University of Cambridge

Department of Engineering

Multiscale modelling of woven and knitted fabric membranes

This thesis is submitted for the degree of Doctor of Philosophy

by

Sumudu Tharanga Herath Herath Mudiyanselage

Trinity Hall

February 2020

Declaration

I wish to declare that, except for commonly understood ideas and concepts, or where specific reference is made to the work of other authors, the contents of this dissertation are original and do not include subject matter that is the outcome of work done in collaboration with others. The research described in this dissertation was conducted in the Department of Engineering at the University of Cambridge from October 2016 to December 2019. This dissertation has not previously been submitted, in part or in whole, to any other university or institution for any degree, diploma, or other qualification. This dissertation is presented in 213 pages, and contains 91 figures and approximately 58,000 words, including references and appendices.

Sumudu Herath Mudiyanselage

Multiscale modelling of woven and knitted fabric membranes

Sumudu Herath Mudiyanselage

Abstract

Light-weight fabric membranes have gained increasing popularity over the past years due to their tailorable structural and material performances. These tailorable properties include stretch forming and deep drawing formability that exhibits excellent stretchability and drapeability properties of textiles and textile composites. Since the inception of computerised numerical control for three-dimensional textile-manufacturing machines, technical textiles paved their way to numerous applications, certainly not limited to; aerospace, biomedical, civil engineering, defence, marine and medical industries. Digital interlooping and digital interlacing technology in additive manufacturing greatly advanced the manufacturing processes of textiles. In this work, we consider two branches of technical fabrics, namely plain-woven and weft-knitted.

Multiscale modelling is the tool of choice for homogenising periodic structures and has been used extensively to model and analyse the mechanical behaviour of woven and knitted fabrics. But there is a plethora of literature discussing the demerits of such conventional multiscale modelling. These demerits include higher computational costs, rigid numerical models, inefficient algorithmic computations and inability to incorporate geometric nonlinearities. We propose a data-driven nonlinear multiscale modelling technique to analyse the complex mechanical behaviour of plain-woven and weft-knitted fabrics with a neat extension to fabric material designing. We show how the integration of statistical learning techniques mitigates the weaknesses of conventional multiscale

modelling. Moreover, we discuss the avenues that will open in many potential fields with regard to material modelling, structural engineering and textile industries.

In the proposed data-driven nonlinear computational homogenisation technique, we efficiently integrate the microscale and macroscale using Gaussian Process Regression (GPR) statistical learning technique. In the microscale, representative volume elements (RVEs) are modelled using finite deformable isogeometric spatial rods and deformation is homogenised using periodic boundary conditions. This finite deformable rod is proficient in handling large deformations, rod-to-rod contacts, arbitrary cross-section definitions and follower loads. Respecting the principle of separation of scales, we construct response databases by applying different homogenised strain states to the RVEs and recording the respective incremental volume-averaged energy values. We use GPR to learn a model using a 5-fold cross-validation technique by optimising the log marginal likelihood. In the macroscale, textiles are modelled as nonlinear orthotropic membranes for which the stresses and material constitutive relations are predicted by the trained GPR model. This coupling between GPR and membrane models is achieved through a systematic and seamless finite element integration using C++ and Python environments. A neat extension to material designing is also discussed with potentials to extend the work into other related fields.

Acknowledgements

There has been a number of people who have been a constant source of encouragement throughout my PhD. Although it will be impossible to list everyone, several of them deserve special acknowledgement. Firstly, I am grateful to my supervisor, Dr Fehmi Cirak, being the source of inspiration and motivation. I am immensely thankful for his support while I was learning the fundamentals of this research field. I would like to thank my advisor Dr Keith Seffen, for helpful discussions and feedback, particularly during the early phase of my research. I also thank my graduate tutor Dr Ramji Venkataramanan for his valuable advices and guidance to work towards my doctorate degree.

It was an absolute joy sharing the lab space with Dr Xiao Xiao, Dr Qiaoling Zhang, Dr Ge Yin, Dr Arion Pons, Dr Eky Febrianto, Kim Jie, Adeeb Arif Kor, Arnaud Vadeboncoeur, Tianyi Liu and others; who were always keen to share their ideas, and willing to give critical and constructive feedback on my work. Special thanks should be given to Dr Malcom and Dr Kosala Bandara for their valuable inputs to my research work. I am extremely grateful for the financial support provided by the Cambridge Trust, Trinity Hall and Postgraduate hardship funding. Without their financial support, it would have been impossible to undertake this research. Lastly, I would like to thank my friends, Ridwan Shariffdeen, Dulan Nakandala, Pasansith Gamalath, Biyanka Ekanayake, Kasun Kariyawasam, Gemunu Kulasekara, Mahendrini Fernando, Kalpanee Jayathilake, Amila Jayasinghe, Mierunalan Seyon, Tisal Edirisinghe, Hansini Mallikarachchi, Hasini Weerasinghe, Thamasha Samarasinghe, Tim Watson, Paul Sabota, Giacomo Torreli and others for their constant support while I discussed my trials and tribulations.

Above all, although this alone is not sufficient, I would like to thank my family for their endless patience and understanding for the past four years. I dedicate this thesis to my beloved sister.

Contents

List of Figures

List of Tables

Nomenclature

This section provides a concise reference describing the abbreviations and notations used throughout this thesis. However, additional abbreviations and notations are also provided in some chapters where they are used only within the respective chapters.

Abbreviations

BVP	Boundary Value Problem
CAD	Computer Aided Design
CNC	Computer Numerical Control
FE/FEM/FEA	Finite Element/ Finite Element Method/ Finite Element Analysis
GPR	Gaussian Process Regression
MLML	Maximum Log Marginal Likelihood
MSE	Mean Squared Error
RVE	Representative Volume Element

Linear algebra operations

$\boldsymbol{p} \cdot \boldsymbol{q}$	standard vector dot(inner) product
$\boldsymbol{p} \times \boldsymbol{q}$	standard vector cross product
$\boldsymbol{p} \otimes \boldsymbol{q}$	standard vector tensor (outer) product
$\boldsymbol{p} \odot \boldsymbol{q}$	standard vector Hadamard (Schur) product
$\|\|\boldsymbol{p}\|\|$	standard vector norm (L_2 norm)
$\mathbf{I}$	3 by 3 Identity matrix
$(\bullet)^{\mathrm{T}}$	transpose of a matrix
$(\bullet) : (\bullet)$	Frobenius inner product between two tensors
$\det(\bullet)$	determinant of a matrix
$\mathrm{Tr}(\bullet)$	trace of a square matrix

Continuum mechanics

$\boldsymbol{F}$	Deformation gradient
$\boldsymbol{E}$	Green-Lagrange strain tensor
J	Jacobian
$\boldsymbol{P}$	First Piola-Kirchhoff stress tensor
$\boldsymbol{S}$	Second Piola-Kirchhoff stress tensor
$\boldsymbol{\mathcal{C}}$	Fourth-order material tensor

1 Introduction

In the appreciation of this research, Chapter 1 begins with the motivations behind the work by highlighting the research gaps and potential applications. Next, the latest developments in textile design, manufacturing technology and data analytics are discussed to provide a background to this research work. Contributions made to different research communities will be discussed in Section 1.3, emphasising on the reduction of the identified research gaps. The chapter concludes by presenting a concise layout of the successive chapters of this thesis.

1.1 Motivation

There is a substantial need for having models that integrate the analysis and design of technical textiles. With recent advances in manufacturing, such as additive manufacturing, high-throughput computing (HTC) and high-performance computing (HPC), the demand for integrated analysis and design of fabrics has never been greater [1–3]. Various applications in aerospace, bio-medical, civil engineering, defence, marine and medical industries largely attract technical textiles for their tailorable structural, mechanical and material performances [4–8]. Hence, from a technical point of view, textiles have advanced far beyond the traditional use of clothing. Therefore, the main objective of this research is to develop a unified framework for the computational homogenisation and design of knitted and woven textile membranes. This unified framework is expected to upgrade the existing analysis and design tools to cater to the booming additive manufacturing technologies.

Formally, a textile is defined as a flexible material consisting of a network of yarns arranged systematically using yarn interlacing or interlooping. Figure 1.1 illustrates a few among many applications of technical textiles. Starting from parachutes [9], top row in

Figure 1.1 depicts other uses of textiles such as, fabric formworks [10], smart clothing [11] implantable medical devices [12]. Bottom row in Figure 1.1 continues to illustrate textile applications in cardiac supports [13], dry suits that remains watertight up to 170 diving hours [14], covert body armors [15] and interior architecture designs [16]. Evolution of the use of fabrics from clothing to advanced applications demands simultaneous advancement in mechanical analysis and design tools. Prevailing textile analysis methods, such as yarn-level modelling [2, 17], multiscale modelling [1, 18–20] and material parameter estimations [21], fail to provide a unified approach to the mechanical analysis and design of textiles. Here, a discussion is supplemented to elaborate on the weaknesses of the existing textile simulators.

Yarn-level modelling is popular in the computer graphics community, where kinematically simplified models are used to perform simulations on textile membranes. Also, yarn-level modelling involves degrees of freedoms in the orders of millions (or even more) and thus inherits a significant computational cost [2]. In reality, however, there is no strict rule to model each and every yarn in the textile, unless one needs to simulate localised deformations such as, snagging, yarn pull-outs, yarn unravelling and yarn fraying. Furthermore, conventional multiscale methods and material parametrisation methods are found to be less computationally expensive compared to yarn-level modelling. However, in conventional multiscale methods, the computational cost increases rapidly when the problem domain is large [39]. This inefficiency is inflated when the macroscale boundary value problem involves material and geometric nonlinearities. A common weakness of the existing modelling techniques is their insensitivity to the changes of the problem domain, material and geometric properties. Due to the inherent computational cost of these existing methods, it is not practically feasible to re-run simulations to investigate for such sensitivities.

The development of unified technical tools, to analyse and design textiles, is found slow-paced and inconsistent compared with the rapid advancements of manufacturing technologies and data analytics. Modern-day technological boost in subtractive and additive manufacturing revolutionises various engineering disciplines, whereas data-driven analytics continue to dominate in a multitude of decision-making processes. Therefore, considering the aforementioned weaknesses of the existing modelling techniques and the recent advances in data-driven techniques, the need arises to develop efficient and unified technical tools to analyse and design technical textiles. This research gap sets the motivation to address these issues by developing data-driven analysis and design tools

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1.1: Diverse applications of technical textiles beyond the conventional use of clothing. (a) Parachutes [9] (b) Fabric formworks [10] (c) Smart clothing [11] (d) Implantable medical devices [12] (e) Cardiac supports [13] (f) Dry suits [14] (g) Covert body armours [15] (h) Interior architecture designs [16]

for knitted and woven technical textiles.

1.2 Background

After identifying the research gaps and motivations behind the forthcoming chapters of this thesis, a brief discussion on the background of this work is presented here. First, conventional analysis and design of textiles are discussed to familiarise the reader with the recent past and present methods to simulate textiles materials. Secondly, a brief discussion furnishes the subtractive and additive manufacturing methods to emphasise the importance of this work in booming technologies. Lastly, current trends in data-analytics are discussed, as our proposed methods rely heavily on data-centric techniques.

Conventional analysis and design of technical textiles

Mechanical analysis of textiles, like any other material, stems from various experimental measures. For textiles, these experimental measures are application dependent and would range from a simple laboratory tensometer to a wind-tunnel testing facility. The complex mechanical behaviour of textiles was a bottleneck for researchers in devising numerical methods to approximate different aspects of textile kinematics. A few of these challenges would include yarn contact modelling, friction modelling, compressive response, initial yarn curvatures and anisotropic material constitutive models. However, through systematic idealisations, researchers paved their way to model the complex textile behaviour numerically.

Noteworthy research, in the field of mechanical behaviour of woven textiles, can be found in the early works of Behre [22], Grosberg et al. [23], Hearle et al. [24], Hearle et al. [25], Ly [26] and Postle et al. [27]. Since then, the tensile behaviour of woven fabrics has been investigated by many researchers using various numerical approaches. Few of these methods include pin-jointed truss models, energy minimisation, multigrid method, laminate theory and curve fitting [28]. Jevsnik et al. [28] provide a rigorous review on such computer models for fabric simulations, where they separately identify geometrical, physical and hybrid models as three main variants of fabric simulations.

The complexity of cloth simulators increased from simple geometrical models to advanced higher-order multiscale and yarn-level models to cater to the booming nano-to-micro scale fibre-based fabric technologies [2, 21]. Yarn-level fabric modelling techniques are out of the scope of this thesis simply due to their extremely expensive computations. Also, unless there is a need to simulate local deformations like snagging, there is no strict requirement to simulate textiles at their yarn-level. Therefore, we resort to using multiscale modelling techniques in the scope of this research work. A detailed discussion on conventional computational homogenisation is provided later in Chapter 5.

Subtractive and additive manufacturing

The inception of subtractive manufacturing, or commonly referred to as Computer Numerical Control (CNC), dates back to the mid 20^{th} century. Collaborative work by Parsons Corporation and Massachusetts Institute of Technology (MIT) achieved significant developments in CNC and eventually released to the industry in 1955 [29]. With

CNC, each object to be manufactured receives a custom computer program, usually written in an international standard language called G-code, stored in and executed by the Machine Control Unit (MCU).

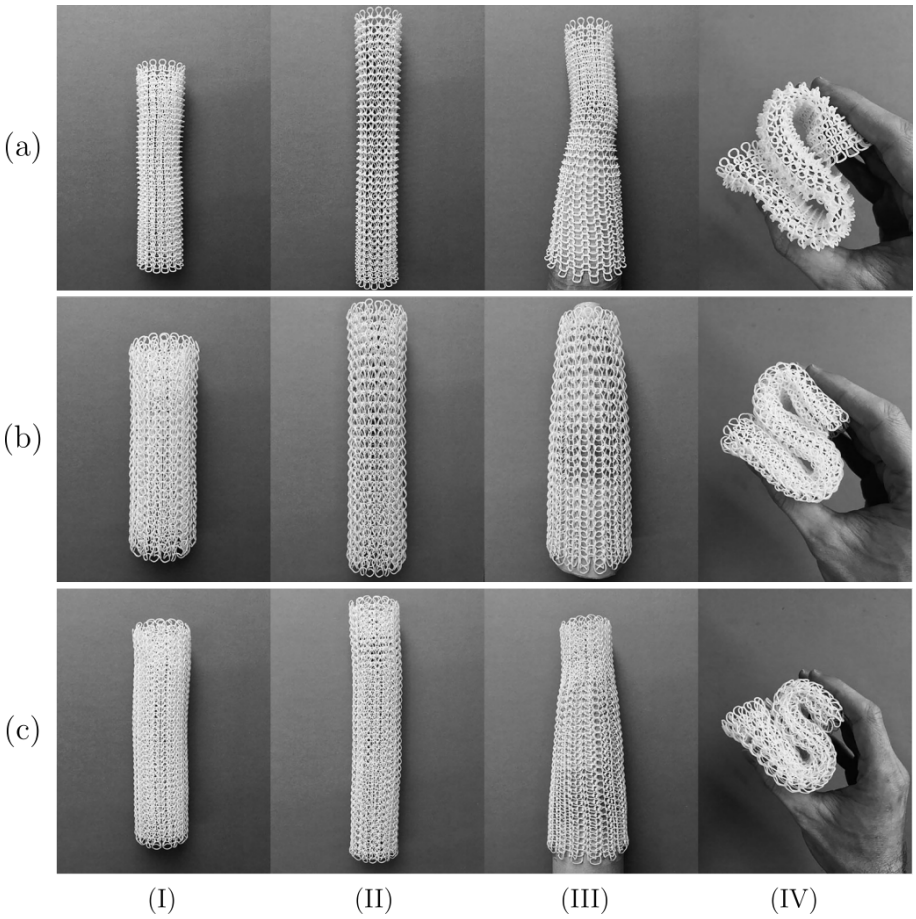


Figure 1.2: Different deformed states of three (a), (b), (c) digitally interlooped tubular forms of knitted textiles. Deformation states are (I) compression (II) extension (III) stretching (IV) bending [30].

Additive manufacturing and 3D printing are two words used interchangeably to discuss the same manufacturing process. Additive manufacturing differs from subtractive manufacturing (CNC) based on the manufacturing process. As the name *subtract*ive implies, it takes a block of solid material and uses sharp rotating tools or cutters to remove (*subtract*) all parts that are not needed to manufacture the part. On the other hand, *add*itive manufacturing involves parts being created (*add*) layer-by-layer using solid material extrusions from a heated nozzle (material extrusion method). Besides, there are other variants of additive manufacturing techniques such as: Directed Energy

Deposition (DED); Material jetting; Powder Bed Fusion (PBF); Laminated Object Manufacturing (LOM) and Ultrasonic Additive Manufacturing (UAM). Rapid prototyping capability of additive manufacturing paved its way to diverse applications in industries, such as medical, jewellery, construction, apparels, architecture, automotive, robotics and communication.

There are a plethora of CNC techniques in cutting and making creative compositions of fabrics made using loom apparatus [31]. However, 3D printing of textiles is a trending topic where many researchers and creative designers work in parallel to add a new dimension to the conventional loom-based textile manufacturing processes [32, 33]. Figure 1.2 illustrates several 3D printed knitted textile specimens under different loading conditions. These specimens are manufactured by digitally interlooping the nylon wires made using Selective Laser Sintering (SLS) technology and expected to revolutionise the apparels industry in the near future [30]. This expected boom is mainly due to the rapid fabrication of textiles as well as the improved sustainability indices of the manufacturing processes [30].

Lattice structures or lattice-based metamaterials have secured a special place among other structural forms due to its dynamic load transferring mechanisms [34]. They are usually comprised of repeating unit cells (UCs) designed and engineered to achieve innovative and desired properties. Through the control of various design parameters, lattice structures can produce unique mechanical, electrical, thermal and acoustic properties, and thus have received compelling research attention over the past few years [35]. The recent boost in lattice structures was due to the rapid progression of additive manufacturing, particularly Selective Laser Melting (SLM) [35]. As shown in Figure 1.3, lattice unit cells can be arbitrarily complex with many nodes and elements in the cell domain. These cell elements are usually modelled as spatial rods (or struts) occupying the full 3D space during deformation. The wide range of applications of lattice structures in manifold engineering fields such as civil, mechanical, robotics, bio-medical, aerospace and acoustics, demand not only accurate and robust computational tools but also require parametric numerical models to refine the expected performance to its optimum.

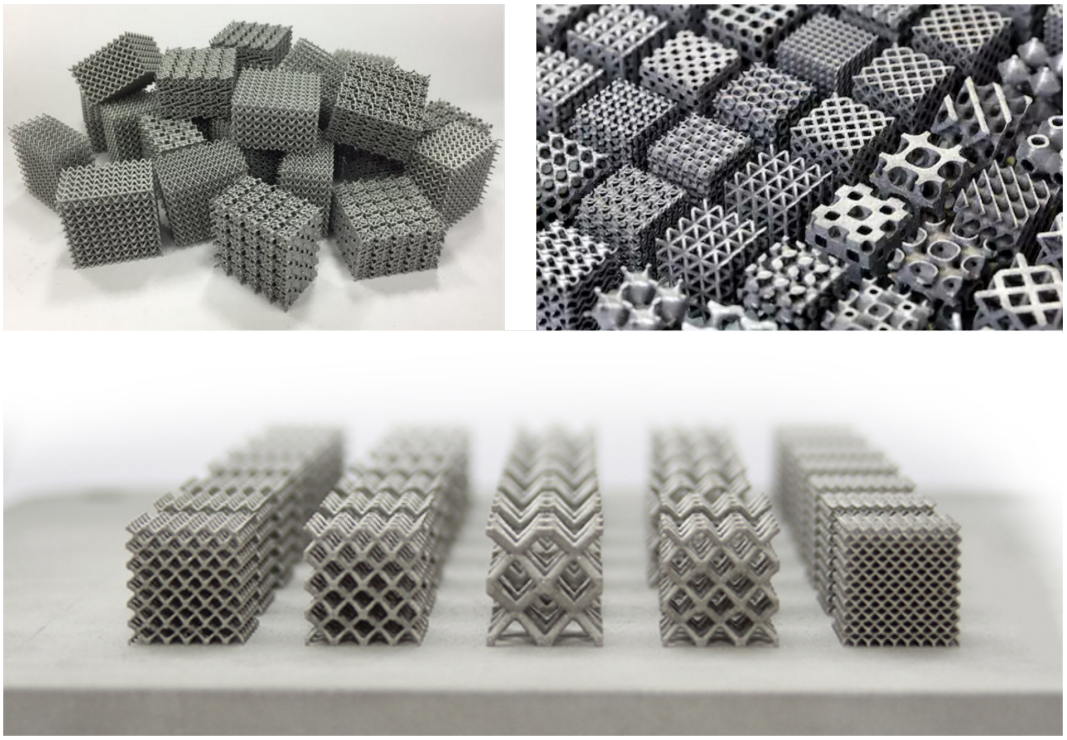


Figure 1.3: 3D printed periodic lattice structures [36]. These lattices possess unique representative volume elements that attract a wide range of computational homogenisation strategies.

Data-driven analytics

Machine learning is a new trending field these days and is seen as a subset of artificial intelligence (AI). Machine learning uses several statistical algorithms to make computers think in a certain way without being explicitly programmed. These algorithms receive an input value (or vector) and predict an output(s) for this through the use of statistical methods. The strong versatility of machine learning outclassed traditional statistical modelling techniques, which paved the way for it to be used in diverse fields such as cyber-security, banking, finance, military, healthcare, robotics, automotive, social media and mass transportation.

Trend identification, handling multi-dimensional and multi-variety data, automation of algorithmic computations are few of the many merits of machine learning that attract computational structural mechanics communities [37]. Notwithstanding, the time taken for model training, the accuracy of predictions, compatibility within a finite element

environment are challenges that need special attention in the process of integrating machine learning methods in a finite element solver. These challenges can be systematically overcome by a thorough understanding of the strengths and weaknesses of various machine learning techniques: neural networks, kernel methods, sparse kernel machines, graph models [37].

Exploiting the novelties in isogeometric analysis, data-centric engineering, additive manufacturing, nonlinear spatial rod models, nonlinear computational homogenisation and machine learning techniques, we are motivated to devise frameworks to analyse the complicated behaviour of technical textiles. The non-existence of a unified data-driven framework for the analysis and design of woven and knitted textiles leaves a research gap that needs to be addressed. The same framework is directly applicable to 3D printed periodic structures discussed earlier.

1.3 Contribution

This thesis proposes a rigorous data-centric computational homogenisation strategy for the analysis and design of technical textiles. If we look at the big picture, the devised method outshines the conventional computational strategies [38–40] for its efficient, robust and flexible model implementations with the aid of statistical learning [41]. Several milestones, which can be recognised as novelties in respective research communities, have also been attained in addition to successfully achieving the final objective.

As known from differential geometry [42], manifolds provide a rigorous framework for describing and analysing surfaces with arbitrary topology [43]. Although there have been several successful attempts to use *smooth* manifold concepts in the computational mechanics domain [44–46], no significant contributions have been made in applying *non-smooth* manifold concepts in the same domain. We introduce a novel set of univariate basis functions for geometric modelling and simulation of non-smooth features in rod structures. Within the framework of isogeometric analysis, we prove these basis functions are accurate in simulating hinges in structures. Continuation of this work has been accepted for publication in Computer Methods in Applied Mechanics and Engineering and the preprint is available on arXiv [47].

The biggest challenge in computational homogenisation of fibrous materials is to develop a rigorous and robust finite element environment for modelling and simulation.

We present a nonlinear isogeometric spatial rod model with discussions on the novel formulation aspects compared to existing implementations [50–52]. These would include analytically exact expressions for initial curvatures, isogeometric analysis of hinged rods, follower (non-conservative) loading and non-zero external stiffness matrices. A wide range of benchmark problems, in the nonlinear regime, has been solved using the proposed rod model for model verification.

Linear material anisotropy in a differential geometric approach has been implemented and verified for orthotropic materials in the small strain regime. This is an important finding to couple microscale with macroscale in linear computational homogenisation by extracting the ABD stiffness matrices from microscale unit-strain analyses [57–59].

Data-driven nonlinear computational homogenisation has been considered by Le et al. [60] and Bessa et al. [61] predominately by using neural networks at the off-line stages of the implementations. In this thesis, GPR is taken as the machine learning tool of choice for model training, testing and predictions. From an application point of view, there is no work done on nonlinear computational homogenisation of technical textiles using machine learning techniques. Thus, the proposed data-driven computational homogenisation and its neat extension to material designing are considered to be novel contributions to the research community.

1.4 Layout

Chapter 2 introduces conventional geometric modelling techniques that are prevalent in isogeometric analysis (IGA) communities. These include B-spline curves, B-spline surfaces, manifold-based smooth curves and manifold-based non-smooth curves. The chapter begins with univariate B-splines aiming to explain several fundamental concepts which form the basis for other geometric modelling techniques. Moreover, chapter qualitatively discusses the merits and demerits associated with the existing modelling techniques under the domain of isogeometric analysis. Chapter 2 ends with the introduction of a set of novel basis functions to simulate hinges of rod structures and surfaces.

Chapter 3 begins with a review of beam theories to feature the evolution of modelling of slender bodies using 1D continuum theories. The chapter categorises a variety of beam theories based on the underlined theory, geometric considerations and kinematic assumptions. Also, this discussion is extended to highlight the advantages and disadvantages

of some beam theories. Next, the chapter outlines rigorous formulations and implementation aspects of a nonlinear isogeometric spatial rod, emphasising on the novelties introduced to mitigate some of the weaknesses in prevailing beam theories. Chapter 3 ends with solving a set of benchmark problems to test the accuracy and efficiency of the proposed nonlinear isogeometric spatial rod model.

Chapter 4 introduces the theory of Kirchhoff-Love thin shells under the domain of differential geometry. A review of thin shell mechanics follows by explaining the formulations and implementations of thin shell kinematics under a finite element analysis (FEA) environment. This discussion is extended to incorporate linear material anisotropy to the thin shell theory. Implementations are verified subsequently using a linear orthotropic material model. Chapter 4 concludes by highlighting the usefulness of the linear material anisotropy of thin shells in linear computational homogenisation.

Chapter 5 kinematically couples the models introduced in chapters 3 and 4 by introducing nonlinear computational homogenisation. Introducing conventional multiscale modelling techniques and highlighting their merits and demerits, the chapter unfolds the basics of machine learning to propose a data-driven multiscale modelling framework to mitigate the inherent weaknesses of conventional multiscale methods. The proposed framework is further elaborated by incorporating the Gaussian Process Regression machine learning technique. Lastly, a neat extension to the data-driven multiscale modelling framework is proposed. This extension will combine data-driven techniques with nonlinear material designs of structures.

Chapter 6 applies the proposed data-driven multiscale modelling and data-driven material designs to simulate technical fabrics in the macroscale. Two families of technical fabrics are considered here to verify the proposed data-driven techniques: plain-woven and weft-knitted fabrics. First, RVE simulations of plain-woven and weft-knitted fabrics are performed and verified. Secondly, data-driven techniques are used to build response databases, Gaussian Process Regression model training and predictions. Next, the trained Gaussian Process regression models are coupled at integrations points of the thin shell finite elements introduced in Chapter 4 to complete the multiscale strategy. Simulations are extended to material designs of plain-woven and weft-knitted fabrics to conclude the chapter.

Chapter 7 summarises the thesis with a few concluding remarks and presents several potential research avenues for future work.

This chapter formally introduced the reader to the big picture of this research by highlighting the research gaps, motivations and contributions. Next chapter is intended to discuss on different geometric modelling techniques mainly focussing on univariate basis functions that are later used to model spatial curves. Chapter 2 can be identified as the foundation of the thesis as it builds the basis for isogeometric analysis of spatial rods.

2 Geometric representation

2.1 B-splines

B-splines are fundamental to many fields: approximation and data fitting [1]; computer graphics; geometric modelling [62, 63]; automated manufacturing; isogeometric analysis [44, 45, 54, 62, 64] and numerical simulation. Hence, B-spline theory can be broadly grouped into: approximation methods (mathematics); modelling techniques (engineering); and geometric algorithms (computer science) [65].

2.1.1 B-spline curves

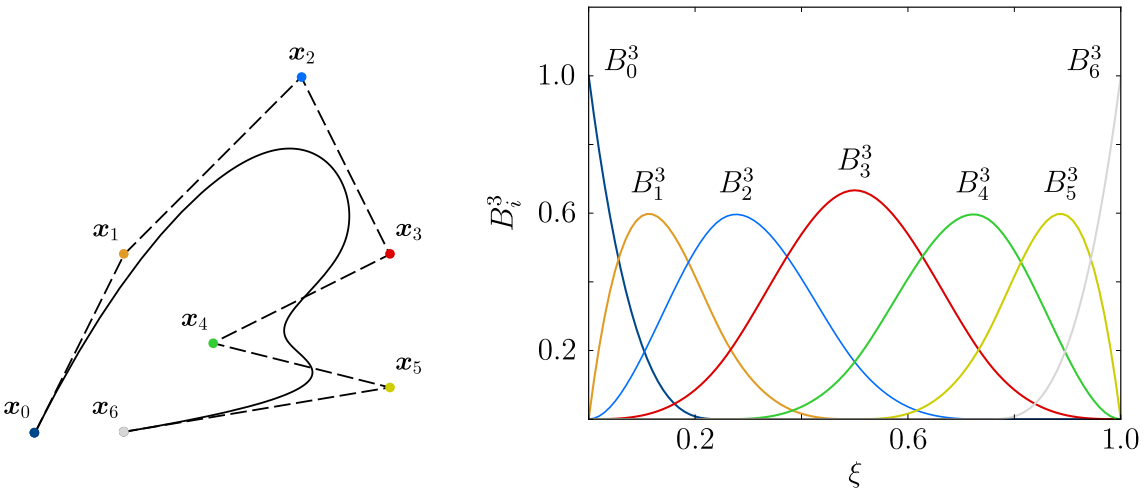
The motivation in discussing B-splines in this thesis is twofold. Firstly, B-splines form the general basis for many geometric modelling techniques like B-spline, Bézier, Subdivision, Non-Uniform Rational B-splines (NURBS), T-splines and Manifold-based modelling [45, 65, 66]. Secondly, B-spline based shape functions are used in isogeometric analysis of finite deformable spatial rods in Chapter 3.

The use of splines for creating free-form curves dates back to 1752 when it was used in ship construction [66]. The development of B-splines in the context of computer-aided design (CAD) predominantly took place in the 1960s with several independent efforts. Some of the significant contributors included Coons, Ferguson, Sabin, de Casteljau and Bézier [66]. In 1970, Gordon and Riesenfeld merged the theory of B-splines and Bézier curves and showed that B-spline curves are a generalisation of Bézier curves [67]. B-spline curves now lie at the core of numerous CAD systems and form the bedrock of computerised geometric modelling techniques.

For convenience, we first focus on univariate B-splines which are widely used in modelling spatial curves. B-splines (or formally known as basis-splines) are piecewise polynomial functions of degree μ, defined on a parametric domain ξ. This is in contrast to Bézier curves which consist of a single polynomial segment. The points where the B-spline functions meet are called the *knots*, ξ_i. A knot vector, $\boldsymbol{\xi}$, for a B-spline curve is a non-decreasing set of parametric coordinates [65] and has a size of $\mu +$ number of control points $+ 1$. A B-spline curve is a spatial parametric curve defined as the linear combination of the univariate B-spline basis functions $B_i^\mu(\xi)$ and control points $\boldsymbol{x}_i$,

$$\boldsymbol{x}(\xi) = \sum_i B_i^\mu(\xi)\boldsymbol{x}_i \; . \tag{2.1}$$

where $B_i^\mu(\xi)$ denotes the B-spline basis function (parameterised by ξ) of degree μ associated with each control point $\boldsymbol{x}_i$. It is highlighted here that in computer graphics literature, it is common to refer to a B-spline by its *order* [68]. The *order* of the spline is one higher than the degree of the B-spline i.e. $order = \mu + 1$.



(a) (b)

Figure 2.1: (a) Control points and the control polygon in dashed-black and the resulting cubic B-spline curve in solid black. (b) Corresponding cubic B-spline basis functions plotted in a parametric space for the knot vector, $\boldsymbol{\xi} = \{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$.

Figure 2.1 is used to illustrate the modelling of B-spline curves using some control points and a predefined B-spline basis. Figure 2.1a depicts the initial control points and the associated control polygon in dashed lines. In addition, Figure 2.1b shows the cubic

B-spline basis functions in a parametric domain $\xi \in [0, 1]$. The resulting limit curve is given by Equation 2.1, which is shown in Figure 2.1a. Furthermore, each control point $\boldsymbol{x}_i$ is assigned to one basis function B_i^3 according to Equation (2.1) and colour coded in Figure 2.1b.

There are several ways to derive B-spline basis functions, e.g. polar form [69], convolution [68], repeated averaging [65, 66, 70] etc. In this thesis, B-spline basis functions are defined using a repeated averaging formula given by

$$\begin{aligned} B_i^0(\xi) &= \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_i^\mu(\xi) &= \frac{\xi - \xi_i}{\xi_{i+\mu} - \xi_i} B_i^{\mu-1}(\xi) + \frac{\xi_{i+\mu+1} - \xi}{\xi_{i+\mu+1} - \xi_{i+1}} B_{i+1}^{\mu-1}(\xi) \end{aligned} \tag{2.2}$$

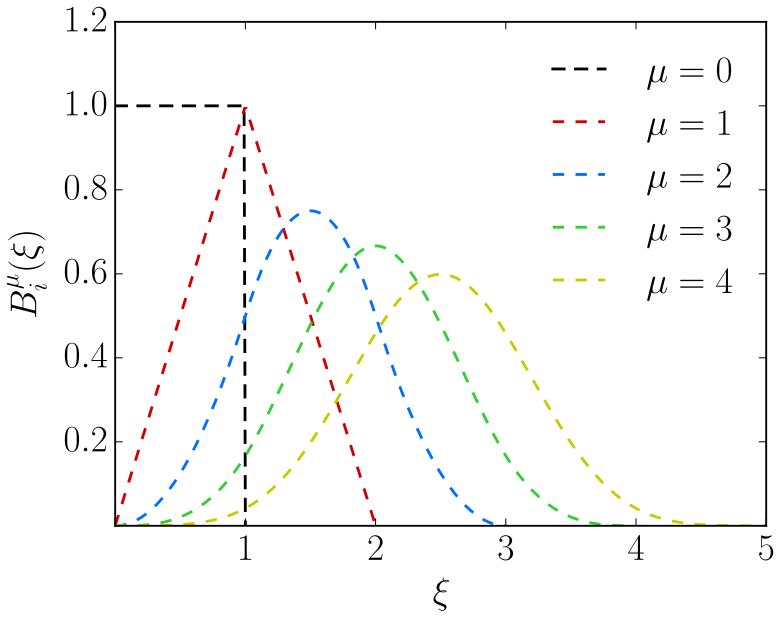


Figure 2.2: Uniform B-splines basis functions for $\mu \in \{0, 1, 2, 3, 4\}$

Figure 2.2 plots the B-spline basis functions on a uniform knot vector for $\mu \in 0, 1, 2, 3, 4$. For $\mu = 0$ it represents the constant spline as given by Equation (2.2). The case of $\mu = 1$ represents linear splines which is identical to linear Lagrange basis (hat) functions [71,72] used in classical finite elements. It can be seen that the B-spline basis functions are supported and non-zero over the knot interval $\xi_i \leq \xi < \xi_{i+\mu+1}$. This compact support

of the basis functions is extremely useful in FEM for maintaining the sparsity of the stiffness and mass matrices.

However, in contrast to the uniform knot vector, if the knot spacing is not equal, the knot vector is said to be non-uniform. The non-uniform vector gives designers more flexibility in embedding local enrichments to geometries [66]. Figures 2.3a and 2.3b show the uniform and non-uniform cubic B-spline basis functions respectively. It should be noted that the continuity of the B-spline curves is dependent on both the polynomial degree, μ, and knot vector definition $\boldsymbol{\xi}$. In general, the basis functions are $C^{\mu-k_i}$ continuous across the knots. Here, k_i denotes the multiplicity of the knot ξ_i. This is clearly demonstrated in Figure 2.3b, where the basis functions are $C^{3-3} = C^0$ at ξ_1.

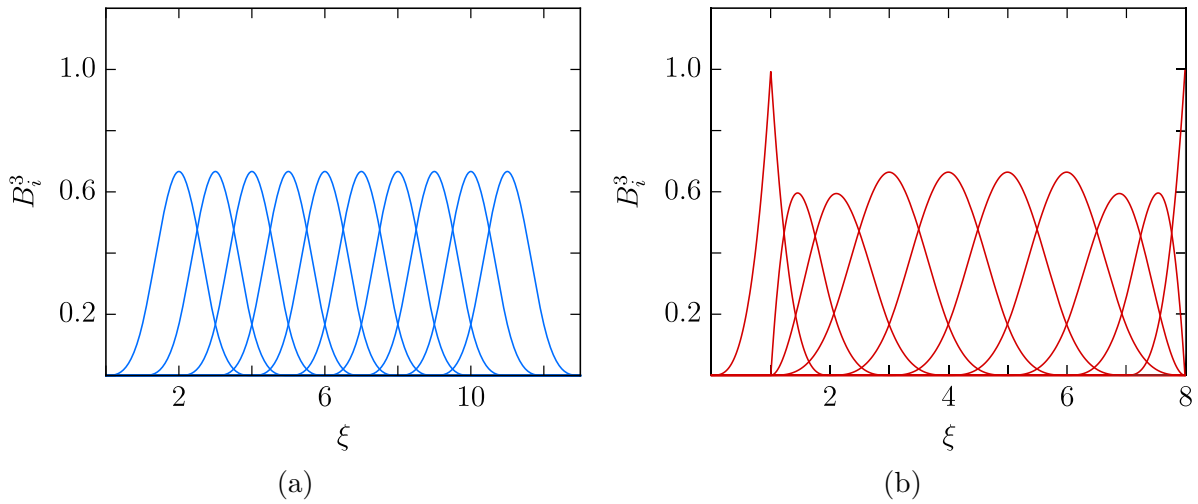


Figure 2.3: (a) Cubic B-spline basis functions over uniform knot vector. (b) Cubic B-spline basis functions over non-uniform knot vector defined as $\boldsymbol{\xi} = \{0, 1, 1, 1, 2, 3, 4, 5, 6, 7, 8, 8, 8, 8\}$.

In most CAD applications, it is prevalent for the first and last knots to be repeated $\mu + 1$ times (see Figure 2.1). A knot vector of this kind is said to be *open* [73]. This circumvents the inclusion of redundant *ghost* control points to generate geometry. This concept is illustrated in Figure 2.4 which shows a cubic B-spline curve generated from the B-spline basis functions over uniform (Figure 2.4a) and open (Figure 2.4b) knot vectors. In the case of a uniform cubic B-spline basis, ghost control points are needed to create a curve through the same set of control points. The number of ghost control points increases with the degree of the basis function. For complex geometries, it is not straightforward to derive the exact positions of these ghost nodes because of their geometric-sensitivity.

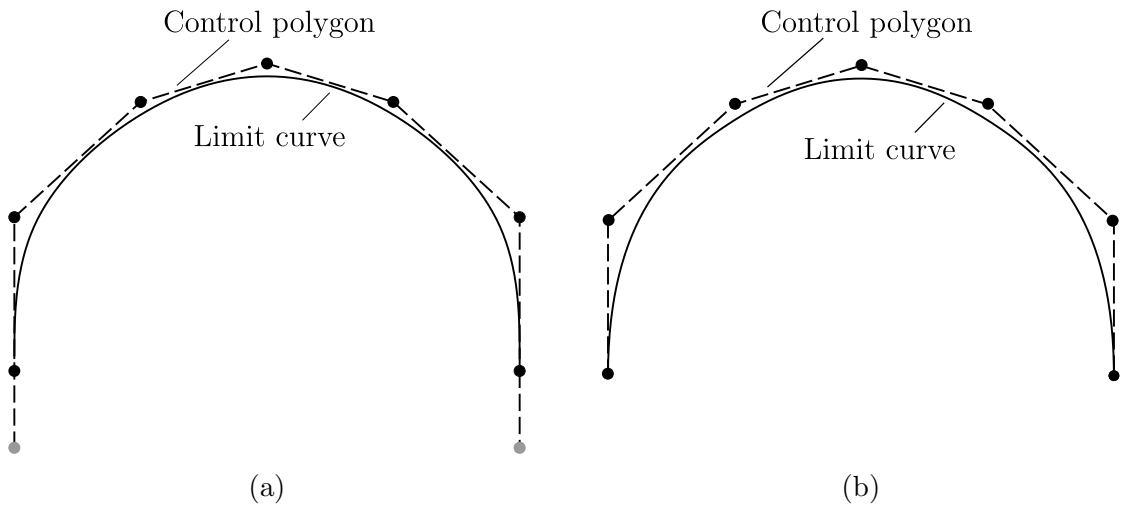


Figure 2.4: Limit cubic B-spline curve for the control mesh (dashed lines) shown for the (a) uniform and (b) open knot sequences. In case of a uniform knot vector, *ghost* control points (grey) are needed to generate a similar curve.

It is also important to investigate the geometric approximations of B-splines of different degrees. Figure 2.5 illustrates the resulting curves for a given set of control points (a fixed control polygon) when the B-spline degree is increased from 1 to 6. Open knot vectors, as shown in Figure 2.3b, are used to circumvent the ambiguity of selecting ghost nodes in generating the curve in Figure 2.5.

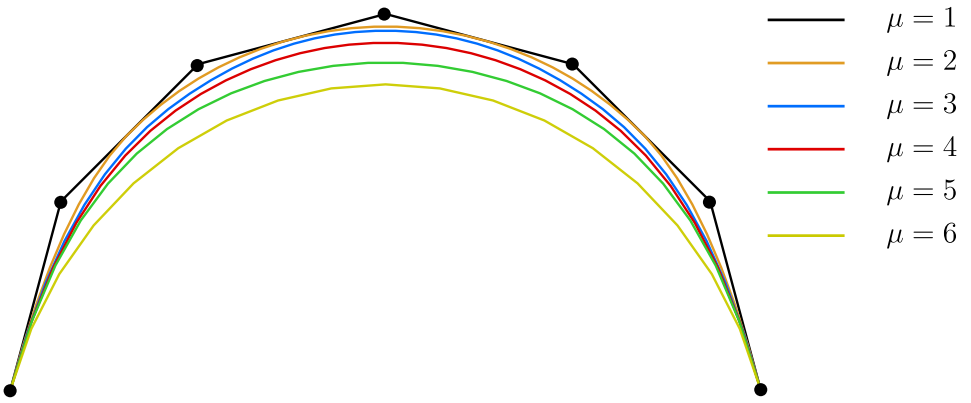


Figure 2.5: Limit curves for a fixed control mesh using different degrees of B-spline basis and respective open knot vectors. For instance, for $\mu = 3$, $\boldsymbol{\xi} = 0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1$

.

It is straightforward to understand the curve generated using the degree 1 B-spline as it linearly interpolates the control points. Linear B-splines give a maximum value of 1

at one control point and a minimum of 0 at any other control point. Therefore, linear B-spline curve (marked in black) in Figure 2.5 passes through every control point of the control polygon. Following a similar rationale, one can observe that the resulting B-spline curves move away from the control polygon when the respective degree of the B-spline basis functions is increased. This behaviour is evident when referring back to Equation (2.1), because the weight $B_i^\mu(\xi)$ on each control point $\boldsymbol{x}_i$ is reduced for increasing B-spline degree (see Figure 2.2). For instance, $B_i^1(\xi)$ is unity when $\xi = \xi_i$. But $B_i^3(\xi)$ always takes a value less than one at $\xi = \xi_i$.

As discussed earlier, B-splines have become the tool of choice in CAD and finite element analysis. There are many properties of B-splines that deserve mention. But without going into unnecessary details, some of the key properties of B-splines in the context of isogeometric analysis are briefly discussed. For detailed discussions on this topic, interested readers are kindly referred to [66].

Partition of unity: One of the most important properties B-spline basis functions possess is that it forms the *partition of unity*. This property partition the unity function at any point of the parametric domain of the B-spline basis. In classical finite element method, partition of unity is a property that the Lagrangian basis functions of any order should satisfy. Hence, it is important that B-spline basis satisfy this requirement in the domain of classical finite elements. This relation can be succinctly written as,

$$\sum_i B_i^\mu(\xi) = 1 \,. \tag{2.3}$$

Intuitively, the partition of unity means that the basis functions can accurately model rigid body motions [74].

Convexity: The B-spline basis functions, and hence, the resulting geometry, are *convex*. A given domain Ω is defined to be convex if one can connect two points in Ω through a line without leaving Ω. Mathematically, a basis function is said to be convex if,

$$\sum_i B_i^\mu(\xi) = 1, \qquad \text{and} \qquad B_i^\mu(\xi) \geq 0 \,. \tag{2.4}$$

In addition to the partition of unity, Figure 2.2 proves the validity of the second condition in Equation 2.4. Since the basis functions are convex, their linear combinations will also be convex. Thus resulting geometry will also be convex.

The convexity property of a B-spline curve is particularly useful because it ensures that the B-spline curve will always be confined within a well-defined region, known as the convex hull (**conv** Ω). It is highlighted here that the convexity property of the functions is stricter than the traditional requirement of the functions to be *affine*. For an affine function, the basis functions only need to obey the partition of unity property. Hence, all convex functions are a special subset of affine functions [45].

Refinability: B-spline basis functions can be refined. This essentially means basis functions at a coarse level ℓ can be constructed by dilating and translating basis functions at a finer level $\ell+1$. This is commonly known as the two scale relation [75]. Refinability property of B-spline basis functions form the basis of subdivision schemes introduced by: Catmull and Clark [76]; and Doo and Sabin [77]. The refinability property of B-splines lies at the heart of many algorithms based on B-splines [70]. For example, the refinability of B-splines is used to construct a hierarchical B-splines basis [78]. Similarly, Rüberg et al. [79] used this property to construct numerically stable approximation spaces for pressure and velocity to solve fluid problems.

Isogeometric analysis using B-spline basis functions

Isogeometric analysis is a numerical technique that integrates finite element analysis into conventional CAD design tools. The primary motive behind the isogeometric analysis is to use the same basis functions used in geometric modelling in the finite element analysis [62, 73]. This can be achieved using many variants of geometric modelling techniques. Most widely used techniques include B-splines, subdivision, NURBS, T-splines and manifold-based modelling. In this thesis, univariate manifold basis functions and B-splines have been used in the isogeometric analysis of spatial rods. Manifold basis functions used in this thesis consist of two variants, namely smooth and non-smooth functions. Moreover, B-spline basis functions of degree 3 and degree 5 have also been used in the modelling and analysis of spatial rods.

We first discuss the isogeometric analysis using B-spline basis functions. Section 2.2 is reserved for discussing the manifold-based isogeometric analysis paradigm. In this thesis, we use univariate cubic and quintic B-spline basis functions to model the rod geometries and approximate the degrees of freedom. The main reason to use two different degrees of B-spline basis functions comes directly from the complexity of the geometry.

Cubic B-spline basis: Bending strain is defined as the change in curvature which is in turn related to the second derivative of the displacement vector. Thus, the minimum smoothness requirement of the basis functions is restricted to C^2. More details on bending strains can be found later in this thesis under Section 3.2.1. Therefore, cubic B-spline basis functions defined on a uniform knot vector is used to model and analyse finite deformable beams.

Figure 2.6 depicts cubic B-splines defined on a uniform knot vector where, for clarity, functions are shown elevated from the univariate parametric domain ξ. Furthermore, for better visualisation, each B-spline function is colour coded with its corresponding central knot.

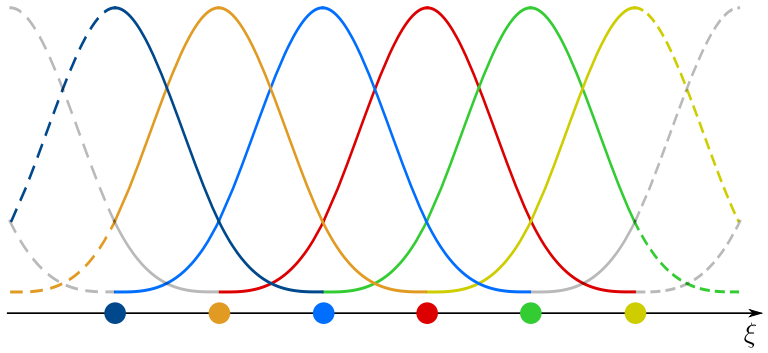


Figure 2.6: Cubic B-spline basis defined over a uniform knot vector. For illustration purpose, the basis functions are assigned to a knot by colour coding and separately shown from the knot intervals.

Due to the symmetry of cubic B-spline basis, we consider only a two-knot interval from Figure 2.6 for illustration purpose. Therefore, taking the interval from blue to green knots and choosing the parametric domain $\xi \in [0, 1]$, we present the basis functions (N^I) and their parametric derivatives $(N^I_{,1}, N^I_{,11}, N^I_{,111})$ as given in Figure 2.7. Notice the presence of inevitable ghost node basis functions (orange and yellow) for the selected knot interval. Therefore, in every parametric element, we have four nodes in total composed of two ghost nodes and two element nodes. A comment is warranted for the third parametric derivative of cubic B-splines. It is evident that cubic B-splines are only C^2 over the knots. But they behave similarly to infinitely differentiable polynomials within two knots. However, numerical experiments show that approximations using third parametric derivatives are prone to higher errors. These erroneous approximations adversely affect the computation of the Frenet-Serret torsion of a space curve [80]. This

issue motivates to use a higher-order B-spline basis, particularly for the modelling and analysis of complex space curves.

