mathbf{P}(\boldsymbol{x}^\mu) = \mathbf{P}^p(\boldsymbol{x}^p)$ . To iteratively solve the non-linear problem in Eq. (2.11), a linearization using the Gateaux derivative around the current state  $\boldsymbol{w}^p$  in direction  $\Delta\boldsymbol{w}^p \in \mathcal{V}^p$  is required and can be written as,

$$\begin{aligned} &DG^p|_{\boldsymbol{w}^p} \cdot (\Delta\boldsymbol{w}^p) \\ &= \int_{\Omega^p} ((\nabla_{\boldsymbol{x}^p} \delta\boldsymbol{w}^p) \mathbf{F}_\mu^{-1}) : \mathfrak{A}^p(\mathbf{F}^p) : ((\nabla_{\boldsymbol{x}^p} \Delta\boldsymbol{w}^p) \mathbf{F}_\mu^{-1}) |\det \mathbf{F}_\mu| d\boldsymbol{x}^p, \end{aligned} \quad (2.12)$$

where  $\mathfrak{A}^p$  is a fourth-order tensor obtained by transforming the material stiffness tensor  $\mathfrak{A} := \nabla_{\mathbf{F}} \mathbf{P}$  from the parameterized to the parent domain. Similarly to the macroscopic problem, the solution can be found using the Newton method, cf. Eq. (2.5). More details are provided in Section 2.2.3.

To summarize, once the transformation map  $\Phi_\mu$  is known, the geometrically parameterized microscopic problem Eq. (2.9) can be solved on the parent domain using Eqs. (2.11) and (2.12). In general, such transformations are not known analytically, but are instead computed numerically. Various methods to compute such maps exist, such as

- free form deformation [120, 139],

- radial basis functions [83, 126],
- registration method [130], and
- mesh-based methods [31, 146, 134].

Further details on how to find these transformations efficiently and especially in the context of reduced order modeling are presented in Section 3.1.1.

### 2.2.1 Effective quantities

For conciseness of notation, the following abbreviations are introduced to denote quantities evaluated at the solution  $\mathbf{w}^{*\text{P}}$ :

$$\mathbf{F}^{*\text{P}} := \bar{\mathbf{F}} + (\nabla_{\mathbf{x}^{\text{P}}} \mathbf{w}^{*\text{P}}) \mathbf{F}_{\mu}^{-1}, \quad (2.13)$$

$$\mathbf{P}^{*\text{P}} := \mathbf{P}^{\text{P}}(\mathbf{F}^{*\text{P}}), \quad (2.14)$$

$$\mathfrak{A}^{*\text{P}} := \mathfrak{A}^{\text{P}}(\mathbf{F}^{*\text{P}}). \quad (2.15)$$

Upon obtaining solution  $\mathbf{w}^{*\text{P}}$  that fulfills Eq. (2.11), the effective stress is computed through standard volume averaging as

$$\begin{aligned} \bar{\mathbf{P}} &:= |\Omega^{\mu}|^{-1} \int_{\Omega^{\mu}} \mathbf{P}^* d\mathbf{x}^{\mu} \\ &= |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \mathbf{P}^{*\text{P}} |\det \mathbf{F}_{\mu}| d\mathbf{x}^{\text{P}}, \end{aligned} \quad (2.16)$$

and the effective stiffness (in index notation) as

$$\begin{aligned} \bar{\mathfrak{A}}_{ijkl} &:= \frac{\partial \bar{P}_{ij}}{\partial \bar{F}_{kl}} \\ &= |\Omega^{\text{P}}|^{-1} \frac{\partial}{\partial \bar{F}_{kl}} \int_{\Omega^{\text{P}}} P_{ij}^{*\text{P}} |\det \mathbf{F}_{\mu}| d\mathbf{x}^{\text{P}} \\ &= |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \mathfrak{A}_{ijmn}^{*\text{P}} \frac{\partial F_{mn}^{*\text{P}}}{\partial \bar{F}_{kl}} |\det \mathbf{F}_{\mu}| d\mathbf{x}^{\text{P}}, \end{aligned} \quad (2.17)$$

with

$$\frac{\partial F_{mn}^{*\text{P}}}{\partial \bar{F}_{kl}} = \delta_{mk} \delta_{nl} + \frac{\partial}{\partial \bar{F}_{kl}} \left( \frac{\partial w_m^{*\text{P}}}{\partial x_r^{\text{P}}} \right) (F_{\mu}^{-1})_{rn}, \quad (2.18)$$

where  $\delta_{mk}$  denotes the Kronecker delta. To determine  $\frac{\partial}{\partial \bar{F}_{kl}} \left( \frac{\partial w_m^{*\text{P}}}{\partial x_r^{\text{P}}} \right)$ , Eq. (2.11) is differentiated with respect to  $\bar{\mathbf{F}}$  to form linear tangent problems that can

be solved. For one particular component  $\bar{F}_{kl}$  (where the indices  $k$  and  $l$  are assumed to be temporarily fixed), the differentiation yields

$$\int_{\Omega^{\text{P}}} \left( (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1} \right) : \boldsymbol{\mathfrak{A}}^{*\text{P}} : (\mathbf{E}_{kl} + (\nabla_{\mathbf{x}^{\text{P}}} \mathbf{q}_{kl}) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}} = 0, \quad (2.19)$$

where a new auxiliary vector field  $\mathbf{q}_{kl} := \frac{\partial \mathbf{w}^{*\text{P}}}{\partial \bar{F}_{kl}} \in \mathcal{V}^{\text{P}}$  has been defined (reflecting the sensitivity of the microfluctuation field with respect to the change of the applied macroscopic loading), and  $\mathbf{E}_{kl} \in \mathbb{R}^{d \times d}$  is a second order tensor with all entries zero, except for the  $kl$ -th entry which is 1. Eq. (2.19) can be rearranged as,

$$\begin{aligned} & \int_{\Omega^{\text{P}}} \left( (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1} \right) : \boldsymbol{\mathfrak{A}}^{*\text{P}} : ((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{q}_{kl}) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}} \\ &= - \left( \int_{\Omega^{\text{P}}} \left( (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1} \right) : \boldsymbol{\mathfrak{A}}^{*\text{P}} |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}} \right) : \mathbf{E}_{kl}. \end{aligned} \quad (2.20)$$

The linear tangent problem in Eq. (2.20) is then solved for all combinations  $k, l = 1, \dots, d$  to obtain  $\mathbf{q}_{kl}$  for each component of  $\bar{\mathbf{F}}$ .

Although not explicitly utilized in this work, the sensitivities of the effective stress  $\bar{\mathbf{P}}$  with respect to the geometrical parameters  $\boldsymbol{\mu}$ , which are required for applications such as shape optimization, can be computed with the geometrically parameterized formulation of the RVE as follows (in index notation),

$$\begin{aligned} \frac{\partial \bar{P}_{ij}}{\partial \mu_k} &= |\Omega^{\text{P}}|^{-1} \frac{\partial}{\partial \mu_k} \int_{\Omega^{\text{P}}} P_{ij}^{*\text{P}} |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}} \\ &= |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \left( \boldsymbol{\mathfrak{A}}_{ijmn}^{*\text{P}} \frac{\partial F_{mn}^{*\text{P}}}{\partial \mu_k} |\det \mathbf{F}_{\boldsymbol{\mu}}| + P_{ij}^{*\text{P}} \frac{\partial |\det \mathbf{F}_{\boldsymbol{\mu}}|}{\partial \mu_k} \right) d\mathbf{x}^{\text{P}}, \end{aligned} \quad (2.21)$$

with

$$\frac{\partial F_{mn}^{*\text{P}}}{\partial \mu_k} = \frac{\partial}{\partial \mu_k} \left( \frac{\partial w_m^{*\text{P}}}{\partial x_r^{\text{P}}} \right) (F_{\boldsymbol{\mu}}^{-1})_{rn} + \frac{\partial w_m^{*\text{P}}}{\partial x_r^{\text{P}}} \frac{\partial (F_{\boldsymbol{\mu}}^{-1})_{rn}}{\partial \mu_k}. \quad (2.22)$$

The integrand is complicated due to the derivatives of  $\mathbf{F}_{\boldsymbol{\mu}}^{-1}$  and  $|\det \mathbf{F}_{\boldsymbol{\mu}}|$ , but, in principle, these derivatives can be computed for a given geometrical mapping  $\boldsymbol{\Phi}_{\boldsymbol{\mu}}$ . Analogously, the sensitivities of the effective stress  $\bar{\mathbf{P}}$  with respect to the material parameters  $\boldsymbol{\lambda}$  could be computed with,

$$\begin{aligned} \frac{\partial \bar{P}_{ij}}{\partial \lambda_k} &= |\Omega^{\text{P}}|^{-1} \frac{\partial}{\partial \lambda_k} \int_{\Omega^{\text{P}}} P_{ij}^{*\text{P}} |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}} \\ &= |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \left( \boldsymbol{\mathfrak{A}}_{ijmn}^{*\text{P}} \frac{\partial F_{mn}^{*\text{P}}}{\partial \lambda_k} + \frac{\partial P_{ij}^{*\text{P}}}{\partial \lambda_k} \right) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}}, \end{aligned} \quad (2.23)$$

with

$$\frac{\partial F_{mn}^{*p}}{\partial \lambda_k} = \frac{\partial}{\partial \lambda_k} \left( \frac{\partial w_m^{*p}}{\partial x_r^p} \right) (F_{\mu}^{-1})_{rn}. \quad (2.24)$$

For practical reasons, if the effective stress  $\bar{\mathbf{P}}$  can be assumed to vary smoothly with the parameters  $\boldsymbol{\mu}$  and  $\boldsymbol{\lambda}$ , finite differences could be used to approximate these sensitivities.

### 2.2.2 Scale coupling and boundary conditions

To ensure a proper physical coupling of macro- and microscale, the so-called **Hill-Mandel Condition** [54] must be fulfilled, which states that the averaged microscopic virtual work has to be equal to the macroscopic virtual work, i.e.,

$$|\Omega^{\mu}|^{-1} \int_{\Omega^{\mu}} \mathbf{P} : \delta \mathbf{F} d\mathbf{x}^{\mu} = \bar{\mathbf{P}} : \delta \bar{\mathbf{F}}, \quad (2.25)$$

Furthermore, the average of the microscopic deformation gradient is typically forced to be equal to the applied macroscopic deformation gradient, i.e.,

$$|\Omega^{\mu}|^{-1} \int_{\Omega^{\mu}} \mathbf{F} d\mathbf{x}^{\mu} = \bar{\mathbf{F}}. \quad (2.26)$$

It has been shown in, e.g., [119], that by prescribing specific boundary conditions for the fluctuation field  $\mathbf{w}$ , both conditions in Eqs. (2.25) and (2.26) are always fulfilled. Two frequently employed ones are the kinematic boundary conditions (KBC) and periodic boundary conditions (PBC) with,

$$\text{(KBC)} \quad \mathbf{w} = \mathbf{0} \quad \text{on } \partial\Omega^{\mu}, \quad (2.27)$$

$$\text{(PBC)} \quad \mathbf{w} \text{ periodic} \quad \text{on } \partial\Omega^{\mu}. \quad (2.28)$$

The corresponding function spaces are then

$$\text{(KBC)} \quad \mathcal{V} := \{\mathbf{v} \in (H^1(\Omega^{\mu}))^d \mid \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega^{\mu}\}, \quad (2.29)$$

$$\text{(PBC)} \quad \mathcal{V} := \{\mathbf{v} \in (H^1(\Omega^{\mu}))^d \mid \mathbf{v} \text{ periodic on } \partial\Omega^{\mu}\}, \quad (2.30)$$

or on the parent domain

$$\text{(KBC)} \quad \mathcal{V}^p := \{\mathbf{v} \in (H^1(\Omega^p))^d \mid \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega^p\}, \quad (2.31)$$

$$\text{(PBC)} \quad \mathcal{V}^p := \{\mathbf{v} \in (H^1(\Omega^p))^d \mid \mathbf{v} \text{ periodic on } \partial\Omega^p\}. \quad (2.32)$$

For most examples considered in this thesis, PBC are utilized, since they have shown empirically to provide a good estimate for effective quantities, while KBC often predict an overly stiff behavior [127, 62].

**Remark 2.2.1** *Periodic boundary conditions are typically enforced through two methods: Lagrange multipliers [88] or static condensation [71]. While the former method is easier to implement, it leads to a saddle point structure and a slightly larger system due to additional constraints that enforce the periodic boundary conditions. The latter method is more difficult to implement, but the resulting global stiffness matrix is symmetric positive definite. In this work, we employ static condensation to enforce periodic boundary conditions.*

### 2.2.3 Discretization and numerical solution

By following a standard Galerkin finite element discretization, the microscopic problem in Eqs. (2.11) and (2.12) can be solved. With  $\mathbf{w}^p \approx \mathbf{w}_h^p \in \mathcal{V}_h^p \subset \mathcal{V}^p$ , with  $\dim \mathcal{V}_h^p = \mathcal{N}$ , the number of degrees of freedom of the discretization, the internal force vector  $\mathbf{f} \in \mathbb{R}^{\mathcal{N}}$  and global stiffness matrix  $\mathbf{K} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$  can be derived from Eqs. (2.11) and (2.12), resulting in the following non-linear system of equations

$$\mathbf{f}(\mathbf{w}) = \mathbf{0}, \quad (2.33)$$

where  $\mathbf{w} \in \mathbb{R}^{\mathcal{N}}$  is the column matrix of unknown coefficients of the discretized fluctuation field. To find the solution  $\mathbf{w}$  of Eq. (2.33), the Newton method can be employed, cf. Eq. (2.5). Starting from a starting vector  $\mathbf{w}^0$  and Newton iteration number  $m = 0$ , compute for  $m = 1, \dots$ ,

$$\begin{aligned} \mathbf{K}(\mathbf{w}^m) \Delta \mathbf{w} &= -\mathbf{f}(\mathbf{w}^m), \\ \mathbf{w}^{m+1} &= \mathbf{w}^m + \Delta \mathbf{w}, \end{aligned} \quad (2.34)$$

until  $\|\mathbf{f}(\mathbf{w}^m)\|_2 \leq \varepsilon_{\text{newton}}$  with  $\varepsilon_{\text{newton}}$  a user-defined tolerance.

### 2.2.4 Computational costs

There are several aspects of the microscopic problem that make it computationally expensive:

1. Since typically, realistic microstructures contain fine geometrical features requiring fine meshes to capture their RVE geometry, the dimension of the solution space  $\mathcal{N}$  becomes large. The solution of the linear system of equations in Eq. (2.34) scales with  $\mathcal{O}(\mathcal{N}^2) - \mathcal{O}(\mathcal{N}^3)$ , making it computationally expensive for large  $\mathcal{N}$ .
2. In every Newton iteration, it is necessary to assemble the global force vector  $\mathbf{f}(\mathbf{w})$  and stiffness matrix  $\mathbf{K}(\mathbf{w})$ . This procedure scales linearly with the number of elements and Gauss points, thus, becoming expensive for fine meshes.

3. To compute the effective stiffness, it is necessary to solve several linear tangent problems in Eq. (2.20), which again scales with  $\mathcal{O}(\mathcal{N}^2) - \mathcal{O}(\mathcal{N}^3)$ .





## Chapter 3

# A non-intrusive reduced order model for first-order computational homogenization

The content of this chapter is based on the following publications:

- Guo, T., Rokoš, O., & Veroy, K. (2021). Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 384, 113924.
- Guo, T., Silva, F.A.B., Rokoš, O., & Veroy, K. (2022). Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method: Extension to geometrical parameterizations. *Computer Methods in Applied Mechanics and Engineering*, 401, 115636.

The two-scale problem arising in first-order computational homogenization, as presented in Chapter 2, is often discretized and solved using the finite element method (FEM) on both scales, leading to a nested FE scheme, also known as FE2. Although accurate for problems where scale separation of the micro- and macroscale can be assumed, the FE2 method tends to be computationally very expensive, since the microscopic problem needs to be solved at every quadrature point of the macroscopic problem. In multi-query contexts such as optimization or material design, it is therefore necessary to accelerate the solution of the microscopic problem.

As already discussed in Section 1.2.1, several non-intrusive methods have been developed to accelerate and accurately approximate the microscopic problem. However, such methods typically require large datasets and focus only on the effective quantities, while neglecting the microscopic physics, making the interpretation of the predictions more difficult. In this chapter, we derive a surrogate model for the microscopic problem that accounts for the microscopic physics with the following key features: (1) the evaluation of the effective quantities (stress and stiffness) is fast and accurate, (2) the stress field for all geometries can be recovered and automatically fulfill the microscopic balance laws and periodicity, (3) the method is non-intrusive, and (4) the sensitivities are available and can be readily used for optimization and material design.

To emphasize that a quantity depends on coordinates  $\boldsymbol{x}^P$  and/or input parameters  $(\bar{\boldsymbol{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ , the dependencies are stated explicitly in this chapter. To distinguish coordinates and parameters, a semicolon is employed, e.g.,  $P^P(\boldsymbol{x}^P; \bar{\boldsymbol{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ .

### 3.1 Surrogate modeling

To lower the computational cost of the microscopic problem, we approximate the microscopic problem using a data-driven non-intrusive reduced order model, constructed by proper orthogonal decomposition (POD) and Gaussian process regression (GPR). Similar methods were proposed in the literature to construct non-intrusive reduced order models for the displacement or strain field, see, e.g., [45, 64, 21, 129]. The proposed method differs from these works in that the non-intrusive reduced order model is constructed directly for the stress field  $\boldsymbol{P}$ . As seen in Fig. 2.1, both displacement and strain are not required by the macroscopic solver. If the displacement or strain was predicted, the stresses would still have to be calculated, which means that the microscopic constitutive law has to be implemented anew, negating the advantages of a non-intrusive method. Furthermore, depending on the material model the evaluation might be costly and can compromise the online speed-up significantly. It is also not

obvious how one would derive the effective stiffness for that case. On the other hand, reducing the stresses directly circumvents the above-mentioned disadvantages, and both the effective stress and stiffness can be rapidly obtained.

Another key difference here is that we consider geometrical parameters in our work. To treat different geometries, it is essential to define a parent domain and to transform all snapshots onto that domain. To find such transformations, we first present in this section a surface-based deformation method for finding such geometrical transformations by solving an auxiliary problem based on linear elasticity, which can be rapidly solved. Hereafter, we show how to construct the surrogate model and replace the microscopic simulation.

### 3.1.1 Transformation of snapshots

When simulating different geometries of the microstructure with independent simulation meshes, the obtained snapshots will be incompatible as the field variable is only known at different discrete points. To apply POD on such a set of snapshots, one would first need to interpolate each snapshot onto the same underlying grid of discrete points, leading to additional interpolation errors. To circumvent this, it is common to introduce a parent domain, and to transform each simulation snapshot onto the parent domain.

In general, such transformations are not known analytically, but obtained by solving an auxiliary problem. In the context of reduced order modeling, many existing works obtain geometrical transformations with space deformation techniques, such as free form deformation [120] or radial basis functions [101], see, e.g., [116, 115, 95, 83, 139, 25]. The biggest advantage of these methods is that the computational costs of computing the transformation map only scale with a pre-defined number of control points and is independent of the original simulation mesh, thus allowing for an efficient solution. However, numerous control points might be required to describe complicated geometries or to enforce certain parts of the domain to remain fixed. For our specific parameterized microscopic problem, as defined in Section 2.2, it is important to keep the outer boundaries fixed.

Another approach to obtain such transformations are surface-based deformation techniques, where an auxiliary diffusion-type PDE problem is posed and solved on the simulation mesh of the original problem. Hence, the transformation map has exactly the same flexibility as the solution field in the original simulation. Furthermore, one can ensure by construction that the obtained transformation is a bijective map. Lastly, enforcing certain parts to remain fixed is straightforward by prescribing conditions on these parts, making the surface-based deformation techniques more favorable for our use case. Below,

we directly employ zero Dirichlet boundary conditions to fix the RVE boundaries, since RVE size and topology will always remain fixed while the internal microstructure will be subject to change.

We propose to find the transformations by solving an auxiliary problem based on linear elasticity, which is closely related to techniques presented in [146, 59, 31, 16]. Even though the auxiliary problem formulated this way is more expensive to solve than the one arising from space deformation techniques, the solution can be substantially accelerated with standard reduced basis (RB) methods, as shown below.

The main idea is as follows: for any geometrical parameterization, the movement of certain parts of the geometry is known, while other parts are fixed. This movement can be prescribed onto a parent mesh and then an auxiliary problem solved to obtain the transformation map for all geometrical parameters. The example below illustrates this approach.

**2D RVE with elliptical inclusion** Consider an RVE defined on  $\Omega^\mu = [0, 1]^2$  consisting of a parameterized elliptical inclusion embedded in a homogeneous matrix, see Fig. 3.1. The major and minor axes of the inclusion are parameterized. Assuming a parent domain  $\Omega^p$  with a given circular inclusion, it can be deformed into any of the parameterized domains  $\Omega^\mu$ , by moving the points on the circular interface into the shape of the elliptical inclusion while keeping the outer boundaries fixed. By defining this transformation map as

$$\Phi_\mu : \Omega^p \rightarrow \Omega^\mu, \mathbf{x}^p \mapsto \mathbf{x}^\mu = \Phi_\mu(\mathbf{x}^p) = \mathbf{x}^p + \mathbf{d}(\mathbf{x}^p; \mu) \quad (3.1)$$

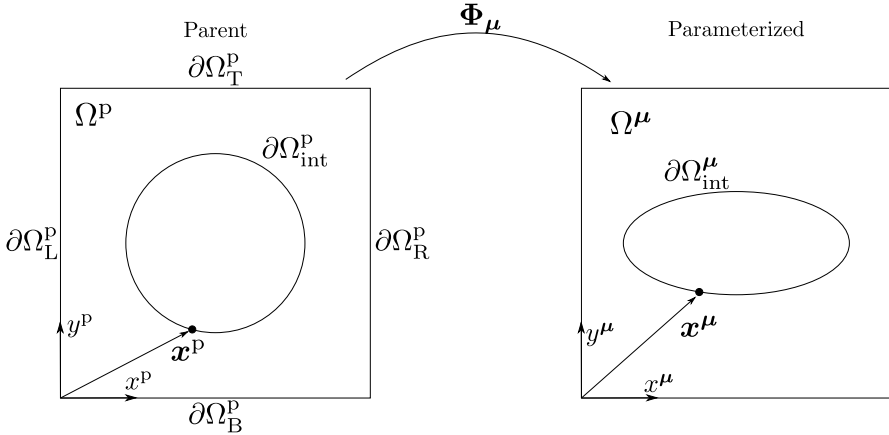
with  $\mathbf{d}$  the transformation displacement, this can now be translated into the following linear elasticity auxiliary problem:

$$\nabla_{\mathbf{x}^p} \cdot \left( \mathbf{c}^{\text{aux}} : \frac{1}{2} \left( \nabla_{\mathbf{x}^p} \mathbf{d} + (\nabla_{\mathbf{x}^p} \mathbf{d})^T \right) \right) = \mathbf{0} \quad \text{in } \Omega^p, \quad (3.2a)$$

$$\mathbf{d} = \mathbf{0} \quad \text{on } \partial\Omega^p, \quad (3.2b)$$

$$\mathbf{d} = \mathbf{x}^\mu(\mathbf{x}^p) - \mathbf{x}^p \quad \text{on } \partial\Omega_{\text{int}}^p, \quad (3.2c)$$

where  $\partial\Omega^p = \partial\Omega_L^p \cup \partial\Omega_R^p \cup \partial\Omega_B^p \cup \partial\Omega_T^p$  denotes the union of the left, right, bottom and top RVE boundaries, and  $\mathbf{x}^\mu(\mathbf{x}^p)$  is known for all points on the parent interface  $\partial\Omega_{\text{int}}^p$ . The elasticity tensor  $\mathbf{c}^{\text{aux}}$  is assumed to be constant throughout the whole domain, assumed in the form of Hooke's law, and specified by the Young's modulus  $E^{\text{aux}}$  and Poisson's ratio  $\xi^{\text{aux}}$ . Since the problem is cast in a purely geometric manner, the Young's modulus has no influence on the transformation map, as it only changes the magnitude of the stresses that are of no interest here. The Poisson's ratio  $\xi^{\text{aux}}$  changes the compressibility



**Figure 3.1:** Definition of parent and parameterized domain. The transformation map  $\Phi_\mu$  maps from the parent  $\Omega^P$  to the parameterized  $\Omega^\mu$  domain. The transformation displacement  $\mathbf{d}(\mathbf{x}^P; \mu)$  on the boundaries is fixed. The points on both the parent  $\partial\Omega_{int}^P$  and parameterized interface  $\partial\Omega_{int}^\mu$  are known and used to prescribe the transformation along the interface. As an example, the indicated point  $\mathbf{x}^P$  in the parent domain is displaced onto the indicated point  $\mathbf{x}^\mu$  in the parameterized domain.

of the material, hence affecting the transformation, and its influence on the approximation will be investigated in Section 3.3. The boundary condition in Eq. (3.2b) is chosen such that the parameterized domain  $\Phi_\mu(\Omega^P) = [0, 1]^2$  remains the same, i.e., covers the parent domain  $\Omega^P$ . This also means that the transformation preserves the volume, i.e.,  $|\Omega^P| = |\Phi_\mu(\Omega^P)|$ . Moreover, this way a periodic quantity remains periodic after transformation. Finally, Eq. (3.2c) prescribes the transformation displacements along the interface to deform the circle into an ellipse.

**Remark 3.1.1** *In principle, one could replace Eq. (3.2b) with periodic boundary conditions and fix the RVE at the corner points only to allow for more flexible transformations. However, the added complexity did not yield any significant improvement in accuracy in our test problems.*

Discretizing the auxiliary problem in Eq. (3.2) with FE yields a linear system of equations

$$\mathbf{A}\mathbf{d} = \mathbf{b}(\mu), \quad (3.3)$$

where the size of  $\mathbf{A}$ ,  $\mathbf{d}$  and  $\mathbf{b}(\mu)$  scale with the full mesh dimensionality  $\mathcal{N}$  and Eq. (3.3) needs to be solved for many right-hand sides (one for each geometry), where each solution is typically computationally expensive. However, there are several ways to accelerate the solution: since the stiffness matrix  $\mathbf{A}$  does not

depend on the parameters  $\boldsymbol{\mu}$  and is symmetric positive definite, for moderate values of  $\mathcal{N}$ , a Cholesky decomposition  $\mathbf{A} = \mathbf{L}\mathbf{L}^T$  can be computed once and then a forward and backward substitution can be used to solve Eq. (3.3) for any  $\mathbf{b}(\boldsymbol{\mu})$ . For even larger values of  $\mathcal{N}$ , the reduced basis method [103, 52] can be used: with a proper orthogonal decomposition (POD) a reduced basis for the transformation displacement can be found with  $\mathbf{d} = \mathbf{V}\hat{\mathbf{d}}$  with  $\mathbf{V} \in \mathbb{R}^{\mathcal{N} \times N}$  and  $\dim(\hat{\mathbf{d}}) = N \ll \mathcal{N}$  and the reduced system is obtained via a Galerkin projection, i.e.,

$$\begin{aligned} \mathbf{V}^T \mathbf{A} \mathbf{V} \hat{\mathbf{d}} &= \mathbf{V}^T \mathbf{b}(\boldsymbol{\mu}), \\ \Rightarrow \hat{\mathbf{A}} \hat{\mathbf{d}} &= \mathbf{V}^T \mathbf{b}(\boldsymbol{\mu}), \end{aligned} \quad (3.4)$$

where the reduced stiffness matrix  $\hat{\mathbf{A}} := \mathbf{V}^T \mathbf{A} \mathbf{V}$  only needs to be computed once. In Section 3.3, it will be shown that  $N$  is equal to the number of geometrical parameters  $\boldsymbol{\mu}$ . However, as pointed out in [126], the term on the right hand side  $\mathbf{V}^T \mathbf{b}(\boldsymbol{\mu})$  is in general not affinely decomposable, meaning that for each new value of  $\boldsymbol{\mu}$ , a matrix-vector product  $\mathbf{V}^T \mathbf{b}(\boldsymbol{\mu})$  must be computed which depends linearly on the full problem size  $\mathcal{O}(\mathcal{N})$ . If this is prohibitive, then an empirical interpolation method [5, 23] could be used to approximate  $\mathbf{b}(\boldsymbol{\mu})$  and then the complexity of the solution of the auxiliary problem becomes independent of  $\mathcal{N}$ .

After solving Eq. (3.3) for a given  $\boldsymbol{\mu}$ ,  $\mathbf{d}(\mathbf{x}^p; \boldsymbol{\mu})$  is known at every grid point of the parent mesh and hence the map  $\Phi_{\boldsymbol{\mu}}$  is obtained. To transform the original snapshots  $\mathbf{P}(\mathbf{x}^{\mu}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ , computed on the parameterized domain  $\Omega^{\mu}$ , onto the parent domain  $\mathbf{P}^p(\mathbf{x}^p; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ , they have to be evaluated at the transformed parent coordinates, i.e.,

$$\mathbf{P}^p(\mathbf{x}^p; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := \mathbf{P}(\Phi_{\boldsymbol{\mu}}(\mathbf{x}^p); \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{P}(\mathbf{x}^{\mu}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}), \quad \forall \mathbf{x}^p \in \Omega^p. \quad (3.5)$$

**Remark 3.1.2** *If the original snapshot  $\mathbf{P}(\mathbf{x}^{\mu}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$  is obtained on an independent mesh, one needs to interpolate it onto the transformed parent coordinates  $\Phi_{\boldsymbol{\mu}}(\mathbf{x}^p)$ . However, this interpolation can introduce non-physical effects, i.e., periodicity and/or balance of linear momentum might not be fulfilled anymore. To ensure a physical transformation and avoid interpolation errors, we solve the auxiliary problem in Eq. (3.2) to generate simulation meshes for each geometry  $\Omega^{\mu}$  (by adding the transformation displacement  $\mathbf{d}(\mathbf{x}^p; \boldsymbol{\mu})$  to each node of  $\Omega^p$ ) and solve the physical simulation in Eq. (2.9). Equivalently, one could compute the transformation gradient  $\mathbf{F}_{\boldsymbol{\mu}}$  and solve the physical simulation directly on the parent domain using Eqs. (2.11) and (2.12).*

### 3.1.2 Proper orthogonal decomposition

After solving the microscopic problem for different input parameter combinations  $\{\bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}\}_{i=1}^{N_{\text{pod}}}$ , stress snapshots  $\{\mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})\}_{i=1}^{N_{\text{pod}}}$  are obtained on the parent domain that each fulfill the balance of linear momentum in Eq. (2.11). Here,  $N_{\text{pod}}$  denotes the number of snapshots. Each stress snapshot is typically defined on the quadrature points and  $\mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in L^2(\Omega^{\text{P}})$ , with  $L^2(\Omega^{\text{P}})$  the space of square integrable functions on  $\Omega^{\text{P}}$ .

With the available snapshots, the goal is to learn an effective constitutive law, i.e.,  $(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mapsto (\bar{\mathbf{P}}, \bar{\boldsymbol{\alpha}})$ , that is not only fast to evaluate and accurate, but it is also possible to recover the microscopic stress fields. Furthermore, the predicted fields should always fulfill the microscopic balance laws and periodicity. To achieve that, first the weak form in Eq. (2.11) is rewritten with,

$$\begin{aligned} & \int_{\Omega^{\text{P}}} ((\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}} = 0, \\ \Leftrightarrow & \int_{\Omega^{\text{P}}} (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) : (\mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{F}_{\boldsymbol{\mu}}^{-T} |\det \mathbf{F}_{\boldsymbol{\mu}}|) d\mathbf{x}^{\text{P}} = 0, \quad (3.6) \\ \Leftrightarrow & \int_{\Omega^{\text{P}}} (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) : \mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) d\mathbf{x}^{\text{P}} = 0, \end{aligned}$$

where the weighted stress  $\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := \mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{F}_{\boldsymbol{\mu}}^{-T} |\det \mathbf{F}_{\boldsymbol{\mu}}|$  is defined. By employing a proper orthogonal decomposition (POD) on the available weighted stress snapshots (obtained by multiplying the stress snapshots with  $\mathbf{F}_{\boldsymbol{\mu}^{(i)}}^{-T} |\det \mathbf{F}_{\boldsymbol{\mu}^{(i)}}|$ ), the weighted stress is approximated as a linear combination of constant global basis functions  $\mathbf{B}_l(\mathbf{x}^{\text{P}}) \in L^2(\Omega^{\text{P}})$  with parameter-dependent coefficients  $\alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}$ ,

$$\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \approx \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{B}_l(\mathbf{x}^{\text{P}}), \quad (3.7)$$

where  $L$  is the number of basis functions. The basis functions  $\mathbf{B}_l$  are computed as linear combinations of the  $N_{\text{pod}}$  training snapshots, i.e.,

$$\mathbf{B}_l(\mathbf{x}^{\text{P}}) = \sum_{i=1}^{N_{\text{pod}}} c_{il} \mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}), \quad (3.8)$$

where  $c_{il}$  are coefficients determined by POD. The algorithm employed to compute POD is provided in Appendix B. Additionally, the basis functions are orthonormal with respect to  $L^2(\Omega^{\text{P}})$ , i.e.,

$$(\mathbf{B}_i, \mathbf{B}_j)_{L^2(\Omega^{\text{P}})} := \int_{\Omega^{\text{P}}} \mathbf{B}_i(\mathbf{x}^{\text{P}}) : \mathbf{B}_j(\mathbf{x}^{\text{P}}) d\mathbf{x}^{\text{P}} = \delta_{ij}, \quad (3.9)$$

and periodic, as they are computed from a linear combination of periodic weighted stress fields. Inserting Eqs. (3.7) and (3.8) in Eq. (3.6) yields,

$$\begin{aligned}
& \int_{\Omega^{\text{P}}} (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) : \mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) d\mathbf{x}^{\text{P}} \\
& \approx \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \int_{\Omega^{\text{P}}} (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) : \mathbf{B}_l(\mathbf{x}^{\text{P}}) d\mathbf{x}^{\text{P}} \\
& = \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \sum_{i=1}^{N_{\text{pod}}} c_{il} \underbrace{\int_{\Omega^{\text{P}}} (\nabla_{\mathbf{x}^{\text{P}}} \delta \mathbf{w}^{\text{P}}) : \mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}) d\mathbf{x}^{\text{P}}}_{=0, \text{ since each snapshot fulfills Eq. (3.6)} = 0,
\end{aligned} \tag{3.10}$$

and shows that the balance equation will always be fulfilled with the POD basis. This means that, independent of the coefficients  $\alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ , the resulting field will always fulfill the balance of linear momentum in Eq. (2.11).

**Remark 3.1.3** *As we do not have access to the microscopic displacement nor the microscopic deformation gradient, it is not possible to verify the Hill-Mandel condition in Eq. (2.25) or that the averaged microscopic deformation gradient is equal to the applied macroscopic deformation gradient, see Eq. (2.26).*

**Effective quantities** Assuming for now that we have a way of finding the coefficients  $\alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ , the effective stress can be computed from the approximated weighted stress in Eq. (3.7). In Eq. (2.16), the effective stress is defined as

$$\bar{\mathbf{P}}(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}}. \tag{3.11}$$

By rearranging the definition of the weighted stress with

$$\begin{aligned}
\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{F}_{\boldsymbol{\mu}}^{-T} |\det \mathbf{F}_{\boldsymbol{\mu}}|, \\
\Leftrightarrow \mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= \mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{F}_{\boldsymbol{\mu}}^T |\det \mathbf{F}_{\boldsymbol{\mu}}|^{-1},
\end{aligned} \tag{3.12}$$

and inserting together with Eq. (3.7) in Eq. (3.11) yields

$$\begin{aligned}
\bar{\mathbf{P}}(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) &= |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{F}_{\boldsymbol{\mu}}^T d\mathbf{x}^{\text{P}} \\
&\approx |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \left( \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{B}_l(\mathbf{x}^{\text{P}}) \right) \mathbf{F}_{\boldsymbol{\mu}}^T d\mathbf{x}^{\text{P}} \\
&= |\Omega^{\text{P}}|^{-1} \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \int_{\Omega^{\text{P}}} \mathbf{B}_l(\mathbf{x}^{\text{P}}) \mathbf{F}_{\boldsymbol{\mu}}^T d\mathbf{x}^{\text{P}}.
\end{aligned} \tag{3.13}$$



In order to have a rapid online phase, the integrals in Eq. (3.13) must be precomputed in the offline stage. Due to the specific form of the basis functions  $\mathbf{B}_l$ , Eq. (3.8), it can be shown that the integrals in Eq. (3.13) are invariant with respect to  $\mathbf{F}_\mu^T$ , i.e.,

$$\int_{\Omega^P} \mathbf{B}_l(\mathbf{x}^P) \mathbf{F}_\mu^T d\mathbf{x}^P = \int_{\Omega^P} \mathbf{B}_l(\mathbf{x}^P) d\mathbf{x}^P, \quad (3.14)$$

which can easily be precomputed since  $\mathbf{B}_l$  is known; the proof of the identity in Eq. (3.14) is provided in Appendix C. With

$$\bar{\mathbf{B}}_l := |\Omega^P|^{-1} \int_{\Omega^P} \mathbf{B}_l(\mathbf{x}^P) d\mathbf{x}^P, \quad (3.15)$$

the expression for the effective stress in Eq. (3.13) becomes

$$\bar{\mathbf{P}}(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \bar{\mathbf{B}}_l, \quad (3.16)$$

Interestingly, the prediction of the effective stresses is completely independent of the auxiliary problem. The auxiliary problem only needs to be computed in order to recover the stress field from the predicted weighted stress field.

The effective constitutive stiffness and sensitivities with respect to material and geometrical parameters are given as

$$\bar{\mathfrak{A}} = \nabla_{\bar{\mathbf{F}}} \bar{\mathbf{P}} = \sum_{l=1}^L \bar{\mathbf{B}}_l \otimes \nabla_{\bar{\mathbf{F}}} \alpha_l, \quad (3.17)$$

$$\nabla_{\boldsymbol{\lambda}} \bar{\mathbf{P}} = \sum_{l=1}^L \bar{\mathbf{B}}_l \otimes \nabla_{\boldsymbol{\lambda}} \alpha_l, \quad (3.18)$$

$$\nabla_{\boldsymbol{\mu}} \bar{\mathbf{P}} = \sum_{l=1}^L \bar{\mathbf{B}}_l \otimes \nabla_{\boldsymbol{\mu}} \alpha_l. \quad (3.19)$$

**Remark 3.1.4** *In general, the effective stiffness in Eq. (3.17) does not fulfill major symmetries, i.e.,  $\bar{\mathfrak{A}}_{ijkl} = \bar{\mathfrak{A}}_{klij}$ . This means that the global stiffness matrix of the macroscopic solver will be non-symmetric, which is a disadvantage of the proposed method. In our considered two-scale examples, this did not cause any problems.*

### 3.1.3 Regression model

From Eqs. (3.16)–(3.19), it is clear that all effective quantities can be rapidly predicted, when coefficients  $\alpha_l$  and their derivatives are known. These coefficients can be approximated with regression models. However, the choice of

the regression model depends on the type of microscopic material model. For history-dependent material behavior, such as plasticity, the same input variables might lead to different stress values (and, hence, coefficients), depending on the previous state. To approximate such coefficients, a regression model that contains internal state variables, such as recurrent neural networks, are required. In this chapter, we focus on the hyperelastic and therefore history-independent class of materials. These materials are characterized by a (poly-)convex strain energy density function. Two hyperelastic material models are provided in Appendix A.1. Due to the convexity, each input corresponds to exactly one stress and coefficient, and a wide range of regression techniques can be used, such as radial basis functions [44], GPR [45, 64, 144] or neural networks [53]. A comparison of different regression models in approximating the coefficients was carried out in [10], revealing that GPRs can be as accurate as NNs, while being easier to train. Apart from that, GPR offers some other advantages:

- It can reconstruct the training data perfectly, i.e., it reproduces the exact solution at the training points.
- Depending on the chosen kernel, the regression function has a specified smoothness and its derivatives can be obtained by closed-form expressions.
- The trained GPR model returns an uncertainty measure for each evaluation, which can be used to estimate the regression error or to develop an active learning scheme [76, 45, 64, 144].

Due to these advantages, in this work, we learn regression models for  $\alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$  with GPRs.

**Data preparation** For the regression, the training data that have been collected for POD can be reused. However, as will be shown in Section 3.2, more training data is in general required for an accurate regression model than for an accurate POD basis. For that reason, it is assumed that we have  $N_{\text{reg}} \geq N_{\text{pod}}$  weighted stress snapshots  $\mathbf{W}^{\text{P},(i)}$  that are generated for parameters  $(\bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})$  with  $i = 1, 2, \dots, N_{\text{reg}}$ . The optimal projection (in  $L^2(\Omega^{\text{P}})$  sense) of the  $i$ -th snapshot on the POD basis is given by

$$\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}) \approx \sum_{l=1}^L (\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}), \mathbf{B}_l)_{L^2(\Omega^{\text{P}})} \mathbf{B}_l, \quad (3.20)$$

and hence  $\alpha_l^{(i)} = (\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}), \mathbf{B}_l)_{L^2(\Omega^{\text{P}})}$  for  $l = 1, \dots, L$ . All coefficients for all snapshots  $\alpha_l^{(i)}$  are collected together with the input data. Subsequently, a mapping between the parameters and each coefficient is constructed

independently with GPR, i.e.,

$$\hat{\alpha}_l : \mathbb{P} \rightarrow \mathbb{R}, (\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \stackrel{\text{GPR}}{\mapsto} \hat{\alpha}_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \approx \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \quad (3.21)$$

**Gaussian process regression** For ease of notation, we assume for this section that all inputs  $(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{P}$  are concatenated and written as one vector of parameters  $\mathbf{X} \in \hat{\mathbb{P}}$  with  $\hat{\mathbb{P}}$  the flattened parameter space.

Each coefficient  $\hat{\alpha}_l(\mathbf{X})$  is assumed to be distributed as a Gaussian process (GP) with a zero mean function and a symmetric semi-positive definite kernel  $k_\theta : \hat{\mathbb{P}} \times \hat{\mathbb{P}} \rightarrow \mathbb{R}^+$ ,  $(\mathbf{X}, \mathbf{X}') \mapsto k_\theta(\mathbf{X}, \mathbf{X}')$ ,

$$\hat{\alpha}_l \sim \mathcal{GP}(0, k_\theta(\mathbf{X}, \mathbf{X}')). \quad (3.22)$$

Here,  $\mathbb{R}^+$  contains all positive real numbers and 0. The form of the kernel  $k_\theta(\mathbf{X}, \mathbf{X}')$  is chosen by the user, and each kernel has hyperparameters  $\boldsymbol{\theta}$  that are fitted to the training data. In this work, we use, same as in [45], the automatic relevance determination (ARD) squared exponential kernel,

$$k_\theta(\mathbf{X}, \mathbf{X}') = \sigma_f^2 \exp\left(-\frac{1}{2} \sum_{k=1}^{N_{\text{in}}} \frac{(X_k - X'_k)^2}{l_k^2}\right), \quad (3.23)$$

where  $\boldsymbol{\theta} = [\sigma_f, l_1, l_2, \dots, l_{N_{\text{in}}}]$  and  $N_{\text{in}} = \dim \hat{\mathbb{P}}$  is the total number of input parameters.

By Bayesian inference and conditioning on the available training data  $\{\mathbf{X}^{(i)}, \alpha_l(\mathbf{X}^{(i)})\}_{i=1}^{N_{\text{reg}}}$ , here compactly written as  $\mathbf{X}_{\text{tr}} := [\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(N_{\text{reg}})}]^T$  and  $\mathbf{y}_{\text{tr}} := [\alpha_l(\mathbf{X}^{(1)}), \dots, \alpha_l(\mathbf{X}^{(N_{\text{reg}})})]^T$ , the posterior GP  $\hat{\alpha}_l^*$  at a testing point  $\mathbf{x}$  is given by

$$\hat{\alpha}_l^*(\mathbf{x}) | (\mathbf{X}_{\text{tr}}, \mathbf{y}_{\text{tr}}) \sim \mathcal{GP}(m^*(\mathbf{x}), k^*(\mathbf{x}, \mathbf{x}')), \quad (3.24)$$

$$m^*(\mathbf{x}) = \mathbf{y}_{\text{tr}}^T \mathbf{K}_{\text{tr}}^{-1} \mathbf{k}_\theta(\mathbf{X}_{\text{tr}}, \mathbf{x}), \quad (3.25)$$

$$k^*(\mathbf{x}, \mathbf{x}') = k_\theta(\mathbf{x}, \mathbf{x}') - \mathbf{k}_\theta(\mathbf{X}_{\text{tr}}, \mathbf{x})^T \mathbf{K}_{\text{tr}}^{-1} \mathbf{k}_\theta(\mathbf{X}_{\text{tr}}, \mathbf{x}'), \quad (3.26)$$

where  $\mathbf{K}_{\text{tr}} := [k_\theta(\mathbf{X}^{(i)}, \mathbf{X}^{(j)})]_{i,j=1}^{N_{\text{reg}}} \in \mathbb{R}^{N_{\text{reg}} \times N_{\text{reg}}}$  is a matrix that can be pre-computed. The term  $\mathbf{k}_\theta(\mathbf{X}_{\text{tr}}, \mathbf{x}) := [k_\theta(\mathbf{X}^{(1)}, \mathbf{x}), \dots, k_\theta(\mathbf{X}^{(N_{\text{reg}})}, \mathbf{x})]^T$  needs to be evaluated during the online phase. This way, for each testing input  $\mathbf{x}$  a Gaussian distribution  $\mathcal{N}(m^*(\mathbf{x}), k^*(\mathbf{x}, \mathbf{x}))$  for the coefficient  $\hat{\alpha}_l^*$  can be obtained. As the ARD kernel is infinitely smooth, the derivatives of  $\hat{\alpha}_l^*$  with respect to the inputs exist and can be obtained through closed-form expressions. The optimal hyperparameters  $\boldsymbol{\theta}$  are determined with a maximum likelihood estimation, as presented in [110]. More information and a broad overview on GPRs can be found in [110].