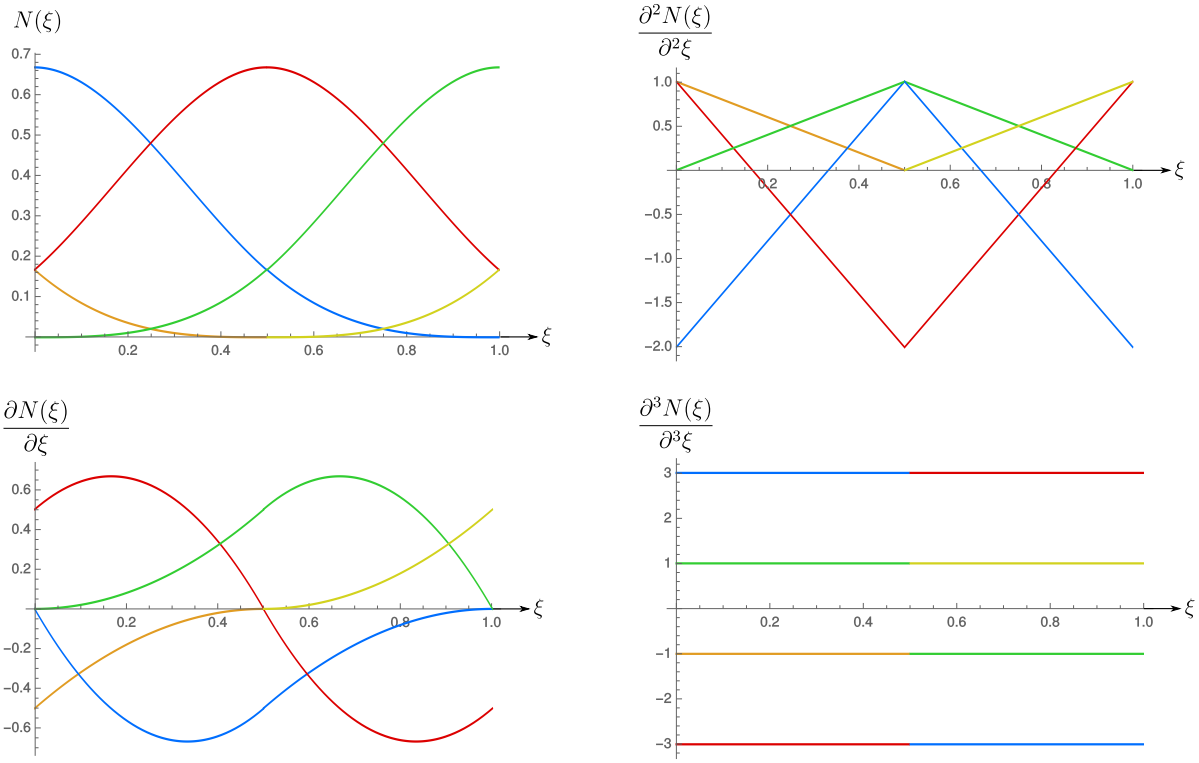


Figure 2.7: Piecewise cubic B-spline basis functions and their parametric derivatives defined over $\xi = [0, 1]$

Quintic B-spline basis: For a general smooth spatial curve, Frenet-Serret [80,81] formulas demand the existence of a torsion constant defined everywhere on the curve. This torsion constant is defined using the third parametric derivative of the curve (see Equation (3.28)). Hence, requiring the basis functions to possess C^3 continuity. Further details on the Frenet-Serret formulas are provided in Section 3.2.1 and Equation (3.27). Moreover, considering the symmetry of B-spline basis, the quintic B-spline basis was chosen in the modelling and analysis of complex rod geometries.

Following similar steps from the cubic B-spline basis, we illustrate quintic B-spline basis and the respective derivatives in Figure 2.9. We pick the interval from blue to green knots and choosing the parametric domain $\xi \in [0, 1]$, we present the basis functions and their derivatives in Figure 2.9. Here, notice the presence of four ghost node basis functions (dark blue, orange, yellow and grey) for the selected knot interval. Therefore

in every parametric element, we have six nodes in total, composed of four ghost nodes and two element nodes.

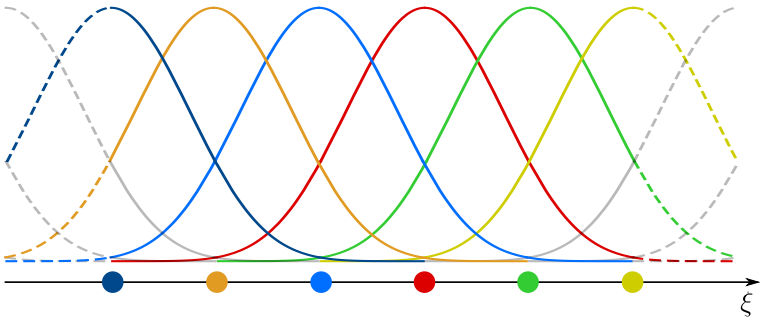


Figure 2.8: Quintic B-spline basis defined over a uniform knot vector. For illustration purpose, the basis functions are assigned to a knot by colour coding and shown separately from the knot intervals.

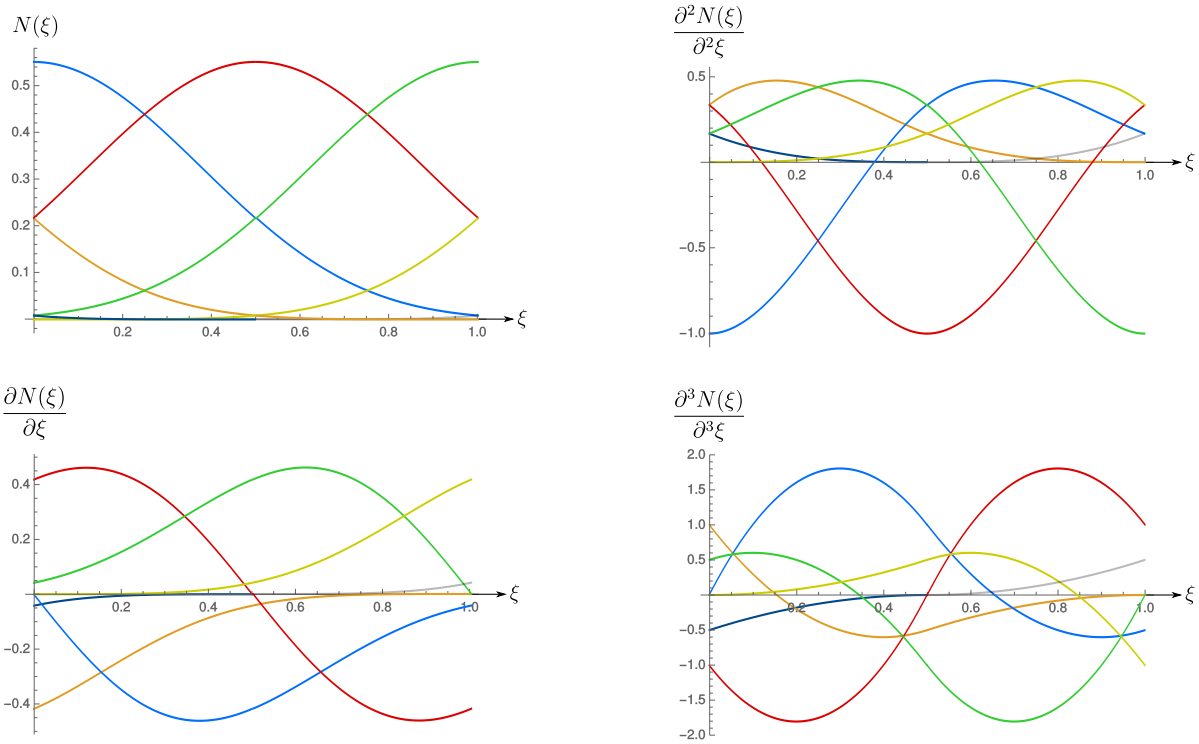


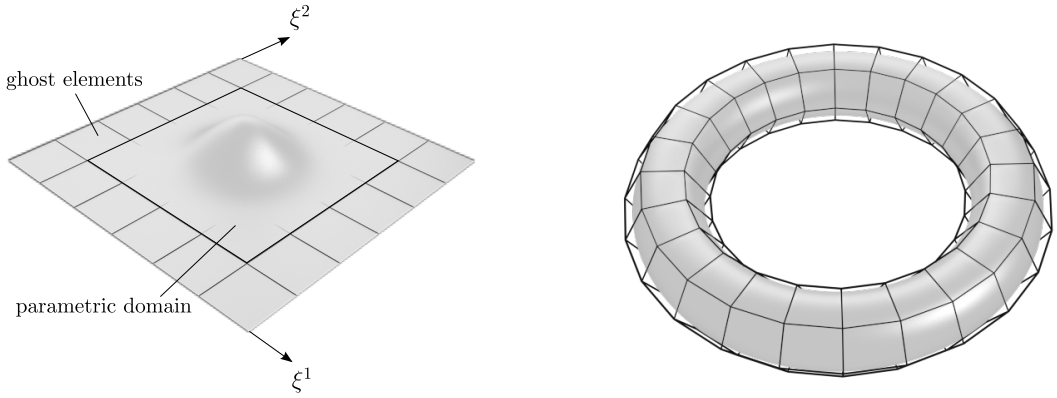
Figure 2.9: Piecewise quintic B-spline basis functions and their parametric derivatives defined over $\xi = [0, 1]$

2.1.2 B-spline surfaces

B-spline curves can be generalised to construct surfaces and solids using tensor product formalism. In case of surfaces, the knot vectors can be defined in each of the two coordinate directions in the parametric space. The tensor product B-spline surface is constructed using control points $\boldsymbol{x}_{i,j}$ and univariate B-spline basis functions $B(\xi^1)$, $B(\xi^2)$ as,

$$\boldsymbol{x}(\xi^1, \xi^2) = \sum_i \sum_j B_i(\xi^1) B_j(\xi^2) \boldsymbol{x}_{i,j} \ . \tag{2.5}$$

Figure 2.10a shows the tensor product uniform cubic B-spline basis function plotted over a parametric domain. Also, notice the ghost nodes in one-ring neighbourhood of the same parametric domain. Similar to the one-dimensional case, two-dimensional basis functions can also be constructed over a non-uniform knot vector. Figure 2.10b shows the control polygon for the torus and the underlying limit surface generated using cubic B-splines.



(a) (b)

Figure 2.10: (a) Uniform cubic B-spline tensor product basis function. (b) Control polygon and the limit surface of a torus. Image credits to Majeed [45]

We restrict further discussions on B-spline surface construction and analysis, as this thesis focusses more on the formulations and implementations of the isogeometric analysis of spatial rods. Chapter 4, however, uses subdivision basis functions in the isogeometric analysis of thin-shells. Since the isotropic material variant of the in-house thin-shell solver is not the author's original work, more details on two-dimensional B-spline basis

functions are not provided here. Readers are kindly invited to refer to the doctoral theses of Majeed [45] and Long [64] for additional information.

It can be appreciated that the B-splines offer several attractive properties for modelling and analysis of structures. Though, it remains a popular tool amongst designers and analysts, B-spline based modelling lacks the capability to exactly represent complex objects, particularly conic sections, commonly used in different engineering fields [73].

2.2 Manifold-based geometric modelling

Many B-spline variants, as explained in Section 2.1.1, attract geometric modellers for numerous reasons. Inherent weaknesses of each geometric modelling technique, including B-spline variants, lead to another that addresses those weaknesses. The inception of manifold-based geometric modelling addresses a few prevalent demerits of existing geometric modelling techniques. The need for additional post-processing operations with NURBS and the lack of higher-order continuity near extraordinary vertices for subdivision schemes are two common issues that have been addressed by manifold-based geometric modelling technique [44, 45]. Manifold-based geometric modelling is also one of the many variants of B-spline based geometric modelling techniques, where it follows the popular manifold paradigm [82] to generate surfaces of arbitrary topology with the desired C^k continuity. In Section 2.2.1, the work on manifold-based smooth curves by Majeed et al. [45] is briefly mentioned to familiarise the reader with the concept. Later, in Section 2.2.2, a novel set of basis functions is introduced to model and analyse manifold-based non-smooth curves.

2.2.1 Manifold-based smooth curves

The popular and most intuitive example to explain a manifold is to think of the work of a cartographer as manifold and atlas concepts arise naturally in the context of creating a world atlas (see figure 2.11). A cartographer usually draws parts of the world map on rectangular sheets and combine them together to get the atlas. The world map itself is a complicated surface, but each of the drawings is on a simpler plane. The cartographer decides about how much of the world to draw on each page, and how to *flatten* that

part (curved surface) of the world onto the page (planar sheet). Every part of the world shows up on at *least* one page, and many places show up more than once.

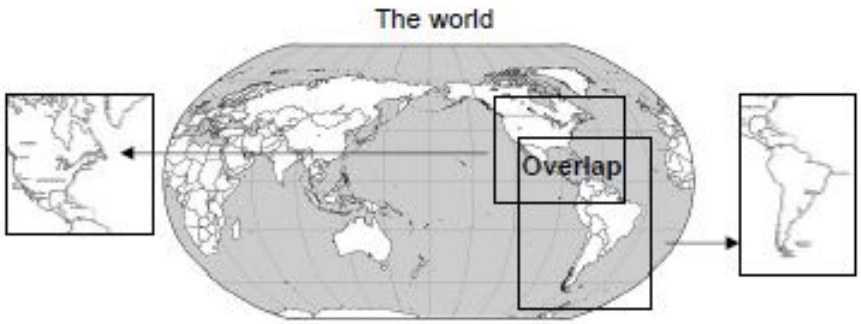


Figure 2.11: Creating two pages (charts) of a world atlas (manifold). Image credits to Grimm et al. [83].

Manifold-based geometric modelling of curves requires three main ingredients. They are namely transition functions, blending functions and local polynomial functions. To this end, we define a planar patch ($\hat{\Omega}_i$) and its projection onto the physical domain (Ω) as Ω_i. The relationship between Ω and possibly overlapping Ω_i is given by,

$$\Omega = \bigcup_{i \in \mathbb{N}} \Omega_i \tag{2.6}$$

Three ingredients that play equally important roles in manifold-based modelling are discussed briefly to establish the basics of manifold-based isogeometric analysis.

Transition functions: Each planar patch $\hat{\Omega}_i$ is uniquely defined by a mapping function $\boldsymbol{\varphi}$ that maps $\hat{\Omega}_i$ onto Ω_i on Ω. However, for robust geometry construction, a smooth navigation between two planar patches is imperative. This navigation between planar patches is facilitated by the *transition functions*. Navigation from $\hat{\Omega}_i$ to $\hat{\Omega}_j$ is formally written as $\boldsymbol{t}_{ji}$ and given as a function composition by,

$$\boldsymbol{t}_{ji} = \boldsymbol{\varphi}_j^{-1} \circ \boldsymbol{\varphi}_i \tag{2.7}$$

Blending functions: Blending functions (w_i) are chosen to respect the partition of unity and be smooth everywhere in the planar patch domain. Blending functions are defined on planar patches and can be a set of normalised arbitrary functions w_i, as given in Equation (2.8). Previous work on manifold-based modelling, by Majeed [45], uses cubic

B-splines as the blending functions because cubic B-splines satisfy all the aforementioned requirements. Figure 2.12a illustrates the univariate blending functions (solid lines) on five overlapping patches.

$$w_i = \frac{\hat{w}_i}{\sum_j \hat{w}_j} \ . \tag{2.8}$$

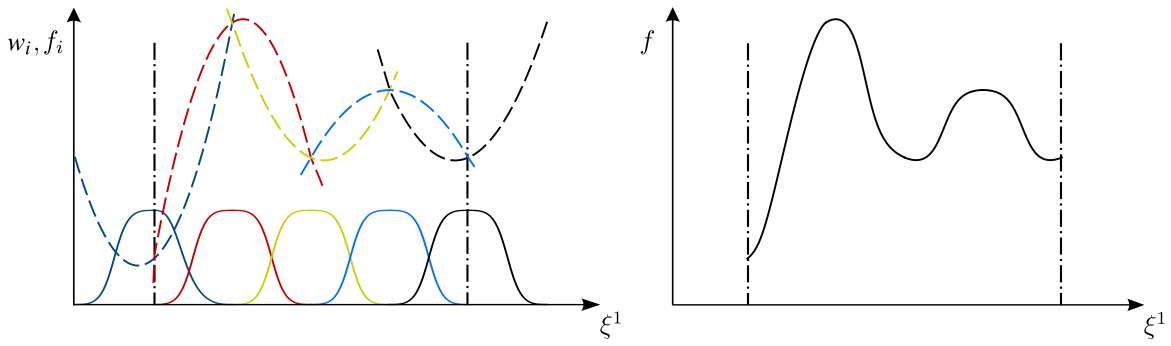
Local polynomial functions: Geometric features of the final curve f is predominantly governed by the local contributions from planar patches. We define arbitrary functions (dashed lines) over the planar patches as shown in Figure 2.12a. Restricting the arbitrary functions to polynomials, the local contributions are formally termed *local polynomial functions*. Local polynomial contributions (f_i) are written as,

$$f_i = \mathbf{c}_i \cdot \boldsymbol{p} \ . \tag{2.9}$$

where $\mathbf{c_i} = [c_1, c_2, c_3 \cdots]_i^{\mathrm{T}}$ is a vector of co-efficients and $\boldsymbol{p} = [x^0, x^1, x^2, \cdots]^{\mathrm{T}}$ is the monomial basis. Now we write the manifold-based global approximant f, using Equations (2.8) and (2.9) as,

$$f = \sum_i w_i f_i = \sum_i w_i \left(\mathbf{c}_i \cdot \boldsymbol{p}_i\right) \ . \tag{2.10}$$

The resulting smooth curve from Equation (2.10) is shown in Figure 2.12b.



(a) (b)

Figure 2.12: Construction of a smooth curve on a domain with five overlapping patches. (a) Blending functions w_i (solid) and the local polynomials f_i (dashed) are shown. (b) The obtained smooth curve f is shown over the overlapping patches. Image credits to [45].

Now we derive the manifold-based basis functions for one-dimensional meshes. Here, a

patch with three control points, that we refer to as a one-ring domain with respect to the mid point, is considered. This one-ring of a vertex is considered to be a planar patch, and we take Lagrange basis functions ($\boldsymbol{L}(\xi)$) [71] as the local polynomial functions. This choice invites us to write the co-efficient vector in Equation (2.9) simply as the coordinates of the one-ring vertices. We consider a point $\mathrm{P}(x^1, x^2, x^3)$ on the smooth curve and for simplicity, we write the following equations considering only x^1. The same formulations can be applied to the other two coordinates.

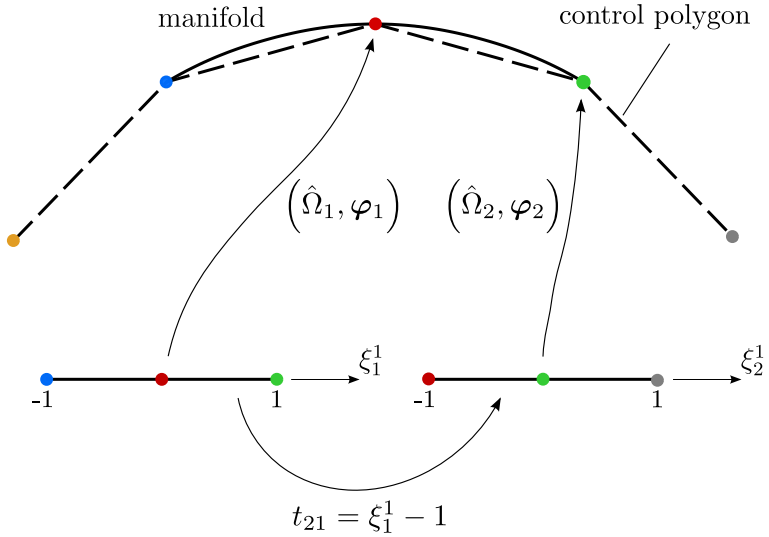


Figure 2.13: A control polygon (dashed lines) with the resulting smooth curve f (solid lines). On each of the two shown one-rings, the blending function is a normalised cubic B-spline and the local polynomial is a quadratic polynomial. Image credits to [45].

Following from Equation (2.10), we extend the formulations for Lagrange polynomials as,

$$x^1(\xi_i) = \sum_j w_j(\xi_i)(\boldsymbol{x}_j^1 \cdot \boldsymbol{L}(\xi_i))\,. \tag{2.11}$$

It is highlighted that $\boldsymbol{L}(\xi_i)$ has three entries, as is the case for quadratic Lagrange basis functions. Now we expand the vector dot product in Equation (2.11) to arrive at the manifold-based smooth basis functions.

$$x^1(\xi_i) = \sum_j w_j(\xi_i) \left(\sum_{I=1}^{3} x_{j,I}^1 L_I(\xi_i) \right) = \sum_{I=1}^{3} \left(\sum_j w_j(\xi_i) L_I(\xi_i) \right) x_{j,I}^1 = \sum_{I=1}^{3} N_I(\xi_i) x_{j,I}^1\,. \tag{2.12}$$

where N_I are the three basis functions corresponding to the three vertices in the patch.

Due to symmetry, Figure 2.14 depicts the basis functions and their derivatives in half of a one-ring patch. Here, a special note is made regarding the smoothness of the manifold basis functions. The overall smoothness of the manifold basis functions depends on the individual smoothness of blending functions and local polynomial functions. Polynomials are infinitely differentiable, and therefore the smoothness of the basis functions solely depends on the smoothness of the blending functions. Thus, using normalised B-splines as blending functions restrict the smoothness of the final curve to be C^2. Readers are recommended to see both Figures 2.13 and 2.14 in conjunction to get a better idea with the colour-coded vertices and basis functions.

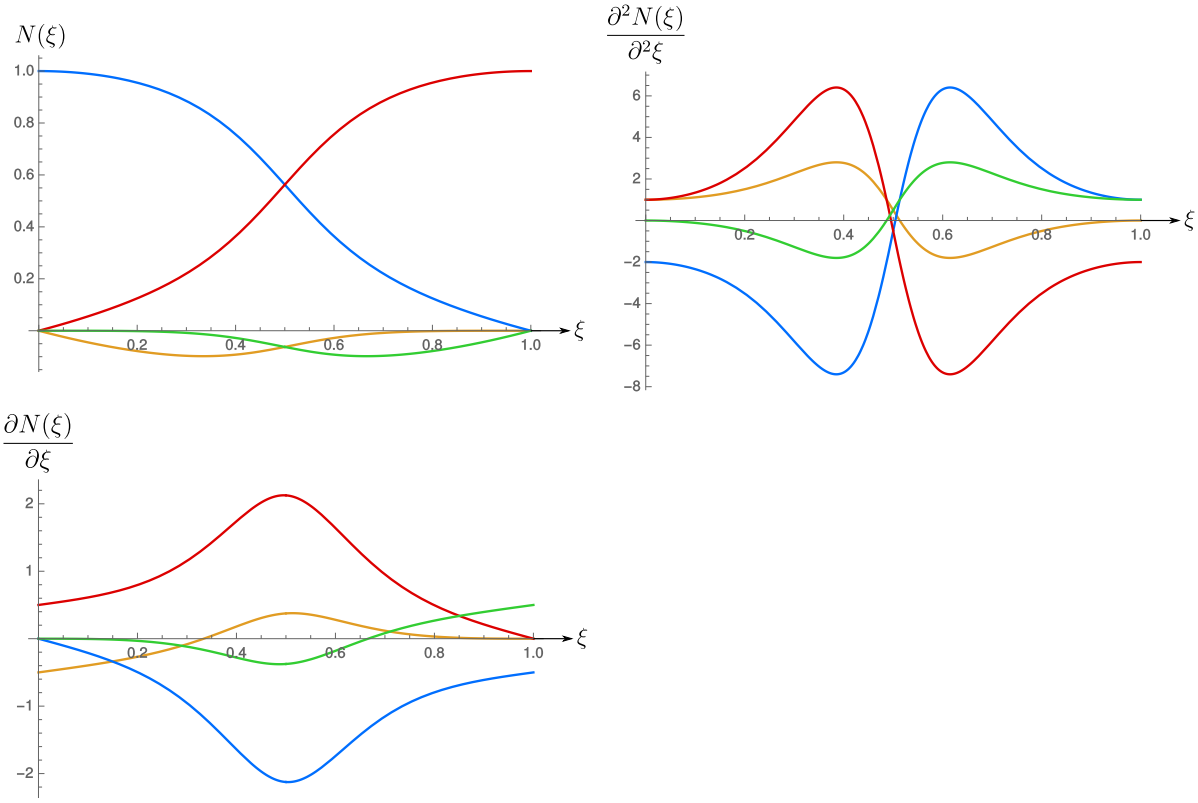


Figure 2.14: Manifold basis functions and their parametric derivatives in an element. There are four non-zero basis functions with two negative ones corresponding to vertices on the neighbouring elements.

2.2.2 Manifold-based non-smooth curves

In general mathematical analysis, the smoothness of a function is a property measured by the number of derivatives it has which are continuous. A smooth function is a function

that has derivatives of all orders everywhere in its domain. For instance, it is intuitive that any polynomial, which takes the form $f = \boldsymbol{c} \cdot \boldsymbol{p}$ (where $\boldsymbol{p}$ is the monomial basis and $\boldsymbol{c}$ contains the coefficients), is C^∞ continuous, or in other words, f has derivatives of all orders. Similarly the function, $g = |x|$ is C^0 continuous, as its first derivative is not defined at $x = 0$.

There are many practical examples for the existence of non-smooth features of a geometry. Figure 2.15 shows a few of such non-smooth features that are commonly seen in mechanical and civil engineering applications.

(a) (b) (c)

Figure 2.15: Non-smooth geometric features in engineering applications. (a) Origami-inspired structures [84] (b) Three-hinged bridges [85] (c) Architectural forms [86].

Extended finite element method, or commonly referred to as XFEM in associated communities, is useful for the approximation of solutions with pronounced non-smooth characteristics in small parts of the computational domain, for example near discontinuities and singularities. In these cases, standard numerical methods such as the finite element method (FEM) or finite volume method (FVM) often exhibit poor accuracy [87]. A detailed discussion on XFEM is not intended to be carried out here. Instead, the general formulation of XFEM is outlined as,

$$\boldsymbol{u}(\xi) = \sum_{i \in I} N_i(\xi) \boldsymbol{u}_i + \sum_{i \in I^*} M_i(\xi) \boldsymbol{a}_i \tag{2.13}$$

where the first term on the right-hand side is the standard finite element approximation and the second term serves the local enrichments to incorporate the non-smooth features. Moreover, $I^* \subset I$ is referred to nodal subset of enrichment, $\boldsymbol{a}_i$ are unknown enrichments and $M_i(\xi)$ are local enrichment functions at node i.

Alternatively, one can think of incorporating non-smooth features into the geometry

using the manifold paradigm. Options would be to replace either local polynomial or blending functions with a non-smooth function so that the resulting function would contain non-smooth geometric features. It is not straightforward to replace the blending functions, B-splines in this context, with a non-smooth function because only $\mu = 0, 1$ B-splines are non-smooth. Due to the lack of flexibility in blending functions, it is intended to replace local polynomial with a non-smooth piecewise function. By choosing non-smooth functions as the local polynomial functions in Equation (2.9), we expect that the resulting basis functions will carry more information on the non-smooth features.

Local polynomial approximation

Figure 2.16 (middle) depicts the choice of non-smooth local polynomial defined on a planar patch $\hat{\Omega}_2$. The choice of first order Lagrangian basis functions for non-smooth local polynomial has its own advantages. Mainly and most importantly it has the non-smooth point at a node and satisfies partition of unity in the one-ring domain. In addition, functions have zero second derivatives everywhere (except at the non-smooth point), thus seem feasible to simulate a perfect hinge at the non-smooth nodal point. Furthermore, it does not require to have ghost nodes as it is clear that each element in the one-ring of $\hat{\Omega}_2$ has only two associated local polynomials.

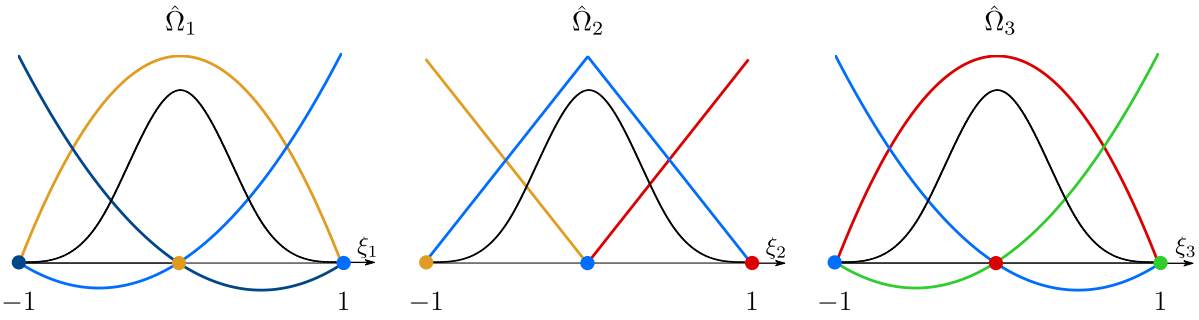


Figure 2.16: Introducing the non-smooth local polynomials onto patch $\hat{\Omega}_2$. On each planar patch $\hat{\Omega}$, a cubic B-spline (black solid lines) and a set of node colour-coded local polynomial functions (coloured solid lines) are shown.

Construction of non-smooth manifold basis functions

Construction procedure of the non-smooth manifold basis functions follows the same steps as explained in detail under the Section 2.2.1. Additionally, the non-smooth local

polynomials defined on planar patch $\hat{\Omega}_2$ in Figure 2.16 are written as,

$$L_j(\xi_2) = \begin{cases} -\xi_2 & \xi_2 \in [-1,0] \quad j=1 \\ 1+\xi_2 & \xi_2 \in [-1,0] \quad j=2 \\ 1-\xi_2 & \xi_2 \in [0,1] \quad j=2 \\ \xi_2 & \xi_2 \in [0,1] \quad j=3 \end{cases} \tag{2.14}$$

Figure 2.17 presents the derived non-smooth manifold basis functions and their first and second derivatives. Note that all second derivatives are zero at the non-smooth point (blue node).

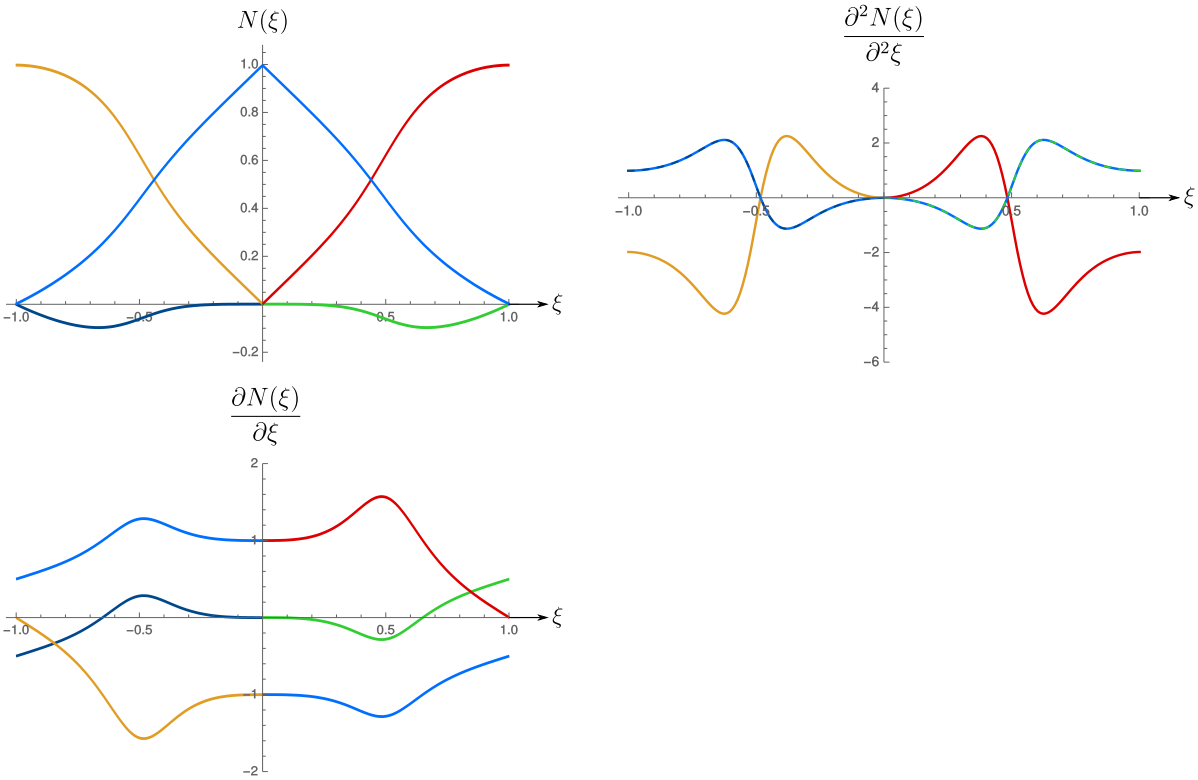


Figure 2.17: Non-smooth manifold basis functions and their parametric derivatives in two neighbouring elements. There are three non-zero basis functions in an element with one negative basis function corresponding to the vertex on the neighbouring element.

One-dimensional meshes

Smooth and non-smooth manifold basis functions are next used to generate a curve for a given control polygon. To generate the desired geometry on a one-dimensional

mesh, we simply use the Equation (2.12) and substitute for N_I with manifold basis functions and $\boldsymbol{x}_I$ with nodal coordinates (x^1, x^2, x^3) of each vertex I of the control polygon. For instance, as shown in Figure 2.18, we approximate a one-dimensional mesh using manifold-based smooth and non-smooth basis functions to distinguish the curve approximation of the two.

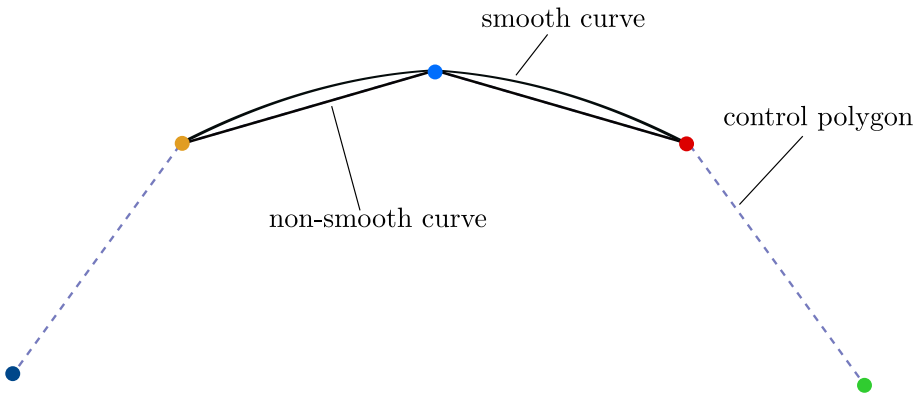


Figure 2.18: Smooth curve (using smooth manifold basis functions) and non-smooth curve (using non-smooth maifold basis functions) generated for a given control polygon

Both smooth and non-smooth curves go through the vertices of the control polygon, as interpolating Lagrangian basis functions were used as local polynomials in both cases. If the local polynomials were selected differently, the shape of the curves in Figure 2.18 would differ based on the qualities of each local polynomial function.

Isogeometric analysis using manifold basis functions

Both manifold-based smooth and non-smooth basis functions can be used in isogeometric analysis. Isogeometric analysis using smooth manifold basis functions is implemented following a similar approach, as explained in Section 2.1. A special note is made for the observed sub-optimal convergence rates when using smooth manifold basis functions [44, 45]. This sub-optimality is attained because the constructed smooth manifold basis functions are *rational* piecewise polynomials due to normalisation. This normalisation also demands the use of more Gaussian integration points than the standard rule for non-rational polynomials. Implementation details of smooth manifold basis functions are discussed extensively in Majeed [45] and hence, detailed explanations are not be

presented in this thesis. However, we aim to discuss the important aspects of univariate non-smooth manifold basis functions in the domain of spatial rods.

We consider the moment balance of a straight Euler-Bernoulli beam [88] given by,

$$M = EI\frac{\mathrm{d}^2 w}{\mathrm{d}x^2} \tag{2.15}$$

where M is the bending moment, EI is the bending stiffness, w is the transverse displacement and x is measured along the beam centreline. We use the discretisation given by,

$$w = \sum_{I=1}^{\mathrm{NBF}} N^I(x) w_I \tag{2.16}$$

where NBF is the number of basis functions, N^I is the basis function corresponding to the I^{th} node and w_I are the transverse nodal displacements. Straightness of the rod implies constant Jacobians, thus it is safe to assume the equivalence of x as the parametric coordinate. Substituting Equation (2.16) in (2.15) we write,

$$\begin{aligned} M &= EI\frac{\left(\mathrm{d}^2 \sum_{J=1}^{\mathrm{NBF}} N^J(x)\right)}{\mathrm{d}x^2} w_J \\ &= EI N^J_{,11}(x) w_J \end{aligned} \tag{2.17}$$

where Equation (2.17) implies a summation over $J = 1, 2, \cdots, \mathrm{NBF}$.

Now we can compare Equation (2.17) in combination with Figures 2.14 and 2.17. We clearly see in the smooth manifold basis functions (see Figure 2.14), moment transfer is allowed among the elements. This is because, within any element, there is no strictly zero second derivatives of the basis functions. Then, the only possibility for zero moment condition in Equation (2.16) would be to have either all $w_J = 0$ (stress-free equilibrium) or all $w_J =$ constant (rigid body motion). These two simple solutions are easily understood for any type of basis functions, because for internal strain energetic consistency, we write,

$$\mathrm{c}\sum_{J=1}^{\mathrm{NBF}} N^J_{,1} = 0 \qquad\qquad \hat{\mathrm{c}}\sum_{J=1}^{\mathrm{NBF}} N^J_{,11} = 0 \tag{2.18}$$

where c and $\hat{\mathrm{c}}$ are two arbitrary constants. Zero values of c and $\hat{\mathrm{c}}$ resemble a stress-free equilibrium condition and any non-zero values denote a rigid body motion of the body.

However, it is interesting to see Figure 2.17 in comparison with Equation (2.17). For any non-constant arbitrary values of w_J, a zero moment condition is reached because all the second derivatives vanish at the node where piecewise Lagrange polynomials were introduced as the local polynomials (blue node of $\hat{\Omega}_2$ in Figure 2.16). In the context of structural mechanics, this phenomenon is known as a *hinge*. Therefore, non-smooth manifold basis functions can be used to simulate perfect point hinges in beams with no additional constraints. These point hinges can be easily extended to simulate line hinges between two surfaces in 3D space. On a structured mesh, this is accomplished using a tensor product formalism of the univariate basis functions.

This chapter was intended to facilitate the readers on several aspects. First, readers were acknowledged of the conventional and recent developments of geometric modelling techniques highlighting the role of the B-spline basis. Next, the isogeometric analysis of spatial curves using B-splines basis was discussed to familiarise the readers with the concept and to specify the types of basis functions that will be used later in this thesis. Next, smooth and non-smooth manifold basis functions were presented as alternative tools for isogeometric analysis. Non-smooth manifold basis functions were discussed in detail because it carries novelty for its ability to simulate hinges in rod structures. In Chapter 3, we aim to provide the readers with rigorous formulations and detailed implementations of a finite deformable spatial rod model. Chapter 2 contributes directly to Chapter 3 by providing univariate smooth and non-smooth basis functions for the isogeometric analysis of finite deformable spatial rods.

3 Nonlinear isogeometric spatial rods

3.1 Preliminaries

Modelling of slender bodies using 3D continuum mechanics theory has been greatly replaced by large deformation beam theories mainly due to two reasons. First, the reduced 1D continuum theories lead to immense computational savings not only because of the vast reduction in degrees of freedom but also due to the simplified kinematics of the slender body. Secondly, from a numerical standpoint, beam theories produce efficient and well-posed numerical formulations and yield robust and high-fidelity algorithms for computations. We identify a family of beam theories proposed and implemented by many authors under different sets of assumptions to simplify the geometry, material models and mechanics. Promising original work on the geometrically exact beam theories by Simo [89] and subsequent work by Simo and Vu-Quoc [50] laid a solid foundation to the research topic and related communities. Other significant contributions to beam theories can be found at Simo and Vu-Quoc [90], Cardona and Geradin [91], Kondoh et al. [92], Dvorkin et al. [93], Jelenic et al. [94]. In addition, the most recent works of Meier et al. [51, 81, 95, 96] and Bauer et al. [52] furnish the beam theories with more efficient algorithmic computations and rigorous formulations.

This chapter unfolds with a review of beam theories, highlighting the evolutions and idealisations by different authors to simplify the geometry, material models and beam kinematics. Next, our motivation to implement the nonlinear isogeometric spatial beam model for the simulation of fibre-based RVEs under the domain of nonlinear multiscale modelling will be discussed. Section 3.2 is reserved for the rigorous formulations of the nonlinear spatial beam within the isogeometric analysis domain. This section brings two novelties to the existing formulations. First, the Jacobian-based analytical derivations for initial curvatures have been proposed to compute the internal forces and Hessian terms accurately. Secondly, our computational model can capture the kinematics from

external loadings with non-zero tangents (with respect to degrees of freedom). In other words, our model facilitates the inclusion of Hessian terms that naturally arise from external loadings. Section 3.3 elaborates on the numerical modelling of beam contacts which is used later in the microscale analyses of textile RVEs used in Chapter 7. Section 3.4 concludes the Chapter 3 by verifying the implemented numerical models using a set of carefully selected benchmark problems. These benchmark problems are aimed to cover various aspects of beam kinematics, contacts, large deformations, non-conservative loading, buckling, hinges, initial curvatures and full 3D geometries.

3.1.1 Review of beam theories

As discussed briefly in Section 3.1, there is a wide range of beam theories developed during the past few decades. Since there is no unanimity in naming each of these beam theories, it is not an easy task to provide a broad review of beam theories. However, in this thesis, various beam theories are classified into three main categories.

- **Underlined theory**: There are two variants of the theory behind beam kinematics. First, we have the linear beam theory where the equilibrium solution is attained by solving the equilibrium equations once. Thus, linear beam theory is valid to solve for infinitesimal deformations and strains. The original Euler-Bernoulli and Timoshenko beam theories were developed under linear theory. This is the main reason any structural mechanics course [97] or text book [71, 88, 98] has the standard deformation-independent stiffness matrices for truss and beam elements. Secondly, we have the nonlinear theory where both deformations and/or rotations can be arbitrarily large. The equilibrium solution is attained using an iterative process such as Newton-Raphson scheme. There are several variants of nonlinear beam theories, but usually originated from Kirchhoff-Love or Simo–Reissner beam theories [51, 52, 89, 99, 100].

- **Geometric considerations**: Many authors simplify the beam kinematics by idealising the beam geometric properties. A set of beam theories rely on initially straight beams and thus does not account for initial curvatures. This reliance is valid in analysing portal frames, trusses, straight beams, crane arms etc., where straight beam pieces are assembled or cast to manufacture the final component. In contrast, the recent boost in 3D printing technology and traditional hot-rolled metal beam bending exposed the engineers to optimise the beam structures by

utilising the arching action arising from initially curved beams. Moreover, analysis of periodic structures like lattices, steel meshes, woven and knitted fabrics also require to consider the effects of initial curvatures. This motivated to develop geometrically exact beam theories by taking the initial beam curvatures into the formulations.

Besides, another common geometric consideration is to constrain the beam cross-sections. Some beam theories can only work with isotropic cross-sections where both major and minor second moments of area are identical. For instance, this type of beam theories can be used in analysing beams with circular cross-sections. Interested readers are invited to refer to Meier et al. [101] for further implementation details on geometric considerations of beam theories.

- **Kinematic assumptions**: The classical Euler-Bernoulli and Kirchhoff beam theories assume vanishing shear deformations in beam kinematics. On the other hand, Timoshenko beam theory extends the Euler-Bernoulli beam theory to the shear-flexible regime by allowing for transverse shear deformations. In addition, torsion-free variants of Simo-Reissner and Kirchhoff-Love beam theories assume zero torsion deformation in spatial beams [101, 102] whereas the general Simo-Reissner beam theory [50], Bauer et al. [52] and Meier et al. [51] allow the torsional deformations.

  Also, a set of beam theories exists that can be identified as kinematically reduced models. Reduced beam kinematics restrict the deformation space of the beam body by removing one plane of deformations. Such beam models can only deform in the 2D space rather than in the full 3D space. Hence torsional deformations vanish naturally, and there will be only one mode of bending. This type of beam models can be found at Nadler et al. [18] and Warren [103].

  Another class of beam theories considers nonlinear material models in beam formulations. Most of the aforementioned beam theories simplify the constitutive models by assuming an isotropic linear elastic or small strain Saint Venant-Kirchhoff hyperelastic material models [50–52]. However, the research works by Orzechowski et al. [104], Maqueda et al. [105] and Hawileh et al. [106] utilise nonlinear material models in their beam simulations. For instance, Neo-Hookean and Mooney-Rivlin materials are two well-established and widely used nonlinear material models in the computational mechanics community [104].

The categorisation based on the underlined theory, geometric considerations and kinematic assumptions lead to a better understanding of the scope of beam theories. However, as expected, there are overlaps between the three categories. Therefore, each beam theory has its own strengths compared to each other. This motivates to derive rigorous formulations for a high-fidelity nonlinear beam theory that produces efficient algorithmic computations for analysis.

3.1.2 Motivation

The level of details and the depth of formulations of the desired beam model comes directly from the problems at hand. In this thesis, we model RVEs of plain-woven and weft-knitted textiles for multiscale modelling and simulations. These two problems demand a set of requirements to be fulfilled in the spatial rod model that we aim to implement. We detail the requirements based on the classification used in Section 3.1.1. The motivation behind implementing a high-fidelity spatial rod model is to capture all the necessary details to simulate the mechanical behaviour of textile RVEs accurately. Thus, we do not intend to include warping effects or complicated material models to strengthen the rod model as there will only be a negligible effect on the final result. Also, such complications can be safely neglected as we have proof from many published articles that experimentally validated real yarns do not display such complex behaviours [1, 18, 20, 52, 107].

Table 3.1 summarises the required settings for the rod model that will be used in RVE simulations, later in Chapter 6. To exploit the advantages of isogeometric analysis [73], we use the same set of basis functions used in geometric modelling to approximate the finite element solution.

Table 3.1: Level of details required by the rod model for multiscale modelling of woven and knitted fabrics

Category	Description
Underlined theory	Nonlinear continuum mechanics with large deformations. Iterative solution scheme: Newton-Raphson method. Compatible with nonlinear contact modelling
Geometric considerations	Initially curved rods. Fibrous materials have arbitrarily large initial curvatures. Isotropic cross-sections. However, a neat extension is proposed to include anisotropic cross-sections.
Kinematic assumptions	Shear-free Kirchhoff-Love kinematics to model textile yarns. Presence of torsional deformations that prevails in fibrous materials. Spatial (full 3D) rod that consists all three displacements and one torsional degree of freedom. Saint Venant-Kirchhoff hyperelastic material model for small strains.

3.2 Finite deformable spatial rods

3.2.1 Kinematics of spatial rods

In the undeformed (reference) configuration, the rod body is characterised by the centreline S and a set of cross-sections attached to S. Under applied loads (or displacements) the rod deforms and adopts a deformed (current) configuration. Deformed configuration is defined by the centreline s and rotated/twisted cross-sections attached to s. Rod centrelines are parameterised by the convective coordinate θ^1 and any material point in the rod body, in either configuration, is defined using the centreline and convective parameters θ^2, θ^3 which span the cross-section of the rod. In this chapter, any uppercase vector quantity belongs to the undeformed configuration, whereas a lowercase vector quantity applies to the deformed configuration. In addition, commonly used notations and definitions are tabulated in Table 3.2.

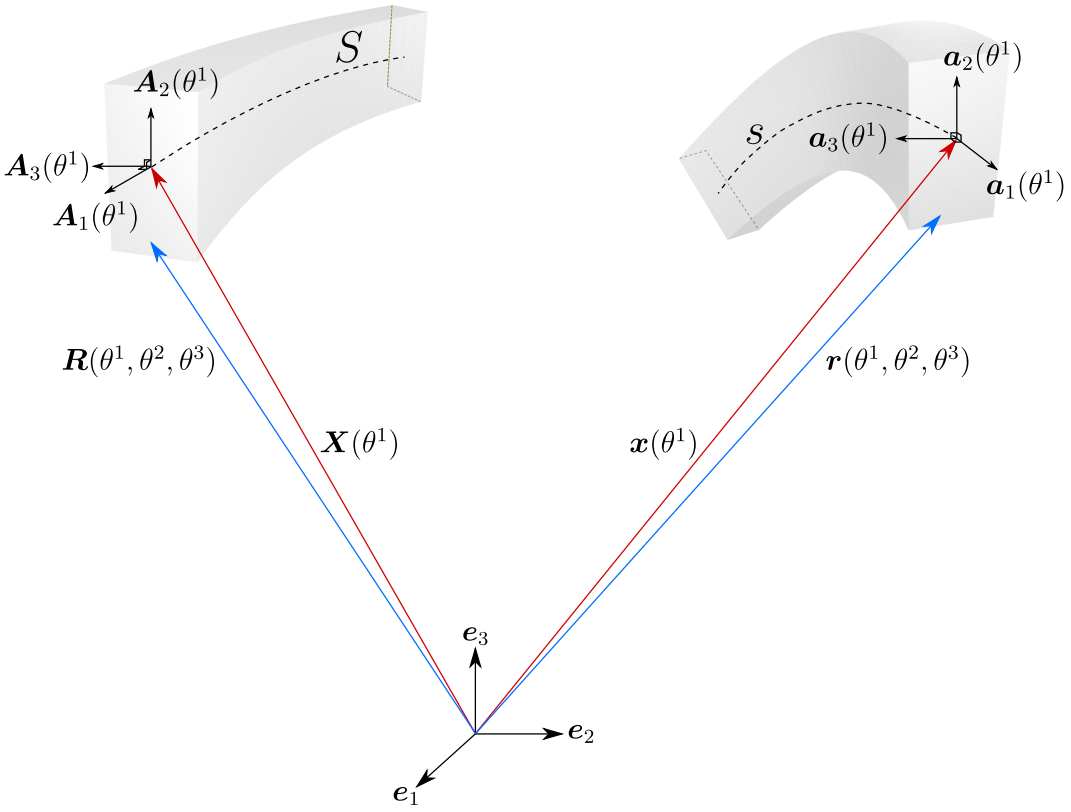


Figure 3.1: Geometric description of a spatial rod in the undeformed (left) and deformed (right) configurations. A rectangular cross-section is considered here for illustration purpose.

Table 3.2: Notations and definitions

Greek indices	take values $2, 3$
Latin indices	take values $1, 2, 3$
$\bullet_{,i}$	comma denotes differentiation w.r.t. θ^i
$\boldsymbol{g}_i = \frac{\partial \boldsymbol{r}}{\partial \theta^i}$	covariant base vectors
$\boldsymbol{g}^i = \frac{\partial \theta^i}{\partial \boldsymbol{r}}$	contravariant base vectors
$\boldsymbol{a}_i = \frac{\partial \boldsymbol{x}}{\partial \theta^i}$	covariant base vectors for $\theta^2 = \theta^3 = 0$
$\boldsymbol{a}^i = \frac{\partial \theta^i}{\partial \boldsymbol{x}}$	contravariant base vectors for $\theta^2 = \theta^3 = 0$
$g_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j$	components of covariant metric tensor
$g^{ij} = \boldsymbol{g}^i \cdot \boldsymbol{g}^j$	components of contravariant metric tensor

The position vectors of a material point in the undeformed and deformed configurations

can be written as,

$$\boldsymbol{R}(\theta^1, \theta^2, \theta^3) = \boldsymbol{X}(\theta^1) + \theta^2 \boldsymbol{A}_2(\theta^1) + \theta^3 \boldsymbol{A}_3(\theta^1) \tag{3.1}$$

$$\boldsymbol{r}(\theta^1, \theta^2, \theta^3) = \boldsymbol{x}(\theta^1) + \theta^2 \boldsymbol{a}_2(\theta^1) + \theta^3 \boldsymbol{a}_3(\theta^1) \ . \tag{3.2}$$

where $\theta^2 = [-h/2, h/2]$ and $\theta^3 = [-b/2, b/2]$. Here h, b indicate the *h*eight and *b*readth of the rectangular cross-section of the rod shown in Figure 3.1. Moreover, $\boldsymbol{X}(\theta^1)$ and $\boldsymbol{x}(\theta^1)$ are the centreline position vectors and $\boldsymbol{A}_i$, $\boldsymbol{a}_i$ form the orthogonal base vectors aligned with the moving vector triads along the centrelines. To reduce the notational overload, vector parameterisation is dropped from quantities in the subsequent formulations. For instance, $\boldsymbol{X}$ is used instead of $\boldsymbol{X}(\theta^1)$.

Kinematic relation between the two configurations is given with the aid of the displacement vector $\boldsymbol{u}$ as,

$$\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{u} \ . \tag{3.3}$$

Using the notations and definitions in Table 3.2 we write the deformation gradient of the rod as,

$$\boldsymbol{F} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{R}} = \frac{\partial \boldsymbol{r}}{\partial \theta^i} \frac{\partial \theta^i}{\partial \boldsymbol{R}} = \boldsymbol{g}_i \otimes \boldsymbol{G}^i \ . \tag{3.4}$$

Next, we derive the Green–Lagrange strain tensor of the rod. To this end, first recall the definition of the Green–Lagrange strain tensor of a solid as,

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} - \mathbf{I}) = \frac{1}{2}(g_{ij} - G_{ij})\boldsymbol{G}^i \otimes \boldsymbol{G}^j = E_{ij}\boldsymbol{G}^i \otimes \boldsymbol{G}^j \ . \tag{3.5}$$

The covariant base vectors $\boldsymbol{G}_i$ and $\boldsymbol{g}_i$ take the forms,

$$\boldsymbol{G}_1 = \frac{\partial \boldsymbol{R}}{\partial \theta^1} = \boldsymbol{A}_1 + \theta^2 \boldsymbol{A}_{2,1} + \theta^3 \boldsymbol{A}_{3,1} \tag{3.6a}$$

$$\boldsymbol{g}_1 = \frac{\partial \boldsymbol{r}}{\partial \theta^1} = \boldsymbol{a}_1 + \theta^2 \boldsymbol{a}_{2,1} + \theta^3 \boldsymbol{a}_{3,1} \tag{3.6b}$$

$$\boldsymbol{G}_2 = \boldsymbol{A}_2 \tag{3.6c}$$

$$\boldsymbol{g}_2 = \boldsymbol{a}_2 \tag{3.6d}$$

$$\boldsymbol{G}_3 = \boldsymbol{A}_3 \tag{3.6e}$$

$$\boldsymbol{g}_3 = \boldsymbol{a}_3 \ . \tag{3.6f}$$

Now we define the moving vector triads along the centrelines as,

$$\boldsymbol{A}_1 = \frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}\theta^1} \tag{3.7}$$

$$\boldsymbol{a}_1 = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\theta^1} \tag{3.8}$$

$$\boldsymbol{a}_\alpha = \Lambda \boldsymbol{A}_\alpha \ , \tag{3.9}$$

where Λ denotes a mapping of the reference vectors $\boldsymbol{A}_\alpha$ onto the corresponding deformed vectors $\boldsymbol{a}_\alpha$. For later use, normalised tangent vectors of the centrelines are defined as,

$$\boldsymbol{A}_{1u} = \frac{\boldsymbol{A}_1}{||\boldsymbol{A}_1||} \qquad \boldsymbol{a}_{1u} = \frac{\boldsymbol{a}_1}{||\boldsymbol{a}_1||} \ . \tag{3.10}$$

For shear-free Euler-Bernoulli rods we can express the orthonormal vector triad relationships as,

$$\boldsymbol{A}_3 = \boldsymbol{A}_{1u} \times \boldsymbol{A}_2 \qquad \boldsymbol{a}_3 = \boldsymbol{a}_{1u} \times \boldsymbol{a}_2 \tag{3.11a}$$

$$\boldsymbol{A}_i \cdot \boldsymbol{A}_\beta = \delta_{i\beta} \qquad \boldsymbol{a}_i \cdot \boldsymbol{a}_\beta = \delta_{i\beta} \ , \tag{3.11b}$$

where $\delta_{i\beta}$ is the standard Kronecker-delta operator.

Base vectors $\boldsymbol{A}_\alpha$ and $\boldsymbol{a}_\alpha$ are chosen as unit vectors and are aligned with the principal directions of the cross-section (i.e. the Eigen directions of the second moment of area).

Next, we consider the rotation of the convective triad from the undeformed to the deformed configuration. We make a Bernoulli type assumption on the shear deformations so that the orthogonal triad $\boldsymbol{a}_i$ can be mapped from the orthogonal triad $\boldsymbol{A}_i$ using a 3D rotation $\boldsymbol{\Lambda}$ as given in Equation (3.9). This 3D rotation is composed by two distinct rotations $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$. This 3D rotation is illustrated in Figure 3.2.

1. $\boldsymbol{\Lambda}_1$ - Rotation about $\boldsymbol{A}_1$, which resembles a torsion of the cross-section of an angle ϑ, and takes the cross-section into an intermediate configuration denoted by the triad $\boldsymbol{a}_i'$. This angle ϑ is the *only* independent variable in this rotation and therefore, it naturally becomes a degree of freedom at a given cross-section of the rod. So we can write $\boldsymbol{\Lambda}_1 = \boldsymbol{\Lambda}_1(\vartheta)$.

2. $\boldsymbol{\Lambda}_2$ - Intermediate configuration is now rotated from $\boldsymbol{a}_1'(\equiv \boldsymbol{A}_1)$ to $\boldsymbol{a}_1$ to complete the 3D rotation. $\boldsymbol{\Lambda}_2$ involves a rotation in the rod centreline and hence, it is

dependent *only* on the deformed position of the rod centreline ($\boldsymbol{x}$). Thus, we can write $\boldsymbol{\Lambda}_2 = \boldsymbol{\Lambda}_2(\boldsymbol{u})$.

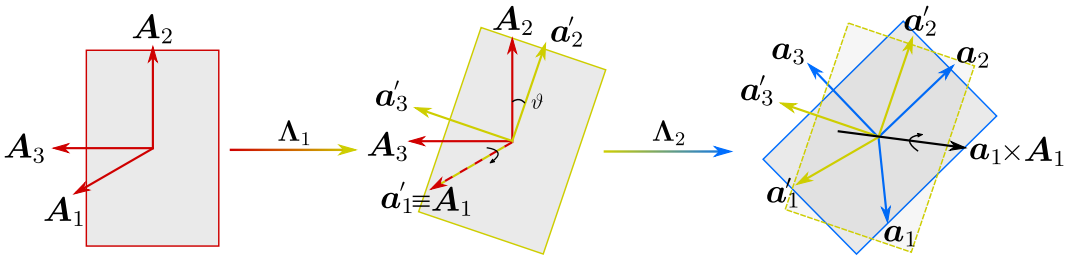


Figure 3.2: Decomposition of the 3D rotation of the cross-section. The transformation from the undeformed to deformed configurations is accomplished using an intermediate step.

The complete 3D rotation can now be written as,

$$\boldsymbol{\Lambda}(\boldsymbol{u}, \vartheta) = \boldsymbol{\Lambda}_2(\boldsymbol{u})\boldsymbol{\Lambda}_1(\vartheta) \ . \tag{3.12}$$

Substituting Equation (3.12) in Equation (3.9) we arrive at,

$$\boldsymbol{a}_\alpha = \boldsymbol{\Lambda}_2(\boldsymbol{u}) \underbrace{\boldsymbol{\Lambda}_1(\vartheta)\boldsymbol{A}_\alpha}_{\boldsymbol{a}'_\alpha} \ . \tag{3.13}$$

Rodrigues' rotation formula

Rodrigues' rotation formula is of paramount importance when formulations involve a vector rotated by a known angle around a known axis. Rodrigues' rotation formula has been widely used in a plethora of computational research works [50, 89, 91, 108, 109]. It has the form,

$$\boldsymbol{v}_{\text{rot}} = \boldsymbol{v}\cos\omega + (\boldsymbol{k} \times \boldsymbol{v})\sin\omega + \boldsymbol{k}\ (\boldsymbol{k} \cdot \boldsymbol{v})(1 - \cos\omega) \ , \tag{3.14}$$

where a vector $\boldsymbol{v}$ is rotated by an angle ω around the axis given by a vector $\boldsymbol{k}$ to arrive at the rotated vector $\boldsymbol{v}_{\text{rot}}$. The same formula can take another form,

$$\boldsymbol{v}_{\text{rot}} = \boldsymbol{\Lambda}_{\text{rot}}\boldsymbol{v} \ , \tag{3.15}$$

where

$$\mathbf{\Lambda}_{\text{rot}} = \mathbf{I} + \sin\omega\mathbf{K} + (1 - \cos\omega)\mathbf{K}^2 \ , \tag{3.16}$$

with $\mathbf{K} = \mathbf{K}(\boldsymbol{k})$.

We use Rodrigues' rotation formula for the rotation $\mathbf{\Lambda}_1$ where rotation axis is given by the undeformed normalised tangent vector $\boldsymbol{A}_{1u}$ and angle of rotation is ϑ, as illustrated in Figure 3.2. Subsequently, we use the smallest rotation formula [51, 81, 110] for the rotation $\mathbf{\Lambda}_2$. These rotations take forms,

$$\mathbf{\Lambda}_1(\vartheta) = \mathbf{I} + \sin\vartheta\mathbf{K}(\boldsymbol{A}_{1u}) + (1 - \cos\vartheta)\mathbf{K}(\boldsymbol{A}_{1u})\mathbf{K}(\boldsymbol{A}_{1u}) \ , \tag{3.17}$$

$$\mathbf{\Lambda}_2(\boldsymbol{u}) = \mathbf{I} - \frac{(\boldsymbol{A}_{1u} + \boldsymbol{a}_{1u})\boldsymbol{a}_{1u}^{\mathrm{T}}}{1 + \boldsymbol{a}_{1u} \cdot \boldsymbol{A}_{1u}} = \mathbf{I} - \frac{(\boldsymbol{A}_{1u} + \boldsymbol{a}_{1u}) \otimes \boldsymbol{a}_{1u}}{1 + \boldsymbol{a}_{1u} \cdot \boldsymbol{A}_{1u}} \ , \tag{3.18}$$

where

$$\mathbf{K}(\boldsymbol{A}_{1u}) = \begin{bmatrix} 0 & -A_{1u}^z & A_{1u}^y \\ A_{1u}^z & 0 & -A_{1u}^x \\ -A_{1u}^y & A_{1u}^x & 0 \end{bmatrix} \qquad \text{with} \qquad \boldsymbol{A}_{1u} = \begin{bmatrix} A_{1u}^x \\ A_{1u}^y \\ A_{1u}^z \end{bmatrix} \ . \tag{3.19}$$

In the case of $\boldsymbol{A}_{1u} = \boldsymbol{a}'_{1u} = \boldsymbol{a}_{1u}$, or in other words when the rod is in its reference configuration, a special note is warranted for Equation (3.13). Obviously $\mathbf{\Lambda}_1 = \mathbf{I}$ in (3.17) and $\mathbf{\Lambda}_2 = \mathbf{I} - \boldsymbol{a}_{1u} \otimes \boldsymbol{a}_{1u}$. Moreover, after some algebraic simplifications, we get the expected $\boldsymbol{a}_\alpha = \boldsymbol{A}_\alpha$ using the definition of the tensor product $(\boldsymbol{A}_{1u} \otimes \boldsymbol{A}_{1u})\boldsymbol{A}_\alpha = (\boldsymbol{A}_\alpha \cdot \boldsymbol{A}_{1u})\boldsymbol{A}_{1u} = 0$. For later use, we differentiate the rotation formulae given in Equations (3.17) and (3.18) and write the expressions as,

$$\mathbf{\Lambda}_{1,1} = \sin\vartheta\mathbf{K}_{,1} + \cos\vartheta\mathbf{K}\vartheta_{,1} + (\mathbf{K}\mathbf{K}_{,1} + \mathbf{K}_{,1}\mathbf{K})(1 - \cos\vartheta) + \sin\vartheta\mathbf{K}\mathbf{K}\vartheta_{,1} \tag{3.20}$$

where

$$\mathbf{K}_{,1} = \mathbf{K}(\boldsymbol{A}_{1u,1}) \tag{3.21}$$

$$\boldsymbol{A}_{1u,1} = \frac{\boldsymbol{A}_{1,1}}{||\boldsymbol{A}_1||} - \boldsymbol{A}_1\frac{(\boldsymbol{A}_1 \cdot \boldsymbol{A}_{1,1})}{||\boldsymbol{A}_1||^3} \ , \tag{3.22}$$

and

$$\boldsymbol{\Lambda}_{2,1} = -\frac{(\boldsymbol{A}_{1u,1}\otimes\boldsymbol{a}_{1u}+\boldsymbol{A}_{1u}\otimes\boldsymbol{a}_{1u,1}+\boldsymbol{a}_{1u}\otimes\boldsymbol{a}_{1u,1}+\boldsymbol{a}_{1u,1}\otimes\boldsymbol{a}_{1u})}{1+\boldsymbol{a}_{1u}\cdot\boldsymbol{A}_{1u}} + \frac{((\boldsymbol{A}_{1u}+\boldsymbol{a}_{1u})\otimes\boldsymbol{a}_{1u})(\boldsymbol{a}_{1u,1}\cdot\boldsymbol{A}_{1u}+\boldsymbol{A}_{1u,1}\cdot\boldsymbol{a}_{1u})}{(1+\boldsymbol{a}_{1u}\cdot\boldsymbol{A}_{1u})^2} \ . \tag{3.23}$$

It is important to precisely obtain the base vectors $\boldsymbol{A}_i$ of the undeformed geometry. This information is vital when the geometry is in full 3D space and has anisotropic cross-sections (i.e. rectangular, triangular, etc.). This is achieved, again, using the Rodrigues' formula by moving the base vectors along the rod starting from a point where base vectors are exactly known [52]. It can also be achieved by first using the Frenet-Serret formulas [51, 80, 81, 111] and then rotating the base vectors along the principal direction of the cross-sections [51, 95]. However in literature, fabric yarns are modelled as rods with circular cross-sections [1, 17–20, 112–115] which allows us to use the Frenet-Serret formulas directly due to the isotropy of the yarn cross-section.

Frenet–Serret formulae in differential geometry

The orthonormal **TNB** frame of a differentiable space curve $\boldsymbol{r}$ is defined on the parametric domain θ^1 as,

$$\mathbf{T} = \frac{\boldsymbol{r}_{,1}}{||\boldsymbol{r}_{,1}||} \tag{3.24}$$

$$\mathbf{N} = \frac{\mathbf{T}_{,\mathbf{1}}}{||\mathbf{T}_{,\mathbf{1}}||} = \frac{\boldsymbol{r}_{,1}\times(\boldsymbol{r}_{,11}\times\boldsymbol{r}_{,1})}{||\boldsymbol{r}_{,1}||\ ||\boldsymbol{r}_{,11}\times\boldsymbol{r}_{,1}||} \tag{3.25}$$

$$\mathbf{B} = \mathbf{T}\times\mathbf{N} = \frac{\boldsymbol{r}_{,1}\times\boldsymbol{r}_{,11}}{||\boldsymbol{r}_{,1}\times\boldsymbol{r}_{,11}||} \ . \tag{3.26}$$

Here, the vector $\mathbf{T}$ is identified as the normalised tangent vector along $\boldsymbol{r}$. In addition, normal ($\mathbf{N}$) and bi-normal ($\mathbf{B}$) vectors complete the vector triad at any given point on $\boldsymbol{r}$. Note that there can be several combinations of $\mathbf{N}$ and $\mathbf{B}$ to form the vector triad with the uniquely defined $\mathbf{T}$. However, above equations yield one combination of them. The parametric derivative of the **TNB** frame takes the form,

$$\begin{bmatrix}\mathbf{T}_{,\mathbf{1}}\\ \mathbf{N}_{,\mathbf{1}}\\ \mathbf{B}_{,\mathbf{1}}\end{bmatrix} = ||\boldsymbol{r}_{,1}|| \begin{bmatrix}0 & \hat{\kappa} & 0\\ -\hat{\kappa} & 0 & \tau\\ 0 & -\tau & 0\end{bmatrix} \begin{bmatrix}\mathbf{T}\\ \mathbf{N}\\ \mathbf{B}\end{bmatrix} \ , \tag{3.27}$$

where

$$\hat{\kappa} = \frac{||\boldsymbol{r}_{,1} \times \boldsymbol{r}_{,11}||}{||\boldsymbol{r}_{,1}||^3} \qquad \text{and} \qquad \tau = \frac{\det[\boldsymbol{r}_{,1}, \boldsymbol{r}_{,11}, \boldsymbol{r}_{,111}]}{||\boldsymbol{r}_{,1} \times \boldsymbol{r}_{,11}||^2} \ . \tag{3.28}$$

For planar curves, τ in Equation (3.28) vanishes. However, for general space curves, τ exists and demands for the existence of $\boldsymbol{r}_{,111}$. For higher accuracy and better convergence, the existence of $\boldsymbol{r}_{,111}$ enforces a stringent requirement for the basis functions to possess minimum C^3 continuity everywhere in the problem domain. This will be explained in detail in Section 3.2.2.

Now we can explicitly express the strain components in the Green-Lagrange strain tensor. To this end, we substitute the covariant base vectors in Equation (3.6) into Equation (3.5) and using the definitions in Table 3.2 we write,

$$E_{ij} = \frac{1}{2}(g_{ij} - G_{ij}) = \frac{1}{2}(\boldsymbol{g}_i \cdot \boldsymbol{g}_j - \boldsymbol{G}_i \cdot \boldsymbol{G}_j).$$

Simplifying for E_{11} and neglecting the higher orders of θ^α we get,

$$\begin{aligned} E_{11} &= \frac{1}{2}(\boldsymbol{g}_1 \cdot \boldsymbol{g}_1 - \boldsymbol{G}_1 \cdot \boldsymbol{G}_1) \\ &= \frac{1}{2}((\boldsymbol{a}_1 + \theta^2 \boldsymbol{a}_{2,1} + \theta^3 \boldsymbol{a}_{3,1}) \cdot (\boldsymbol{a}_1 + \theta^2 \boldsymbol{a}_{2,1} + \theta^3 \boldsymbol{a}_{3,1}) \\ &- (\boldsymbol{A}_1 + \theta^2 \boldsymbol{A}_{2,1} + \theta^3 \boldsymbol{A}_{3,1}) \cdot (\boldsymbol{A}_1 + \theta^2 \boldsymbol{A}_{2,1} + \theta^3 \boldsymbol{A}_{3,1})) \\ &= \frac{1}{2}(\boldsymbol{a}_1 \cdot \boldsymbol{a}_1 - \boldsymbol{A}_1 \cdot \boldsymbol{A}_1) + \theta^2(\boldsymbol{A}_2 \cdot \boldsymbol{A}_{1,1} - \boldsymbol{a}_2 \cdot \boldsymbol{a}_{1,1}) + \theta^3(\boldsymbol{A}_3 \cdot \boldsymbol{A}_{1,1} - \boldsymbol{a}_3 \cdot \boldsymbol{a}_{1,1}). \end{aligned} \tag{3.29}$$

Following the same steps for other terms of E_{ij} we arrive at,

$$E_{ij} = \begin{bmatrix} E_{11} = \alpha + \theta^2\beta_2 + \theta^3\beta_3 & E_{12} = \theta^3\gamma & E_{13} = -\theta^2\gamma \\ & E_{22} = 0 & E_{23} = 0 \\ \text{symmetric} & & E_{33} = 0 \end{bmatrix} \tag{3.30}$$

where $\alpha(\boldsymbol{u})$, $\beta_2(\boldsymbol{u}, \vartheta)$, $\beta_3(\boldsymbol{u}, \vartheta)$ and $\gamma(\boldsymbol{u}, \vartheta)$ are membrane, bending (about $\boldsymbol{a}_2$), bending (about $\boldsymbol{a}_3$) and torsional shear strains, respectively. Notice the interaction between $\boldsymbol{u}$ and ϑ influences the bending and shear strains, but not membrane strains. This is easily understood because membrane strains are only governed by the rod's axial elongations, but not the rotations. Moreover, assumptions from the Bernoulli beam theory set E_{23}, E_{32} and E_{33} to zero and shear strains E_{12} and E_{21} are induced only from

torsional moments.

$$\alpha(\boldsymbol{u}) = \frac{1}{2}(\boldsymbol{a}_1 \cdot \boldsymbol{a}_1 - \boldsymbol{A}_1 \cdot \boldsymbol{A}_1) \tag{3.31a}$$

$$\beta_2(\boldsymbol{u}, \vartheta) = \boldsymbol{A}_2 \cdot \boldsymbol{A}_{1,1} - \boldsymbol{a}_2 \cdot \boldsymbol{a}_{1,1} \tag{3.31b}$$

$$\beta_3(\boldsymbol{u}, \vartheta) = \boldsymbol{A}_3 \cdot \boldsymbol{A}_{1,1} - \boldsymbol{a}_3 \cdot \boldsymbol{a}_{1,1} \tag{3.31c}$$

$$\gamma(\boldsymbol{u}, \vartheta) = \frac{1}{2}(\boldsymbol{a}_2 \cdot \boldsymbol{a}_{3,1} - \boldsymbol{A}_2 \cdot \boldsymbol{A}_{3,1}) \tag{3.31d}$$

Following orthogonality identities have been used in deriving the strain components as given in Equation (3.31).

$$\boldsymbol{a}_1 \cdot \boldsymbol{a}_{2,1} = -\boldsymbol{a}_{1,1} \cdot \boldsymbol{a}_2 \qquad \boldsymbol{a}_3 \cdot \boldsymbol{a}_{2,1} = -\boldsymbol{a}_{3,1} \cdot \boldsymbol{a}_2 \qquad \boldsymbol{a}_1 \cdot \boldsymbol{a}_{3,1} = -\boldsymbol{a}_{1,1} \cdot \boldsymbol{a}_3. \tag{3.32}$$

It is straightforward to derive expressions in Equation (3.32) by differentiating the orthogonality conditions. For instance, we know $\boldsymbol{a}_1 \cdot \boldsymbol{a}_2 = 0$, and differentiation with respect to θ^1 yields $\boldsymbol{a}_1 \cdot \boldsymbol{a}_{2,1} + \boldsymbol{a}_{1,1} \cdot \boldsymbol{a}_2 = 0$.

3.2.2 Discretisation of the energy functional

We consider the total potential energy of a hyperelastic rod to derive the rod equilibrium equations in the weak form as,

$$\Pi(\boldsymbol{x}) = \Pi_{\text{int}}(\boldsymbol{x}) + \Pi_{\text{ext}}(\boldsymbol{x}) \tag{3.33}$$

$$= \int_{\text{V}} W(\alpha, \beta_2, \beta_3, \gamma)\text{dV} + \Pi_{\text{ext}}(\boldsymbol{x}) \ , \tag{3.34}$$

where W is the strain energy density. For hyperelastic materials, strain energy density is a function of the Green-Lagrange strain tensor. Hence, using strain quantities in Equation (3.31), we write $W = W(\boldsymbol{E}) = W(\alpha, \beta_2, \beta_3, \gamma)$. Moreover, we introduce the second Piola-Kirchhoff stress tensor which is defined for a hyperelastic material [116] as,

$$\boldsymbol{S} = \frac{\partial W(\alpha, \beta_2, \beta_3, \gamma)}{\partial \boldsymbol{E}} \ . \tag{3.35}$$

Second Piola-Kirchhoff stress tensor in Equation (3.35) together with its energy conjugate Green-Lagrange strain tensor defines the strain energy density of a hyperelastic material as,

$$W(\alpha, \beta_2, \beta_3, \gamma) = \boldsymbol{S} : \boldsymbol{E} \ . \tag{3.36}$$

At equilibrium, the first variation of the total potential energy is zero. This resembles a nonlinear convex optimisation problem to be equilibrated at the minimum potential energy [117, 118]. Equilibrium condition for the system in Equation (3.33) is derived using the principle of virtual work as,

$$\delta\Pi(\boldsymbol{x}) = \delta\Pi_{\text{int}}(\boldsymbol{x}) + \delta\Pi_{\text{ext}}(\boldsymbol{x}) = 0 \ . \tag{3.37}$$

Using the relation $\delta(\bullet) = \dfrac{\partial(\bullet)}{\partial(*)}\delta(*) + \dfrac{\partial(\bullet)}{\partial(\star)}\delta(\star) + \ \ldots$ we simplify Equation (3.37) as

$$\delta\Pi(\boldsymbol{x}) = \underbrace{\int_{\text{V}} \left(\frac{\partial W}{\partial \boldsymbol{u}}\right) \text{dV}}_{\boldsymbol{f}_u} \cdot \delta\boldsymbol{u} + \underbrace{\int_{\text{V}} \left(\frac{\partial W}{\partial \vartheta}\right) \text{dV}}_{f_\vartheta} \cdot \delta\vartheta + \delta\Pi_{\text{ext}}(\boldsymbol{x}) = 0 \ . \tag{3.38}$$

where internal forces $\boldsymbol{f}_u$ and f_ϑ are defined for arbitrary perturbations $\delta\boldsymbol{u}$ and $\delta\vartheta$, respectively.

The next task is to derive closed form expressions to compute the integrals in Equation (3.38). At this juncture, attention is guided towards simplifying the integral computations using parametric volume mapping.

Volume integration of a quantity $(\bullet)$ in the convected parametric domain θ^i takes the form,

$$\int_{\text{V}} (\bullet)\ \text{dV} = \int_b \int_h \int_S (\bullet)\sqrt{G}\ \text{d}\theta^1 \text{d}\theta^2 \text{d}\theta^3 \ , \tag{3.39}$$

where $\sqrt{G}$ is the Jacobian derived as,

$$\begin{aligned}
\sqrt{G} &= \det[\boldsymbol{G}_1 \ \ \boldsymbol{G}_2 \ \ \boldsymbol{G}_3] \\
&= \boldsymbol{G}_1 \cdot (\boldsymbol{G}_2 \times \boldsymbol{G}_3) \\
&= (\boldsymbol{A}_1 + \theta^2 \boldsymbol{A}_{2,1} + \theta^3 \boldsymbol{A}_{3,1}) \cdot (\boldsymbol{A}_2 \times \boldsymbol{A}_3) \\
&= (\boldsymbol{A}_1 + \theta^2 \boldsymbol{A}_{2,1} + \theta^3 \boldsymbol{A}_{3,1}) \cdot \frac{\boldsymbol{A}_1}{||\boldsymbol{A}_1||} \\
&= ||\boldsymbol{A}_1|| + \theta^2 \frac{\boldsymbol{A}_{2,1} \cdot \boldsymbol{A}_1}{||\boldsymbol{A}_1||} + \theta^3 \frac{\boldsymbol{A}_{3,1} \cdot \boldsymbol{A}_1}{||\boldsymbol{A}_1||} \\
\sqrt{G} &= ||\boldsymbol{A}_1|| - \theta^2 \frac{\boldsymbol{A}_{1,1} \cdot \boldsymbol{A}_2}{||\boldsymbol{A}_1||} - \theta^3 \frac{\boldsymbol{A}_{1,1} \cdot \boldsymbol{A}_3}{||\boldsymbol{A}_1||} \ .
\end{aligned} \tag{3.40}$$

The Euler-Bernoulli beam theory assumes thin beam cross-sections with vanishing shear strains, whereby the beam height remains unchanged during deformation. This assump-

tion leads to analytically tractable integrals with respect to θ^α. Hence, we can use the metric $\sqrt{G}$ to express the current infinitesimal arc-length $\mathrm{d}s$ as,

$$\begin{aligned}
\mathrm{d}s &= \sqrt{G}\mathrm{d}\theta^1 \\
&= \left(||\boldsymbol{A}_1|| - \theta^2 \frac{\boldsymbol{A}_{1,1}\cdot\boldsymbol{A}_2}{||\boldsymbol{A}_1||} - \theta^3 \frac{\boldsymbol{A}_{1,1}\cdot\boldsymbol{A}_3}{||\boldsymbol{A}_1||} \right) \mathrm{d}\theta^1 \qquad (3.41) \\
&= \underbrace{\left(1 - \theta^2 \underbrace{\frac{\boldsymbol{A}_{1,1}\cdot\boldsymbol{A}_2}{\boldsymbol{A}_1\cdot\boldsymbol{A}_1}}_{\kappa_2} - \theta^3 \underbrace{\frac{\boldsymbol{A}_{1,1}\cdot\boldsymbol{A}_3}{\boldsymbol{A}_1\cdot\boldsymbol{A}_1}}_{\kappa_3} \right)}_{\mu} \underbrace{||\boldsymbol{A}_1||\mathrm{d}\theta^1}_{\mathrm{d}S} \qquad (3.42) \\
\mathrm{d}s &= (1 - \kappa_2\theta^2 - \kappa_3\theta^3) = \eta \mathrm{d}S \ . \qquad (3.43)
\end{aligned}$$

where κ_2 and κ_3 are the initial curvatures of the rod in two principal directions of the cross-section. This reveals an important computational aspect of spatial rods because the accuracy of internal forces rely heavily on these initial curvatures [52]. It is evident that, for initially straight rods, $\boldsymbol{A}_{1,1}$ remains parallel to $\boldsymbol{A}_1$ (thus orthogonal to $\boldsymbol{A}_\alpha$) and therefore has vanishing initial curvatures due to the orthogonality of the vector triad $\boldsymbol{A}_i$. For initially curved planar rods, one of the curvatures will be zero, simply due to one non-existent $\boldsymbol{A}_{\alpha,1}$. Readers are reminded of the orthogonality identities given in Equation (3.32).

Rod internal forces

Using Equations (3.36) and (3.38) now we write the internal forces as,

$$\boldsymbol{f_u} = \int_{\mathrm{V}} \boldsymbol{S} : \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{u}} \mathrm{dV} \ , \qquad (3.44)$$

$$f_\vartheta = \int_{\mathrm{V}} \boldsymbol{S} : \frac{\partial \boldsymbol{E}}{\partial \vartheta} \mathrm{dV} \ . \qquad (3.45)$$

Next, the assumptions in Euler-Bernoulli beam theory and volume integral simplifications in Equations (3.40) and (3.43) are used to simplify the internal forces as,

$$\boldsymbol{f_u} = \int_b \int_h \int_S \left(S^{11} \frac{\partial E_{11}}{\partial \boldsymbol{u}} + 2S^{12} \frac{\partial E_{12}}{\partial \boldsymbol{u}} + 2S^{13} \frac{\partial E_{13}}{\partial \boldsymbol{u}} \right) \eta \mathrm{d}\theta^2 \mathrm{d}\theta^3 \mathrm{d}S \ , \qquad (3.46)$$

$$f_\vartheta = \int_b \int_h \int_S \left(S^{11} \frac{\partial E_{11}}{\partial \vartheta} + 2S^{12} \frac{\partial E_{12}}{\partial \vartheta} + 2S^{13} \frac{\partial E_{13}}{\partial \vartheta} \right) \eta \mathrm{d}\theta^2 \mathrm{d}\theta^3 \mathrm{d}S \ . \qquad (3.47)$$

We assume that the strains are small so that we can use the St. Venant–Kirchhoff constitutive equation [119] given by,

$$\boldsymbol{S} = \boldsymbol{\mathcal{C}} : \boldsymbol{E} \ . \tag{3.48}$$

Next, we express $\boldsymbol{E}$, $\boldsymbol{S}$ and $\boldsymbol{\mathcal{C}}$ defined in the convected coordinates as,

$$\boldsymbol{E} = E_{ij}\boldsymbol{G}^i \otimes \boldsymbol{G}^j \tag{3.49}$$

$$\boldsymbol{S} = S^{ij}\boldsymbol{G}_i \otimes \boldsymbol{G}_j \tag{3.50}$$

$$\boldsymbol{\mathcal{C}} = \mathcal{C}^{ijkl}\boldsymbol{G}_i \otimes \boldsymbol{G}_j \otimes \boldsymbol{G}_k \otimes \boldsymbol{G}_l \ . \tag{3.51}$$

Now the non-zero second Piola-Kirchhoff stress components can be written as,

$$S^{11} = \mathcal{C}^{1111}E_{11} \tag{3.52a}$$

$$S^{12} = \mathcal{C}^{1212}E_{12} \tag{3.52b}$$

$$S^{13} = \mathcal{C}^{1313}E_{13} \ . \tag{3.52c}$$

It is highlighted, for instance, that $\mathcal{C}^{1111}$ in Equation (3.52a) is defined in the convective coordinates as given in Equation (3.51). Hence, a closed form for $\mathcal{C}^{ijkl}$ in Equation (3.51) is derived as,

$$\mathcal{C}^{ijkl} = (\boldsymbol{G}^i \otimes \boldsymbol{G}^j) : \boldsymbol{\mathcal{C}} : (\boldsymbol{G}^k \otimes \boldsymbol{G}^l) \ . \tag{3.53}$$

After some algebraic simplifications using definitions in Table 3.2 we arrive at [42],

$$\mathcal{C}^{ijkl} = \lambda G^{ij}G^{kl} + \mu(G^{ik}G^{jl} + G^{il}G^{jk}) \ . \tag{3.54}$$

where λ and μ are the Lamé parameters defined for an isotropic material. Now we substitute for stress components in Equations (3.52) and replace them in Equations (3.46) and (3.47). Exploiting the symmetry of the fourth-order material tensor $\boldsymbol{\mathcal{C}}$, we now use shortened indices, for instance $\mathcal{C}^{1212}$ as $\mathcal{C}^{12}$.

$$\boldsymbol{f_u} = \int_b \int_h \int_S \left(\mathcal{C}^{11}E_{11}\frac{\partial E_{11}}{\partial \boldsymbol{u}} + 2\mathcal{C}^{12}E_{12}\frac{\partial E_{12}}{\partial \boldsymbol{u}} + 2\mathcal{C}^{13}E_{13}\frac{\partial E_{13}}{\partial \boldsymbol{u}} \right) \eta \mathrm{d}\theta^2 \mathrm{d}\theta^3 \mathrm{d}S \tag{3.55}$$

$$f_\vartheta = \int_b \int_h \int_S \left(\mathcal{C}^{11}E_{11}\frac{\partial E_{11}}{\partial \vartheta} + 2\mathcal{C}^{12}E_{12}\frac{\partial E_{12}}{\partial \vartheta} + 2\mathcal{C}^{13}E_{13}\frac{\partial E_{13}}{\partial \vartheta} \right) \eta \mathrm{d}\theta^2 \mathrm{d}\theta^3 \mathrm{d}S \ . \tag{3.56}$$

Now Green-Lagrange strains and their derivatives are computed using Equation (3.31) and substituted in Equations (3.55) and (3.56). At this juncture, to maintain the clarity of this chapter, detailed derivations of internal forces are included in the Appendix A.1.1.

To this end, we write the internal forces in a compact form as,

$$\boldsymbol{f_u} = \int_S \left[\left(n\frac{\partial \alpha}{\partial \boldsymbol{u}} + \sum_{j=2}^{3} m_j \frac{\partial \beta_j}{\partial \boldsymbol{u}} + q\frac{\partial \gamma}{\partial \boldsymbol{u}} \right) - \mathcal{C}^{11} \sum_{j=2}^{3} I_j \left(\beta_j \frac{\partial \alpha}{\partial \boldsymbol{u}} + \alpha \frac{\partial \beta_j}{\partial \boldsymbol{u}} \right) \kappa_j \right] \mathrm{d}S \tag{3.57}$$

$$f_\vartheta = \int_S \left[\left(\sum_{j=2}^{3} m_j \frac{\partial \beta_j}{\partial \vartheta} + q\frac{\partial \gamma}{\partial \vartheta} \right) - \mathcal{C}^{11} \sum_{j=2}^{3} I_j \left(\alpha \frac{\partial \beta_j}{\partial \vartheta} \right) \kappa_j \right] \mathrm{d}S \ . \tag{3.58}$$

where membrane (n), bending (m_ω) and torsion (q) stress resultants take the forms,

$$n = \frac{\partial W^s}{\partial \alpha} = \lambda G^{11} G^{11} A \alpha \tag{3.59a}$$

$$m_\omega = \frac{\partial W^b}{\partial \beta_\omega} = \lambda G^{11} G^{11} I_\omega \beta_\omega \tag{3.59b}$$

$$q = \frac{\partial W^t}{\partial \gamma} = 4\mu G^{11} (I_2 + I_3)\gamma \ . \tag{3.59c}$$

Table 3.3 summarises the geometric and material properties used in Equations (3.59a), (3.59b) and (3.59c). Same properties will be used consistently throughout this chapter.

Table 3.3: Rod geometric and material properties

$A = bh$	cross-section area of the rod
$I_2 = \frac{hb^3}{12}$	second moment area of the rod about $\boldsymbol{A}_2$
$I_3 = \frac{bh^3}{12}$	second moment area of the rod about $\boldsymbol{A}_3$
$I_{\text{polar}} = I_2 + I_3$	polar moment area of the rod about $\boldsymbol{A}_1$
λ	first Lamé parameter of the rod
μ	second Lamé parameter of the rod

Next, we discretise the rod centreline and approximate the unknown displacement ($\boldsymbol{u}$) and torsion (ϑ) functions using shape functions N^I and nodal degrees of freedom. We

write,

$$\boldsymbol{X}(\theta^1) = \sum_{I=1}^{NSF} N^I(\theta^1)\boldsymbol{X}_I \tag{3.60a}$$

$$\boldsymbol{x}(\theta^1) = \sum_{I=1}^{NSF} N^I(\theta^1)\boldsymbol{x}_I \tag{3.60b}$$

$$\boldsymbol{u}(\theta^1) = \sum_{I=1}^{NSF} N^I(\theta^1)\boldsymbol{u}_I \tag{3.60c}$$

$$\vartheta(\theta^1) = \sum_{I=1}^{NSF} N^I(\theta^1)\vartheta_I \ , \tag{3.60d}$$

where NSF denotes the number of shape functions. Differentiation of any quantity in Equation (3.60), with respect to convective parameter θ^1 in the discretised domain is written as,

$$(\bullet)_{,1} = \sum_{I=1}^{NSF} N^I_{,1}(\theta^1)(\bullet)_I \ , \qquad (\bullet)_{,11} = \sum_{I=1}^{NSF} N^I_{,11}(\theta^1)(\bullet)_I \ . \tag{3.61}$$

Next important task in hand is to present explicit expressions for strain gradients to precisely conclude the formulations for internal forces in Equations (3.57) and (3.58). This is accomplished using the discretised domain of the rod as presented in Equation (3.60).

Strain gradients

First we formulate strain gradients with respect to nodal displacement degrees of freedoms $\boldsymbol{u}_I$

$$\frac{\partial \alpha(\boldsymbol{u})}{\partial \boldsymbol{u}_I} = \boldsymbol{a}_1 N^I_{,1} \tag{3.62}$$

$$\frac{\partial \beta_\omega(\boldsymbol{u},\vartheta)}{\partial \boldsymbol{u}_I} = -\frac{\partial \boldsymbol{a}_\omega}{\partial \boldsymbol{u}_I}\cdot \boldsymbol{a}_{1,1} - \boldsymbol{a}_\omega \cdot \frac{\partial \boldsymbol{a}_{1,1}}{\partial \boldsymbol{u}_I} \tag{3.63}$$

$$\frac{\partial \gamma(\boldsymbol{u},\vartheta)}{\partial \boldsymbol{u}_I} = \frac{1}{2}\left(\frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{u}_I}\cdot \boldsymbol{a}_{3,1} + \boldsymbol{a}_2 \cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I}\right) \tag{3.64}$$

where

$$\frac{\partial \boldsymbol{a}_1}{\partial \boldsymbol{u}_I} = N^I_{,1}\mathbf{I} \tag{3.65}$$

$$\frac{\partial \boldsymbol{a}_{1,1}}{\partial \boldsymbol{u}_I} = N^I_{,11}\mathbf{I} \tag{3.66}$$

$$\frac{\partial \boldsymbol{a}_\omega}{\partial \boldsymbol{u}_I} = \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_\omega \tag{3.67}$$

$$\boldsymbol{a}_{\omega,1} = \boldsymbol{\Lambda}_{2,1}\boldsymbol{\Lambda}_1 \boldsymbol{A}_\omega + \boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_{1,1}\boldsymbol{A}_\omega + \boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_1 \boldsymbol{A}_{\omega,1} \tag{3.68}$$

$$\frac{\partial \boldsymbol{a}_{\omega,1}}{\partial \boldsymbol{u}_I} = \frac{\partial \boldsymbol{\Lambda}_{2,1}}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_\omega + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_{1,1}\boldsymbol{A}_\omega + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_{\omega,1} \tag{3.69}$$

Here, the derivatives $\boldsymbol{\Lambda}_{1,1}, \boldsymbol{\Lambda}_{2,1}$ follow Equations (3.20) and (3.23), respectively. In addition, the gradients $\frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}$ and $\frac{\partial \boldsymbol{\Lambda}_{2,1}}{\partial \boldsymbol{u}_I}$ are included in the Appendix A.1.1 due to their tediousness.

Next, we derive strain gradients with respect to nodal torsional degrees of freedoms ϑ_I

$$\frac{\partial \beta_\omega(\boldsymbol{u},\vartheta)}{\partial \vartheta_I} = -\frac{\partial \boldsymbol{a}_\omega}{\partial \vartheta_I}\cdot \boldsymbol{a}_{1,1} \tag{3.70}$$

$$\frac{\partial \gamma(\boldsymbol{u},\vartheta)}{\partial \vartheta_I} = \frac{1}{2}\left(\frac{\partial \boldsymbol{a}_2}{\partial \vartheta_I}\cdot \boldsymbol{a}_{3,1} + \boldsymbol{a}_2 \cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \vartheta_I}\right) \tag{3.71}$$

where

$$\frac{\partial \boldsymbol{a}_\omega}{\partial \vartheta_I} = \boldsymbol{\Lambda}_2 \frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_I}\boldsymbol{A}_\omega \tag{3.72}$$

$$\frac{\partial \boldsymbol{a}_{\omega,1}}{\partial \vartheta_I} = \boldsymbol{\Lambda}_{2,1}\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_I}\boldsymbol{A}_\omega + \boldsymbol{\Lambda}_2\frac{\partial \boldsymbol{\Lambda}_{1,1}}{\partial \vartheta_I}\boldsymbol{A}_\omega + \boldsymbol{\Lambda}_2\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_I}\boldsymbol{A}_{\omega,1}\ . \tag{3.73}$$

Furthermore, the gradient of the rotation matrix $\boldsymbol{\Lambda}_1$, with respect to ϑ_I, is simplified

using the chain-rule of partial differentiation as,

$$\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_I} = \frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta}\frac{\partial \vartheta}{\partial \vartheta_I} \tag{3.74}$$

$$\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta} = \cos\vartheta\mathbf{K} + \sin\vartheta\mathbf{K}\mathbf{K} \quad \text{and} \quad \frac{\partial \vartheta}{\partial \vartheta_I} = N^I \tag{3.75}$$

$$\begin{aligned}\frac{\partial \boldsymbol{\Lambda}_{1,1}}{\partial \vartheta_I} &= \cos\vartheta\mathbf{K}_{,1}N^I - \sin\vartheta\mathbf{K}\vartheta_{,1}N^I + \cos\vartheta\mathbf{K}N^I_{,1} \\ &\quad + \sin\vartheta(\mathbf{K}\mathbf{K}_{,1} + \mathbf{K}_{,1}\mathbf{K})N^I + \cos\vartheta\mathbf{K}\mathbf{K}\vartheta_{,1}N^I + \sin\vartheta\mathbf{K}\mathbf{K}\vartheta_{,1}N^I_{,1}\ .\end{aligned} \tag{3.76}$$

This formally concludes the rod internal forces formulations. Next, we consider the external forces acting on the rod. To avoid the notational ambiguity, external force vector is written as $\boldsymbol{f}_{\text{ext}}$.

Rod external forces

The external forces result from the external virtual work due to the applied tractions at the traction boundary Γ. We write the external virtual work as,

$$\delta\Pi_{\text{ext}} = \int_{\text{V}} \left(\boldsymbol{f}_{\text{ext}[\boldsymbol{u}]} \cdot \delta\boldsymbol{u}\right) \text{dV} + \int_{\text{V}} \left(\boldsymbol{f}_{\text{ext}[\vartheta]} \cdot \delta\vartheta\right) \text{dV} \tag{3.77}$$

Here, readers are made aware that in this thesis, external forces $\boldsymbol{f}_{\text{ext}[\boldsymbol{u}]}$ will only be considered in the subsequent formulations and implementations. Hence, we take $\boldsymbol{f}_{\text{ext}} = \boldsymbol{f}_{\text{ext}[\boldsymbol{u}]}$. External virtual work is written as,

$$\delta\Pi_{\text{ext}} = \int_{\text{V}} \left(\boldsymbol{f}_{\text{ext}} \cdot \delta\boldsymbol{u}\right) \text{dV} = \int_{\text{V}} \left(\boldsymbol{b} \cdot \delta\boldsymbol{r}\right) \text{dV} + \int_{\Gamma} \left(\mathbf{t} \cdot \delta\boldsymbol{r}\right) \text{dA} \tag{3.78}$$

where $\boldsymbol{b}$ and $\mathbf{t}$ are body force density and external linear tractions, respectively.