After training a GPR for each coefficient independently, yielding in total  $L$  GPR models, the surrogate model for the microscopic simulation is complete. Given a testing input, the coefficients and their derivatives can be inferred, and together with Eqs. (3.16)–(3.19), the effective quantities can be computed. The proposed method will be referred to as PODGPR in the following.

**Remark 3.1.5** *The macroscopic balance of angular momentum states that*

$$\overline{\mathbf{P}}\overline{\mathbf{F}}^T = \overline{\mathbf{F}}\overline{\mathbf{P}}^T \quad (3.27)$$

*has to be fulfilled for the effective 1PK stress  $\overline{\mathbf{P}}$ . Inserting Eq. (3.16) in Eq. (3.27) yields*

$$\sum_{l=1}^L \alpha_l \overline{\mathbf{B}}_l \overline{\mathbf{F}}^T = \sum_{l=1}^L \alpha_l \overline{\mathbf{F}} \overline{\mathbf{B}}_l^T, \quad (3.28)$$

*which can be expanded into four scalar equations in 2D (9 equations in 3D),*

$$\sum_{l=1}^L \alpha_l (\overline{B}_{l,11} \overline{F}_{11} + \overline{B}_{l,12} \overline{F}_{12}) = \sum_{l=1}^L \alpha_l (\overline{F}_{11} \overline{B}_{l,11} + \overline{F}_{12} \overline{B}_{l,12}), \quad (3.29)$$

$$\sum_{l=1}^L \alpha_l (\overline{B}_{l,11} \overline{F}_{21} + \overline{B}_{l,12} \overline{F}_{22}) = \sum_{l=1}^L \alpha_l (\overline{F}_{11} \overline{B}_{l,21} + \overline{F}_{12} \overline{B}_{l,22}), \quad (3.30)$$

$$\sum_{l=1}^L \alpha_l (\overline{B}_{l,21} \overline{F}_{11} + \overline{B}_{l,22} \overline{F}_{12}) = \sum_{l=1}^L \alpha_l (\overline{F}_{21} \overline{B}_{l,11} + \overline{F}_{22} \overline{B}_{l,12}), \quad (3.31)$$

$$\sum_{l=1}^L \alpha_l (\overline{B}_{l,21} \overline{F}_{21} + \overline{B}_{l,22} \overline{F}_{22}) = \sum_{l=1}^L \alpha_l (\overline{F}_{21} \overline{B}_{l,21} + \overline{F}_{22} \overline{B}_{l,22}), \quad (3.32)$$

*that constrain the values that  $\{\alpha_l\}_{l=1}^L$  can take. It is clear that Eqs. (3.29) and (3.32) are automatically fulfilled because individual terms on both sides are equal, and that Eqs. (3.30) and (3.31) represent the same equation. Therefore, the only equation that constrains the values of  $\{\alpha_l\}_{l=1}^L$  can be written as*

$$\sum_{l=1}^L \alpha_l (\overline{B}_{l,11} \overline{F}_{21} + \overline{B}_{l,12} \overline{F}_{22} - \overline{F}_{11} \overline{B}_{l,21} - \overline{F}_{12} \overline{B}_{l,22}) = 0. \quad (3.33)$$

*Given a deformation gradient  $\overline{\mathbf{F}}$ , and by defining*

$$\gamma_l(\overline{\mathbf{F}}) := \overline{B}_{l,11} \overline{F}_{21} + \overline{B}_{l,12} \overline{F}_{22} - \overline{F}_{11} \overline{B}_{l,21} - \overline{F}_{12} \overline{B}_{l,22} \quad (3.34)$$

for  $l = 1, \dots, L$ , one obtains the following orthogonality constraint that has to be fulfilled

$$\boldsymbol{\alpha} \perp \boldsymbol{\gamma}, \quad (3.35)$$

where  $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_L]^T$  and  $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_L]^T$ . For the 3D case, three orthogonality constraints can be derived analogously.

When using GPRs to predict the coefficients  $\hat{\boldsymbol{\alpha}}^*$ , these will in general not fulfill Eq. (3.35). One can correct the coefficients by taking only the orthogonal part, i.e.,

$$\boldsymbol{\alpha}^\perp = \hat{\boldsymbol{\alpha}}^* - \frac{\hat{\boldsymbol{\alpha}}^* \cdot \boldsymbol{\gamma}}{\boldsymbol{\gamma} \cdot \boldsymbol{\gamma}} \boldsymbol{\gamma}. \quad (3.36)$$

Using  $\boldsymbol{\alpha}^\perp$  in Eq. (3.16), the balance of angular momentum in Eq. (3.27) is fulfilled. The effective sensitivities defined in Eqs. (3.17)–(3.19) also need to be corrected, and the equations are provided in Appendix D. In our numerical examples, we predicted the effective stress with and without the correction and obtained negligible differences.

**Remark 3.1.6** The term  $k^*(\mathbf{x}, \mathbf{x})$  in Eq. (3.26) acts as an uncertainty measure for the prediction. It can be used as an indicator of the regression error or to develop an adaptive sampling scheme. In this work, we do not utilize the uncertainty measure, and it is therefore sufficient to evaluate only the mean function to predict the coefficient  $\hat{\alpha}_i^*(\mathbf{x}) = m^*(\mathbf{x})$ .

**Remark 3.1.7** As the matrix  $\mathbf{K}_{\text{tr}}$  must be inverted for the prediction, the computational cost of the online evaluation scales with the number of samples via  $\mathcal{O}(N_{\text{reg}}^3)$ . For large numbers of training data, the prediction can become prohibitively expensive. For such cases, methods such as sparse GPs [105] have been proposed.

**Remark 3.1.8** For all GPR models, the Python library GPy [42] was used. To solve the maximum likelihood estimation, the L-BFGS-B algorithm was used. In the numerical tests, we did not encounter any problems during hyperparameter tuning and the convergence of the L-BFGS-B algorithm was smooth.

### 3.1.4 Comparison to neural networks

After the pioneering works by Ghaboussi et al. [38, 39] and recent advances in deep learning, many papers have used methods of deep learning to extract a constitutive model from pairs of stress and deformation data by training a

deep neural network (DNN), e.g. [56, 73, 40, 94, 142]. Frequently, their data is generated by solving the microscopic simulation for different input parameters, analogous to what is done in this work. However, one difference is that in this work the (weighted) stress field is collected while the other works only collect the averaged stress and dispose of the stress field. A brief comparison between both approaches is given below:

1. **Training:** The DNN approach essentially performs a regression directly on the stress and deformation data. This means that a mapping from  $\mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^{3 \times 3}$  (or  $\mathbb{R}^6 \rightarrow \mathbb{R}^6$  in the case where the stress and deformation measures are symmetric) has to be learned. On the other hand, the proposed method PODGPR first compresses the microscopic stress field solutions into a few independent coefficients which are then learned, i.e., regression maps from  $\mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}$  or  $\mathbb{R}^6 \rightarrow \mathbb{R}$  are learned.
2. **Implementation:** Both surrogate models, after they have been trained, can be easily adopted into any simulation software as they are both non-intrusive and therefore entirely independent of the software used to solve the microscale simulation.
3. **Evaluation:** The DNN approach needs to compute one forward pass through the neural network to get the effective stress for a given deformation. The effective stiffness can then be computed with one backward pass with automatic differentiation. However, the microstructural stresses, which are desirable for revealing local stress concentrations and designing improved microstructures, cannot be obtained. In PODGPR, the effective stress and stiffness can be computed with Eqs. (3.16) and (3.17), but in addition the (weighted) stress field can also be fully recovered with Eq. (3.7).

### 3.1.5 Offline–online decomposition

For convenience, the offline–online decomposition is summarized in Algorithm 1.

---

**Algorithm 1** Offline–online decomposition of the proposed PODGPR framework with microstructures parameterized with external loading  $\bar{\mathbf{F}}$ , microstructural material parameters  $\boldsymbol{\lambda}$  and geometrical features  $\boldsymbol{\mu}$ .

---

**Offline Stage:**

- 1: Define a parent domain  $\Omega^{\text{P}}$  and its finite element discretization.
- 2: Generate parameter samples  $\{\bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}\}_{i=1}^{N_{\text{reg}}}$  from a random distribution on  $\mathbb{P}$ .
- 3: For each different set of geometrical parameters  $\boldsymbol{\mu}^{(i)}$ , solve the auxiliary problem in Eq. (3.2) to obtain the transformation map  $\Phi_{\boldsymbol{\mu}^{(i)}}$ .
- 4: Run full RVE simulations to obtain stress snapshots  $\mathbf{P}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})$ .
- 5: Compute POD of weighted stress  $\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})$  on the parent domain, cf. Eq. (3.7).
- 6: Project weighted stress snapshots onto POD basis and learn GPRs for the POD coefficients.
- 7: Compute  $\bar{\mathbf{B}}_l$  using Eq. (3.15).

**Online Stage:**

- 1: Given a new parameter set  $(\bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ , evaluate  $\hat{\alpha}_l^*$  with trained GPRs.
  - 2: Compute effective quantities with Eqs. (3.16)–(3.19).
- 

## 3.2 Numerical examples for fixed geometry

In this section, the performance of PODGPR is demonstrated for a fixed geometry, i.e., there is only a single geometrical parameters  $\boldsymbol{\mu}$ ,  $\mathbf{F}_{\boldsymbol{\mu}} = \mathbf{I}$  and transformations  $\Phi_{\boldsymbol{\mu}}$  are identity maps, and the weighted stress  $\mathbf{W}^{\text{P}}$  is equivalent to the stress  $\mathbf{P}^{\text{P}}$ . To illustrate the influence of the number of basis functions  $L$ , number of samples used for the POD basis construction  $N_{\text{pod}}$ , and number of samples used for the regression  $N_{\text{reg}}$ , two single-scale examples with different 2D RVEs are presented. The third example involves a two-scale problem, in which the results obtained with PODGPR are compared with the full FE2 solution.

All numerical examples are defined in 2D under plane strain conditions, although the proposed methodology can easily be extended to 3D microstructures. All quantities are normalized and considered dimensionless. The RVEs are assumed to be of size  $[0, 1]^2$ . The polar decomposition (cf. Remark 2.1.1) is employed to reduce the number of load parameters, and the surrogate model is constructed for the stretch tensor  $\bar{\mathbf{U}}$ . All microscopic simulations are conducted with the FE framework MOOSE [99].

To measure the accuracy of PODGPR on test data, the following relative

error measure for the effective stress is defined:

$$\varepsilon_{\bar{\mathbf{P}}} := \frac{\|\bar{\mathbf{P}}^{\text{truth}} - \bar{\mathbf{P}}^{\text{PODGPR}}\|_F}{\|\bar{\mathbf{P}}^{\text{truth}}\|_F}, \quad (3.37)$$

where  $\|\bullet\|_F$  denotes the Frobenius norm,  $\bar{\mathbf{P}}^{\text{truth}}$  the effective stress from the full FE simulation and  $\bar{\mathbf{P}}^{\text{PODGPR}}$  the effective stress resulting from PODGPR.

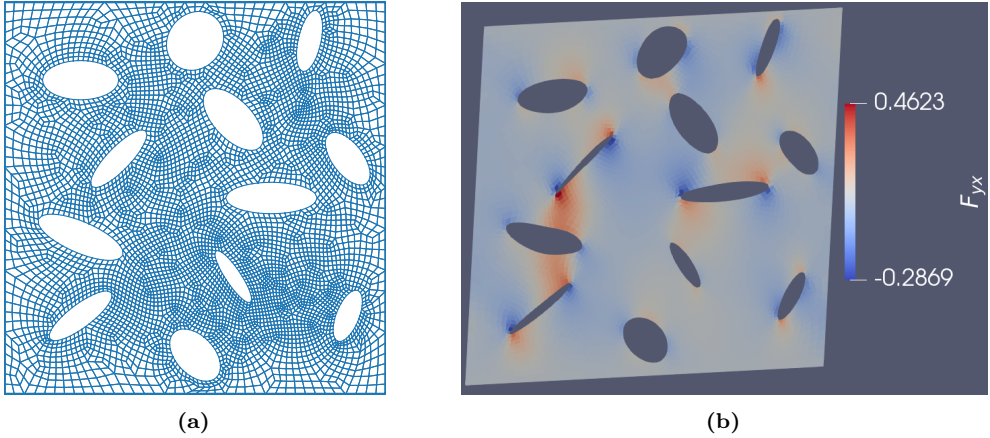
For comparison, deep feedforward neural networks with different architectures are also trained with the available deformation and stress data collected from the RVE simulations. A brief introduction to feedforward neural networks is provided in Appendix E. Regarding the network architecture, different neural networks with  $N_h = 1$  or 2 hidden layers with each  $N_n = 20$  or 50 neurons are tested. The input and output layers have  $N_{\text{in}}$  and 4 neurons. Apart from the output layer, an ELU activation function is applied after each layer. Before training, the inputs and stress data are normalized to  $[0, 1]$ . All networks are trained with a mean squared error loss function, the Adam optimizer with a learning rate of  $1 \times 10^{-4}$  and a batch size of 32 for 10000 epochs. The training is performed with the Python package PyTorch [98].

### 3.2.1 Porous material

The porous microstructure depicted in Fig. 3.2a is considered for the first example, where the pores account for 14% of the total area. Four-node quadrilateral elements with four quadrature points are employed for the FE discretization. The matrix material is modeled as a Neo-Hookean material with material parameters  $C_1 = 1$  and  $D_1 = 1$ . The definition of the material model is provided in Appendix A.1.1. Kinematic boundary conditions for the microscopic fluctuation field ( $\mathbf{w}^p = \mathbf{0}$  on the boundaries) and macroscopic stretches of  $\bar{U}_{xx} - 1, \bar{U}_{yy} - 1, \bar{U}_{xy} \in [-0.05, 0.05]$  are considered. Due to the geometry of the problem, such external deformations lead to much higher local strains, see Fig. 3.2b for an exemplary deformation for  $[\bar{U}_{xx}, \bar{U}_{yy}, \bar{U}_{xy}] = [0.95, 0.95, 0.05]$ . If larger macroscopic loads were considered, some pores might close, requiring contact detection, which is outside the scope of the developed methodology. In this example, material parameters  $\boldsymbol{\lambda}$  are fixed, hence this problem has  $N_{\text{in}} = 3$  varying parameters,  $\bar{U}_{xx}, \bar{U}_{yy}, \bar{U}_{xy}$ .

**Data generation** To investigate the number of pre-computations needed for an accurate representation, a set of 500 training snapshots for training PODGPR is sampled via a Sobol sequence sampling procedure, and another set of 1000 test snapshots is generated randomly from a uniform distribution to evaluate the accuracy of the surrogate model. It was observed in [11] that



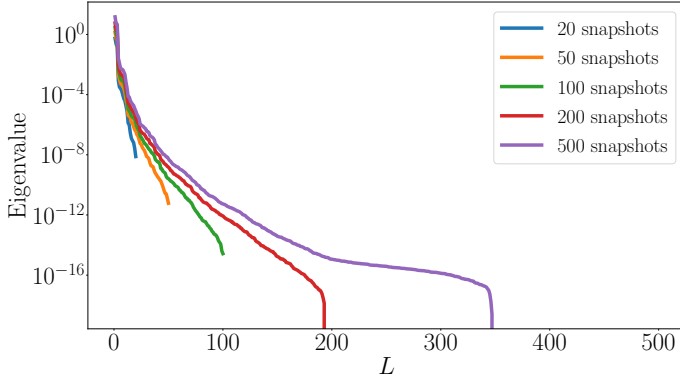


**Figure 3.2:** a) The considered porous microstructure consists of a Neo-Hookean matrix material with a volume ratio of 86%. The mesh consists of 6000 bilinear elements and 6293 nodes. b) An exemplary local deformation gradient  $F_{yx}$  with  $[\bar{U}_{xx}, \bar{U}_{yy}, \bar{U}_{xy}] = [0.95, 0.95, 0.05]$ . The  $yx$ -component ranges from  $[-0.2869, 0.4623]$  and is hence much larger than the prescribed macroscopic deformation.

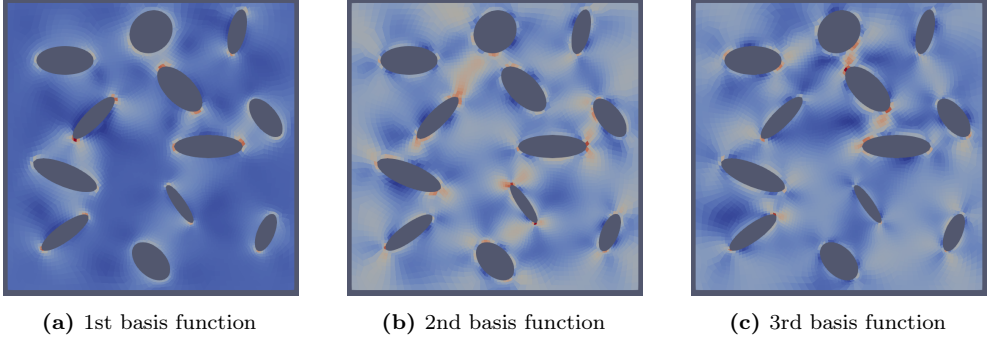
Sobol sequence sampling fills the parameter space more evenly as compared to random sampling and leads to better results.

**Eigenvalues** The eigenvalues of the correlation matrix of the stress field (cf. Eq. (B.2)) for different numbers of training snapshots  $N_{\text{pod}}$  are plotted in Fig. 3.3. For all cases, a quick decay is observed, indicating the reducibility of the problem. The magnitude of the first three POD basis functions is plotted in Fig. 3.4. It can be seen that the basis functions specifically capture the stress concentrations around the pores. Subsequently,  $L = 20$  basis functions are considered which corresponds to an energy  $\mathcal{E}_{\text{pod}}$  of 99.9996% (see Eq. (B.4) for the definition).

**Influence of  $N_{\text{pod}}$  and  $N_{\text{reg}}$**  For  $L = 20$  basis functions, combinations of  $N_{\text{pod}} \in \{20, 50, 100, 200, 500\}$  and  $N_{\text{reg}} \in \{50, 100, 200, 300, 400, 500\}$  are used for PODGPR. All 1000 test data are evaluated and the resulting mean and maximum error of the stresses are plotted in Fig. 3.5. It can be observed that for all cases the mean error is below 0.1%. From  $N_{\text{pod}} = 20$  to  $N_{\text{pod}} = 50$ , a significant improvement in the error can be observed. However, for  $N_{\text{pod}} > 50$ , the error only changes marginally. As  $N_{\text{reg}}$  is increased, more data becomes available for the regression and therefore the mapping in Eq. (3.21) becomes increasingly more accurate. Nevertheless, even using only 50 snapshots for both basis and regression yields a mean error of less than 0.1% and a maximum error



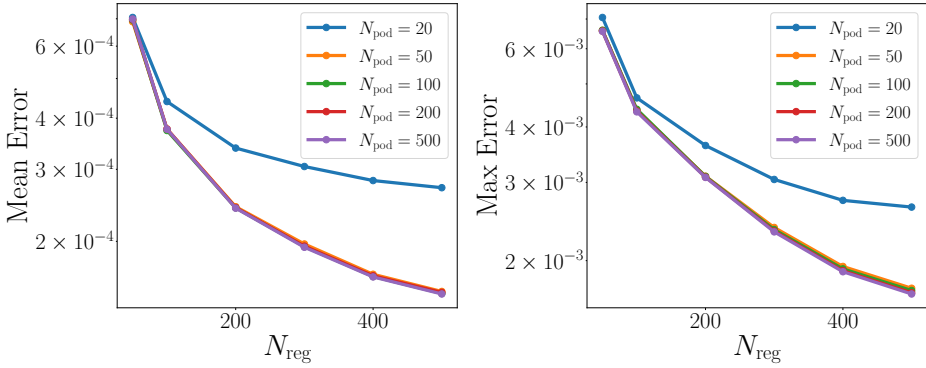
**Figure 3.3:** Eigenvalues of the correlation matrix for different numbers of snapshots  $N_{\text{pod}}$  used for POD.



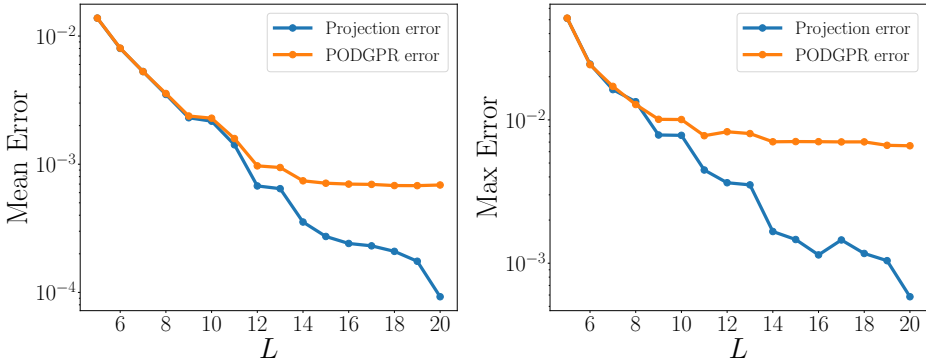
**Figure 3.4:** Von Mises stress of the first three POD basis functions of the microscopic stress field.

of roughly 0.65%.

**Influence of  $L$**  Using  $N_{\text{pod}} = N_{\text{reg}} = 50$ , the influence of the number of basis functions  $L$  is then investigated. The relative errors of the effective stress are given in Fig. 3.6. In addition, the projection error, computed by projecting the true stress field onto the POD basis, is also plotted to compare the quality of the regression in Eq. (3.21). The projection error can be interpreted as a lower bound for the accuracy of PODGPR, for the case when the regression error is zero. As seen in Fig. 3.6, both the mean and maximum error of the first 8 basis functions nearly perfectly match. However, for a larger number of basis functions, the discrepancy slowly grows and the error flattens. Generally, the POD coefficients get increasingly more oscillatory with increasing  $L$  and hence require more data for an accurate regression. To show that with increasing  $N_{\text{reg}}$ ,



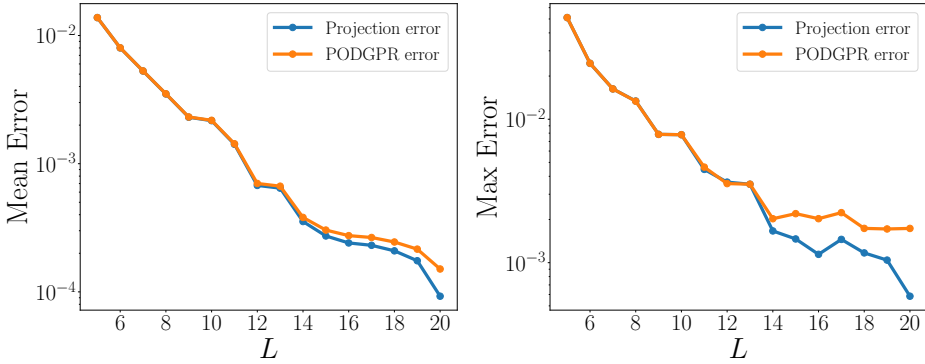
**Figure 3.5:** Comparison of stress errors for different combinations of  $N_{\text{pod}}$  and  $N_{\text{reg}}$  for  $L = 20$ .



**Figure 3.6:** Comparison of stress errors for different  $L$  with  $N_{\text{pod}} = N_{\text{reg}} = 50$ .

the regression error decreases and approaches the projection error, the error over  $L$  for  $N_{\text{pod}} = 50$  with  $N_{\text{reg}} = 500$  is also plotted in Fig. 3.7. For this case, the mean error matches the projection error for 13 basis functions, while the maximum error also gets much closer to the projection error. Note that the maximum error of PODGPR is sometimes slightly lower than the projection error. This can be explained with errors in the regression that lead to a stress field which after averaging ends up closer to the high fidelity solution than the best projection. This is entirely random and as  $N_{\text{reg}}$  is further increased, the regression error will tend towards zero and the curves will match eventually.

**Comparison with neural networks** All available  $N_{\text{reg}} = 500$  training snapshots and all 1000 test snapshots are set as training and validation dataset for training different neural networks, as described in the introduction of this



**Figure 3.7:** Comparison of stress errors for different  $L$  with  $N_{\text{pod}} = 50$  and  $N_{\text{reg}} = 500$ .

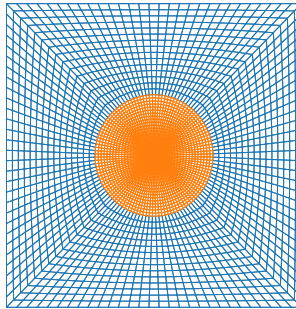
**Table 3.1:** Validation and effective stress error for different feedforward neural network architectures. The lowest values in each column have been highlighted.

Architecture	Validation Loss	$\varepsilon_{\bar{\mathbf{P}}}^{\text{mean}}$	$\varepsilon_{\bar{\mathbf{P}}}^{\text{max}}$
$N_h = 1, N_n = 20$	$2.33 \times 10^{-7}$	<b>0.0016</b>	0.0199
$N_h = 1, N_n = 50$	<b><math>1.72 \times 10^{-7}</math></b>	0.0017	<b>0.0136</b>
$N_h = 2, N_n = 20$	$2.4 \times 10^{-7}$	0.0019	0.0184
$N_h = 2, N_n = 50$	$1.94 \times 10^{-7}$	0.0018	0.0166

section. The validation loss (computed as the mean squared error on the testing snapshots after training) and relative errors  $\varepsilon_{\bar{\mathbf{P}}}$  obtained for each architecture are given in Table 3.1, where the best architecture is highlighted. All architectures produce similar results, with the second architecture  $N_h = 1, N_n = 50$  yielding the best results with an average and maximum error of 0.17% and 1.36%. On the other hand, PODGPR already yields a lower mean and maximum error of around 0.065% and 0.65% when using only 50 training data points, showing that PODGPR can utilize the information of each snapshot more efficiently than the neural network. When 500 snapshots are used for the regression in PODGPR, a mean and max error of less than 0.02% and 0.2% is achieved, hence outperforming the neural network.

### 3.2.2 Fiber reinforced material

In the second example, the considered microstructural RVE consists of two different phases: a soft matrix and a stiff fiber material. Both materials are Neo-Hookean (see Appendix A.1.1 for the definition), but the matrix has parameters  $C_1 = 1, D_1 = 1$ , whereas the fiber has variable parameters  $C_1 = D_1 \in [50, 150]$ , corresponding to a Young's modulus that is 50–150 times



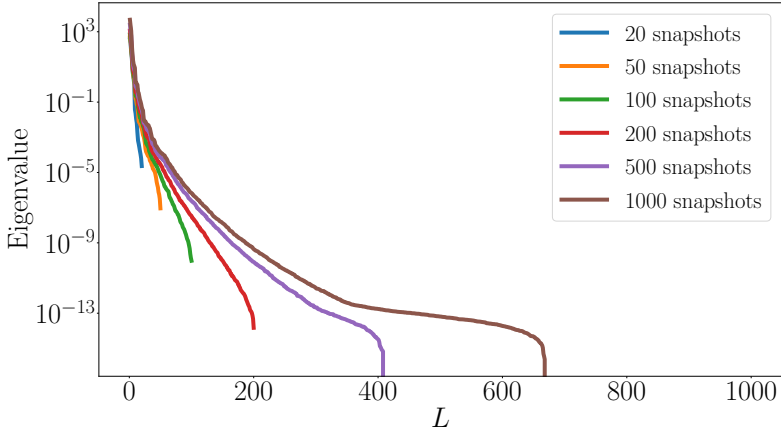
**Figure 3.8:** The considered microstructure consisting of a Neo-Hookean matrix material and a Neo-Hookean fiber material. The fiber volume fraction is 12.56%. The mesh consists of 4321 eight-node elements and 13080 nodes.

higher than the matrix, with the Poisson's ratio remaining the same ( $\nu = 0.25$ ), cf. Eq. (A.4). The geometry used is depicted in Fig. 3.8, where the matrix (blue) completely surrounds the fiber (orange). Eight node quadrilateral elements with four quadrature points are employed. The volume fraction of the fiber is 12.56%. Kinematic boundary conditions (defined in Section 2.2.2),  $\bar{U}_{xx} - 1, \bar{U}_{yy} - 1, \bar{U}_{xy} - 1 \in [-0.3, 0.3]$  and fiber parameters of  $C_1 = D_1 \in [50, 150]$  are considered. Therefore, the considered problem has  $N_{\text{in}} = 4$  varying parameters.

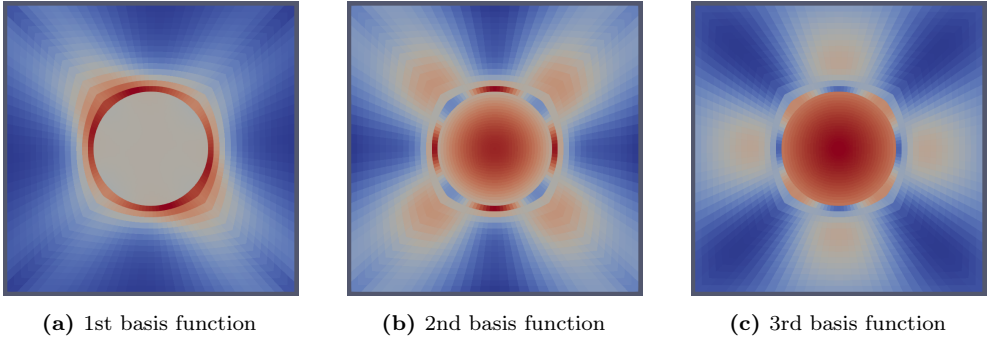
**Data generation** A set of 1000 training snapshots and another set of 1000 test snapshots are generated. The first set contains the  $2^4 = 16$  corner points of the parameter domain and the remaining points are sampled from a Sobol sequence, while the second set is generated from a random uniform distribution.

**Eigenvalues** The eigenvalues of the correlation matrix (cf. Eq. (B.2)) for different numbers of training snapshots are plotted in Fig. 3.9. Similar to the last example, for all cases, a quick decay can be observed, showing the reducibility of the problem. The magnitude of the stress of the first three POD basis functions are plotted in Fig. 3.10. Subsequently,  $L = 20$  basis functions are considered which correspond to an energy  $\mathcal{E}_{\text{pod}}$  of 99.9901% (see Eq. (B.4) for the definition).

**Influence of  $N_{\text{pod}}$  and  $N_{\text{reg}}$**  For  $L = 20$  basis functions, combinations of  $N_{\text{pod}} \in \{20, 50, 100, 200, 500\}$  and  $N_{\text{reg}} \in \{100, 200, 300, 400, 500, 1000\}$  are tested. The mean and maximum error plots can be seen in Fig. 3.11. It can be observed that for all cases with  $N_{\text{reg}} > 50$ , the mean error is below 1%. The mean error improves as more snapshots are considered for the POD basis, suggesting that there is still new information in the snapshots, which fine-tunes the optimal



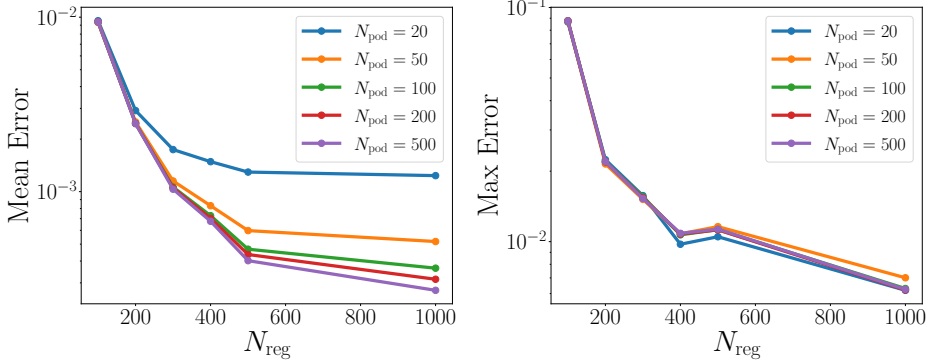
**Figure 3.9:** Eigenvalues of the correlation matrix for different numbers of snapshots used for POD.



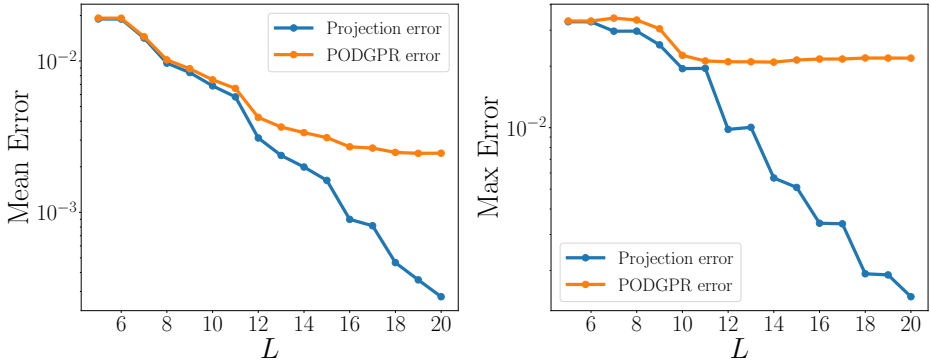
**Figure 3.10:** Von Mises stress of the first three POD basis functions of the microscopic stress field.

basis. The mean error reaches around 0.04% for  $N_{\text{reg}} = 500$  snapshots with a maximum error of around 1%. Increasing  $N_{\text{reg}}$  from 500 to 1000 does not improve the mean error significantly, meaning that the projection error has already been reached with 500 snapshots and  $L = 20$  basis functions. Weighing the approximation error over the number of full computations needed,  $N_{\text{pod}} = N_{\text{reg}} = 200$  is considered in the subsequent analysis.

**Influence of  $L$**  Using  $N_{\text{pod}} = N_{\text{reg}} = 200$ , the influence of  $L$  is investigated. The mean and maximum error of the approximation error over the number of basis functions  $L$  used are shown in Fig. 3.12. The projection error is also plotted to reveal the quality of the regression of Eq. (3.21). As seen from Fig. 3.12, the mean and maximum error for the first 11 basis functions



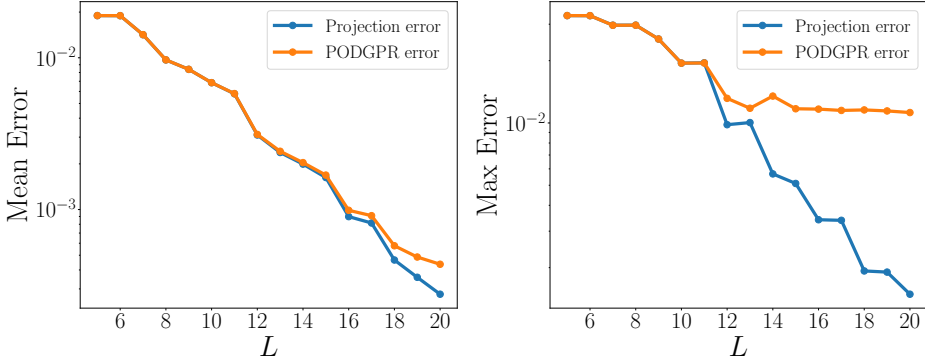
**Figure 3.11:** Comparison of stress errors for different combinations of  $N_{\text{pod}}$  and  $N_{\text{reg}}$  for  $L = 20$ .



**Figure 3.12:** Comparison of stress errors for different  $L$  with  $N_{\text{pod}} = 200$  and  $N_{\text{reg}} = 200$ .

match the projection error quite well, and afterwards flatten out, similar to the last example. When the number of snapshots  $N_{\text{reg}}$  is increased to 500, the mean error matches the projection error for 16 basis functions, as seen in Fig. 3.13, while the maximum error also slightly improves.

**Comparison with neural network** Similarly to the last example, deep forward neural networks with different architectures are trained for comparison. The available 500 training data and 1000 testing data are used for training and validation. The same network architecture configurations as described in the introduction are tested, with one additional combination  $N_{\text{h}} = 2, N_{\text{n}} = 100$ . The results are given in Table 3.2. The fourth architecture  $N_{\text{h}} = 2, N_{\text{n}} = 50$  performs the best with an average error of 0.39% and a maximum error of 2.06%. Same as for the previous example, PODGPR outperforms the neural network. With



**Figure 3.13:** Comparison of stress errors for different  $L$  with  $N_{\text{pod}} = 200$  and  $N_{\text{reg}} = 500$ .

**Table 3.2:** Validation loss and effective stress error for different feedforward neural network architectures. The lowest values in each column have been highlighted in bold face.

Architecture	Validation Loss	$\varepsilon_{\overline{\mathbf{P}}}^{\text{mean}}$	$\varepsilon_{\overline{\mathbf{P}}}^{\text{max}}$
$N_h = 1, N_n = 20$	$1.7 \times 10^{-6}$	0.0077	0.0368
$N_h = 1, N_n = 50$	$8.55 \times 10^{-7}$	0.0056	0.0289
$N_h = 2, N_n = 20$	$5.36 \times 10^{-7}$	0.0047	<b>0.0176</b>
$N_h = 2, N_n = 50$	<b><math>2.97 \times 10^{-7}</math></b>	<b>0.0039</b>	0.0206
$N_h = 2, N_n = 100$	$7.91 \times 10^{-7}$	0.0052	0.0315

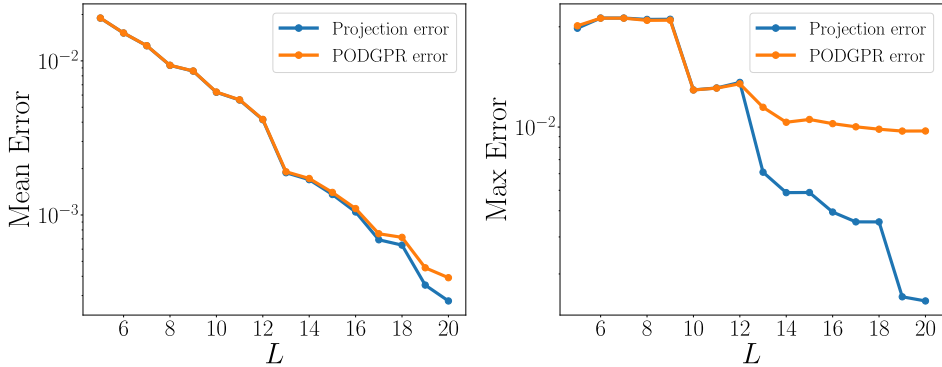
only 200 training snapshots, PODGPR achieves roughly the same accuracy as the NN using 500 snapshots. With 500 employed snapshots, PODGPR reaches a mean and maximum error of 0.04% and 1%, hence outperforming the neural network, while also being able to recover the microscopic stress field.

**Remark on periodic boundary conditions** For the same microstructure (Fig. 3.8), an analogous analysis using periodic boundary conditions was considered, defined in Section 2.2.2. In contrast to kinematic boundary conditions, the fluctuation displacement field  $\mathbf{w}^p$  is assumed to be periodic on the RVE boundary. The obtained results regarding the approximation quality are shown in Fig. 3.14, where  $N_{\text{pod}} = 200$  and  $N_{\text{reg}} = 500$  was chosen. In comparison to Fig. 3.13, the results are comparable and hence it can be concluded that PODGPR works independently of the chosen boundary conditions.

### 3.2.3 Two-scale Cook's membrane problem

To show the performance of PODGPR in a two-scale simulation, the learned constitutive model for the fiber reinforced RVE with  $N_{\text{pod}} = N_{\text{reg}} = 200$  is em-





**Figure 3.14:** Fiber reinforced material with periodic boundary conditions: Comparison of stress errors for different  $L$  with  $N_{\text{pod}} = 200$  and  $N_{\text{reg}} = 500$ .

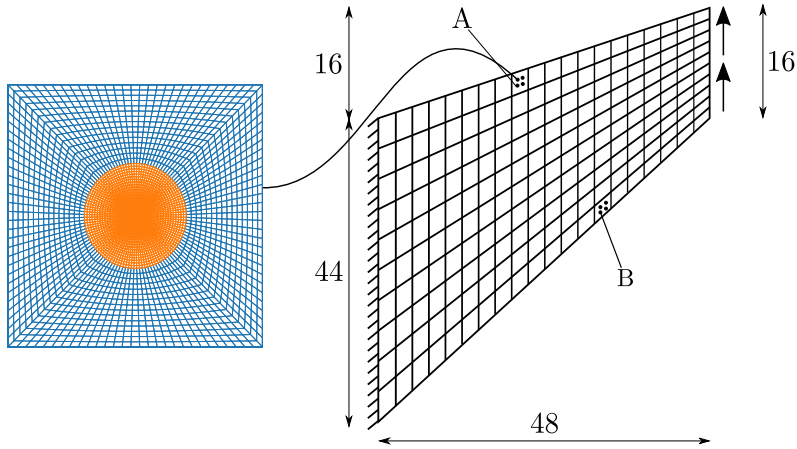
bedded inside a FE solver. For the macroscopic problem the Cook’s membrane, consisting of the fiber reinforced RVE, is chosen. The geometry and mesh of the membrane are given in Fig. 3.15. The macroscopic mesh consists of 200 bilinear quadrilateral elements with 4 quadrature points. The quadrature points A and B, as marked on the figure, denote points in which the microscopic stress fields are compared against the reference solution. The reference solution is obtained by running a full two-scale FE2 simulation. The left side of the sample is fixed and the right side is loaded in five time steps with a vertical traction of 0.1, which leads to local deformations of up to 30%. The material parameters of the fiber are taken to be fixed with  $C_1 = D_1 = 100$  in this example.

The  $yx$ -component of the macroscopic stress  $\bar{P}_{yx}$  obtained by the full FE2 and FE with PODGPR simulation are given in Fig. 3.16, while the microscopic stress  $P_{yx}$  at point A and B are shown in Fig. 3.17 and Fig. 3.18. The relative error defined as  $\varepsilon_{P_{yx}} := |P_{yx}^{\text{FE2}} - P_{yx}^{\text{ROM}}| / \langle |P_{yx}^{\text{FE2}}| \rangle$ , with  $\langle |P_{yx}^{\text{FE2}}| \rangle$  the averaged absolute stress, is also shown (Figs. 3.16c, 3.17c and 3.18c). As can be seen, the shape of the stress field of both solutions is indistinguishable. Even though the relative errors for the microscopic problem reach a maximum of 7% near the interface of matrix and fiber, after homogenization the highest error reduces to merely 1%. This discrepancy is due to the fact that the method tries to reduce the  $L_2$ -norm of the error in the stress field and therefore allows locally high errors.

Using 48 cores<sup>1</sup>, the FE2 simulation takes around 100 minutes while the simulation with PODGPR is completed within 1 minute on a single core<sup>2</sup>, resulting in a speed-up of about three orders of magnitude. For PODGPR,

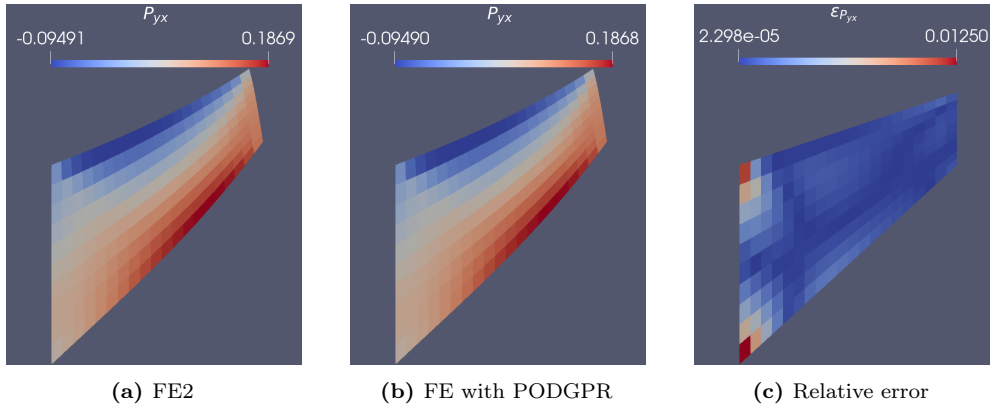
<sup>1</sup>Intel Xeon Platinum 8260

<sup>2</sup>Intel Core i7-8750H

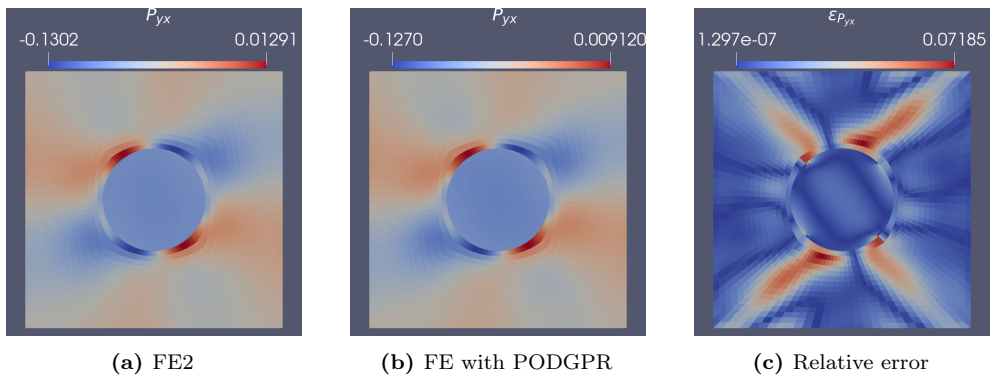


**Figure 3.15:** The left side was fixed while a traction in the vertical direction of 0.1 was applied along the right side. The mesh consisted of 200 bilinear elements and 231 nodes. The microstructural stress field at the marked quadrature points A and B will be shown later.