In this context, we limit our attention only to linear tractions. Moreover, due to rigid cross-sections, Cauchy and Piola tractions give the same expression for a linear traction $\mathbf{t}$ as [64],

$$\mathbf{t} = s^i_{\text{f}}\boldsymbol{e}_i - \theta^2 s^2_{\text{m}}\boldsymbol{a}_{1u} - \theta^3 s^3_{\text{m}}\boldsymbol{a}_{1u}\ . \tag{3.79}$$

In Equation (3.79), the first term will result in forces in e_i directions. The second and third terms define follower force couples equivalent to moments around the two principal

axes of the rod cross-section. Hence, we understand s_{f}^{i} as a normal stress in direction e_i and s_{m}^{ω} as a bending stress per unit width in respective direction ω. In the case of follower loads, e_i of the first term in Equation (3.79) is replaced by the direction of the follower load so that the force will follow the nodal rotations. Substituting from Equations (3.2) and (3.79) in (3.78) we get,

$$\begin{aligned}\int_{\Gamma} \mathbf{t} \cdot \delta \boldsymbol{r} \mathrm{dA} &= \int_b \int_h (s_{\mathrm{f}}^{i} \boldsymbol{e}_i - \theta^2 s_{\mathrm{m}}^{2} \boldsymbol{a}_{1u} - \theta^3 s_{\mathrm{m}}^{3} \boldsymbol{a}_{1u}) \cdot (\delta \boldsymbol{x} + \theta^2 \delta \boldsymbol{a}_2 + \theta^3 \delta \boldsymbol{a}_3) \Big|_{\theta^1 = \partial S} \mathrm{d}\theta^2 \mathrm{d}\theta^3 \\ &= \underbrace{bhs_{\mathrm{f}}^{i}}_{P^i} \boldsymbol{e}_i \cdot \delta \boldsymbol{x} - \underbrace{\frac{bh^3}{12} s_{\mathrm{m}}^{2}}_{M^2} \boldsymbol{a}_{1u} \cdot \delta \boldsymbol{a}_2 - \underbrace{\frac{b^3 h}{12} s_{\mathrm{m}}^{3}}_{M^3} \boldsymbol{a}_{1u} \cdot \delta \boldsymbol{a}_3 \Big|_{\theta^1 = \partial S} ,\end{aligned} \tag{3.80}$$

where coupling between membrane and bending forces have been omitted in the formulations. Using the orthogonality conditions, $\boldsymbol{a}_{1u} \cdot \delta \boldsymbol{a}_2 = -\boldsymbol{a}_2 \cdot \delta \boldsymbol{a}_{1u}$ and $\boldsymbol{a}_{1u} \cdot \delta \boldsymbol{a}_3 = -\boldsymbol{a}_3 \cdot \delta \boldsymbol{a}_{1u}$, the external virtual work simplifies to,

$$\int_{\Gamma} \mathbf{t} \cdot \delta \boldsymbol{r} \mathrm{dA} = \left(\boldsymbol{P} \cdot \delta \boldsymbol{x} + M^2 \boldsymbol{a}_2 \cdot \delta \boldsymbol{a}_{1u} + M^3 \boldsymbol{a}_3 \cdot \delta \boldsymbol{a}_{1u} \right) \Big|_{\theta^1 = \partial S} \tag{3.81}$$

Now we can extract the external force vector using Equations (3.81) and (3.77) as,

$$\boldsymbol{f}_{\mathrm{ext}} = \left(\boldsymbol{P} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} + M^2 \boldsymbol{a}_2 \cdot \frac{\partial \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}} + M^3 \boldsymbol{a}_3 \cdot \frac{\partial \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}} \right) \Big|_{\theta^1 = \partial S} . \tag{3.82}$$

Lastly, the discretised form of the external force is written as,

$$\boldsymbol{f}_{\mathrm{ext}}^{I} = \left(\boldsymbol{P} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}_I} + M^2 \boldsymbol{a}_2 \cdot \frac{\partial \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}_I} + M^3 \boldsymbol{a}_3 \cdot \frac{\partial \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}_I} \right) \Big|_{\theta^1 = \partial S} . \tag{3.83}$$

Linearised algebraic system

It is evident that the equilibrium Equation (3.38) is nonlinear in $\boldsymbol{u}$ and ϑ, thus also in discretised $\boldsymbol{u}_I$ and ϑ_I. Therefore, we linearise the equilibrium equations [120] by defining a residual force as

$$\boldsymbol{f}_{\mathrm{res}}(\boldsymbol{u}, \vartheta) = \boldsymbol{f}_{\mathrm{int}}(\boldsymbol{u}, \vartheta) - \boldsymbol{f}_{\mathrm{ext}}(\boldsymbol{u}, \vartheta) = 0 \ . \tag{3.84}$$

Now we use the first-order Taylor expansion to linearise the stationary condition in Equation (3.38) as,

$$\boldsymbol{f}_{\text{res}}(\boldsymbol{u}_{k+1}, \vartheta_{k+1}) \approx \boldsymbol{f}_{\text{res}}(\boldsymbol{u}_k, \vartheta_k) + \Delta\boldsymbol{u}\frac{\partial \boldsymbol{f}_{\text{res}}(\boldsymbol{u}_k, \vartheta_k)}{\partial \boldsymbol{u}} + \Delta\vartheta\frac{\partial \boldsymbol{f}_{\text{res}}(\boldsymbol{u}_k, \vartheta_k)}{\partial \vartheta} = \boldsymbol{0} \tag{3.85}$$

$$\Delta\boldsymbol{u} = \boldsymbol{u}_{k+1} - \boldsymbol{u}_k \tag{3.86}$$

$$\Delta\vartheta = \vartheta_{k+1} - \vartheta_k \ . \tag{3.87}$$

Now we write the two linear systems of equations to be solved for $\boldsymbol{u}_J$ and ϑ_J, realising the Equation (3.38) is valid for any arbitrary $\delta\boldsymbol{u}$ and $\delta\vartheta$. Furthermore, to avoid the notational overload, we use $\boldsymbol{F}$ to denote the $\boldsymbol{f}_{\text{res}}$ and also drop the variable dependencies $(\boldsymbol{u}_I, \vartheta_I)$ in the discretised domain. Also, we distinguish the residual forces, according to Equations (3.57), (3.58) and follow the same subscripts in deriving the system of equations as follows.

$$(\boldsymbol{F}_{\boldsymbol{u}}^{k+1})_I = (\boldsymbol{F}_{\boldsymbol{u}}^{k})_I + \Delta\boldsymbol{u}_J \cdot \frac{\partial (\boldsymbol{F}_{\boldsymbol{u}}^{k})_I}{\partial \boldsymbol{u}_J} + \Delta\vartheta_J \frac{\partial (\boldsymbol{F}_{\boldsymbol{u}}^{k})_I}{\partial \vartheta_J} = \boldsymbol{0} \tag{3.88a}$$

$$(F_{\vartheta}^{k+1})_I = (F_{\vartheta}^{k})_I + \Delta\boldsymbol{u}_J \cdot \frac{\partial (F_{\vartheta}^{k})_I}{\partial \boldsymbol{u}_J} + \Delta\vartheta_J \frac{\partial (F_{\vartheta}^{k})_I}{\partial \vartheta_J} = 0 \ . \tag{3.88b}$$

At this juncture, attention is given to the enforcement of Dirichlet (essential) and Neumann (natural) boundary conditions. In this context, we consider only linear boundary conditions. Therefore, there is no stringent requirement to linearise the boundary conditions. Among many methods to enforce boundary conditions, i.e. Lagrange multipliers, Penalty method, Nitsche's method, we use the Lagrange multiplier method to enforce the boundary conditions due to its simple implementation and ability to extract the boundary forces directly [117, 118].

$$(\boldsymbol{B}_{\boldsymbol{u}})_{IJ}\Delta\boldsymbol{u}_J - (\boldsymbol{w}_{\boldsymbol{u}})_I = \boldsymbol{0} \tag{3.89a}$$

$$(B_{\vartheta})_{IJ}\Delta\vartheta_J - (w_{\vartheta})_I = 0 \ , \tag{3.89b}$$

where $\boldsymbol{B}_{\boldsymbol{u}}$, B_{ϑ} refer to the matrices containing respective weights of linear constraints. Furthermore, the vector $\boldsymbol{w} = \{\boldsymbol{w}_{\boldsymbol{u}}, w_{\vartheta}\}$ contains any prescribed displacements or axial rotations, respectively. We rewrite the boundary enforced linearised Equations (3.88)

using Equation (3.89) as,

$$(\boldsymbol{F}_{\boldsymbol{u}}^{k})_I + \Delta\boldsymbol{u}_J \cdot \frac{\partial(\boldsymbol{F}_{\boldsymbol{u}}^{k})_I}{\partial \boldsymbol{u}_J} + \Delta\vartheta_J \frac{\partial(\boldsymbol{F}_{\boldsymbol{u}}^{k})_I}{\partial \vartheta_J} + (B_{\boldsymbol{u}}^{\mathrm{T}})_{IJ}(\Delta\boldsymbol{\lambda}_{\boldsymbol{u}})_J = \mathbf{0} \tag{3.90a}$$

$$(F_{\vartheta}^{k})_I + \Delta\boldsymbol{u}_J \cdot \frac{\partial(F_{\vartheta}^{k})_I}{\partial \boldsymbol{u}_J} + \Delta\vartheta_J \frac{\partial(F_{\vartheta}^{k})_I}{\partial \vartheta_J} + (B_{\vartheta}^{\mathrm{T}})_{IJ}(\Delta\lambda_{\vartheta})_J = 0 \ , \tag{3.90b}$$

where the vector $\Delta\boldsymbol{\lambda} = \{\Delta\boldsymbol{\lambda}_{\boldsymbol{u}}, \Delta\lambda_{\vartheta}\}$ contains the respective boundary enforcing Lagrange multipliers. Using Equations (3.88) and (3.89), now we complete the full linear system of equations by writing,

$$\left[\begin{array}{c|c|c} \dfrac{\partial(\boldsymbol{F}_{\boldsymbol{u}})_I}{\partial \boldsymbol{u}_J} & \dfrac{\partial(\boldsymbol{F}_{\boldsymbol{u}})_I}{\partial \vartheta_J} & (\boldsymbol{B}_{\boldsymbol{u}}^{\mathrm{T}})_{IJ} \\ \hline \dfrac{\partial(F_{\vartheta})_I}{\partial \boldsymbol{u}_J} & \dfrac{\partial(F_{\vartheta})_I}{\partial \vartheta_J} & (B_{\vartheta}^{\mathrm{T}})_{IJ} \\ \hline (\boldsymbol{B}_{\boldsymbol{u}})_{IJ} & (B_{\vartheta})_{IJ} & \mathbf{0} \end{array}\right] \left\{\begin{array}{c} \Delta\boldsymbol{u}_J \\ \hline \Delta\vartheta_J \\ \hline \Delta\boldsymbol{\lambda}_J \end{array}\right\} = \left\{\begin{array}{c} -(\boldsymbol{F}_{\boldsymbol{u}})_I \\ \hline -(F_{\vartheta})_I \\ \hline \boldsymbol{w}_I \end{array}\right\} \tag{3.91}$$

The linear system of equations in Equation (3.91) requires to determine the Hessian terms to complete the formulations of spatial rod kinematics. Hessian matrix is symmetric and can be computed by differentiating the total potential energy twice or differentiating residual forces once with respect to the degrees of freedom.

Hessian matrix terms

Hessian matrix, in general, includes the contributions from both internal and external forces. Hessian component from internal forces is usually termed the *stiffness matrix*. For linear problems, stiffness matrices do exist, but are independent of the kinematic variables $\boldsymbol{u}$ and ϑ. In addition, external forces also contribute to the Hessian when they have non-zero tangents with respect to $\boldsymbol{u}$ and ϑ. For instance, consider a follower load that is coupled to one (or a combination) of the base vectors. This load follows the nodal rotations and therefore possess a tangent which gives rise to a Hessian contribution. In this thesis, we consider both internal and external Hessian contributions and illustrate the accurate modelling through examples provided in Section 3.4. We first consider

the Hessian terms from the internal forces. To this end, we need to formulate the main four stiffness contributions as given in Equation (3.91). For clarity and to avoid excessively long equations in the main body of the thesis, detailed formulations are included in Appendix A.1.2. Stiffness contributions from initial curvatures are also presented separately.

$$\begin{aligned}\frac{\partial(\boldsymbol{F_u})_I}{\partial \boldsymbol{u}_J} = &\frac{\partial n}{\partial \boldsymbol{u}_J}\frac{\partial \alpha}{\partial \boldsymbol{u}_I} + n\frac{\partial^2 \alpha}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} + \sum_{j=2}^{3}\left(\frac{\partial m^j}{\partial \boldsymbol{u}_J}\frac{\partial \beta_j}{\partial \boldsymbol{u}_I} + m^j\frac{\partial^2 \beta_j}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J}\right) \\ &+ \frac{\partial q}{\partial \boldsymbol{u}_J}\frac{\partial \gamma}{\partial \boldsymbol{u}_I} + q\frac{\partial^2 \gamma}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} \ .\end{aligned} \tag{3.92}$$

$$\frac{\partial(\boldsymbol{F_u})_I}{\partial \vartheta_J} = \sum_{j=2}^{3}\left(\frac{\partial m^j}{\partial \vartheta_J}\frac{\partial \beta_j}{\partial \boldsymbol{u}_I} + m^j\frac{\partial^2 \beta_j}{\partial \boldsymbol{u}_I \partial \vartheta_J}\right) + \frac{\partial q}{\partial \vartheta_J}\frac{\partial \gamma}{\partial \boldsymbol{u}_I} + q\frac{\partial^2 \gamma}{\partial \boldsymbol{u}_I \partial \vartheta_J} \ . \tag{3.93}$$

Due to the symmetry of the stiffness matrix we have the following relationship,

$$\frac{\partial(F_\vartheta)_I}{\partial \boldsymbol{u}_J} = \left[\frac{\partial(\boldsymbol{F_u})_I}{\partial \vartheta_J}\right]^{\mathrm{T}} \ . \tag{3.94}$$

Lastly, $\dfrac{\partial(F_\vartheta)_I}{\partial \vartheta_J}$ is simplified as,

$$\frac{\partial(F_\vartheta)_I}{\partial \vartheta_J} = \sum_{j=2}^{3}\left(\frac{\partial m^j}{\partial \vartheta_J}\frac{\partial \beta_j}{\partial \vartheta_I} + m^j\frac{\partial^2 \beta_j}{\partial \vartheta_I \partial \vartheta_J}\right) + \frac{\partial q}{\partial \vartheta_J}\frac{\partial \gamma}{\partial \vartheta_I} + q\frac{\partial^2 \gamma}{\partial \vartheta_I \partial \vartheta_J} \ . \tag{3.95}$$

Stiffness contributions due to initial curvatures can be written as,

$$\frac{\partial(\boldsymbol{F_u})_I}{\partial \boldsymbol{u}_J} = -\mathcal{C}^{11}\sum_{j=2}^{3}\kappa_j I_j\left(\frac{\partial \beta_j}{\partial \boldsymbol{u}_J}\frac{\partial \alpha}{\partial \boldsymbol{u}_I} + \beta_j\frac{\partial^2 \alpha}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} + \frac{\partial \alpha}{\partial \boldsymbol{u}_J}\frac{\partial \beta_j}{\partial \boldsymbol{u}_I} + \alpha\frac{\partial^2 \beta_j}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J}\right) \tag{3.96}$$

$$\frac{\partial(F_\vartheta)_I}{\partial \boldsymbol{u}_J} = -\mathcal{C}^{11}\sum_{j=2}^{3}\kappa_j I_j\left(\frac{\partial \beta_j}{\partial \vartheta_I}\frac{\partial \alpha}{\partial \boldsymbol{u}_J} + \alpha\frac{\partial^2 \beta_j}{\partial \boldsymbol{u}_J \partial \vartheta_I}\right) \tag{3.97}$$

$$\frac{\partial(F_\vartheta)_I}{\partial \vartheta_J} = -\mathcal{C}^{11}\sum_{j=2}^{3}\kappa_j I_j\left(\alpha\frac{\partial^2 \beta_j}{\partial \vartheta_I \partial \vartheta_J}\right) \ . \tag{3.98}$$

Note that the stiffness entries from initial curvatures will be added to respective stiffness terms given in Equations (3.92) - (3.95).

Next, we consider the stiffness entries due to external forces with non-zero tangents. To this end, we identify follower loads and moments (given by follower load couples in Equation (3.83)) give rise to non-zero tangents. Therefore, we write the external stiffness matrix contributions as,

$$\frac{\partial (\boldsymbol{f}_{\text{ext}})_I}{\partial \boldsymbol{u}_J} = \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{u}_J} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}_I} + \sum_{j=2}^{3} M^j \left(\frac{\partial \boldsymbol{a}_j}{\partial \boldsymbol{u}_J} \cdot \frac{\partial \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}_I} + \boldsymbol{a}_j \cdot \frac{\partial^2 \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} \right) . \tag{3.99}$$

Note that the first term in Equation (3.99) will be zero for fixed loads.

Section 3.2 detailed some of the key aspects of the nonlinear isogeometric rod model that will be used to model and analyse the textile RVEs. These formulations achieved the objectives and required features set in Table 3.1 in the context of modelling the yarns of knitted and woven textiles.

3.3 Contact formulation for spatial rods

Beam-to-beam collision modelling is still a demanding topic in the related research communities due to its complexity and limitations [81, 107, 111, 121–126]. Currently, there are two different mechanical modelling techniques, namely Point-to-point and Line-to-line, where each of these techniques has its own merits and demerits. In this thesis, we closely follow the collision models initially suggested by Wriggers et al. [121] and further developed by [81, 107, 111].

3.3.1 Point-to-point contact model

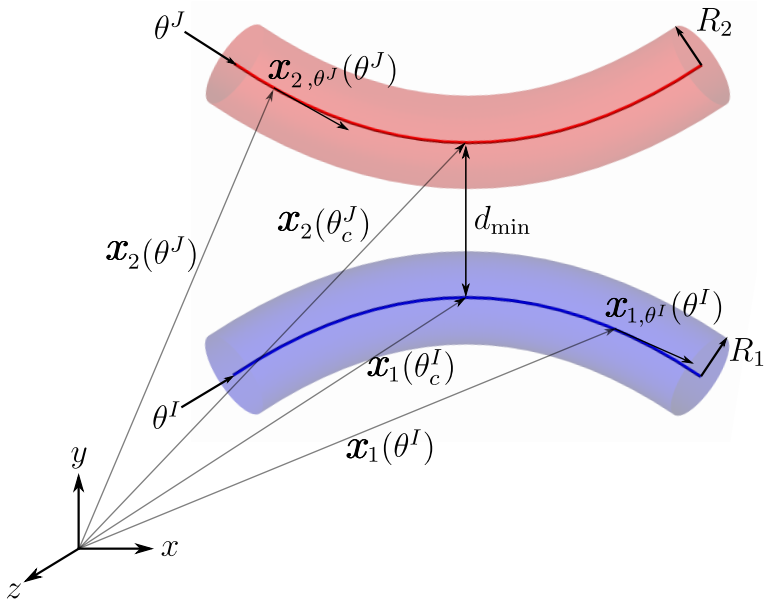


Figure 3.3: Point-to-point collision modelling of two spatial rods.

In this approach, a discrete contact force at the closest point of the rods is formulated by minimising the penetration depth if contact occurs. Here, we use two parameterisations for two rods given by parametric equations $\boldsymbol{x}_1(\theta_1^I)$ and $\boldsymbol{x}_2(\theta_1^J)$. To avoid the notational overload, subscript '1' is dropped from θ_1 in subsequent derivations.

First, we find the distance between two material points of two spatial rods as,

$$d(\theta^I, \theta^J) = ||\boldsymbol{x}_1(\theta^I) - \boldsymbol{x}_2(\theta^J)|| \ . \tag{3.100}$$

and it reaches a minimum closer to a potential collision location as shown in Figure 3.3. This is written as,

$$d_{\min} = \min_{\theta^I, \theta^J} d(\theta^I, \theta^J) = d(\theta_c^I, \theta_c^J) \ . \tag{3.101}$$

The unknown parametric locations at contact θ_c^I and θ_c^J (in Equation (3.101)) can be found by solving the following two systems considering the orthogonality between the material point vector difference and their tangents of the rods.

$$\boldsymbol{x}_{1,\theta^I}^{\mathrm{T}}(\theta_c^I)(\boldsymbol{x}_1(\theta_c^I) - \boldsymbol{x}_2(\theta_c^J)) = 0 \tag{3.102a}$$

$$\boldsymbol{x}_{2,\theta^J}^{\mathrm{T}}(\theta_c^J)(\boldsymbol{x}_1(\theta_c^I) - \boldsymbol{x}_2(\theta_c^J)) = 0 \ . \tag{3.102b}$$

Solution for θ_c^I and θ_c^J is attained using the Newton-Raphson iterative scheme by linearising the orthogonality conditions given in Equations (3.102).

$$\begin{aligned}
\left[\boldsymbol{x}_{2,\theta^I}^{\mathrm{T}}(\theta_i^J)(\boldsymbol{x}_1(\theta^I)-\boldsymbol{x}_2(\theta_i^J))\right]_{,\theta^I}\Delta\theta^I &= -\boldsymbol{x}_{2,\theta^I}^{\mathrm{T}}(\theta_i^J)(\boldsymbol{x}_1(\theta^I)-\boldsymbol{x}_2(\theta_i^J)) \\
\left[\boldsymbol{x}_{2,\theta^J}^{\mathrm{T}}(\theta_i^J)(\boldsymbol{x}_1(\theta^I)-\boldsymbol{x}_2(\theta_i^J))\right]_{,\theta^J}\Delta\theta^J &= -\boldsymbol{x}_{2,\theta^J}^{\mathrm{T}}(\theta_i^J)(\boldsymbol{x}_1(\theta^I)-\boldsymbol{x}_2(\theta_i^J)) \\
\theta_{i+1}^I &= \theta_i^I + \Delta\theta^I \\
\theta_{i+1}^J &= \theta_i^J + \Delta\theta^J
\end{aligned} \tag{3.103}$$

After locating possible collision points θ_c^I and θ_c^J, next task is to define the non-penetration condition using a gap function g as,

$$g \geq 0 \qquad \text{with} \qquad g = d_{\min} - R_1 - R_2 \ . \tag{3.104}$$

Energy contribution from collisions (if present) is added to the variational problem using widely used Lagrange multiplier method. Here, we derive only the contribution from collisions which needs to be added to the total potential energy given in Equation (3.33). To this end, we write the energy contributions $\Pi_{c\lambda}$ as,

$$\Pi_{c\lambda} = \lambda\langle g\rangle \qquad \text{with} \qquad \langle x\rangle = \begin{cases} x & x \leq 0 \\ 0 & x > 0 \ . \end{cases} \tag{3.105}$$

First variation of the energy yields its weak form contribution given by,

$$\delta\Pi_{c\lambda} = \lambda\delta g + \delta\lambda\langle g\rangle = \lambda(\delta\boldsymbol{x}_1(\theta_c^I) - \delta\boldsymbol{x}_2(\theta_c^J))\cdot\boldsymbol{n} + \delta\lambda\langle g\rangle \ . \tag{3.106}$$

where $\boldsymbol{n} = \dfrac{\boldsymbol{x}_1(\theta_c^I)-\boldsymbol{x}_2(\theta_c^J)}{||\boldsymbol{x}_1(\theta_c^I)-\boldsymbol{x}_2(\theta_c^J)||}$ is the unit normal vector at the collision points, along which the equal and opposite contact forces are applied to avoid penetration.

Additional co-efficients to the Lagrange multiplier array are given by,

$$\frac{\partial\Pi_{c\lambda}}{\partial\boldsymbol{u}_I} = \lambda\left(\frac{\partial\boldsymbol{x}_{1c}}{\partial\boldsymbol{u}_I} - \frac{\partial\boldsymbol{x}_{2c}}{\partial\boldsymbol{u}_I}\right)\cdot\boldsymbol{n} \tag{3.107}$$

$$\frac{\partial\Pi_{c\lambda}}{\partial\lambda} = \langle g\rangle \ . \tag{3.108}$$

Point-to-point contact scheme performs well for large contact angles [111], where the contact angle is defined as the angle between the two tangent vectors at the contact

point. But sometimes it is difficult to find a unique closest point when the contact angle is small, especially in an iterative scheme like Equation (3.103), and hence does not perform well in the small contact angle regime [81]. It is in contrast to the expectations of line-to-line contacts. Line-to-line contacts perform well for smaller contact angles. From a mechanics standpoint, it is understood that line-to-line contacts are not feasible to be formed when the contact angles are larger. In this thesis, the line-to-line contact formulation of Meier et al. [81] is closely followed.

3.3.2 Line-to-line contact model

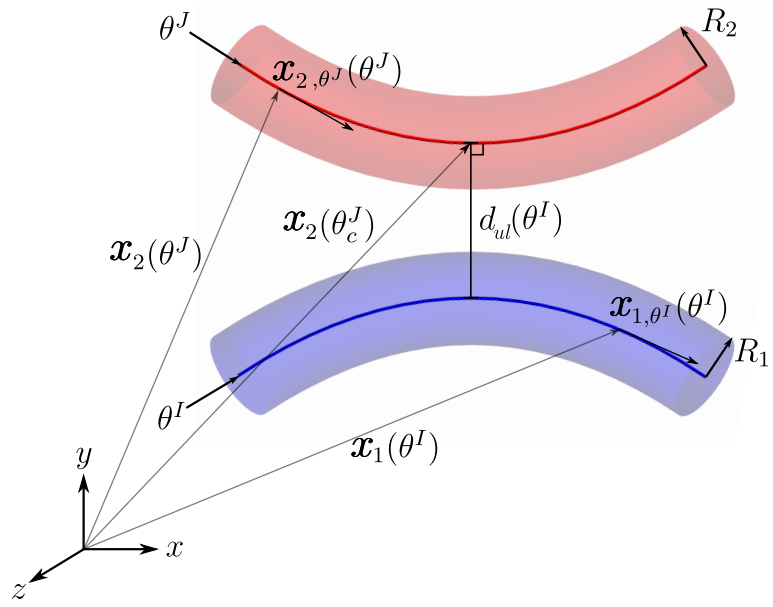


Figure 3.4: Line-to-line collision modelling of two spatial rods.

Here, a distinction has to be made between a master beam ($\boldsymbol{x}_2(\theta^J)$) and a slave beam ($\boldsymbol{x}_1(\theta^I)$). The closest master point θ_c^J to a given slave point θ^I is determined as solution of the following unilateral (ul) minimal distance problem,

$$d(\theta^I, \theta^J) = ||\boldsymbol{x}_1(\theta^I) - \boldsymbol{x}_2(\theta^J)|| \tag{3.109}$$

$$d_{ul}(\theta^I) = \min_{\theta^J} d(\theta^I, \theta^J) = d(\theta^I, \theta_c^J) \ . \tag{3.110}$$

This leads only to one orthogonality condition to be satisfied for θ_c^J and takes the form,

$$\boldsymbol{x}_{2,\theta^J}^{\mathrm{T}}(\theta_c^J)(\boldsymbol{x}_1(\theta^I) - \boldsymbol{x}_2(\theta_c^J)) = 0 \ . \tag{3.111}$$

Solution for θ_c^J is attained using Newton-Raphson iterative scheme by linearising the orthogonality condition.

$$\left[\boldsymbol{x}_{2,\theta^J}^{\mathrm{T}}(\theta_i^J)(\boldsymbol{x}_1(\theta^I)-\boldsymbol{x}_2(\theta_i^J))\right]_{,\theta^J}\Delta\theta^J = -\boldsymbol{x}_{2,\theta^J}^{\mathrm{T}}(\theta_i^J)(\boldsymbol{x}_1(\theta^I)-\boldsymbol{x}_2(\theta_i^J)) \tag{3.112}$$

$$\theta_{i+1}^J = \theta_i^J + \Delta\theta^J \tag{3.113}$$

Unlike in a point-to-point contact model, the normal vector $\boldsymbol{n}(\theta^I)$ is perpendicular to the master beam at θ_c^J but not to the slave beam. Therefore, it is given as a function of θ^I. Next, we write the non-penetration condition for the line-to-line contact model as,

$$g(\theta^I) \geq 0 \;\forall\; \theta^I \qquad \text{with} \qquad g(\theta^I) = d_{ul}(\theta^I) - R_1 - R_2 \;. \tag{3.114}$$

Similar to the point-to-point contact model, the energy from collisions is included in the variational form using the Lagrange multiplier method. Now the Lagrange multiplier acts as a line load along the contact region δl that transfers a distributed load to the two rod bodies.

$$\Pi_{c\lambda} = \int_{\delta l} \lambda(\theta^I) g(\theta^I)\; \mathrm{d}l \tag{3.115}$$

Thus the virtual work expression takes the form,

$$\begin{aligned}
\delta\Pi_{c\lambda} &= \int_{\delta l} \left(\lambda(\theta^I)\delta g(\theta^I) + \delta\lambda(\theta^I) g(\theta^I)\right) \mathrm{d}l \\
&= \int_{\delta l} \left(\lambda(\theta^I)(\delta\boldsymbol{x}_1 - \delta\boldsymbol{x}_{2c})\cdot\boldsymbol{n}(\theta^I) + \delta\lambda(\theta^I) g(\theta^I)\right) \mathrm{d}l \\
\text{with} \qquad \boldsymbol{n}(\theta^I) &= \frac{\boldsymbol{x}_1(\theta^I) - \boldsymbol{x}_2(\theta_c^J)}{||\boldsymbol{x}_1(\theta^I) - \boldsymbol{x}_2(\theta_c^J)||}
\end{aligned} \tag{3.116}$$

Additional co-efficients to the Lagrange Multiplier arrays are given by,

$$\frac{\partial\Pi_{c\lambda}}{\partial\boldsymbol{u}_I} = \int_{\delta l} \lambda(\theta^I) \left(\frac{\partial\boldsymbol{x}_1(\theta^I)}{\partial\boldsymbol{u}_I} - \frac{\partial\boldsymbol{x}_2(\theta_c^J)}{\partial\boldsymbol{u}_I}\right)\cdot\boldsymbol{n}(\theta^I)\; \mathrm{d}l \tag{3.117}$$

$$\frac{\partial\Pi_{c\lambda}}{\partial\lambda_I} = \int_{\delta l} \frac{\partial\lambda}{\partial\lambda_I} g(\theta^I)\; \mathrm{d}l \tag{3.118}$$

Algorithm 1 outlines the key steps of the C++ algorithm developed to model and simulate the finite deformable beam models under the isogeometric analysis paradigm.

Algorithm 1 Nonlinear isogeometric 3D rod analysis

// run the analysis preprocessor
1: **preprocessor** : definition of the geometry, loading, material and cross section properties
// define the reference configuration of the structure
2: **initialise** : $\boldsymbol{f}_{\text{ext}} = \mathbf{0},\ \boldsymbol{x} = \boldsymbol{X},\ \boldsymbol{\vartheta} = \boldsymbol{\vartheta}_0,\ \boldsymbol{f}_{res} = \mathbf{0}$
// start step-loading the structure
3: **for** `i` $\leq$ `load steps` **do**
4: **find** $\Delta\boldsymbol{f}_{\text{ext}}$
5: **set** $\boldsymbol{f}_{\text{ext}} \leftarrow \boldsymbol{f}_{\text{ext}} + \Delta\boldsymbol{f}_{\text{ext}}$
6: **set** $\boldsymbol{f}_{\text{res}} \leftarrow \boldsymbol{f}_{\text{res}} - \Delta\boldsymbol{f}_{\text{ext}}$
// equilibrate the structure at every load step
7: **while** $(||\boldsymbol{f}_{\text{res}}||/||\boldsymbol{f}_{\text{ext}}|| <$ `tolerance`$)$ **do**
8: **solve** $\mathbf{K}\boldsymbol{u}_{(\boldsymbol{u},\vartheta)} = -\boldsymbol{f}_{\mathbf{res}}$
9: **update** $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{u}_{\boldsymbol{u}}$ and $\boldsymbol{\vartheta} \leftarrow \boldsymbol{\vartheta} + \boldsymbol{u}_{\vartheta}$
10: **compute** vector triad ($\boldsymbol{a}_1,\ \boldsymbol{a}_2$ and $\boldsymbol{a}_3$) and their derivatives
11: **compute** $\boldsymbol{f}_{\text{int}}$ and $\mathbf{K}$
12: **update** $\boldsymbol{f}_{\text{res}} \leftarrow \boldsymbol{f}_{\text{int}} - \boldsymbol{f}_{\text{ext}}$
13: **end while**
14: **end for**

3.4 Model verification

In this section, several benchmark problems are solved using the spatial rod model developed in Section 3.2. Every problem follows a similar illustration framework, where at first, the problem description (initial geometry, material properties, cross-section details) is presented. Next, the comparisons are made with a sketch of the deformed shapes to show the large deformations. It is noted that the units of each problem is taken from the respective reference article and thus some problems take SI units and others are presented with no units.

3.4.1 Membrane-bending interaction

Clamped-hinged deep circular arch subject to point load

This example has been considered by a number of authors [50–52, 127] and the exact solution based on the Kirchhoff-Love theory is given by DaDeppo and Schmidt [128]. We use smooth manifold basis functions to uniformly discretise the arch into 40 finite elements. The finite element mesh consist of 43 nodes including two ghost nodes, as explained in Section 2.2.

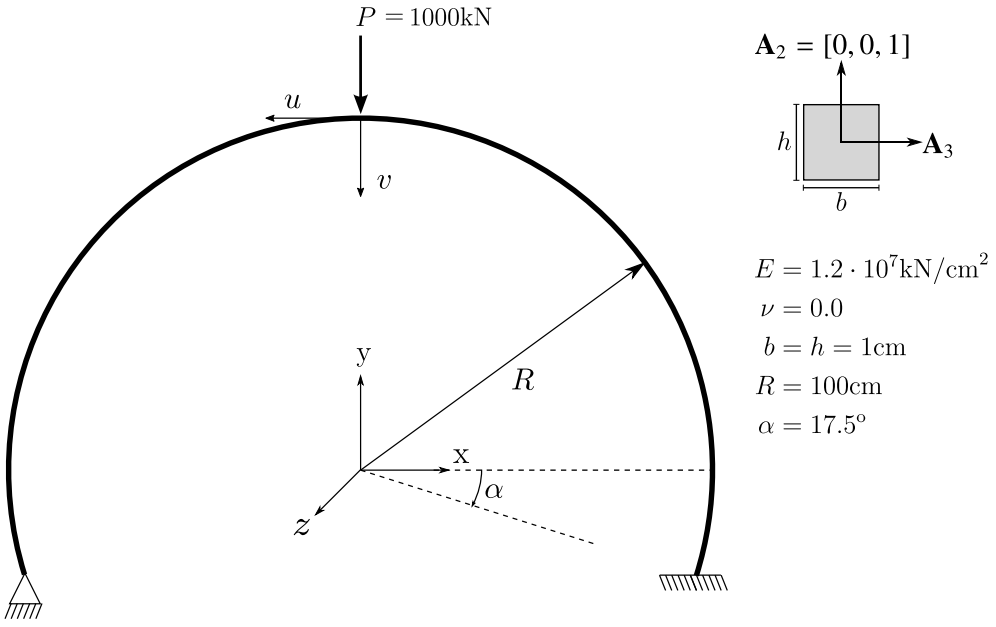
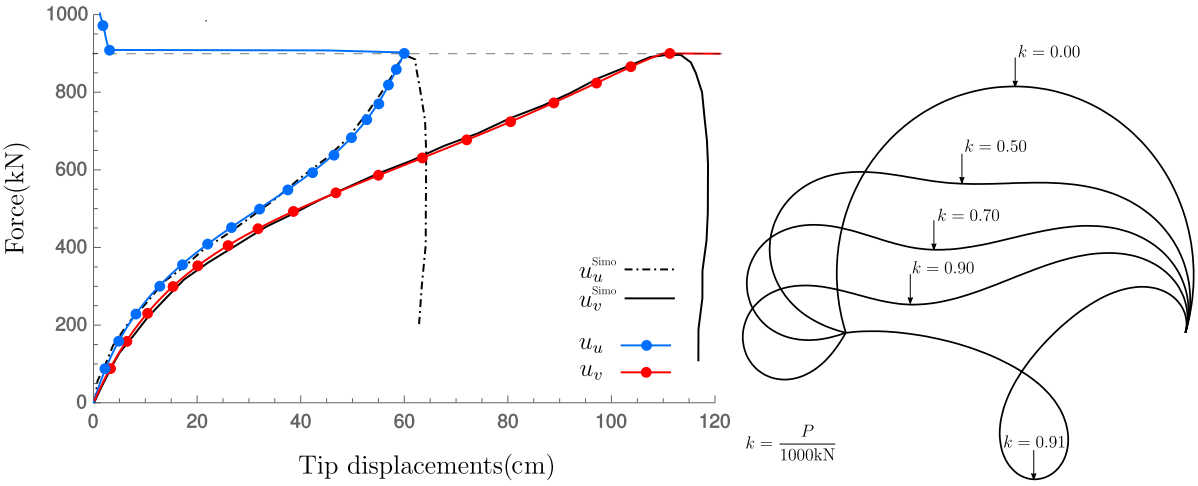


Figure 3.5: Problem description of a deep circular arch with a fixed load applied at the crest.



(a) (b)

Figure 3.6: (a) Comparison of numerical results of the deep circular arch until buckling. (b) Obtained deformed shapes of the deep circular arch at different load levels.

3.4.2 Membrane-bending-torsion interaction

Cantilever 45-degree bend subject to a tip fixed load

This example has been considered by several authors [50, 52, 100, 129, 130] under a fixed tip load. This example not only considers large deformations of the rod but also takes into account the interactions between membrane, bending and torsion actions during deformations. Moreover, it includes additional terms from initial curvature and hence provides a good benchmark to test the accuracy of the developed spatial rod model. We use smooth manifold basis functions to uniformly discretise the arch into 20 finite elements. The finite element mesh consist of 23 nodes including two ghost nodes, as explained in Section 2.2.

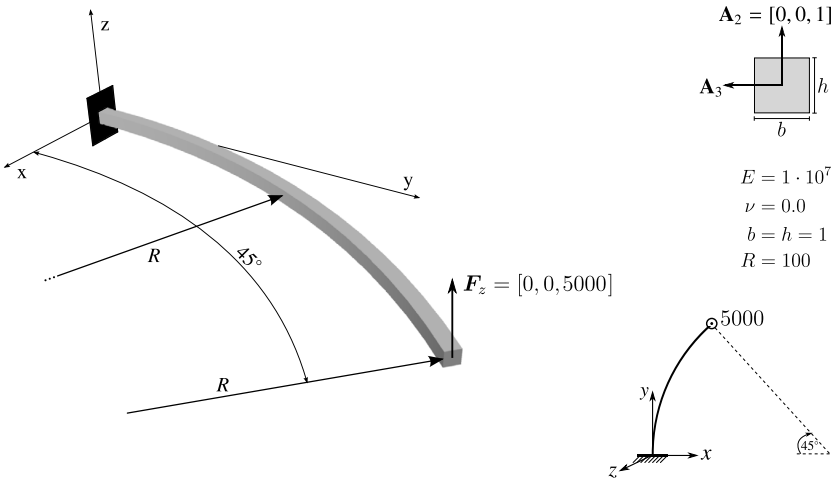
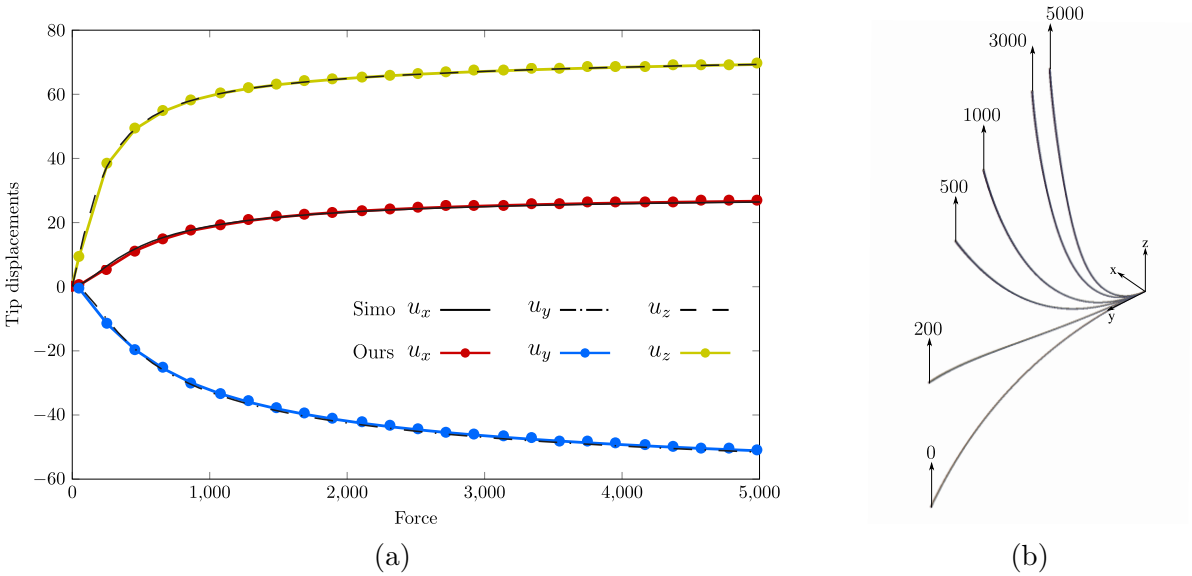


Figure 3.7: Problem description of a tip loaded 45° degree cantilever.



(a) (b)

Figure 3.8: (a) Comparison of numerical results of the tip loaded 45° cantilever. (b) Deformed shape of the cantilever at different load levels.

Helicoidal spring subject to a tip fixed load

We consider another membrane-bending-torsion interaction problem. However, the *reference* geometry of the helicoidal spring is now in full 3D unlike the tip loaded *planar* cantilever problem. We use the geometry, material and cross-section properties used in [52, 131] and compare the numerical results of the tip displacement by referring to

Bauer et al. [52]. A uniform finite element mesh of 40 elements and 46 nodes (including six ghost nodes) is used for the isogeometric analysis using quintic B-spline basis functions. We define the initial geometry as

$$\boldsymbol{X}(\theta^1) = \{10\sin(2\pi\theta^1), 10\cos(2\pi\theta^1), 20\theta^1\}, \qquad \theta^1 \in [0,1] \tag{3.119}$$

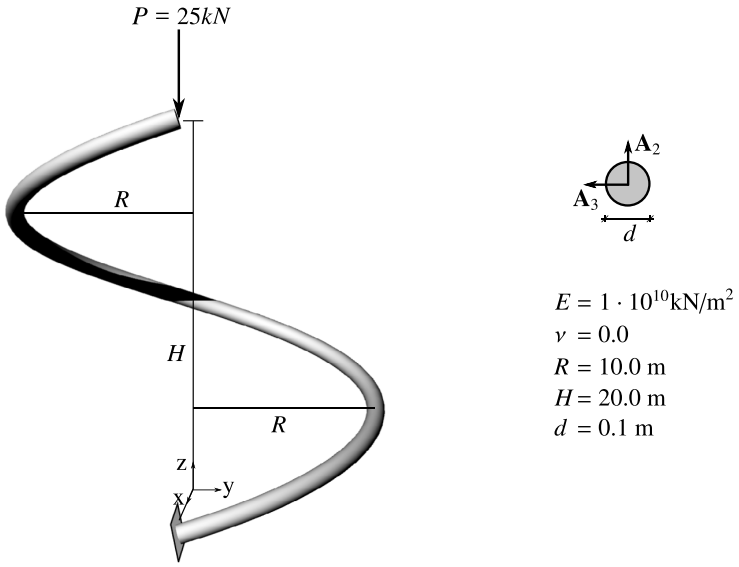
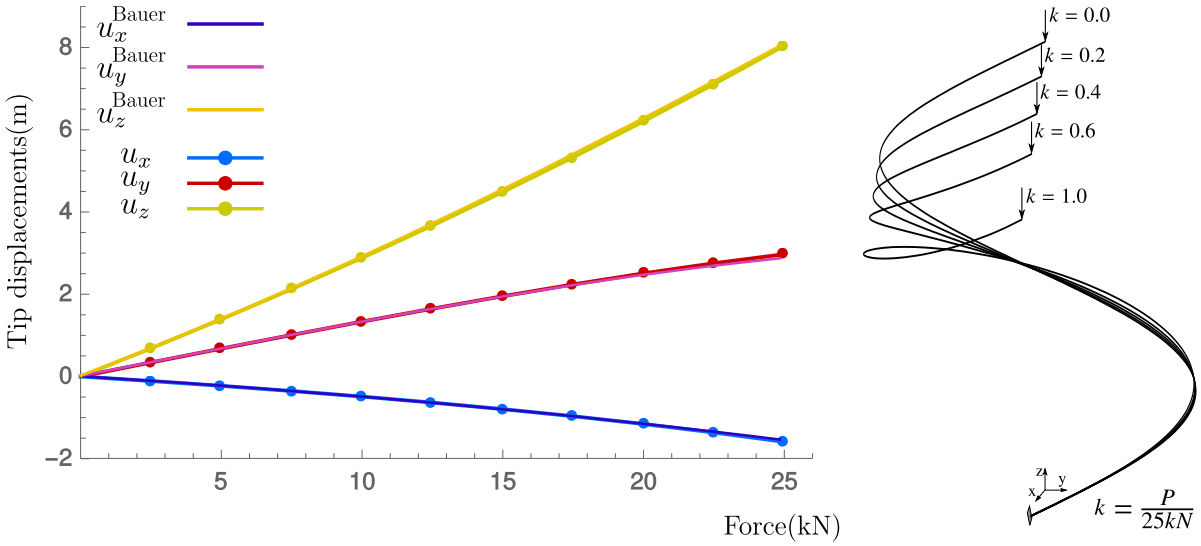


Figure 3.9: Problem description of a tip loaded spatial helicoidal spring



(a) (b)

Figure 3.10: (a) Comparison of numerical results of the tip loaded spatial helicoidal spring. (b) Deformed shape of the spring at different load levels.

3.4.3 Benchmark problems

A few carefully selected benchmark example problems are numerically solved and presented below following the same illustration framework introduced in Section 3.4. Each problem has its uniqueness and addresses at least one of the key features in rod kinematics, formulations, geometry and implementation aspects.

Clamped-clamped shallow circular arch subject to a midspan point load

This example has been considered by a number of authors [52, 100, 129] hence represents a well-established benchmark example. A shallow arch is deformed using displacement control method and analysed for nonlinear buckling and post-buckling behaviour. Numerical results show an excellent accuracy to the reference solution of Lo [100], and the comparison is presented in Figure 3.10a. We conclude that post-buckling behaviour can correctly be analysed with the presented spatial rod formulations. We use smooth manifold basis functions to discretise the arch into 20 equal finite elements. The finite element mesh consist of 23 nodes including two ghost nodes, as explained in Section 2.2.

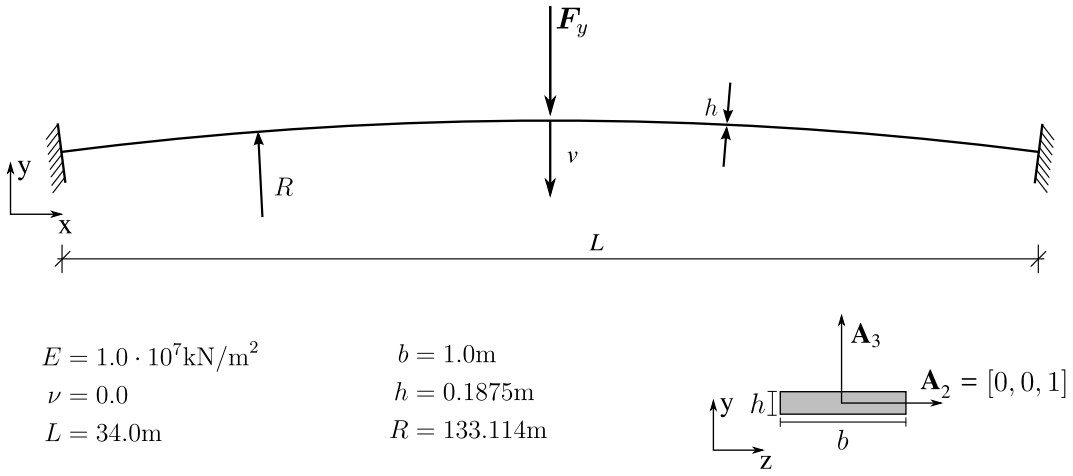
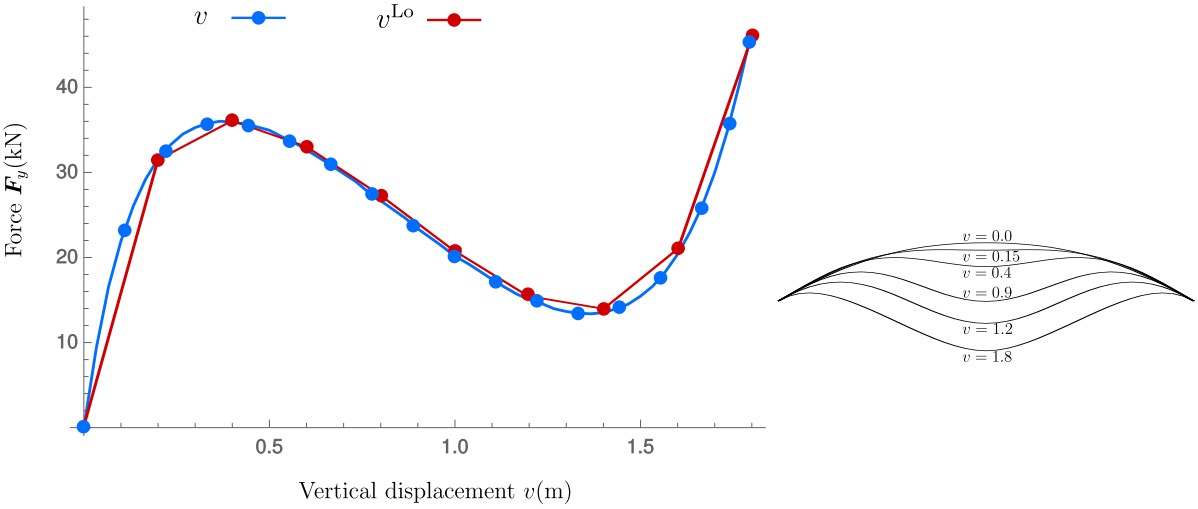


Figure 3.11: Problem description of a shallow circular arch with prescribed vertical displacement control at the crest.



(a) (b)

Figure 3.12: (a) Comparison of numerical results of the shallow arch in the post-buckling regime. (b) Deformed shapes of the shallow arch at different load levels.

Cantilever beam subject to a follower tip load

This problem allows us to demonstrate the inclusion of non-zero external force tangents in the Hessian. Formulations given in Equations (3.96), (3.97) and (3.98) apply here, where the load magnitude ($||\boldsymbol{F}_{\boldsymbol{a}_3}||$) is attached to the deformed normal vector $\boldsymbol{a}_3$. Therefore, during deformation, load follows the nodal rotation to maintain the 90° angle with the cantilever tip. This problem has been analysed by many authors [50,127], and in this thesis, our numerical solution is compared with the results of Simo et al. [50]. We use cubic B-spline basis functions to discretise the cantilever into 20 equal finite elements. The finite element mesh consist of 23 nodes including two ghost nodes.

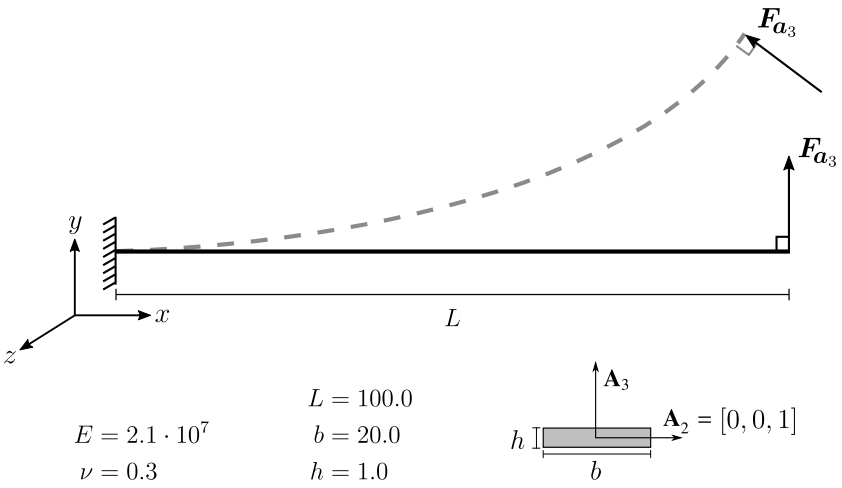


Figure 3.13: Problem description of the cantilever subject to a follower tip load.

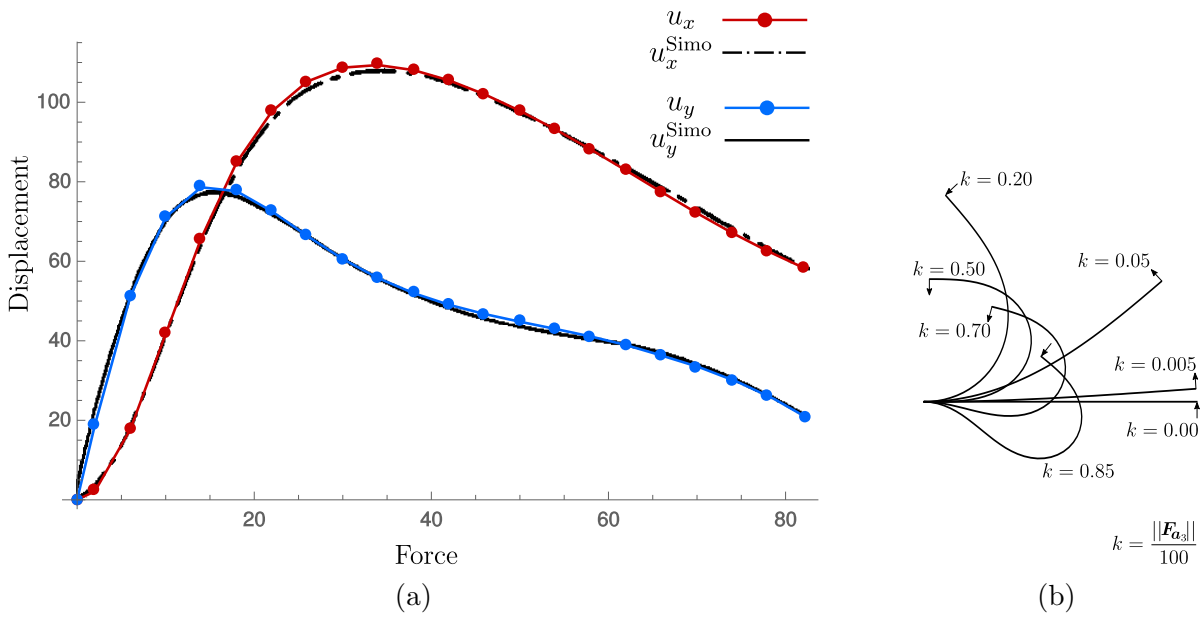


Figure 3.14: (a) Comparison of numerical results of the cantilever subject to a follower tip load. (b) Deformed shape of the cantilever at different load magnitudes.

Hinged beam

This problem is used to elaborate on the linear isogeometric analysis using non-smooth manifold basis functions introduced in Section 2.2.2. We introduce the non-smooth feature, a hinge in this problem, by mapping the C^0-smooth point (see Figure 2.17) of non-smooth manifold basis functions onto the physical domain where the hinge is

located. We use smooth and non-smooth manifold basis functions to discretise the hinged beam into 40 equal finite elements. Moreover, the finite element mesh consist of 43 nodes including two ghost nodes.

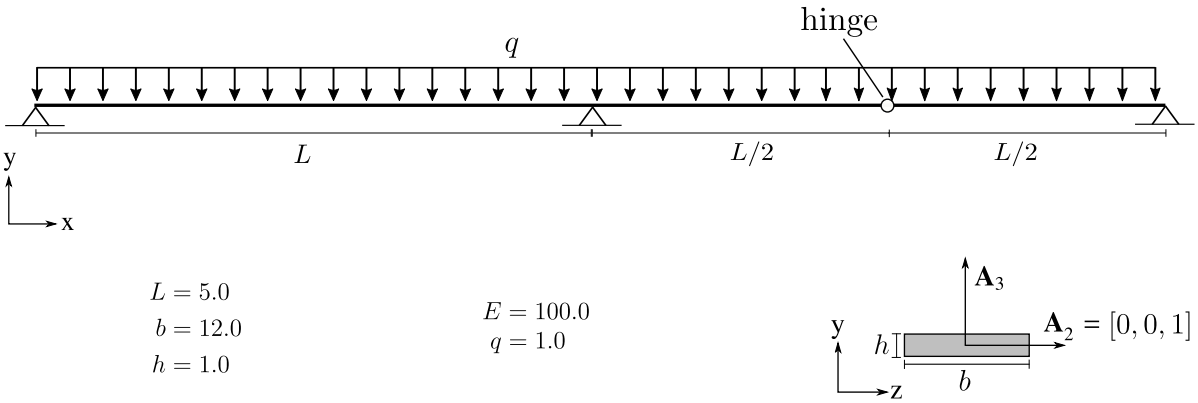
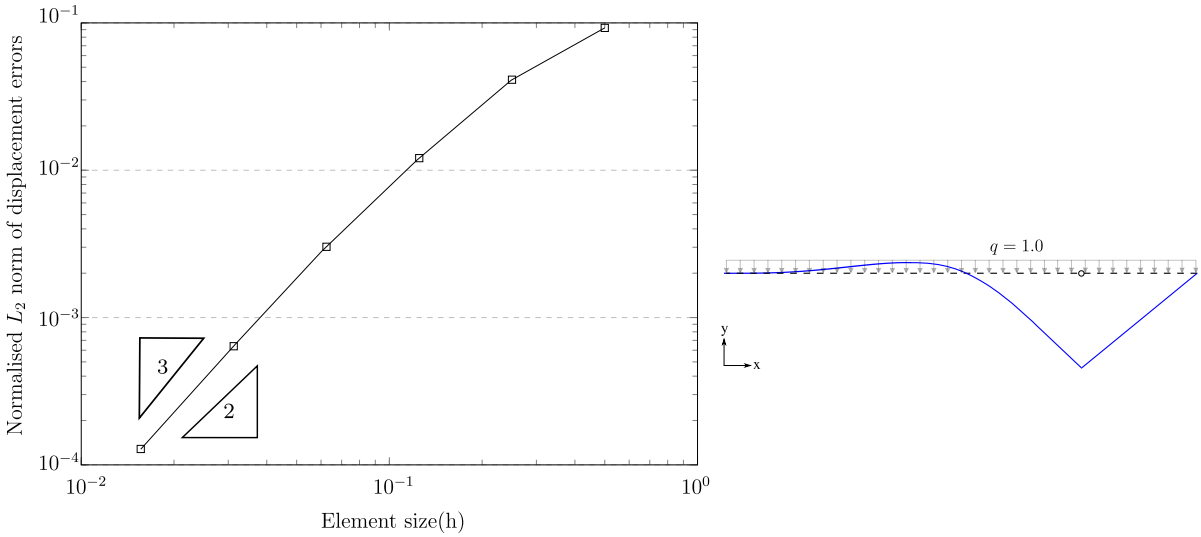


Figure 3.15: Problem description of a hinged beam subject to uniform loading



(a) (b)

Figure 3.16: (a) Error rate and convergence of the L_2 norm of transverse displacement errors. (b) Deformed shape of the hinged beam under uniform loading.

Figure 3.16a shows the convergence rate of using manifold basis functions in isogeometric analysis. We obtained near optimum convergence rate of 2.82. Manifold basis functions being rational polynomials (due to normalisation) is identified as the reason for this sub-optimal convergence rate. Optimum convergence rate for this type of analysis would be $p + 1(= 3)$, where p is the local polynomial degree [132].

Figure 3.16b depicts the deformed shape of the hinged beam. As expected, the deformed shape consists of a non-smooth (C^0) point that corresponds to the hinge location. Moreover, vanishing moments at the hinge can be visually verified by observing linearly varied displacements close to the hinge. We derive the following displacement functions from the statically determinant problem in Figure 3.15, for the L_2 error calculation to plot 3.16a. This L_2 norm error is defined as $\sqrt{\int_\Omega (w - w_h)^2 \mathrm{d}\Omega / \int_\Omega w^2 \mathrm{d}\Omega}$ where w is the transverse displacement, $(\bullet)_h$ denotes the finite element solution and Ω denotes the physical domain of the beam body.

$$w(x) = \begin{cases} \dfrac{qLx^3}{24EI} - \dfrac{qx^4}{24EI} + \dfrac{qL(x-L)^3}{4EI} & 0 \leq x \leq 1.5L \\ \dfrac{qL(x-1.5L)^3}{24EI} - \dfrac{q(x-1.5L)^4}{24EI} + \dfrac{qL^3(x-1.5L)}{12EI} - \dfrac{5qL^4}{128EI} & 1.5L < x \leq 2L \ . \end{cases}$$

Rod contacts

It is vital to numerically prove the accuracy of the rod contact models discussed in Section 3.3. Here, two examples are presented, each from point-to-point and line-to-line contact schemes to verify the accuracy of the implemented nonlinear rod contact models.

Point-to-point contacts

We use the large sliding problem considered in Weeger et al. [107] to verify the point-to-point contact simulations. Two equally long perpendicular cantilevers, as shown in Figure 3.17, are initially defined to be in contact ($g = 0$), and rod 2 is loaded 10 times more in global y direction compared to the loading in global z direction. The reason for this loading is twofold. First, heavy loading in y is to study the change in contact locations when the rod 1 undergoes large sliding deformation. The second reason is to check the stability of the formed contacts so that rods will neither separate nor penetrate into each other. Rod numbering is denoted by a superscript and contact force components are identified as ${}^c f_x^i, {}^c f_y^i, {}^c f_z^i$, where i is the rod number. Moreover, two cantilevers are initially in contact at the parametric coordinates $\theta^I = 0.7$ and $\theta^J = 0.5$ respectively. We use cubic B-spline basis functions to discretise each beam into 20 equal finite elements. Moreover, for each beam, the finite element mesh consist of 23 nodes including two ghost nodes.

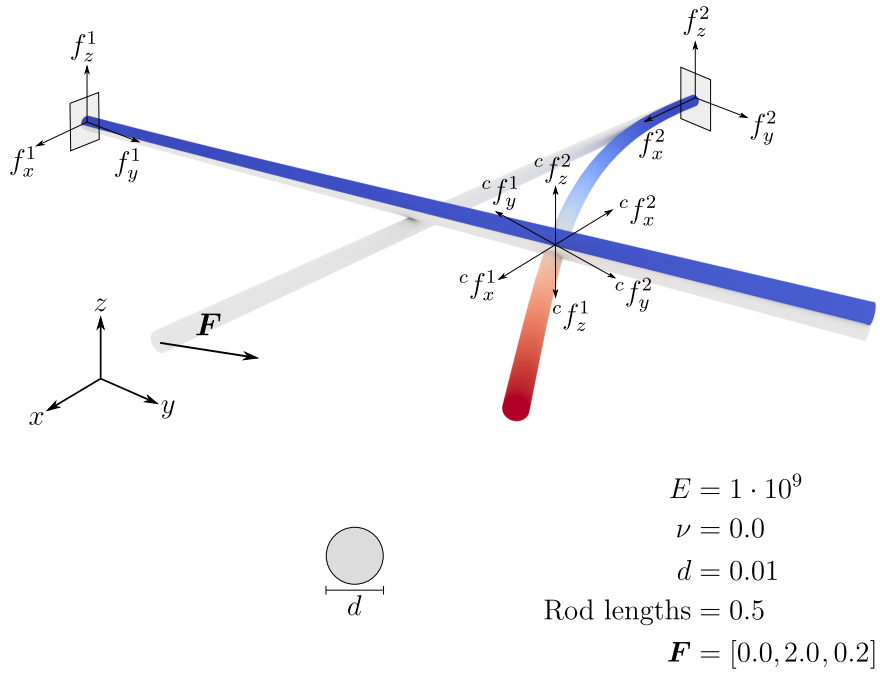


Figure 3.17: Problem description of two perpendicular cantilevers undergoing a large sliding deformation. Initial geometry is given in grey and deformed geometry is shown with its displacement magnitude contours.

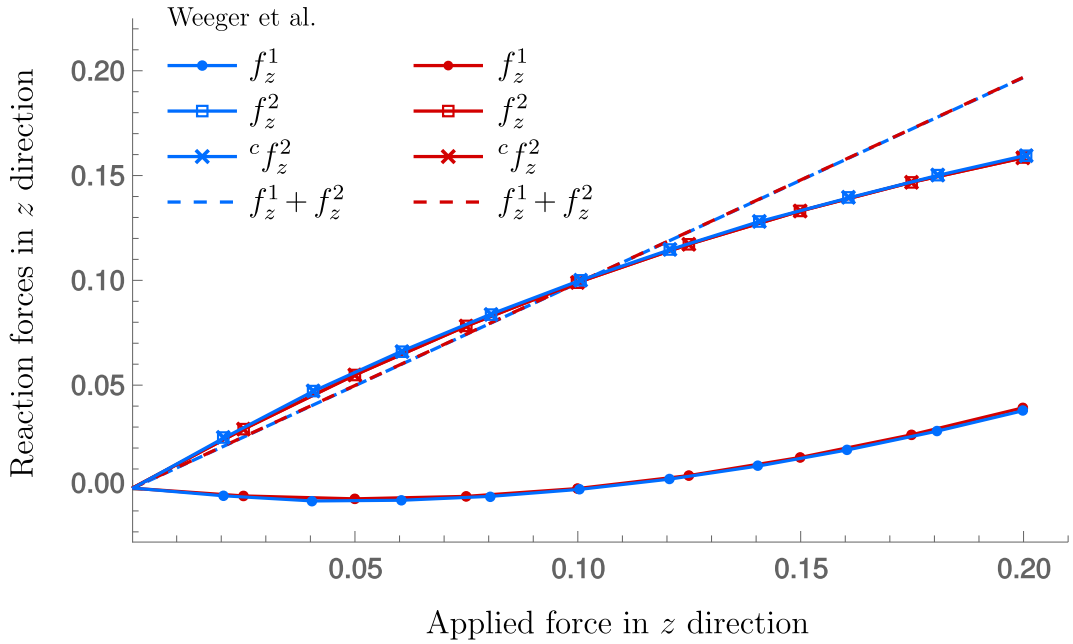


Figure 3.18: Comparison of numerical results of two perpendicular cantilevers undergoing a large sliding deformation.

Line-to-line contacts

Two, initially in contact, parallel rods problem in Meier et al. [81] is considered to verify the implemented line-to-line contact model. In this problem, two cantilever rods are twisted by angle 4π from their free ends. Here, the displacement controlled twisting makes the centres of the free ends of cantilevers to follow a circular path but line-to-line contacts enforce the rods to deform into a double helix. Free end rotations are depicted in Figure 3.19 and colour contours refer to the magnitude of displacements. This problem is provided for visual verification. Hence, details of the material and cross-section properties are not presented. Readers are invited to read Meier et al. [81] for further details. We use cubic B-spline basis functions to discretise each beam into 20 equal finite elements. Moreover, for each beam, the finite element mesh consist of 23 nodes including two ghost nodes.

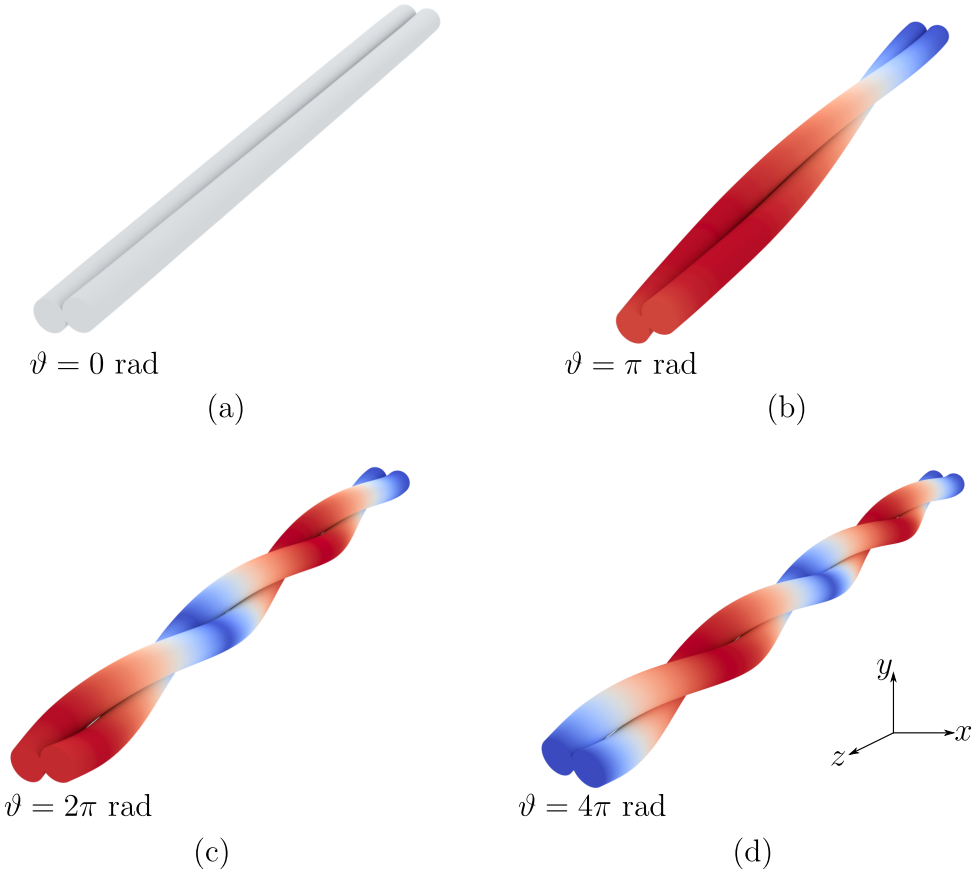


Figure 3.19: Two initially straight and parallel rods in contact. Displacement-controlled twisting into a double-helical shape.

This chapter formed the bedrock of simulating spatial rods undergoing large deformations. In this thesis, this rod model furnishes the basis of simulating microscale RVEs in Chapters 5 and 6. Next, for a twoscale homogenisation strategy, we need to implement a macroscale simulator. Considering our applications, we aim to present a Kirchhoff-Love thin-shell solver for the analysis of linear anisotropic thin-shells. Thus, Chapter 4 is reserved to supplement the rod-based microscale RVEs of Chapter 3 with macroscale thin-shells to complete the twoscale homogenisation scheme.

4 Anisotropic Kirchhoff-Love thin-shells

4.1 Preliminaries

Shells are common structural elements used in diverse engineering applications, including pressure vessels, submarine hulls, wings and fuselages of aeroplanes, structures of nuclear power plants, liquid storage tanks and many other structures. They are also found in nature in the form of skulls, eggs and sea-shells. Manifold uses of shell structures naturally posit the need for accurate shell analysis tools. These analysis tools originate from the theory of shells which can be recognised as a generalisation of plate and membrane theories [133]. In the analysis of shells, 3D continuum mechanics is reduced to its 2D variant by neglecting through the thickness deformations [56, 64, 133, 134]. Hence, the thickness of a shell is taken as a constant when integrating stress resultants along the thickness direction. Classical Plate Theory (CPT), or widely known as Kirchhoff-Love (KL) theory, formed the basis for many shell theories. KL theory, in contrast to shear deformable shell theories, assumes zero transverse shear deformations and became the tool of choice in the analysis of thin-shells. The First-order Shear Deformation Theory (FSDT) and Third-order Shear Deformation Theory (TSDT) became popular in the analysis of thick-shells where the former assumes a constant transverse shear and the latter allows the variation of transverse shear through the shell thickness. FSDT, commonly referred to as Mindlin-Reissner plate theory, and TSDT, commonly referred to as Reddy's plate theory, are applicable when the ratio between the span of plate-bending curvature and the plate thickness is approximately $20:1$ or less.

In this thesis, we use the membrane variant of the KL theory of shells because the textile applications considered in Chapter 6 can be classified as thin-membranes. Section 4.2 presents general formulations of a finite deformable shear-free thin-shell. There is a

plethora of research articles and books in relation to shear-deformable shells. Interested readers are kindly invited to refer to Long et al. [56], Reddy [133], Timoshenko [134].

4.2 Review of thin-shell mechanics

4.2.1 Kinematics of shells

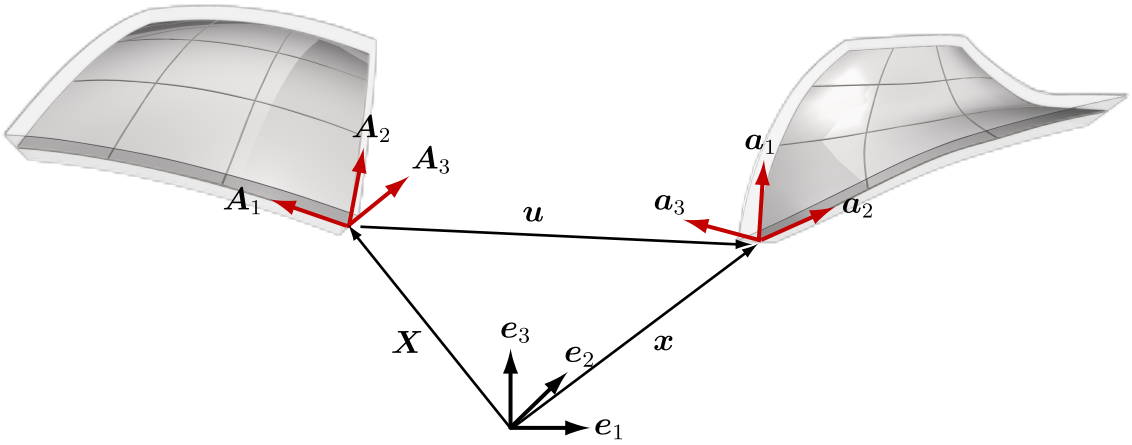


Figure 4.1: Kinematics of shells in curvilinear coordinates. $\boldsymbol{A}_i$ and $\boldsymbol{a}_i$ represents non-orthogonal covariant basis vectors in reference and deformed configurations respectively [45].

In this section, first, a summary of the kinematics and deformation of the Kirchhoff-Love shells is presented. The finite deformation of a shell is characterised by using the energy functional approach. The corresponding finite element weak form is derived by employing the principle of virtual work. This section closely follows the rigorous formulations presented by Cirak et al. [53] and Long et al. [56]. For in-depth discussions on finite deformable shell formulations using differential geometry, readers are encouraged to refer to Ciarlet [42] and Wriggers [119].

As previously mentioned, the KL theory of shells is one of the simplest theories to describe shell deformations. It is a direct extension of the Euler-Bernoulli beam theory to plates. Similar to Euler-Bernoulli beam theory, KL theory assumes that the mid-surface of the shells remain normal and that there are no through-the-thickness shear deformations. This assumption is generally valid for thin-shells [133].

Table 4.1: Notations and definitions

Greek indices	take values $1, 2$
Latin indices	take values $1, 2, 3$
$\bullet_{,i}$	comma denotes differentiation with respect to θ^i
$\boldsymbol{a}_\alpha = \dfrac{\partial \boldsymbol{x}}{\partial \theta^\alpha}$	covariant base vectors for $\theta^3 = 0$
$\boldsymbol{a}^\alpha = \dfrac{\partial \theta^\alpha}{\partial \boldsymbol{x}}$	contravariant base vectors for $\theta^3 = 0$
$\boldsymbol{g}_i = \dfrac{\partial \boldsymbol{r}}{\partial \theta^i}$	covariant base vectors
$\boldsymbol{g}^i = \dfrac{\partial \theta^i}{\partial \boldsymbol{r}}$	contravariant base vectors
$g_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j$	components of covariant metric tensor
$g^{ij} = \boldsymbol{g}^i \cdot \boldsymbol{g}^j$	components of contravariant metric tensor
$\boldsymbol{g}_i = g_{ij}\boldsymbol{g}^j$	covariant and contravariant base vectors relationship
$\boldsymbol{g}^i = g^{ij}\boldsymbol{g}_j$	summed over $j = 1, 2, 3$
$\boldsymbol{a}_3 = \dfrac{\boldsymbol{a}_1 \times \boldsymbol{a}_2}{\lvert \boldsymbol{a}_1 \times \boldsymbol{a}_2 \rvert}$	orthonormal vector to the shell mid-surface

In KL formulation of shells, the geometry and kinematics of the shell are approximated by considering only the shell mid-surface. Figure 4.1 summarises the kinematic description of the shell in the curvilinear coordinates. For notation consistency, this section uses uppercase and lowercase letters to represent quantities in the reference and deformed configurations, respectively. In Figure 4.1, $\boldsymbol{A}_i$ and $\boldsymbol{a}_i$ denote the orthogonal covariant basis vectors; $\boldsymbol{e}_i$ represents the orthonormal Cartesian basis; and $\boldsymbol{X}$ and $\boldsymbol{x}$ are position vectors of an arbitrary mid-surface material point. Then, the position vector $\boldsymbol{r}$ of points belonging to the shell of thickness t is written as,

$$\boldsymbol{R}(\theta^1, \theta^2, \theta^3) = \boldsymbol{X}(\theta^1, \theta^2) + \theta^3 \boldsymbol{A}_3(\theta^1, \theta^2), \qquad -t/2 \leq \theta^3 \leq t/2 \tag{4.1a}$$

$$\boldsymbol{r}(\theta^1, \theta^2, \theta^3) = \boldsymbol{x}(\theta^1, \theta^2) + \theta^3 \boldsymbol{a}_3(\theta^1, \theta^2), \qquad -t/2 \leq \theta^3 \leq t/2\ , \tag{4.1b}$$

where $\boldsymbol{A}_3$ and $\boldsymbol{a}_3$ are the unit normal vector of the shell and $\boldsymbol{\theta} = (\theta^1, \theta^2, \theta^3)$ is a parametric coordinate. Using the kinematic definitions in Table 4.1, deformation gradient $\boldsymbol{F}$

can be written as,

$$\boldsymbol{F} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{R}} = \frac{\partial \boldsymbol{r}}{\partial \theta^i}\frac{\partial \theta^i}{\partial \boldsymbol{R}} = \boldsymbol{g}_i \otimes \boldsymbol{G}^i \ . \tag{4.2}$$

Green-Lagrange strain tensor E_{ij} in curvilinear coordinates is defined as,

$$\begin{aligned} \boldsymbol{E} &= \frac{1}{2}(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} - \mathbf{I}) \\ &= \frac{1}{2}(g_{ij} - G_{ij})\boldsymbol{G}^i \otimes \boldsymbol{G}^j \\ &= E_{ij}\boldsymbol{G}^i \otimes \boldsymbol{G}^j \ , \end{aligned} \tag{4.3}$$

where the covariant metric tensors g_{ij} and G_{ij} are defined as,

$$g_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j, \qquad G_{ij} = \boldsymbol{G}_i \cdot \boldsymbol{G}_j, \tag{4.4a}$$

$$\boldsymbol{g}_\alpha = \frac{\partial \boldsymbol{r}}{\partial \theta^\alpha} = \boldsymbol{a}_\alpha + \theta^3 \boldsymbol{a}_{3,\alpha} \ , \qquad \boldsymbol{G}_\alpha = \frac{\partial \boldsymbol{R}}{\partial \theta^\alpha} = \boldsymbol{A}_\alpha + \theta^3 \boldsymbol{A}_{3,\alpha} \tag{4.4b}$$

$$\boldsymbol{x}_{,\alpha} = \boldsymbol{a}_\alpha = \frac{\partial \boldsymbol{x}}{\partial \theta^\alpha}, \qquad \boldsymbol{X}_{,\alpha} = \boldsymbol{A}_\alpha = \frac{\partial \boldsymbol{X}}{\partial \theta^\alpha} \ . \tag{4.4c}$$

For KL thin-shells, following kinematic relations hold in both configurations.

$$\boldsymbol{a}_3 = \frac{\boldsymbol{a}_1 \times \boldsymbol{a}_2}{||\boldsymbol{a}_1 \times \boldsymbol{a}_2||} \ , \qquad \boldsymbol{a}_3 \cdot \boldsymbol{a}_3 = 1 \ , \qquad \boldsymbol{a}_\alpha \cdot \boldsymbol{a}_3 = 0 \ . \tag{4.5}$$

Using the relations in Equations (4.3), (4.4) and (4.5) and dropping the higher-order terms, the Green-Lagrange strain tensor is written as,

$$\begin{aligned} E_{\alpha\beta} &= \alpha_{\alpha\beta} + \theta^3 \beta_{\alpha\beta} \\ \alpha_{\alpha\beta} &= \frac{1}{2}(\boldsymbol{a}_\alpha \cdot \boldsymbol{a}_\beta - \boldsymbol{A}_\alpha \cdot \boldsymbol{A}_\beta) \\ \beta_{\alpha\beta} &= \boldsymbol{a}_\alpha \cdot \boldsymbol{a}_{3,\beta} - \boldsymbol{A}_\alpha \cdot \boldsymbol{A}_{3,\beta} \ , \end{aligned} \tag{4.6}$$

where $\alpha_{\alpha\beta}$ and $\beta_{\alpha\beta}$ are the membrane and bending strains respectively. The simplified expression for the Green-Lagrange strain tensor in Equation (4.6) implies that the deformation of a shell is completely characterised by the deformation of the shell mid-surface. Moreover, $\alpha_{\alpha\beta}$ represents the stretching of the shell mid-surface, i.e. in-plane strains and $\beta_{\alpha\beta}$ captures the change of shell curvature, i.e. out-of-plane strains. Strains $E_{\alpha\beta}$ furnish the kinematic description to define the plane stress state of a thin-shell. Therefore, terms $E_{\alpha 3}, E_{3\alpha}$ naturally vanish to satisfy the plane stress state.

Now we write the deformed configuration of a shell given by,

$$\boldsymbol{x}(\theta^1, \theta^2) = \boldsymbol{X}(\theta^1, \theta^2) + \boldsymbol{u}(\theta^1, \theta^2) \,, \tag{4.7}$$

where $\boldsymbol{u}$ represents the spatial displacement of the shell mid-surface. Differentiating Equation (4.7) with respect to θ^α and substituting the resulting expressions for $\boldsymbol{x}_{,\alpha} \equiv \boldsymbol{a}_\alpha$ in Equation (4.6) leads to,

$$\alpha_{\alpha\beta} = \frac{1}{2}(\boldsymbol{u}_{,\alpha}.\boldsymbol{A}_\beta + \boldsymbol{A}_\alpha \cdot \boldsymbol{u}_{,\beta}) \tag{4.8a}$$

$$\begin{aligned}\beta_{\alpha\beta} = &-\boldsymbol{u}_{,\alpha\beta} \cdot \boldsymbol{A}_3 + \\ &\frac{1}{||\boldsymbol{A}_1 \times \boldsymbol{A}_2||}[\boldsymbol{u}_{,1} \cdot (\boldsymbol{A}_{\alpha,\beta} \times \boldsymbol{A}_2) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{A}_1 \times \boldsymbol{A}_{\alpha,\beta})] + \\ &\frac{\boldsymbol{a}_3 \cdot \boldsymbol{a}_{\alpha,\beta}}{||\boldsymbol{A}_1 \times \boldsymbol{A}_2||}[\boldsymbol{u}_{,1} \cdot (\boldsymbol{A}_2 \times \boldsymbol{A}_3) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{A}_3 \times \boldsymbol{A}_1)] \,.\end{aligned} \tag{4.8b}$$

For additional details of derivations, see Ciarlet [42]. The transformed expression, in Equation (4.6), for $\alpha_{\alpha\beta}$ and $\beta_{\alpha\beta}$, and hence the Green-Lagrange strain tensor $E_{\alpha\beta}$ is now expressed in terms of the *known* reference configuration and the *unknown* displacements which need to be solved. This is evident as the strain components in Equation (4.8) are furnished by the derivatives of displacements and covariant basis vectors of the reference configuration. To avoid tedious equations, the term $\boldsymbol{a}_3 \cdot \boldsymbol{a}_{\alpha,\beta}$ in Equation (4.8b) is left unaltered.

4.2.2 Discretisation of the shell energy functional

In this section, we aim to arrive at the discretised weak form of the shell at equilibrium. This is achieved by first deriving the weak form of a shell using the principle of virtual work. Constitutive relations are then integrated to relate the stresses to strains. Lastly, we discretise the shell weak form in a linearised algebraic domain.

Weak form of equilibrium equations

We consider the total potential energy of a hyperelastic shell to derive the shell equilibrium equations in the weak form, given by,

$$\Pi(\boldsymbol{r}) = \Pi_{\text{int}}(\boldsymbol{r}) + \Pi_{\text{ext}}(\boldsymbol{r}) \ , \tag{4.9}$$

where Π_{int} is the internal and Π_{ext} is the external potential energy. At equilibrium, the first variation of the potential energy is zero, which can be written concisely as,

$$\delta\Pi = \delta\Pi_{\text{int}} + \delta\Pi_{\text{ext}} = 0 \ . \tag{4.10}$$

The internal potential energy is the integral of the strain energy density W. For a hyperelastic material, strain energy density is a function of the Green-Lagrange strain tensor $\boldsymbol{E}$ [116]. Therefore, we can write the internal potential energy as,

$$\Pi_{\text{int}} = \int_{\Omega}\int_{-\frac{t}{2}}^{\frac{t}{2}} W(\boldsymbol{E})\,\bar{j}\ \mathrm{d}\theta^3\,\mathrm{d}\Omega \ . \tag{4.11}$$

Here, the Jacobian $\bar{j}$, takes into account the curvature of the shell in the integration across the thickness and takes the form,

$$\bar{j} = \frac{|(\boldsymbol{G}_1 \times \boldsymbol{G}_2)\cdot\boldsymbol{G}_3|}{|(\boldsymbol{A}_1 \times \boldsymbol{A}_2)\cdot\boldsymbol{A}_3|} \ . \tag{4.12}$$

The variation of the internal energy can be derived using the identity $\delta(\bullet) = \frac{\partial(\bullet)}{\partial(\star)}\delta(\star) + \frac{\partial(\bullet)}{\partial(*)}\delta(*) + \cdots$. And knowing that for a hyperelastic material internal energy is a function of $\boldsymbol{E}$, we write,

$$\delta\Pi_{\text{int}} = \int_{\Omega}\int_{-\frac{t}{2}}^{\frac{t}{2}} \frac{\partial W(\boldsymbol{E})}{\partial \boldsymbol{E}} : \delta\boldsymbol{E}\bar{j}\ \mathrm{d}\theta^3\,\mathrm{d}\Omega = \int_{\Omega}\int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{S} : \delta\boldsymbol{E}\bar{j}\ \mathrm{d}\theta^3\,\mathrm{d}\Omega \ , \tag{4.13}$$

where $\boldsymbol{S}$ is the second Piola-Kirchhoff stress tensor defined for plane stress conditions. After introducing the Green-Lagrange strain in Equation (4.6), the preceding equation takes the form,

$$\delta\Pi_{\text{int}} = \int_{\Omega}\int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{S} : (\delta\boldsymbol{\alpha} + \theta^3\delta\boldsymbol{\beta})\bar{j}\ \mathrm{d}\theta^3\,\mathrm{d}\Omega \ . \tag{4.14}$$

Next, the membrane (or stretching) stress and moment resultants are defined by further simplifying the Equation (4.14). These can be written as,

$$\boldsymbol{n} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{S}\,\overline{j}\;\mathrm{d}\theta^3\;, \tag{4.15a}$$

$$\boldsymbol{m} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \theta^3 \boldsymbol{S}\,\overline{j}\;\mathrm{d}\theta^3\;. \tag{4.15b}$$

Equations (4.15) and (4.14) enable us to write the variation of the internal energy as an integral over the shell mid-surface as,

$$\delta\Pi_{\text{int}} = \int_{\Omega} (\boldsymbol{n} : \delta\boldsymbol{\alpha} + \boldsymbol{m} : \delta\boldsymbol{\beta})\;\mathrm{d}\Omega\;. \tag{4.16}$$

Recall from Equation (4.7) that the independent variable in the thin-shell model is the mid-surface position vector $\boldsymbol{x}$. Therefore, the variation of the internal energy with respect to $\boldsymbol{x}$ is given by,

$$\delta\Pi_{\text{int}} = \int_{\Omega} \left(\boldsymbol{n} : \frac{\partial\boldsymbol{\alpha}}{\partial\boldsymbol{x}} + \boldsymbol{m} : \frac{\partial\boldsymbol{\beta}}{\partial\boldsymbol{x}}\right) \cdot \delta\boldsymbol{x}\;\mathrm{d}\Omega\;. \tag{4.17}$$

Next, we derive the variation of the external potential energy of the shell. The boundary of the shell consists of the top ($\theta^3 = t/2$), bottom ($\theta^3 = -t/2$) and lateral surfaces (Γ) of which, for brevity, only the lateral surface $[-t/2,\, t/2] \times \Gamma$ is considered. The external potential energy of the shell with the external body force vector $\boldsymbol{b}$ and the external lateral surface traction vector $\boldsymbol{f}$ is given by,

$$\Pi_{\text{ext}} = -\int_{\Omega}\int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{b}\cdot(\boldsymbol{x} + \theta^3\boldsymbol{a}_3)\overline{j}\;\mathrm{d}\theta^3\,\mathrm{d}\Omega - \int_{\Gamma}\int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{f}\cdot(\boldsymbol{x} + \theta^3\boldsymbol{a}_3)\overline{j}\;\mathrm{d}\theta^3\,\mathrm{d}\Gamma\;, \tag{4.18}$$

where the kinematic assumption in Equation (4.1b) has been used. Furthermore, simplifying Equation (4.18), we arrive at the following external force ($\boldsymbol{p}$, $\boldsymbol{r}$) and moment

resultants ($\boldsymbol{q},\ \boldsymbol{s}$).

$$\boldsymbol{p} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{b}\bar{j}\ \mathrm{d}\theta^3 \qquad \boldsymbol{q} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \theta^3 \boldsymbol{b}\bar{j}\ \mathrm{d}\theta^3 \tag{4.19a}$$

$$\boldsymbol{r} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \boldsymbol{f}\bar{j}\ \mathrm{d}\theta^3 \qquad \boldsymbol{s} = \int_{-\frac{t}{2}}^{\frac{t}{2}} \theta^3 \boldsymbol{f}\bar{j}\ \mathrm{d}\theta^3\ . \tag{4.19b}$$

Thus, the variation of the external energy with respect to the mid-surface position $\boldsymbol{x}$ can be expressed as,

$$\delta\Pi_{\text{ext}} = -\int_{\Omega} \left(\boldsymbol{p} + \boldsymbol{q}\frac{\partial \boldsymbol{a}_3}{\partial \boldsymbol{x}}\right) \cdot \delta\boldsymbol{x}\ \mathrm{d}\Omega - \int_{\Gamma} \left(\boldsymbol{r} + \boldsymbol{s}\frac{\partial \boldsymbol{a}_3}{\partial \boldsymbol{x}}\right) \cdot \delta\boldsymbol{x}\ \mathrm{d}\Gamma\ . \tag{4.20}$$

Summation of the variations of the internal (4.17) and external (4.20) energy eventually yields the weak form of the shell equilibrium equation,

$$\left(\frac{\partial \Pi_{\text{int}}(\boldsymbol{x})}{\partial \boldsymbol{x}} + \frac{\partial \Pi_{\text{ext}}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) \cdot \delta\boldsymbol{x} = 0\ . \tag{4.21}$$

Equilibrium Equation (4.21) is also referred to as the principle of virtual work or principle of minimum potential energy.

Constitutive equations

We consider the generalised linear anisotropic material constitutive model in the Voigt notations as,

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} \mathcal{C}^{11} & \mathcal{C}^{12} & \mathcal{C}^{13} & \mathcal{C}^{14} & \mathcal{C}^{15} & \mathcal{C}^{16} \\ & \mathcal{C}^{22} & \mathcal{C}^{23} & \mathcal{C}^{24} & \mathcal{C}^{25} & \mathcal{C}^{26} \\ & & \mathcal{C}^{33} & \mathcal{C}^{34} & \mathcal{C}^{35} & \mathcal{C}^{36} \\ & & & \mathcal{C}^{44} & \mathcal{C}^{45} & \mathcal{C}^{46} \\ & \text{sym.} & & & \mathcal{C}^{55} & \mathcal{C}^{56} \\ & & & & & \mathcal{C}^{66} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ 2\epsilon_4 \\ 2\epsilon_5 \\ 2\epsilon_6 \end{bmatrix}\ , \tag{4.22}$$

where the Voigt indices $1 = 11$, $2 = 22$, $3 = 33$, $4 = 23$, $5 = 13$ and $6 = 12$ expands Equation (4.22) to the standard fourth order tensor notation. For instance, fourth order tensorial quantity $\mathcal{C}^{1111}$ is written as $\mathcal{C}^{11}$ following the Voigt notation [136].

Equation (4.22) is commonly referred to as the generalised Hooke's law [71]. It linearly

maps strains (ϵ_{kl}) to stresses (σ_{ij}) using a linear material constitutive model $\mathcal{C}^{ijkl}$. Hence we can expand Equation (4.22) in index notations as,

$$\sigma_{ij} = \mathcal{C}^{ijkl}\epsilon_{kl} \ . \tag{4.23}$$

A tensor of any order, has its components attached to a tensorial basis. We can write the fourth-order material tensor $\boldsymbol{\mathcal{C}}$ in the orthonormal Cartesian basis $\boldsymbol{e}_i$ as,

$$\boldsymbol{\mathcal{C}} = \mathcal{C}^{ijkl}\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l \ . \tag{4.24}$$

But the same tensor can be written with respect to a different orthonormal basis $\bar{\boldsymbol{e}}_i$ as,

$$\boldsymbol{\mathcal{C}} = \mathcal{C}^{ijkl}\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l = \bar{\mathcal{C}}^{pqrs}\bar{\boldsymbol{e}}_p \otimes \bar{\boldsymbol{e}}_q \otimes \bar{\boldsymbol{e}}_r \otimes \bar{\boldsymbol{e}}_s \ . \tag{4.25}$$

Evidently, we can transform Equation (4.25) using standard tensor algebra to express the components $\bar{\mathcal{C}}^{pqrs}$ as,

$$\begin{aligned}
\bar{\mathcal{C}}^{pqrs} &= (\bar{\boldsymbol{e}}^p \otimes \bar{\boldsymbol{e}}^q) : \boldsymbol{\mathcal{C}} : (\bar{\boldsymbol{e}}^r \otimes \bar{\boldsymbol{e}}^s) \\
&= (\bar{\boldsymbol{e}}^p \otimes \bar{\boldsymbol{e}}^q) : \mathcal{C}^{ijkl}\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \otimes \boldsymbol{e}_l : (\bar{\boldsymbol{e}}^r \otimes \bar{\boldsymbol{e}}^s) \\
&= (\bar{\boldsymbol{e}}^p \otimes \bar{\boldsymbol{e}}^q) : \mathcal{C}^{ijkl}\boldsymbol{e}_i \otimes \boldsymbol{e}_j(\boldsymbol{e}_k \cdot \bar{\boldsymbol{e}}^r)(\boldsymbol{e}_l \cdot \bar{\boldsymbol{e}}^s) \\
&= \mathcal{C}^{ijkl}(\boldsymbol{e}_i \cdot \bar{\boldsymbol{e}}^p)(\boldsymbol{e}_j \cdot \bar{\boldsymbol{e}}^q)(\boldsymbol{e}_k \cdot \bar{\boldsymbol{e}}^r)(\boldsymbol{e}_l \cdot \bar{\boldsymbol{e}}^s)
\end{aligned} \tag{4.26}$$

where $\bar{\boldsymbol{e}}^p$ is the dual vector of $\bar{\boldsymbol{e}}_p$ [135]. Equations (4.26) reveal a pivotal basis conversion which is used to map material tensor from global Cartesian coordinates to local covariant base vectors as shown in Figure 4.1.

To this end, using Equations (4.3) and (4.25) we can derive the second Piola-Kirchhoff stress tensor for small strains, in the convected coordinates as,

$$\boldsymbol{S} = \boldsymbol{\mathcal{C}} : \boldsymbol{E} \tag{4.27}$$

$$\begin{aligned}
&= [\hat{\mathcal{C}}^{ijkl}\boldsymbol{G}_i \otimes \boldsymbol{G}_j \otimes \boldsymbol{G}_k \otimes \boldsymbol{G}_l] : [E_{kl}\boldsymbol{G}^k \otimes \boldsymbol{G}^l] \\
&= \hat{\mathcal{C}}^{ijkl}E_{kl}\boldsymbol{G}_i \otimes \boldsymbol{G}_j(\boldsymbol{G}_k \cdot \boldsymbol{G}^k)(\boldsymbol{G}_l \cdot \boldsymbol{G}^l) \\
&= \hat{\mathcal{C}}^{ijkl}E_{kl}\boldsymbol{G}_i \otimes \boldsymbol{G}_j \\
&= S^{ij}\boldsymbol{G}_i \otimes \boldsymbol{G}_j \ ,
\end{aligned} \tag{4.28}$$

where the dual vector identity $\boldsymbol{G}_k \cdot \boldsymbol{G}^k = \delta^k_k = 1$ in Table 4.1 has been used to simplify

the terms. Also, Equation (4.28) provides the second Piola-Kirchhoff stress tensor in the covariant basis.

Next, using Equations (4.25) and (4.26) we express the material tensor to facilitate the constitutive relation in convected coordinates as,

$$\hat{\mathcal{C}}^{pqrs} = \mathcal{C}^{ijkl}(\boldsymbol{e}_i \cdot \boldsymbol{G}^p)(\boldsymbol{e}_j \cdot \boldsymbol{G}^q)(\boldsymbol{e}_k \cdot \boldsymbol{G}^r)(\boldsymbol{e}_l \cdot \boldsymbol{G}^s) \ . \tag{4.29}$$

At this juncture, we have gathered material tensor $\boldsymbol{\mathcal{C}}$ and kinematic relations $\boldsymbol{S}$, $\boldsymbol{E}$ in convected coordinates for a general linear anisotropic material. However, it is not convenient to operate on the implicit form of the material tensor as given in Equation (4.29). Therefore, first, we introduce an isotropic material to derive an explicit expression to Equation (4.29). We write the material tensor for an isotropic solid [136] using the Voigt notation as,

$$\boldsymbol{\mathcal{C}} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ & 1-\nu & \nu & 0 & 0 & 0 \\ & & 1-\nu & 0 & 0 & 0 \\ & & & 1-2\nu & 0 & 0 \\ & \text{sym.} & & & 1-2\nu & 0 \\ & & & & & 1-2\nu \end{bmatrix}, \tag{4.30}$$

where E and ν are the Young's modulus and Poisson's ratio, respectively. Components in Equation (4.30) can be written in a compact form as,

$$\mathcal{C}^{ijkl} = \lambda \delta^{ij}\delta^{kl} + \mu(\delta^{ik}\delta^{jl} + \delta^{il}\delta^{jk}), \tag{4.31}$$

where $\lambda = \dfrac{E\nu}{(1+\nu)(1-2\nu)}$ and $\mu = \dfrac{E}{2(1+\nu)}$ are the Lamé material parameters [136].