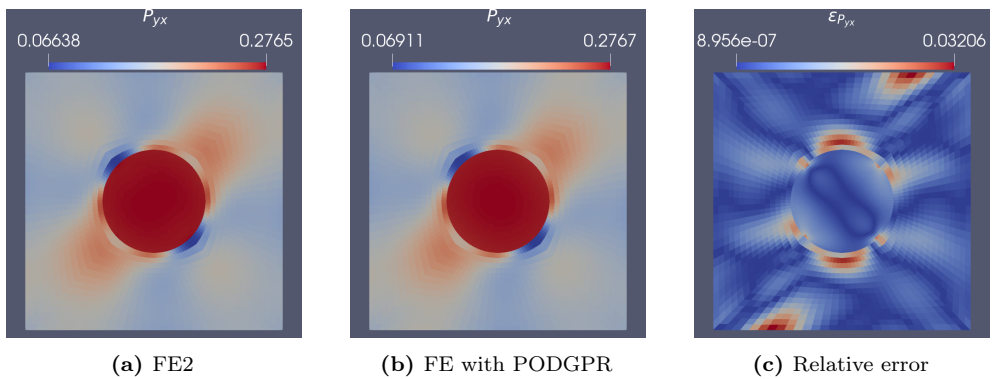
200 RVE simulations have to be pre-computed, which takes less than one hour on a single core. Performing the POD and GPR to construct PODGPR takes around one minute.



**Figure 3.16:** Macroscopic stress field.



**Figure 3.17:** Microscopic stress field at Point A.



**Figure 3.18:** Microscopic stress field at Point B.

### 3.3 Numerical examples for varying geometry

In this section, the performance of PODGPR is demonstrated for varying geometries. PODGPR is first applied onto two geometrically parameterized microstructures to showcase the generality of the approach and its accuracy. Then, a two-scale Cook's membrane problem is shown to illustrate the speed-up and its potential applications.

Same as in Section 3.2, all examples are defined in 2D under plane strain conditions, and all quantities are normalized and considered dimensionless. All RVEs are assumed to be of size  $[0, 1]^2$ , and the polar decomposition (cf. Remark 2.1.1) is employed to reduce the number of load parameters. The nonlinear physical simulations are solved within the Finite Element framework MOOSE [99] and the linear auxiliary problems are solved with an in-house code<sup>3</sup> written in Python. As mentioned in Remark 3.1.2, to obtain physically consistent transformations without the need for interpolation, we use the auxiliary problem to generate simulation meshes.

To quantify the quality of the approximation, the following two error measures are introduced:

1. Relative error of stress field

$$\varepsilon_{\mathbf{P}} = \frac{\|\mathbf{P}^{\text{truth}} - \mathbf{P}^{\text{PODGPR}}\|_{L^2(\Omega^\mu)}}{\|\mathbf{P}^{\text{truth}}\|_{L^2(\Omega^\mu)}}, \quad (3.38)$$

2. Relative error of effective stress

$$\varepsilon_{\bar{\mathbf{P}}} = \frac{\|\bar{\mathbf{P}}^{\text{truth}} - \bar{\mathbf{P}}^{\text{PODGPR}}\|_F}{\|\bar{\mathbf{P}}^{\text{truth}}\|_F}, \quad (3.39)$$

where  $(\bullet)^{\text{truth}}$  and  $(\bullet)^{\text{PODGPR}}$  indicate the full and approximate solution, and  $\|(\bullet)\|_{L^2(\Omega^\mu)}$  and  $\|(\bullet)\|_F$  denote the  $L^2(\Omega^\mu)$  and the Frobenius norm, with  $\Omega^\mu$  the parameterized domain. The average errors for given testing datasets are defined as:

$$\bar{\varepsilon}_{\mathbf{P}} = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \varepsilon_{\mathbf{P}}^n, \quad \bar{\varepsilon}_{\bar{\mathbf{P}}} = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \varepsilon_{\bar{\mathbf{P}}}^n, \quad (3.40)$$

where  $N_{\text{test}}$  is the number of testing snapshots and  $\varepsilon_{\mathbf{P}}^n$  and  $\varepsilon_{\bar{\mathbf{P}}}^n$  correspond to the relative errors of the  $n$ -th snapshot.

For comparison, we also trained several deep feed-forward neural networks for the effective stress using the same data for each example, similar to the

<sup>3</sup>The implementation can be found on <https://github.com/theronguo/auxiliary-problem>.

results in Section 3.2. All considered neural networks have as many inputs as the number of parameters, 4 outputs for each stress component and 2 hidden layers each with  $N_n$  neurons. We trained four architectures with  $N_n \in \{50, 100, 200, 300\}$ . These four architectures will be referred to as NN1, NN2, NN3 and NN4. ELU activation functions are applied on each layer apart from the last layer. For the optimization, the mean squared error loss function is chosen and optimized with the Adam optimizer [66] with a learning rate of  $1 \times 10^{-4}$  and a batch size of 32 for 10000 epochs. The training is performed with the Python package PyTorch [98].

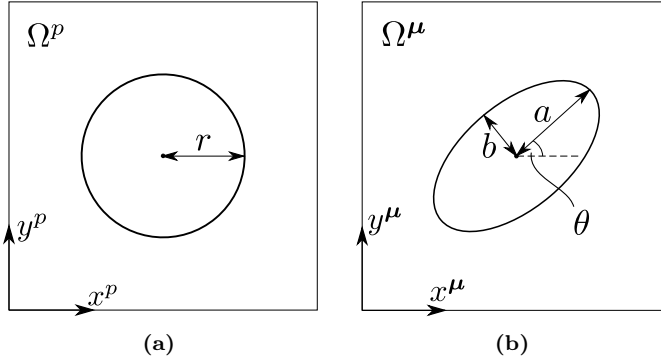
### 3.3.1 Composite microstructure with an elliptical fiber

In the first example, a composite structure, consisting of a soft matrix and an elliptical stiff fiber around the center of the domain  $\mathbf{X}_c = [0.5, 0.5]^T$ , is considered. Three geometrical parameters that parameterize the fiber shape, the semi-major axis  $a$ , semi-minor axis  $b$  and a rotation angle  $\theta$ , are considered, see Fig. 3.19b. Together with the three loading directions  $\bar{U}_{xx}$ ,  $\bar{U}_{yy}$  and  $\bar{U}_{xy}$ , this problem has  $N_{\text{in}} = 6$  parameters. For the matrix material a Neo-Hookean material model (see Appendix A.1.1 for the definition) with  $C_1 = 1$  and  $D_1 = 1$  is chosen, while for the fiber material a Neo-Hookean material model with  $C_1 = 100$  and  $D_1 = 100$  is assumed, corresponding to a stiffness that is 100 times higher than the matrix material. Both materials have Poisson's ratio 0.25. The considered parameter ranges are given in Table 3.3. Lower and upper bounds for the macroscopic stretches  $\bar{\mathbf{U}}$  are chosen such that the solution of the microstructural problem converges for all parameter configurations: for some geometrical parameters, a few elements of the parent mesh might become highly distorted after applying the geometrical transformation; together with the high contrast of material stiffness of both materials, larger magnitudes than 0.15 of the components  $\bar{\mathbf{U}} - \mathbf{I}$  lead to convergence issues. If larger magnitudes have to be considered, a different parent domain could be employed or a smaller parameter space of the geometrical parameters could be chosen.

**Table 3.3:** 6 parameters with corresponding ranges. The parameters  $a$ ,  $b$  and  $\theta$  are the geometrical parameters describing the interface while  $\bar{U}_{xx}$ ,  $\bar{U}_{yy}$  and  $\bar{U}_{xy}$  are external loading parameters.

$a$	$b$	$\theta$	$\bar{U}_{xx} - 1$	$\bar{U}_{yy} - 1$	$\bar{U}_{xy}$
[0.1, 0.35]	[0.1, 0.35]	$[-\pi/2, \pi/2]$	[-0.15, 0.15]	[-0.15, 0.15]	[-0.15, 0.15]

**Setup of the auxiliary problem** For the parent domain, a domain with a circular inclusion with radius  $a = b = r = 0.225$  is chosen, see Fig. 3.19a. The



**Figure 3.19:** Parent and parameterized domain. (a) The chosen parent domain with a circular interface with fixed radius  $a = b = r = 0.225$ , and (b) the parameterized domain, characterized by semi-major axis  $a$ , semi-minor axis  $b$  and angle  $\theta$ . The parent mesh consists of a total of 20769 nodes and 10306 6-node triangular elements.

employed simulation mesh consists of 20769 nodes and 10665 6-node triangular elements. The transformation of the circular interface into the elliptical interface can be given as:

$$\mathbf{x}^\mu(\mathbf{x}^p) = \hat{\mathbf{R}} \begin{bmatrix} a/r & 0 \\ 0 & b/r \end{bmatrix} \begin{pmatrix} \tilde{r}(\mathbf{x}^p - \mathbf{X}_c) \cos(\tilde{\theta}(\mathbf{x}^p - \mathbf{X}_c) - \theta) \\ \tilde{r}(\mathbf{x}^p - \mathbf{X}_c) \sin(\tilde{\theta}(\mathbf{x}^p - \mathbf{X}_c) - \theta) \end{pmatrix} + \mathbf{X}_c, \quad (3.41)$$

where

$$\hat{\mathbf{R}}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (3.42)$$

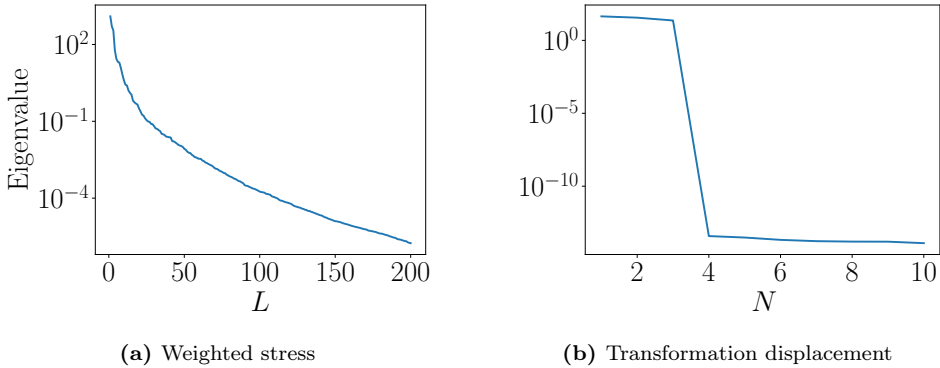
$$\tilde{r}(\mathbf{x}^p) = \sqrt{\mathbf{x}^{pT} \mathbf{x}^p}, \quad (3.43)$$

$$\tilde{\theta}(\mathbf{x}^p) = \arctan2(x^p, y^p), \quad (3.44)$$

with  $\mathbf{x}^p = [x^p, y^p]^T$  a column matrix of each of the nodal positions located at the interface. The  $\arctan2$ -function of Eq. (3.44) is an extension of the  $\arctan$ -function defined as,

$$(x, y) \mapsto \begin{cases} \arctan(y/x) & x > 0, \\ \arctan(y/x) + \pi & x < 0, y \geq 0, \\ \arctan(y/x) - \pi & x < 0, y < 0, \\ \pi/2 & x = 0, y > 0, \\ -\pi/2 & x = 0, y < 0, \\ \text{undefined} & x = 0, y = 0. \end{cases} \quad (3.45)$$

The Poisson's ratio  $\xi^{\text{aux}}$  for the auxiliary problem is first chosen to be 0.3. In the results below, its influence on the accuracy of obtained results is discussed.

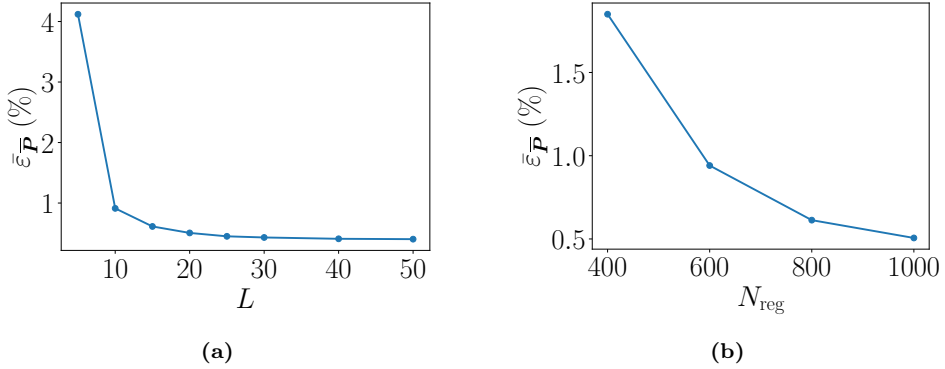


**Figure 3.20:** Eigenvalues of the correlation matrix for (a) weighted stress and (b) transformation displacement. For the weighted stress the eigenvalues decay quickly, while for the transformation displacement only the first three modes are nonzero. This means that the transformation displacement can be represented with only three basis functions.

**Data generation** In total  $N_{\text{reg}} = 1000$  training snapshots are generated, of which  $2^6 = 64$  snapshots are at the corners of the 6-dimensional parameter space and the rest are sampled from a Sobol sequence [123]. For testing, another  $N_{\text{test}} = 500$  snapshots are generated from a random uniform distribution.

**POD of weighted stress and transformation displacement** The eigenvalues of the correlation matrix for the weighted stress are given in Fig. 3.20a. It can be observed that the eigenvalues decay quickly, indicating a good reducibility. To show that the auxiliary problem in Eq. (3.3) can be reduced drastically, the eigenvalues of the correlation matrix for the transformation displacement are also shown in Fig. 3.20b. All but three eigenvalues are essentially zero, showing that the auxiliary problem can be solved with three basis functions.

**Approximation errors** The average approximation error of the effective stress on the 500 testing snapshots of PODGPR for different numbers of basis functions  $L$  and training snapshots  $N_{\text{reg}}$  is shown in Fig. 3.21. All  $N_{\text{reg}}$  training snapshots are used for both the POD and GPR. In Fig. 3.21a, the error decays rapidly for the first few basis functions. For  $L = 20$  an error of roughly 0.5% is reached. However, taking into account more basis functions barely improves the performance since the coefficients get increasingly more oscillatory with increasing number and, hence, more difficult to approximate with a GPR model, similar to the results in Section 3.2. From Fig. 3.21b, we see that a higher number of snapshots is crucial for the accuracy of PODGPR. Data shown corresponds to  $L = 20$  basis functions. For small datasets the error increases



**Figure 3.21:** (a) Average error of the effective stress over number of basis functions with  $N_{\text{reg}} = 1000$  training snapshots. The error curve decays rapidly for the first few basis functions and then flattens out. (b) Average error of the effective stress over number of training snapshots used for  $L = 20$  basis functions. The error increases drastically when fewer training snapshots are used.

exponentially, indicating a poor approximation of the first 20 POD coefficients.

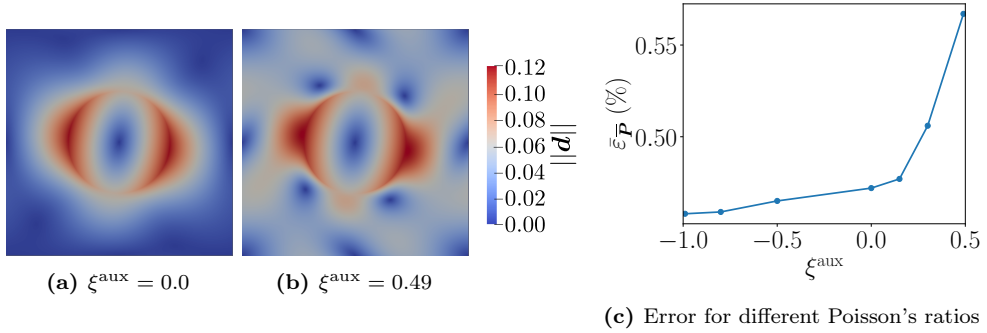
In Table 3.4, the approximation quality of PODGPR with  $L = 20$  basis functions is compared with the four neural networks. The projection error (defined as the error of projecting the truth solution onto the reduced basis) with  $L = 20$  basis functions is also given. PODGPR approximates the effective stress better than all the NN models at least by a factor of 2. Moreover, it is nearly able to reach the error of the projection error in both error measures, indicating that with  $N_{\text{reg}} = 1000$  training snapshots the first 20 POD coefficients can be well captured. The error in the stress field is less than 0.1%, while for the effective stress the error is 0.5%. Furthermore, by comparing the results with Fig. 3.21b, it can be seen that the best results obtained by the neural networks are reached by PODGPR with only  $N_{\text{reg}} = 600$  snapshots, showing that PODGPR is more data efficient than the neural networks in this case.

**Table 3.4:** Approximation errors for different methods. The trained surrogate models are tested on 500 testing snapshots. PODGPR outperforms all NN models by a factor 2.

	Projection	PODGPR	NN1	NN2	NN3	NN4
$\bar{\epsilon}_{\mathbf{P}}$	$8.17 \times 10^{-4}$	$9.49 \times 10^{-4}$	n.a.	n.a.	n.a.	n.a.
$\bar{\epsilon}_{\mathbf{P}}$	$2.74 \times 10^{-3}$	$5.06 \times 10^{-3}$	$2.66 \times 10^{-2}$	$1.53 \times 10^{-2}$	$1.02 \times 10^{-2}$	$9.8 \times 10^{-3}$

**Influence of Poisson's ratio of the auxiliary problem** In this example, the influence of the auxiliary Poisson's ratio on the approximation of the effective stress is investigated, considering  $\xi^{\text{aux}} \in \{-0.99, -0.8, -0.5, 0.0$ ,





**Figure 3.22:** Norm of transformation displacement with  $a = 0.104$ ,  $b = 0.291$ ,  $\theta = -8.44^\circ$  for (a)  $\xi^{\text{aux}} = 0.0$  and (b)  $\xi^{\text{aux}} = 0.49$ . While the former leads to localized deformations, the latter affects the entire domain. (c) The mean error plotted over the Poisson's ratio  $\xi^{\text{aux}}$ . The errors decrease with decreasing  $\xi^{\text{aux}}$ , but remain on the same order of magnitude  $\mathcal{O}(10^{-3})$ .

$0.15, 0.3, 0.49\}$ . Example transformation maps for  $\xi^{\text{aux}} \in \{0.0, 0.49\}$  are shown in Figs. 3.22a and 3.22b and the obtained errors are plotted over the Poisson's ratio in Fig. 3.22c. For this example, the lower the Poisson's ratio the better the approximation. Nevertheless, all error values are close to each other (ranging from 0.46% to 0.58%), although the transformation displacement field differs significantly, see Figs. 3.22a and 3.22b. This empirical result suggests that the choice of the Poisson's ratio is insignificant and, since there is no practical way of finding the best value,  $\xi^{\text{aux}} = 0.3$  is adopted hereafter.

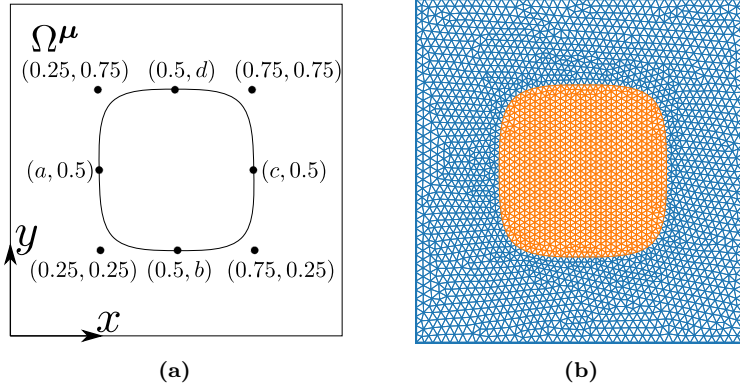
### 3.3.2 Composite microstructure with a B-spline controlled inclusion shape

In the second example, an inclusion with a shape that is described by a B-spline with eight control points is considered, see Fig. 3.23a. The  $x$ -coordinate of the center left and right control point and the  $y$ -coordinate of the middle top and bottom control point are parameterized, resulting in four geometrical parameters  $a$ ,  $b$ ,  $c$  and  $d$ . The curve is then interpolated with cubic polynomials using the Python library NURBS-Python [12]. The same material parameters are chosen as in the first example, with parameter ranges given in Table 3.5. A few example geometries are shown in Fig. 3.24 to show the variety of shapes covered by this parameterization. The lower and upper bounds for  $\bar{U}$  are chosen in the same way as in the previous example in Section 3.3.1.

**Setup of the auxiliary problem** For the parent domain, the midpoint of the parameter domain is selected, i.e.,  $a = b = 0.25$  and  $c = d = 0.75$ . The corresponding geometry and mesh, consisting of 11296 nodes and 5833 6-node

**Table 3.5:** 7 parameters with corresponding ranges. The parameters  $a$ ,  $b$ ,  $c$  and  $d$  are geometrical parameters describing the interface, see Fig. 3.23 for the explanation, while  $\bar{U}_{xx}$ ,  $\bar{U}_{yy}$  and  $\bar{U}_{xy}$  are external loading parameters.

$a$	$b$	$c$	$d$	$\bar{U}_{xx} - 1$	$\bar{U}_{yy} - 1$	$\bar{U}_{xy}$
[0.1, 0.4]	[0.1, 0.4]	[0.6, 0.9]	[0.6, 0.9]	[-0.15, 0.15]	[-0.15, 0.15]	[-0.15, 0.15]



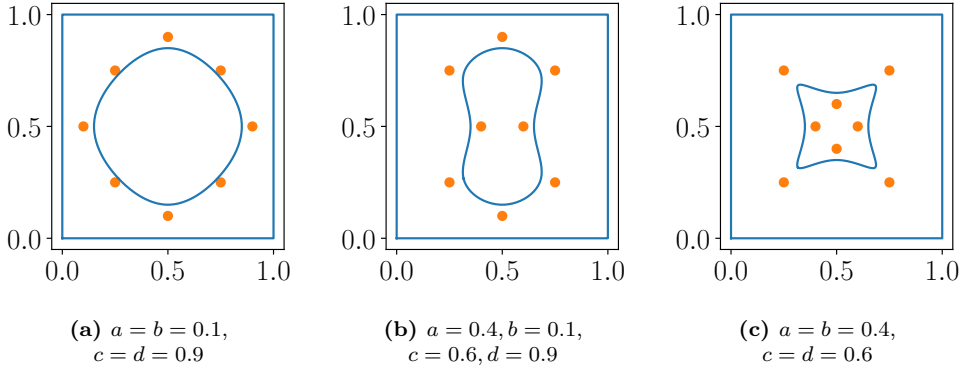
**Figure 3.23:** Parameterized domain. (a) The interface is spanned by eight control points. Out of those, four control points can move in one direction, which are controlled by the geometrical parameters  $a$ ,  $b$ ,  $c$  and  $d$ . (b) The parent geometry with  $a = b = 0.25$  and  $c = d = 0.75$  is chosen and the mesh consists of 11296 nodes and 5566 6-node triangular elements.

triangular elements, are shown in Fig. 3.23.

**Data generation** In total  $N_{\text{reg}} = 1000$  training snapshots are again generated from a Sobol sequence [123], while another  $N_{\text{test}} = 500$  snapshots are generated from a random uniform distribution for testing.

**POD of weighted stress and transformation displacement** The eigenvalues of the correlation matrix for both weighted stress and transformation displacement are depicted in Fig. 3.25. A fast decay can be observed for the weighted stress, while all but four eigenvalues are essentially zero for the transformation displacement.

**Approximation errors** The average approximation errors on the 500 testing snapshots of the projection error (projection of truth solution onto the reduced basis) with  $L = 50$  basis functions, PODGPR with  $L = 50$  basis functions and the four neural networks are reported in Table 3.6. From the results it can be observed that PODGPR nearly reaches the projection error, showing that the first 50 coefficients are well approximated by the GPR models. Furthermore,



**Figure 3.24:** Example geometries. The control points are shown in orange color and the resulting interface in blue color.

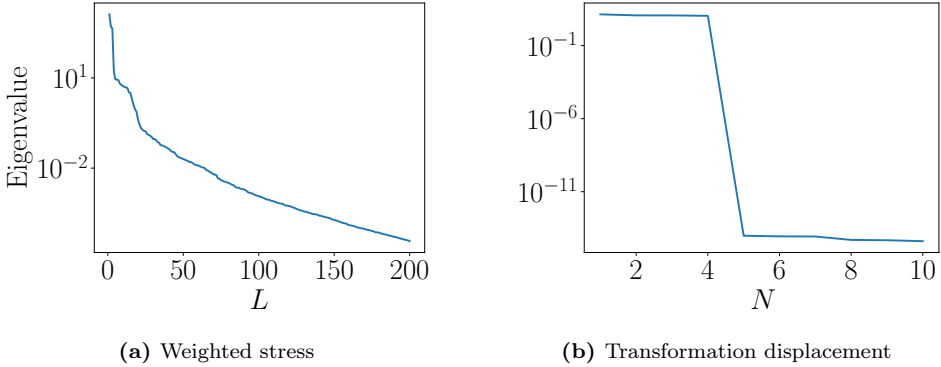
it outperforms all neural network architectures by a factor of 2, reaching an average error of 0.131% in effective stress.

**Table 3.6:** Mean errors for different methods. The trained surrogate models are tested on 500 testing snapshots. PODGPR outperforms all NN models by a factor 2.

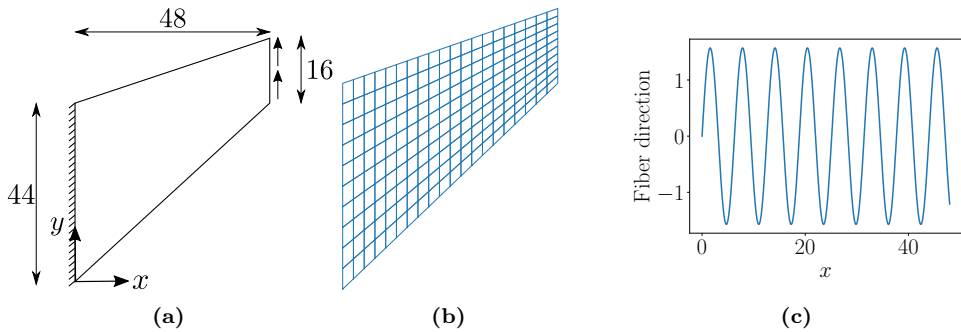
	Projection	PODGPR	NN1	NN2	NN3	NN4
$\bar{\varepsilon}_{\mathcal{P}}$	$1.79 \times 10^{-4}$	$1.85 \times 10^{-4}$	n.a.	n.a.	n.a.	n.a.
$\bar{\varepsilon}_{\bar{\mathcal{P}}}$	$1.09 \times 10^{-3}$	$1.31 \times 10^{-3}$	$3.3 \times 10^{-3}$	$2.5 \times 10^{-3}$	$2.2 \times 10^{-3}$	$2.3 \times 10^{-3}$

### 3.3.3 Geometrically parameterized two-scale Cook's membrane problem

While the last two examples dealt with the construction of the surrogate model for the microscale, in this example the surrogate model is employed in a full two-scale Cook's membrane problem. Here, the microstructure from Section 3.3.1 with an elliptical inclusion is considered. The geometry of the Cook's membrane and its mesh are shown in Fig. 3.26. The mesh consists of 200 quadrilateral elements with 4 quadrature points, resulting in 800 microstructure evaluations required for a single Newton iteration. The microstructural parameters  $a$  and  $b$  are assumed to be constant with  $a = 0.35$  and  $b = 0.1$ , corresponding to an ellipse, while the angle  $\theta$  is a function in the  $x$ -coordinate with  $\theta(x) = (\pi \sin x)/2$ , in order to test the performance of the surrogate model under rapidly varying fiber directions (see Fig. 3.26c for an illustration of the function). A similar test problem was considered in [111]. The left side of the membrane is fixed, while a vertical traction of 0.05 is applied on the right edge, which leads to overall deformations within the training range of the surrogate



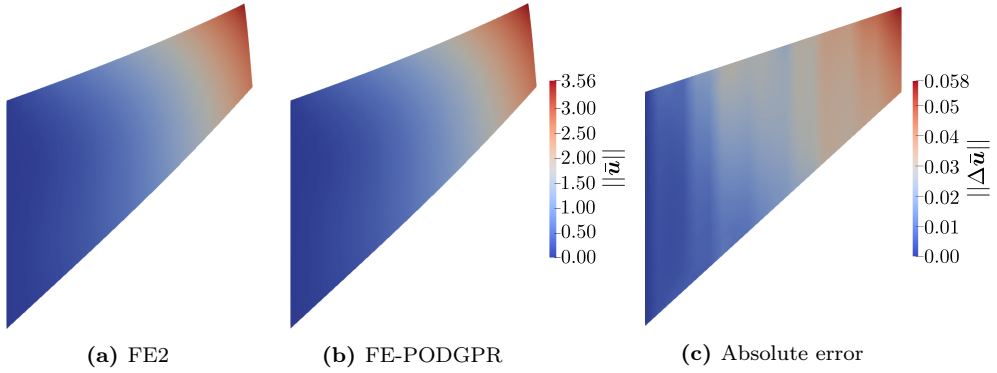
**Figure 3.25:** Eigenvalues of the correlation matrix for (a) weighted stress and (b) transformation displacement. Same as for the last example, the eigenvalues of weighted stress decay exponentially. For the transformation displacement only the first 4 modes are nonzero, meaning it can be represented with 4 basis functions.



**Figure 3.26:** (a) The geometry of the Cook's membrane and (b) corresponding simulation mesh used. The mesh consists of 231 nodes and 200 4-node quadrilateral elements. (c) The fiber direction depends on the  $x$ -coordinate as  $\theta(x) = \frac{\pi}{2} \sin x$ .

model ( $|\bar{U}_{xx} - 1|, |\bar{U}_{yy} - 1|, |\bar{U}_{xy}| < 0.15$ ).

A full FE2 simulation and a FE simulation using PODGPR are run and the obtained displacement fields  $\bar{\mathbf{u}}$  are compared in Fig. 3.27. The PODGPR surrogate model constructed with  $L = 20$  basis functions from  $N_{\text{reg}} = 1000$  training snapshots is used. The obtained displacement fields shown in Figs. 3.27a and 3.27b are almost identical. To further quantify the error, the magnitude of the difference between both displacement solutions is shown in Fig. 3.27c. Here we observe that the highest absolute error is at the top right corner and the error increases from left to right. Comparing the error and the actual value at the top right corner, the relative error corresponds to  $0.058/3.56 = 1.6\%$ . Furthermore, vertical stripes with similar errors can be seen, which corres-



**Figure 3.27:** The norm of the displacement obtained for (a) FE2 and (b) FE with PODGPR. The displacement fields are nearly identical. (c) The magnitude of the absolute error  $\Delta \bar{\mathbf{u}} := \bar{\mathbf{u}}^{\text{FE2}} - \bar{\mathbf{u}}^{\text{PODGPR}}$  between both solutions. The error is around two orders of magnitude lower than the displacement values and increases from left to right. Vertical stripes are visible, corresponding to the rapidly varying fiber orientations, showing that some angles are better or worse approximated.

pond to the quickly varying fiber directions in the  $x$ -coordinate, meaning that some angles are better or worse approximated. The quality of approximation for different angles depends on the sampling of the training snapshots. Moreover, the compliance  $f_c := \mathbf{f}_{\text{ext}} \cdot \bar{\mathbf{u}}$ , where  $\mathbf{f}_{\text{ext}}$  corresponds to the externally applied vertical traction, is computed for both methods, yielding  $f_c^{\text{truth}} = 2.227$  and  $f_c^{\text{PODGPR}} = 2.202$ , implying a relative error of 1.1%. This is an important quantity, often used in optimization problems.

The execution times<sup>4</sup> for both cases are reported in Table 3.7. For the construction of PODGPR, 1000 snapshots are generated, involving each time an auxiliary problem and a full simulation to be solved, taking roughly 4 hours. With the data available, the construction of PODGPR with 20 basis functions takes around 10 minutes. After this offline computation, the online speed-up is on the order of 1000 as compared to the full two-scale simulation.

<sup>4</sup>All operations were executed using four subprocesses on an Intel Core i7-8750H.

**Table 3.7:** Run times. The offline stage for constructing PODGPR takes roughly 260 minutes. Then, the Cook’s membrane problem can be solved within 0.5 minutes, while at the same time the full FE2 simulation takes around 1080 minutes for a single forward solution.

	FE2	FE-PODGPR
Offline	n.a	Auxiliary Problem: $\approx 10$ min Snapshot Generation: $\approx 240$ min PODGPR: $\approx 10$ min
Online	$\approx 1080$ min	$\approx 0.5$ min

### 3.4 Conclusions

In this chapter, by combining proper orthogonal decomposition, Gaussian process regression, and a PDE-based transformation method for the treatment of geometrical parameters, we developed a non-intrusive surrogate model that can learn an effective constitutive model and replace the microscopic problem arising in first-order computational homogenization. With the reduced model, the effective quantities can be rapidly and accurately inferred for a wide range of geometries and material parameters. Moreover, the microscopic stress fields can be fully recovered and visualized, and they automatically fulfill the underlying microscopic governing equations by construction. Additionally, the effective sensitivities with respect to the microstructural parameters can be easily computed, which could be utilized for two-scale shape optimization problems or inverse problems. Due to the non-intrusive nature of the method, it can be easily implemented into any existing finite element solver.

Several numerical examples were studied to assess and validate the methodology. First, the framework was tested for microstructures with fixed geometry. For the two considered microstructures, involving a porous and a fiber-reinforced material, the proposed method captured the effective stress accurately with a mean error of 0.1%. The trained surrogate model was then employed inside a macroscopic simulation, and the reduced two-scale simulation was compared with the full two-scale FE2 simulation, where it reached high accuracy on both macro- and microscale, and an online speed-up of the order of 1000 was observed. Subsequently, geometrically parameterized microstructures were considered. For two microstructures, each described by several geometrical parameters, the surrogate model approximated the stresses with an average error of less than 1%. Embedded in a two-scale problem, where a high variation in geometrical parameters throughout the macroscopic domain was considered, it accelerated the simulation by a factor of 1000, while maintaining high accuracy.

Although this method is powerful, there exist several limitations:

1. For geometrical parameters leading to very severe geometrical variations, the proposed transformation method leads to highly irregular and distorted meshes. A possible remedy to resolve this issue is to correct the distorted elements by mesh refinement, or to use multiple parent geometries with local surrogate models. In the online stage, the closest cluster could be chosen to evaluate the surrogate model.
2. Another challenge is the requirement of data. Even though the methodology proved to be more data efficient than feedforward neural networks in our examples, still a rather large amount of training data is needed. Possible solutions include multi-fidelity methods or adaptive sampling schemes.
3. The treatment of history-dependent material behavior, such as plasticity, is challenging. For such materials, Gaussian process regression cannot be used, but instead recurrent neural networks, such as long short-term memory (LSTM) or gated recurrent unit (GRU) networks, that contain an internal state, have to be considered. However, these methods typically require vast amounts of data, and generating these amounts of data might become infeasible for complicated microstructures, especially in 3D.





## Chapter 4

# An intrusive reduced order model for first-order computational homogenization

The content of this chapter is based on the following publication:

- Guo, T., Rokoš, O., & Veroy, K. (2024). A reduced order model for geometrically parameterized two-scale simulations of elasto-plastic microstructures under large deformations. *Computer Methods in Applied Mechanics and Engineering*, 418, 116467.

In Chapter 3, a data-driven non-intrusive surrogate model of the microscopic problem in first-order computational homogenization (FE2) was derived through a combination of proper orthogonal decomposition (POD), Gaussian process regression (GPR) and a geometrical transformation method. After training, the surrogate model is highly efficient as it only involves the evaluation of few GPR models and the addition of small matrices. However, the method requires rather large amounts of data and, more crucially, extensions to history-dependent material behaviors, such as plasticity, are difficult.

In this chapter, an intrusive surrogate model for the geometrically parameterized microscopic PDE is proposed by employing the reduced basis method (RBM) [103] and the empirical cubature method (ECM) [48], later referred to as PODECM. Both methods reduce the complexity of the microscopic FE model, such that it can be solved more efficiently. As the PDE is still being solved, history-dependent material behavior can be treated in straightforward manner. Similar to the method in Chapter 3, the surrogate model is constructed on the parent domain  $\Omega^P$ . For any new geometry  $\Omega^\mu$ , the linear auxiliary problem can be rapidly solved and the reduced model adapted to each geometry.

## 4.1 Surrogate modelling

### 4.1.1 Reduced basis method

When solving the microscopic problem in Eq. (2.11) for complex problems and geometries, typically a fine mesh is required for the full FE model, leading to a high-dimensional solution space  $\mathcal{V}_h^P$  for the fluctuation displacement field  $\mathbf{w}^P$  with  $\dim \mathcal{V}_h^P = \mathcal{N}$ , see Section 2.2.3. The idea of the RBM is to approximate the solution field with global parameter-independent basis functions and parameter-dependent coefficients, i.e.,

$$\mathbf{w}^P(\mathbf{x}^P; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \approx \sum_{n=1}^N a_n(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{v}_n(\mathbf{x}^P), \quad (4.1)$$

where  $N$  is the number of basis functions, typically much smaller than the dimension of the FE space, i.e.,  $N \ll \mathcal{N}$ . The basis functions,  $\{\mathbf{v}_n\}_{n=1}^N$ , span a subset of  $\mathcal{V}_h^P$  and can be obtained by applying POD on a set of pre-computed full solutions for different parameter values. More information on POD and how to compute the basis functions  $\{\mathbf{v}_n\}_{n=1}^N$  is given in Appendix B, and more information on the RBM can be found in [103].

By utilizing the POD space for both the trial and test space and inserting  $\mathbf{w}^P$  from Eq. (4.1) into Eqs. (2.11) and (2.12), the components for the

reduced internal force vector  $\mathbf{f}^{\text{POD}} \in \mathbb{R}^N$  and reduced global stiffness matrix  $\mathbf{K}^{\text{POD}} \in \mathbb{R}^{N \times N}$  can be derived as

$$f_i^{\text{POD}}(\mathbf{a}) := \int_{\Omega^{\text{P}}} ((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \mathbf{P}^{\text{P}}(\mathbf{F}^{\text{P}}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}}, \quad (4.2)$$

$$K_{ij}^{\text{POD}}(\mathbf{a}) := \int_{\Omega^{\text{P}}} ((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \boldsymbol{\mathfrak{A}}^{\text{P}}(\mathbf{F}^{\text{P}}) : ((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_j) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}}, \quad (4.3)$$

$$\mathbf{F}^{\text{P}}(\mathbf{a}) := \bar{\mathbf{F}} + \left( \sum_{n=1}^N a_n \nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_n \right) \mathbf{F}_{\boldsymbol{\mu}}^{-1}, \quad (4.4)$$

where  $\mathbf{a} = [a_1, \dots, a_N]^T$  is the column matrix of unknown coefficients to be solved for, and  $i, j = 1, \dots, N$  span over all basis functions. Analogously to Eqs. (2.33) and (2.34), the resulting non-linear system of equations,

$$\mathbf{f}^{\text{POD}}(\mathbf{a}) = \mathbf{0}, \quad (4.5)$$

can be solved using Newton method:

$$\begin{aligned} \mathbf{K}^{\text{POD}}(\mathbf{a}^m) \Delta \mathbf{a} &= -\mathbf{f}^{\text{POD}}(\mathbf{a}^m), \\ \mathbf{a}^{m+1} &= \mathbf{a}^m + \Delta \mathbf{a}. \end{aligned} \quad (4.6)$$

### 4.1.2 Empirical cubature method

Even though the solution field and linear system of equations have been reduced to dimension  $N \ll \mathcal{N}$ , computing the components of the force vector in Eq. (4.2) and global stiffness matrix in Eq. (4.3) still requires integrating over the RVE. For the full integration, a numerical quadrature rule (usually based on Gauss quadrature) with integration points and corresponding weights  $\{(\hat{\mathbf{x}}_q, \hat{w}_q)\}_{q=1}^{\hat{Q}}$ , where  $\hat{Q}$  is the total number of integration points, is employed, i.e.,

$$f_i^{\text{POD}}(\mathbf{a}) \approx \sum_{q=1}^{\hat{Q}} \hat{w}_q [((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \mathbf{P}^{\text{P}}(\mathbf{F}^{\text{P}}) |\det \mathbf{F}_{\boldsymbol{\mu}}|] \Big|_{\hat{\mathbf{x}}_q}, \quad (4.7)$$

for  $i = 1, \dots, N$ . Analogous expression holds for Eq. (4.3). For a fine mesh,  $\hat{Q}$  is very large and evaluating Eq. (4.7) thus leads to high computational costs, especially in cases in which evaluation of the constitutive law  $\mathbf{P}^{\text{P}}(\mathbf{F}^{\text{P}})$  is expensive, such as in plasticity. To address this issue, we employ ECM, which was proposed in Hernández et al. [48] for a fixed geometry, and generalize it to parameterized geometries.

The idea of ECM is to find a subset of points  $\{\mathbf{x}_q\}_{q=1}^Q \subset \{\hat{\mathbf{x}}_q\}_{q=1}^{\hat{Q}}$ , with  $Q \ll \hat{Q}$  among the set of all integration points, with corresponding weights  $\{w_q\}_{q=1}^Q$

that approximates Eq. (4.7) up to a user-defined error  $\varepsilon_{\text{ECM}}$ . To find such a subset that approximates Eq. (4.7) well for all admissible geometrical parameters  $\boldsymbol{\mu}$ , Eq. (4.7) is first rewritten as

$$\begin{aligned} f_i^{\text{POD}}(\mathbf{a}) &= \sum_{q=1}^{\hat{Q}} \hat{w}_q \left[ (\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) : \underbrace{(\mathbf{P}^{\text{P}}(\mathbf{F}^{\text{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-T} |\det \mathbf{F}_{\boldsymbol{\mu}}|)}_{\mathbf{W}^{\text{P}} :=} \right] \Big|_{\hat{\mathbf{x}}_q} \\ &= \sum_{q=1}^{\hat{Q}} \hat{w}_q [(\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) : \mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})] \Big|_{\hat{\mathbf{x}}_q}, \end{aligned} \quad (4.8)$$

where the weighted stress  $\mathbf{W}^{\text{P}}$  has been defined in analogy with Eq. (3.6). To remove the parameter dependence of the integrand in Eq. (4.8) (the term in the square bracket), the weighted stress is approximated by another reduced basis, i.e.,

$$\mathbf{W}^{\text{P}}(\mathbf{x}^{\text{P}}; \bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \approx \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mathbf{B}_l(\mathbf{x}^{\text{P}}), \quad (4.9)$$

where  $\{\mathbf{B}_l\}_{l=1}^L$  is a set of  $L$  basis functions obtained using POD, which are orthonormal with respect to  $L^2(\Omega^{\text{P}})$ , i.e.,

$$\int_{\Omega^{\text{P}}} \mathbf{B}_m : \mathbf{B}_n \, d\mathbf{x}^{\text{P}} = \delta_{mn}. \quad (4.10)$$

Inserting Eq. (4.9) into Eq. (4.8) and rearranging yields,

$$f_i^{\text{POD}}(\mathbf{a}) \approx \sum_{l=1}^L \alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \sum_{q=1}^{\hat{Q}} \hat{w}_q [(\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) : \mathbf{B}_l] \Big|_{\hat{\mathbf{x}}_q}, \quad i = 1, \dots, N. \quad (4.11)$$

Since Eq. (4.11) should be accurate for any choice of coefficients  $\alpha_l(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ , all the  $N \cdot L$  terms in Eq. (4.11) that approximate the integral have to be approximated as accurately as possible. Hence, the goal becomes to find a subset  $Q (\ll \hat{Q})$  of integration points with corresponding weights  $\{(\mathbf{x}_q, w_q)\}_{q=1}^Q$  that approximates Eq. (4.11) well, i.e.,

$$\sum_{q=1}^{\hat{Q}} \hat{w}_q [(\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) : \mathbf{B}_l] \Big|_{\hat{\mathbf{x}}_q} \approx \sum_{q=1}^Q w_q [(\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) : \mathbf{B}_l] \Big|_{\mathbf{x}_q}, \quad (4.12)$$

for  $i = 1, \dots, N$  and  $l = 1, \dots, L$ . These  $Q$  points and corresponding weights are found using a greedy algorithm, as proposed in [48], which is briefly summarized

in Appendix F. The algorithm is terminated when the mean squared error of all  $N \cdot L$  terms is less than a user-defined tolerance  $\varepsilon_{\text{ECM}}$ .

Compared to the original algorithm for a fixed geometry, as proposed in [48], the only differences are that the weighted stress  $\mathbf{W}^{\text{P}}$  is employed instead of the stress  $\mathbf{P}^{\text{P}}$  and that the parent domain  $\Omega^{\text{P}}$  is considered instead of a fixed domain  $\Omega$ . With the ECM integration rule, the hyperreduced force vector and global stiffness matrix are computed as

$$f_i^{\text{ECM}} := \sum_{q=1}^Q w_q [((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \mathbf{P}^{\text{P}}(\mathbf{F}^{\text{P}}) |\det \mathbf{F}_{\boldsymbol{\mu}}|] |_{\mathbf{x}_q}, \quad (4.13)$$

$$K_{ij}^{\text{ECM}} := \sum_{q=1}^Q w_q [((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \mathfrak{A}^{\text{P}}(\mathbf{F}^{\text{P}}) : ((\nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_j) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) |\det \mathbf{F}_{\boldsymbol{\mu}}|] |_{\mathbf{x}_q}. \quad (4.14)$$

**Remark 4.1.1** *The computational costs of the ECM greedy algorithm as proposed in [48] increase drastically with the number of selected integration points, since for every selected point a non-negative least squares problem needs to be solved. As pointed out in [22], rank-one updates can be used with the least squares solver for better efficiency. Such a refined version of the ECM algorithm was presented in [50]. For the numerical examples considered in this work, the original ECM algorithm in [48] was sufficiently fast and we did not use the algorithmically improved version.*

### 4.1.3 Effective quantities

Once the new set of integration points and weights is found, the integrands of Eqs. (4.13) and (4.14) only need to be evaluated at the points  $\{\mathbf{x}_q\}_{q=1}^Q$  during the solution of the reduced problem. This also means that the stress and stiffness fields are available at these points only. To compute the effective stress (cf. Eq. (2.16)), the most straightforward method is to use the integration rule obtained by ECM, i.e.,

$$\begin{aligned} \bar{\mathbf{P}} &= |\Omega^{\text{P}}|^{-1} \int_{\Omega^{\text{P}}} \mathbf{P}^{*\text{P}} |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\text{P}} \\ &\approx |\Omega^{\text{P}}|^{-1} \sum_{q=1}^Q w_q (\mathbf{P}^{*\text{P}} |\det \mathbf{F}_{\boldsymbol{\mu}}|) |_{\mathbf{x}_q}. \end{aligned} \quad (4.15)$$

Since the stress field  $\mathbf{P}^{*\text{P}}$  is known at all integration points  $\{\mathbf{x}_q\}_{q=1}^Q$ , the effective stress can be directly evaluated. The method yields very accurate results in the

examples considered below in Section 4.2. However, it should be noted that there is currently no guarantee that the integration rule found by ECM will generally be accurate for the computation of the effective stress. In general, the effective stress can have two sources of error as compared to the full solution: one comes from the solution of the reduced system and one from an inaccurate integration of the obtained stress field. The ECM integration points are selected such that the first error is minimized, but this also indirectly affects the second one to decrease, although not as quickly. This can be observed in the results of the first numerical example presented in Section 4.2.1. To ensure an accurate integration of the effective stress, it could be included into the ECM algorithm as a criterion. This idea is implemented in the hyperreduction algorithm proposed in Section 5.2.2 for second-order computational homogenization.

As discussed in Section 2.2.1, derivatives  $\nabla_{\bar{\mathbf{F}}}\mathbf{w}^{*P}$  are needed to find the effective stiffness  $\bar{\mathbf{A}}$ , see Eq. (2.17). For each component of  $\bar{\mathbf{F}}$ , the linear tangent problem of Eq. (2.20) needs to be solved. By employing the trial space of the fluctuation field for the auxiliary function  $\mathbf{q}_{kl}$  (which represents the sensitivity of  $\mathbf{w}^{*P}$  with respect to the  $kl$ -th component of  $\bar{\mathbf{F}}$ ), i.e.,

$$\mathbf{q}_{kl} = \sum_{n=1}^N q_n \mathbf{v}_n(\mathbf{x}^P), \quad (4.16)$$

and the integration rule found by ECM, the following linear system of equations results for each  $\bar{F}_{kl}$ :

$$\mathbf{K}^{*P}\mathbf{q} = \mathbf{b}, \quad (4.17)$$

where  $\mathbf{q} = [q_1, \dots, q_N]^T$  is the column matrix of unknowns to be solved for and

$$K_{ij}^{*P} = \sum_{q=1}^Q w_q [((\nabla_{\mathbf{x}^P}\mathbf{v}_i)\mathbf{F}_{\mu}^{-1}) : \mathbf{A}^{*P} : ((\nabla_{\mathbf{x}^P}\mathbf{v}_j)\mathbf{F}_{\mu}^{-1}) |\det \mathbf{F}_{\mu}|]_{\mathbf{x}_q}, \quad (4.18)$$

$$b_i = - \left( \sum_{q=1}^Q w_q [((\nabla_{\mathbf{x}^P}\mathbf{v}_i)\mathbf{F}_{\mu}^{-1}) : \mathbf{A}^{*P} |\det \mathbf{F}_{\mu}|]_{\mathbf{x}_q} \right) : \mathbf{E}_{kl}. \quad (4.19)$$

Note that the matrix  $\mathbf{K}^{*P} \in \mathbb{R}^{N \times N}$  is exactly the same as the global stiffness matrix  $\mathbf{K}$  of Eq. (4.14) evaluated at the solution  $\mathbf{w}^{*P}$ . After solving the tangent problem for each component of  $\bar{\mathbf{F}}$ , the effective stiffness  $\bar{\mathbf{A}}$  can be computed (in index notation) as

$$\bar{\mathbf{a}}_{ijkl} = |\Omega^P|^{-1} \sum_{q=1}^Q w_q \left( \frac{\partial P_{ij}^{*P}}{\partial \bar{F}_{kl}} |\det \mathbf{F}_{\mu}| \right) \Big|_{\mathbf{x}_q}, \quad (4.20)$$

where

$$\frac{\partial \mathbf{P}^{*\mathbb{P}}}{\partial \bar{\mathbf{F}}_{kl}} = \mathfrak{A}^{*\mathbb{P}} : (\mathbf{E}_{kl} + ((\nabla_{\mathbf{x}^{\mathbb{P}}} \mathbf{q}_{kl}) \mathbf{F}_{\boldsymbol{\mu}}^{-1})). \quad (4.21)$$

#### 4.1.4 Summary

For convenience, the offline-online decomposition for constructing and solving the surrogate model, referred to as PODECM, is summarized in Algorithm 2.

---

**Algorithm 2** Offline-online decomposition of the proposed PODECM framework with microstructures parameterized with external loading  $\bar{\mathbf{F}}$ , microstructural material parameters  $\boldsymbol{\lambda}$  and geometrical features  $\boldsymbol{\mu}$ .

---

##### Offline Stage:

- 1: Define a parent domain  $\Omega^{\mathbb{P}}$  and its finite element discretization.
- 2: Generate parameter samples  $\{\bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}\}_{i=1}^{N_{\text{train}}}$  from a random distribution on  $\mathbb{P}$ , where  $N_{\text{train}}$  is the number of training samples.
- 3: For each different set of geometrical parameters  $\boldsymbol{\mu}^{(i)}$ , solve the auxiliary problem in Eq. (3.2) to obtain the transformation map  $\Phi_{\boldsymbol{\mu}^{(i)}}$ .
- 4: Compute  $\mathbf{F}_{\boldsymbol{\mu}^{(i)}}^{-1}$  and  $\det \mathbf{F}_{\boldsymbol{\mu}^{(i)}}$  for each parameter sample  $\boldsymbol{\mu}^{(i)}$ , then run full RVE simulations (Eqs. (2.11) and (2.12)) for  $(\bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})$  and collect fluctuation displacements  $\mathbf{w}^{\mathbb{P}}(\mathbf{x}^{\mathbb{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})$  and weighted stress snapshots  $\mathbf{W}^{\mathbb{P}}(\mathbf{x}^{\mathbb{P}}; \bar{\mathbf{F}}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})$ .
- 5: Compute POD for the fluctuation displacements and weighted stresses, see Eqs. (4.1) and (4.9).
- 6: Run ECM algorithm and find integration points and weights, see Eq. (4.12) and Appendix F.
- 7: Assemble the reduced system matrix and forcing vector for the auxiliary problem in Eq. (3.4) by applying POD and DEIM, see Section 3.1.1.

##### Online Stage:

- 1: Given a new parameter set  $(\bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ , solve the reduced auxiliary problem in Eq. (3.4) and compute  $\mathbf{F}_{\boldsymbol{\mu}^*}^{-1}$  and  $\det \mathbf{F}_{\boldsymbol{\mu}^*}$ .
  - 2: Solve reduced problem for  $(\bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$  with Eqs. (4.13) and (4.14).
  - 3: Compute effective stress using Eq. (4.15).
  - 4: Solve the linear tangent problem Eq. (4.17) for each component of  $\bar{\mathbf{F}}$ .
  - 5: Compute components of the effective stiffness with Eqs. (4.20) and (4.21).
-

## 4.2 Numerical examples

The proposed framework, referred to as PODECM, is first tested on a non-linear composite microstructure under various loading conditions and analyzed in depth regarding its capabilities and accuracy. The RVE consists of an elastoplastic matrix with stiff inclusions of variable size and is considered under non-monotonic loading. The surrogate model is analyzed in terms of the number of basis functions of the fluctuation displacement field  $N$ , number of basis functions of the weighted stress  $L$  and the ECM integration error tolerance  $\varepsilon_{\text{ECM}}$ . Subsequently, a two-scale problem involving a porous microstructure under non-monotonic loading conditions and varying porosities is studied to illustrate the accuracy and speed-up of PODECM in a two-scale setting.

All experiments are defined in two dimensions under plane strain conditions. The RVEs are assumed to be of size  $[0, 1]^2$  and all quantities are assumed to be normalized and hence dimensionless. Since the macroscopic deformation gradient  $\bar{\mathbf{F}}$  can always be decomposed into a rotation  $\bar{\mathbf{R}}$  and a symmetric stretch tensor  $\bar{\mathbf{U}}$  with a polar decomposition, i.e.,  $\bar{\mathbf{F}} = \bar{\mathbf{R}}\bar{\mathbf{U}}$ , see Remark 2.1.1, it is sufficient to generate training data for the stretch tensor  $\bar{\mathbf{U}}$ , having only 3 independent components (6 in 3D). In all examples, the material parameters  $\boldsymbol{\lambda}$  are assumed to be constant, but the geometry can vary with geometrical parameters  $\boldsymbol{\mu}$ .

To measure the quality of the approximation, the following error measures to compare the full FE simulations against PODECM solutions are defined:

1. Error of effective stress

$$\varepsilon_{\bar{\mathbf{P}}} = \frac{\sum_{k=1}^K \|\bar{\mathbf{P}}^{\text{truth}}(\bar{\mathbf{U}}^k) - \bar{\mathbf{P}}^{\text{PODECM}}(\bar{\mathbf{U}}^k)\|_F}{\sum_{k=1}^K \|\bar{\mathbf{P}}^{\text{truth}}(\bar{\mathbf{U}}^k)\|_F}, \quad (4.22)$$

where  $\bar{\mathbf{P}}^{\text{PODECM}}(\bar{\mathbf{U}}^k)$  and  $\bar{\mathbf{P}}^{\text{truth}}(\bar{\mathbf{U}}^k)$  denote the effective stress obtained with PODECM and FE for  $\bar{\mathbf{U}}^k$ ,  $\|\bullet\|_F$  denotes the Frobenius norm,  $K$  is the total number of loading steps and  $\bar{\mathbf{U}}^k$  is the applied external load at load step  $k$ .

2. Error of fluctuation field

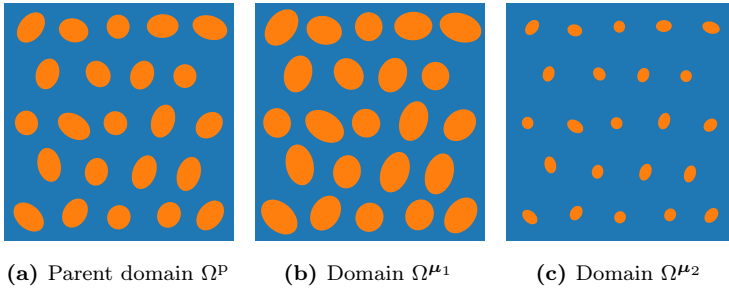
$$\varepsilon_{\mathbf{w}} = \frac{\sum_{k=1}^K \|\mathbf{w}^{\text{truth}}(\bar{\mathbf{U}}^k) - \mathbf{w}^{\text{PODECM}}(\bar{\mathbf{U}}^k)\|_{\mathcal{V}}}{\sum_{k=1}^K \|\mathbf{w}^{\text{truth}}(\bar{\mathbf{U}}^k)\|_{\mathcal{V}}}, \quad (4.23)$$

where  $\mathbf{w}^{\text{PODECM}}$  and  $\mathbf{w}^{\text{truth}}$  denote the fluctuation displacement field obtained with PODECM and FE, and  $\|\bullet\|_{\mathcal{V}}^2 = (\bullet, \bullet)_{\mathcal{V}}$ , see Eq. (2.10). Recall that the integral in Eq. (2.10) is defined over the parameterized domain  $\Omega^{\boldsymbol{\mu}}$ .



### 4.2.1 Elasto-plastic composite RVE with random inclusions

The considered RVE in this example consists of two phases, an elasto-plastic matrix and stiff elastic inclusions. The geometry of the parent domain is shown in Fig. 4.1a, where the volume fraction of the inclusions is 23.4%. For the geometrical parameterization, one geometrical parameter  $\boldsymbol{\mu} = \{\zeta\}$  that scales the size of the inclusions uniformly (and is proportional to the volume fraction of the inclusions) is introduced, see Figs. 4.1b and 4.1c showing two example domains for distinct values of  $\zeta$ . Six-noded quadratic triangular elements are used in conjunction with three quadrature points per element. In total, the mesh has 62194 degrees of freedom, 15450 triangular elements and 46350 quadrature points.

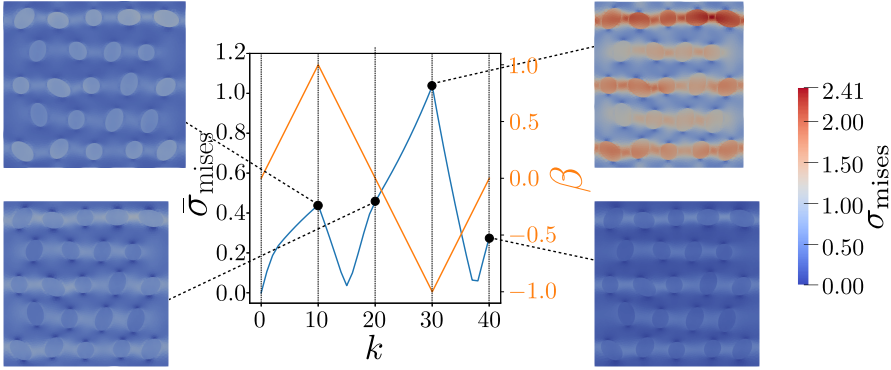


**Figure 4.1:** Parent with two parameterized domains and simulation mesh. (a) The parent domain consists of a matrix material (blue) with 23 random elliptical inclusions (orange). The problem has one geometrical parameter  $\zeta$  that scales all ellipses uniformly ( $\zeta = 1$  for parent domain). (b) A parameterized domain for  $\zeta = 1.2$  and (c) for  $\zeta = 0.5$ .

For the constitutive model of both matrix and inclusion the small-strain  $J_2$ -plasticity model with linear isotropic hardening is chosen and extended to large strains with the method presented in Cuitino and Ortiz [24]. The details of the plasticity model are provided in Appendix A.2. For the matrix, the following material parameters are selected: a Young's modulus  $E = 10$ , Poisson's ratio  $\nu = 0.3$ , yield stress  $\sigma_{y0} = 0.2$  and hardening constant  $H = 5$ . For the inclusions,  $E = 100$  and  $\nu = 0.3$  are selected, corresponding to a stiffness contrast ratio of 10 between both components. Since no plastic deformation is assumed for the inclusions, their yield stress is set to a large value such that yielding never occurs.

Three loading  $\bar{U}_{xx}$ ,  $\bar{U}_{xy}$ ,  $\bar{U}_{yy}$  and one geometrical  $\zeta$  parameters are considered with bounds  $\zeta \in [0.5, 1.2]$ ,  $\bar{U}_{xx} \in [0.9, 1.1]$ ,  $\bar{U}_{yy} \in [0.9, 1.1]$  and  $\bar{U}_{xy} \in [-0.1, 0.1]$ . Through  $\zeta$ , the volume fraction of the inclusions is varied from 5.85% to 33.7%. For each sample, the macroscopic stretch tensor  $\bar{\mathbf{U}} = \begin{bmatrix} \bar{U}_{xx} & \bar{U}_{xy} \\ \bar{U}_{xy} & \bar{U}_{yy} \end{bmatrix}$  is

applied to the RVE with  $\beta(k)\bar{\mathbf{U}}$ , where  $\beta(k)$  is a piecewise linear amplitude function with load step  $k \in \{0, 1, \dots, 40\}$ . The chosen amplitude function is shown in Fig. 4.2 in orange, together with the evolution of the effective von Mises stress for an example with  $\zeta = 1.010$ ,  $\bar{U}_{xx} = 1.1$ ,  $\bar{U}_{yy} = 1.0$ ,  $\bar{U}_{xy} = 0.0$ , as well as the local von Mises stress fields at steps  $k = \{10, 20, 30, 40\}$ . The von Mises stress field is computed from the 1PK stress as follows: first, the Cauchy stress is computed with  $\boldsymbol{\sigma} = \mathbf{P}\mathbf{F}^T(\det \mathbf{F})^{-1}$ ; then the von Mises stress is given by  $\sigma_{\text{mises}} = \sqrt{\frac{3}{2} \text{Dev}(\boldsymbol{\sigma}) : \text{Dev}(\boldsymbol{\sigma})}$ , with  $\text{Dev}(\boldsymbol{\sigma}) := \boldsymbol{\sigma} - \frac{1}{3} \text{tr}(\boldsymbol{\sigma})\mathbf{I}$  the deviatoric stress. Even though only macroscopic strains of up to 10% are applied, local strains reach values up to 83% (not shown in Fig. 4.2).

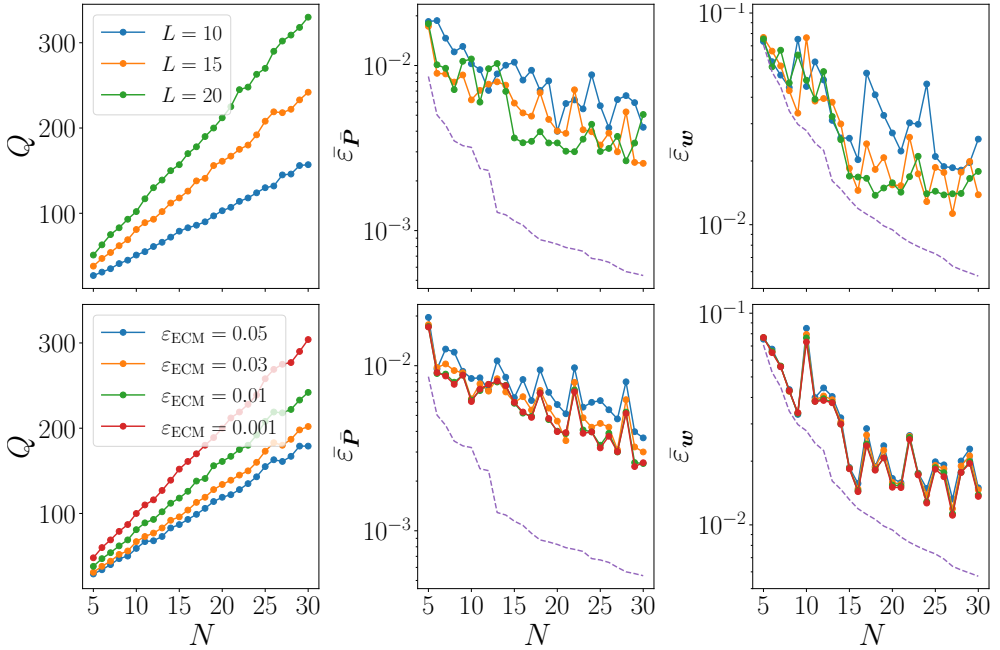


**Figure 4.2:** Macroscopic von Mises stress  $\bar{\sigma}_{\text{mises}}$  and amplitude function  $\beta$  plotted over  $k$  for  $\zeta = 1.010$ ,  $\bar{U}_{xx} = 1.1$ ,  $\bar{U}_{yy} = 1.0$ ,  $\bar{U}_{xy} = 0.0$ . At  $k = \{10, 20, 30, 40\}$ , microstructural von Mises stress fields  $\sigma_{\text{mises}}$  are shown. The von Mises stress is non-zero at  $k = \{20, 40\}$  due to residual plastic deformation.

**Results** In total  $N_{\text{train}} = 20$  samples are generated from a Sobol sequence to train PODECM whereas 100 testing samples are generated from a uniform distribution to test PODECM. Each sample consists of 40 snapshots for each load step.

The accuracy and speed-up of PODECM depends on the number of basis functions used for the fluctuation displacement field  $N$  and the number of quadrature points  $Q$ . While  $N$  is typically chosen directly,  $Q$  depends on the choice of the number of basis functions used for the weighted stress  $L$  and the ECM integration error  $\varepsilon_{\text{ECM}}$ .

To study the influence of  $L$  on the resulting number of quadrature points  $Q$  and mean errors in effective stress and fluctuation field on the testing dataset, several combinations of  $N$  and  $L$  for a fixed  $\varepsilon_{\text{ECM}} = 10^{-2}$  are tested, with resulting errors shown in the top row of Fig. 4.3. The projection error



**Figure 4.3:** The left column shows the number of quadrature points  $Q$  obtained from ECM for different choices of number of basis functions of fluctuation field  $N$ , number of basis functions of weighted stress  $L$ , and ECM integration error  $\varepsilon_{ECM}$ . The middle and right columns show the average errors of the effective stress and the fluctuation field when tested on the testing data for different choices of  $N$ ,  $L$  and  $\varepsilon_{ECM}$ . The top row assumes a fixed  $\varepsilon_{ECM} = 0.01$ , while the bottom row assumes a fixed  $L = 15$ .

(for  $N$  basis functions and using full integration) is shown as well. It can be clearly seen that the number of quadrature points  $Q$  increases drastically with increasing  $N$  and  $L$ , as more information needs to be integrated accurately. In fact, a roughly linear relationship  $Q \propto NL$  can be recognized. For the mean errors, a higher  $L$  leads to better results on average, although we observe that errors fluctuate significantly, and for some values of  $N$  a worse approximation is obtained with a higher  $L$ . This occurs since the ECM algorithm is a greedy algorithm, meaning that it does not necessarily find an optimal set of integration points. When more basis functions are included into the algorithm, a completely different set of points may be found that finally leads to a worse approximation. It can furthermore be observed that the gap between the projection error and the PODECM solution grows larger for increasing  $N$ . This is because the basis functions typically become more oscillatory and difficult to approximate with higher  $N$ , see, e.g., [45, 46], and thus require significantly more quadrature points for a good approximation. It is interesting that the gap for the errors in the fluctuation field are smaller than the ones in the effective stress, i.e.,

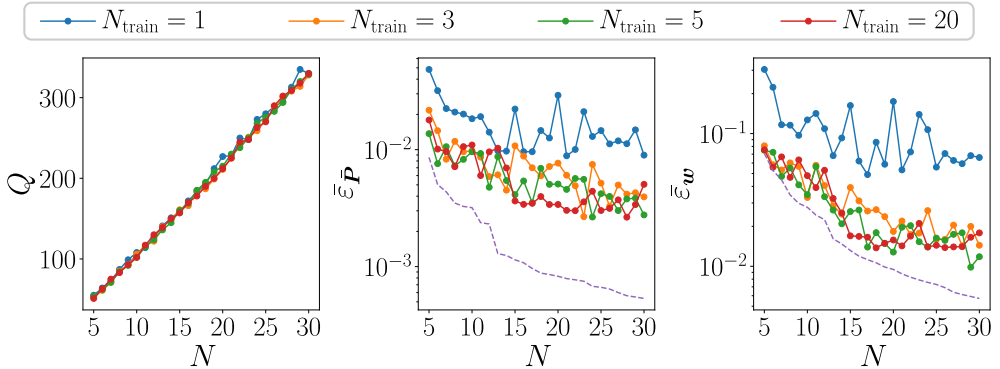
the difference between the computed PODECM and the projection errors are much higher for the effective stress as compared to the fluctuation field. This happens because the ECM integration points and weights are primarily selected to integrate the weak form accurately, and using them to compute the effective stress introduces an additional approximation error, cf. Section 4.1.3.

Several combinations of  $N$  and  $\varepsilon_{\text{ECM}}$  for a fixed  $L = 15$  are next tested to study the influence of  $\varepsilon_{\text{ECM}}$  on the number of quadrature points and approximation errors. Obtained results are shown in the bottom row of Fig. 4.3. Similarly to the previous analysis, a lower  $\varepsilon_{\text{ECM}}$  leads to more quadrature points  $Q$  and a lower mean error in the effective stress and fluctuation field on average, as the integrals are approximated more accurately. Interestingly, lowering the tolerance from 0.01 to 0.001 does not significantly improve the approximation quality, even though substantially more quadrature points are included, meaning that the errors can be attributed to the higher modes of the weighted stress (the additional quadrature points barely contain any information). Therefore, choosing a tolerance smaller than  $\varepsilon_{\text{ECM}} = 0.01$  leads to no improvement.

From Fig. 4.3 we further observe that the errors of the fluctuation field are considerably higher (order of magnitude) than the errors of the effective stresses. This results from the fact that the POD basis functions aim to minimize the  $H^1(\Omega^P)$  error, and thus approximate the field accurately on average rather than locally, suggesting a favorable approximation for averaged quantities such as the effective stress.

To test the data efficiency of PODECM, the reduced model is trained for different numbers of training data  $N_{\text{train}} = \{1, 3, 5, 20\}$  with  $L = 20$  and  $\varepsilon_{\text{ECM}} = 0.01$ . The number of integration points and corresponding errors in the effective stress and fluctuations are shown in Fig. 4.4. It can clearly be seen that the number of integration points barely changes for different  $N_{\text{train}}$ , and that the errors converged already for  $N_{\text{train}} = 3$ . No noticeable improvements with  $N_{\text{train}} = 5$  and 20 can be observed, showing that PODECM is very data efficient. Even with  $N_{\text{train}} = 1$ , the errors in the effective stress are below 5%.

To conclude, the more basis functions  $N$  and  $L$  are used and the lower the integration error  $\varepsilon_{\text{ECM}}$  is chosen, the more accurate the final result is. However, at the same time the surrogate model grows in size and the speed-up decreases. A user must thus make a compromise between accuracy and cost. For this example, the speed-up correlates nearly linearly with  $Q$ , i.e., if the number of quadrature points is reduced by a factor of 100, this results in a speed-up of roughly 100. In contrast, the number of basis functions  $N$  only plays a minor role for the speed-up. For this example, use of  $N = 20$ ,  $L = 20$  and  $\varepsilon_{\text{ECM}} = 0.01$  lead to a reduction in the number of degrees of freedom from 62194 to 20 and quadrature points from 46350 to 212, suggesting a speed-up on the order of roughly 200.



**Figure 4.4:** Errors of PODECM for different numbers of training samples. The number of quadrature points does not change. For  $N_{\text{train}} = 3$ , the errors in effective stress and fluctuation displacement are already converged, showing that PODECM is very data efficient.

**Remark 4.2.1** *Intuitively, one might consider not selecting  $L$  as an independent parameter, but as a (non-linear) function of  $N$  with  $L \geq N$ , since the weighted stress acts as a non-linear function on the fluctuation field, and thus is expected to require more basis functions. However, for  $L = N$ , one obtains a roughly quadratic relationship for the number of quadrature points  $Q$  and  $N$  with  $Q \propto LN = N^2$ . If a higher number of  $N$  is necessary to have a sufficiently large solution space,  $Q$  quickly becomes very large and speed-ups of PODECM become very small. By treating  $L$  as an independent parameter and allowing  $L < N$ , the resulting number of integration points can be controlled and decreased. Furthermore, we observed in the numerical tests that, if a maximum number of integration points is specified, we often obtained better results for  $N > L$  rather than  $N \leq L$ , as long as  $L$  is large enough.*

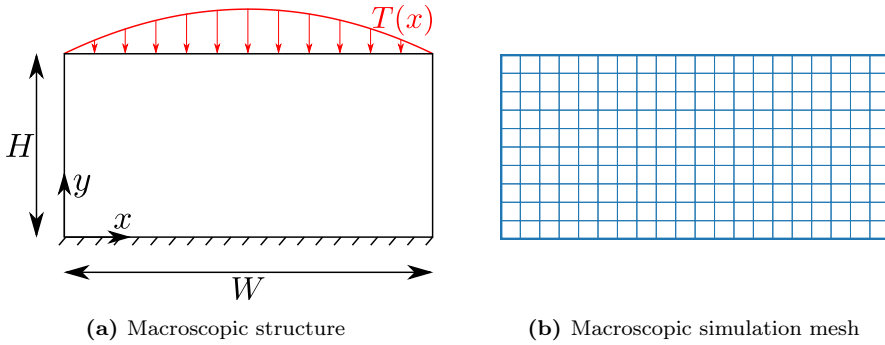
## 4.2.2 Two-scale compression with porous microstructure

In the second example, the macroscopic structure, depicted in Fig. 4.5 together with the employed simulation mesh, is compressed under an external loading  $T(x)$ . Here, we assume  $H = 1$ ,  $W = 2$ ,  $T(x) = \bar{T}(1 - (2x/W - 1)^2)$ ,  $x \in [0, W]$ , with  $\bar{T}$  the magnitude of the applied load. The simulation mesh has 1322 degrees of freedom, 200 8-noded quadrilateral elements, and in total 800 quadrature points. The structure is assumed to have a porous microstructure, modelled by the parameterized RVE, shown in Fig. 4.6, and the same material model as the matrix material in the previous example. Such microstructures (with circular holes, i.e.,  $a = b$ ) have been considered in several works, see, e.g., [9, 150, 2, 114], due to their auxetic behavior under compression, i.e., negative Poisson's ratio. During compression, the center part of the material

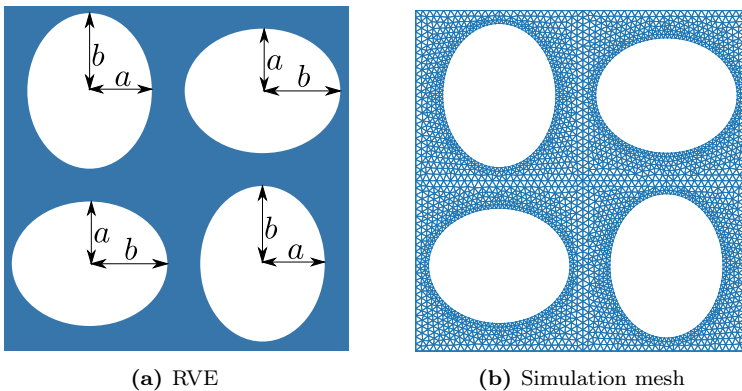
starts to rotate, thus pulling the material from the sides inwards. In our work, we define two independent parameters, namely the volume fraction of the voids,  $v_{\text{void}} := 4\pi ab$ , and the ratio of the semi-major axis  $b$  to semi-minor axis  $a$  of each hole,  $\kappa := b/a$ . The semi-minor axis  $a$  and semi-major axis  $b$  depend on  $v_{\text{void}}$  and  $\kappa$  as

$$a(v_{\text{void}}, \kappa) = \sqrt{v_{\text{void}}/(4\pi\kappa)}, \quad (4.24)$$

$$b(v_{\text{void}}, \kappa) = \kappa \cdot a(v_{\text{void}}, \kappa). \quad (4.25)$$



**Figure 4.5:** Geometry (a) and mesh (b) of the considered macroscopic structure. The body is fixed on the bottom and an external compression force  $T$  is applied on the top. The mesh consists of 1322 degrees of freedom, 200 8-noded quadrilateral elements and 4 quadrature points per element.



**Figure 4.6:** Geometry (a) and mesh (b) of the porous RVE. The elliptical holes are all characterized by the same semi-minor axis  $a$  and semi-major axis  $b$ , and are parameterized by the volume fraction of the pores  $v_{\text{void}}$  and the ratio  $\kappa = b/a$ . The employed simulation mesh has 21042 degrees of freedom, 4964 6-noded triangular elements and in total 14892 quadrature points. The parent domain corresponds to  $\kappa = 1.25$  and  $v_{\text{void}} = 0.45$ .

Depending on the values of the parameters, the resulting effective properties change significantly. To illustrate this, linear analyses of this RVE for different parameters have been carried out, similarly to [151], where a small compression in the  $y$ -direction with  $\Delta u_y = 0.001$  has been applied, while allowing the RVE to contract freely in the  $x$ -direction. With the resulting displacement in the  $x$ -direction,  $\Delta u_x$ , the Poisson's ratio in the initial state can be estimated as

$$\nu^{\text{eff}} = -\frac{\Delta u_x}{\Delta u_y}. \quad (4.26)$$

Similarly, the initial Young's modulus is estimated as

$$E^{\text{eff}} = \frac{\bar{P}_{yy}}{\Delta u_y}, \quad (4.27)$$

where  $\bar{P}_{yy}$  is the  $yy$ -component of the effective stress. For parameter ranges  $v_{\text{void}} \in [0.4, 0.5]$  and  $\kappa \in [1.01, 1.5]$ , the estimated Poisson's ratio and Young's modulus are plotted in Fig. 4.7. It can be observed that removing material (by increasing  $v_{\text{void}}$  while keeping  $\kappa$  fixed) or increasing  $\kappa$  while keeping  $v_{\text{void}}$  fixed both lead to a softer response with lower Young's modulus. While the Poisson's ratio is barely affected by  $v_{\text{void}}$  for values of  $\kappa$  close to 1, the effect becomes apparent for larger values of  $\kappa$ . In particular, for  $\kappa \geq 1.4$ , the Poisson's ratio changes from a positive value to a negative one. Therefore, by tuning  $v_{\text{void}}$  and  $\kappa$ , the RVE behavior can be significantly modified.

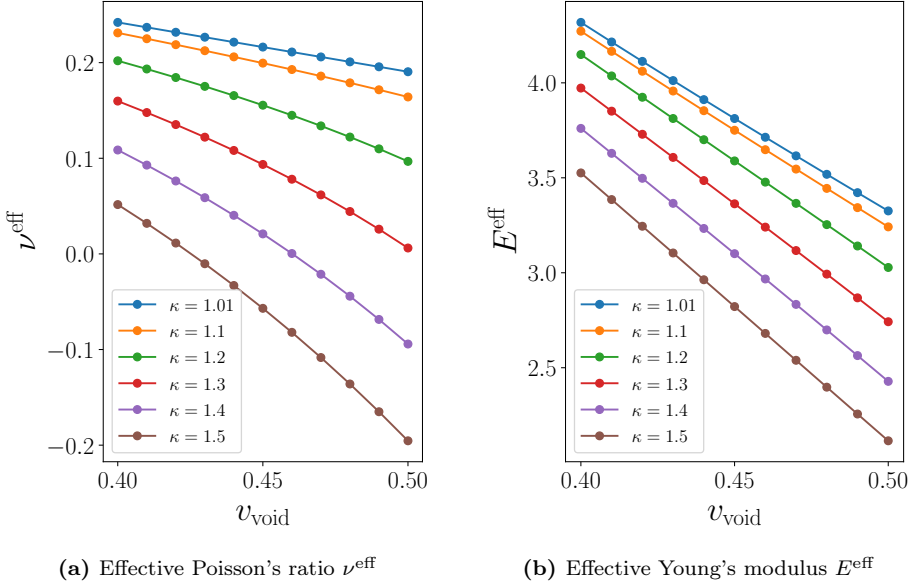
The parameters  $v_{\text{void}}$  and  $\kappa$  are chosen to vary smoothly through the macrostructural domain as

$$\kappa(x, y) = 1.5 - (1.5 - 1.01)y, \quad (4.28)$$

$$v_{\text{void}}(x, y) = 0.4 + (0.5 - 0.4)(1 - x)^2. \quad (4.29)$$

The parent mesh is selected with  $\kappa = 1.25$  and  $v_{\text{void}} = 0.45$ . The external loading is applied in  $K = 50$  load steps. In the first 25 load steps, the applied load  $\bar{T}$  increases linearly from 0 to 0.2. In the next 25 steps  $\bar{T}$  is decreased linearly from 0.2 to 0.

**Results** Several two-scale simulations with different PODECM surrogate models are run and compared to the full reference FE2 solution. To compare the accuracy of the surrogate solutions, the compliance  $C := \int_{\Gamma} T(x)u_y(x)dx$ , where  $\Gamma$  denotes the top horizontal edge of the macrostructure and  $u_y$  its vertical displacement, is computed at every load step. The compliance is an important quantity, often employed in optimization problems. Subsequently,



**Figure 4.7:** Initial effective Poisson's ratio (a) and Young's modulus (b) of RVE for different values of  $v_{\text{void}}$  and  $\kappa$ .

the relative error in compliance  $\varepsilon_C$  and the relative error averaged over all load steps  $\bar{\varepsilon}_C$  are defined as

$$\varepsilon_{C,k} := \frac{|C_k - C_k^{\text{FE2}}|}{|C_k^{\text{FE2}}|}, \quad \bar{\varepsilon}_C := \frac{1}{K} \sum_{n=1}^K \varepsilon_{C,k}, \quad (4.30)$$

where the subscript  $k$  denotes the  $k$ -th load step and  $C^{\text{FE2}}$  is the compliance computed with the full solution.

The tested PODECM models are generated for different numbers of training samples  $N_{\text{train}}$  and number of basis functions  $N$ . The training data is sampled from  $\bar{U}_{xx} \in [0.85, 1]$ ,  $\bar{U}_{yy} \in [0.85, 1]$ ,  $\bar{U}_{xy} \in [-0.15, 0.15]$ ,  $v_{\text{void}} \in [0.4, 0.5]$  and  $\kappa \in [1.01, 1.5]$  with a Sobol sequence. Each sample consists of 40 load steps, where the sampled macroscopic stretch tensor is applied to the RVE with a piecewise linear amplitude function that is linearly increased from 0 to 1 for the first 20 load steps and then linearly decreased from 1 to 0 for the last 20 steps. For the ECM algorithm, the number of basis functions for the weighted stress  $L$  and the integration error  $\varepsilon_{\text{ECM}}$  are assumed as fixed with  $L = 20$  and  $\varepsilon_{\text{ECM}} = 0.01$  for all models. The exact settings for each surrogate model are summarized in Table 4.1, alongside the averaged relative error in compliance  $\bar{\varepsilon}_C$ , and the run time and online speed-up in comparison to the full FE2 solution. For comparison, the total number of degrees of freedom



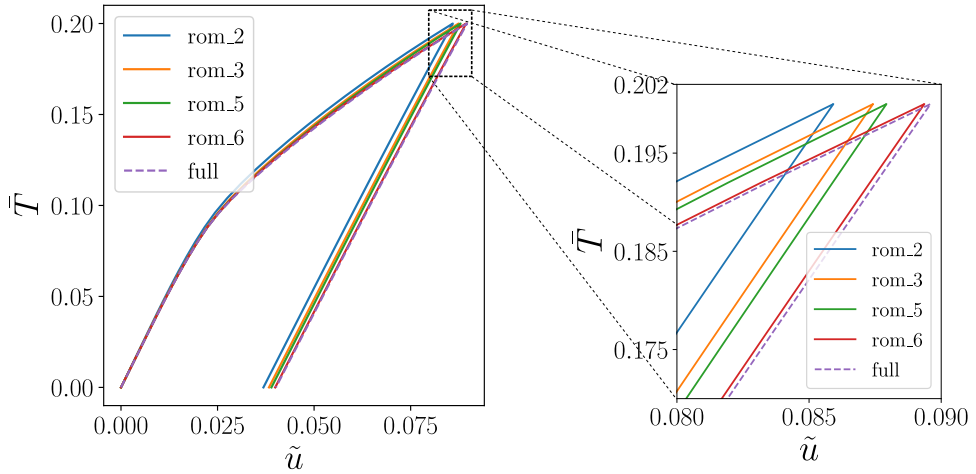
and quadrature points of the full FE model of the RVE are also provided. As can be seen, for all surrogate models the number of quadrature points are reduced by a factor of up to 100 which results in high speed-ups up to 100 times, while errors are below 5% for all models. By increasing  $N$  from 10 to 50 for rom\_1 to rom\_5, the error decreases from 4.74% to 1.54%, whereas the speed-up reduces from 95 to 29. Including more training samples with the same number of basis functions  $N = 50$  (from  $N_{\text{train}} = 20$  for rom\_5 to  $N_{\text{train}} = 50$  for rom\_6) improves the error from 1.54% to 0.39% while the speed-up remains roughly the same. This means that by increasing the sample size, the first 50 basis functions contain more general information that result in a better approximation. When more than 50 samples are used for rom\_7 and rom\_8, the error remains roughly the same, implying that 50 samples are sufficient for this problem and including more training data does not improve the results.

**Table 4.1:** Summary of results for full FE2 and different PODECM surrogate models. All computations are executed using 20 cores of an Intel Platinum 8260. Reduced order models are generated for different numbers of training samples  $N_{\text{train}}$  and basis functions of the displacement  $N$ . The number of quadrature points  $Q$  follow from the ECM algorithm with a fixed number of basis functions of the weighted stress field  $L = 20$  and an integration error of  $\varepsilon_{\text{ECM}} = 0.01$ . All reduced order models achieve errors less than 5% with speed-ups up to 100 times. By generating more training data and maintaining the same  $N$  (rom\_5 and rom\_6), better results are achieved. However, using more than 50 training samples (rom\_7 and rom\_8) leads to no further improvements, suggesting that 50 samples are sufficient for this problem.

	$N_{\text{train}}$	$N$	$Q$	$\bar{\varepsilon}_C$	run time	online speed-up
full	-	21042	14892	-	12573s	-
rom_1	20	10	132	4.74%	133s	94.53
rom_2	20	20	259	4.66%	195s	64.48
rom_3	20	30	372	2.28%	260s	48.36
rom_4	20	40	490	2.19%	337s	37.31
rom_5	20	50	595	1.54%	438s	28.71
rom_6	50	50	591	0.39%	426s	29.51
rom_7	80	50	579	0.58%	422s	29.79
rom_8	100	50	577	0.65%	411s	30.59

In Fig. 4.8 the force-displacement curve is shown for the FE2 and a few selected surrogate solutions. The displacement is defined as the vertical displacement at the mid point of the top edge (which is also the maximal displacement). It can be observed that all surrogate models underpredict the displacement, indicating that the surrogate models overpredict the stiffness of the macrostructure.

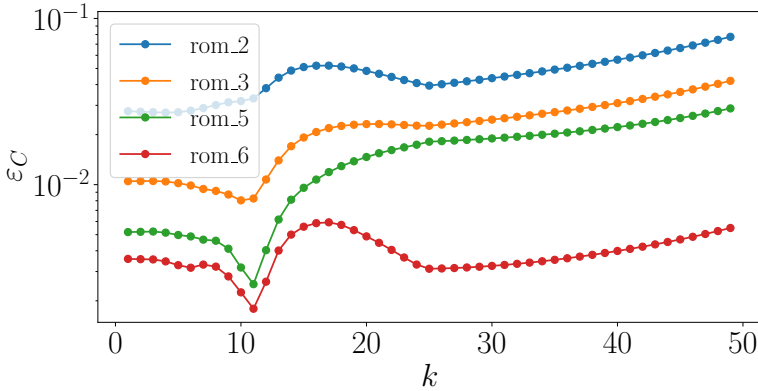
The relative error in compliance is plotted over the load steps  $k$  in Fig. 4.9.



**Figure 4.8:** Force-displacement curves of two-scale simulations. The displacement  $\tilde{u}$  is defined as the vertical displacement of the mid point on the top edge, cf. Fig. 4.5a. The structure starts deforming plastically for  $\bar{T} > 0.07$  and a residual displacement of roughly  $\tilde{u} = 0.04$  remains after unloading. All surrogate models achieve accurate results, although they generally predict a slightly stiffer response than the full FE2 solution.

For all models, the error slowly increases over  $k$ . The reason for this behavior is that all training samples are generated for simple loading cases, where the macroscopic stretch tensor is linearly varied in only one direction throughout the entire simulation. In the macroscopic simulation, the macroscopic stretch tensor for one integration point generally does not evolve along one direction, but changes its direction continuously, leading to highly complicated deformation paths and histories that are not included in the training data. To tackle this problem, random loading paths during training could be used, as performed in, e.g., [94, 141], to generate a more general surrogate model. Solely increasing the number of samples from 20 to 50 (rom\_5 to rom\_6) also decreases the observed errors to less than 1% for all load steps. This shows that PODECM generalizes well to loading paths that are not part of the training data.

Regarding the offline run times (for one core of an Intel Platinum 8260): computing each training sample takes approximately 2–3 minutes. With the training data available, computing the POD for each model takes up to around one minute. Finally, selecting the ECM points takes up to roughly two minutes for the PODECM models with  $N = 50$ .



**Figure 4.9:** Relative error  $\varepsilon_C$  in compliance over load step  $k$ . All surrogate models (specified in Table 4.1) begin with a lower error which slowly grows with  $k$ . By increasing the sample size of the training data (comparing rom\_5 and rom\_6), the prediction becomes more accurate for the same number of basis functions; the errors for all load steps for rom\_6 are below 1%.