Now Equation (4.31) is substituted in Equation (4.29) to arrive at the material tensor components in convected coordinates as given in Equation (4.32). Detailed algebraic formulations are not presented here. Interested readers are kindly referred to Ciarlet et al. [42] and Itskov [137].

$$\hat{\mathcal{C}}^{ijkl} = \lambda G^{ij}G^{kl} + \mu(G^{ik}G^{jl} + G^{il}G^{jk}) \ . \tag{4.32}$$

For thin-shells, it is reasonable to assume plane stress conditions during deformation [55].

Therefore, material components in Equation (4.32) are further simplified, respecting plane stress conditions as,

$$\hat{\mathcal{C}}^{\alpha\beta\gamma\delta} = \frac{E\nu}{(1-\nu^2)} G^{\alpha\beta} G^{\gamma\delta} + \frac{E}{2(1+\nu)} (G^{\alpha\gamma} G^{\beta\delta} + G^{\alpha\delta} G^{\beta\gamma}) \ . \tag{4.33}$$

Discretisation of the shell weak form

The proposed kinematics, in Equations (4.8), contains the second-order derivatives of the deformed shell mid-surface. Hence, smooth, or more formally H^2-conforming [56], shape functions are necessary for a robust finite element discretisation. In this thesis, we use subdivision surfaces for the discretisation of the shell weak form. Further details on the subdivision surfaces can be found at Farin et al. [66]. Moreover, additional information regarding the isogeometric analysis implementations can be found at Cirak et al. [53,55,138]. Adhering to aforementioned prerequisites, we can write the discretised shell mid-surface configurations and displacement degrees of freedom as,

$$\boldsymbol{X}(\theta^1, \theta^2) = \sum_{I=1}^{NSF} N^I(\theta^1, \theta^2) \boldsymbol{X}_I \tag{4.34a}$$

$$\boldsymbol{x}(\theta^1, \theta^2) = \sum_{I=1}^{NSF} N^I(\theta^1, \theta^2) \boldsymbol{x}_I \tag{4.34b}$$

$$\boldsymbol{u}(\theta^1, \theta^2) = \sum_{I=1}^{NSF} N^I(\theta^1, \theta^2) \boldsymbol{u}_I \ , \tag{4.34c}$$

where NSF is the number of vertices in one-neighbourhood of an element and N^I are the subdivision shape functions.

In the discretised domain we can write the discretised residual force vector $\hat{\boldsymbol{f}}^I_{\text{res}}$ using Equation (4.21) for an arbitrary $\delta\boldsymbol{x}_I$ as,

$$\hat{\boldsymbol{f}}^I_{\text{res}} = \left(\sum_{K=1}^{NEL} \frac{\partial \Pi^K_{\text{int}}}{\partial \boldsymbol{x}_I} + \sum_{K=1}^{NEL} \frac{\partial \Pi^K_{\text{ext}}}{\partial \boldsymbol{x}_I} \right) = \boldsymbol{0} \ , \tag{4.35}$$

where NEL denotes the total number of elements in the discretised shell domain. Moreover, the foregoing summations in Equation (4.35) imply the presence of a mapping

between the global (mesh-wide) and local (element-specific) vertex numbering. For example, the internal force contribution of an element K to a vertex I is computed with,

$$\hat{\boldsymbol{f}}_{\text{int}}{}^{IK} = \int_{\Omega_K} \left(\boldsymbol{n} : \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{x}_I} + \boldsymbol{m} : \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{x}_I} \right) \mathrm{d}\Omega \ . \tag{4.36}$$

To this end, we can replace Equations (4.27), (4.33) in Equation (4.15) to explicitly write the internal stress resultants as,

$$\boldsymbol{n} = \frac{Et}{1-\nu^2} \mathbf{H} \boldsymbol{\alpha} \tag{4.37a}$$

$$\boldsymbol{m} = \frac{Et^3}{12(1-\nu^2)} \mathbf{H} \boldsymbol{\beta} \ , \tag{4.37b}$$

where $\mathbf{H}$ takes the form,

$$\mathbf{H} = \begin{bmatrix} (G^{11})^2 & \nu G^{11} G^{22} + (1-\nu)(G^{12})^2 & G^{11} G^{12} \\ & (G^{22})^2 & G^{22} G^{12} \\ \text{sym.} & & \frac{1}{2}[(1-\nu) G^{11} G^{22} + (1+\nu)(G^{12})^2] \end{bmatrix} . \tag{4.38}$$

It should be noted that Voigt notation has been used in vectorising the stress resultants and strains in Equation (4.37).

A standard Newton-Raphson iterative solver is used to find the equilibrium solution of the discretised shell. This solution process closely follows the steps given in Section 3.2.2, hence not repeated herein. For detailed and rigorous analytical expressions for Hessian terms, readers are encouraged to refer to the doctoral theses of Majeed M. [45] and Long Q. [64].

4.3 Linear material anisotropy of thin-shells

The motivation for presenting the formulations in Section 4.2 is twofold. First, the principle of virtual work leads to analytically derive the internal forces of a hyperelastic material, as given in Equation (4.17). This derivation, however, comes one step before assigning a material constitutive model to the system. As shown in Equation (4.29), we can incorporate different material models to the structural system. Such incorporation can be directly exploited to consolidate the linear material response from computational homogenisation of thin-shell structures.

Secondly, and most importantly, this step would be the entry point of the data-driven homogenised material response, arriving from data-driven multiscale modelling, explained in Section 5.3.

4.3.1 Orthotropic material symmetry

A material is said to be orthotropic if it is characterised by symmetry with respect to three mutually-orthogonal planes, by reflections from which material properties remain unchanged. The axes normal to these planes are called principal material directions. This leads to 9 independent material parameters to uniquely define an orthotropic solid. The set of all orthogonal mappings that do not violate the material symmetry forms a group referred to as symmetry group [139,140]. Let $\boldsymbol{l}_i \cdot \boldsymbol{l}_j = \delta_{ij}$ be unit base vectors in the principal material directions. Then, the orthotropic symmetry group can be described by structural tensors defined as,

$$\boldsymbol{L}_i = \boldsymbol{l}_i \otimes \boldsymbol{l}_i, \qquad i = 1,2,3. \tag{4.39}$$

Using the above structural tensors $\boldsymbol{L}_i$ along with the St.Venant-Kirchhoff material parameters a_{ij} $(a_{ij} = a_{ji},\ i \neq j)$ and b_{ij} $(b_{ij} = b_{ji},\ i \neq j)$, we can define the hyperelastic strain energy W as a function of the Green-Lagrange strain tensor as [137],

$$W(\boldsymbol{E}) = \frac{1}{2} \sum_{i,j}^{3} a_{ij} \operatorname{Tr}(\boldsymbol{E}\boldsymbol{L}_i) \operatorname{Tr}(\boldsymbol{E}\boldsymbol{L}_j) + \sum_{i,j\neq i}^{3} b_{ij} \operatorname{Tr}(\boldsymbol{E}\boldsymbol{L}_i \boldsymbol{E}\boldsymbol{L}_j) \ . \tag{4.40}$$

One could interpret the terms $\boldsymbol{E}\boldsymbol{L}_i$ as oriented strain tensors along the material directions. The constitutive relations and elastic moduli corresponding to the strain energy function, in Equation (4.40), are of the form,

$$\boldsymbol{S} = \frac{\partial W}{\partial \boldsymbol{E}} = \sum_{i,j}^{3} a_{ij} \operatorname{Tr}(\boldsymbol{E}\boldsymbol{L}_i)\boldsymbol{L}_j + 2 \sum_{i,j\neq i}^{3} b_{ij} \boldsymbol{L}_i \boldsymbol{E} \boldsymbol{L}_j, \tag{4.41}$$

$$\boldsymbol{\mathcal{C}} = \frac{\partial^2 W}{\partial \boldsymbol{E}^2} = \sum_{i,j}^{3} a_{ij} \boldsymbol{L}_i \times \boldsymbol{L}_j + 2 \sum_{i,j\neq i}^{3} b_{ij} (\boldsymbol{L}_i \otimes \boldsymbol{L}_j)^S \ . \tag{4.42}$$

where $(\bullet)^S$ denotes a symmetrisation operator on fourth-order tensors [137] and operators $\times$ and $\otimes$ in Equation (4.42) are defined for second-order tensors $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$

as,

$$(\boldsymbol{A} \times \boldsymbol{B}) : \boldsymbol{C} = (\boldsymbol{B} : \boldsymbol{C})\boldsymbol{A} \qquad \boldsymbol{A} \otimes \boldsymbol{B} : \boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}\boldsymbol{C} \ . \tag{4.43}$$

Material parameters a_{ij} and b_{ij} can be easily understood when we overlap the principal material directions $\boldsymbol{l}_i$ with the Cartesian base vectors $\boldsymbol{e}_i$, i.e. $\boldsymbol{l}_i = \boldsymbol{e}_i$. Then we obtain the text-book definition of an orthotropic solid constitutive model given by,

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} E_1\dfrac{1-\nu_{23}\nu_{32}}{\Delta} & E_1\dfrac{\nu_{12}+\nu_{32}\nu_{13}}{\Delta} & E_3\dfrac{\nu_{31}+\nu_{32}\nu_{21}}{\Delta} & 0 & 0 & 0 \\ & E_2\dfrac{1-\nu_{31}\nu_{13}}{\Delta} & E_2\dfrac{\nu_{23}+\nu_{13}\nu_{21}}{\Delta} & 0 & 0 & 0 \\ & & E_3\dfrac{1-\nu_{12}\nu_{21}}{\Delta} & 0 & 0 & 0 \\ & & & b_{23} & 0 & 0 \\ \text{sym.} & & & & b_{13} & 0 \\ & & & & & b_{12} \end{bmatrix} , \tag{4.44}$$

where $\Delta = (1-\nu_{12}\nu_{21}-\nu_{32}\nu_{23}-\nu_{13}\nu_{31}-2\nu_{12}\nu_{23}\nu_{31})$. We identify 9 independent material parameters [136]; three Young's moduli ($E_1,\ E_2,\ E_3$), three Shear moduli ($b_{23},\ b_{13},\ b_{12}$) and three Poisson's ratios ($\nu_{23},\ \nu_{13},\ \nu_{12}$). Moreover, we can relate the material parameters a_{ij} in Equation (4.40) by comparing entries in Equation (4.44) as $a_{ij} = \mathcal{C}^{ij}$ for $i, j = 1, 2, 3$.

Recalling an orthotropic material is a generalised representation of a transversely isotropic and an isotropic material, we can easily derive the following expressions using Equations (4.42) and (4.44).

Isotropic materials

For isotropic materials, we set $E_1 = E_2 = E_3 = E$, $\nu_{23} = \nu_{13} = \nu_{12} = \nu$ and obtain $b_{23} = b_{13} = b_{12} = \frac{E}{2(1+\nu)}$. This relation was derived earlier in Section 4.2.2. Therefore, to avoid repetition, detailed formulations will not be presented here. However, expressions for transversely isotropic material are presented as following.

Transversely isotropic materials

In the case of transversely isotropic material symmetry, all directions orthogonal to the principal material one ($\boldsymbol{l}_1$) become equivalent. This equivalence can be taken into

account by setting,

$$E_2 = E_3, \qquad \nu_{12} = \nu_{13}, \qquad \nu_{21} = \nu_{31}, \qquad b_{12} = b_{13}, \qquad b_{23} = \frac{E_2}{2(1+\nu_{23})}. \tag{4.45}$$

where the five independent material parameters can be identified as $E_1,\ E_2,\ \nu_{12},\ b_{12},\ b_{23}$. Interestingly, the same constitutive equation is presented in a compact form in the basis $\boldsymbol{e}_i$ [141]

$$\begin{aligned} \mathcal{C}^{pqrs} = {} & \lambda\delta^{pq}\delta^{rs} + \mu_T\left(\delta^{pr}\delta^{qs} + \delta^{ps}\delta^{qr}\right) + \eta_1\left(\delta^{pq}l^r l^s + \delta^{rs}l^p l^q\right) \\ & + \left(\mu_L - \mu_T\right)\left(\delta^{qs}l^p l^r + \delta^{qr}l^p l^s + \delta^{ps}l^q l^r + \delta^{pr}l^q p^s\right) + \eta_2 l^p l^q l^r l^s \end{aligned} \tag{4.46}$$

where principal material direction one is expressed dropping the index as $\boldsymbol{l}(=\boldsymbol{l}_1)$. Moreover, l^p refers to the p^{th} component of the vector $\boldsymbol{l}$. Five independent material parameters are identified as $\lambda, \mu_T, \mu_L, \eta_1, \eta_2$. To get a clearer picture we can express Equation (4.46) in the Cartesian basis by writing $\boldsymbol{l} = \boldsymbol{e}_1 = [1, 0, 0]$.

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} (\lambda - 2\mu_T + 4\mu_L + 2\eta_1 + \eta_2) & \lambda + \eta_1 & \lambda + \eta_1 & 0 & 0 & 0 \\ & \lambda + 2\mu_T & \lambda & 0 & 0 & 0 \\ & & \lambda + 2\mu_T & 0 & 0 & 0 \\ & & & 2\mu_T & 0 & 0 \\ & \text{sym.} & & & 2\mu_L & 0 \\ & & & & & 2\mu_L \end{bmatrix} \tag{4.47}$$

Orthotropic materials

A major drawback of integrating an orthotropic material model into a curvilinear coordinate system exists due to the existence of three different vector bases: global Cartesian ($\boldsymbol{e}_i$), local convected ($\boldsymbol{G}_i$) and material principal directions ($\boldsymbol{l}_i$). Therefore, material representations in Equations (4.42) and (4.44) pose as bottlenecks to proceed with derivations in the convected coordinate system. For an isotropic material, for its simplicity, an elegant analytical expression is formulated as given in Equation 4.32. This basis inconsistency issue is addressed by, first expressing the orthotropic material model in a more concise form, like Equation (4.46) for transversely isotropic materials. Secondly, basis conversions given in Equation (4.29) are performed algorithmically at every integration point of each shell element.

Now we continue from Equation (4.44) to derive a compact form, similar to Equation (4.46), for orthotropic materials. This is accomplished in the Cartesian basis, after some linear algebraic simplifications of Equation (4.42) as,

$$\begin{aligned}\mathcal{C}^{pqrs} &= a_{11}\; l_1^p l_1^q l_1^r l_1^s + a_{22}\; l_2^p l_2^q l_2^r l_2^s + a_{33}\; l_3^p l_3^q l_3^r l_3^s \\ &+ a_{12}\,(l_1^p l_1^q l_2^r l_2^s + l_2^p l_2^q l_1^r l_1^s) + a_{13}\,(l_1^p l_1^q l_3^r l_3^s + l_3^p l_3^q l_1^r l_1^s) + a_{23}\,(l_2^p l_2^q l_3^r l_3^s + l_3^p l_3^q l_2^r l_2^s) \\ &+ 2b_{12}\,(l_1^p l_1^r l_2^q l_2^s + l_2^p l_2^r l_1^q l_1^s) + 2b_{13}\,(l_1^p l_1^r l_3^q l_3^s + l_3^p l_3^r l_1^q l_1^s) + 2b_{23}\,(l_2^p l_2^r l_3^q l_3^s + l_3^p l_3^r l_2^q l_2^s)\;\;,\end{aligned} \tag{4.48}$$

where shear moduli entries are restructured to maintain the symmetry of terms $\mathcal{C}^{pqrs}$. Similar to transversely isotropic materials, l_1^p refers to the p^{th} component of the principal material direction $\boldsymbol{l}_1$. For instance, setting $\boldsymbol{l}_i = \boldsymbol{e}_i$ we arrive at,

$$\begin{aligned}\mathcal{C}^{1111} &= 1^4 a_{11} + 0^4 a_{22} + 0^4 a_{33} \\ &+ a_{12}(1^2 0^2 + 0^2 1^2) + a_{13}(1^2 0^2 + 0^2 1^2) + a_{23}(0^2 0^2 + 0^2 0^2) \\ &+ 2b_{12}(1^2 0^2 + 0^2 1^2) + 2b_{13}(1^2 0^2 + 0^2 1^2) + 2b_{23}(0^2 0^2 + 0^2 0^2) \\ &= a_{11}\;.\end{aligned}$$

Derivation of the compact Equations (4.46), (4.48) is important for the algorithmic implementation of transversely isotropic and orthotropic shells. Compact equations not only reduce the complex tensor algebra on structural tensors but also open the possibility to analytically express the basis conversions according to Equation (4.29). Moreover, compact equations intuitively reveal the dependency of principal material directions on the material constants a_{ij} and b_{ij}. Material tensor is further simplified for plane stress conditions of orthotropic materials. Detailed derivations of this formulations are given in Jones M. [136].

4.3.2 Model verification

We consider the analytical solution derived by Dobyns et al. [142], for simply supported plates, to verify and study the convergence of the results with the proposed solution. First, we discuss the analytical formulations derived by Dobyns et al. [142] in a general context. Next, we define the problem description and solve it using the analytical expressions by Dobyns et al. [142] and the proposed method.

Analytical solution of a simply supported orthotropic square plate

For consistency, in this short section, we use similar notations used in Dobyns et al. [142].

Equations of motion for orthotropic plates developed by Whitney and Pagano [143], including initial stress resultants N_x^0, N_y^0 and a foundation stiffness K, take the form,

$$D_{11}\psi_{x,xx} + D_{66}\psi_{x,yy} + (D_{12} + D_{66})\psi_{y,xy} - \kappa A_{55}\psi_x - \kappa A_{55}w_{,x} + m_x = I\ddot{\psi}_x \quad (4.49a)$$

$$D_{22}\psi_{y,yy} + D_{66}\psi_{y,xx} + (D_{12} + D_{66})\psi_{x,xy} - \kappa A_{44}\psi_y - \kappa A_{44}w_{,y} + m_y = I\ddot{\psi}_y \quad (4.49b)$$

$$\kappa A_{55}\psi_{x,x} + (\kappa A_{55} + N_x^0)w_{,xx} + \kappa A_{44}\psi_{y,y} + (\kappa A_{44} + N_y^0)w_{,yy} + p_z + Kw = P\ddot{w}\ , \quad (4.49c)$$

where w is the z direction displacement and ψ_x and ψ_y are the shear rotations in x and y directions respectively. Partial differentiation with respect to x or y is denoted by a comma while differentiation with respect to time is denoted by a dot. Moreover, p_z, m_x and m_y are the distributed loads on the plate. Also, κ is a shear correction factor commonly taken as $\pi^2/12$ (or $5/6$). A_{ij} and D_{ij} are material parameters given by,

$$(A_{ij}, D_{ij}) = \int_{-h/2}^{h/2} Q_{ij}(1, z^2)\,\mathrm{d}z \qquad i, j = 1, 2, 6 \quad (4.50a)$$

$$A_{ij} = \int_{-h/2}^{h/2} C_{ij}\,\mathrm{d}z \qquad i, j = 4, 5 \quad (4.50b)$$

$$(P, I) = \int_{-h/2}^{h/2} \rho(1, z^2)\,\mathrm{d}z\ , \quad (4.50c)$$

where Q_{ij} are reduced in-plane stiffnesses and C_{ij} are transverse shear stiffnesses.

Simply supported boundary conditions for a rectangular plate of uniform thickness h with dimensions a and b take the form,

$$w = \psi_{x,x} = 0 \quad \text{at} \quad x = 0, x = a \quad (4.51a)$$

$$w = \psi_{y,y} = 0 \quad \text{at} \quad y = 0, y = b\ . \quad (4.51b)$$

For a static load, solutions to equation (4.49) subjected to boundary conditions in equa-

tion (4.51) are given by,

$$\Psi_x = A_{mn}\cos(m\pi x/a)\sin(n\pi n/b) \tag{4.52a}$$
$$\Psi_y = B_{mn}\sin(m\pi x/a)\cos(n\pi n/b) \tag{4.52b}$$
$$W = C_{mn}\sin(m\pi x/a)\sin(n\pi n/b)\ , \tag{4.52c}$$

with the loading function given by,

$$p_z = q_{mn}\sin(m\pi x/a)\cos(n\pi n/b) \tag{4.53}$$

where q_{mn} are the terms of a Fourier series representation of the load. Substituting equations (4.52), (4.53) into the equations of motion (4.49) produces,

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12} & L_{22} & L_{23} \\ L_{13} & L_{23} & L_{33} \end{bmatrix} \begin{Bmatrix} A_{mn} \\ B_{mn} \\ C_{mn} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ q_{mn} \end{Bmatrix}\ , \tag{4.54}$$

where

$$L_{11} = D_{11}(m\pi/a)^2 + D_{66}(n\pi/b)^2 + \kappa A_{55} \qquad L_{12} = (D_{12} + D_{66})(m\pi/a)(n\pi/b)$$
$$L_{22} = D_{66}(m\pi/a)^2 + D_{22}(n\pi/b)^2 + \kappa A_{44} \qquad L_{13} = \kappa A_{55}(m\pi/a)$$
$$L_{33} = (\kappa A_{55} + N_x^0)(m\pi/a)^2 + (\kappa A_{44} + N_y^0)(n\pi/b)^2 + K \qquad L_{23} = \kappa A_{44}(n\pi/b)\ .$$

Solving equation (4.54) for A_{mn}, B_{mn} and C_{mn} we get,

$$A_{mn} = \frac{(L_{12}L_{23} - L_{22}L_{13})q_{mn}}{\det}\ , B_{mn} = \frac{(L_{12}L_{13} - L_{11}L_{23})q_{mn}}{\det}\ , C_{mn} = \frac{(L_{11}L_{22} - L_{12}^2)q_{mn}}{\det}\ ,$$

where 'det' is the determinant of the matrix in equation (4.54).

Problem description: A simply supported square plate of uniform thickness 10^{-5}mm, with dimensions $a = b = 10$mm, was applied a pressure load as defined in equation (4.53) for $m = n = 1$ and $q_{11} = 10^{-10}\text{N/mm}^2$. Following material parameters (units in N/mm^2) were used both in obtaining the analytical solution and finite element solution. Moreover, a 20×20 structured grid was used as the quadrilateral mesh for this problem

and subdivision shape functions of order three were used as the basis functions.

$$a_{11} = 10005.40 \quad a_{12} = 21.323 \quad a_{13} = 3.748 \quad a_{22} = 101.286 \quad a_{23} = 3.551$$
$$a_{33} = 10.125 \quad b_{12} = 745.0 \quad b_{23} = 154.0 \quad b_{13} = 234.0 \tag{4.55}$$

Moreover, we set $\boldsymbol{l}_i = \boldsymbol{e}_i$.

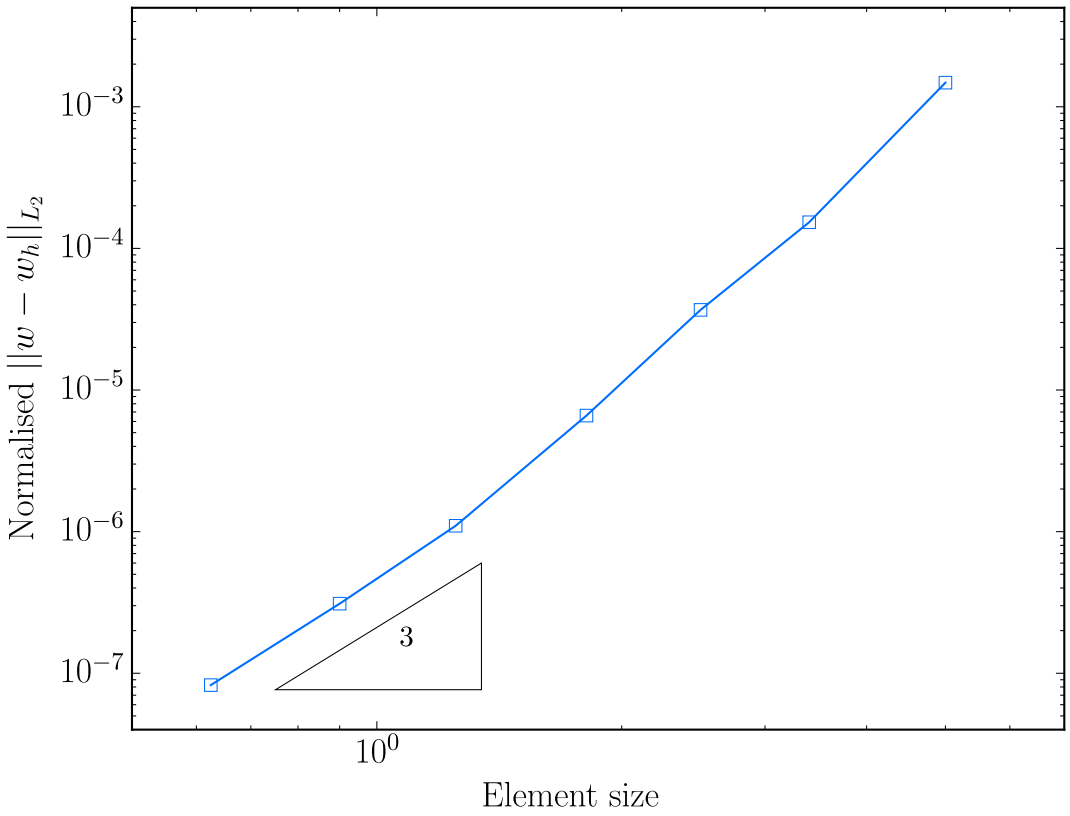


Figure 4.2: Convergence of the normalised L_2 norm error of the simply supported orthotropic square plate under a uniform pressure load

Figure 4.2 shows the convergence rate of the normalised L_2 norm error of the simply supported orthotropic square plate. Normalised L_2 error is defined as $||w - w_h||_{L_2} = \sqrt{\int_\Omega (w - w_h)^2 \mathrm{d}\Omega / \int_\Omega w^2 \mathrm{d}\Omega}$. Optimal convergence has been attained by the h-refinement of the structured mesh used in this analysis.

This thesis does not intend to discuss further on linear material models (for linear homogenisation) as the primary aim was set to analyse textile structures using nonlinear computational homogenisation. Instead, this chapter is concluded by presenting the

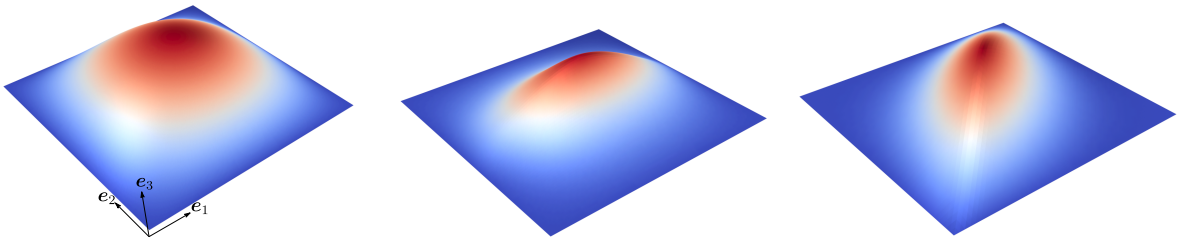
following examples to showcase different aspects of the linear anisotropic thin-shell formulations.

4.3.3 Additional examples

In this section, additional examples are provided to showcase different aspects of the formulations derived in Section 4.3. All the following examples use the material constants given in Equation (4.55). A comparatively higher value for a_{11} in Equation (4.55) is chosen intentionally to highlight the alignment of displacement contours along $\boldsymbol{l}_1$ direction.

Simply supported square plate

The same square plate used in the previous example is considered here. A 20×20 structured grid was used as the quadrilateral mesh for these problems and subdivision shape functions of order three were used as the basis functions. Moreover, principal material directions were chosen as $\boldsymbol{l}_i = \boldsymbol{e}_i$ for the first two problems and $\boldsymbol{l}_1 = [1/\sqrt{2}, 1/\sqrt{2}, 0]$, $\boldsymbol{l}_2 = [-1/\sqrt{2}, 1/\sqrt{2}, 0]$, $\boldsymbol{l}_3 = [0, 0, 1]$ for the third problem in Figure 4.3. Figure 4.3 illustrates the importance of the principal material directions of orthotropic materials on the deformed configuration. For instance, observe the deformations in Figures 4.3b and 4.3c. Under the same loading (mid-point concentrated load), square plate deflects completely distinct to each other. Figure 4.3b deforms predominantly in $\boldsymbol{l}_1 (= \boldsymbol{e}_1)$ direction simply due to the higher stiffness entry a_{11} given in Equation (4.55). Similarly, deformation in Figure 4.3c results as a higher resultant stiffness in $\boldsymbol{l}_1$ direction. However, for uniform pressure loading, deformation dependency on principal material directions is insignificant (see Figure 4.3a).

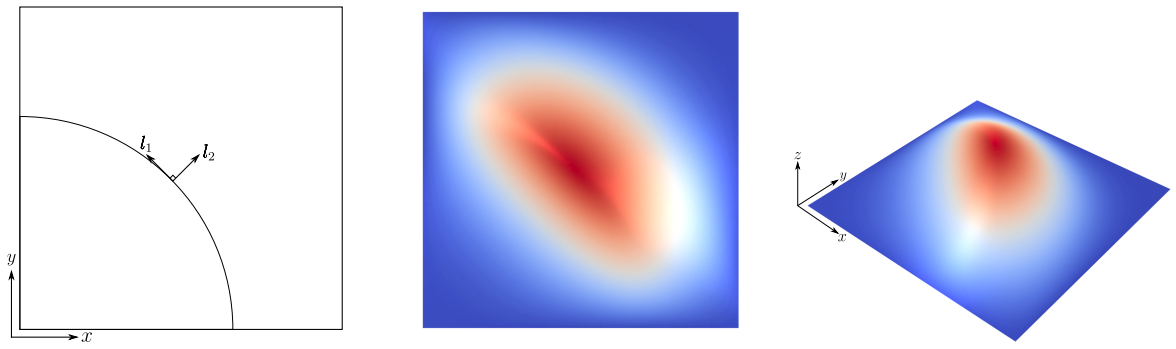


(a) (b) (c)

Figure 4.3: Displacement contours of three identical orthotropic plates under different loading and principal material directions. (a) Uniform pressure loading with $\boldsymbol{l}_i = \boldsymbol{e}_i$ (b) Mid-point concentrated force with $\boldsymbol{l}_i = \boldsymbol{e}_i$ (c) Mid-point concentrated force with $\boldsymbol{l}_1 = [1/\sqrt{2}, 1/\sqrt{2}, 0]$, $\boldsymbol{l}_2 = [-1/\sqrt{2}, 1/\sqrt{2}, 0]$, $\boldsymbol{l}_3 = [0, 0, 1]$

It is straightforward to implement orthotropic materials with fixed principal material directions in most of the existing commercial FEA software. However, the use of commercial software, in the presence of complex orthotropic material directions, might limit the users' scope of analyses or require significant manual intervention. Our implementations can easily incorporate such complex material directions, as shown in Figure 4.4.

Figure 4.4a shows the spatially varying in-plane orthotropic principal directions $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$, which follow tangential and radial directions of a circle, respectively. Hence, the third material direction is taken as $\boldsymbol{l}_3 = [0, 0, -1]$. We use a 20×20 structured grid as the quadrilateral mesh for this problem and subdivision shape functions of order three as the basis functions. For clarity, only one circular material curve is shown in Figure 4.4a. Figures 4.4b and 4.4c clearly illustrate the displacement contours and their alignment with the material principal directions.

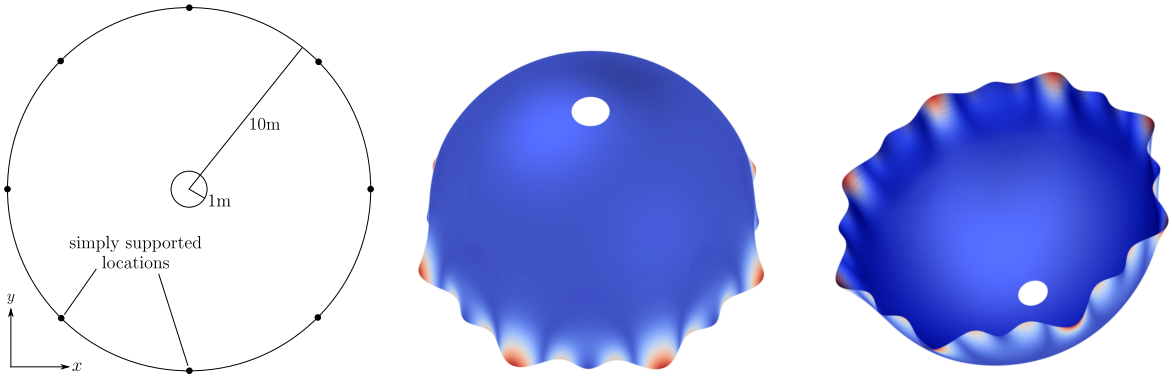


(a) (b) (c)

Figure 4.4: Deformation of a complex orthotropic plate subject to a mid–point out of plane loading. (a) Problem description. Displacement magnitude contours in (b) top view and (c) side view.

Parachute canopy-like structure

Next, we simulate a parachute canopy-like structure to visualise the deformation under uniform internal pressure of 10^{-10}MPa. As shown in Figure 4.5a, hemispherical canopy is fixed at 8 different locations of the open edge of the base and the pressure is applied on the interior side of the canopy. The base diameter was chosen as 10m and canopy thickness was set to 10^{-5}m. Moreover, principal material directions of the canopy, $\boldsymbol{l}_1$, $\boldsymbol{l}_2$ and $\boldsymbol{l}_3$, were assigned as radial, polar and azimuthal directions of the hemisphere, respectively.



(a) (b) (c)

Figure 4.5: Deformation of an orthotropic hemispherical canopy-like structure subject to a pressure loading. (a) Problem description where principal material directions are chosen to be in radial, polar and azimuthal directions of the hemisphere. Displacement magnitude contours in (b) exterior and (c) interior of the canopy.

High-fidelity models for microscale and macroscale simulations have been introduced respectively in Chapters 3 and 4. Now we use Chapter 5 to kinematically assemble them to arrive at a data-driven computational homogenisation framework for textile materials. A neat extension to textile material design is also proposed in the forthcoming chapter. One highlight of the subsequent chapter is that it proposes a statistical learning technique that outshines the conventional computational homogenisation schemes in many aspects.

5 Computational homogenisation

This chapter begins with a discussion on conventional multiscale techniques and their evolution. It also includes a discussion on the strengths and weaknesses of conventional multiscale methods. In Section 5.1, fundamental principles of multiscale methods have been briefly discussed for a twoscale problem. This includes formulations of a conventional computational homogenisation scheme to attain the equilibrium solution by averaging field variables over a boundary value problem domain. Next, the latest developments in multiscale methods will be discussed to merit their strengths to mitigate the weaknesses identified in conventional computational homogenisation schemes.

Section 5.2 is reserved for introducing readers to machine learning. Starting with general facets of machine learning, this section is concluded with detailed discussions and formulations concerning Gaussian Process Regression (GPR) Bayesian machine learning technique. In this thesis, we use the term 'machine learning' to refer to Bayesian machine learning. Also, we drop the slight differences between machine learning and statistical learning and use the two interchangeably. The next section elaborates on the proposed data-driven homogenisation technique, highlighting the merits of data-driven methods to mitigate the weaknesses mentioned in Section 5.1. Section 5.3 also discusses the implementation details of the proposed scale transition framework, error and sensitivity analyses. Chapter 5 is concluded with a proposed framework for data-driven material designing under the domain of computational homogenisation. This is identified as a neat extension to the proposed data-driven homogenisation strategy.

For fibre-based periodic structures, like woven and knitted textiles, Chapter 5 blends the models introduced in Chapters 3 and 4 elegantly using GPR machine learning. Intuitively, applications of this nature are not limited to technical textiles, but can be used in the homogenisation of 3D printed lattices, textile composites and digitally interlooped structures.

5.1 Conventional multiscale modelling

For the clarity of the upcoming discussions, we establish the following abbreviations, notations and definitions given in Table 5.1.

Table 5.1: Abbreviations, notations and definitions

$(\bullet)_{\mathrm{M}}$	quantity in Macroscale
$(\bullet)_{\mathrm{m}}$	quantity in microscale
V_0	volume in the reference configuration
$\nabla_{0\mathrm{m}}$	gradient operator with respect to reference microscale configuration
AI	Artificial intelligence
RQ	Rational Quadratic covariance function
SE	Squared Exponential covariance function

5.1.1 Preliminaries

Modern multiscale methods have deep-rooted in powerful state-of-the-art nonlinear computational techniques addressing scientific and engineering problems of different scales. Multiscale methods gained increasing popularity in the past few decades due to well-understood scale transitions and properly developed fundamental principles on the intrinsic roles of each scale. Multiscale methods, in general, aim to predict the macroscopic behaviour of engineering materials through a systematic and consistent modelling of the fundamental mechanics and physics of underlying microstructures. These microstructures can be of different scales. A proper understanding of the microscale behaviour, evolution, and mechanical response of materials is critical in modelling such microstructures. Applications of multiscale methods are diverse, but certainly not limited to: heterogeneous solids [39, 144], complex fluids [145], microfluidics [146], heat transfer [147] , anisotropic electrical conductivity [148] and neuroscience [149].

Accurate coupling of different scales, respecting the mechanics and compatibility conditions, lies at the core of multiscale strategies. A number of methods have been proposed to achieve this. *Homogenisation*, a common reference in mechanics communities [38, 39, 150–152] also termed *coarse-graining* in the physics community [153], plays a major role in plethora of multiscale methods. In this thesis, we interchangeably use the terms homogenisation and multiscale modelling by dropping the distinctions between

them. Ivo Babuska, in 1976, originally coined the term homogenisation which essentially is based on averaging theorems between the different scales. Original works of homogenisation date back to more than a century when the interest for micromechanics of heterogeneous material became prevalent. Voigt (1887), Sachs (1928), Reuss (1929) and Taylor (1938) led the early steps in homogenisation schemes, and significant contributions were later added by Eshelby (1957), Kroner (1958), Hashin and Shtrikman (1963), Hill (1965), Mori and Tanaka (1973) and many others [154].

Different classifications have been suggested to categorise multiscale methods. One classification divides the manifold multiscale methods as concurrent, hierarchical and hybrid methods [39, 155]. In this thesis, and also the method of choice in the mechanics community, we focus on the hierarchical multiscale methods. In the hierarchical methods, the scales are coupled in a hierarchical manner, which implies that different scales are considered and linked in the same part of a domain. The hierarchical link is usually established through volume averaging of field variables [39], and different scales are solved in parallel to arrive at the global equilibriums in all scales. Moreover, this thesis focusses only on twoscale problems where two distinct scales are named as macroscale and microscale (or finescale). On a big picture, the hierarchical link for general twoscale problems follows the traditional loop strategy as depicted in Figure 5.1.

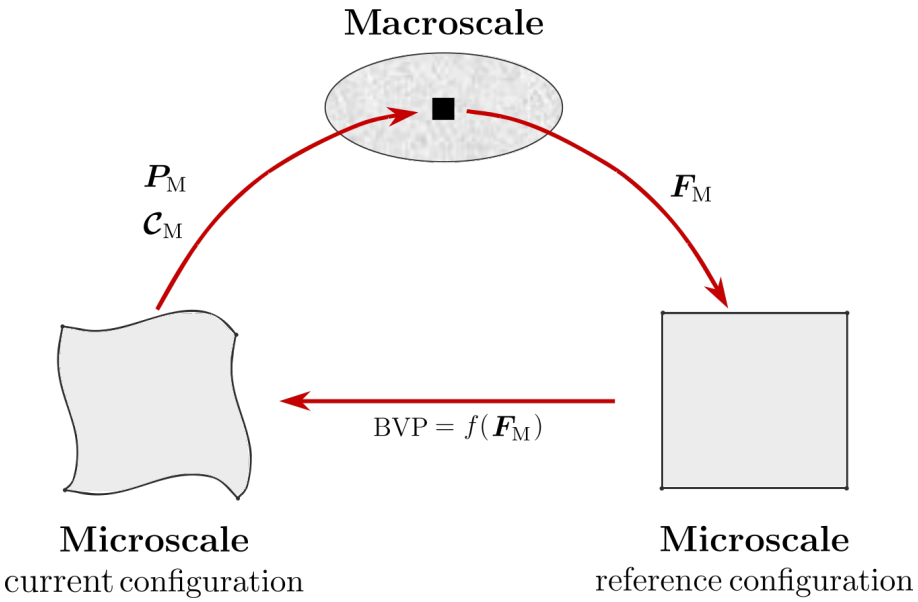


Figure 5.1: Conventional computational homogenisation scheme for a twoscale problem. $\boldsymbol{F}_{\mathrm{M}}$, $\boldsymbol{P}_{\mathrm{M}}$ and $\boldsymbol{\mathcal{C}}_{\mathrm{M}}$ are macroscale deformation gradient, First Piola-Kirchhoff stress and material constitutive tensor, respectively.

Next, a general twoscale problem is formulated to familiarise the reader with the first principles of computational homogenisation. Table 5.1 summarises some of the commonly used notations and definitions in this chapter. Moreover, for brevity, detailed derivations of certain formulations have been intentionally omitted in this thesis. Interested readers are kindly invited to refer to Geers et al. [38, 39, 150, 156] for further details.

5.1.2 Twoscale problem

Macroscale problem

At the macroscopic scale, in the absence of body forces, momentum balance can be expressed as the divergence of the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ as [38],

$$\nabla_{\mathrm{M}} \cdot \boldsymbol{P}_{\mathrm{M}}^{\mathrm{T}} = \boldsymbol{0}. \tag{5.1}$$

A constitutive relation between the stress and kinematic quantities needs to be postulated for the completion of the macroscale problem given in Equation (5.1). In Section 4.2, we formulated equations required to completely solve the kinematics of thin-shells by assuming a linear isotropic material model. However, for materials such as composites, textiles, glass, biological tissues and polymers, not only the material models become nonlinear and anisotropic but also tend to be locally heterogeneous. A conventional computational homogenisation scheme, as shown in Figure 5.1, plays a vital role in tackling these issues and proven to be exceptionally effective [39]. In a broad view, computational homogenisation technique systematically extracts the constitutive response numerically from rigorous computations of a microstructural Representative Volume Element (RVE) to solve the macroscale equilibrium equation.

Deformation gradient $\boldsymbol{F}_{\mathrm{M}}$ is the macroscale quantity that departs from the macroscale as an input to the microscale boundary value problem. Therefore, Equation (5.1) is written in terms of the first Piola-Kirchhoff stress tensor, which is the energy conjugate of the deformation gradient. To this end, the relationships between the commonly used

stress tensors take the forms [116, 120],

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-\mathrm{T}} \tag{5.2a}$$

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-\mathrm{T}} \tag{5.2b}$$

$$\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S} \ , \tag{5.2c}$$

where $\boldsymbol{P}$, $\boldsymbol{S}$ and $\boldsymbol{\sigma}$ are first Piola-Kirchhoff, second Piola-Kirchhoff and Cauchy stress tensors, respectively. Moreover, J is the Jacobian determinant and $\boldsymbol{F}$ is the deformation gradient.

Microscale problem

An appropriate RVE, capable of capturing relevant microscale kinematic fluctuations, should be selected to kinematically couple with the macroscale. This, however, should respect the separation of scales principle, such that the selected RVE should be finer than the characteristic length of the relevant macroscopic field variation, but sufficiently larger than microfluctuations [39]. With a characteristic RVE in hand, next steps involve deriving expressions to relate the microscale deformation gradient to the macroscale deformation gradient. Herein, we use the Taylor expansion to express variations in an arbitrary macroscale position vector. This position vector is denoted as $\boldsymbol{X}$ and $\boldsymbol{x}$ in the reference (or undeformed) and deformed configurations, respectively.

The Taylor expansion of the nonlinear deformation map takes the series form as given in Equation (5.3). This expansion has been supplemented by an additive component $\boldsymbol{w}$, that can be interpreted as the microstructural contribution on the macroscale. Considering the first two terms on the right-hand side of Equation (5.3) lead to first-order computational homogenisation whereas taking more than two terms yield the expansion of a higher-order computational homogenisation strategy.

$$\Delta\boldsymbol{x} = \boldsymbol{w} + \boldsymbol{F}_{\mathrm{M}} \cdot \Delta\boldsymbol{X} + \frac{1}{2}\Delta\boldsymbol{X} \cdot (\nabla_{\mathrm{M}} \cdot \boldsymbol{F}_{\mathrm{M}}) \cdot \Delta\boldsymbol{X} + \cdot\cdot\cdot\cdot\cdot \tag{5.3}$$

In this thesis, we resort to the first order computational homogenisation and therefore Equation (5.3) takes the truncated form,

$$\Delta\boldsymbol{x} = \boldsymbol{w} + \boldsymbol{F}_{\mathrm{M}} \cdot \Delta\boldsymbol{X} \ . \tag{5.4}$$

Prior to establishing the relationship between macroscale and microscale deformation gradients, it is important to discuss the boundary enforcement of RVEs. These boundary conditions are specifically designed to treat microfluctuations $\boldsymbol{w}$ in Equation (5.4).

Microscale RVE boundary conditions

A solution to the microscale boundary value problem is attained after the compatible enforcement of boundary conditions of the RVE. Figure 5.2 is a 2D schematic of a generalised RVE which is used to discuss the enforcement of boundary conditions. Moreover, Γ refers to the RVE boundary and the subscripts T, B, L, R refer to Top, Bottom, Left and Right, respectively. Boundary nodes are labelled in roman numerals.

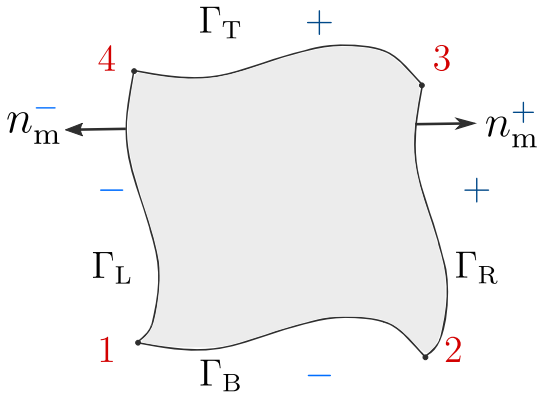


Figure 5.2: Schematic of a generalised RVE

We identify three main boundary enforcement techniques commonly used in the computational homogenisation communities.

- **Displacement boundary conditions**: This sets the microfluctuations at the RVE boundary to zero. Thus, equation (5.4) simplifies to,

$$\Delta \boldsymbol{x} = \boldsymbol{F}_{\mathrm{M}} \cdot \Delta \boldsymbol{X} \ . \tag{5.5}$$

  This relation also means that the linear mapping of the macroscopic deformation gradient is applied over the RVE and solely governs the positions of the RVE boundary. In the literature, this is known as the Taylor (or Voigt) assumption [38].

- **Traction boundary conditions**: $\boldsymbol{P}_{\mathrm{M}}$ is prescribed on the RVE based on Sachs (or Reuss) assumption. But this is not widely used in computational homogenisation as it leads to unsatisfactory results [39].

- **Periodic boundary conditions**: Considering the local periodicity of a RVE, periodic boundary conditions are written as,

$$\boldsymbol{x}_{\mathrm{m}}^{+} - \boldsymbol{x}_{\mathrm{m}}^{-} = \boldsymbol{F}_{\mathrm{M}} \cdot (\boldsymbol{X}_{\mathrm{m}}^{+} - \boldsymbol{X}_{\mathrm{m}}^{-}) \ . \tag{5.6}$$

  Equation (5.6) requires the microfluctuations $\boldsymbol{w}_{\mathrm{m}}$ and the normal vectors at the boundary to be anti-periodic. Hence we write,$\boldsymbol{w}_{\mathrm{m}}^{+} = \boldsymbol{w}_{\mathrm{m}}^{-}$ and $\boldsymbol{n}_{\mathrm{m}}^{+} = \boldsymbol{n}_{\mathrm{m}}^{-}$

5.1.3 Scale transitions

Interactions between the two scales occur at two different levels of the computational homogenisation process. First, macro-micro transition, where a macroscale quantity (usually $\boldsymbol{F}_{\mathrm{M}}$) departs from the macroscale to the microscale devises the boundary value problem at the microscale. Secondly, in micro-macro transition, the stress resultants (usually $\boldsymbol{P}_{\mathrm{M}}$) and their tangents are transferred from microscale to macroscale to complete the cyclic solution process as illustrated in Figure 5.1. In literature, this is also known as the handshake or coupling process [18, 38].

Macro-Micro scale transition

The most commonly used scale transition to establish the macro-to-micro coupling is the kinematical averaging relation. It requires the volume averaged microscale deformation gradient tensor $\overline{\boldsymbol{F}}_{\mathrm{m}}$, to be equal to the macroscale deformation gradient tensor $\boldsymbol{F}_{\mathrm{M}}$, which takes the form,

$$\boldsymbol{F}_{\mathrm{M}} = \overline{\boldsymbol{F}}_{\mathrm{m}} = \frac{1}{V_{0\mathrm{m}}} \int_{V_{0\mathrm{m}}} \boldsymbol{F}_{\mathrm{m}} \mathrm{dV} \ . \tag{5.7}$$

Microscale deformation gradient in Equation (5.7) is further expanded using Equation (5.4) and simplified using the divergence theorem to arrive at the following relation.

$$\frac{1}{V_{0\mathrm{m}}} \int_{V_{0\mathrm{m}}} \boldsymbol{F}_{\mathrm{m}} \mathrm{dV}_{0\mathrm{m}} = \boldsymbol{F}_{\mathrm{M}} + \int_{\Gamma_{0\mathrm{m}}} \boldsymbol{w} \otimes \boldsymbol{n}_{\mathrm{m}} \mathrm{d}\Gamma \ . \tag{5.8}$$

Thus, comparing the equations (5.7) and (5.8), it is evident that, the second term on the right-hand side of equation (5.8) should vanish consistently based on the contributions from microfluctuations $\boldsymbol{w}$. It has been proven that the use of any boundary conditions enlisted in section 5.1.2 will satisfy this requirement [38, 39].