## 4.3 Conclusions

In this chapter, we developed a reduced order model, termed PODECM, by combining the proper orthogonal decomposition (POD), the empirical cubature method (ECM) and a geometrical transformation method. Unlike the model presented in Chapter 3, PODECM can treat history-dependent material behavior and accelerate the microscopic problem for various material and geometrical parameters, making it viable for two-scale optimization problems.

The framework was first tested on a single-scale problem involving an RVE of a composite microstructure that consisted of a soft elasto-plastic matrix with stiff inclusions of variable size, controlled by a single geometrical parameter. With PODECM, the number of degrees of freedom and integration points was reduced to a fraction of the full FE model while maintaining a high accuracy in effective stress. The performance of PODECM was further evaluated for a two-scale simulation, in which a porous microstructure, characterized by two geometrical parameters, was considered. Both geometrical parameters were varied throughout the macrostructure, and depending on their values, the effective Poisson's ratio changed from positive to negative. For this example, different PODECM models were obtained with good accuracies (of errors less than 1%), while achieving speed-ups up to 100.

The biggest drawback of PODECM is its intrusiveness, since it is necessary to change both the assembly procedure and the basis functions of an existing microscopic solver. If this is not a problem, PODECM offers several advantages in comparison to non-intrusive methods, such as the one presented in Chapter 3.

As the underlying microscopic PDE is still being solved, only a small amount of training data is required to construct a good approximation of the full model, and history-dependent material behavior does not need to be specially treated, making the framework very general.

# Chapter 5

## A reduced order model for second-order computational homogenization

The content of this chapter is based on the following publication (in preparation):

- Guo, T., Kouznetsova, V.G., Geers, M.G.D., Veroy, K., & Rokoš, O. (n.a.). Reduced order modeling for second-order computational homogenization with applications to geometrically parameterized elastomeric metamaterials. In preparation.

In previous chapters, the reduced order modelling of first-order computational homogenization (CH) has been treated. However, when employing first-order CH for mechanical metamaterials, the results may be inaccurate because the typical length scale of the microstructural features may become comparable with that of the macrostructural engineering application, or non-local effects such as buckling may emerge. To obtain better results, second-order CH formulations have been considered. Similar to the first-order formulation, the two-scale simulations are computationally expensive and must be accelerated. To the best of our knowledge, there are no reduced order models for second-order CH in the literature.

Here, we present a reduced order model for second-order CH, by following a similar approach as in Chapter 4: we first use a proper orthogonal decomposition (POD) to reduce the number of degrees of freedom, and propose a novel hyperreduction scheme specifically tailored for this problem that identifies a reduced number of integration points and weights.

After reviewing the theory on second-order CH and specifying the employed formulation in Section 5.1, the proposed ROM, including the novel hyperreduction algorithm, is presented in Section 5.2. To validate the ROM, two numerical examples are discussed in Section 5.3 and obtained results compared with reference solutions in terms of accuracy and efficiency. A summary on the findings with final remarks is provided in Section 5.4.

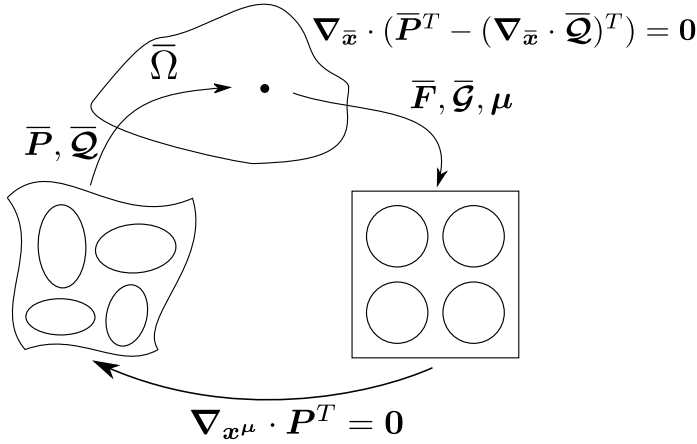
## 5.1 Second-order computational homogenization

The second-order CH formulation contains the second gradient of the displacement field, thus introducing a length-scale associated with the length-scale of the underlying unit cell, making it possible to capture size and non-local effects [90, 136]. The formulation of the micro- and macroscopic problem as well as their scale coupling employed in this work is discussed in the subsections below. A schematic sketch of the two-scale problem is depicted in Fig. 5.1.

### 5.1.1 Macroscopic problem

In second-order CH, the macroscopic problem is based on a strain gradient formulation [90, 136] to model non-local effects of the microstructure. Consider a body  $\bar{\Omega} \subset \mathbb{R}^d$  with outer boundaries  $\partial\bar{\Omega}$ ,  $d = 2, 3$  the space dimension, and a position vector  $\bar{\mathbf{x}} \in \bar{\Omega}$ . The governing partial differential equation (PDE) has the form (with body forces neglected for brevity) [69],

$$\nabla_{\bar{\mathbf{x}}} \cdot (\bar{\mathbf{P}}(\bar{\mathbf{F}}, \bar{\mathbf{G}}, \boldsymbol{\mu})^T - (\nabla_{\bar{\mathbf{x}}} \cdot \bar{\mathbf{Q}}(\bar{\mathbf{F}}, \bar{\mathbf{G}}, \boldsymbol{\mu}))^T) = \mathbf{0}, \quad (5.1)$$



**Figure 5.1:** Two-scale formulation in second-order computational homogenization. At every macroscopic point, the deformation gradient  $\bar{\mathbf{F}}$  and its gradient  $\bar{\mathcal{G}}$  are used to prescribe the loading for the microscopic problem, which after solving returns an effective stress  $\bar{\mathbf{P}}$  and couple stress  $\bar{\mathbf{Q}}$ . The parameter  $\mu$  describes the shape of the RVE. More information on  $\mu$  are provided in Section 5.1.2.

where  $\bar{\mathbf{F}} := \mathbf{I} + \nabla_{\bar{\mathbf{x}}}\bar{\mathbf{u}}$  is the macroscopic deformation gradient with  $\bar{\mathbf{u}}(\bar{\mathbf{x}})$  being the macroscopic displacement field and  $\mathbf{I}$  the identity tensor,  $\bar{\mathcal{G}} := \nabla_{\bar{\mathbf{x}}}(\bar{\mathbf{F}}^T)$  is the gradient of the (transposed) deformation gradient, i.e., a third-order tensor with symmetry  $\bar{\mathcal{G}}_{ijk} = \bar{\mathcal{G}}_{kji}$ , and  $\mu$  contains additional shape parameters;  $\bar{\mathbf{P}}$  denotes the second-order first Piola-Kirchhoff (1PK) stress tensor, and  $\bar{\mathbf{Q}}$  is a third-order tensor, often referred to as couple or double stress tensor [90, 136]. Unlike in the first-order microscopic problem presented in Section 2.2, here we assume that the RVE has fixed material parameters and does not depend on  $\lambda$ . By multiplying Eq. (5.1) with a test function  $\delta\bar{\mathbf{u}}$  and utilizing the divergence theorem, the following weak form can be derived [70],

$$\begin{aligned} & \int_{\bar{\Omega}} \left( \bar{\mathbf{P}}(\bar{\mathbf{F}}, \bar{\mathcal{G}}, \mu) : \delta\bar{\mathbf{F}} + \bar{\mathbf{Q}}(\bar{\mathbf{F}}, \bar{\mathcal{G}}, \mu) : \delta\bar{\mathcal{G}} \right) d\bar{\mathbf{x}} \\ & = \int_{\partial\bar{\Omega}_N} (\delta\bar{\mathbf{u}} \cdot (\bar{\mathbf{P}} - \nabla \cdot \bar{\mathbf{Q}}) \cdot \mathbf{n} + (\mathbf{n} \cdot \bar{\mathbf{Q}}) : \delta\bar{\mathbf{F}}) d\bar{\mathbf{x}} \end{aligned} \quad (5.2)$$

where  $\delta\bar{\mathbf{F}} := \nabla_{\bar{\mathbf{x}}}\delta\bar{\mathbf{u}}$  and  $\delta\bar{\mathcal{G}} := \nabla_{\bar{\mathbf{x}}}(\delta\bar{\mathbf{F}}^T)$  are introduced,  $\partial\bar{\Omega}_N$  denotes the boundaries with prescribed Neumann boundary conditions, and  $\mathbf{n}$  is the outward unit normal vector. In addition, there are Dirichlet boundaries  $\partial\bar{\Omega}_D$ , with  $\partial\bar{\Omega}_D \cap \partial\bar{\Omega}_N = \emptyset$  and  $\partial\bar{\Omega}_D \cup \partial\bar{\Omega}_N = \partial\bar{\Omega}$ , where values of the displacement  $\bar{\mathbf{u}}$  and its gradient  $\nabla_{\bar{\mathbf{x}}}\bar{\mathbf{u}}$  are prescribed. In this work, we consider only Dirichlet boundary conditions, and thus neglect the terms on the right hand side, i.e., Eq. (5.2)

becomes

$$\int_{\Omega} \left( \bar{\mathbf{P}}(\bar{\mathbf{F}}, \bar{\mathbf{G}}, \boldsymbol{\mu}) : \delta \bar{\mathbf{F}} + \bar{\mathbf{Q}}(\bar{\mathbf{F}}, \bar{\mathbf{G}}, \boldsymbol{\mu}) : \delta \bar{\mathbf{G}} \right) d\bar{\mathbf{x}} = 0. \quad (5.3)$$

The relation between  $(\bar{\mathbf{P}}, \bar{\mathbf{Q}})$  and  $(\bar{\mathbf{F}}, \bar{\mathbf{G}}, \boldsymbol{\mu})$  is established by solving the microscopic boundary value problem, which is defined on a representative volume element (RVE) and discussed in more detail in Section 5.1.2. In particular, we assume that  $\boldsymbol{\mu}$  contains parameters that describe the shape of the RVE.

In order to solve the problem in Eq. (5.3), the second gradient of  $\bar{\mathbf{u}}$  is required. To this end, Lesicar et al. [74] employed 36 degrees of freedom (DOF)  $C^1$ -triangular elements. Wu et al. [140] utilized an enriched discontinuous Galerkin method combined with a penalty method to enforce  $C^1$ -continuity weakly. Other works reformulate the problem in Eq. (5.3) with a mixed formulation instead [69, 79, 113], which is used also in this work. The idea of the mixed formulation is to introduce an independent deformation gradient field  $\widehat{\mathbf{F}}$  which is coupled with the deformation gradient computed from the displacement field through Lagrange multipliers  $\bar{\mathbf{L}}$ . With  $\widehat{\mathbf{F}}$ , its (transposed) gradient  $\widehat{\mathbf{G}} := \nabla_{\bar{\mathbf{x}}} \widehat{\mathbf{F}}^T$ , and  $\bar{\mathbf{L}}$ , the problem in Eq. (5.3) can be rewritten as

$$\bar{G}(\bar{\mathbf{u}}, \widehat{\mathbf{F}}, \bar{\mathbf{L}}) := \int_{\Omega} \left( \bar{\mathbf{P}} : \delta \bar{\mathbf{F}} + \bar{\mathbf{Q}} : \delta \widehat{\mathbf{G}} + \delta(\bar{\mathbf{L}} : (\widehat{\mathbf{F}} - \bar{\mathbf{F}})) \right) d\bar{\mathbf{x}} = 0, \quad (5.4)$$

where arguments of  $\bar{\mathbf{P}}$  and  $\bar{\mathbf{Q}}$  have been dropped for brevity. Inserting  $\delta \bar{\mathbf{F}} = \nabla_{\bar{\mathbf{x}}} \delta \bar{\mathbf{u}}$ ,  $\delta \widehat{\mathbf{G}} = \nabla_{\bar{\mathbf{x}}} \delta \widehat{\mathbf{F}}^T$  and

$$\delta(\bar{\mathbf{L}} : (\widehat{\mathbf{F}} - \bar{\mathbf{F}})) = (\widehat{\mathbf{F}} - \bar{\mathbf{F}}) : \delta \bar{\mathbf{L}} + \bar{\mathbf{L}} : \left( \delta \widehat{\mathbf{F}} - \nabla_{\bar{\mathbf{x}}} \delta \bar{\mathbf{u}} \right) \quad (5.5)$$

into Eq. (5.4) yields:

$$\begin{aligned} & \bar{G}(\bar{\mathbf{u}}, \widehat{\mathbf{F}}, \bar{\mathbf{L}}) \\ &= \int_{\Omega} \left( (\bar{\mathbf{P}} - \bar{\mathbf{L}}) : \nabla_{\bar{\mathbf{x}}} \delta \bar{\mathbf{u}} + \bar{\mathbf{L}} : \delta \widehat{\mathbf{F}} + \bar{\mathbf{Q}} : \nabla_{\bar{\mathbf{x}}} \delta \widehat{\mathbf{F}}^T + (\widehat{\mathbf{F}} - \bar{\mathbf{F}}) : \delta \bar{\mathbf{L}} \right) d\bar{\mathbf{x}}. \end{aligned} \quad (5.6)$$

Linearization of Eq. (5.6), required by the macroscopic iterative Newton solver, around a state  $(\bar{\mathbf{u}}, \widehat{\mathbf{F}}, \bar{\mathbf{L}})$  in directions  $(\Delta \bar{\mathbf{u}}, \mathbf{0}, \mathbf{0})$ ,  $(\mathbf{0}, \Delta \widehat{\mathbf{F}}, \mathbf{0})$  and  $(\mathbf{0}, \mathbf{0}, \Delta \bar{\mathbf{L}})$  gives:

$$\begin{aligned} & D\bar{G}|_{\bar{\mathbf{u}}, \widehat{\mathbf{F}}, \bar{\mathbf{L}}} \cdot (\Delta \bar{\mathbf{u}}, \mathbf{0}, \mathbf{0}) \\ &= \int_{\Omega} \left( \nabla_{\bar{\mathbf{x}}} \delta \bar{\mathbf{u}} : \nabla_{\bar{\mathbf{F}}} \bar{\mathbf{P}} + \nabla_{\bar{\mathbf{x}}} \delta \widehat{\mathbf{F}}^T : \nabla_{\bar{\mathbf{F}}} \bar{\mathbf{Q}} - \delta \bar{\mathbf{L}} \right) : \nabla_{\bar{\mathbf{x}}} \Delta \bar{\mathbf{u}} d\bar{\mathbf{x}}, \end{aligned} \quad (5.7)$$



$$\begin{aligned}
& D\bar{G}|_{\bar{\mathbf{u}}, \widehat{\mathbf{F}}, \bar{\mathbf{L}}} \cdot (\mathbf{0}, \Delta\widehat{\mathbf{F}}, \mathbf{0}) \\
&= \int_{\bar{\Omega}} \left( (\nabla_{\bar{\mathbf{x}}}\delta\bar{\mathbf{u}} : \nabla_{\widehat{\mathbf{G}}}\bar{\mathbf{P}} + \nabla_{\bar{\mathbf{x}}}\delta\widehat{\mathbf{F}}^T : \nabla_{\widehat{\mathbf{G}}}\bar{\mathbf{Q}}) : \nabla_{\bar{\mathbf{x}}}\Delta\widehat{\mathbf{F}}^T + \delta\bar{\mathbf{L}} : \Delta\widehat{\mathbf{F}} \right) d\bar{\mathbf{x}},
\end{aligned} \tag{5.8}$$

$$D\bar{G}|_{\bar{\mathbf{u}}, \widehat{\mathbf{F}}, \bar{\mathbf{L}}} \cdot (\mathbf{0}, \mathbf{0}, \Delta\bar{\mathbf{L}}) = \int_{\bar{\Omega}} \left( -\nabla_{\bar{\mathbf{x}}}\delta\bar{\mathbf{u}} + \delta\widehat{\mathbf{F}} \right) : \Delta\bar{\mathbf{L}} d\bar{\mathbf{x}}, \tag{5.9}$$

where  $\nabla_{\bar{\mathbf{F}}}\bar{\mathbf{P}}$ ,  $\nabla_{\bar{\mathbf{F}}}\bar{\mathbf{Q}}$ ,  $\nabla_{\widehat{\mathbf{G}}}\bar{\mathbf{P}}$  and  $\nabla_{\widehat{\mathbf{G}}}\bar{\mathbf{Q}}$  are the macroscopic tangents evaluated at  $(\bar{\mathbf{F}}(\bar{\mathbf{u}}), \widehat{\mathbf{G}}(\widehat{\mathbf{F}}), \boldsymbol{\mu})$ . Given a suitable discretization for  $\bar{\mathbf{u}}$ ,  $\widehat{\mathbf{F}}$  and  $\bar{\mathbf{L}}$ , the system of Eqs. (5.6)–(5.9) can be solved with the finite element method, when a constitutive relation between  $(\bar{\mathbf{P}}, \bar{\mathbf{Q}})$  and  $(\bar{\mathbf{F}}, \widehat{\mathbf{G}}, \boldsymbol{\mu})$  is established. Different combinations of displacement, deformation gradient and Lagrange multiplier shape functions were considered and tested in Kouznetsova et al. [69]. For the numerical examples in this work, quadrilateral elements with eight displacement nodes, four deformation gradient nodes and one Lagrange multiplier node per element are chosen.

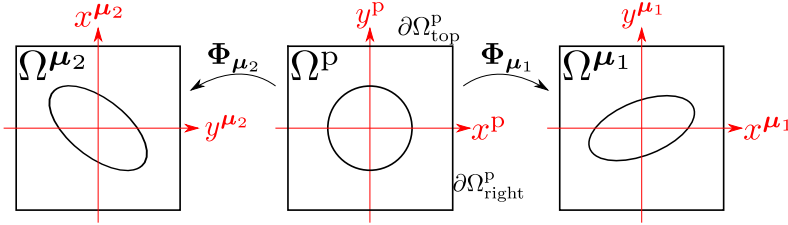
### 5.1.2 Parameterized microscopic problem

To evaluate  $\bar{\mathbf{P}}$  and  $\bar{\mathbf{Q}}$  and their derivatives in Eqs. (5.6)–(5.9), the microscopic problem needs to be solved at every macroscopic integration point. Here, we follow the formulation as presented in Kouznetsova et al. [69], where the microscopic problem is modelled as a standard Cauchy continuum. For brevity, a fixed macroscopic material point is assumed, and the dependence on the macroscopic coordinates is omitted in the definition of the microscopic problem provided below.

Consider a family of domains  $\Omega^\mu \subset \mathbb{R}^d$ , parameterized by parameters  $\boldsymbol{\mu} \in \mathbb{P}_\mu$  with parameter space  $\mathbb{P}_\mu$  and spanned by position vectors  $\mathbf{x}^\mu \in \Omega^\mu$ , see Fig. 5.2. For all  $\boldsymbol{\mu}$ , the outer boundaries and topology of  $\Omega^\mu$  are assumed to remain fixed. As a consequence, the volume  $|\Omega^\mu|$  remains constant for all  $\boldsymbol{\mu}$ . Additionally, it is assumed that there exists a parent domain  $\Omega^p := \Omega^{\mu_p}$  with  $\boldsymbol{\mu}_p \in \mathbb{P}_\mu$ , which can be transformed into any  $\Omega^\mu$  with a transformation map  $\Phi_\mu : \Omega^p \rightarrow \Omega^\mu$ ,  $\mathbf{x}^p \mapsto \mathbf{x}^\mu$ , transformation gradient  $\mathbf{F}_\mu := \nabla_{\mathbf{x}^p}\Phi_\mu$  and  $d\mathbf{x}^\mu = |\det \mathbf{F}_\mu| d\mathbf{x}^p$ . For a fixed domain, i.e.,  $\Omega^p = \Omega^\mu$ , the transformations  $\Phi_\mu$  are identity maps, with  $\mathbf{F}_\mu = \mathbf{I}$  and  $|\det \mathbf{F}_\mu| = 1$ .

To obtain such transformation maps  $\Phi_\mu$ , we solve the auxiliary problem as proposed in Section 3.1.1. The key idea of the method is to define an auxiliary linear elasticity problem on the parent domain that can be solved for the transformation displacement  $\mathbf{d}$ , which can then be utilized to compute the

transformation  $\Phi_\mu$  with  $\Phi_\mu(\mathbf{x}^P) = \mathbf{I} + \mathbf{d}(\mathbf{x}^P)$ . The displacement  $\mathbf{d}$  is fixed on the outer boundaries with zero Dirichlet boundary conditions and prescribed on parts of the domain that are known from the parameterization (as an example, the circular interface of  $\Omega^P$  in Fig. 5.2 is deformed into the elliptical interfaces in  $\Omega^{\mu_1}$  and  $\Omega^{\mu_2}$ ). Subsequently, the auxiliary problem can be solved to find the entire field  $\mathbf{d}(\mathbf{x}^P)$ . For all RVEs considered in this chapter, the auxiliary problem is solved with  $E^{\text{aux}} = 1$  MPa and  $\xi^{\text{aux}} = 0.25$  (defining the elasticity tensor of the auxiliary problem).



**Figure 5.2:** Family of RVE domains  $\Omega^\mu$  parameterized through parameters  $\mu$ . A parent domain  $\Omega^P$  can be defined which can be transformed through transformations  $\Phi_\mu$  into any of the RVE domains  $\Omega^\mu$ .

The microscopic displacement field  $\mathbf{u}(\mathbf{x}^\mu)$  is assumed to consist of a mean field  $\bar{\mathbf{u}}(\mathbf{x}^\mu)$  and a fluctuation field  $\mathbf{w}(\mathbf{x}^\mu)$ , i.e.,  $\mathbf{u}(\mathbf{x}^\mu) = \bar{\mathbf{u}}(\mathbf{x}^\mu) + \mathbf{w}(\mathbf{x}^\mu)$ . The mean field  $\bar{\mathbf{u}}$  is fully prescribed through the macroscopic quantities  $(\bar{\mathbf{F}}, \widehat{\mathbf{G}})$  with

$$\bar{\mathbf{u}}(\mathbf{x}^\mu) := (\bar{\mathbf{F}} - \mathbf{I})\mathbf{x}^\mu + \frac{1}{2}(\mathbf{x}^\mu \cdot \widehat{\mathbf{G}}) \cdot \mathbf{x}^\mu. \quad (5.10)$$

Subsequently, the microscopic deformation gradient can be defined as

$$\mathbf{F} := \mathbf{I} + \nabla_{\mathbf{x}^\mu} \mathbf{u} = \bar{\mathbf{F}} + \mathbf{x}^\mu \cdot \widehat{\mathbf{G}} + \nabla_{\mathbf{x}^\mu} \mathbf{w}. \quad (5.11)$$

Compared to the deformation gradient in first-order formulation, cf. Eq. (2.7), there is an additional term due to the strain gradient  $\widehat{\mathbf{G}}$ , making it possible to communicate non-local effects such as bending of the RVE. The governing microscopic PDE is given by,

$$\nabla_{\mathbf{x}^\mu} \cdot \mathbf{P}^T(\mathbf{F}) = \mathbf{0}, \quad (5.12)$$

which can be written into the weak form by multiplying with a test function  $\delta \mathbf{w}$  and applying the divergence theorem (cf. Eq. (2.9)),

$$G(\mathbf{w}) = \int_{\Omega^\mu} \nabla_{\mathbf{x}^\mu} \delta \mathbf{w} : \mathbf{P} \left( \bar{\mathbf{F}} + \mathbf{x}^\mu \cdot \widehat{\mathbf{G}} + \nabla_{\mathbf{x}^\mu} \mathbf{w} \right) d\mathbf{x}^\mu \stackrel{!}{=} 0, \quad (5.13)$$

where  $\mathbf{P}$  is the microscopic second-order 1PK stress tensor, and the macroscopic quantities  $(\bar{\mathbf{F}}, \widehat{\mathbf{G}})$  act as external forcing terms. For now, no constitutive model at the microscale level is specified, but it is assumed that  $\mathbf{P}$  is a non-linear function of the deformation gradient  $\mathbf{F}$ . The microscopic problem can thus be stated as follows: given  $(\bar{\mathbf{F}}, \widehat{\mathbf{G}}, \boldsymbol{\mu})$ , find  $\mathbf{w}$  that fulfills Eq. (5.13) for all  $\delta\mathbf{w}$ . To remove the dependence of the integral on parameters  $\boldsymbol{\mu}$ , Eq. (5.13) can be transformed to the parent domain with the transformation map  $\Phi_{\boldsymbol{\mu}}$ , i.e.,

$$G^{\mathbf{P}}(\mathbf{w}^{\mathbf{P}}) = \int_{\Omega^{\mathbf{P}}} ((\nabla_{\mathbf{x}^{\mathbf{P}}} \delta\mathbf{w}^{\mathbf{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \mathbf{P}^{\mathbf{P}}(\mathbf{F}^{\mathbf{P}}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\mathbf{P}} \stackrel{!}{=} 0, \quad (5.14)$$

where  $\mathbf{w}^{\mathbf{P}}(\mathbf{x}^{\mathbf{P}}) := (\mathbf{w} \circ \Phi_{\boldsymbol{\mu}})(\mathbf{x}^{\mathbf{P}}) = \mathbf{w}(\mathbf{x}^{\boldsymbol{\mu}})$ ,  $\delta\mathbf{w}^{\mathbf{P}}(\mathbf{x}^{\mathbf{P}}) := \delta\mathbf{w}(\mathbf{x}^{\boldsymbol{\mu}})$ ,  $\mathbf{P}^{\mathbf{P}}(\mathbf{x}^{\mathbf{P}}) := \mathbf{P}(\mathbf{x}^{\boldsymbol{\mu}})$ , and

$$\mathbf{F}^{\mathbf{P}} = \bar{\mathbf{F}} + \Phi_{\boldsymbol{\mu}}(\mathbf{x}^{\mathbf{P}}) \cdot \widehat{\mathbf{G}} + (\nabla_{\mathbf{x}^{\mathbf{P}}} \mathbf{w}^{\mathbf{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1}. \quad (5.15)$$

Hereafter, we write  $\mathbf{x}^{\boldsymbol{\mu}}(\mathbf{x}^{\mathbf{P}})$  instead of  $\Phi_{\boldsymbol{\mu}}(\mathbf{x}^{\mathbf{P}})$  for brevity. To find the  $\mathbf{w}^{\mathbf{P}}$  that fulfills Eq. (5.14) for all  $\delta\mathbf{w}^{\mathbf{P}}$ , the linearization of Eq. (5.14) around a state  $\mathbf{w}^{\mathbf{P}}$  in direction  $\Delta\mathbf{w}^{\mathbf{P}}$  is required,

$$\begin{aligned} & DG^{\mathbf{P}}|_{\mathbf{w}^{\mathbf{P}}} \cdot (\Delta\mathbf{w}^{\mathbf{P}}) \\ &= \int_{\Omega^{\mathbf{P}}} ((\nabla_{\mathbf{x}^{\mathbf{P}}} \delta\mathbf{w}^{\mathbf{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) : \mathfrak{A}^{\mathbf{P}}(\mathbf{F}^{\mathbf{P}}) : ((\nabla_{\mathbf{x}^{\mathbf{P}}} \Delta\mathbf{w}^{\mathbf{P}}) \mathbf{F}_{\boldsymbol{\mu}}^{-1}) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\mathbf{P}}, \end{aligned} \quad (5.16)$$

where  $\mathfrak{A}^{\mathbf{P}} := \nabla_{\mathbf{F}} \mathbf{P}^{\mathbf{P}}$  is the fourth-order stiffness tensor on the parent domain.

To ensure a proper scale transition of the kinematic quantities, different authors derived and proposed additional constraints on the fluctuation field  $\mathbf{w}^{\mathbf{P}}$ . In Kouznetsova et al. [69], periodic boundary conditions (PBC) for  $\mathbf{w}^{\mathbf{P}}$  are assumed and the following constraints are derived for a rectangular RVE:

$$\int_{\partial\Omega_{\text{top}}^{\mathbf{P}}} \mathbf{w}^{\mathbf{P}} d\mathbf{x}^{\mathbf{P}} = \mathbf{0}, \quad (5.17)$$

$$\int_{\partial\Omega_{\text{right}}^{\mathbf{P}}} \mathbf{w}^{\mathbf{P}} d\mathbf{x}^{\mathbf{P}} = \mathbf{0}, \quad (5.18)$$

where  $\partial\Omega_{\text{top}}^{\mathbf{P}}$  and  $\partial\Omega_{\text{right}}^{\mathbf{P}}$  denote the top and right edge of the RVE  $\Omega^{\mathbf{P}}$ , see Fig. 5.2. In addition, all four corners of the RVE are fixed, i.e.,  $\mathbf{w}^{\mathbf{P}} = \mathbf{0}$ . Since it is assumed that  $\Omega^{\boldsymbol{\mu}}$  has fixed outer boundaries for all  $\boldsymbol{\mu}$ , constraints in Eqs. (5.17) and (5.18) are independent of  $\boldsymbol{\mu}$ . In follow-up works, other authors developed slightly different formulations (see, e.g., [15, 79, 113, 140, 148]). In [113], the authors compared different formulations and pointed out that fixing the corners in the formulation in Kouznetsova et al. [69] leads to stress

concentrations and artificial effects at the corners. Instead of fixing the corners, other formulations introduce an additional equation that constrains the rigid body motion with,

$$\int_{\Omega^\mu} \mathbf{w} d\mathbf{x}^\mu = \int_{\Omega^p} \mathbf{w}^p |\det \mathbf{F}_\mu| d\mathbf{x}^p = \mathbf{0}. \quad (5.19)$$

The complete microscopic model employed in this work consists of Eqs. (5.14)–(5.19) together with PBC for  $\mathbf{w}^p$ . Lagrange multipliers are used to enforce the constraints in Eqs. (5.17)–(5.19) and PBC, resulting in a saddle point problem.

To solve the microscopic problem, the fluctuation displacement is typically discretized with finite elements (FE), i.e.,

$$\mathbf{w}^p(\mathbf{x}^p) \approx \mathbf{N}(\mathbf{x}^p)\mathbf{w}, \quad (5.20)$$

where  $\mathbf{N}(\mathbf{x}^p) \in \mathbb{R}^{d \times \mathcal{N}}$  denotes the FE shape functions,  $\mathbf{w} \in \mathbb{R}^{\mathcal{N}}$  the coefficients of the discretized displacement field, and  $\mathcal{N}$  the total number of DOFs. Subsequently, the weak form in Eq. (5.14), together with the constraints, can be written as

$$\begin{aligned} \mathbf{f}(\mathbf{w}) + \mathbf{C}^T \mathbf{m} &= \mathbf{0}, \\ \mathbf{C} \mathbf{w} &= \mathbf{0}, \end{aligned} \quad (5.21)$$

where  $\mathbf{f} \in \mathbb{R}^{\mathcal{N}}$  is the global internal force vector, the constraint matrix  $\mathbf{C} \in \mathbb{R}^{N_c \times \mathcal{N}}$  is derived from the constraints in Eqs. (5.17)–(5.19) and PBC, with  $N_c$  the number of constraint equations, and  $\mathbf{m} \in \mathbb{R}^{N_c}$  are the corresponding Lagrange multipliers. Using the Newton method, the non-linear system of equations in Eq. (5.21) can be solved for  $\mathbf{w}$  and  $\mathbf{m}$ ,

$$\begin{aligned} \begin{bmatrix} \mathbf{K}(\mathbf{w}^k) & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{w} \\ \mathbf{m} \end{bmatrix} &= \begin{bmatrix} -\mathbf{f}(\mathbf{w}^k) \\ \mathbf{0} \end{bmatrix}, \\ \mathbf{w}^{k+1} &= \mathbf{w}^k + \Delta \mathbf{w}, \end{aligned} \quad (5.22)$$

where  $\mathbf{K} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$  is the global stiffness matrix computed from Eq. (5.16), and  $k$  is the Newton iteration number. Eq. (5.22) is iterated with increasing  $k$  until  $\|\mathbf{f}(\mathbf{w}^k) + \mathbf{C}^T \mathbf{m}\|_2 \leq \varepsilon_{\text{newton}}$  with  $\varepsilon_{\text{newton}}$  a user-defined tolerance. For more information on the FE method and discretization of weak forms, we refer to [6].

### 5.1.3 Effective quantities

After the microscopic problem has been solved and a solution  $\mathbf{w}^{*p}$  obtained, the effective stress  $\bar{\mathbf{P}}$ , couple stress  $\bar{\mathbf{Q}}$  and their corresponding derivatives

with respect to  $\bar{\mathbf{F}}$  and  $\widehat{\mathcal{G}}$  must be computed. For conciseness of notation, the following microscopic quantities are introduced:

$$\mathbf{F}^{*P} := \bar{\mathbf{F}} + \mathbf{x}^\mu \cdot \widehat{\mathcal{G}} + (\nabla_{\mathbf{x}^P} w^{*P}) \mathbf{F}_\mu^{-1}, \quad (5.23)$$

$$\mathbf{P}^{*P} := \mathbf{P}^P(\mathbf{F}^{*P}), \quad (5.24)$$

$$\mathfrak{A}^{*P} := \mathfrak{A}^P(\mathbf{F}^{*P}), \quad (5.25)$$

which correspond to the microstructural deformation gradient, 1PK stress and related stiffness tensors, evaluated at the solution on the parent domain. Expressions for the effective stress  $\bar{\mathbf{P}}$  and couple stress  $\bar{\mathcal{Q}}$  were derived in [70, Section 3.3], which after transformation to the parent domain read,

$$\bar{\mathbf{P}} := \frac{1}{|\Omega^P|} \int_{\Omega^P} \mathbf{P}^{*P} |\det \mathbf{F}_\mu| d\mathbf{x}^P, \quad (5.26)$$

$$\bar{\mathcal{Q}} := \frac{1}{|\Omega^P|} \int_{\Omega^P} \frac{1}{2} \left( \mathbf{P}^{*PT} \otimes \mathbf{x}^\mu + \mathbf{x}^\mu \otimes \mathbf{P}^{*P} \right) |\det \mathbf{F}_\mu| d\mathbf{x}^P. \quad (5.27)$$

The effective stiffness components, derived by differentiating the above stress and couple stress quantities then read,

$$\frac{\partial \bar{P}_{ij}}{\partial \bar{F}_{kl}} = \frac{1}{|\Omega^P|} \int_{\Omega^P} \frac{\partial P_{ij}^{*P}}{\partial \bar{F}_{kl}} |\det \mathbf{F}_\mu| d\mathbf{x}^P, \quad (5.28)$$

$$\frac{\partial \bar{P}_{ij}}{\partial \widehat{\mathcal{G}}_{mno}} = \frac{1}{|\Omega^P|} \int_{\Omega^P} \frac{\partial P_{ij}^{*P}}{\partial \widehat{\mathcal{G}}_{mno}} |\det \mathbf{F}_\mu| d\mathbf{x}^P, \quad (5.29)$$

$$\frac{\partial \bar{\mathcal{Q}}_{ijk}}{\partial \bar{F}_{mn}} = \frac{1}{|\Omega^P|} \int_{\Omega^P} \frac{1}{2} \left( \frac{\partial P_{ji}^{*P}}{\partial \bar{F}_{mn}} x_k^\mu + x_i^\mu \frac{\partial P_{jk}^{*P}}{\partial \bar{F}_{mn}} \right) |\det \mathbf{F}_\mu| d\mathbf{x}^P, \quad (5.30)$$

$$\frac{\partial \bar{\mathcal{Q}}_{ijk}}{\partial \widehat{\mathcal{G}}_{mno}} = \frac{1}{|\Omega^P|} \int_{\Omega^P} \frac{1}{2} \left( \frac{\partial P_{ji}^{*P}}{\partial \widehat{\mathcal{G}}_{mno}} x_k^\mu + x_i^\mu \frac{\partial P_{jk}^{*P}}{\partial \widehat{\mathcal{G}}_{mno}} \right) |\det \mathbf{F}_\mu| d\mathbf{x}^P. \quad (5.31)$$

In the above,

$$\frac{\partial P_{ij}^{*P}}{\partial \bar{F}_{kl}} = \mathfrak{A}_{ijmn}^{*P} \left( \delta_{mk} \delta_{nl} + \frac{\partial}{\partial \bar{F}_{kl}} \left( \frac{\partial w_m^{*P}}{\partial x_r^P} \right) F_{\mu, rn}^{-1} \right), \quad (5.32)$$

and

$$\begin{aligned}
\frac{\partial P_{ij}^{*p}}{\partial \widehat{\mathcal{G}}_{mno}} &= \mathfrak{A}_{ijkl}^{*p} \left( \frac{\partial \left( x_r^\mu \widehat{\mathcal{G}}_{rkl} \right)}{\partial \widehat{\mathcal{G}}_{mno}} + \frac{\partial}{\partial \widehat{\mathcal{G}}_{mno}} \left( \frac{\partial w_k^{*p}}{\partial x_s^p} \right) F_{\mu,sl}^{-1} \right) \\
&= \mathfrak{A}_{ijkl}^{*p} \left( x_r^\mu \delta_{rm} \delta_{kn} \delta_{lo} + \frac{\partial}{\partial \widehat{\mathcal{G}}_{mno}} \left( \frac{\partial w_k^{*p}}{\partial x_s^p} \right) F_{\mu,sl}^{-1} \right) \\
&= \mathfrak{A}_{ijkl}^{*p} \left( x_m^\mu \delta_{kn} \delta_{lo} + \frac{\partial}{\partial \widehat{\mathcal{G}}_{mno}} \left( \frac{\partial w_k^{*p}}{\partial x_s^p} \right) F_{\mu,sl}^{-1} \right).
\end{aligned} \tag{5.33}$$

To determine  $\frac{\partial}{\partial \bar{F}_{kl}} \left( \frac{\partial w_m^{*p}}{\partial x_r^p} \right)$  and  $\frac{\partial}{\partial \widehat{\mathcal{G}}_{mno}} \left( \frac{\partial w_k^{*p}}{\partial x_s^p} \right)$ , Eq. (5.14) is differentiated with respect to  $\bar{\mathbf{F}}$  and  $\widehat{\mathcal{G}}$  to derive linear tangent problems that can be solved to find the corresponding sensitivities. As an example, for one particular component  $\bar{F}_{kl}$  (where the indices  $k$  and  $l$  are assumed to be temporarily fixed), the differentiation yields

$$\frac{\partial G^p(\mathbf{w}^{*p})}{\partial \bar{F}_{kl}} = \int_{\Omega^p} \left( (\nabla_{\mathbf{x}^p} \delta \mathbf{w}^p) \mathbf{F}_\mu^{-1} \right) : \frac{\partial \mathbf{P}^{*p}}{\partial \bar{F}_{kl}} |\det \mathbf{F}_\mu| d\mathbf{x}^p = 0, \tag{5.34}$$

which can be rearranged with Eq. (5.32) as,

$$\begin{aligned}
&\int_{\Omega^p} \left( (\nabla_{\mathbf{x}^p} \delta \mathbf{w}^p) \mathbf{F}_\mu^{-1} \right) : \mathfrak{A}^{*p} : \left( (\nabla_{\mathbf{x}^p} \mathbf{q}_{kl}) \mathbf{F}_\mu^{-1} \right) |\det \mathbf{F}_\mu| d\mathbf{x}^p \\
&= - \left( \int_{\Omega^p} \left( (\nabla_{\mathbf{x}^p} \delta \mathbf{w}^p) \mathbf{F}_\mu^{-1} \right) : \mathfrak{A}^{*p} |\det \mathbf{F}_\mu| d\mathbf{x}^p \right) : \mathbf{E}_{kl}
\end{aligned} \tag{5.35}$$

where a new auxiliary vector field  $\mathbf{q}_{kl} := \frac{\partial \mathbf{w}^{*p}}{\partial \bar{F}_{kl}}$  is defined, reflecting the sensitivity of the microfluctuation field with respect to the change of the macroscopic deformation gradient, and  $\mathbf{E}_{kl} \in \mathbb{R}^{d \times d}$  is a second-order tensor with all entries zero except for the  $kl$ -th entry which is 1. The linear problem of Eq. (5.35) is solved for all combinations  $k, l = 1, \dots, d$  to obtain  $\mathbf{q}_{kl}$  for each component of  $\bar{\mathbf{F}}$ . The same procedure is followed for  $\widehat{\mathcal{G}}$ . With an auxiliary vector field  $\mathbf{q}_{mno} := \frac{\partial \mathbf{w}^{*p}}{\partial \widehat{\mathcal{G}}_{mno}}$ , the differentiation of Eq. (5.14) for one particular component  $\widehat{\mathcal{G}}_{mno}$  (where the indices  $m, n$  and  $o$  are assumed to be temporarily fixed) yields, together with Eq. (5.33),

$$\begin{aligned}
&\int_{\Omega^p} \left( (\nabla_{\mathbf{x}^p} \delta \mathbf{w}^p) \mathbf{F}_\mu^{-1} \right) : \mathfrak{A}^{*p} : \left( (\nabla_{\mathbf{x}^p} \mathbf{q}_{mno}) \mathbf{F}_\mu^{-1} \right) |\det \mathbf{F}_\mu| d\mathbf{x}^p \\
&= - \left( \int_{\Omega^p} \mathbf{x}^\mu \otimes \left( (\nabla_{\mathbf{x}^p} \delta \mathbf{w}^p) \mathbf{F}_\mu^{-1} \right) : \mathfrak{A}^{*p} |\det \mathbf{F}_\mu| d\mathbf{x}^p \right) : \mathcal{E}_{mno},
\end{aligned} \tag{5.36}$$

where  $\mathcal{E}_{mno} \in \mathbb{R}^{d \times d \times d}$  is a third-order tensor with all entries zero except for the  $mno$ -th entry which is 1. The linear problem of Eq. (5.36) is then solved for all combinations  $m, n, o = 1, \dots, d$  to obtain  $\mathbf{q}_{mno}$  for each component of  $\widehat{\mathcal{G}}$ .

## 5.2 Reduced order modeling

Since the microscopic problem is solved at every macroscopic integration point, its solution must be efficient. Due to the often complicated RVE geometry, a fine discretization (using, e.g., finite elements) is required, resulting in a large number of DOFs and integration points, which entail a costly solution. Furthermore, computing the effective quantities for a fine RVE mesh presents another computationally expensive operation. To construct a reduced order model (ROM) for a more efficient solution, we employ two reduction techniques: (1) we utilize proper orthogonal decomposition (POD) to reduce the number of DOFs in Section 5.2.1; (2) to find a more efficient integration scheme, we propose in Section 5.2.2 a novel hyperreduction algorithm that uses ideas of the empirical cubature method [48], which is specifically suited for the second-order CH formulation.

### 5.2.1 Proper orthogonal decomposition

To reduce the number of DOFs, the fluctuation displacement field  $\mathbf{w}^P$  is approximated with a reduced basis [52, 103], i.e.,

$$\mathbf{w}^P \approx \sum_{n=1}^N a_n \mathbf{v}_n, \quad (5.37)$$

where  $N$  is typically much smaller than  $\mathcal{N}$ , i.e.,  $N \ll \mathcal{N}$ . The global basis functions,  $\{\mathbf{v}_n\}_{n=1}^N$ , are obtained by applying POD to a set of pre-computed snapshots of  $\mathbf{w}^P$  for different parameter values  $(\bar{\mathbf{F}}, \widehat{\mathcal{G}}, \boldsymbol{\mu})$ . Since each of the basis functions is computed from a linear combination of pre-computed periodic solutions that fulfill the constraints in Eqs. (5.17)–(5.19), every basis function is periodic and also fulfills Eqs. (5.17) and (5.18). This implies that any solution  $\mathbf{w}^P$  that is represented by Eq. (5.37) always fulfills these conditions. However, the basis functions will only fulfill the constraint in Eq. (5.19) if a fixed geometry is assumed for the RVE, i.e.,  $\boldsymbol{\mu}$  is constant. For varying geometries, the constraint is violated due to the influence of  $|\det \mathbf{F}_\mu|$ . Nevertheless, in our numerical examples, tests with and without enforcing Eq. (5.19) through Lagrange multipliers were run and only insignificant differences of the solutions were observed. For that reason, we do not enforce Eq. (5.19) for the ROM.

This has the added advantage that no constraints have to be considered for the ROM and, thus, more efficient solvers for the resulting system of linear equations can be utilized.

By inserting Eq. (5.37) into Eqs. (5.14) and (5.16) and assuming a Galerkin projection, the components of the internal force  $\mathbf{f} \in \mathbb{R}^N$  and global stiffness matrix  $\mathbf{K} \in \mathbb{R}^{N \times N}$  can be computed:

$$f_i(\mathbf{a}) := \int_{\Omega^P} ((\nabla_{\mathbf{x}^P} \mathbf{v}_i) \mathbf{F}_\mu^{-1}) : \mathbf{P}^P(\mathbf{F}^P) |\det \mathbf{F}_\mu| d\mathbf{x}^P, \quad (5.38)$$

$$K_{ij}(\mathbf{a}) := \int_{\Omega^P} ((\nabla_{\mathbf{x}^P} \mathbf{v}_i) \mathbf{F}_\mu^{-1}) : \mathfrak{A}^P(\mathbf{F}^P) : ((\nabla_{\mathbf{x}^P} \mathbf{v}_j) \mathbf{F}_\mu^{-1}) |\det \mathbf{F}_\mu| d\mathbf{x}^P, \quad (5.39)$$

$$\mathbf{F}^P(\mathbf{a}) = \bar{\mathbf{F}} + \mathbf{x}^\mu \cdot \widehat{\mathbf{G}} + \left( \sum_{n=1}^N a_n \nabla_{\mathbf{x}^P} \mathbf{v}_n \right) \mathbf{F}_\mu^{-1}, \quad (5.40)$$

for all  $i, j = 1, \dots, N$ . The column matrix  $\mathbf{a} = [a_1, \dots, a_N]^T$  contains the unknown coefficients to be solved for.

## 5.2.2 Hyperreduction

While the reduced system of Eqs. (5.38) and (5.39) only has  $N$  DOFs, computing the integrals in Eqs. (5.38) and (5.39) (as well as Eqs. (5.26)–(5.33), (5.35) and (5.36) for the effective quantities) requires integration over the whole full finite element mesh (typically with many Gauss quadrature points). To accelerate this computation, a more efficient integration scheme (i.e., fewer integration points and corresponding weights) is sought that closely approximates the numerical integration with Gaussian quadrature of the following four quantities:

- Internal force  $\mathbf{f} = [f_1, \dots, f_N]^T$  in Eq. (5.38):

$$\begin{aligned} f_i &= \int_{\Omega^P} ((\nabla_{\mathbf{x}^P} \mathbf{v}_i) \mathbf{F}_\mu^{-1}) : \mathbf{P}^P |\det \mathbf{F}_\mu| d\mathbf{x}^P \\ &= \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_i : \mathbf{W}^P d\mathbf{x}^P, \end{aligned} \quad (5.41)$$

for all  $i = 1, \dots, N$  and where the weighted stress  $\mathbf{W}^P := \mathbf{P}^P \mathbf{F}_\mu^{-T} |\det \mathbf{F}_\mu|$  is defined (in analogy with Eq. (3.6)).



- Effective stress  $\bar{\mathbf{P}}$  in Eq. (5.26):

$$\begin{aligned}
 \bar{\mathbf{P}} &= \frac{1}{|\Omega^{\mathbf{P}}|} \int_{\Omega^{\mathbf{P}}} \mathbf{P}^{\mathbf{P}} |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\mathbf{P}} \\
 &= \frac{1}{|\Omega^{\mathbf{P}}|} \int_{\Omega^{\mathbf{P}}} \mathbf{W}^{\mathbf{P}} \mathbf{F}_{\boldsymbol{\mu}}^T d\mathbf{x}^{\mathbf{P}} \\
 &= \frac{1}{|\Omega^{\mathbf{P}}|} \int_{\Omega^{\mathbf{P}}} \mathbf{W}^{\mathbf{P}} d\mathbf{x}^{\mathbf{P}},
 \end{aligned} \tag{5.42}$$

where in the last line the invariance of the integral with respect to  $\mathbf{F}_{\boldsymbol{\mu}}^T$  was used, which was proven in Appendix C. This implies that the accurate integration of the effective stress is equivalent to the accurate integration of the weighted stress.

- Effective couple stress  $\bar{\mathbf{Q}}$  in Eq. (5.27):

$$\begin{aligned}
 \bar{\mathbf{Q}} &= \frac{1}{|\Omega^{\mathbf{P}}|} \int_{\Omega^{\mathbf{P}}} \frac{1}{2} \left( \mathbf{P}^{\mathbf{P}T} \otimes \mathbf{x}^{\boldsymbol{\mu}} + \mathbf{x}^{\boldsymbol{\mu}} \otimes \mathbf{P}^{\mathbf{P}} \right) |\det \mathbf{F}_{\boldsymbol{\mu}}| d\mathbf{x}^{\mathbf{P}} \\
 &= \frac{1}{|\Omega^{\mathbf{P}}|} \int_{\Omega^{\mathbf{P}}} \boldsymbol{\mathcal{Y}}^{\mathbf{P}} d\mathbf{x}^{\mathbf{P}},
 \end{aligned} \tag{5.43}$$

where the weighted couple stress  $\boldsymbol{\mathcal{Y}}^{\mathbf{P}} := \frac{1}{2} \left( \mathbf{P}^{\mathbf{P}T} \otimes \mathbf{x}^{\boldsymbol{\mu}} + \mathbf{x}^{\boldsymbol{\mu}} \otimes \mathbf{P}^{\mathbf{P}} \right) |\det \mathbf{F}_{\boldsymbol{\mu}}|$  is defined.

- Volume:

$$V := |\Omega^{\mathbf{P}}| = \int_{\Omega^{\mathbf{P}}} d\mathbf{x}^{\mathbf{P}}. \tag{5.44}$$

Even though the integration of the volume does not necessarily have to be accurate, it helps to stabilize the non-negative least squares problem used to find the new integration points, as seen below. In particular, it leads to fewer weights that are equal to 0.

**Algorithm** To find an efficient integration scheme, we use ideas from the empirical cubature method (ECM), which was previously applied to find integration points and weights for the efficient integration of the internal force, as proposed by Hernández et al. [48] and extended to varying geometries in Chapter 4. In the first step, similarly to the fluctuation field  $\mathbf{w}^{\mathbf{P}}$ , snapshots of the weighted stress  $\mathbf{W}^{\mathbf{P}}$  and weighted couple stress  $\boldsymbol{\mathcal{Y}}^{\mathbf{P}}$  are collected for different parameter values  $(\bar{\mathbf{F}}, \widehat{\boldsymbol{\mathcal{G}}}, \boldsymbol{\mu})$ . Utilizing POD, two sets of basis functions

for  $\mathbf{W}^{\text{P}}$  and  $\mathbf{Y}^{\text{P}}$ ,  $\{\mathbf{B}_m\}_{m=1}^M$  and  $\{\mathcal{H}_l\}_{l=1}^L$ , are found, with which  $\mathbf{W}^{\text{P}}$  and  $\mathbf{Y}^{\text{P}}$  can then be approximated, i.e.,

$$\mathbf{W}^{\text{P}} \approx \sum_{m=1}^M \alpha_m \mathbf{B}_m, \quad (5.45)$$

$$\mathbf{Y}^{\text{P}} \approx \sum_{l=1}^L \beta_l \mathcal{H}_l. \quad (5.46)$$

Inserting Eqs. (5.45) and (5.46) into Eqs. (5.41)–(5.43) yields

$$f_i \approx \sum_{m=1}^M \alpha_m \int_{\Omega^{\text{P}}} \nabla_{\mathbf{x}^{\text{P}}} \mathbf{v}_i : \mathbf{B}_m d\mathbf{x}^{\text{P}}, \quad \forall i = 1, \dots, N, \quad (5.47)$$

$$\bar{\mathbf{P}} \approx \sum_{m=1}^M \alpha_m \frac{1}{|\Omega^{\text{P}}|} \int_{\Omega^{\text{P}}} \mathbf{B}_m d\mathbf{x}^{\text{P}}, \quad (5.48)$$

$$\bar{\mathcal{Q}} \approx \sum_{l=1}^L \beta_l \frac{1}{|\Omega^{\text{P}}|} \int_{\Omega^{\text{P}}} \mathcal{H}_l d\mathbf{x}^{\text{P}}. \quad (5.49)$$

Since Eqs. (5.47)–(5.49) should be accurately integrated for any choice of coefficients  $\alpha_m$  and  $\beta_l$ , all the occurring integrals in the sums should be approximated accurately. Together with the volume equation in Eq. (5.44), integration points and weights are sought that approximate these  $NM + d^2M + d^3L + 1$  integrals accurately. The factors  $d^2$  and  $d^3$  arise due to the number of components of  $\bar{\mathbf{P}}$  and  $\bar{\mathcal{Q}}$ .

Assume for now the known set of Gaussian integration points  $\{\hat{\mathbf{x}}_q, \hat{w}_q\}_{q=1}^{\hat{Q}}$  corresponding to the fully resolved discretization, where  $\hat{Q}$  is the total number of Gauss integration points,  $\hat{\mathbf{x}}_q$  their positions and  $\hat{w}_q$  their weights. By defining

$$\mathbf{1} := [1, \dots, 1]^T \in \mathbb{R}^{\hat{Q}}, \quad (5.50)$$

and the flattened basis functions  $\underline{\mathbf{B}}_m \in \mathbb{R}^{d^2}$  and  $\underline{\mathcal{H}}_l \in \mathbb{R}^{d^3}$  (in 2D, i.e.,  $d = 2$ ),

$$\underline{\mathbf{B}}_m := [B_{m,11}, B_{m,12}, B_{m,21}, B_{m,22}]^T, \quad (5.51)$$

$$\underline{\mathcal{H}}_l := [\mathcal{H}_{l,111}, \mathcal{H}_{l,112}, \mathcal{H}_{l,121}, \mathcal{H}_{l,122}, \mathcal{H}_{l,211}, \mathcal{H}_{l,212}, \mathcal{H}_{l,221}, \mathcal{H}_{l,222}]^T, \quad (5.52)$$

for all  $m = 1, \dots, M$  and  $l = 1, \dots, L$ , the numerical approximation of the  $NM + d^2M + d^3L + 1$  integrals with the full Gauss quadrature can be written

in algebraic form

$$\underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ \mathbf{1}^T \end{bmatrix}}_{=: \mathbf{A}} \underbrace{\begin{bmatrix} \hat{w}_1 \\ \vdots \\ \hat{w}_{\hat{Q}} \end{bmatrix}}_{=: \hat{\mathbf{w}}} = \underbrace{\begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ V \end{bmatrix}}_{=: \mathbf{b}}, \quad (5.53)$$

where

$$\mathbf{b}_1 = \begin{bmatrix} \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_1 : \mathbf{B}_1 d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_1 : \mathbf{B}_M d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_N : \mathbf{B}_1 d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_N : \mathbf{B}_M d\mathbf{x}^P \end{bmatrix} \in \mathbb{R}^{NM}, \quad (5.54)$$

$$\mathbf{b}_2 = \begin{bmatrix} \int_{\Omega^P} \underline{\mathbf{B}}_1 d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \underline{\mathbf{B}}_M d\mathbf{x}^P \end{bmatrix} \in \mathbb{R}^{d^2 M}, \quad (5.55)$$

$$\mathbf{b}_3 = \begin{bmatrix} \int_{\Omega^P} \underline{\mathcal{H}}_1 d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \underline{\mathcal{H}}_L d\mathbf{x}^P \end{bmatrix} \in \mathbb{R}^{d^3 L}. \quad (5.56)$$

and

$$\mathbf{A}_1 = \begin{bmatrix} (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_1)|_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_1)|_{\hat{\mathbf{x}}_{\hat{Q}}} \\ \vdots & & \vdots \\ (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_M)|_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_M)|_{\hat{\mathbf{x}}_{\hat{Q}}} \\ \vdots & & \vdots \\ (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_1)|_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_1)|_{\hat{\mathbf{x}}_{\hat{Q}}} \\ \vdots & & \vdots \\ (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_M)|_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_M)|_{\hat{\mathbf{x}}_{\hat{Q}}} \end{bmatrix} \in \mathbb{R}^{NM \times \hat{Q}}, \quad (5.57)$$

$$\mathbf{A}_2 = \begin{bmatrix} \underline{\mathbf{B}}_1(\hat{\mathbf{x}}_1) & \dots & \underline{\mathbf{B}}_1(\hat{\mathbf{x}}_{\hat{Q}}) \\ \vdots & & \vdots \\ \underline{\mathbf{B}}_M(\hat{\mathbf{x}}_1) & \dots & \underline{\mathbf{B}}_M(\hat{\mathbf{x}}_{\hat{Q}}) \end{bmatrix} \in \mathbb{R}^{d^2 M \times \hat{Q}}, \quad (5.58)$$

$$\mathbf{A}_3 = \begin{bmatrix} \underline{\mathcal{H}}_1(\hat{\mathbf{x}}_1) & \dots & \underline{\mathcal{H}}_1(\hat{\mathbf{x}}_{\hat{Q}}) \\ \vdots & & \vdots \\ \underline{\mathcal{H}}_L(\hat{\mathbf{x}}_1) & \dots & \underline{\mathcal{H}}_L(\hat{\mathbf{x}}_{\hat{Q}}) \end{bmatrix} \in \mathbb{R}^{d^3 L \times \hat{Q}}. \quad (5.59)$$

The system in Eq. (5.53) can be equivalently rewritten into

$$\underbrace{\begin{bmatrix} \hat{\mathbf{A}}_1 \\ \hat{\mathbf{A}}_2 \\ \hat{\mathbf{A}}_3 \\ \mathbf{1}^T \end{bmatrix}}_{=:\hat{\mathbf{A}}} \hat{\mathbf{w}} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ V \end{bmatrix}}_{=:\hat{\mathbf{b}}}, \quad (5.60)$$

with

$$\hat{\mathbf{A}}_1 = \mathbf{A}_1 - \frac{1}{V} \mathbf{b}_1 \otimes \mathbf{1}, \quad (5.61)$$

$$\hat{\mathbf{A}}_2 = \mathbf{A}_2 - \frac{1}{V} \mathbf{b}_2 \otimes \mathbf{1}, \quad (5.62)$$

$$\hat{\mathbf{A}}_3 = \mathbf{A}_3 - \frac{1}{V} \mathbf{b}_3 \otimes \mathbf{1}, \quad (5.63)$$

which is convenient for the definition of residuals of the algorithm discussed below.

The goal now is to select a subset of integration points  $\{\mathbf{x}_q\}_{q=1}^Q$  from the set of all integration points, i.e.,  $\{\mathbf{x}_q\}_{q=1}^Q \subset \{\hat{\mathbf{x}}_q\}_{q=1}^{\hat{Q}}$ , such that  $Q \ll \hat{Q}$ , with corresponding weights  $\{w_q\}_{q=1}^Q$  obtained by minimizing the following weighted

non-negative least squares residual,

$$\begin{aligned}
\mathbf{w}^{\text{LS}} &= \arg \min_{\mathbf{w} \geq \mathbf{0}} \left\| \hat{\mathbf{b}} - \hat{\mathbf{A}}_{\bullet \mathcal{I}} \mathbf{w} \right\|_{\Sigma} \\
&= \arg \min_{\mathbf{w} \geq \mathbf{0}} \left\| \hat{\mathbf{A}} \hat{\mathbf{w}} - \hat{\mathbf{A}}_{\bullet \mathcal{I}} \mathbf{w} \right\|_{\Sigma} \\
&= \arg \min_{\mathbf{w} \geq \mathbf{0}} \left\| \hat{\mathbf{r}}(\mathbf{w}) \right\|_{\Sigma},
\end{aligned} \tag{5.64}$$

where  $\|\mathbf{a}\|_{\Sigma} := \mathbf{a}^T \Sigma \mathbf{a}$  and the residual

$$\hat{\mathbf{r}}(\mathbf{w}) := \hat{\mathbf{A}} \hat{\mathbf{w}} - \hat{\mathbf{A}}_{\bullet \mathcal{I}} \mathbf{w} \tag{5.65}$$

are defined,  $\mathcal{I}$  denotes a set of non-repeating indices with  $|\mathcal{I}| = Q$  and  $\hat{\mathbf{A}}_{\bullet \mathcal{I}}$  is the submatrix of  $\hat{\mathbf{A}}$  with  $Q$  selected columns according to the entries of  $\mathcal{I}$ . The matrix  $\Sigma$  is a weighting matrix with a block diagonal structure

$$\Sigma = \begin{bmatrix} c_1 \Sigma_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & c_2 \Sigma_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & c_3 \Sigma_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Sigma_4 \end{bmatrix}, \tag{5.66}$$

where each block corresponds to one of the approximated quantities and is defined as,

$$\Sigma_1 = \text{diag}(\sigma_1^w \sigma_1^W, \dots, \sigma_1^w \sigma_M^W, \dots, \sigma_N^w \sigma_1^W, \dots, \sigma_N^w \sigma_M^W) \in \mathbb{R}^{NM \times NM}, \tag{5.67}$$

$$\Sigma_2 = \text{diag}(\underbrace{\sigma_1^W, \dots, \sigma_1^W}_{d^2 \text{ times}}, \dots, \underbrace{\sigma_M^W, \dots, \sigma_M^W}_{d^2 \text{ times}}) \in \mathbb{R}^{d^2 M \times d^2 M}, \tag{5.68}$$

$$\Sigma_3 = \text{diag}(\underbrace{\sigma_1^{\mathcal{Y}}, \dots, \sigma_1^{\mathcal{Y}}}_{d^3 \text{ times}}, \dots, \underbrace{\sigma_L^{\mathcal{Y}}, \dots, \sigma_L^{\mathcal{Y}}}_{d^3 \text{ times}}) \in \mathbb{R}^{d^3 L \times d^3 L}, \tag{5.69}$$

$$\Sigma_4 = \text{diag}(1) \in \mathbb{R}^{1 \times 1}. \tag{5.70}$$

The entries  $\sigma_i^w, \sigma_m^W, \sigma_l^{\mathcal{Y}}$  for all  $i = 1, \dots, N$ ,  $m = 1, \dots, M$ , and  $l = 1, \dots, L$  correspond to the ordered normalized singular values of the POD of the fluctuation field  $\mathbf{w}^p$ , weighted stress  $\mathbf{W}^p$  and weighted couple stress  $\mathcal{Y}^p$ , with  $\sigma_1^w = \sigma_1^W = \sigma_1^{\mathcal{Y}} = 1$ . The parameters  $c_1, c_2, c_3$  enable control over the importance of each of the approximated quantities. Their influence is illustrated in Section 5.3. The blocks of the weighting matrix  $\Sigma$  are chosen in this fashion to promote the integration scheme to approximate the basis functions corresponding to the larger singular values more accurately than the ones corresponding to smaller singular values. The indices in  $\mathcal{I}$  are selected one by one, similarly

to the greedy algorithm presented in [48]. The exact algorithm on the selection is provided in Algorithm 3. For the algorithm, the residual  $\hat{\mathbf{r}}$  is split into four parts,

$$\hat{\mathbf{r}} = [\hat{\mathbf{r}}_1^T, \hat{\mathbf{r}}_2^T, \hat{\mathbf{r}}_3^T, \hat{\mathbf{r}}_4^T]^T \quad (5.71)$$

where  $\hat{\mathbf{r}}_i$  for  $i = 1, \dots, 4$  are the residuals for each quantity. Independent residuals are introduced to check that each quantity is approximated accurately up to a precision depending on the choice of tolerances  $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ , with,

$$\begin{aligned} r_1 &:= \frac{\|\hat{\mathbf{r}}_1\|_{\Sigma_1}}{\text{tr } \Sigma_1} < \varepsilon_1, & r_2 &:= \frac{\|\hat{\mathbf{r}}_2\|_{\Sigma_2}}{\text{tr } \Sigma_2} < \varepsilon_2, \\ r_3 &:= \frac{\|\hat{\mathbf{r}}_3\|_{\Sigma_3}}{\text{tr } \Sigma_3} < \varepsilon_3, & r_4 &:= \frac{\|\hat{\mathbf{r}}_4\|_{\Sigma_4}}{V} < \varepsilon_4, \end{aligned} \quad (5.72)$$

where  $r_i$  for  $i = 1, \dots, 4$  are the standardized norms of the residuals. As will be shown in Section 5.3, all  $r_i$  generally decay with different rates, which can, however, be tuned with the parameters  $c_1, c_2, c_3$ . The lowest number of quadrature points can be achieved when all residuals reach the desired tolerances at the same time. This will be demonstrated in Section 5.3.

---

**Algorithm 3** Integration point selection algorithm
 

---

**Offline Stage:**  $\hat{\mathbf{A}}, \hat{\mathbf{b}}, \Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4, c_1, c_2, c_3, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, k_{\max}$ 
**Online Stage:**  $\mathbf{w}^{\text{LS}}, \mathcal{I}$ 

 Initialize empty list of selected columns  $\mathcal{I} \leftarrow \emptyset$ 

 Initialize list of candidate indices  $\mathcal{C} \leftarrow \{1, \dots, \hat{Q}\}$ 

 Set iteration number  $k \leftarrow 0$ 

 Set initial residual  $\hat{\mathbf{r}} \leftarrow \hat{\mathbf{b}}$ 
**while**  $k < k_{\max}$  **do**
 $k \leftarrow k + 1$ 

 Find the column  $i$  of  $\hat{\mathbf{A}}$  with

$$i = \arg \max_{j \in \mathcal{C}} \frac{\hat{\mathbf{A}}_{\bullet j}^T \Sigma \hat{\mathbf{r}}}{\sqrt{\hat{\mathbf{A}}_{\bullet j}^T \Sigma \hat{\mathbf{A}}_{\bullet j}}}$$

 Add selected index  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$ 

 Remove selected index from candidates  $\mathcal{C} \leftarrow \mathcal{C} \setminus \{i\}$ 

 Solve Eq. (5.64) for  $\mathbf{w}^{\text{LS}}$ 

 Compute residuals  $\hat{\mathbf{r}}_i$  according to Eq. (5.65)

**if** all conditions in Eq. (5.72) are fulfilled **then**
 $\mathbf{return} \mathbf{w}^{\text{LS}}, \mathcal{I}$ 
 $\triangleright$  Algorithm is converged

**end if**
**end while**


---

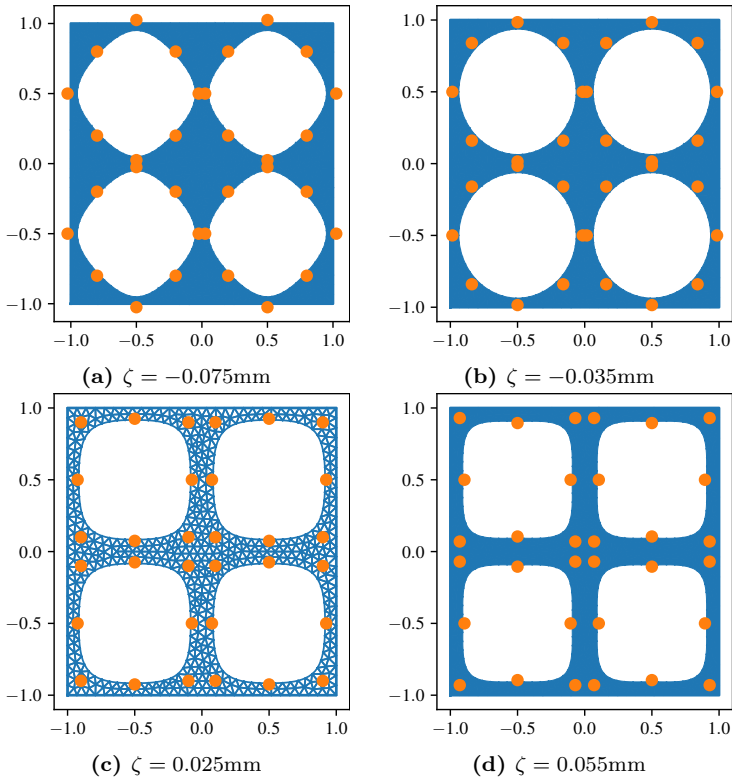
### 5.3 Numerical examples

To examine and illustrate different features of the proposed ROM, two macroscopic example problems with a parameterized microstructure are studied in two dimensions and under plane strain conditions. The results are compared against the full two-scale second-order CH solution (later referred to as CH2) as well as the direct numerical simulation (DNS), where the microstructure is fully resolved at the macroscale. The ROM is discussed in detail in the first example, whereas the second example shows a possible application, in which a full DNS might not be feasible anymore (especially in three dimensions), but with the ROM an excellent approximation can be computed in a reasonable amount of time.

For both examples, a metamaterial-based RVE with four identical holes is selected, motivated by *Specimen 1* in Bertoldi et al. [9]. The size of the RVE is  $2\text{ mm} \times 2\text{ mm}$  and the local coordinate system is chosen in the center of the domain, i.e., the domain of the RVE is given by  $[-1\text{ mm}, 1\text{ mm}] \times [-1\text{ mm}, 1\text{ mm}]$ . Each hole is described by a cubic B-spline with eight control points, of which the coordinates are parameterized by one geometrical parameter  $\boldsymbol{\mu} = \{\zeta\}$ . For the top right hole, the coordinates (in mm) of the control points are  $(0.05 + \zeta, 0.5)$ ,  $(0.125 - \zeta, 0.125 - \zeta)$ ,  $(0.5, 0.05 + \zeta)$ ,  $(0.875 + \zeta, 0.125 - \zeta)$ ,  $(0.95 - \zeta, 0.5)$ ,  $(0.875 + \zeta, 0.875 + \zeta)$ ,  $(0.5, 0.95 - \zeta)$ ,  $(0.125 - \zeta, 0.875 + \zeta)$ . The coordinates of the control points for the other holes are obtained by shifting the coordinates of the top right hole by 1mm in the  $x$ - and/or  $y$ -direction, and the same  $\zeta$  is assumed for each hole. The geometry of the RVE is shown for different values of  $\zeta = \{-0.075\text{mm}, -0.035\text{mm}, 0.025\text{mm}, 0.055\text{mm}\}$  in Fig. 5.3. The parent domain  $\Omega^p$ , chosen with  $\zeta = 0.025\text{mm}$ , and its simulation mesh, consisting of 4882 DOFs and 1066 six-noded triangular elements, are shown in Fig. 5.3c. A mesh convergence study was conducted to ensure that the effective quantities obtained with this mesh are converged with respect to the element size. Note that the control points (in orange color) are allowed to lie outside the RVE domain, as long as the resulting B-spline curves do not intersect with the outer boundary of the RVE.

Depending on  $\zeta$ , the shape of the holes changes from a circular shape to a square-like one. For circular holes, it is known that, due to the symmetry, the RVE buckles locally under compression and exhibits auxetic behavior [9], i.e., under uniaxial compression in one direction the RVE shortens in the perpendicular direction. On the other hand, square-like holes promote global buckling on the macroscale instead of local buckling. This is illustrated in Section 5.3.1, where significantly different behaviors of the macrostructure are observed when varying  $\zeta$ .





**Figure 5.3:** Example geometries for (a)  $\zeta = -0.075\text{mm}$ , (b)  $\zeta = -0.035\text{mm}$ , (c)  $\zeta = 0.025\text{mm}$ , and (d)  $\zeta = 0.055\text{mm}$ . The control points defining the hole shapes are shown in orange and the matrix material in blue. Depending on  $\zeta$ , the shape of the holes is more circular or square-like, and the RVE more prone to local or global buckling. The parent domain  $\Omega^P$  is chosen for  $\zeta = 0.025\text{mm}$  with a simulation mesh, consisting of 1066 six-noded triangular elements with 4882 DOFs.

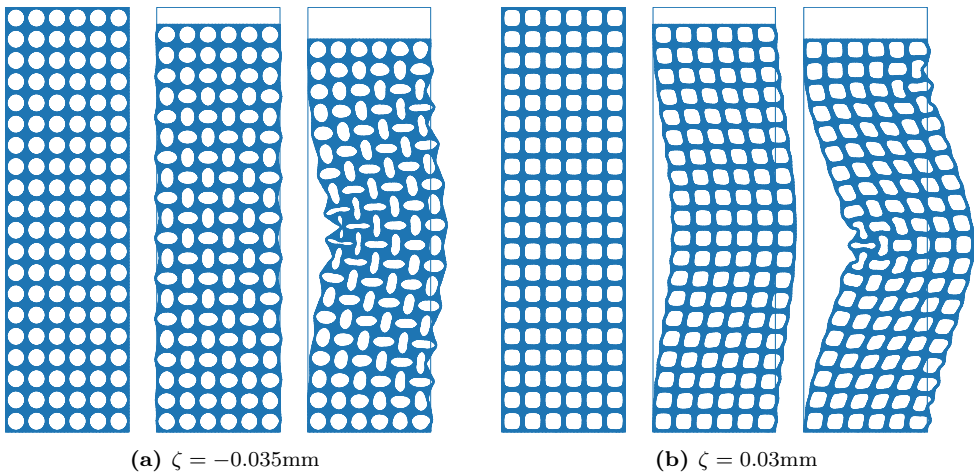
The RVE material is modelled as a hyperelastic Mooney–Rivlin material with strain energy density function

$$W(\mathbf{F}) = C_1(I_1 - 3) + C_2(I_1 - 3)^2 - 2C_1 \log J + \frac{K}{2}(J - 1)^2, \quad (5.73)$$

where  $I_1 := \text{tr } \mathbf{C}$  is the first invariant of the right Cauchy–Green tensor  $\mathbf{C} := \mathbf{F}^T \mathbf{F}$  and  $J = \det \mathbf{F}$  characterizes the volume change. The constants  $C_1$ ,  $C_2$  and  $K$  are material parameters, which are set to  $C_1 = 0.55 \text{ MPa}$ ,  $C_2 = 0.3 \text{ MPa}$ , and  $K = 55 \text{ MPa}$ , according to the experimental data in Bertoldi et al. [8].

### 5.3.1 Uniaxial compression of perforated plate

In the first example, uniaxial compression of a rectangular perforated plate of size  $W \times H$  (width  $W = 6$  mm, height  $H = 20$  mm) in the longitudinal direction is considered. The top edge is compressed up to 7.5% strain, while the bottom edge is fixed, and the geometrical parameter  $\zeta$  is assumed to be constant throughout the macrostructure. As the reference solution, full DNS solutions, where the microstructure is fully resolved and for which triangular six-noded elements are used, are computed for various values of  $\zeta$ . For each DNS model, the number of elements is approximately 32,000 and number of DOFs 140,000. For  $\zeta = \{-0.035\text{mm}, 0.03\text{mm}\}$ , the undeformed state together with the deformed solutions at 4% and 7.5% strain are shown in Fig. 5.4. While for  $\zeta = -0.035\text{mm}$  the macrostructure first buckles locally (patterning of holes) and then globally (whole macrostructure buckles), for  $\zeta = 0.03\text{mm}$  the structure first buckles globally and then locally, implying that the overall behavior of the macrostructure changes drastically for different  $\zeta$ .



**Figure 5.4:** DNS solutions for  $\zeta = -0.035\text{mm}$  in (a),  $\zeta = 0.03\text{mm}$  in (b). In each panel, the undeformed (left) and deformed states at 4% (middle) and 7.5% (right) compression are shown. For  $\zeta = -0.035\text{mm}$  the structure first buckles locally and then globally, while for  $\zeta = 0.03\text{mm}$  the structure first buckles globally with subsequent local patterning.

For the homogenized plate, a uniform mesh, consisting of two elements in the horizontal and four elements in the vertical direction with a total of 32 quadrature points, is chosen, amounting to 32 microscopic problems that must be solved for each macroscopic Newton iteration. Regarding the boundary conditions, the displacement is fixed at the bottom edge and prescribed at the top edge with value  $\tilde{u}$ , while the  $xx$ - and  $yx$ -components of the deformation

gradient are fixed to  $\widehat{F}_{xx} = 1$  and  $\widehat{F}_{yx} = 0$  at the top and bottom edge. To prevent zero energy modes corresponding to the components  $\widehat{F}_{xy}$  and  $\widehat{F}_{yy}$ , the deformation gradient at the bottom left point is fully fixed with  $\widehat{\mathbf{F}} = \mathbf{I}$ .

To construct the ROM, training data must be generated by solving the microscopic problem for different input parameters  $(\overline{\mathbf{F}}, \widehat{\mathcal{G}}, \boldsymbol{\mu})$ , which span an 11-dimensional parameter space in 2D. In total, 100 samples are generated for the loading parameters  $(\overline{\mathbf{F}}, \widehat{\mathcal{G}})$  via a Sobol sequence sampling with parameter bounds provided in Table 5.1. Since the macrostructure is compressed up to 7.5% in  $y$ -direction and locally higher deformations might occur, the lower bound for  $\overline{F}_{yy} - 1$  is chosen as  $-0.1$ . Since the RVE behaves auxetically, the lower bound for  $\overline{F}_{xx} - 1$  is also assumed to be  $-0.1$ . The upper bound for both  $\overline{F}_{xx} - 1$  and  $\overline{F}_{yy} - 1$  is chosen as  $0.02$  to capture some tensile behavior of the RVE. Due to the global buckling, large shear strains might occur and bounds of  $[-0.1, 0.1]$  are chosen for  $\overline{F}_{xy}$  and  $\overline{F}_{yx}$ . Bounds for  $\widehat{\mathcal{G}}$  are difficult to estimate without prior knowledge. Here, every component is assumed to range from  $-0.05 \text{ mm}^{-1}$  to  $0.05 \text{ mm}^{-1}$ , which for the RVE size of  $2 \text{ mm} \times 2 \text{ mm}$  can result in maximal deformations in the range of  $[-0.1, 0.1]$  with  $\mathbf{F} - \mathbf{I} = \mathbf{x}^p \cdot \widehat{\mathcal{G}}$  and  $\mathbf{x}^p \in \Omega^p = [-1 \text{ mm}, 1 \text{ mm}]^2$ .

Subsequently, all the samples are divided into five groups, each with 20 samples and assigned one value of  $\zeta = \{-0.05 \text{ mm}, -0.025 \text{ mm}, 0.0 \text{ mm}, 0.025 \text{ mm}, 0.05 \text{ mm}\}$ . For each sample, the macroscopic loads are applied to the RVE with  $(t\overline{\mathbf{F}}, t\widehat{\mathcal{G}})$ , where  $t \in [0, 1]$  is increased linearly from 0 to 1 in 20 equidistant load steps, resulting in 20 snapshots per sample. In total, 2000 snapshots are obtained which are all used for the construction of the ROM.

**Table 5.1:** Parameter bounds used for sampling training data for the ROM. The bounds for  $\overline{\mathbf{F}}$  are motivated by the applied macroscopic compression loads and the auxeticity of the RVE. The bounds for  $\widehat{\mathcal{G}}$  are chosen randomly.

$\overline{F}_{xx} - 1$	$\overline{F}_{xy}$	$\overline{F}_{yx}$	$\overline{F}_{yy} - 1$
$[-0.1, 0.02]$	$[-0.1, 0.1]$	$[-0.1, 0.1]$	$[-0.1, 0.02]$
$\widehat{\mathcal{G}}_{xxx} [\text{mm}^{-1}]$	$\widehat{\mathcal{G}}_{xxy} [\text{mm}^{-1}]$	$\widehat{\mathcal{G}}_{xyx} [\text{mm}^{-1}]$	$\widehat{\mathcal{G}}_{xyy} [\text{mm}^{-1}]$
$[-0.05, 0.05]$	$[-0.05, 0.05]$	$[-0.05, 0.05]$	$[-0.05, 0.05]$

**Results** The accuracy and efficiency of the ROM depends on several factors:

- the number of basis functions for the fluctuation displacement  $N$ , the weighted stress  $M$  and the weighted couple stress  $L$ ,

- the error tolerances  $\varepsilon_1$ ,  $\varepsilon_2$ ,  $\varepsilon_3$  and  $\varepsilon_4$ , and
- the hyperparameters  $c_1$ ,  $c_2$  and  $c_3$  that control the weighting matrix  $\Sigma$  of the weighted least squares problem in Eqs. (5.64) and (5.66).

To choose a reasonable amount of basis functions, the singular values of POD are often utilized, as they give an indication on the information loss due to the reduction. Given the ordered singular values  $\{\sigma_i\}_{i=1}^{N_{\text{POD}}}$  of POD, a criterion can be defined with:

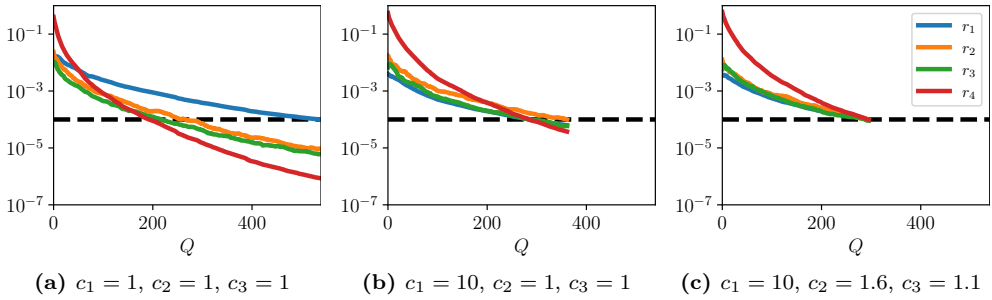
$$1 - \frac{\sum_{i=1}^{N_{\text{POD}}} \sigma_i^2}{\sum_{i=1}^{N_S} \sigma_i^2} < \mathcal{E}_{\text{POD}}, \quad (5.74)$$

where  $N_S$  denotes the total number of snapshots and  $\mathcal{E}_{\text{POD}}$  is a user-specified tolerance. The number of basis functions is then selected to be equal to the smallest  $N_{\text{POD}}$ , for which Eq. (5.74) is fulfilled. For the weighted stress and couple stress, good results were obtained with  $\mathcal{E}_{\text{POD}} = 5 \times 10^{-3}$ , for which  $M = 28$  and  $L = 28$  were found. For the fluctuation field, three values of  $\mathcal{E}_{\text{POD}} = 1 \times 10^{-4}$ ,  $1 \times 10^{-5}$  and  $1 \times 10^{-6}$  were considered, which resulted in  $N = 48$ , 78 and 112 basis functions.

Regarding the error tolerances  $\varepsilon_1$ ,  $\varepsilon_2$ ,  $\varepsilon_3$  and  $\varepsilon_4$ , numerical tests for different values were carried out and a good balance in terms of accuracy and efficiency was found for  $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 1 \times 10^{-4}$ .

The choice of the hyperparameters  $c_1$ ,  $c_2$  and  $c_3$  affects the rates with which each of the standardized norm of residuals  $r_1$ ,  $r_2$ ,  $r_3$  and  $r_4$  (see Eq. (5.72) for the definition) decreases over the number of selected quadrature points  $Q$ . In general, the lowest number of quadrature points can be found when  $c_1$ ,  $c_2$  and  $c_3$  are tuned such that all  $r_i$  fall below their corresponding tolerances at roughly the same time. In Fig. 5.5, the decay of each  $r_i$  over the selected number of quadrature points with  $N = 48$ ,  $M = 28$ ,  $L = 28$  and  $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 1 \times 10^{-4}$  is shown for different choices of  $c_1$ ,  $c_2$  and  $c_3$ . For  $c_1 = c_2 = c_3 = 1$  (see Fig. 5.5a), it can be clearly seen that  $r_1$  drops much more slowly than  $r_2$ ,  $r_3$  and  $r_4$ , resulting in a total of  $Q = 539$  quadrature points. When increasing  $c_1$  to 10 (see Fig. 5.5b),  $r_1$  drops more quickly, ending up in a total number of  $Q = 362$  quadrature points. Finally, for values of  $c_1 = 10$ ,  $c_2 = 1.6$  and  $c_3 = 1.1$  (see Fig. 5.5c) all tolerances are achieved at roughly the same time with only  $Q = 296$  quadrature points. For  $N = 78$  and 112,  $Q = 318$  and 337 quadrature points are found, when all other hyperparameters are kept constant.

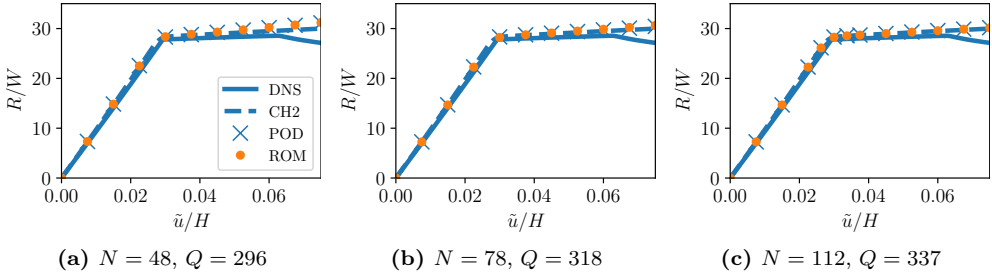
To evaluate the accuracy of the ROM for  $N = 48$ , 78 and 112 basis functions for the displacement, the two-scale compression of the perforated plate is solved for  $\zeta = -0.035\text{mm}$  (recall that the training data was sampled for



**Figure 5.5:** Decay of standardized norm of residuals  $r_1, r_2, r_3$  and  $r_4$  over the number of selected integration points with  $N = 48, M = 28, L = 28$  and  $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 1 \times 10^{-4}$ . The hyperparameters  $c_1, c_2$  and  $c_3$  control the rate of decay for each  $r_i$ , which can result in completely different numbers of quadrature points: (a)  $Q = 539$ , (b)  $Q = 362$ , whereas (c) only  $Q = 296$  quadrature points are selected.

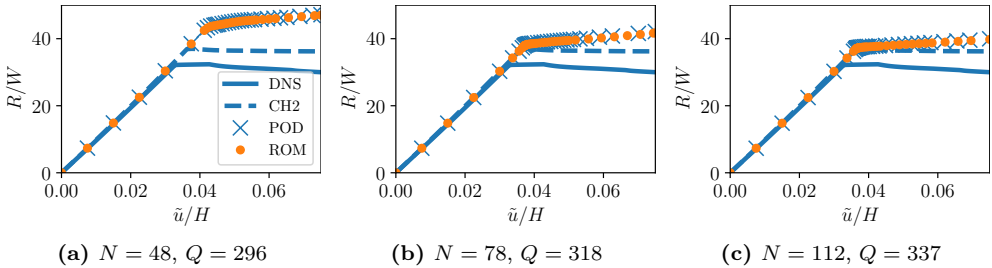
$\zeta = \{-0.05\text{mm}, -0.025\text{mm}, 0.0\text{mm}, 0.025\text{mm}, 0.05\text{mm}\}$ ), and the total resulting reaction force  $R$  acting on the top edge is plotted over the prescribed displacement  $\tilde{u}$  and compared to the DNS, CH2 and POD solutions in Fig. 5.6 ( $R$  and  $\tilde{u}$  are normalized with the width  $W$  and height  $H$  of the plate to yield nominal quantities). Here, POD denotes the solution obtained with the POD basis in Eq. (5.37), but with full integration of the reduced system, i.e., computing the integrals in Eqs. (5.38) and (5.39) (as well as Eqs. (5.26)–(5.33), (5.35) and (5.36)) with Gauss quadrature. It can clearly be seen that the ROM closely follows the POD solution, showing that the reduced integration is very accurate. It is also clear that the prebuckling stage and buckling point predicted by CH2 are sufficiently accurate already for  $N = 48$ , but small deviations from the CH2 solution can be observed for increasing  $\tilde{u}$  in the postbuckling stage. This error is decreased by increasing  $N$  to 78 or 112. Moreover, it can be observed that the CH2 solution predicts a slightly higher prebuckling stiffness than the DNS, which was also observed in [125], and is unable to predict the second (global) buckling point (at around 6% strain), illustrating some modeling limitations of the CH2 scheme. A full two-scale simulation based on first-order CH (FE2) was also run, however, the solver was unable to converge when reaching buckling, demonstrating issues of FE2 for buckling problems and advantages of the CH2 scheme.

In Fig. 5.7, a similar comparison is performed for  $\zeta = 0.03\text{mm}$  with  $N = 48, 78$  and 112. It can be clearly seen that the POD and ROM solutions match well for all  $N$ , but both overestimate the (post-)buckling behavior of the CH2 solution. Even though with increasing  $N$  the buckling point is predicted more accurately, the postbuckling stiffness is captured poorly. This suggests that the randomly generated training data does not properly cover the kinematics



**Figure 5.6:** Force-displacement curves for  $\zeta = -0.035\text{mm}$  (i.e., circular holes buckling first locally) for different numbers of basis functions  $N$  for the fluctuation field. The ROM solution closely follows the POD solution, implying that the proposed hyperreduction algorithm yields accurate results. For increasing number of fluctuation displacement basis functions  $N$ , the POD and ROM solution both approach that of CH2.

during the postbuckling stage and more representative training data is required for good approximations. Compared to the DNS solution, CH2 overpredicts the buckling point and the prebuckling stiffness. Nevertheless, the postbuckling stiffness is captured quite accurately. Moreover, CH2 is again unable to detect the second buckling which occurs at around  $\tilde{u}/H = 0.042$ . For this example, the FE2 solver was also run, which again was unable to converge at the buckling point.



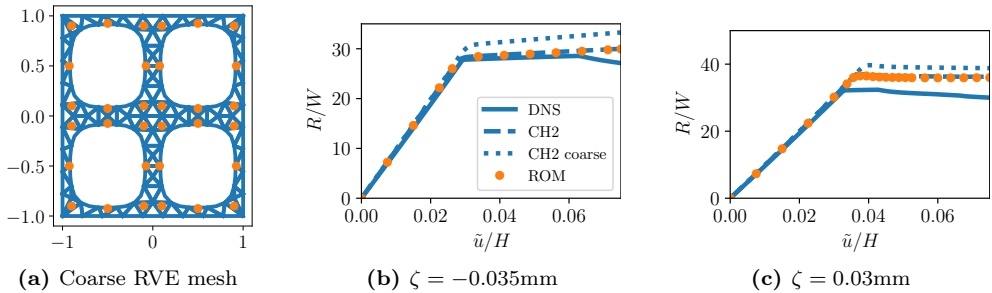
**Figure 5.7:** Force-displacement curves for  $\zeta = 0.03\text{mm}$  (i.e., square-like holes buckling first globally) for different numbers of basis functions  $N$  for the fluctuation field. Similarly to  $\zeta = -0.035\text{mm}$ , the ROM solution closely follows the POD solution. Both POD and ROM solutions approximate the CH2 solution poorly, implying that the randomly sampled training data is not representative for the global buckling of the macrostructure.

To demonstrate that the results for  $\zeta = 0.03\text{mm}$  are improved by employing a more representative training dataset, we generated another training dataset by employing the following procedure:

1. First, we solved the full CH2 problem with a coarsened RVE mesh (142 six-noded elements with 746 DOFs, see Fig. 5.8a) for  $\zeta = \{-0.05\text{mm}, -0.025\text{mm}, 0.0\text{mm}, 0.025\text{mm}, 0.05\text{mm}\}$ .