Micro-Macro scale transition

The micro-macro scale transition relation is usually established based on the Hill–Mandel condition or macrohomogeneity condition [157]. This condition requires the volume average of the virtual of work performed on the microscale RVE to be equal to the variation of local work on the macroscale. Formulated in terms of work conjugates $\boldsymbol{F}$ and $\boldsymbol{P}$, the Hill–Mandel condition reads,

$$\frac{1}{V_{0\mathrm{m}}}\int_{V_{0\mathrm{m}}} \boldsymbol{P}_{\mathrm{m}} : \delta\boldsymbol{F}_{\mathrm{m}}^{\mathrm{T}}\mathrm{dV} = \boldsymbol{P}_{\mathrm{M}} : \delta\boldsymbol{F}_{\mathrm{M}}^{\mathrm{T}} \ . \tag{5.9}$$

Substituting Equation (5.8) in (5.9) for microscale deformation gradient tensor, and using the divergence theorem, macroscale first Piola-Kirchhoff stress tensor is written as,

$$\boldsymbol{P}_{\mathrm{M}} = \frac{1}{V_{0\mathrm{m}}}\int_{\Gamma_{0\mathrm{m}}} \boldsymbol{p}_{\mathrm{m}} \otimes \boldsymbol{X}_{\mathrm{m}}\mathrm{d}\Gamma \tag{5.10}$$

where $\boldsymbol{p}_{\mathrm{m}} = \boldsymbol{P}_{\mathrm{m}} \cdot \boldsymbol{n}_{\mathrm{m}}$. Pleaser refer to Geers et al. [39] for the rigorous derivations. Furthermore, using the following identity and the divergence theorem, the boundary surface integral of equation (5.10) can be transformed into a volume integral as given in Equation (5.11).

$$\nabla_{0\mathrm{m}} \cdot (\boldsymbol{P}_{\mathrm{m}}^{\mathrm{T}} \otimes \boldsymbol{X}_{\mathrm{m}}) = (\nabla_{0\mathrm{m}} \cdot \boldsymbol{P}_{\mathrm{m}}^{\mathrm{T}}) \otimes \boldsymbol{X}_{\mathrm{m}} + \boldsymbol{P}_{\mathrm{m}} \cdot (\nabla_{0\mathrm{m}}\boldsymbol{X}_{\mathrm{m}}) = \boldsymbol{P}_{\mathrm{m}}$$

where $\nabla_{0\mathrm{m}}\boldsymbol{X}_{\mathrm{m}} = \mathbf{I}$ and $\nabla_{0\mathrm{m}} \cdot \boldsymbol{P}_{\mathrm{m}}^{\mathrm{T}} = \mathbf{0}$ in the absence of body forces of the RVE.

$$\boldsymbol{P}_{\mathrm{M}} = \frac{1}{V_{0\mathrm{m}}}\int_{V_{0\mathrm{m}}} \boldsymbol{P}_{\mathrm{m}}\mathrm{dV} \tag{5.11}$$

Thus, based on the Hill–Mandel energy conservation relation, the macroscale first Piola–Kirchhoff stress tensor can simply be identified as the volume average of the microscale first Piola–Kirchhoff stress tensor, as given in Equation (5.11).

Extraction of the material constitutive model

To solve the nonlinear macroscale problem in Equation (5.1), it requires to compute the tangents of internal forces with respect to strain components at every integration point. This requires the extraction of, not only the macroscopic stress tensor $\boldsymbol{P}_{\mathrm{M}}$, but also the material tangent matrix $\boldsymbol{\mathcal{C}}_{\mathrm{M}}$ (see micro-macro transition in Figure 5.1). In linear computational homogenisation, it is not necessary to compute the material tangents at every integration point, as the *linearity* priori assumes the material tangent to be constant at any material point in the macroscale. In this case, we can use the linear anisotropic material models introduced in Section 4.3. In computational homogenisation literature, such constant material tangent matrices are abbreviated as ABD stiffness matrices [57–59,158]. In a 6×6 ABD matrix, 3×3 sub-stiffness-matrices A, D and B stand for membrane, bending and coupling between membrane and bending, respectively.

Extraction of the material tangent matrices has been approached using a forward difference method by Miehe [159] and Nadler et al. [18]. Moreover, schemes that employ a direct condensation of the constrained degrees of freedoms are considered to extract the (integration) point-wise material tangent matrices [38,39,156,160,161]. Detailed discussions on these methods are not presented here as we employ a data-driven technique in Section 5.3 to extract the stresses and material tangent matrices.

5.1.4 Recent developments

In the past few years, significant progress had been made in bridging the mechanics of materials into other disciplines, e.g. downscaling to materials science or upscaling to structural engineering. Also, upon the rapid technological developments in 3D printing and CNC, engineers look for more advanced and optimised structures to achieve time and cost savings. A few of these structures would include lattices [34], bio-inspired structures [162] and 3D printing of polymers onto textiles [163]. Alongside these technological advancements, we notice some improvements in multiscale modelling techniques to handle the complexity of such structures.

Damanpack et al. [34] and Ptochos et al. [164] compare and validate homogenised unit cell response of a lattice structure to that of a large scale experimental results from 3D printed lattices. Their motive was not to seamlessly integrate the macroscale to

microscale, but to experimentally validate the nonlinear multiscale response to experimental results. Ahmadi et al. [165] experimented on open-cell porous biomaterials made of diamond lattice unit cells. They used Euler-Bernoulli and Timoshenko beam theories to validate them in a finite element environment.

On a broader topic, research works of Weeger et al. [1], Dinh et al. [20], Vassiliadis et al. [114], Kueh et al. [58], Nadler et al. [18] and Nilakantan et al. [166] fall under computational homogenisation of technical textiles. Some of these works predict the mechanical response of a textile RVE and validate against corresponding experimental results [20, 58, 114]. They do not aim to seamlessly couple the macroscale with microscale in a nested form. However, the coupling between macroscale and microscale can be handy in most cases when one needs to consider local enrichments, multiphase microstructures and testing of RVE parameters [39]. For this coupling, Miehe [159] and Nadler et al. [18] used a forward difference technique to combine the two scales whereas Weeger et al. [1] used a B-spline surface parameterisation technique to homogenise the constitutive behaviour of knitted textiles.

Based on the underlined theory, computational homogenisation can be classified as linear and nonlinear. In linear homogenisation, the coupling between two scales is achieved simply by formulating the ABD stiffness matrix from RVE simulations [57–59, 158]. Such ABD matrices are subsequently used in the macroscale simulations to couple the macroscale strains to stress resultants. In nonlinear homogenisation, however, scales are coupled at every integration point in the macroscale domain. Due to the complexity and higher computational costs involved in nonlinear homogenisation, researchers aim to simplify the material models by introducing material model parameterisations [1, 21].

The latest addition to the family of multiscale modelling techniques comes from the works of Bessa et al. [61, 167] and Liu et al. [168] where data-driven techniques have been explored within the domain of nonlinear computational homogenisation. In this thesis, we closely follow such data-driven techniques to implement a multiscale modelling framework for plain-woven and weft-knitted textiles.

5.2 Gaussian Process Regression

From a historical point of view, one can argue for significant distinctions in statistical modelling, machine learning and artificial intelligence (AI). However, with the evolution

of data science, we can see the overlaps and relations of each of these key terms. In a big picture, we see machine learning is one of the biggest subsets of AI. Statistical modelling departs the problem with a known model, to be trained using the available data to solve the problem. However, in machine learning, model selection is crucial as the problem at hand does not reveal any information about the model. Therefore, machine learning uses the available data to train a model first and then to solve the problem. This relation is illustrated in Figure 5.3.

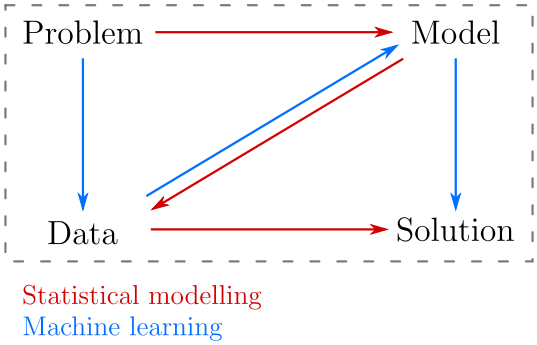


Figure 5.3: Statistical modelling versus Machine learning

In this chapter, we do not intend to discuss about AI as it is not directly relevant to the context of this thesis. However, a broad overview is presented in Section 5.2.1 about machine learning, highlighting the latest developments and motivations to include machine learning under the domain of multiscale modelling. This broad discussion is reserved for the readers coming from a non-data-science background.

5.2.1 Background

Machine learning, effortlessly, became a hot topic in the past decade, mainly due to the rapid increase of computational capabilities to implement and optimise machine learning algorithms. In addition, a wide range of applications in diverse industries such as banking, retail, medical, defence, social media, and energy, not only made machine learning the tool of choice to make predictions, but also attracted manifold research communities to upgrade the current practices to the next level. The term *Machine learning* was first coined by Arthur Samuel as the "Field of study that gives computers the ability to learn without being explicitly programmed" where he studied machine learning with the aid of playing checkers [169]. Later in 1997, Tom Mitchell provided a

broad definition, widely quoted in the related communities as "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E" [170]. For detailed discussions on different aspects of machine learning and machine learning algorithms, readers are encouraged to refer to Bishop [37], Goodfellow et al. [171] and Rogers et al. [41].

To this end, some technical terms are listed as follows and used consistently throughout this section. Most of these terms follow from its original meanings.

- dataset/database: a collection of many examples points. These example points are also named data points.
- data point: one example point of a dataset.
- inputs: inputs are the independent variables of a dataset.
- output(s): output(s) is/are the dependent variable(s) in the dataset.
- features: features of a dataset refer to the different labels in the input space.
- targets: targets of a dataset refer to the different labels in the output space.

For instance, we denote a dataset as $(\boldsymbol{x}_i^j, \boldsymbol{y}_i^k)$ with $j = 1, 2, 3$, $i = 1, 2, \cdots, 100$ and $k = 1, 2$. We expand the dataset for clarity as

$$(\boldsymbol{x}_i^j, \boldsymbol{y}_i^k) = \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & y_1^1 & y_1^2 \\ x_2^1 & x_2^2 & x_2^3 & y_2^1 & y_2^2 \\ x_3^1 & x_3^2 & x_3^3 & y_3^1 & y_3^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{99}^1 & x_{99}^2 & x_{99}^3 & y_{99}^1 & y_{99}^2 \\ x_{100}^1 & x_{100}^2 & x_{100}^3 & y_{100}^1 & y_{100}^2 \end{bmatrix} \tag{5.12}$$

Dataset in Equation (5.12) has 100 *data points* (i), 3 *features* (j), 2 *targets* (k) and ($100 \times 3 =$) 300 *inputs* and ($100 \times 2 =$) 200 *outputs*.

Using intuitive examples, now we discuss each of these three components; Tasks (T), Performance measures (P) and Experiences (E).

Task - T

Machine learning automates the solution process of some problems (tasks) that are too complex to be solved using fixed programs or human intervention. We formally introduce a *task* as the activity that a machine learning algorithm learns to achieve. Some of the commonly used machine learning tasks are listed below. For further applications of machine learning tasks, readers are invited to refer to Goodfellow et al. [171].

Classification: Classification tasks require the computer program to specify which of k categories some inputs belong into. Machine learning algorithm produces a function $f : \mathbb{R}^n \longrightarrow \{1, \cdots, k\}$. Examples of classification tasks include object recognition [172] and facial recognition tools in social media. Classification tasks are designed to produce discrete outputs.

Regression: In regression type tasks, the computer program is required to output a numerical value for a given input, i.e. $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$. Regression type tasks are common in future predictions of energy usage, customer demand, insurance claims, security pricing etc. Regression is the machine learning task used in this thesis in the context of multiscale modelling.

Translation: Machine translation involves translating a series of symbols in one language to a series of symbols of another language. This is widely used in the context of Natural Language Processing (NLP) [173].

Anomaly detection: As the name implies, the computer program is intended to filter unusual or atypical entries in a set of events or objects. One common application is credit card fraud detection where unusual transactions are flagged automatically to identify any abnormal purchases.

Performance measure - P

It is crucial to measure the performance of a machine learning algorithm quantitatively. Usually, performance measures are task-dependent. For instance, the accuracy of a classification task is measured by the error rate as the expected $0-1$ loss [171]. Mean Squared Error (MSE) can be used as a performance measure for regression problems. Notwithstanding the task-specific performance measures, any performance measure should be tested on a set of data, that has not been seen by the machine learning algorithm, during

the training process. At this juncture, we formally distinguish 'model unseen data' as **test data** and 'model seen data' as **training data**. In literature, three distinct datasets can be found namely training, validation and testing datasets. But in this thesis, the distinction between validation and test datasets has been dropped for the explanation purpose [61].

Experience - E

Machine learning algorithms are broadly categorised as supervised, unsupervised and reinforcement learning [37,171,174,175]. These three categories control what the machine learning algorithms are expected to experience during the learning process.

Supervised learning algorithms: experience a dataset containing features that already have been associated with a feature label or a target label. For example, take Iris dataset [176]. For each labelled length and width of sepals and petals, Fisher classified iris plants into three different species.

Unsupervised learning algorithms: experience a dataset containing features but not explicitly labelled like in supervised learning. The aim is to learn the useful properties of the structure of this dataset and to categorise or cluster similar data points.

Reinforced learning algorithms: experience an interaction with the environment and learn by rewarding or punishing the output produced. This type of algorithms is used in optimising the winning strategies of games like chess and checkers.

There are many machine learning algorithms, based on different underlying principles, to learn from data and make predictions on unseen or new data points. These include, but not certainly limited to linear regression, linear classification models, neural networks, kernel methods, sparse kernel machines, graphical models and mixture models. Readers are recommended to read through Bishop [37] for clear, eloquent and descriptive explanations on each of these methods. But in this thesis, attention is only given to one of the commonly used Bayesian machine learning technique, **Gaussian Process Regression**.

Gaussian process machine learning is also termed Kriging [61, 167, 177] in the literature and has promising merits to be chosen over other machine learning techniques. These

advantages include the ability to perform well with relatively smaller datasets, non-parametric, well-established theoretical basis and less uncertain decisions required than a neural network algorithm [174]. However, it suffers from large dimensionality issues and inability to operate on relatively bigger datasets simply due to the expensive matrix inverse computations. Aforementioned merits motivate the choice of using Gaussian processes in this context due to the use of relatively smaller datasets and specificity of the problem.

5.2.2 Gaussian Process Regression - A deep dive

First, for clarity, we present Table 5.2 that contains frequently used notations, abbreviations and definitions.

Table 5.2: Notations, abbreviations and definitions used in Gaussian Process Regression

$\mathcal{GP}$	$\mathcal{G}$aussian $\mathcal{P}$rocess
n	number of training data points
d	number of input features
$\boldsymbol{x}_p$	the p^{th} training input data point
$\boldsymbol{x}_*$	a testing data point
$(\bullet)_*$	denotes a quantity in testing dataset
X	the $d \times n$ design matrix of training inputs; $X(q,p) = x_p^q$
X_*	matrix of testing inputs
y_p	the p^{th} training output (or target)
y_*	a target in the testing outputs
$\mathbf{0}$	a vector of zeros
$k(\boldsymbol{x}_p, \boldsymbol{x}_q)$	covariance (or kernel) function evaluated at $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$
K or $K(X,X)$	$n \times n$ covariance (Gram) matrix where $K(p,q) = k(\boldsymbol{x}_p, \boldsymbol{x}_q)$
K_* or $K(X,X_*)$	$n \times n_*$ covariance matrix between training and testing inputs
$\boldsymbol{k}_*$ or $\boldsymbol{k}(\boldsymbol{x}_*)$	$n \times 1$ vector $(K(X,\boldsymbol{x}_*))$ between training inputs and *one* testing input
$\boldsymbol{\Theta}$	vectorised hyperparameters
ℓ	lengthscale hyperparameter
σ_f	scaling hyperparameter
σ_n	variance of Gaussian noise

This section is reserved for introducing the readers to Gaussian Process Regression, abbreviated as GPR consistently in the subsequent text. We closely follow the guidebook of Rasmussen and Williams [174] in this section to establish the theoretical basis for

explaining the role of GPR in multiscale modelling. We formally start with the definition of a Gaussian process as, 'A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution' [174]. Moreover, a Gaussian process is completely specified by its mean function and covariance function.

We define the mean function $m(\boldsymbol{x})$ and the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ of a real Gaussian process $f(\boldsymbol{x})$ as,

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})], \tag{5.13}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))] \ , \tag{5.14}$$

where $\mathbb{E}[(\bullet)]$ refers to the expected value of a quantity $(\bullet)$. Moreover, $m(\boldsymbol{x})$ is further decomposed as,

$$m(\boldsymbol{x}) = \boldsymbol{n}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\beta} \tag{5.15}$$

with $\boldsymbol{n} = [n_1(\boldsymbol{x}), n_2(\boldsymbol{x}), \cdots, n_u(\boldsymbol{x})]^{\mathrm{T}}$ being u known functions. They can be constant, linear, quadratic, exponential etc., functions. $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_u]^{\mathrm{T}}$ is a vector of unknown co-efficients to be determined.

We also write the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ using a squared exponential (Gaussian covariance) function as,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \mathrm{exp}\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}}\mathbf{M}(\boldsymbol{x} - \boldsymbol{x}')\right) \ , \tag{5.16}$$

where σ_f^2 is the scaling hyperparameter and $\mathbf{M}$ is a diagonal matrix containing the lengthscale hyperparameters ($\ell_i \ \ i = 1, 2, \cdots, n$). We define $\boldsymbol{\Theta} = \{\ell_i, \sigma_f^2\}^{\mathrm{T}}$ as a vector containing lengthscale and scaling hyperparameters. Simplest form of $\mathbf{M}$ would be in the form $\mathbf{M} = \ell^{-2}\mathbf{I}$ where ℓ is a constant, i.e. $\ell_i = \ell$. Note that there are many choices for covariance function. Gaussian covariance in Equation (5.16) is used only for introducing GPR to the reader. Further details on possible covariance functions are discussed later in this section.

Now we write the Gaussian process as,

$$f(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right). \tag{5.17}$$

We briefly acknowledge the reader on the following definitions and relations used in probability theories [171, 178].

- **Random variable**: A discrete or continuous variable which takes up possible values, whose outcomes are numerical from a random phenomenon. For instance, a discrete random variable x can take x_1 and x_2 out of a random process.

- **Joint probability**: Consider two random variables x and y. Probability of having $\mathrm{x} = x$ and $\mathrm{y} = y$ simultaneously (or jointly) is named the joint probability of two random variables. We denote it as $P(\mathrm{x} = x, \mathrm{y} = y)$ or $P(x, y)$.

- **Marginal probability**: The probability of any single event occurring unconditioned on any other events. For instance, considering two discrete random variables x and y, we write the marginal probability of $\mathrm{x} = x$ as $P(\mathrm{x} = x)$,

$$\forall x \in \mathrm{x},\ P(\mathrm{x} = x) = \sum_{y} P(\mathrm{x} = x, \mathrm{y} = y)\ . \tag{5.18}$$

- **Conditional probability**: The probability of some event, given that (conditioned on) some other event has happened is called conditional probability. We write the conditional probability of two random variables x and y as,

$$P(\mathrm{y} = y|\mathrm{x} = x) = \frac{P(\mathrm{y} = y, \mathrm{x} = x)}{P(\mathrm{x} = x)}\ . \tag{5.19}$$

- **Bayes' theorem**: This describes the probability of an event, based on prior knowledge of conditions that might be related to the event. We write the following based on Equations (5.18) and (5.19),

$$P(\mathrm{y} = y, \mathrm{x} = x) = P(\mathrm{y} = y|\mathrm{x} = x)P(\mathrm{x} = x) = P(\mathrm{x} = x|\mathrm{y} = y)P(\mathrm{y} = y) \tag{5.20}$$

$$\implies P(\mathrm{y} = y|\mathrm{x} = x) = \frac{P(\mathrm{x} = x|\mathrm{y} = y)P(\mathrm{y} = y)}{P(\mathrm{x} = x)} \tag{5.21}$$

$$\text{or } \implies P(\mathrm{x} = x|\mathrm{y} = y) = \frac{P(\mathrm{y} = y|\mathrm{x} = x)P(\mathrm{x} = x)}{P(\mathrm{y} = y)}\ . \tag{5.22}$$

  where Equations (5.21) and (5.22) are two forms to interpret Bayes' theorem.

Prediction with noise-free observations

Now we move forward to an in-depth discussion on using GPR to train a model and make predictions. Marginalisation property implied from the GPR definition in Equation (5.17) leads us to write, for a general case,

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \ , \tag{5.23}$$

where $\boldsymbol{\mu}$ is the vectorised mean and Σ is the covariance matrix of the normal distribution $\mathcal{N}$. Then Equation (5.23) must also satisfy $\boldsymbol{y}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_{11})$ with $\boldsymbol{y}_1$, $\boldsymbol{\mu}_1$ and Σ_{11} being relevant parts of $\boldsymbol{y}$, $\boldsymbol{\mu}$ and Σ, respectively.

Equation (5.15) revealed that the choice of mean function is arbitrary. To this end, we assume a zero mean as a *prior*, and therefore, the joint distribution of training and test outputs are written as,

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \ , \tag{5.24}$$

where we decompose the covariance matrix Σ in Equation (5.23) following the definitions given in Table 5.2. We remind the reader on the following identity in relevance to conditioning and marginalisation of joint distributions.

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} A & C \\ C^{\mathrm{T}} & B \end{bmatrix}\right) \tag{5.25a}$$

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, A) \tag{5.25b}$$

$$\boldsymbol{x}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}_x + CB^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y), A - CB^{-1}C^{\mathrm{T}}) \ . \tag{5.25c}$$

We can use Equation (5.25) directly on the joint distribution in Equation (5.24). Moreover, predictions based on the conditioning of the joint Gaussian prior distribution on the observations and test inputs follows a Gaussian distribution similar to Equation (5.25c) as,

$$\begin{aligned} \boldsymbol{y}_*|X_*, X, \boldsymbol{y} \sim \mathcal{N}(&K(X_*,X)K(X,X)^{-1}\boldsymbol{y}, \\ &K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)) \ . \end{aligned} \tag{5.26}$$

Hence the best estimate for $\boldsymbol{y}_*$ is the mean of the distribution (denoted as $\overline{\boldsymbol{y}}_*$) in Equation

(5.26) given by,

$$\overline{\boldsymbol{y}}_* = K(X_*, X)K(X, X)^{-1}\boldsymbol{y} \ , \tag{5.27}$$

and the uncertainty of the estimate is given by the variance of the Gaussian distribution as,

$$\text{var}(\boldsymbol{y}_*) = K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*). \tag{5.28}$$

In the presence of noisy-observations, $K(X, X)$ in Equations (5.24), (5.26), (5.27) and (5.28) will simply be replaced by $K(X, X) + \sigma_n^2\mathbf{I}$ where the independent observation errors ϵ are assumed to be of a Gaussian distribution, i.e. $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$.

Hyperparameter estimation

Equation (5.26) furnishes the Gaussian distribution of the predictions (conditioned on the database). However, matrix K remains unknown because we did not specify the hyperparameters of the covariance functions. For instance, squared exponential covariance function given in Equation (5.16) has two hyperparameters σ_f^2 and ℓ that control the entries in K. Hence, it is necessary to systematically arrive at the best possible hyperparameters that will not overfit the trained GPR model. We use the Maximum Log Marginal Likelihood (MLML) approach to obtain the optimum hyperparameters that maximise the log marginal likelihood.

First, we establish the relationships between posterior, likelihood, prior and marginal likelihood given by,

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}} \ , \qquad P(\boldsymbol{\Theta}|\boldsymbol{y}, X) = \frac{P(\boldsymbol{y}|X, \boldsymbol{\Theta})P(\boldsymbol{\Theta})}{P(\boldsymbol{y}|X)} \ . \tag{5.29}$$

To this end, we write the marginal likelihood using Equations (5.18) and (5.29) as,

$$P(\boldsymbol{y}|X) = \int P(\boldsymbol{y}|X, \boldsymbol{\Theta})P(\boldsymbol{\Theta})\mathrm{d}\boldsymbol{\Theta} \ , \tag{5.30}$$

where model parameters are conditioned on hyperparameters.

Marginal likelihood in Equation (5.30) can be further simplified in order to obtain an analytically tractable form of the integral as follows [174]. This is achieved by first taking the logarithm of the marginal likelihood function given in Equation (5.30) and

simplifying the terms to arrive at,

$$\log\, P(\boldsymbol{y}|X) = -\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}K^{-1}\boldsymbol{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi\ , \tag{5.31}$$

where $|K|$ denotes the determinant of K. Additionally, the term $\frac{n}{2}\log 2\pi$ in (5.31) is a normalising constant that depends only on the size of the training database n.

Determination of the hyperparameters $\boldsymbol{\Theta}$ departs from maximising the log marginal likelihood as given in Equation (5.31). Gradient-based optimisation tools require the partial derivatives of the log marginal likelihood with respect to hyperparameters. This is written as,

$$\begin{aligned}\frac{\partial}{\partial\Theta_j}\log\, P(\boldsymbol{y}|X) &= \frac{1}{2}\boldsymbol{y}^{\mathrm{T}}K^{-1}\frac{\partial K}{\partial\Theta_j}K^{-1}\boldsymbol{y} - \frac{1}{2}\mathrm{Tr}\left(K^{-1}\frac{\partial K}{\partial\Theta_j}\right)\\ &= \frac{1}{2}\mathrm{Tr}\left(\left(\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}} - K^{-1}\right)\frac{\partial K}{\partial\Theta_j}\right) \qquad \text{where } \boldsymbol{\alpha} = K^{-1}\boldsymbol{y}\ .\end{aligned} \tag{5.32}$$

Note that computational complexity of training a GPR model lies on the inversion of the Gram matrix K. Standard matrix inversion methods for $n \times n$ positive definite symmetric matrices require time $\mathcal{O}(n^3)$. Therefore, GPR machine learning is inefficient in handling very large databases.

Covariance functions

The Gram matrix K, also referred to as the design matrix, requires a stringent requirement of being positive semidefinite [174]. Therefore, any arbitrary function of input pairs $\boldsymbol{x}$ and $\boldsymbol{x}'$ will not, in general, be a valid covariance (or kernel) function. We discuss, in brief, the possible covariance functions that can be used to produce the design matrix K. All covariance functions are broadly categorised as *stationary* and *non-stationary*. A stationary covariance function is a function of $\boldsymbol{x} - \boldsymbol{x}'$. All other covariance functions come under non-stationary covariance functions. Both stationary and non-stationary covariance functions can be further divided into subcategories. However, we do not intend to go into further details. Interested readers are invited to refer to Rasmussen and Williams [174] for more information.

Now we start introducing some of the widely used covariance functions.

Squared exponential covariance function: This was earlier introduced in Equation (5.16). Squared exponential covariance function is infinitely differentiable and usually carries two hyperparameters: lengthscale (ℓ) and scaling (σ_f^2) hyperparameters.

The Matérn class of covariance functions: The Matérn class of covariance function is stationary and identified as a generalization of the squared exponential covariance function. It has an additional parameter ν which controls the smoothness of the resulting covariance function. We write the commonly used two forms of Matérn covariance functions as,

$$k_{\nu=3/2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(1 + \frac{\sqrt{3}r}{\ell}\right) \exp\left(-\frac{\sqrt{3}r}{\ell}\right) \tag{5.33a}$$

$$k_{\nu=5/2}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}r}{\ell}\right) \ , \tag{5.33b}$$

where $r = |\boldsymbol{x} - \boldsymbol{x}'|$ and ν, ℓ, σ_f are identified as the hyperparameters. Moreover, Equations (5.33a) and (5.33b) are, respectively, at least once or twice differentiable. A more general equation for Matérn class of covariance functions can be found in the literature [174, 175].

γ-Exponential covariance function: This family of covariance functions includes both the exponential and squared exponential functions given by,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(-(r/\ell)^\gamma\right) \qquad 0 < \gamma \leq 2 \ , \tag{5.34}$$

where γ, ℓ and σ_f^2 are the hyperparameters.

Rational quadratic covariance function: The Rational quadratic covariance function can be seen as a scale mixture (an infinite sum) of squared exponential covariance functions with different characteristic lengthscales given by,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \left(1 + \frac{(\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}} \mathbf{M} (\boldsymbol{x} - \boldsymbol{x}')}{2\alpha}\right)^{-\alpha} \ , \tag{5.35}$$

where $\alpha > 0$, ℓ and σ_f^2 are identified as hyperparameters of the rational quadratic covariance function.

Squared Exponential-Sine covariance function: This is a popular choice in modelling periodic functions and can be seen as a combination of sine and exponential functions.

It takes the form,

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left[-2\left(\sin(\pi/p * r)/\ell\right)^2\right] \ , \tag{5.36}$$

where $p > 0$, ℓ and σ_f^2 are identified as hyperparameters of the squared Exponential-Sine covariance function.

Dot-product covariance function: The dot-product covariance function is non-stationary and takes the following form.

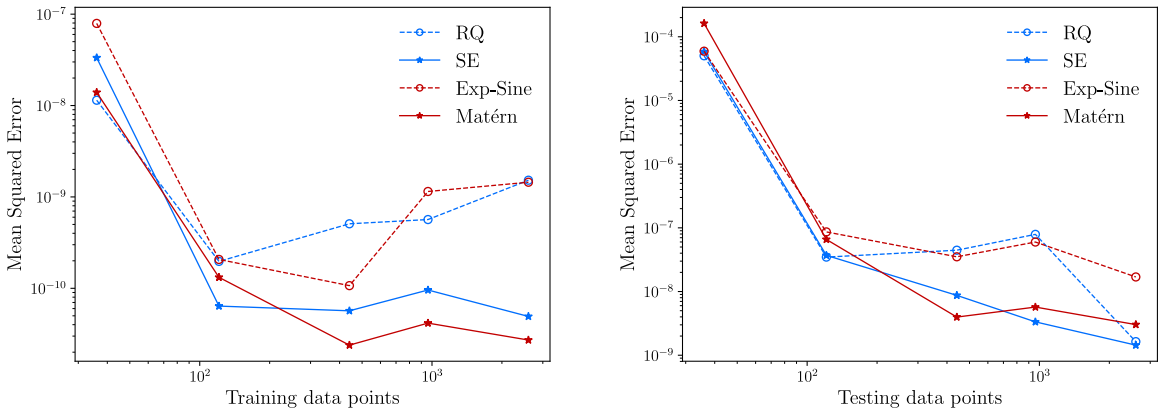
$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2(\sigma_0^2 + \boldsymbol{x} \cdot \boldsymbol{x}')^p \ , \tag{5.37}$$

where $p > 0$, σ_0^2 and σ_f^2 are identified as hyperparameters of the dot-product covariance function.

Different covariance functions mentioned above can be combined to form new covariance functions that can be used in GPR. A more detailed discussion on this topic can be found in Bishop [37].

Next, a comparative study is carried out to investigate the influence of covariance functions on the prediction errors (see Figure 5.4). For this, a sample dataset from weft-knitted textile microscale simulations has been used. Further details on the weft-knitted fabric microscale simulations can be found later in Section 6.2. Furthermore, tabulated results of generating Figure 5.4 and corresponding machine learning algorithm settings can be found in Appendix A.2.1.

Additional details on database construction, mean squared error (MSE) calculations and the use of scikit-learn library is provided in Section 5.3. Specifically, database construction is given in Equation (5.46) and MSE of energy prediction is defined in Equation (5.48).



(a) (b)

Figure 5.4: Influence of covariance functions on the energy potential mean squared errors of (a) training data and (b) testing data. RQ and SE are abbreviated, respectively, for Rational Quadratic and Squared Exponential covariance functions.

As we see from Figure 5.4 Matérn and squared exponential covariance functions tend to converge better than the other two. Choice of covariance functions is purely governed by the nature of application and type of data that forms databases.

Partial derivatives of the posterior mean

Many applications require the partial derivatives of predictions with respect to the input variables. In the computational mechanics context of a hyperelastic material, predicted strain energy (ψ) needs to be differentiated with respect to the Green-Lagrange strain tensor ($\boldsymbol{E}$) components to arrive at the second Piola-Kirchhoff stress tensor ($\boldsymbol{S}$) components. Moreover, because differentiation is a linear operation, a derivative of a Gaussian process remains a Gaussian process [179].

Consider a single new data point as $\boldsymbol{x}_*$. Thus $K(X_*, X)$ matrix has one row and d number of columns. The aim is to find the partial derivatives with respect to this new test point $\boldsymbol{x}_*$, so we have a closed-form equation for the first derivative of the posterior

mean obtained in Equation (5.27).

$$\begin{aligned}\frac{\partial \overline{y}_*}{\partial \boldsymbol{x}_*} &= \frac{\partial}{\partial \boldsymbol{x}_*}\left(K(X_*, X)K(X, X)^{-1}\boldsymbol{y}\right) \\ &= \frac{\partial K(X_*, X)}{\partial \boldsymbol{x}_*}K(X, X)^{-1}\boldsymbol{y}\end{aligned} \tag{5.38}$$

To this end, we choose a squared exponential covariance function to derive the analytical partial derivatives of the predicted mean. This choice is justified by the following reasons.

- Smoothness - Squared exponential covariance function is infinitely differentiable, therefore very smooth.
- Analytically tractable - It is convenient to derive the compact forms upon differentiation and integration.
- Widely used in literature - Many research communities use squared exponential covariance functions [61, 167, 180, 181].
- Accuracy - A good accuracy compared to other covariance functions. See Figure 5.4.

Now we write the vector $k(\boldsymbol{x}_*, \boldsymbol{x}_i)$ as,

$$k(\boldsymbol{x}_*, \boldsymbol{x}_i) = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_* - \boldsymbol{x}_i)^{\mathrm{T}}\mathbf{M}(\boldsymbol{x}_* - \boldsymbol{x}_i)\right) \ , \tag{5.39}$$

and its derivatives with respect to the new input $\boldsymbol{x}_*$ as,

$$\frac{\partial k(\boldsymbol{x}_*, \boldsymbol{x}_i)}{\partial \boldsymbol{x}_*} = -\mathbf{M}(\boldsymbol{x}_* - \boldsymbol{x}_i)k(\boldsymbol{x}_*, \boldsymbol{x}_i) \ . \tag{5.40}$$

Substituting Equation (5.40) in Equation (5.38) produces the partial derivative of the posterior mean $\overline{\boldsymbol{y}}_*$ with respect to $\boldsymbol{x}_*$. This is written as,

$$\frac{\partial \overline{y}_*}{\partial \boldsymbol{x}_*} = -\mathbf{M}\left[\hat{X}_*^{\mathrm{T}}\left(\boldsymbol{k}(\boldsymbol{x}_*, X)^{\mathrm{T}} \odot K(X, X)^{-1}\boldsymbol{y}\right)\right] \ , \tag{5.41}$$

where $\hat{X}_*$ is given by $[\boldsymbol{x}_* - \boldsymbol{x}_1, \boldsymbol{x}_* - \boldsymbol{x}_2, ..., \boldsymbol{x}_* - \boldsymbol{x}_n]^{\mathrm{T}}$ and $\odot$ denotes an element-wise product, also known as Hadamard or Schur product. That is, for two vectors of the

same size (say $N \times 1$), $(\boldsymbol{a} \odot \boldsymbol{b})_i = a_i b_i$ for $i = 1, 2, \cdots, N$, where summation convention is not implied over i.

Next, second-order partial derivatives are obtained following a similar manner. In a computational mechanics context, these second derivatives are analogous to the fourth-order material tangent tensor. That is, $\boldsymbol{\mathcal{C}} = \partial\psi/\partial\boldsymbol{E}\partial\boldsymbol{E}$. In addition, $\boldsymbol{\alpha} = K(X,X)^{-1}\boldsymbol{y}$ has been used henceforth to simplify the expressions as it is not a $\boldsymbol{x}_*$ dependent. First, for convenience, Equation (5.41) is expanded to arrive at,

$$\frac{\partial \overline{y}_*}{\partial \boldsymbol{x}_*} = -\mathbf{M}\left[(\boldsymbol{x}_* - \boldsymbol{x}_1)k(\boldsymbol{x}_*, \boldsymbol{x}_1)\alpha_1 + \ldots + (\boldsymbol{x}_* - \boldsymbol{x}_n)k(\boldsymbol{x}_*, \boldsymbol{x}_n)\alpha_n\right] \ . \tag{5.42}$$

Now we differentiate Equation (5.42) to derive the second partial derivative of the posterior mean as,

$$\begin{aligned}
\frac{\partial^2 \overline{y}_*}{\partial \boldsymbol{x}_* \partial \boldsymbol{x}_*} &= \frac{\partial}{\partial \boldsymbol{x}_*}[-\mathbf{M}\hat{X}_*^{\mathrm{T}}\left(\boldsymbol{k}(\boldsymbol{x}_*, X)^{\mathrm{T}} \odot \boldsymbol{\alpha}\right)] \\
&= -\mathbf{M}\left(\frac{\partial \hat{X}_*^{\mathrm{T}}}{\partial \boldsymbol{x}_*}\left(\boldsymbol{k}(\boldsymbol{x}_*, X)^{\mathrm{T}} \odot \boldsymbol{\alpha}\right) + \hat{X}_*^{\mathrm{T}}\frac{\partial \boldsymbol{k}(\boldsymbol{x}_*, X)^{\mathrm{T}} \odot \boldsymbol{\alpha}}{\partial \boldsymbol{x}_*}\right) \ .
\end{aligned} \tag{5.43}$$

Using the expanded Equation (5.42), same Equation (5.43) can be obtained, which looks more elegant for algorithmic implementations.

$$\begin{aligned}
\frac{\partial^2 \overline{y}_*}{\partial \boldsymbol{x}_* \partial \boldsymbol{x}_*} &= -\mathbf{M}\frac{\partial}{\partial \boldsymbol{x}_*}\left[(\boldsymbol{x}_* - \boldsymbol{x}_1)k(\boldsymbol{x}_*, \boldsymbol{x}_1)\alpha_1 + \ldots + (\boldsymbol{x}_* - \boldsymbol{x}_n)k(\boldsymbol{x}_*, \boldsymbol{x}_n)\alpha_n\right] \\
&= -\mathbf{M}\left[\frac{\partial(\boldsymbol{x}_* - \boldsymbol{x}_1)}{\partial \boldsymbol{x}_*}k(\boldsymbol{x}_*, \boldsymbol{x}_1)\alpha_1 + (\boldsymbol{x}_* - \boldsymbol{x}_1)\frac{\partial k(\boldsymbol{x}_*, \boldsymbol{x}_1)}{\partial \boldsymbol{x}_*}\alpha_1 + \ldots\ldots\right] \\
&= -\mathbf{M}\left[\ \mathbf{I}k(\boldsymbol{x}_*, \boldsymbol{x}_1)\alpha_1 - (\boldsymbol{x}_* - \boldsymbol{x}_1) \otimes (\mathbf{M}(\boldsymbol{x}_* - \boldsymbol{x}_1))k(\boldsymbol{x}_*, \boldsymbol{x}_1)\alpha_1 + \ldots\ldots\right] \\
&= -\mathbf{M}\left[(\mathbf{I} - (\boldsymbol{x}_* - \boldsymbol{x}_1) \otimes (\mathbf{M}(\boldsymbol{x}_* - \boldsymbol{x}_1)))\, k(\boldsymbol{x}_*, \boldsymbol{x}_1)\alpha_1 + \ldots\ldots\right] \\
\frac{\partial^2 \overline{y}_*}{\partial \boldsymbol{x}_* \partial \boldsymbol{x}_*} &= -\mathbf{M}\sum_{i=1}^{n}\left(\mathbf{I} - (\boldsymbol{x}_* - \boldsymbol{x}_i) \otimes (\mathbf{M}(\boldsymbol{x}_* - \boldsymbol{x}_i))\right) k(\boldsymbol{x}_*, \boldsymbol{x}_i)\alpha_i \ .
\end{aligned} \tag{5.44}$$

5.3 Data-driven multiscale modelling

5.3.1 Scale transition framework

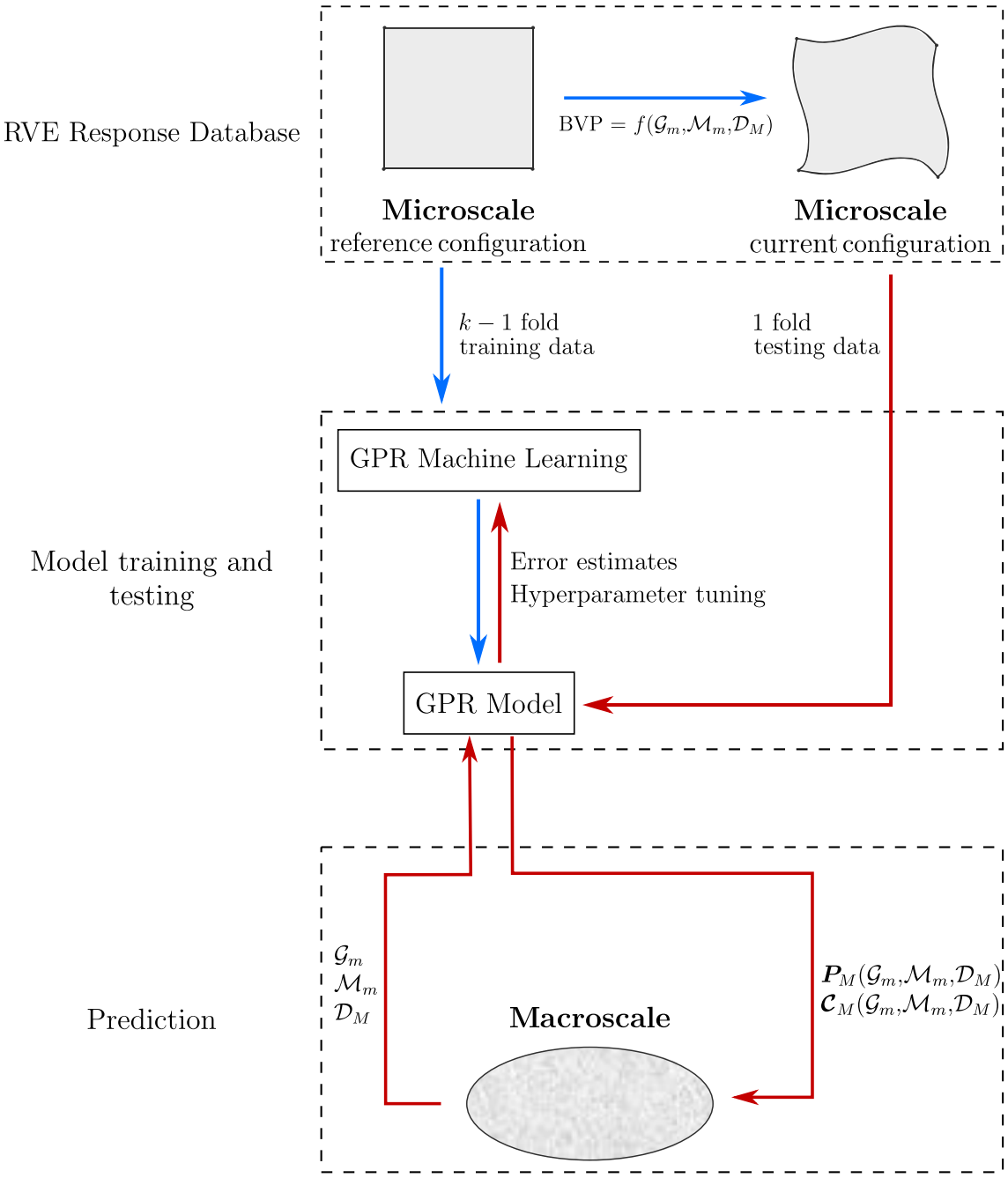


Figure 5.5: Data-driven multiscale modelling and material designing framework using GPR machine learning

Having discussed the basic principles of conventional multiscale modelling and GPR machine learning, now we amalgamate them to introduce a homogenisation technique (see Figure 5.5) for the analysis of textile materials. The novelty in the proposed technique can be identified as a neat extension to the recent work by Bessa et al. [61]. Bessa et al. used neural networks in multiscale modelling and worked predominantly towards the designing of material systems. Our work, first, includes a complete multiscale modelling technique to simulate thin membranes. Here, we train a GPR model using a response database constructed by microscale RVE simulations. This GPR model is later integrated to our in-house thin-shell solver to simulate membranes in the macroscale. Secondly, we embed non-deformation measures, i.e. geometric and material properties, into the response database to enhance the flexibility and versatility of the proposed multiscale modelling technique. This eventually leads to a range of potential applications in multiscale modelling and material systems designing. Figure 5.5 illustrates the proposed data-driven homogenisation strategy.

In a big picture, as shown in Figure 5.5, we identify three main sub-processes in the proposed multiscale strategy, namely: RVE response database construction; GPR Model training and testing; Making predictions.

5.3.2 RVE response database construction

Development of a complete space-filling response database is pivotal in machine learning. This is achieved by designing experiments to simulate microscale RVEs constrained by predefine BVPs. These BVPs, as shown in Figure 5.5, has three main ingredients, namely $\mathcal{G}$eometric, $\mathcal{M}$aterial and $\mathcal{D}$eformation.

1. $\mathcal{G}$eometric properties - Geometric properties include RVE spatial dimensions, yarn cross-section properties (for fibre-based RVEs), initial curvatures (or angles), waviness parameters of sinusoidal RVEs etc. Care should be taken when there are intrinsic relationships between two geometric properties. Geometric properties govern the material design aspects of the proposed framework in Figure 5.5.

2. $\mathcal{M}$aterial properties - Material properties of a RVE refer to the Young's moduli and Poisson's ratios of a general heterogeneous linear material model. Material nonlinearities can also be included in a systematic manner. Material properties also govern the material design aspects of the proposed framework in Figure 5.5.

3. $\mathcal{D}$eformation states - Deformation states provide the basis of multiscale modelling. Usually, deformation states are taken to be any strain tensor (small strain, Euler-Almansi,logarithmic, Green-Lagrange), deformation gradient or a combination of displacements and rotations. In this thesis, we use the deformation gradient to apply boundary constraints, as given in Equation (5.6), and compute Green-Lagrange tensor components to construct the response database.

It is important to establish design domains to the aforementioned three ingredients of RVE experiments. This ensures that the response database is well-defined and complete. For instance, we consider two geometric, two material and three plane-stress Green-Lagrange strain descriptors to construct a response database. Following the notations in Table 5.1 and Figure 5.5 we write,

$$\mathcal{G}_{\mathrm{m}} = \begin{cases} \mathcal{G}^1_{\mathrm{m}} & \mathcal{G}^1_{\mathrm{m}} \in [a^1_{\mathrm{m}}, b^1_{\mathrm{m}}] \\ \mathcal{G}^2_{\mathrm{m}} & \mathcal{G}^2_{\mathrm{m}} \in [a^2_{\mathrm{m}}, b^2_{\mathrm{m}}] \end{cases} \tag{5.45a}$$

$$\mathcal{M}_{\mathrm{m}} = \begin{cases} \mathcal{M}^1_{\mathrm{m}} = \mathrm{E} & \mathrm{E} \in [10^5, 10^9] \\ \mathcal{M}^2_{\mathrm{m}} = \nu & \nu \in [0.0, 0.5] \end{cases} \tag{5.45b}$$

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}^1_{\mathrm{M}} = E_{11} & E_{11} \in [-0.5, 1.5] \\ \mathcal{D}^2_{\mathrm{M}} = E_{22} & E_{22} \in [-1.5, 1.0] \\ \mathcal{D}^3_{\mathrm{M}} = E_{12} & E_{12} \in [-0.5, 0.5], \end{cases} \tag{5.45c}$$

where positive a_m and b_m ($a_m < b_m$) bound the two geometric descriptors. Note that the above values and descriptors are arbitrarily chosen and used only to elaborate on the response database construction process.

Note: If we use only $\mathcal{D}_{\mathrm{M}}$ to construct the response database, this is equivalent to data-driven computational homogenisation. Further expansion of the response database domain (adding $\mathcal{G}_m$, $\mathcal{M}_m$) leads to data-driven homogenisation integrated to materials system designing. The former is discussed in this section, and the latter is explored in detail under Section 5.4.

At this juncture, we express the response databases for data-driven multiscale modelling. As this decision is purely macroscale application-based, we resort to plane-stress thin-shell mechanics introduced in Section 4.2.1 for the response database construction. Hence, a plane-stress response database is defined as $(\boldsymbol{x}_i, y_i)$ where $\boldsymbol{x} = \{E_{11}, E_{22}, E_{12}\}$

and $y = \psi$. In other words, we furnish in-plane strains and strain energy density (ψ) in the construction of a response database. We use energy density as the target output for energetic consistency of thin-shell mechanics. Moreover, the same choice allows us to write the Second Piola-Kirchhoff stress tensor $\boldsymbol{S} = \frac{\partial \psi}{\partial \boldsymbol{E}}$ and Material constitutive tensor $\boldsymbol{\mathcal{C}} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}}$.