2. This way, for each value of  $\zeta$  a loading trajectory of values of  $\{(\bar{\mathbf{F}}, \widehat{\mathbf{G}})\}$  for each of the 32 macroscale quadrature points is collected.
3. For each  $\zeta$ , the microscopic problem is solved with the fine mesh (see Fig. 5.3c) along all 32 trajectories  $\{(\bar{\mathbf{F}}, \widehat{\mathbf{G}})\}$ , and snapshots of the fluctuation displacement  $\mathbf{w}^p$ , weighted stress  $\mathbf{W}^p$  and couple stress  $\mathbf{y}^p$  are gathered.
4. All snapshots computed with the fine mesh are utilized to construct the ROM.

The resulting ROM for  $N = 48$  displacement basis functions has  $Q = 282$  quadrature points (with  $M = 28$  and  $L = 28$ ), and the resulting force-displacement curves for  $\zeta = -0.035\text{mm}$  and  $\zeta = 0.03\text{mm}$  are shown in Figs. 5.8b and 5.8c. The ROM solution approaches the CH2 solution nearly perfectly for both cases, showing the importance of the training dataset. Additionally, the results of the full CH2 solution with the coarse RVE mesh are also shown, which shows much less accurate (post-)buckling behavior.



**Figure 5.8:** (a) Employed coarse RVE mesh for generating a more representative training dataset. The resulting ROM has  $N = 48$  displacement basis functions and  $Q = 282$  integration points, and its solution closely follows the CH2 solution for both (b)  $\zeta = -0.035\text{mm}$  and (c)  $\zeta = 0.03\text{mm}$ . The result of the full CH2 model with the coarse RVE mesh is also shown for comparison.

All simulations were executed on an Intel<sup>®</sup> Xeon<sup>®</sup> Platinum 8260 processor. Computing the DNS solutions for  $\zeta = -0.035\text{mm}$  and  $0.03\text{mm}$  took 194s and 459s with one thread. The significant differences in computational times are caused by the global buckling, which requires many more load steps for convergence as compared to the local buckling. The ROM with  $N = 48$ , after the offline stage is completed, took 23s and 76s for both simulations with one thread, achieving an online speed-up of 6-8x as compared to the DNS solver. With  $N = 78$  and  $N = 112$ , both simulations took 51s ( $\zeta = -0.035\text{mm}$ ) and 231s ( $\zeta = 0.03\text{mm}$ ), and 163s ( $\zeta = -0.035\text{mm}$ ) and 360s ( $\zeta = 0.03\text{mm}$ ).

Concerning the offline stage of the randomly generated dataset, with one thread, 100 samples with each 20 load steps were computed in 1020 s, and constructing the ROM took another 80 s. For the more representative training dataset, the generation took significantly longer, as full two-scale simulations need to be run. Note that the offline and online stage with the ROM can easily be parallelized, since all RVEs can be solved independently, which would increase the speed-up. On the other hand, DNS parallelization is less straightforward and more difficult to achieve. CH2 and POD took much longer as compared to the DNS since the considered scale separation is relatively low.

While this example problem might not be suitable for homogenization since the DNS solution can be obtained quickly, it shows that the ROM can accurately approximate the POD solution, i.e., the proposed algorithm for finding a sparse integration scheme works well. Moreover, the POD solution approaches the CH2 solution (provided the training data is representative), which in turn approximates the DNS well.

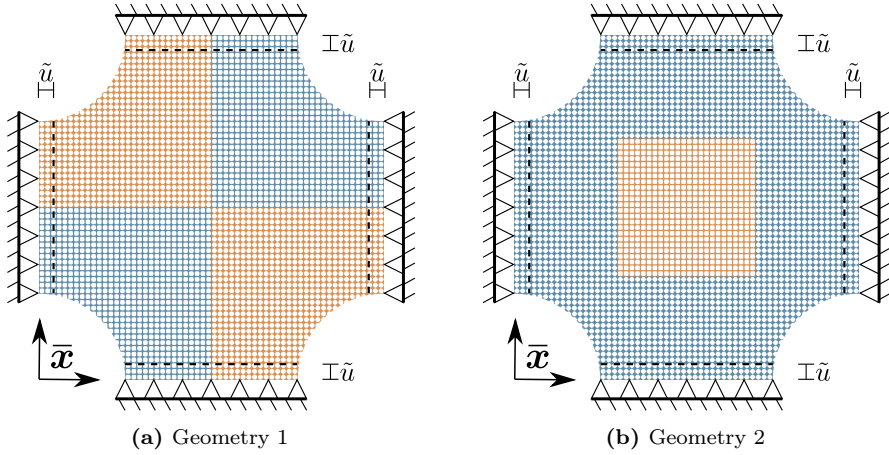
### 5.3.2 Biaxial compression of graded cruciform

The second example deals with the biaxial compression of a graded cruciform-shaped macrostructure with varying hole shapes (i.e., spatially varying  $\zeta$  field) throughout the domain, see Fig. 5.9. Each side edge has length 30 mm and the cut out parts at each corner are quarter circles with a radius of 15 mm. Both example parameterizations shown in Fig. 5.9 are considered and computed with the DNS, CH2 and the ROM solver. The discretized DNS problem for Fig. 5.9a has 3,475,044 DOFs and 800,889 elements and 3,627,610 DOFs and 839,580 elements for Fig. 5.9b. Each side edge is compressed by 2% in the normal direction, while being fixed in the tangential direction. For CH2 and the ROM, additionally, the  $xx$ - and  $yx$ -components of the deformation gradient are fixed to  $\hat{F}_{xx} = 1$  and  $\hat{F}_{yx} = 0$  on the top and bottom edge, and the  $xy$ - and  $yy$ -components of the deformation gradient are fixed to  $\hat{F}_{xy} = 0$  and  $\hat{F}_{yy} = 1$  on the left and right edge. The simulation meshes employed for CH2 and the ROM are shown in Fig. 5.10.

The already trained ROM (with randomly generated training data) is re-used here with  $N = 48, 78$  and 112 displacement basis functions with  $Q = 296, 318$  and 337 integration points, and referred to as ROM48, ROM78 and ROM112.

**Results** Both example geometries are solved with DNS, CH2 and the three ROM solvers (ROM48, ROM78 and ROM112). The resulting force-displacement curves are plotted in Fig. 5.11. For this example, the FE2 solver was also run and able to converge, although it took more than ten times as many loading

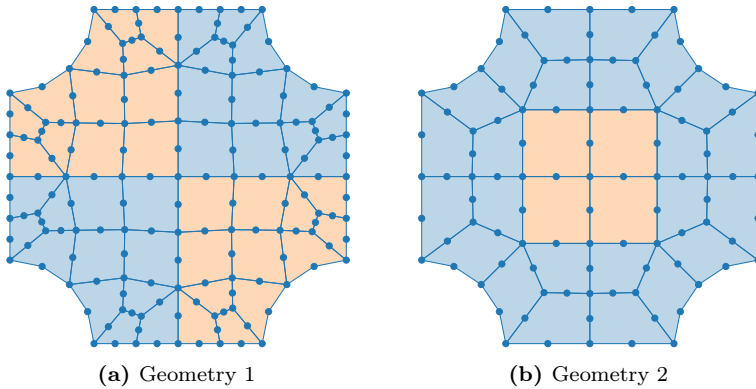




**Figure 5.9:** Two geometries are solved with the DNS solver. (a)  $\zeta = 0.03\text{mm}$  is set for the top right and bottom left part (in blue), and  $\zeta = -0.05\text{mm}$  for the top left and bottom right part (orange). (b)  $\zeta = 0.05\text{mm}$  is set in the center (orange) for  $\bar{\mathbf{x}} \in [18\text{ mm}, 42\text{ mm}] \times [18\text{ mm}, 42\text{ mm}]$  and  $\zeta = -0.075\text{mm}$  elsewhere (blue). Six-noded triangular elements are employed for both geometries, resulting in (a) 3,475,044 DOFs and 800,889 elements, and (b) 3,627,610 DOFs and 839,580 elements.

steps as the DNS solver. The FE2 results are shown in Fig. 5.11 as well. For the first geometry (see Fig. 5.11a), it can clearly be seen that all ROM solutions can recover the CH2 solution nearly perfectly. As compared to the DNS solution, the CH2 solution again predicts a higher pre-buckling stiffness, but the buckling load and post-buckling stage are predicted quite well. With FE2, the pre- and post-buckling stiffness are predicted accurately, but the structure buckles too early. For the second geometry (see Fig. 5.11b), both FE2 and CH2 solutions cannot capture the correct buckling load. The pre- and post-buckling stiffness are predicted slightly more accurately with FE2. Since in this example, the center part is set to  $\zeta = -0.075\text{mm}$ , which is outside the training data (sampled from  $\zeta = \{-0.05\text{mm}, -0.025\text{mm}, 0.0\text{mm}, 0.025\text{mm}, 0.05\text{mm}\}$ ), ROM48 is not able to follow the CH2 solutions closely. The error reduces with  $N = 78$  and 112. For both geometries, both components of the displacement field at the final loading are shown for the DNS and ROM48 solutions in Figs. 5.12 and 5.13, where clearly the trend and magnitudes of the displacement fields are comparable.

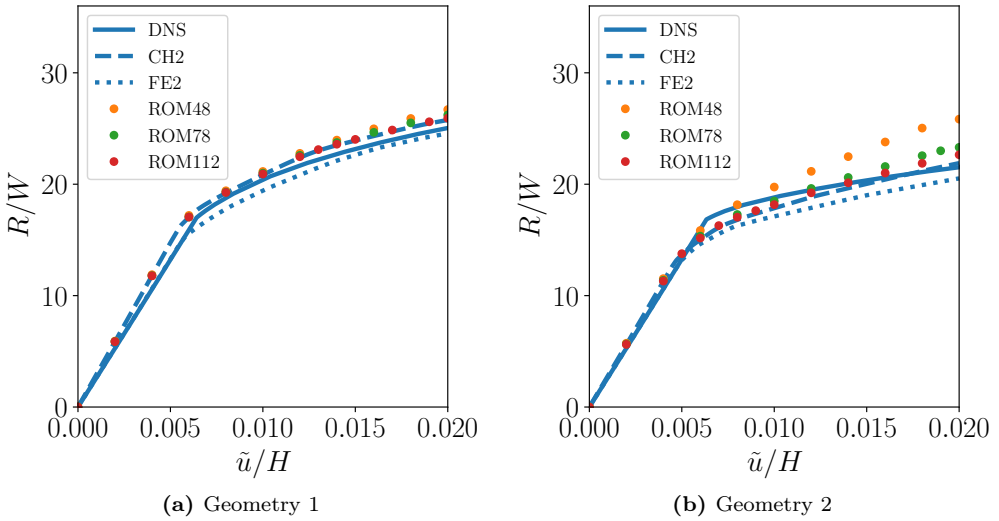
Concerning the run times (using one thread of Intel<sup>®</sup> Xeon<sup>®</sup> Platinum 8260), solving both problems with the DNS solver took 9274s and 8847s, while with ROM48 it took 176s and 113s, implying online speed-ups of 53 and 78 times. With ROM78, the run times were 331s and 275s, meaning speed-ups of 28 and 32 times, and with ROM112, 865s and 593s elapsed until the problems



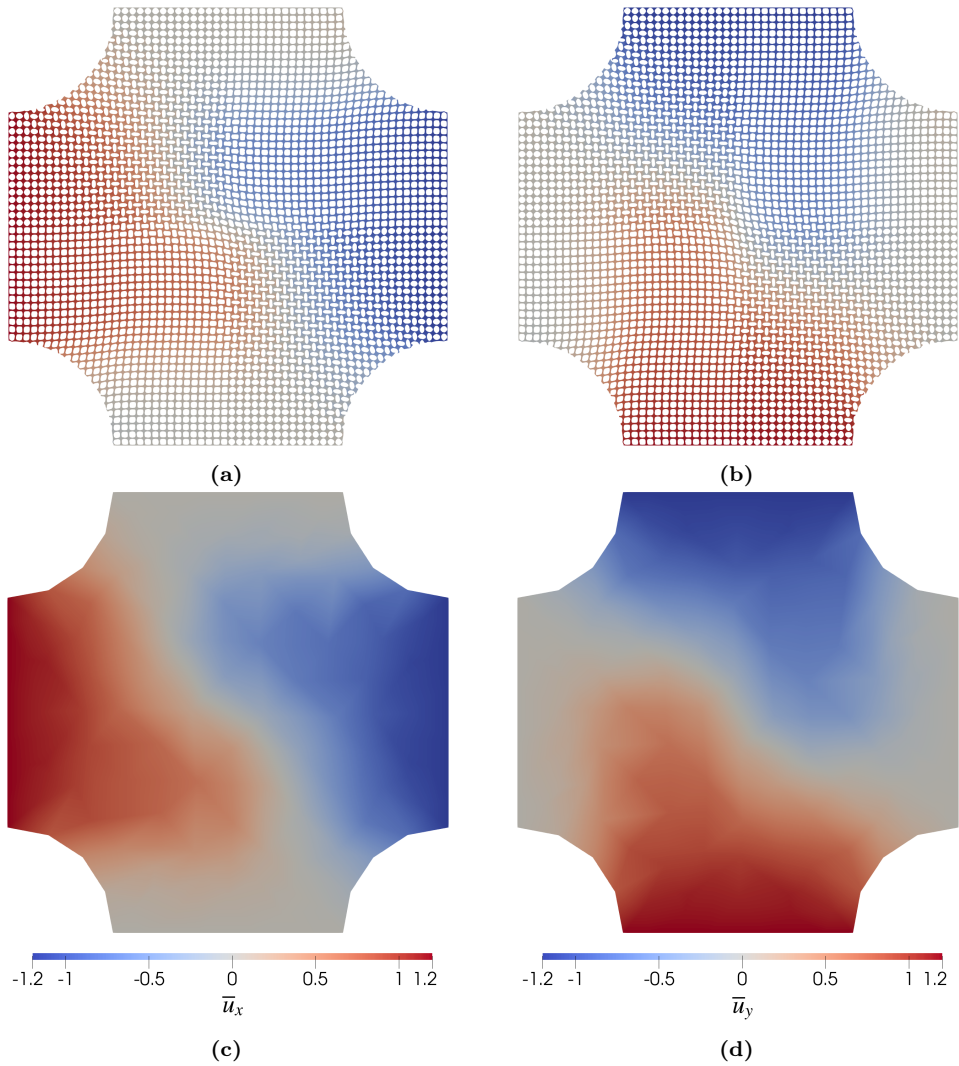
**Figure 5.10:** Mesh of the homogenized cruciform used for CH2 and ROM. (a)  $\zeta = 0.03\text{mm}$  is set for the blue elements and  $\zeta = -0.05\text{mm}$  for the orange elements. (b) The orange and blue elements correspond to  $\zeta = 0.05\text{mm}$  and  $\zeta = -0.075\text{mm}$ . The meshes have (a) 48 and (b) 28 elements with four quadrature points each.

were solved, which still amounts to online speed-ups of 11 and 15 times. The computational costs of the offline stage are the same as reported for the previous example in Section 5.3.1. The obtained speed-ups could be greatly increased by using more threads due to the superior scaling of the multi-scale formulation over the DNS. The run times for FE2 and CH2 are again much higher than the run times of DNS.

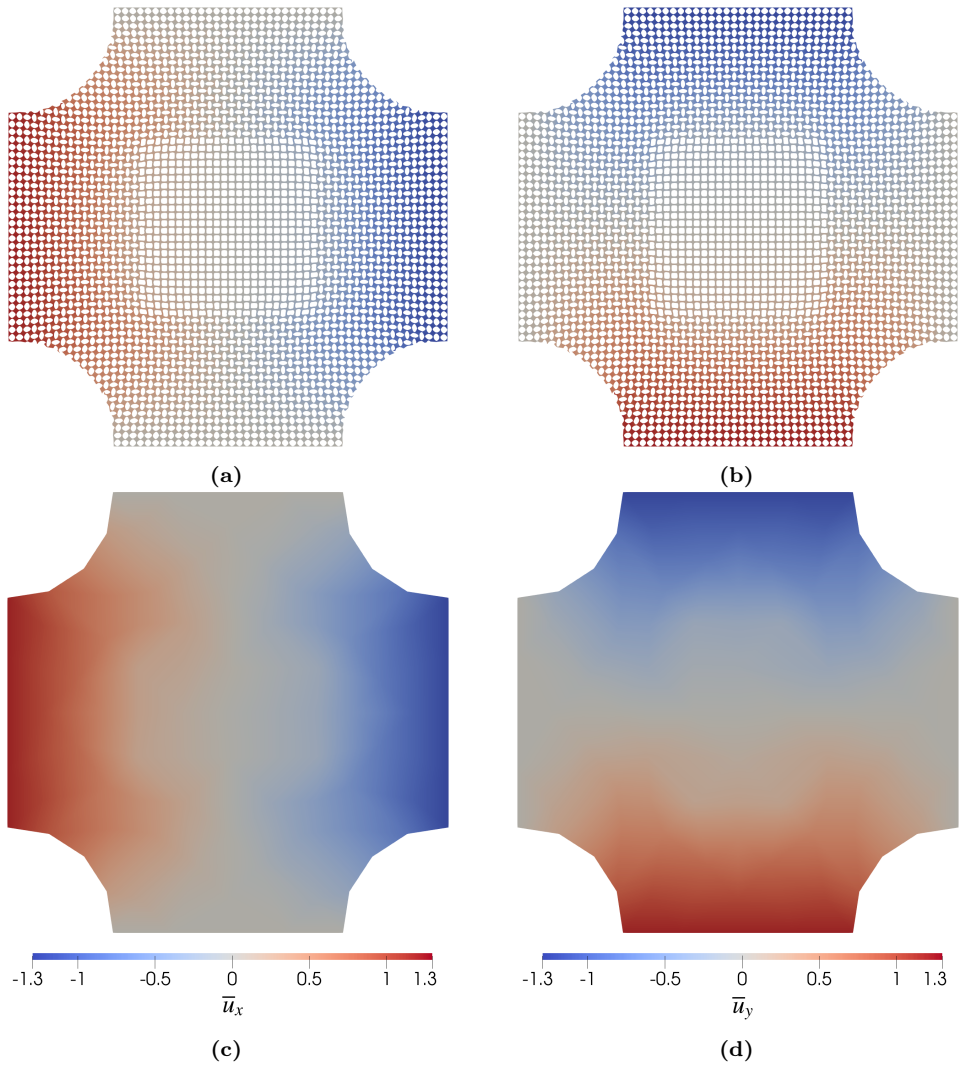
The DNS solution took more than two hours (with one thread), mostly because of the detection of instabilities (i.e., checking the system matrix for negative eigenvalues and eigenvalues close to zero). If a large-scale problem (in 3D) was considered, the DNS solution might become infeasible, since (1) detecting negative eigenvalues is computationally expensive, and (2) negative eigenvalues of the system matrix may cause problems for iterative solvers, while direct solvers become too computationally expensive for large systems. On the other hand, the ROM solution should remain relatively computationally cheap, since the solver can be easily parallelized by solving all RVE problems at the macroscopic integration points in parallel. An additional advantage of the ROM is that, after training, different geometrical parameters inside the macrostructure can be easily treated, while for the DNS the meshing can become expensive and challenging, especially for 3D problems. This makes this ROM an interesting candidate for the material design of buckling structures.



**Figure 5.11:** Force-displacement curves for both example geometries of Figs. 5.9 and 5.10 obtained for DNS, CH2, FE2 and three ROMs with different numbers of basis functions  $N$  and quadrature points  $Q$ . The FE2 solver was able to converge for both geometries, but required more than 10 times as many loading steps as the DNS to converge. (a) All ROM solutions are very close to the CH2 solution. Both CH2 and FE2 solution approximate the DNS quite well. (b) The ROMs are not able to recover the CH2 solution accurately, since  $\zeta = -0.075\text{mm}$  is outside the training data. For higher number of basis functions, the approximation becomes increasingly more accurate, and for ROM112 only small deviations are observed during the post-buckling stage. Both CH2 and FE2 solutions are unable to predict the DNS solution very well.



**Figure 5.12:** Displacement fields for geometry 1 obtained with DNS ((a) and (b)) and with ROM48 ((c) and (d)). The shear band forming along the diagonal is clearly captured.



**Figure 5.13:** Displacement fields for geometry 2 obtained with DNS ((a) and (b)) and with ROM48 ((c) and (d)). The sharp transition in the center due to the sharp change in  $\zeta$  is more or less captured.

## 5.4 Conclusions

In this chapter, we proposed a reduced order model (ROM) for second-order computational homogenization (CH2), based on proper orthogonal decomposition and a novel hyperreduction method that uses ideas from the empirical cubature method and is specifically suited for CH2. Several aspects on the derivation of the reduced system, including the treatment of constraints and geometrical parameterizations, expressions for the effective quantities, and the novel hyperreduction algorithm were discussed. Afterwards, the ROM was tested on two numerical examples, in which the macrostructures are compressed and multi-scale buckling occurs. The ROM solutions were critically evaluated by comparison against the results obtained by the direct numerical simulation (DNS), the full CH2 and the first-order computational homogenization (FE2) models. The first example demonstrated that the proposed hyperreduction algorithm identified integration points and weights that yield accurate results for several parameterizations of the microstructure. The second example considered a more complex application, in which the geometry of the microstructures is varied within the macroscopic domain, and for which the DNS solution takes a substantial amount of time to compute. When employing the ROM for this problem, speed-ups ranging from 10 to 80 as compared to the DNS were achieved with one thread. These speed-ups could be further increased by employing more threads, since, in general, the multi-scale problem scales much better than the DNS.

The comparison of methods additionally revealed some inherent deficiencies of FE2 and CH2. Even though the FE2 solver yielded fairly accurate results for the second example, it required an extensive amount of load steps to complete the simulation. For the first example it was not able to converge at all. On the other hand, the CH2 model is more robust and solved both examples. However, the accuracy was similar to that of the FE2 model, despite the more complicated strain gradient formulation.

To the best of our knowledge, this work is the first attempt of accelerating an enriched computational homogenization formulation. Although we proposed a reduced order model for second-order computational homogenization, we are confident that our findings and employed methods also extend to other formulations, e.g., based on micromorphic computational homogenization. As different parameterizations of the microstructure can be treated as well, interesting applications can be realized with this framework, such as two-scale shape optimization problems, design of materials and uncertainty quantification.

# Chapter 6

## Discussion and outlook

The optimization and design of macrostructures with complex and heterogeneous microstructures is typically infeasible, since the numerical simulation of such structures requires extremely fine meshes, making it computationally expensive. To solve such systems more efficiently, two-scale methods based on computational homogenization, which model both macro- and microscopic features separately, are often employed. The two-scale system is then solved in coupled manner. However, the resulting formulation often still remains computationally expensive due to the repeated solution of the microscopic problem.

Many existing works attempt to accelerate the microscopic problem with non-intrusive or intrusive model order reduction techniques. While many successful methods have been proposed, they typically require a large amount of data and thus are impractical to be used, have a poor interpretability since microscopic fields cannot be recovered, or can only be applied to fixed microstructures, thereby restricting their applicability in optimization and design. Furthermore, existing methods are currently limited to first-order computational homogenization. Enriched formulations, such as second-order computational homogenization, have not been considered thus far, which may be beneficial for problems with small scale separation.

In this thesis, we addressed these issues by developing several reduced-order models to accelerate the microscopic problem in first- and second-order computational homogenization for microstructures that are parameterized by material and geometrical parameters. For first-order computational homogenization, two models were proposed. The first model, named PODGPR and presented in Chapter 3, is a non-intrusive method designed to predict the microscopic stress field. It combines proper orthogonal decomposition with Gaussian process regression, complemented by geometrical transformations.

Through proper orthogonal decomposition, the stress field is compressed into a few reduced coefficients, which are then approximated using Gaussian process regression models. With PODGPR, stress fields that always fulfill the underlying microscopic governing equations as well as the corresponding effective stress and stiffness can be rapidly predicted. We conducted extensive tests on the method, applying it to a range of hyperelastic microstructures including porous and fiber-reinforced materials with varying material and geometrical parameters. The method achieved considerable speed-ups while maintaining high accuracy. Furthermore, when compared with feedforward neural networks, PODGPR showcased superior accuracy and data efficiency. When integrated into a two-scale problem with macrostructural variations, PODGPR accelerated the solution by several orders of magnitude as compared with the full two-scale simulation. However, the extension of PODGPR to history-dependent material behavior is challenging.

To allow the treatment of history-dependent materials, our second model, termed PODECM and presented in Chapter 4, is an intrusive approach and integrates proper orthogonal decomposition with the empirical cubature method. By employing POD, the number of degrees of freedom associated with the microscopic problem is substantially reduced, while the empirical cubature method enables a more efficient system assembly. The performance of PODECM was first studied in a single-scale problem, specifically an RVE of a composite microstructure comprising a soft elasto-plastic matrix with stiff inclusions of variable size, controlled by a single geometrical parameter. Subsequently, we evaluated the performance of PODECM in a two-scale simulation where a porous elasto-plastic microstructure, characterized by two geometrical parameters, was analyzed under non-monotonous loading. Both geometrical parameters were varied throughout the macrostructure, leading to changes in the effective Poisson's ratio from positive to negative. Various PODECM models were constructed, all exhibiting excellent accuracy and achieving significant speed-ups.

Lastly, we introduced a reduced-order model for second-order computational homogenization in Chapter 5. This leverages the proper orthogonal decomposition combined with a novel hyperreduction method tailored specifically for the second-order formulation. We assessed this novel surrogate model using two test cases: a perforated plate under uniaxial compression, which exhibits both local and global buckling instabilities, and a punched cruciform subjected to biaxial compression. Compared to the direct numerical simulation, where the microstructure is fully resolved at the macroscale, and the full second-order solution, our surrogate model demonstrated high accuracy and computational efficiency.



---

While our proposed reduced-order models have achieved excellent results in addressing the two-scale computational homogenization problem, there remain several open questions and exciting avenues for further research:

- **Extension of PODGPR to inelastic materials:** The PODGPR method introduced in Chapter 3 is currently limited to elastic microstructures. An interesting direction would be to extend this approach to inelastic microstructures and explore the use of recurrent neural networks to predict the POD coefficients instead of Gaussian process regression. This raises the question of whether such a model could outperform directly learning the effective constitutive model with recurrent neural networks [40, 141, 78, 94].
- **Adaptive sampling with GPR:** The uncertainty measure of Gaussian process regression is currently not utilized in this work. It has been applied in related works for adaptive sampling [45, 64, 144]. Could such a scheme be effectively employed in computational homogenization, particularly given its high-dimensional parameter spaces, to enhance the efficiency of data sampling?
- **PODGPR for second-order computational homogenization:** The performance of PODGPR has not been investigated in the context of second-order computational homogenization, which is characterized by a high-dimensional parameter space due to strain gradients. How would PODGPR perform, especially when compared to the developed intrusive model?
- **Understanding empirical cubature method:** The empirical cubature method has empirically demonstrated high accuracy in computational homogenization. However, more theoretical understanding on why this method works so effectively is crucial and presents an interesting research question.
- **Extension to 3D and realistic applications:** All numerical examples, that were discussed in this thesis, were defined in 2D under plane strain conditions. While the extension of all proposed methods to 3D appears straightforward, the higher-dimensional loading parameter space in 3D may introduce unforeseen challenges. Investigating this and assessing the computational savings due to reduced order models in 3D would be valuable.
- **Application in inverse or optimization problems:** Despite achieving errors of less than 1% in forward solutions of the two-scale problem,

these errors may potentially lead to very different results for inverse or optimization problems. It would be intriguing to employ the surrogate models for solving inverse or optimization problems and compare the results to those obtained from the full model.

- **Hyperreduction in other formulations:** The developed hyperreduction model in Chapter 5 is highly versatile and can be adapted for other formulations in computational homogenization, such as micromorphic computational homogenization. Assessing this technology in such formulations would be an interesting avenue to explore.

# Appendix A

## Constitutive models

### A.1 Hyperelastic material models

Hyperelastic materials are characterized by history-independent behavior. Their behavior is described through a strain energy density function  $W(\mathbf{F})$ . From that, the second-order 1PK stress tensor  $\mathbf{P}$  and the fourth-order stiffness tensor  $\mathfrak{A}$  are derived with

$$\mathbf{P}(\mathbf{F}) = \frac{\partial W}{\partial \mathbf{F}}, \quad (\text{A.1})$$

$$\mathfrak{A}(\mathbf{F}) = \frac{\partial \mathbf{P}}{\partial \mathbf{F}} = \frac{\partial^2 W}{\partial \mathbf{F} \otimes \partial \mathbf{F}}. \quad (\text{A.2})$$

#### A.1.1 Neo-Hookean material

There exist several formulations for Neo-Hookean materials that have slight differences. In this work, we consider the following definition for the strain energy density function:

$$\begin{aligned} W(\mathbf{F}, \boldsymbol{\lambda}) &= C_1(\text{tr}(\mathbf{C}(\mathbf{F})) - 3 - 2 \ln(J(\mathbf{F}))) + D_1(J(\mathbf{F}) - 1)^2, \\ \mathbf{C}(\mathbf{F}) &= \mathbf{F}^T \mathbf{F}, \\ J(\mathbf{F}) &= \det \mathbf{F}, \end{aligned} \quad (\text{A.3})$$

where  $C_1$  and  $D_1$  are the material parameters stored in  $\boldsymbol{\lambda} = [C_1, D_1]^T$ . The parameters  $C_1$  and  $D_1$  are related to the Young's modulus  $E$  and Poisson's ratio  $\nu$  through,

$$E = \frac{2C_1(3D_1 + 2C_1)}{C_1 + D_1}, \quad \nu = \frac{D_1}{2(C_1 + D_1)}. \quad (\text{A.4})$$

### A.1.2 Bertoldi-Boyce material model

For the studies of elastomeric metamaterials, the authors in [9] employed the following Mooney-Rivlin material model:

$$\begin{aligned}
 W(\mathbf{F}, \boldsymbol{\lambda}) &= C_1(\text{tr}(\mathbf{C}(\mathbf{F})) - 3 - 2 \ln(J(\mathbf{F}))) \\
 &\quad + C_2(\text{tr}(\mathbf{C}(\mathbf{F})) - 3)^2 + D_1(J(\mathbf{F}) - 1)^2, \\
 \mathbf{C}(\mathbf{F}) &= \mathbf{F}^T \mathbf{F}, \\
 J(\mathbf{F}) &= \det \mathbf{F},
 \end{aligned} \tag{A.5}$$

where  $C_1$ ,  $C_2$  and  $D_1$  are the material parameters stored in  $\boldsymbol{\lambda} = [C_1, C_2, D_1]^T$ . Their values were experimentally determined as  $C_1 = 0.55$  MPa,  $C_2 = 0.3$  MPa and  $D_1 = 27.5$  MPa.

## A.2 Plasticity model

The small-strain  $J_2$ -plasticity model with linear isotropic hardening model obeys:

$$\boldsymbol{\sigma} = \mathfrak{C} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^{\text{pl}}), \tag{A.6}$$

$$f^{\text{yield}} = \|\boldsymbol{\sigma}\|_{\text{mises}} - (\sigma_{y0} + H\xi), \tag{A.7}$$

$$\mathbf{r} = \frac{\partial f}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}} \frac{\text{Dev}(\boldsymbol{\sigma})}{\sqrt{\text{Dev}(\boldsymbol{\sigma}) : \text{Dev}(\boldsymbol{\sigma})}}, \tag{A.8}$$

$$\dot{\xi} = \gamma, \tag{A.9}$$

$$\dot{\boldsymbol{\epsilon}}^{\text{pl}} = \gamma \mathbf{r}, \tag{A.10}$$

$$\gamma \geq 0, \quad f^{\text{yield}} \leq 0, \quad \gamma f^{\text{yield}} = 0, \tag{A.11}$$

where  $\boldsymbol{\epsilon}$  is the small-strain tensor,  $\boldsymbol{\epsilon}^{\text{pl}}$  the plastic strain tensor,  $\mathfrak{C}$  is the fourth-order elasticity tensor that can be fully specified by Young's modulus  $E$  and Poisson's ratio  $\nu$ , and  $\boldsymbol{\sigma}$  is the corresponding stress tensor;  $f^{\text{yield}}$  defines the yield surface,  $\text{Dev}(\bullet)$  takes the deviatoric part of a tensor ( $\bullet$ ),  $\|\bullet\|_{\text{mises}} = \sqrt{\frac{3}{2} \text{Dev}(\bullet) : \text{Dev}(\bullet)}$  computes the von Mises stress,  $H$  is the hardening constant,  $\sigma_{y0}$  yield stress,  $\mathbf{r}$  the plastic flow direction,  $\xi$  the equivalent plastic strain that defines the isotropic hardening of the yield surface, and  $\gamma$  is the consistency parameter. The dot above a quantity denotes the (pseudo-)time derivative, which is typically approximated with an implicit Euler scheme with some discretized time step. However, since we consider rate-independent plasticity here, the time step can always be multiplied with the  $\gamma$  and does not play

any role. Instead, time steps become load steps, and sometimes very small load steps are required for the global Newton solver to converge. Given the last converged step  $k$  with internal variables  $(\boldsymbol{\epsilon}^{pl,k}, \xi^k)$  and a new input strain  $\boldsymbol{\epsilon}^{k+1}$ , the new internal variables are obtained with  $\boldsymbol{\epsilon}^{pl,k+1} = \boldsymbol{\epsilon}^{pl,k} + \gamma \mathbf{r}^{k+1}$  and  $\xi^{k+1} = \xi^k + \gamma$ , where  $\mathbf{r}^{k+1}$  is the new plastic flow direction, computed from the new stress  $\boldsymbol{\sigma}^{k+1}$ . The value for the consistency parameter  $\gamma$  is computed such that the Kuhn-Tucker conditions Eq. (A.11) are fulfilled.  $\xi$  and  $\boldsymbol{\epsilon}^{pl}$  are both assumed to be 0 and  $\mathbf{0}$  at the start, i.e.,  $\xi^0 = 0$  and  $\boldsymbol{\epsilon}^{pl,0} = \mathbf{0}$ . For more information on the material model see Simo and Hughes [121].

Following the procedure of Cuitino et al. [24], by employing a multiplicative split of the deformation gradient  $\mathbf{F} = \mathbf{F}^{el} \mathbf{F}^{pl}$ , where  $\mathbf{F}$  is split into its elastic  $\mathbf{F}^{el}$  and plastic  $\mathbf{F}^{pl}$  part with  $\mathbf{F}^{pl} = \mathbf{I}$  at the beginning, the elastic logarithmic strain can be defined as

$$\mathbf{C}_{\log}^{el} := \ln \mathbf{C}^{el} = \ln((\mathbf{F}^{el})^T \mathbf{F}^{el}). \quad (\text{A.12})$$

By interpreting the elastic logarithmic strain  $\mathbf{C}_{\log}^{el}$  as the small-strain tensor  $\boldsymbol{\epsilon}$ , the small-strain constitutive model defined in Eqs. (A.6)–(A.11) is used to compute the stress tensor  $\hat{\mathbf{S}}$  on the intermediate configuration,

$$\hat{\mathbf{S}} := 2\boldsymbol{\sigma}(\mathbf{C}_{\log}^{el}) : \nabla_{\mathbf{C}^{el}} \mathbf{C}_{\log}^{el}, \quad (\text{A.13})$$

while the 1PK stress is recovered from  $\hat{\mathbf{S}}$  as

$$\mathbf{P} = (\mathbf{F}^{el})^{-1} \hat{\mathbf{S}} (\mathbf{F}^{pl})^{-T}. \quad (\text{A.14})$$

Instead of evolving the plastic strain with Eq. (A.10), the plastic deformation gradient  $\mathbf{F}^{pl}$  is evolved according to the following incremental form

$$\mathbf{F}^{pl,k+1} = \exp(\gamma \mathbf{r}^{k+1}) \mathbf{F}^{pl,k}, \quad (\text{A.15})$$

where  $\mathbf{F}^{pl,0} = \mathbf{I}$ .



# Appendix B

## Proper orthogonal decomposition

Principal component analysis (PCA) is a powerful tool in data science to find a low-dimensional approximation of a dataset in Euclidean space. By introducing an appropriate function metric, PCA can be generalized for functions in Hilbert spaces and is referred to as proper orthogonal decomposition (POD). POD can utilize the correlation of solution snapshots of a partial differential equation (PDE) for different parameters and find a low-dimensional orthonormal basis, often reducing the number of unknowns to a small fraction of the dimension of the original high-fidelity model. The procedure for computing the low-dimensional basis is outlined in the following.

Given a function  $\mathbf{u} \in \mathcal{V}$ , where  $\mathcal{V}$  is a Hilbert space equipped with an inner product  $(\bullet, \bullet)_{\mathcal{V}}$ , that is the solution to a PDE, parameterized by parameters  $\boldsymbol{\mu}$ . We are looking for a low-dimensional representation of  $\mathbf{u}$  with

$$\mathbf{u}(\mathbf{x}; \boldsymbol{\mu}) = \sum_{n=1}^N a_n(\boldsymbol{\mu}) \mathbf{v}_n(\mathbf{x}), \quad (\text{B.1})$$

with parameter-dependent coefficients  $a_n(\boldsymbol{\mu}) \in \mathbb{R}$ , fixed global basis functions  $\mathbf{v}_n(\mathbf{x}) \in \mathcal{V}$ , and  $N$  the number of basis functions.

To compute the basis functions  $\mathbf{v}_n(\mathbf{x})$ , first,  $N_{\text{pod}}$  high-fidelity simulations for different parameters are computed and solution snapshots collected with  $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(N_{\text{pod}})}$ . Then, the correlation matrix  $\mathbf{C} \in \mathbb{R}^{N_{\text{pod}} \times N_{\text{pod}}}$  is formed by computing the inner product between every pair of two snapshots,

$$C_{ij} = \left( \mathbf{u}^{(i)}, \mathbf{u}^{(j)} \right)_{\mathcal{V}}, \quad (\text{B.2})$$

for  $i, j = 1, 2, \dots, N_{\text{pod}}$ . After obtaining the correlation matrix, an (approximate) eigenvalue decomposition is computed,

$$\mathbf{C} \approx \sum_{n=1}^N \lambda_n \mathbf{y}_n \otimes \mathbf{y}_n, \quad (\text{B.3})$$

with eigenvalues  $\lambda_n$  and eigenvectors  $\mathbf{y}_n$ . The number of basis functions  $N \leq N_{\text{pod}}$  is often determined from the energy criterion

$$\frac{\sum_{n=1}^N \lambda_n}{\sum_{n=1}^{N_{\text{pod}}} \lambda_n} > \mathcal{E}_{\text{pod}}, \quad (\text{B.4})$$

where  $\mathcal{E}_{\text{pod}}$  corresponds to the energy captured by  $N$  basis functions and is specified by the user. Moreover, the eigenvalues can also reveal if a problem is reducible using POD: if the eigenvalues decay rapidly, the solution manifold can be accurately captured by a few basis functions.

Finally, the  $n$ -th basis function can be computed with

$$\mathbf{v}_n = \sum_{i=1}^{N_{\text{pod}}} c_{in} \mathbf{u}^{(i)}, \quad (\text{B.5})$$

where  $c_{in} = (\mathbf{y}_n)_i / \sqrt{\lambda_n}$  and  $(\mathbf{y}_n)_i$  denotes the  $i$ -th component of  $\mathbf{y}_n$ .

By construction, the POD basis is orthonormal in  $\mathcal{V}$ , i.e.,

$$(\mathbf{v}_i, \mathbf{v}_j)_{\mathcal{V}} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}. \quad (\text{B.6})$$



# Appendix C

## Proof of Eq. (3.14)

With  $\mathbf{B}_l$  the  $l$ -th basis function of the weighted stress, cf. Eq. (3.7), we want to prove that

$$\int_{\Omega^p} \mathbf{B}_l(\mathbf{x}^p) \mathbf{F}_\mu^T d\mathbf{x}^p \stackrel{!}{=} \int_{\Omega^p} \mathbf{B}_l(\mathbf{x}^p) d\mathbf{x}^p \quad (\text{C.1})$$

for all  $\boldsymbol{\mu}$  and all  $l$ . First of all,  $\mathbf{F}_\mu$  can be written as

$$\mathbf{F}_\mu = \mathbf{I} + \nabla_{\mathbf{x}^p} \mathbf{d}_\mu \quad (\text{C.2})$$

with  $\mathbf{d}_\mu(\mathbf{x}^p) \equiv \mathbf{d}(\mathbf{x}^p; \boldsymbol{\mu})$  the transformation displacement, so the left-hand side of Eq. (C.1) splits into

$$\int_{\Omega^p} \mathbf{B}_l(\mathbf{x}^p) \mathbf{F}_\mu^T d\mathbf{x}^p = \int_{\Omega^p} \mathbf{B}_l(\mathbf{x}^p) d\mathbf{x}^p + \int_{\Omega^p} \mathbf{B}_l(\mathbf{x}^p) (\nabla_{\mathbf{x}^p} \mathbf{d}_\mu)^T d\mathbf{x}^p. \quad (\text{C.3})$$

To prove Eq. (C.1), we thus need to show that

$$\int_{\Omega^p} \mathbf{B}_l(\mathbf{x}^p) (\nabla_{\mathbf{x}^p} \mathbf{d}_\mu)^T d\mathbf{x}^p \stackrel{!}{=} \mathbf{0}. \quad (\text{C.4})$$

Without loss of generality, assume only a single training snapshot, obtained for parameters  $(\bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$  on a domain  $\Omega^{\boldsymbol{\mu}^*}$ , and a transformation map  $\Phi_{\boldsymbol{\mu}^*} : \Omega^p \rightarrow \Omega^{\boldsymbol{\mu}^*}$ ,  $\mathbf{x}^{\boldsymbol{\mu}^*} = \Phi_{\boldsymbol{\mu}^*}(\mathbf{x}^p)$ . Thus, there is only a single basis function,

$$\mathbf{B}_1(\mathbf{x}^p) = P(\Phi_{\boldsymbol{\mu}^*}(\mathbf{x}^p); \bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{F}_{\boldsymbol{\mu}^*}^{-T} |\det \mathbf{F}_{\boldsymbol{\mu}^*}|, \quad (\text{C.5})$$

and Eq. (C.4) becomes

$$\int_{\Omega^p} P(\Phi_{\boldsymbol{\mu}^*}(\mathbf{x}^p); \bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{F}_{\boldsymbol{\mu}^*}^{-T} |\det \mathbf{F}_{\boldsymbol{\mu}^*}| (\nabla_{\mathbf{x}^p} \mathbf{d}_\mu)^T d\mathbf{x}^p. \quad (\text{C.6})$$

We now push the integral in Eq. (C.6) forward onto the domain  $\Omega^{\mu^*}$ . By introducing the inverse mapping,

$$\Phi_{\mu^*}^{-1} : \Omega^{\mu^*} \rightarrow \Omega^P, \quad \mathbf{x}^P = \Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*}), \quad d\mathbf{x}^{\mu^*} = |\det \mathbf{F}_{\mu^*}| d\mathbf{x}^P, \quad (\text{C.7})$$

$\mathbf{F}_{\mu^*}$  is transformed with

$$\begin{aligned} \mathbf{F}_{\mu^*} &= \nabla_{\mathbf{x}^P} \Phi_{\mu^*}(\mathbf{x}^P) \\ &= \nabla_{\mathbf{x}^{\mu^*}} (\Phi_{\mu^*}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*}))) (\nabla_{\mathbf{x}^{\mu^*}} \mathbf{x}^P)^{-1} \\ &= (\nabla_{\mathbf{x}^{\mu^*}} \mathbf{x}^P)^{-1}, \end{aligned} \quad (\text{C.8})$$

from which it follows that

$$\mathbf{F}_{\mu^*}^{-T} = (\nabla_{\mathbf{x}^{\mu^*}} \mathbf{x}^P)^T. \quad (\text{C.9})$$

Furthermore, the partial derivative  $\nabla_{\mathbf{x}^P} \mathbf{d}_{\mu}(\mathbf{x}^P)$  can be expressed as

$$\nabla_{\mathbf{x}^P} \mathbf{d}_{\mu}(\mathbf{x}^P) = \nabla_{\mathbf{x}^{\mu^*}} (\mathbf{d}_{\mu}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*}))) (\nabla_{\mathbf{x}^{\mu^*}} \mathbf{x}^P)^{-1}, \quad (\text{C.10})$$

or

$$(\nabla_{\mathbf{x}^P} \mathbf{d}_{\mu})^T = (\nabla_{\mathbf{x}^{\mu^*}} \mathbf{x}^P)^{-T} \left( \nabla_{\mathbf{x}^{\mu^*}} (\mathbf{d}_{\mu}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*}))) \right)^T \quad (\text{C.11})$$

by transposing both sides of Eq. (C.10). With aid of Eqs. (C.7), (C.9) and (C.11), Eq. (C.6) becomes

$$\int_{\Omega^{\mu^*}} \mathbf{P}(\mathbf{x}^{\mu^*}; \bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \left( \nabla_{\mathbf{x}^{\mu^*}} (\mathbf{d}_{\mu}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*}))) \right)^T d\mathbf{x}^{\mu^*}. \quad (\text{C.12})$$

Utilizing the divergence theorem, Eq. (C.12) can be rewritten as

$$\begin{aligned} &\int_{\partial\Omega^{\mu^*}} \mathbf{t}_0 \otimes \mathbf{d}_{\mu}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*})) d\mathbf{x}^{\mu^*} \\ &- \int_{\Omega^{\mu^*}} \mathbf{d}_{\mu}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*})) \otimes (\nabla_{\mathbf{x}^{\mu^*}} \cdot \mathbf{P}(\mathbf{x}^{\mu^*}; \bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)) d\mathbf{x}^{\mu^*}, \end{aligned} \quad (\text{C.13})$$

where  $\mathbf{t}_0 := \mathbf{P}(\mathbf{x}^{\mu^*}; \bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \mathbf{n}$  is the traction vector with  $\mathbf{n}$  the outer unit normal along the boundary  $\partial\Omega^{\mu^*}$ . Using the fact that the training snapshot fulfills the balance of linear momentum  $\nabla_{\mathbf{x}^{\mu^*}} \cdot \mathbf{P}(\mathbf{x}^{\mu^*}; \bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathbf{0}$  on the domain  $\Omega^{\mu^*}$ , the latter part of Eq. (C.13) becomes  $\mathbf{0}$ , and therefore,

$$\begin{aligned} &\int_{\Omega^{\mu^*}} \mathbf{P}(\mathbf{x}^{\mu^*}; \bar{\mathbf{F}}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \left( \nabla_{\mathbf{x}^{\mu^*}} (\mathbf{d}_{\mu}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*}))) \right)^T d\mathbf{x}^{\mu^*} \\ &= \int_{\partial\Omega^{\mu^*}} \mathbf{t}_0 \otimes \mathbf{d}_{\mu}(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*})) d\mathbf{x}^{\mu^*}. \end{aligned} \quad (\text{C.14})$$

---

Due to the definition of the auxiliary problem in Eq. (3.2),  $\mathbf{d}_\mu(\Phi_{\mu^*}^{-1}(\mathbf{x}^{\mu^*}))$  is  $\mathbf{0}$  on the boundary  $\partial\Omega^{\mu^*}$  and therefore the boundary integral in Eq. (C.14) always results in the zero tensor, meaning that the integral in Eq. (C.6) vanishes for all  $\mu$ .

Since each basis function  $\mathbf{B}_l$  is a linear combination of converged weighted stress fields, from the linearity, the integral on the left hand side of Eq. (C.4) vanishes as well, which is what we wanted to prove and, thus, Eq. (C.1) holds. ■



# Appendix D

## Effective sensitivities with corrected coefficients

When correcting the POD coefficients to fulfill the macroscopic balance of angular momentum in Eq. (3.27), the effective stress is computed as,

$$\bar{\mathbf{P}}(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{l=1}^L \alpha_l^\perp(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \bar{\mathbf{B}}_l, \quad (\text{D.1})$$

$$\alpha_l^\perp(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \hat{\alpha}_l^*(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) - \frac{\hat{\boldsymbol{\alpha}}^*(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \cdot \boldsymbol{\gamma}(\bar{\mathbf{F}})}{\boldsymbol{\gamma}(\bar{\mathbf{F}}) \cdot \boldsymbol{\gamma}(\bar{\mathbf{F}})} \gamma_l(\bar{\mathbf{F}}), \quad (\text{D.2})$$

where  $\hat{\boldsymbol{\alpha}}^*(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu})$  are the coefficients predicted by the GPRs and  $\boldsymbol{\gamma}(\bar{\mathbf{F}})$  was defined in Eq. (3.34). The second part in Eq. (D.2) corresponds to the correction term, and the effective sensitivities, as defined in Eqs. (3.17)–(3.19), also need to be corrected by the derivative of this correction term:

$$\bar{\boldsymbol{\alpha}} = \nabla_{\bar{\mathbf{F}}} \bar{\mathbf{P}} = \sum_{l=1}^L \bar{\mathbf{B}}_l \otimes \nabla_{\bar{\mathbf{F}}} \alpha_l^\perp, \quad (\text{D.3})$$

$$\nabla_{\boldsymbol{\lambda}} \bar{\mathbf{P}} = \sum_{l=1}^L \bar{\mathbf{B}}_l \otimes \nabla_{\boldsymbol{\lambda}} \alpha_l^\perp, \quad (\text{D.4})$$

$$\nabla_{\boldsymbol{\mu}} \bar{\mathbf{P}} = \sum_{l=1}^L \bar{\mathbf{B}}_l \otimes \nabla_{\boldsymbol{\mu}} \alpha_l^\perp. \quad (\text{D.5})$$

With the derivative of  $\gamma_l$  in Eq. (3.34) with respect to  $\bar{\mathbf{F}}$

$$\begin{aligned}\nabla_{\bar{\mathbf{F}}}\gamma_l &= \bar{B}_{l,11} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \bar{B}_{l,12} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} - \bar{B}_{l,21} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} - \bar{B}_{l,22} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \bar{B}_{l,21} & \bar{B}_{l,22} \\ \bar{B}_{l,11} & \bar{B}_{l,12} \end{bmatrix},\end{aligned}\quad (\text{D.6})$$

and

$$\nabla_{\bar{\mathbf{F}}}((\gamma \cdot \gamma)^{-1}) = -2(\gamma \cdot \gamma)^{-2} \sum_{l=1}^L \gamma_l \nabla_{\bar{\mathbf{F}}}\gamma_l, \quad (\text{D.7})$$

the derivative of the correction term with respect to  $\bar{\mathbf{F}}$  can be written as:

$$\begin{aligned}&\nabla_{\bar{\mathbf{F}}} \left( \frac{\hat{\boldsymbol{\alpha}}^*(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \cdot \gamma(\bar{\mathbf{F}})}{\gamma(\bar{\mathbf{F}}) \cdot \gamma(\bar{\mathbf{F}})} \gamma_l(\bar{\mathbf{F}}) \right) \\ &= \left( \sum_{l=1}^L \gamma_l \nabla_{\bar{\mathbf{F}}}\hat{\boldsymbol{\alpha}}_l^* \right) (\gamma \cdot \gamma)^{-1} \gamma_l \\ &+ \left( \sum_{l=1}^L \hat{\boldsymbol{\alpha}}_l^* \nabla_{\bar{\mathbf{F}}}\gamma_l \right) (\gamma \cdot \gamma)^{-1} \gamma_l \\ &+ (\hat{\boldsymbol{\alpha}}^* \cdot \gamma) \nabla_{\bar{\mathbf{F}}}((\gamma \cdot \gamma)^{-1}) \gamma_l \\ &+ \frac{\hat{\boldsymbol{\alpha}}^* \cdot \gamma}{\gamma \cdot \gamma} \nabla_{\bar{\mathbf{F}}}\gamma_l.\end{aligned}\quad (\text{D.8})$$

Similarly, the derivative with respect to  $\boldsymbol{\lambda}$  is,

$$\nabla_{\boldsymbol{\lambda}} \left( \frac{\hat{\boldsymbol{\alpha}}^*(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \cdot \gamma(\bar{\mathbf{F}})}{\gamma(\bar{\mathbf{F}}) \cdot \gamma(\bar{\mathbf{F}})} \gamma_l(\bar{\mathbf{F}}) \right) = \left( \sum_{l=1}^L \gamma_l \nabla_{\boldsymbol{\lambda}}\hat{\boldsymbol{\alpha}}_l^* \right) (\gamma \cdot \gamma)^{-1} \gamma_l, \quad (\text{D.9})$$

and with respect to  $\boldsymbol{\mu}$ ,

$$\nabla_{\boldsymbol{\mu}} \left( \frac{\hat{\boldsymbol{\alpha}}^*(\bar{\mathbf{F}}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \cdot \gamma(\bar{\mathbf{F}})}{\gamma(\bar{\mathbf{F}}) \cdot \gamma(\bar{\mathbf{F}})} \gamma_l(\bar{\mathbf{F}}) \right) = \left( \sum_{l=1}^L \gamma_l \nabla_{\boldsymbol{\mu}}\hat{\boldsymbol{\alpha}}_l^* \right) (\gamma \cdot \gamma)^{-1} \gamma_l. \quad (\text{D.10})$$

# Appendix E

## Feedforward neural networks

Feedforward neural networks are regression models that, given some training data  $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{N_{\text{train}}}$ , learn a function  $\mathbf{f}$  that maps inputs  $\mathbf{x}$  to outputs  $\mathbf{y} \approx \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ , where  $\boldsymbol{\theta}$  contains the parameters to be optimized and depends on the network architecture. In general, the network has an input layer, an output layer, and can have so-called hidden layers in between the input and output layer. Each layer has a specified dimension, often referred to as number of neurons. The approximating function  $\mathbf{f}$  can then be written as a recursion with  $\mathbf{x}_0 \equiv \mathbf{x}$ :

$$\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x}_{N_h+1} \quad (\text{E.1})$$

$$\mathbf{x}_{l+1} = \boldsymbol{\sigma}_l(\mathbf{A}_l \mathbf{x}_l + \mathbf{b}_l) \quad (\text{E.2})$$

for  $l = 0, \dots, N_h$ , where  $N_h$  is the number of hidden layers. The matrices  $\mathbf{A}_l \in \mathbb{R}^{N_n^{l+1} \times N_n^l}$  and vectors  $\mathbf{b}_l \in \mathbb{R}^{N_n^{l+1}}$ , also known as weights and biases, contain the parameters to be optimized, where  $N_n^l$  is the number of neurons in the  $l$ -th layer, with  $N_n^0 = \dim \mathbf{x}$  and  $N_n^{N_h+1} = \dim \mathbf{y}$ . The activation functions  $\boldsymbol{\sigma}_l$  are applied element-wise to each entry of their input and usually do not contain any parameters to be optimized. Popular activation functions comprise the RELU-function (rectified linear unit) with

$$\text{RELU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}, \quad (\text{E.3})$$

or ELU-function (exponential linear unit),

$$\text{ELU}(x) = \begin{cases} x & x \geq 0 \\ \alpha(\exp(x) - 1) & x < 0 \end{cases}, \quad (\text{E.4})$$

where  $\alpha$  is chosen by the user, often set to  $\alpha = 1$  by default. The number of parameters  $N_{\boldsymbol{\theta}}$  to be optimized in the NN can be computed with

$$N_{\boldsymbol{\theta}} := \dim \boldsymbol{\theta} = \sum_{l=0}^{N_h} (N_n^{l+1} \cdot N_n^l + N_n^{l+1}), \quad (\text{E.5})$$

if the weights and biases are not further constrained. To find optimal parameters  $\boldsymbol{\theta}^*$ , the following unconstrained optimization problem is posed and solved

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}}}} \sum_{i=1}^{N_{\text{train}}} \mathcal{L}(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}), \quad (\text{E.6})$$

where  $\mathcal{L}$  is the loss function that measures the discrepancy between the predicted output  $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$  with the true output  $\mathbf{y}^{(i)}$ . Often times, the mean squared error (MSE), defined by

$$\mathcal{L}^{\text{MSE}}(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{N_{\text{train}}} (\mathbf{y} - \tilde{\mathbf{y}})^T \cdot (\mathbf{y} - \tilde{\mathbf{y}}), \quad (\text{E.7})$$

is chosen. Since the optimization problem in Eq. (E.6) is non-convex, it is typically approximately solved using a stochastic gradient descent (SGD), such as the Adam optimizer [65]. The gradients, required for the optimization, can be efficiently computed with automatic differentiation. For more information on neural networks and deep learning, we refer to [13, 41].



# Appendix F

## Empirical cubature method

The greedy algorithm to select the reduced set of integration points with corresponding weights  $\{(\mathbf{x}_q, w_q)\}_{q=1}^{\hat{Q}}$  among all integration points and weights of the full model  $\{(\hat{\mathbf{x}}_q, \hat{w}_q)\}_{q=1}^{\hat{Q}}$ , as presented in [48], is briefly summarized here. Starting from Eq. (4.12),

$$\sum_{q=1}^{\hat{Q}} \hat{w}_q [\nabla_{\mathbf{x}^p} \mathbf{v}_i : \mathbf{B}_l] |_{\hat{\mathbf{x}}_q} \approx \sum_{q=1}^Q w_q [\nabla_{\mathbf{x}^p} \mathbf{v}_i : \mathbf{B}_l] |_{\mathbf{x}_q}, \quad (\text{F.1})$$

with  $i = 1, \dots, N$  and  $l = 1, \dots, L$ , both left and right hand side can be compactly written as

$$\hat{\mathbf{b}} = \hat{\mathbf{A}} \hat{\mathbf{w}} \approx \hat{\mathbf{A}}_{\bullet \mathcal{I}} \mathbf{w}, \quad (\text{F.2})$$

where

$$\hat{\mathbf{A}} = \begin{bmatrix} (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_1) |_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_1) |_{\hat{\mathbf{x}}_{\hat{Q}}} \\ \vdots & & \vdots \\ (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_L) |_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_1 : \mathbf{B}_L) |_{\hat{\mathbf{x}}_{\hat{Q}}} \\ \vdots & & \vdots \\ (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_1) |_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_1) |_{\hat{\mathbf{x}}_{\hat{Q}}} \\ \vdots & & \vdots \\ (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_L) |_{\hat{\mathbf{x}}_1} & \dots & (\nabla_{\mathbf{x}^p} \mathbf{v}_N : \mathbf{B}_L) |_{\hat{\mathbf{x}}_{\hat{Q}}} \end{bmatrix} \in \mathbb{R}^{NL \times \hat{Q}}, \quad (\text{F.3})$$

and

$$\hat{\mathbf{w}} = \begin{bmatrix} \hat{w}_1 \\ \vdots \\ \hat{w}_{\hat{Q}} \end{bmatrix} \in \mathbb{R}^{\hat{Q}}, \quad \hat{\mathbf{b}} = \begin{bmatrix} \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_1 : \mathbf{B}_1 d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_1 : \mathbf{B}_L d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_N : \mathbf{B}_1 d\mathbf{x}^P \\ \vdots \\ \int_{\Omega^P} \nabla_{\mathbf{x}^P} \mathbf{v}_N : \mathbf{B}_L d\mathbf{x}^P \end{bmatrix} \in \mathbb{R}^{NL}. \quad (\text{F.4})$$

It is assumed that the integrals in  $\hat{\mathbf{b}}$  are exactly computed through  $\hat{\mathbf{A}}\hat{\mathbf{w}}$ .  $\mathcal{I}$  denotes a set of non-repeating indices with  $|\mathcal{I}| = Q$  and  $\hat{\mathbf{A}}_{\bullet\mathcal{I}} \in \mathbb{R}^{NL \times Q}$  is the submatrix of  $\hat{\mathbf{A}}$  with  $Q$  selected columns according to the entries of  $\mathcal{I}$ , with each column corresponding to one integration point. The weights  $\mathbf{w} = [w_1, \dots, w_Q] \in \mathbb{R}^Q$  are computed from the non-negative least squares problem,

$$\mathbf{w}^{\text{LS}} = \arg \min_{\mathbf{w} \geq \mathbf{0}} \left\| \hat{\mathbf{b}} - \hat{\mathbf{A}}_{\bullet\mathcal{I}} \mathbf{w} \right\|_2. \quad (\text{F.5})$$

The indices in  $\mathcal{I}$  are selected one by one, according to Algorithm 4. The algorithm is terminated if the residual of the least squares problem in Eq. (F.5) is below a user-defined error  $\epsilon_{\text{ECM}}$ , the maximum number of iteration  $k_{\text{max}}$  exceeded, or a specified number of integration points  $Q_{\text{max}}$  determined.

---

**Algorithm 4** Integration point selection algorithm
 

---

**Offline Stage:**  $\hat{\mathbf{A}}, \hat{\mathbf{b}}, \epsilon_{\text{ECM}}, k_{\text{max}}, Q_{\text{max}}$ 
**Online Stage:**  $\mathbf{w}^{\text{LS}}, \mathcal{I}$ 

 Initialize empty list of selected columns  $\mathcal{I} \leftarrow \emptyset$ 

 Initialize list of candidate indices  $\mathcal{C} \leftarrow \{1, \dots, \hat{Q}\}$ 

 Set iteration number  $k \leftarrow 0$ 

 Set initial residual  $\hat{\mathbf{r}} \leftarrow \hat{\mathbf{b}}$ 
**while**  $k < k_{\text{max}}$  **do**
 $k \leftarrow k + 1$ 

 Find the column  $i$  of  $\hat{\mathbf{A}}$  with

$$i = \arg \max_{j \in \mathcal{C}} \frac{\hat{\mathbf{A}}_{\bullet j}^T \hat{\mathbf{r}}}{\sqrt{\hat{\mathbf{A}}_{\bullet j}^T \hat{\mathbf{A}}_{\bullet j}}}$$

 Add selected index  $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$ 

 Remove selected index from candidates  $\mathcal{C} \leftarrow \mathcal{C} \setminus \{i\}$ 

 Solve Eq. (F.5) for  $\mathbf{w}^{\text{LS}}$ 

 Compute residual  $\hat{\mathbf{r}} = \hat{\mathbf{b}} - \hat{\mathbf{A}}_{\bullet \mathcal{I}} \mathbf{w}^{\text{LS}}$ 
**if**  $\|\hat{\mathbf{r}}\|_2 < \epsilon_{\text{ECM}}$  or  $|\mathcal{I}| = Q_{\text{max}}$  **then**
 $\mathbf{return} \mathbf{w}^{\text{LS}}, \mathcal{I}$ 
 $\triangleright$  Algorithm is converged

**end if**
**end while**


---



# Bibliography

- [1] M. M. Ameen, R. H. Peerlings, and M. G. Geers. A quantitative assessment of the scale separation limits of classical and higher-order asymptotic homogenization. *European Journal of Mechanics, A/Solids*, 71:89–100, 9 2018.
- [2] M. M. Ameen, O. Rokoš, R. H. J. Peerlings, and M. G. D. Geers. Size effects in nonlinear periodic materials exhibiting reversible pattern transformations. *Mechanics of Materials*, 124:55–70, 9 2018.
- [3] I. Babuška, U. Banerjee, and J. E. Osborn. Generalized finite element method — main ideas, results and perspective. *International Journal of Computational Methods*, 01(01):67–103, 6 2004.
- [4] N. S. Bakhvalov and G. Panasenko. *Homogenisation: averaging processes in periodic media: mathematical problems in the mechanics of composite materials*, volume 36. Springer Science & Business Media, 2012.
- [5] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 11 2004.
- [6] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John Wiley & sons, 2014.
- [7] A. Bensoussan, J.-L. Lions, and G. Papanicolaou. *Asymptotic analysis for periodic structures*, volume 374. American Mathematical Soc., 2011.
- [8] K. Bertoldi and M. C. Boyce. Mechanically triggered transformations of phononic band gaps in periodic elastomeric structures. *Physical Review B*, 77(5), 2 2008.
- [9] K. Bertoldi, M. C. Boyce, S. Deschanel, S. M. 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*, 56(8):2642–2668, 8 2008.
- [10] A. Bērziņš, J. Helmig, F. Key, and S. Elgeti. Standardized Non-Intrusive Reduced Order Modeling Using Different Regression Models With Application to Complex Flow Problems. *arXiv*, 6 2020.
- [11] M. A. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, and W. K. Liu. A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. *Computer Methods in Applied Mechanics and Engineering*, 320:633–667, 2017.
- [12] O. R. Bingol and A. Krishnamurthy. NURBS-Python: An open-source object-oriented NURBS modeling framework in Python. *SoftwareX*, 9:85–94, 2019.
- [13] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [14] R. Biswas and L. H. Poh. A micromorphic computational homogenization framework for heterogeneous materials. *Journal of the Mechanics and Physics of Solids*, 102:187–208, 2017.
- [15] P. J. Blanco, P. J. Sánchez, E. A. De Souza Neto, and R. A. Feijóo. The method of multiscale virtual power for the derivation of a second order mechanical model. *Mechanics of Materials*, 99:53–67, 8 2016.
- [16] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*, volume 44. A K Peters/CRC Press, 10 2010.
- [17] D. S. Brauer, C. Rüssel, S. Vogt, J. Weisser, and M. Schnabelrauch. Degradable phosphate glass fiber reinforced polymer matrices: mechanical properties and cell response. *Journal of Materials Science: Materials in Medicine*, 19(1):121–127, 2008.
- [18] A. Buhr, L. Iapichino, M. Ohlberger, S. Rave, F. Schindler, and K. Smetana. Localized model reduction for parameterized problems. In *Snapshot-Based Methods and Algorithms*, pages 245–305. De Gruyter, 12 2020.
- [19] A. Buhr and K. Smetana. Randomized local model order reduction. *SIAM Journal on Scientific Computing*, 40(4):A2120–A2151, 2018.

- [20] M. Caicedo, J. L. Mroginski, S. Toro, M. Raschi, A. Huespe, and J. Oliver. High Performance Reduced Order Modeling Techniques Based on Optimal Energy Quadrature: Application to Geometrically Non-linear Multiscale Inelastic Material Modeling. *Archives of Computational Methods in Engineering*, 26(4):771–792, 9 2019.
- [21] R. Chakir and J. K. Hammond. A non-intrusive reduced basis method for elastoplasticity problems in geotechnics. *Journal of Computational and Applied Mathematics*, 337:1–17, 2018.
- [22] T. Chapman, P. Avery, P. Collins, and C. Farhat. Accelerated mesh sampling for the hyper reduction of nonlinear computational models. *International Journal for Numerical Methods in Engineering*, 109(12):1623–1654, 3 2017.
- [23] S. Chaturantabut and D. C. Sorensen. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 1 2010.
- [24] A. Cuitino and M. Ortiz. A material-independent method for extending stress update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics. *Engineering computations*, 1992.
- [25] N. Demo, M. Tezzele, G. Gustin, G. Lavini, and G. Rozza. Shape optimization by means of proper orthogonal decomposition and dynamic mode decomposition. *Technology and Science for the Ships of the Future - Proceedings of NAV 2018: 19th International Conference on Ship and Maritime Research*, pages 212–219, 2018.
- [26] P. Diercks, K. Veroy, A. Robens-Radermacher, and J. F. Unger. Multiscale modeling of linear elastic heterogeneous structures via localized model order reduction. *International Journal for Numerical Methods in Engineering*, 7 2023.
- [27] G. J. Dvorak. Transformation field analysis of inelastic composite materials. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 437(1900):311–327, 5 1992.
- [28] R. Eggersmann, T. Kirchdoerfer, S. Reese, L. Stainier, and M. Ortiz. Model-Free Data-Driven inelasticity. *Computer Methods in Applied Mechanics and Engineering*, 350:81–99, 2019.
- [29] J. D. Eshelby. The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proceedings of the royal society of London. Series A. Mathematical and physical sciences*, 241(1226):376–396, 1957.

- [30] C. Farhat, P. Avery, T. Chapman, and J. Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662, 6 2014.
- [31] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):231–245, 9 1998.
- [32] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method - part I: A faster alternative to the two-level FETI method. *International Journal for Numerical Methods in Engineering*, 50(7):1523–1544, 3 2001.
- [33] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 10 1991.
- [34] F. Feyel. Multiscale FE2 elastoviscoplastic analysis of composite structures. *Computational Materials Science*, 16(1-4):344–354, 12 1999.
- [35] B. Florijn, C. Coulais, and M. Van Hecke. Programmable mechanical metamaterials. *Physical Review Letters*, 113(17), 10 2014.
- [36] S. Forest. Homogenization methods and mechanics of generalized continua - part 2. *Theoretical and Applied Mechanics*, (28-29):113–144, 2002.
- [37] M. G. D. Geers, V. G. Kouznetsova, and W. A. M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics*, 234(7):2175–2182, 8 2010.
- [38] J. Ghaboussi, J. Garrett Jr, and X. Wu. Knowledge-based modeling of material behavior with neural networks. *Journal of engineering mechanics*, 117:132–153, 1991.
- [39] J. Ghaboussi, D. A. Pecknold, M. Zhang, and R. M. Haj-Ali. Autoprogressive training of neural network constitutive models. *International Journal for Numerical Methods in Engineering*, 42(1):105–126, 1998.
- [40] F. Ghavamian and A. Simone. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. *Computer Methods in Applied Mechanics and Engineering*, 357:112594, 2019.



- [41] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
- [42] GPy. GPy: A Gaussian process framework in python, 2012.
- [43] A. M. Greiner, B. Richter, and M. Bastmeyer. Micro-Engineered 3D Scaffolds for Cell Culture Studies. *Macromolecular Bioscience*, 12(10):1301–1314, 2012.
- [44] M. Guénot, I. Lepot, C. Sainvitu, J. Goblet, and R. F. Coelho. Adaptive sampling strategies for non-intrusive POD-based surrogates. *Engineering Computations (Swansea, Wales)*, 30(4):521–547, 2013.
- [45] M. Guo and J. S. Hesthaven. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Computer Methods in Applied Mechanics and Engineering*, 341:807–826, 11 2018.
- [46] T. Guo, O. Rokoš, and K. Veroy. Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 384:113924, 10 2021.
- [47] 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*, 1963.
- [48] J. Hernández, M. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer Methods in Applied Mechanics and Engineering*, 313:687–722, 1 2017.
- [49] J. Hernández, J. Oliver, A. Huespe, M. Caicedo, and J. Cante. High-performance model reduction techniques in computational multiscale homogenization. *Computer Methods in Applied Mechanics and Engineering*, 276:149–189, 7 2014.
- [50] J. A. Hernández. A multiscale method for periodic structures using domain decomposition and ECM-hyperreduction. *Computer Methods in Applied Mechanics and Engineering*, 368:113192, 2020.
- [51] J. A. Hernández, A. Giuliadori, and E. Soudah. Empirical Interscale Finite Element Method (EIFEM) for modeling heterogeneous structures via localized hyperreduction. *Computer Methods in Applied Mechanics and Engineering*, 418, 1 2024.

- [52] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Springer International Publishing, Cham, 2016.
- [53] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.
- [54] R. Hill. On constitutive macro-variables for heterogeneous solids at finite strain. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 326(1565):131–147, 1972.
- [55] T. Y. Hou and X.-H. Wu. A Multiscale Finite Element Method for Elliptic Problems in Composite Materials and Porous Media. *Journal of Computational Physics*, 134(1):169–189, 6 1997.
- [56] D. Huang, J. N. Fuhg, C. Weißenfels, and P. Wriggers. A machine learning based plasticity model using proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 365:113008, 2020.
- [57] D. Z. Huang, K. Xu, C. Farhat, and E. Darve. Learning constitutive relations from indirect observations using deep neural networks. *Journal of Computational Physics*, 416:109491, 9 2020.
- [58] R. Jänicke, S. Diebels, H. G. Sehlhorst, and A. Düster. Two-scale modelling of micromorphic continua : A numerical homogenization scheme. *Continuum Mechanics and Thermodynamics*, 21(4):297–315, 9 2009.
- [59] H. Jasak and Z. Tuković. Automatic mesh motion for the unstructured Finite Volume Method. *Transactions of Famena*, 30(2):1–20, 2006.
- [60] R. M. Jones. *Mechanics of composite materials*. CRC press, 1998.
- [61] Ł. Kaczmarczyk, C. J. Pearce, and N. Bićanić. Scale transition and enforcement of RVE boundary conditions in second-order computational homogenization. *International Journal for Numerical Methods in Engineering*, 74(3):506–522, 2008.
- [62] T. Kanit, S. Forest, I. Galliet, V. Mounoury, and D. Jeulin. Determination of the size of the representative volume element for random composites: statistical and numerical approach. *International Journal of Solids and Structures*, 40(13):3647–3679, 2003.
- [63] K. Karapiperis, L. Stainier, M. Ortiz, and J. E. Andrade. Data-Driven multiscale modeling in mechanics. *Journal of the Mechanics and Physics of Solids*, 147(July 2020):104239, 2021.

- [64] M. Kast, M. Guo, and J. S. Hesthaven. A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 364:112947, 6 2020.
- [65] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 12 2014.
- [66] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, (MI):1–14, 2014.
- [67] T. Kirchdoerfer and M. Ortiz. Data-driven computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 304:81–101, 2016.
- [68] T. Kirchdoerfer and M. Ortiz. Data Driven Computing with noisy material data sets. *Computer Methods in Applied Mechanics and Engineering*, 326:622–641, 2017.
- [69] V. Kouznetsova, M. 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*, 193(48-51):5525–5550, 12 2004.
- [70] V. G. Kouznetsova. Computational homogenization for the multi-scale analysis of multi-phase materials. 2004.
- [71] V. G. Kouznetsova, W. A. M. Brekelmans, and F. P. T. Baaijens. An approach to micro-macro modeling of heterogeneous materials. *Computational Mechanics*, 27(1):37–48, 1 2001.
- [72] A. O. Krushynska, V. G. Kouznetsova, and M. G. Geers. Towards optimal design of locally resonant acoustic metamaterials. *Journal of the Mechanics and Physics of Solids*, 71(1):179–196, 2014.
- [73] B. A. Le, J. Yvonnet, and Q. C. He. Computational homogenization of nonlinear elastic materials using neural networks. *International Journal for Numerical Methods in Engineering*, 104(12):1061–1084, 12 2015.
- [74] T. Lesičar, Z. Tonković, and J. Sorić. A second-order two-scale homogenization procedure using C1 macrolevel discretization. *Computational Mechanics*, 54(2):425–441, 2014.

- [75] K. Linka, M. Hillgärtner, K. P. Abdolazizi, R. C. Aydin, M. Itskov, and C. J. Cyron. Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning. *Journal of Computational Physics*, 429:110010, 2021.
- [76] H. Liu, Y. S. Ong, and J. Cai. A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design. *Structural and Multidisciplinary Optimization*, 57(1):393–416, 2018.
- [77] Z. Liu, M. A. Bessa, and W. K. Liu. Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 306:319–341, 2016.
- [78] H. J. Logarzo, G. Capuano, and J. J. Rimoli. Smart constitutive laws: Inelastic homogenization through machine learning. *Computer Methods in Applied Mechanics and Engineering*, 373:113482, 2021.
- [79] D. J. Luscher, D. L. McDowell, and C. A. Bronkhorst. A second gradient theoretical framework for hierarchical multiscale modeling of materials. *International Journal of Plasticity*, 26(8):1248–1275, 8 2010.
- [80] Y. Maday and E. M. Rønquist. A Reduced-Basis Element Method. *Journal of Scientific Computing*, 17(1):447–459, 12 2002.
- [81] A. Malqvist and D. Peterseim. Localization of Elliptic Multiscale Problems. 10 2011.
- [82] P. D. Mangalgi. Composite materials for aerospace applications. *Bulletin of Materials Science*, 22(3):657–664, 1999.
- [83] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *International Journal for Numerical Methods in Biomedical Engineering*, 28(6-7):604–625, 6 2012.
- [84] F. Masi, I. Stefanou, P. Vannucci, and V. Maffi-Berthier. Thermodynamics-based Artificial Neural Networks for constitutive modeling. *Journal of the Mechanics and Physics of Solids*, 147(October 2020):104277, 2 2021.
- [85] K. Matouš, M. G. D. Geers, V. G. Kouznetsova, and A. Gillman. A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *Journal of Computational Physics*, 330:192–220, 2 2017.

- [86] J. C. Michel, H. Moulinec, and P. Suquet. Effective properties of composite materials with periodic microstructure: a computational approach. *Computer Methods in Applied Mechanics and Engineering*, 172(1):109–143, 1999.
- [87] J. C. Michel and P. Suquet. Nonuniform transformation field analysis. *International Journal of Solids and Structures*, 40(25):6937–6955, 12 2003.
- [88] C. Miehe and A. Koch. Computational micro-to-macro transitions of discretized microstructures undergoing small strains. *Archive of Applied Mechanics (Ingenieur Archiv)*, 72(4-5):300–317, 7 2002.
- [89] C. Miehe, J. Schröder, and J. Schotte. Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials. *Computer methods in applied mechanics and engineering*, 171(3-4):387–418, 1999.
- [90] R. Mindlin and N. Eshel. On first strain-gradient theories in linear elasticity. *International Journal of Solids and Structures*, 4(1):109–124, 1 1968.
- [91] N. Mishra, J. Vondřejc, and J. Zeman. A comparative study on low-memory iterative solvers for FFT-based homogenization of periodic media. *Journal of Computational Physics*, 321:151–168, 9 2016.
- [92] T. Mori and K. Tanaka. Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta metallurgica*, 21(5):571–574, 1973.
- [93] 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*, 157(1-2):69–94, 4 1998.
- [94] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, and M. A. Bessa. Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences*, 116(52):26414–26420, 12 2019.
- [95] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.
- [96] X. Ning, X. Yu, H. Wang, R. Sun, R. E. Corman, H. Li, C. M. Lee, Y. Xue, A. Chempakasseril, Y. Yao, and others. Mechanically active materials in three-dimensional mesostructures. *Science advances*, 4(9):eaat8313, 2018.

- [97] W. Noll, B. D. Coleman, and W. Noll. The Thermodynamics of Elastic Materials with Heat Conduction and Viscosity. *The Foundations of Mechanics and Thermodynamics*, pages 145–156, 1974.
- [98] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [99] C. J. Permann, D. R. Gaston, D. Andrš, R. W. Carlsen, F. Kong, A. D. Lindsay, J. M. Miller, J. W. Peterson, A. E. Slaughter, R. H. Stogner, and R. C. Martineau. MOOSE: Enabling massively parallel multiphysics simulation. *SoftwareX*, 11:100430, 1 2020.
- [100] D. B. Phuong Huynh, D. J. Knezevic, and A. T. Patera. A Static condensation Reduced Basis Element method : Approximation and a posteriori error estimation. *Mathematical Modelling and Numerical Analysis*, 47(1):213–251, 2013.
- [101] M. J. D. Powell. Radial Basis Functions for Multivariable Interpolation: A Review. In *Algorithms for Approximation*, pages 143–167. Clarendon Press, USA, 1987.
- [102] D. Puglia, J. Biagiotti, and J. M. Kenny. A Review on Natural Fibre-Based Composites—Part II. *Journal of Natural Fibers*, 1(3):23–65, 4 2005.
- [103] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations*, volume 92 of *UNITEXT*. Springer International Publishing, Cham, 2016.
- [104] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, 1999.
- [105] J. Quiñonero-Candela and C. E. Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. *J. Mach. Learn. Res.*, 6:1939–1959, 12 2005.
- [106] A. Radermacher, B. A. Bednarczyk, B. Stier, J. Simon, L. Zhou, and S. Reese. Displacement-based multiscale modeling of fiber-reinforced

- composites by means of proper orthogonal decomposition. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):29, 12 2016.
- [107] A. Radermacher and S. Reese. POD-based model reduction with empirical interpolation applied to nonlinear elasticity. *International Journal for Numerical Methods in Engineering*, 107(6):477–495, 8 2016.
- [108] S. A. Ramakrishna. Physics of negative refractive index materials. *Reports on Progress in Physics*, 68(2):449–521, 2 2005.
- [109] M. Raschi, O. Lloberas-Valls, A. Huespe, and J. Oliver. High performance reduction technique for multiscale finite element modeling (HPR-FE2): Towards industrial multiscale FE software. *Computer Methods in Applied Mechanics and Engineering*, 375:113580, 2021.
- [110] C. E. Rasmussen. Gaussian Processes in Machine Learning. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [111] B. D. Reddy and D. van Huyssteen. A virtual element method for transversely isotropic elasticity. *Computational Mechanics*, 64(4):971–988, 10 2019.
- [112] A. Reuß. Berechnung der fließgrenze von mischkristallen auf grund der plastizitätsbedingung für einkristalle. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):49–58, 1929.
- [113] I. A. Rodrigues Lopes and F. M. Andrade Pires. An assessment of multi-scale models based on second-order computational homogenisation. *Computers and Structures*, 259, 1 2022.
- [114] O. Rokoš, M. M. Ameen, R. H. J. Peerlings, and M. G. D. Geers. Micromorphic computational homogenization for mechanical metamaterials with patterning fluctuation fields. *Journal of the Mechanics and Physics of Solids*, 123:119–137, 2019.
- [115] G. Rozza, M. Hess, G. Stabile, M. Tezzele, and F. Ballarin. 1 Basic ideas and tools for projection-based model reduction of parametric partial differential equations. In *Snapshot-Based Methods and Algorithms*, pages 1–47. De Gruyter, 12 2020.

- [116] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 9 2008.
- [117] D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77(1):75–89, 1 2009.
- [118] G. Sachs. Zur Ableitung einer Fließbedingung. *Z. Ver, Dtsch. Ing.*, 72:734–736, 1928.
- [119] 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*, 68(5), 2016.
- [120] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986*, 20(4):151–160, 1986.
- [121] J. C. Simo and T. J. R. Hughes. *Computational inelasticity*, volume 7. Springer Science & Business Media, 2006.
- [122] K. Smetana and A. T. Patera. Optimal local approximation spaces for component-based static condensation procedures. *SIAM Journal on Scientific Computing*, 38(5):A3318–A3356, 2016.
- [123] I. M. Sobol’. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.
- [124] D. Soldner, B. Brands, R. Zabihyan, P. Steinmann, and J. Mergheim. A numerical study of different projection-based model reduction techniques applied to computational homogenisation. *Computational Mechanics*, 60(4):613–625, 10 2017.
- [125] S. O. Sperling, T. Guo, R. H. J. Peerlings, V. G. Kouznetsova, M. G. D. Geers, and O. Rokoš. A comparative study of enriched computational homogenization schemes applied to two-dimensional pattern-transforming elastomeric mechanical metamaterials. *Computational Mechanics*, 1 2024.
- [126] G. Stabile, M. Zancanaro, and G. Rozza. Efficient geometrical parametrization for finite-volume-based reduced order methods. *International Journal for Numerical Methods in Engineering*, 121(12):2655–2682, 6 2020.



- [127] P. Suquet. Elements of homogenization for inelastic solid mechanics. *Homogenization techniques for composite media*, 1987.
- [128] 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*, 21(3):1800864, 2019.
- [129] R. Swischuk, L. Mainini, B. Peherstorfer, and K. Willcox. Projection-based model reduction: Formulations for physics-based machine learning. *Computers & Fluids*, 179:704–717, 1 2019.
- [130] T. Taddei. A Registration Method for Model Order Reduction: Data Compression and Geometry Reduction. *SIAM Journal on Scientific Computing*, 42(2):A997–A1027, 1 2020.
- [131] K. T. Tan, H. H. Huang, and C. T. Sun. Blast-wave impact mitigation using negative effective mass density concept of elastic metamaterials. *International Journal of Impact Engineering*, 64:20–29, 2014.
- [132] G. I. Taylor. Plastic strain in metals. *J. Inst. Metals*, 62:307–324, 1938.
- [133] K. Terada, M. Hori, T. Kyoya, and N. Kikuchi. Simulation of the multi-scale convergence in computational homogenization approaches. *International Journal of Solids and Structures*, 37(16):2285–2311, 2000.
- [134] T. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces—The deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339–351, 2 1992.
- [135] A. Toselli and O. B. Widlund. *Domain Decomposition Methods — Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [136] R. A. Toupin. Elastic materials with couple-stresses. *Archive for Rational Mechanics and Analysis*, 11(1):385–414, 1962.
- [137] W. Voigt. Ueber die Beziehung zwischen den beiden Elasticitätsconstanten isotroper Körper. *Annalen der physik*, 274(12):573–587, 1889.
- [138] L. Wang, J. Boddapati, K. Liu, P. Zhu, C. Daraio, and W. Chen. Mechanical cloak via data-driven aperiodic metamaterial design. *Proceedings of the National Academy of Sciences*, 119(13), 3 2022.

- [139] K. Washabaugh, M. J. Zahr, and C. Farhat. On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries. *54th AIAA Aerospace Sciences Meeting*, 0(January):1–16, 2016.
- [140] L. Wu, M. Mustafa, J. Segurado, and L. Noels. Second-order computational homogenisation enhanced with non-uniform body forces for non-linear cellular materials and metamaterials. *Computer Methods in Applied Mechanics and Engineering*, 407:115931, 3 2023.
- [141] L. Wu, V. D. Nguyen, N. G. Kilingar, and L. Noels. A recurrent neural network-accelerated multi-scale model for elasto-plastic heterogeneous materials subjected to random cyclic and non-proportional loading paths. *Computer Methods in Applied Mechanics and Engineering*, 369:113234, 9 2020.
- [142] P. Wu, J. Sun, X. Chang, W. Zhang, R. Arcucci, Y. Guo, and C. C. Pain. Data-driven reduced order model with temporal convolutional neural network. *Computer Methods in Applied Mechanics and Engineering*, 360:112766, 3 2020.
- [143] K. Xu, D. Z. Huang, and E. Darve. Learning constitutive relations using symmetric positive definite neural networks. *Journal of Computational Physics*, 428:110072, 3 2021.
- [144] M. Yang and Z. Xiao. POD-based surrogate modeling of transitional flows using an adaptive sampling in Gaussian process. *International Journal of Heat and Fluid Flow*, 84(December 2019):108596, 2020.
- [145] M. Yano and A. T. Patera. An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs. *Computer Methods in Applied Mechanics and Engineering*, 344:1104–1123, 2 2019.
- [146] T.-M. Yao and K. K. Choi. 3-D shape optimal design and automatic finite element regriding. *International Journal for Numerical Methods in Engineering*, 28(2):369–384, 2 1989.
- [147] C. Yu, O. L. Kafka, and W. K. Liu. Self-consistent clustering analysis for multiscale modeling at finite strains. *Computer Methods in Applied Mechanics and Engineering*, 349:339–359, 6 2019.
- [148] J. Yvonnet, N. Auffray, and V. Monchiet. Computational second-order homogenization of materials with effective anisotropic strain-gradient behavior. *International Journal of Solids and Structures*, 191-192:434–448, 5 2020.

- 
- [149] 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*, 223(1):341–368, 4 2007.
- [150] G. Zhang, N. Feng, and K. Khandelwal. A computational framework for homogenization and multiscale stability analyses of nonlinear periodic materials. *International Journal for Numerical Methods in Engineering*, 122(22):6527–6575, 11 2021.
- [151] Z. Zhang and A. O. Krushynska. Programmable shape-morphing of rose-shaped mechanical metamaterials. *APL Materials*, 10(8):080701, 8 2022.
- [152] 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, S. O. Kucheyev, N. X. Fang, and C. M. Spadaccini. Ultralight, ultrastiff mechanical metamaterials. *Science*, 344(6190):1373–1377, 2014.



# Summary

## **Model order reduction in two-scale solid mechanics**

To study the non-linear effect of microstructure onto macroscopic properties, scale-bridging methods such as computational homogenization are frequently employed in which the micro- and macrostructure are separately modelled, discretized, and solved concurrently in a coupled manner. Although homogenization techniques bring substantial computational savings in engineering applications, the solution of two-scale systems is still computationally expensive, and even infeasible in multi-query contexts, such as optimization, material design, etc.

To render these analyses tractable, it is necessary to construct surrogate models that can effectively and accurately capture the microscopic behavior over a large parameter and design space of shape, material, and loading parameters. To address this, this dissertation investigates two distinct model order reduction techniques. Within the context of first-order computational homogenization, the first method combines the proper orthogonal decomposition with Gaussian process regression. The method is successfully applied to a two-scale benchmark problem with hyperelastic material behavior of microstructures under varying conditions, such as stiffness and shape. The second method utilizes POD in conjunction with the empirical cubature method and is applied to two-scale examples, which are motivated by mechanical metamaterials and involve elasto-plastic microstructures under large deformations. Additionally, the shape of such microstructures can vary throughout the macroscopic domain, leading to qualitative different effective properties, such as, e.g., positive or negative Poisson's ratios. Finally, extensions towards second-order computational homogenization are considered, utilizing the latter of the two aforementioned methods. It is shown that the developed methodology is capable of handling the high-dimensional parameter space, and further captures the multi-scale buckling phenomena for which the first-order formulation fails. For all considered numerical examples, all methods yield high accuracies with high speed-ups. The advantages and disadvantages are carefully assessed and critically evaluated.



# List of Publications

## Journal articles

4. Sperling, S.O., Guo, T., Peerling, R.H.J., Kouznetsova, V.G., Geers, M.G.D., & Rokoš, O. (2024). A comparative study of enriched computational homogenization schemes applied to two-dimensional pattern-transforming elastomeric mechanical metamaterials.. *Computational Mechanics* 72(6).
3. Guo, T., Rokoš, O., & Veroy, K. (2024). A reduced order model for geometrically parameterized two-scale simulations of elasto-plastic microstructures under large deformations. *Computer Methods in Applied Mechanics and Engineering*, 418, 116467.
2. Guo, T., Silva, F.A.B., Rokoš, O., & Veroy, K. (2022). Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method: Extension to geometrical parameterizations. *Computer Methods in Applied Mechanics and Engineering*, 401, 115636.
1. Guo, T., Rokoš, O., & Veroy, K. (2021). Learning constitutive models from microstructural simulations via a non-intrusive reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 384, 113924.

## In preparation

5. Guo, T., Kouznetsova, V.G., Geers, M.G.D., Veroy, K., & Rokoš, O. (n.a.). Reduced order modeling for second-order computational homogenization with applications to geometrically parameterized elastomeric metamaterials. In preparation.





# Curriculum Vitae

Theron Guo was born on 21st June 1995, in Berlin, Germany. He obtained his Bachelor's and Master's degree in Computational Engineering Science at RWTH Aachen University in Aachen, Germany, in November 2019. In February 2020, he moved to Eindhoven University of Technology, where he joined Karen Veroy's group as a PhD student under the EU Horizon 2020 project "Real-time Data-Informed Multi-scale Computational Methods for Material Design and Processing" (Grant agreement No. 818473).



# Acknowledgments

I would like to express my gratitude to Karen Veroy and Ondřej Rokoš for their supervision and guidance throughout the last couple of years. I am particularly grateful for the freedom in exploring different research directions, and for the numerous opportunities to go to conferences and workshops to meet and network with other researchers. I am also thankful for the collaboration and discussions with Marc Geers and Varvara Kouznetsova.

Moreover, I would like to thank my colleagues at CASA for creating a pleasant working atmosphere with several events throughout the year. I am also grateful for the many interesting conversations with my peers in the Computational Science group, in particular with my office mates, which often helped me gain new insights and generate new ideas.

My sincere thanks goes out to my family and friends. Lastly, I wish to thank Helen for her support, encouragement and love.