5.3.3 GPR Model training and testing

GPR machine learning technique discussed in Section 5.2.2 is now used to train a GPR model using a constructed response database. Python-based scikit-learn library [175] is used for this purpose. Following steps were taken to complete the GPR model training and testing process.

First, the response database used for plane-stress data-driven computational homogenisation is succinctly written as,

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}^1_{\mathrm{M}} = E_{11} & E_{11} \in [E^{\min}_{11}, E^{\max}_{11}] \\ \mathcal{D}^2_{\mathrm{M}} = E_{22} & E_{22} \in [E^{\min}_{22}, E^{\max}_{22}] \\ \mathcal{D}^3_{\mathrm{M}} = E_{12} & E_{12} \in [E^{\min}_{12}, E^{\max}_{12}] \ . \end{cases} \tag{5.46}$$

where $(\bullet)^{\min}$ and $(\bullet)^{\max}$ refer to the predefined minimum and maximum bounds of a quantity $(\bullet)$, respectively. Values for these bounds are user-defined and the range $((\bullet)^{\max} - (\bullet)^{\min})$ depends on the desired application.

GPR model training and testing: We start by randomly shuffling the data points in the response database. Next, the response database is split into equal (approximately) k folds for k fold cross-validation. Empirically proven values for k take values $k = 5, 10$ [182]. The special case of $k = 1$ is known as Leave One Out Cross-Validation (LOOCV). As shown in Figure 5.5, $k-1$ fold data from the randomly shuffled response database is used to train a GPR model and the remaining fold, in turn, is used to test the trained model. In simple terms, the process is repeated k times until every fold becomes a test fold. In this thesis, we use $k = 5$. During this process, it is essential to establish performance indicators to quantitatively and qualitatively assess cross-validation.

Error estimates: We use Mean Squared Error (MSE) and co-efficient of determination (R^2) to quantify the accuracy of cross-validation.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_i)^2} \tag{5.47}$$

$$\text{MSE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.48}$$

where y_i, $\hat{y}$ and $\overline{y}$ stand for target output, predicted output and mean of the target outputs, respectively. Furthermore, as stress is a tensorial quantity, we take the norm of mean squared errors of individual stress components,

$$\text{MSE}(||\boldsymbol{S}||) = \sqrt{\text{MSE}^2(S_{11}, \hat{S}_{11}) + \text{MSE}^2(S_{22}, \hat{S}_{22}) + \text{MSE}^2(S_{12}, \hat{S}_{12})}\ . \tag{5.49}$$

Hyperparameter tuning: Overfitting and underfitting are two of the most prevalent machine learning challenges [41, 171]. There are several ways to circumvent these two challenges.

Overfitting occurs in analyses that regress too closely or exactly to a particular set of data, and may therefore, fail to fit new data or predict future observations reliably. A simple example would be to fit three (non-coincident) collinear points using a polynomial of degree more than one. A quadratic or any higher degree polynomial can exactly represent the given three points but the data, in reality, could be following a simple linear model. Therefore, when a model is overfitted, we observe very low MSE and R^2 close to 1 for the training dataset but not for the testing dataset.

In contrast, underfitting, as the name implies, refers to a model that does not fit well even for the training data. Intuitively, underfitting occurs when the model cannot trace the underlying trend of the training data. Imagine using a linear model to capture the trend of a dataset that in reality follows a high-amplitude sinusoid. Underfitting always leads to a comparatively higher MSE and near-zero R^2 values for the training dataset.

Hyperparameter tuning keeps track for any occurrences of overfitting or underfitting. From our experience, GPR models usually do overfit but not underfit. This happens mainly due to the use of infinitely smooth squared exponential covariance functions. Overfitting is triggered when hyperparameters reach their predefined lower bounds or when the hyperparameter bounds are too wide. In addition to very low training errors and high testing errors, overfitting is revealed when optimised hyperparameters and

MLML significantly differ in each fold of k fold cross-validations. When this happens, the hyperparameter setting is revised so that overfitting is avoided in the final GPR model.

5.3.4 Prediction

Lastly, the trained GPR model is used to make predictions based on new data points departing from the macroscale problem. At the reference (stress-free) configuration, all three strain components (zero in magnitude) will arrive at the trained GPR model to predict the internal strain energy density, first Piola-Kirchhoff stress tensor and material constitutive tensor. Likewise, at every load step and equilibrium iteration, the enlisted strain components at integration points will communicate with the GPR model to make predictions. This step is illustrated in Figure 5.5 as the third key step in the proposed data-driven homogenisation technique.

Sampling of the input feature space is also important for the accurate GPR model training, testing and prediction. This thesis investigates uniform and Sobol sequence sampling methods where the latter is a quasirandom sampling technique [183].

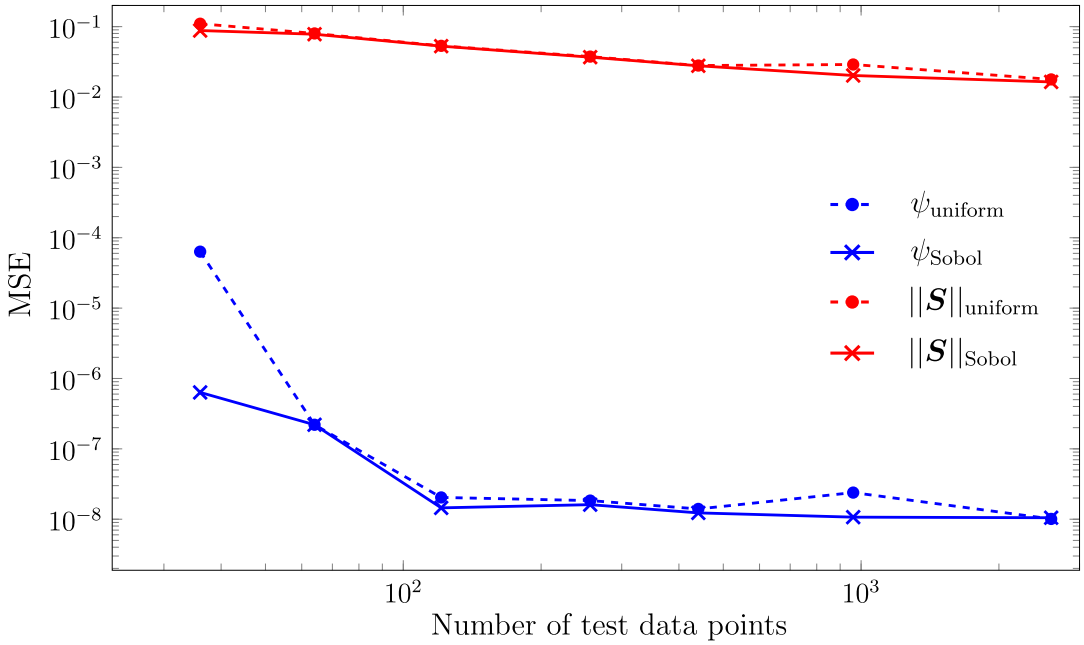


Figure 5.6: Testing errors of the predictions using uniform and Sobol sampling methods.

We use a set of microscale analyses from Section 6.2 to assess the influence from the sampling technique on the GPR model training, testing and prediction. Figure 5.6 shows the comparison of the two sampling techniques. Sobol sequence sampling method performs better for smaller databases in comparison to uniform sampling. However, there is no convincing evidence to reject either of these two methods as they converge similar to each other. This is mainly due to the non-existing redundancies in the input variables. Random sampling is used to detect any redundant variables and, in the case of strain components, it is certain that in-plane strains are independent. Therefore, we observe similar errors in both sampling methods.

One of the main advantages of the proposed technique is its ability to operate off-line from element-wise microscale simulations. As previously discussed, conventional multiscale strategies link every element of the macroscale to a microscale simulation, thus, become computationally very expensive. The proposed data-driven method circumvents this issue by simply feeding all macroscale strains into the trained GPR model and extracting relevant quantities as GPR predictions.

5.4 Design of materials

Data-driven material design is proposed in this section as a neat extension to the proposed data-driven multiscale modelling strategy in the previous section. As illustrated in Figure 5.5 and given in Equation (5.45), we enhance the response database to accommodate material design parameters. In addition to the homogenised strains $\mathcal{D}_{\mathrm{M}}$, we include yarn material parameters $\mathcal{M}_{\mathrm{m}}$ and geometric parameters of RVEs, $\mathcal{G}_{\mathrm{m}}$, to construct the response databases. However, certain prerequisites need to be satisfied when selecting parameters for database construction. These constraints play a major role in developing accurate response databases. Because, one would validate the numerical RVE response against experimental measures, only for a fixed set of geometric and material parameters. Therefore, any material design enhancements of the response databases might give accurate-looking numerical solutions, but such enhancements may not be feasible from a physical or manufacturing standpoint.

5.4.1 Prerequisites

A set of constraints are identified as prerequisites to be satisfied in material designing of woven and knitted textiles. These are enforced systematically so that the final GPR models will produce accurate results and will not lead to any numerical instabilities in the macroscale finite element analyses.

Penetrations/Collisions: Attention must be given to large self-collisions or yarn-to-yarn contacts at the reference configurations. This might happen if one merely increases the radius of yarns without considering the change of thickness of the RVE. Also, large initial penetration depths will cause ill-conditioned constraint enforcements and lead to poor convergence of RVE simulations.

Scalability: It is important to bring all the inputs to a comparable scale for numerical stability. For instance, we will have strains usually within a range of $0.0 \leq |E_{ij}| \leq 1.0$. In contrast, when we try to incorporate true values of yarn Young's modulus ($\mathrm{E} \sim 10^6$) into the response database, model training would find it difficult to converge due to parameter scalability issues. Thus, we normalise varying values of Young's moduli by a base Young's modulus, to scale the input parameters. Hence instead of using true values of E, we define the normalisation as,

$$\mathrm{E}_{\mathrm{scaled}} = \frac{\mathrm{E}}{\hat{\mathrm{E}}} \qquad \text{where} \qquad \hat{\mathrm{E}} = \frac{\mathrm{E}_{\mathrm{max}} + \mathrm{E}_{\mathrm{min}}}{2} \ . \tag{5.50}$$

In a similar way, we normalise the yarn radii R as $\mathrm{R}_{\mathrm{scaled}} = \mathrm{R}/\hat{\mathrm{R}}$.

Above normalisations are not intended to be within a specified range. But sometimes it is preferred to use a min-max normalisation to confine the normalised variable within a given range. Moreover, $\hat{\mathrm{E}}$ in Equation (5.50) can also be a user-defined reference Young's modulus.

Numerical stability: Whether we are using a data-driven approach or conventional multiscale modelling technique, eventually we resort to solving a linear system of equations. Therefore it is crucial to keep the numerical stability of the linear solver during the equilibrium iterations. We aim to maintain the stability of the solver by avoiding the ill-conditioning of the system stiffness matrix and load vector [184]. At this juncture, a set of cases are identified for the ill-conditioning of the system stiffness matrix and the load vector.

– Young's moduli: In the case of woven fabrics, we can have two different yarns in either direction. Minimum and maximum Young's moduli should be chosen carefully so that when we use the minimum of one yarn and the maximum of the other we will not anticipate any numerical instabilities of the linear solvers.

– Yarn Radii: Yarn radii could also be a candidate to make the stiffness matrix ill-conditioned. Unlike the Young's moduli, the influence of yarn radii on the stiffness matrix is not linear. Axial and bending stiffnesses are proportional to yarn radius in the orders of two and four, respectively. Hence, from a numerical standpoint, yarn radii can cause significant numerical instabilities of the linear solver compared to Young's moduli.

– Characteristic lengths: To respect the principle of separation of scales, we choose a macroscale mesh of which the finite elements are coarser than that of the RVE size [18]. Also, classical 2D finite elements are recommended to have a maximum aspect ratio of five but preferably less than three [71]. Therefore, when constructing the response databases, care should be taken in varying the spatial dimensions of RVEs not to violate these aspect ratio restrictions.

Variable dependencies: Attention must be given to dependent variables when constructing the response databases. For instance, the amplitude of sinusoidal plain-woven yarns is sometimes parameterised by the wavelengths [18]. Thus, when we vary wavelengths to construct the response database, a natural need arises to vary the wave amplitudes. But on the other hand, identifying these variable dependencies, we reduce the dimension of the feature space of inputs. This can be interpreted as a simplified version of Principal Component Analysis (PCA) [185], which is analogous to the Principal axis theorem in mechanics. PCA systematically reduces the real dimensions (D) to a set of principle dimensions ($d < D$), to represent the same data, by identifying correlations between features of dimension D [186]. This data restructuring is a pivotal step in machine learning to reduce the computational cost of analysing very large databases.

5.4.2 Framework for data-driven material design

Conditioned on the highlighted prerequisites in Section 5.4.1, we now explain the procedure to use machine learning in a material design framework. In other words, we can see this step as an extension to the data-driven multiscale modelling as we now couple

the scales not only using deformation measures (see Figure 5.1) but also embedding the geometric and material parameters of the yarns (see Figure 5.5). This data-driven material design strategy would now answer to the following questions.

- How would the nonlinear stress-strain relationship look like, if I change the yarn material or RVE geometry?
- How can I compare the homogenised response of two or more different (but comparable) textiles without repeating the same RVE modelling and simulation process?
- How sensitive are the material and geometric parameters on the homogenised response?
- How would the material and geometric parameters interact with each other and influence the overall response in the macroscale?
- Can we enhance this scheme to optimise the mechanical response of technical textiles based on different yarn arrangements?

Next, we discuss the details of including $\mathcal{G}_\mathrm{m}$ and $\mathcal{M}_\mathrm{m}$ in response databases for data-driven material design. Readers are reminded of the notation convention where subscript m is used to denote a property in the microscale.

Response databases for material design are constructed following a similar procedure as data-driven multiscale modelling. However, we include all three design categories ($\mathcal{G}_\mathrm{m}$, $\mathcal{M}_\mathrm{m}$ and $\mathcal{D}_\mathrm{M}$) in the feature space of a response database. Response databases used for material designing take the generalised form,

$$\mathcal{G}_\mathrm{m} = \{\mathcal{G}^i_\mathrm{m}\} \qquad \mathcal{G}^i_\mathrm{m} \in [g^i_\mathrm{min}, g^i_\mathrm{max}] \qquad \text{with} \qquad i = 1, 2, ...i_\mathcal{G} \tag{5.51a}$$

$$\mathcal{M}_\mathrm{m} = \{\mathcal{M}^i_\mathrm{m}\} \qquad \mathcal{M}^i_\mathrm{m} \in [m^i_\mathrm{min}, m^i_\mathrm{max}] \qquad \text{with} \qquad i = 1, 2, ...i_\mathcal{M} \tag{5.51b}$$

$$\mathcal{D}_\mathrm{M} = \{\mathcal{D}^i_\mathrm{M}\} \qquad \mathcal{D}^i_\mathrm{M} \in [d^i_\mathrm{min}, d^i_\mathrm{max}] \qquad \text{with} \qquad i = 1, 2, ...i_\mathcal{D} \tag{5.51c}$$

where $i_\mathcal{G}$, $i_\mathcal{M}$ and $i_\mathcal{D}$ refer to the number of design variables considered in each design category $\mathcal{G}_\mathrm{m}$, $\mathcal{M}_\mathrm{m}$ and $\mathcal{D}_\mathrm{M}$, respectively. Furthermore, we define boundaries for i^th design variable, given as $[(\bullet)^i_\mathrm{min}, (\bullet)^i_\mathrm{max}]$. The schematic of a generalised RVE shown in Figure 5.7 is used to intuitively discuss the data-driven material design. For simplicity, a RVE of a composite material consisting of a matrix (subscript `mat`) and reinforcements

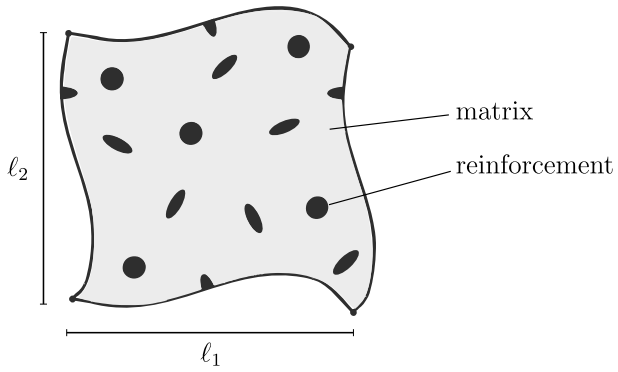


Figure 5.7: Generalised RVE of a composite material with microstructural particles (reinforcement) embedded in a matrix

(subscript `r`) is considered. Response database in Equation (5.51) is expanded for the composite RVE in Figure 5.7 as,

$$\mathcal{G}_{\mathrm{m}} = \begin{cases} \mathcal{G}^1_{\mathrm{m}} = \ell_1 & \ell_1 \in [\ell_1^{\min}, \ell_1^{\max}] \\ \mathcal{G}^2_{\mathrm{m}} = \ell_2 & \ell_2 \in [\ell_2^{\min}, \ell_2^{\max}] \end{cases} \tag{5.52a}$$

$$\mathcal{M}_{\mathrm{m}} = \begin{cases} \mathcal{M}^1_{\mathrm{m}} = \mathrm{E}_{\mathtt{mat}} & \mathrm{E}_{\mathtt{mat}} \in [\mathrm{E}^{\min}_{\mathtt{mat}}, \mathrm{E}^{\max}_{\mathtt{mat}}] \\ \mathcal{M}^2_{\mathrm{m}} = \nu_{\mathtt{mat}} & \nu_{\mathtt{mat}} \in [\nu^{\min}_{\mathtt{mat}}, \nu^{\max}_{\mathtt{mat}}] \\ \mathcal{M}^3_{\mathrm{m}} = \mathrm{E}_{\mathtt{r}} & \mathrm{E}_{\mathtt{r}} \in [\mathrm{E}^{\min}_{\mathtt{r}}, \mathrm{E}^{\max}_{\mathtt{r}}] \\ \mathcal{M}^4_{\mathrm{m}} = \nu_{\mathtt{r}} & \nu_{\mathtt{r}} \in [\nu^{\min}_{\mathtt{r}}, \nu^{\max}_{\mathtt{r}}] \\ \mathcal{M}^5_{\mathrm{m}} = \mathrm{N}_p & \mathrm{N}_p \in [\mathrm{N}^{\min}_p, \mathrm{N}^{\max}_p] \end{cases} \tag{5.52b}$$

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}^1_{\mathrm{M}} = E_{11} & E_{11} \in [E^{\min}_{11}, E^{\max}_{11}] \\ \mathcal{D}^2_{\mathrm{M}} = E_{22} & E_{22} \in [E^{\min}_{22}, E^{\max}_{22}] \\ \mathcal{D}^3_{\mathrm{M}} = E_{12} & E_{12} \in [E^{\min}_{12}, E^{\max}_{12}] \end{cases} \tag{5.52c}$$

where N_p is the number of reinforcement particles in the RVE.

5.4.3 Using the scikit-learn Python library

All the GPR tasks performed in this thesis use the open source machine learning implementation, scikit-learn library [175]. We outline the key steps used for this purpose in Algorithm 2. Algorithm 2 also highlights key functions of the class *sklearn.gaussian_process.GaussianProcessRegressor()* used in the scikit-learn library.

Algorithm 2 GPR model training and testing

// construct the response database

preprocessor : define parametric geometry of the RVE, bounds of the features, number of data points, sampling technique

for `i` $\leq$ `data points` **do**

  define BVP of the RVE

  solve constrained RVE problem

  extract inputs and output(s)

  record data to construct the response database in the form of Equation (5.52)

end for

// prerequisite diagnosis

check : normalised inputs, outlier identification

shuffle : to randomise the response database using *numpy.random.shuffle()*

// start GPR model training using scikit-learn

define covariance function, hyperparameter bounds, optimisation scheme using *kernel* $= \sigma_f^2 \times RBF(length\_scale,\ length\_scale\_bounds=())$

// 5-fold cross-validation

for `i` $\leq$ `5` **do**

  get i^{th} one-fifth of the database

  train GPR model for the remaining four-fifths using *GaussianProcessRegressor(kernel, random_state, normalize_y = True, n_restarts_optimizer = 10).fit(Training inputs, Training output)*

  record optimal hyperparameters, MLML, training time, training MSE and R^2

  test GPR model using the i^{th} one-fifth using *predict(Testing data, return_std)*

  record testing MSE and R^2 using *mean_squared_error* and *r2_score*

  check over-fitting, under-fitting, non-convergence

  tune hyperparameter bounds

end for

// save trained GPR model

save GPR model to as a abc.joblib for later use

The well-trained GPR trained model is subsequently used for the data-driven computational homogenisation and design of textile membranes. The key algorithmic steps are outlined in the Algorithm 3. Algorithm 3 also provides the algorithmic implementation of the nonlinear orthotropic thin-shell solver discussed in detailed under Chapter 4.

Algorithm 3 Data-driven computational homogenisation

// run the analysis preprocessor
preprocessor : definition of the geometry and external loading
// define the reference configuration of the structure
initialise : $\boldsymbol{f}_{\text{ext}} = \boldsymbol{0},\ \boldsymbol{x} = \boldsymbol{X},\ \boldsymbol{f}_{res} = \boldsymbol{0}$
// start step-loading the structure
for `i` $\leq$ `load steps` **do**
  find $\Delta \boldsymbol{f}_{\text{ext}}$
  set $\boldsymbol{f}_{\text{ext}} \leftarrow \boldsymbol{f}_{\text{ext}} + \Delta \boldsymbol{f}_{\text{ext}}$
  set $\boldsymbol{f}_{\text{res}} \leftarrow \boldsymbol{f}_{\text{res}} - \Delta \boldsymbol{f}_{\text{ext}}$
  // equilibrate the structure at every load step
  while $(||\boldsymbol{f}_{\text{res}}||/||\boldsymbol{f}_{\text{ext}}|| <$ `tolerance`$)$ **do**
    load trained GPR model as a abc.joblib
    // for every integration point
    input strain components to abc.joblib
    predict strain energy ψ
    compute $\boldsymbol{f}_{\text{int}}$ and $\mathbf{K}$
    update $\boldsymbol{f}_{\text{res}} \leftarrow \boldsymbol{f}_{\text{int}} - \boldsymbol{f}_{\text{ext}}$
    solve $\mathbf{K}\boldsymbol{u} = -\boldsymbol{f}_{\mathbf{res}}$
    update $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{u}_{\boldsymbol{u}}$
    compute vector triad ($\boldsymbol{a}_1,\ \boldsymbol{a}_2$ and $\boldsymbol{a}_3$) and their derivatives
    update strain components
  end while
end for

This chapter formed an imperative basis for this thesis by extensively discussing the data-driven frameworks for computational homogenisation and material design of textiles. The combined contributions from Chapter 3 on spatial rod-based microscale simulations, and Chapter 4 on thin-shell analysis, brought the ingredients together to solve a twoscale homogenisation problem in Chapter 5. Forthcoming chapter crowns this thesis by presenting two applications of data-driven homogenisation and material design using the proposed framework.

6 Applications

This chapter presents two selected applications of the proposed data-driven homogenisation and material design methods in Chapter 5. Even though one can think of many applications to use the suggested techniques, we limit our attention to two types of technical textiles, namely plain-woven and weft-knitted. Sections 6.1 and 6.2 are dedicated to present the data-driven results of plain-woven and weft-knitted textiles, respectively.

Both applications in this chapter follow a similar structure. First, geometric modelling of a RVE is detailed using illustrations and centreline equations. Next, microscale boundary value problems are defined for uniaxial, biaxial and pure shear deformations of RVEs using periodic boundary conditions. Enforcement of boundary conditions is illustrated with the respective deformed RVEs for visual verification. Visual verification is followed by a quantitative verification of the numerical model in reference to similar problems solved using distinct methods. Next, a yarn-level phenomenological model is simulated to qualitatively discuss and intuitively justify the deformations observed in the phenomenological model. A parametric study is carried out to investigate the sensitivity of the problem parameters on the homogenised response. Lastly, data-driven macroscale analyses are performed as outlined in Section 5.3 and material design aspects are further elaborated.

A special note is warranted on the microscale numerical model verifications. Our nonlinear isogeometric spatial rod model together with contact schemes (see Chapter 3) is used in simulating plain-woven and weft-knitted RVEs. However, in literature, authors use different methods to analyse rod structures that have distinct discretisations, degrees of freedom and other geometric considerations. Moreover, some authors use different boundary conditions (fixed displacements, periodic) to constrain RVEs in the computational homogenisation process [1, 18, 20]. Nevertheless, verifying our results with that of the other authors' works, not only it proves the accuracy of our implementations but also provides an independent certification to our methods.

6.1 Plain-woven textiles

Woven textiles, as the name implies, can be defined as any textile formed by weaving. Woven fabrics are often created on a loom [187], and made of many threads (yarns) woven in warp and weft directions (see Figures 6.1 and 6.2). Precisely, a woven textile is made by interlacing two or more yarns at given angles to one another. Mostly, we come across perpendicular interlacing of yarns as shown in Figure 6.1. However, there are non-perpendicular interlacing variants of woven textiles [58,188] for which one would observe the coupling between axial (tension predominant) and shear responses.

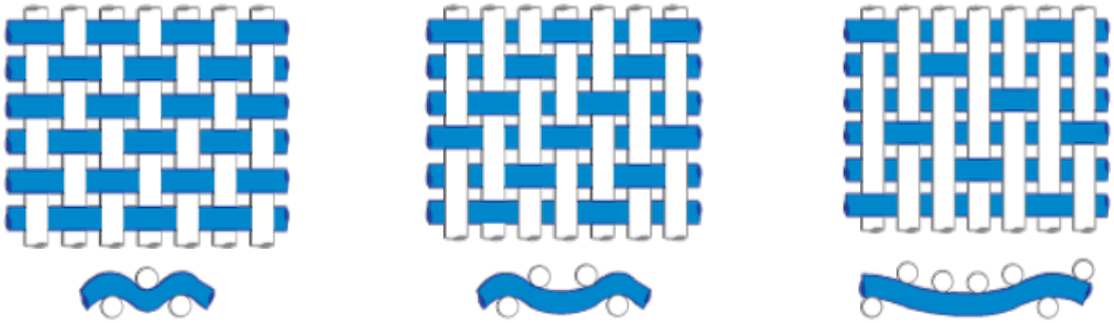


Figure 6.1: Types of weaving techniques of woven textiles [189]

In this thesis, we only consider the plain-woven textiles. However, it is straightforward to apply the proposed analysis tools to homogenise and design other variants of woven textiles. As common to any multiscale strategy, the first task is to identify an element to represent the macroscale continuum. Respecting the prerequisites highlighted in Section 5.1.2, we select a representative volume element as shown in Figure 6.2. Next task is to accurately model the identified analysis-suitable RVE using one of the geometric modelling techniques introduced in Chapter 2. In this section, we use smooth manifold-based basis functions introduced in Section 2.2 for the geometric modelling and analysis of plain-woven RVEs.

6.1.1 RVE Geometry modelling

For notational convenience, we denote any property of warp and weft/fill yarns (see Figure 6.2) using subscripts 1 and 2, respectively.

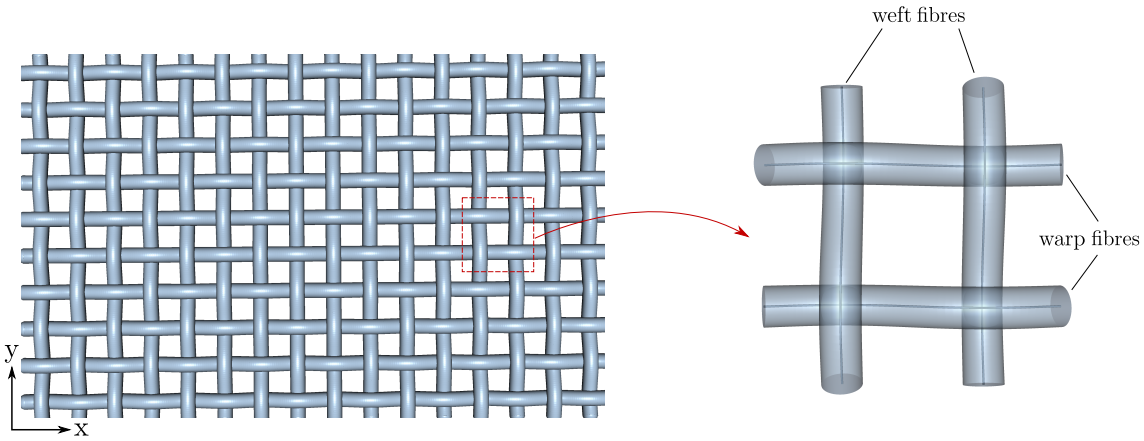


Figure 6.2: Selection of a multiscale analysis-suitable RVE of the plain-woven textile.

As shown in Figure 6.3, we first define the yarn centrelines and render the full yarns by adding a radius to these centrelines. Each yarn centreline is discretised using 20 finite deformable rod elements. Moreover, we define the global Cartesian coordinate system in a manner where x and y axes correspond to the in-plane and z axis denotes the out-of-plane (thickness) direction of the RVE. Additionally, we define the RVE in-plane dimensions as w_i where $i = 1, 2$ refer to warp and weft directions, respectively.

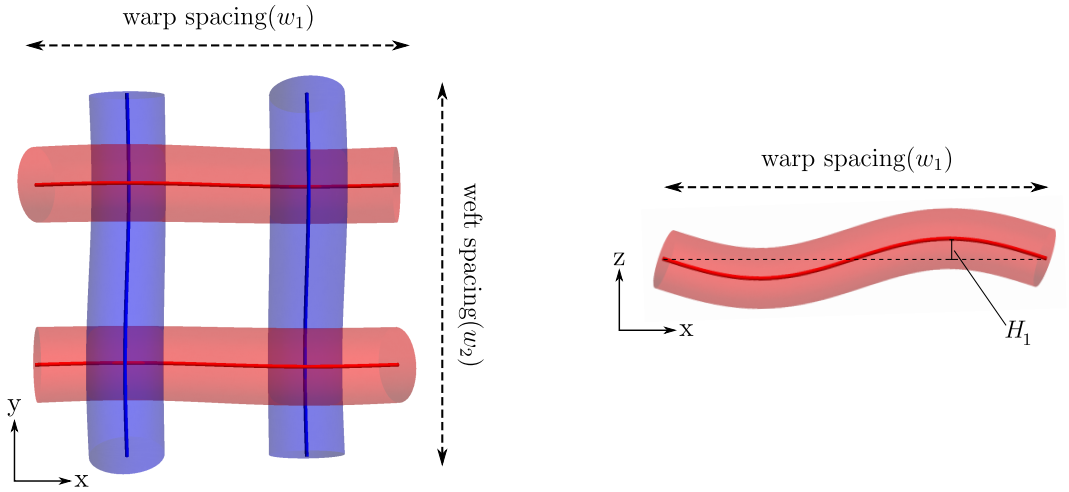


Figure 6.3: Different views of a plain-woven textile RVE. Rod centerlines are shown to define the sinusoidal geometry of a yarn.

For model verification and homogenised RVE result justification, we use the yarn models used by Nadler et al. [18] and Warren [103]. Hence, we define the geometry of a warp

yarn as a smooth sinusoidal curve given by,

$$z = A \sin(Bx) \tag{6.1}$$

where A and B are constants with former scaling the amplitude and latter scaling the wavelength of the sinusoid. For yarns in weft direction, we simply replace the independent variable x in Equation (6.1) by y (see Figure 6.3). Furthermore, constant A is taken as H_i and B is defined as $B = 2\pi/w_i$ for warp ($i = 1$) and weft ($i = 2$) yarns, respectively. According to the geometry definition by Nadler et al. [18], amplitude H_i is taken to be Cw_i/π, where C is a constant. Table 6.1 summarises the material and geometric properties used herein as well as in the RVE simulations of Nadler et al. [18]. In Table 6.1, yarn density is denoted by ρ_i in respective direction i.

Table 6.1: Warp and weft yarn geometric and material parameters

Yarn	Axial stiffness (N)	Bending stiffness (N/mm^2)	w_i(mm)	H_i(mm)	ρ_i(yarns/mm)
warp ($i = 1$)	1156	0.027975	0.95292	0.31684	2.10
weft ($i = 2$)	1165	0.045668	0.94248	0.13345	2.12

6.1.2 Microscale boundary value problems

We employ boundary value problems on plain-woven RVEs to simulate distinct deformation modes as shown in Figure 6.4. Constraint enforcement at the boundary nodes of each yarn is achieved using periodic boundary conditions as previously discussed in Section 5.1.2. At every boundary node, there are four degrees of freedom to be constrained: three displacements and one axial rotation. Thus, the four characters at each boundary node (see Figure 6.4) denote spatial displacements (u_x, u_y, u_z) and axial rotation (ϑ), respectively. Moreover, we use shortened notations $p, f, 0$ to refer to *p*eriodic, *f*ree and *zero*-displacement constraints. Any prescribed displacement is indicated by a component in the displacement vector $\boldsymbol{u}$. For instance, applied displacement in the x-direction is given by u_x (see Figure 6.4).

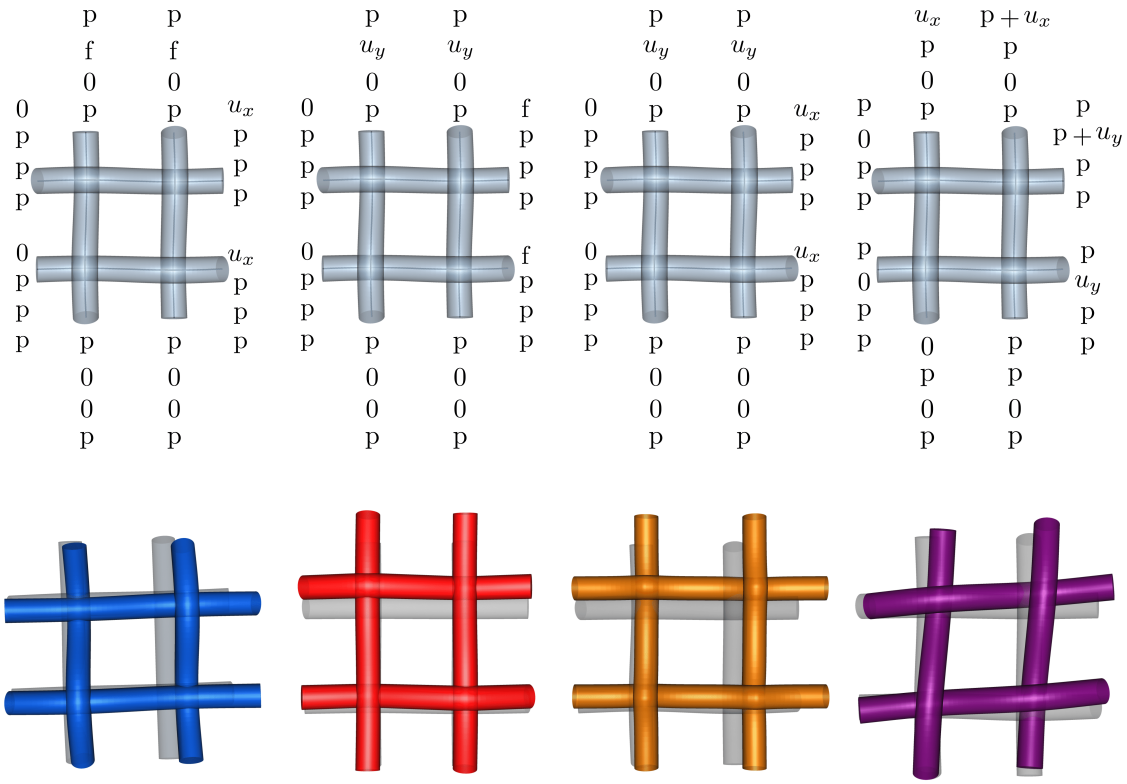


Figure 6.4: Periodic boundary condition enforcement (top row) and respective deformed RVEs (bottom row). At every boundary node, four characters denote the constraint enforcement to each degree of freedom at the respective boundary node. u_i refers to the applied displacements in the i direction. Moreover, $p, f, 0$ refer to *p*eriodic, *f*ree and *zero*-displacement constraints respectively.

Verification of the numerical model

We use the results obtained by Nadler et al. [18] to verify the solutions from microscale boundary value problems defined in Figure 6.4. Nadler et al. use displacement boundary conditions to define the constraints applied to the boundary nodes of the RVEs. In other words, they set the microfluctuations at the boundary to zero, thus resembling Taylor (or Voigt) conditions [38]. Herein, however, we use periodic boundary conditions, as shown in Figure 6.4 to enforce boundary conditions of the RVEs. Motivation to use periodic boundary conditions comes directly from their merits on faster convergence and accurate modelling [38].

Nadler et al. [18] produces *three* main results in their homogenised uniaxial deformation simulations. Therefore, we first simulate homogenised uniaxial tension and compression

tests using the properties in Table 6.1 to verify the mechanical response of RVEs. Homogenised first Piola-Kirchhoff stress tensor ($\boldsymbol{P}$) is extracted using Equation (5.11).

Axial stress resultant (P_{11}): Figure 6.5 depicts the axial response of the plain-woven RVE subjected to a uniaxial deformation (see the left of Figure 6.4). We have considered a 5% compressive strain region to show that, due to yarn contact separation, there will be small compressive stresses developed in the warp yarns due to further crimping. However, the same axial response is different in the tensile strain regime. We see a highly nonlinear response in the strain range $0 - 4\%$ due to decrimping of the warp yarns. Contacts between warp and weft yarns start to develop during this stage and grow rapidly during the decrimp of warp yarns. When applied strain reaches 5%, warp yarns are nearly straightened and therefore the homogenised response becomes less nonlinear. At this stage, yarn contacts are well established (see Figure 6.7) and the homogenised response converges towards a linear relationship compared to the high nonlinearity observed during decrimping.

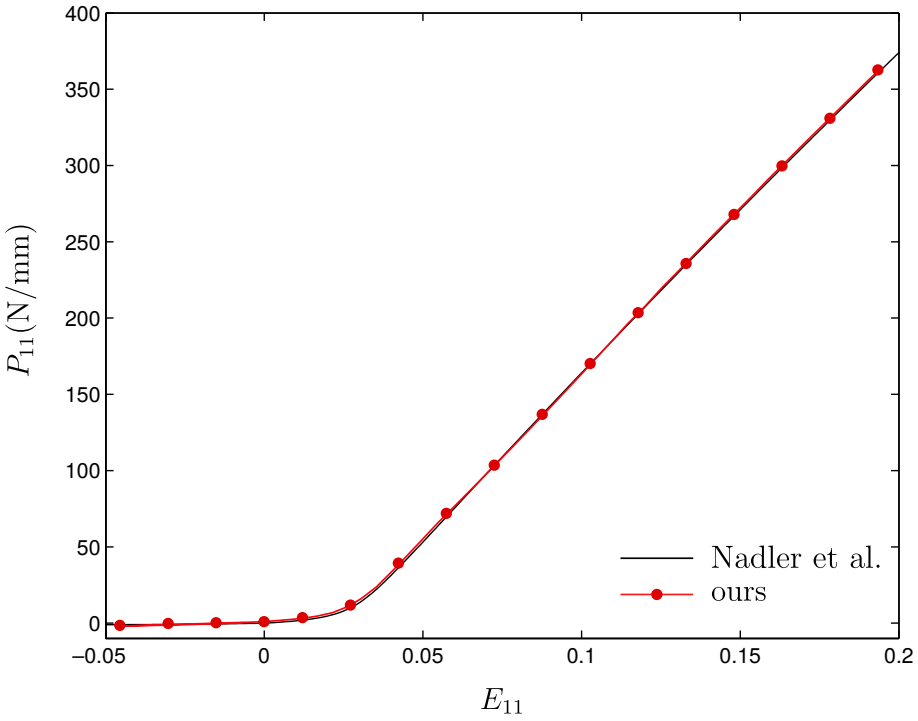


Figure 6.5: Stress resultant P_{11} vs E_{11} for homogeneous uniaxial deformation.

Poisson's effects: Poisson's ratio is computed as the gradient of the graph of transverse strain versus axial strain. For this purpose, we plot the variation in the transverse strain

E_{22} due to the applied strains E_{11} (see Figure 6.6). The compressive strain region does not contribute to the Poisson's ratio due to the separation of yarns, as discussed earlier. As there is no contact force transferred from warp to weft yarns, weft yarns do not deform and thus would not give rise to a transverse strain. Interestingly, a significant variation of the Poisson's ratio is observed in the decrimp stage $(0 - 4\%)$ of warp yarns (see Figure 6.6) mainly due to the contact formation between warp and weft yarns. RVE is observed to be stiffer in transverse deformations when we pay attention to the axial strains higher than 5%. Also, the gradient of the curve in Figure 6.6 gradually decreases in its magnitude as expected, due to the dominant deformations of nearly straightened warp yarns.

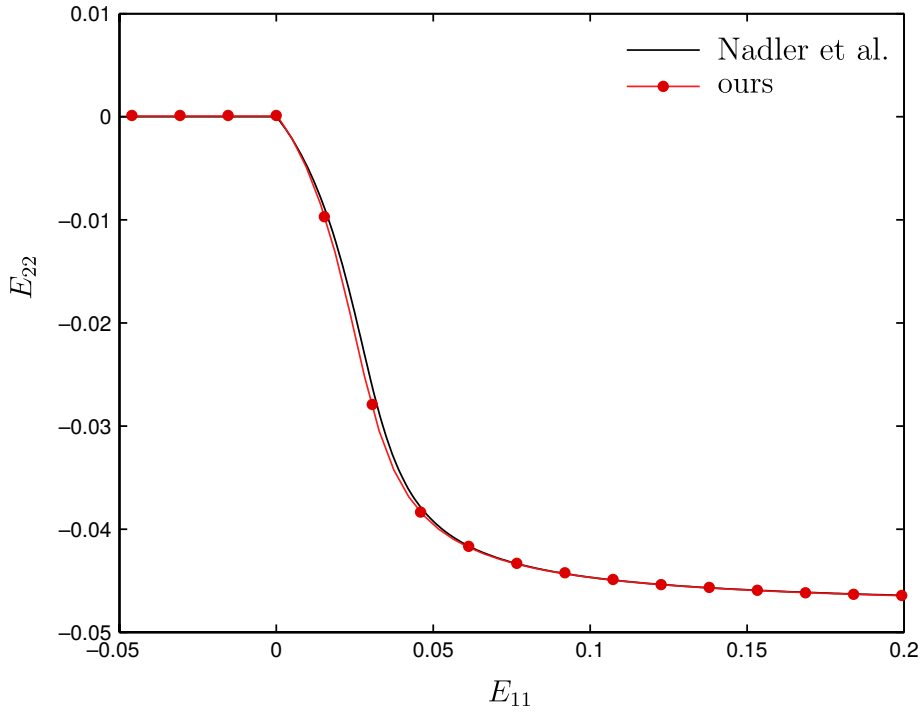


Figure 6.6: E_{22} vs E_{11} for homogeneous uniaxial deformation.

Internal stress developed due to contacts: Internal stress can also be identified as a pressure force that acts in between two yarns at the contact location. One advantage of using the Lagrange multiplier method to enforce contact constraints is that the resulting Lagrange multiplier value corresponds to the contact force at the respective contact location. Denoting these four contact forces as λ_i $(i = 1, 2, 3, 4)$ we derive the internal

stress as,

$$P_0 = \rho_1 \rho_2 \frac{\sum_{i=1}^{4} |\lambda_i|}{4} \tag{6.2}$$

where ρ_1 and ρ_2 denote the yarn densities in warp and weft directions, respectively (see Table 6.1). We use the magnitude values of λ_i in Equation (6.2) to consider only one force from the equal and opposite contact forces. Observations closely follow from both axial stress resultants and Poisson's effects. Compressive strains give no rise to internal stresses due to contact separation. There is a rapid increase of the internal stress during the contact formation (strains $0 - 4\%$) and gradually converges at latter stages of the decrimping process of warp yarns.

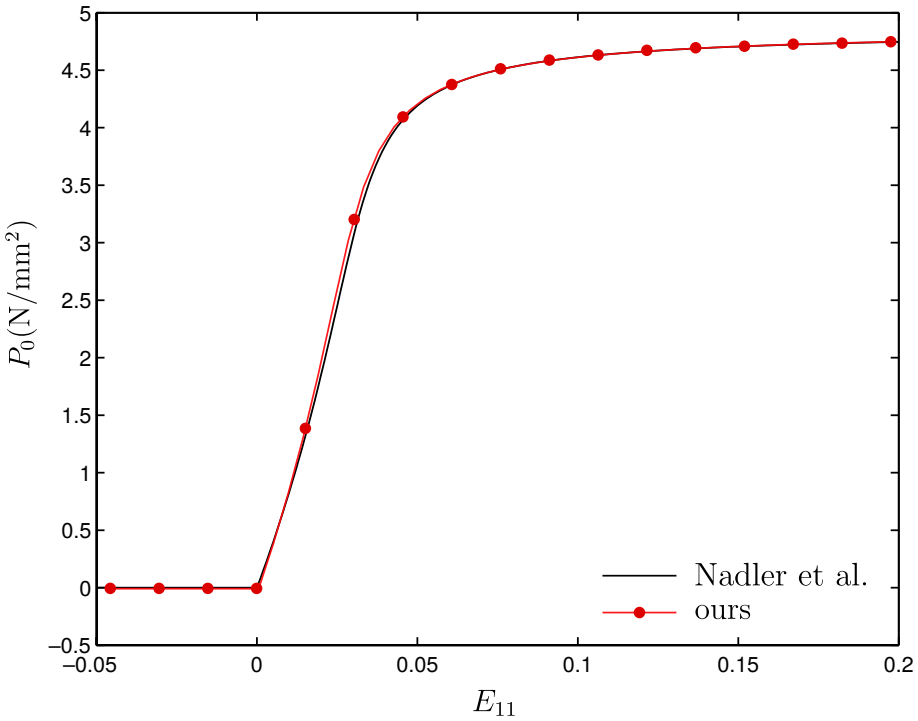


Figure 6.7: Internal stress P_0 vs E_{11} for homogeneous uniaxial deformation.

Shear stress resultant (P_{12}): Zero shear stiffness assumption made by Nadler et al. [18] in their work, is mainly twofold. First, shear stiffness observed in real textiles is very low in comparison to axial stiffnesses. Secondly, beam models used by Nadler et al. [18] and Warren [103] are deformable only in 2D space, thus not allowing to simulate shear deformations in plain-woven RVEs. However, our spatial rod implementation detailed in Section 3.2 allows to enforce pure shear conditions in plain-woven RVEs (see Figure 6.4). Figure 6.8 illustrates the homogenised RVE response under pure shear deformations. As

expected, we observe not only a very low shear stiffness but also a nearly linear shear response of the plain-woven RVE.

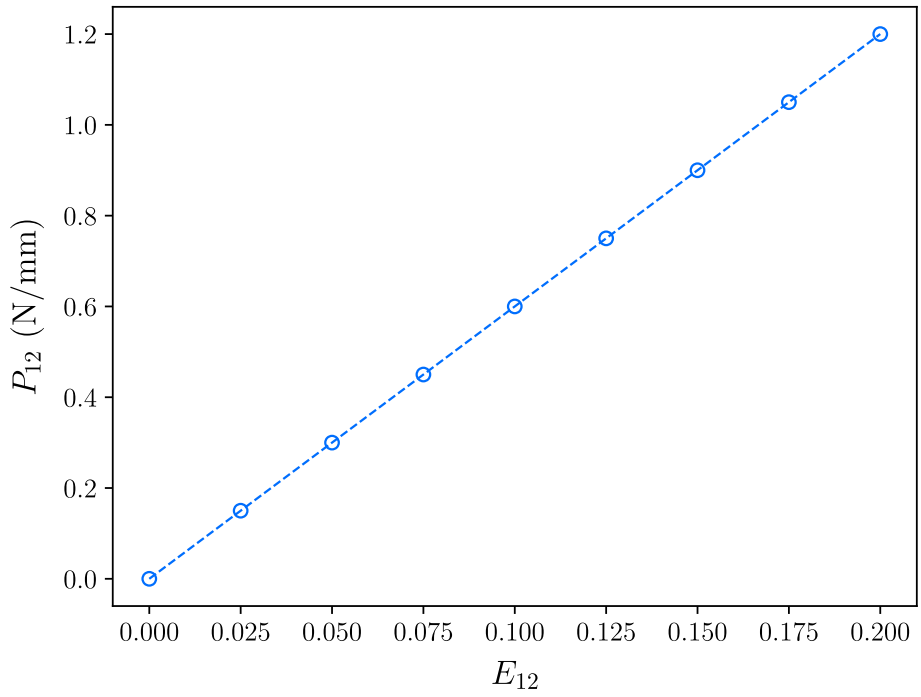


Figure 6.8: Stress resultant P_{12} vs E_{12} for homogeneous pure shear deformation.

Yarn-level simulation of a plain-woven textile

Departing from the main focus on multiscale modelling of plain-woven fabrics, now we aim to develop a phenomenological model using the finite deformation rod model implemented in Section 3.2. The objective of this phenomenological model is to visually verify the deformation of a membrane made out of yarns defined according to Equation (6.1) and properties in Table 6.1. For brevity, we consider applying unidirectional forces to a 16×16 grid of yarns in a displacement controlled manner. Figure 6.9 visualises the deformation of the yarn-level model subject to a $E_{11} = 10\%$ strain level.

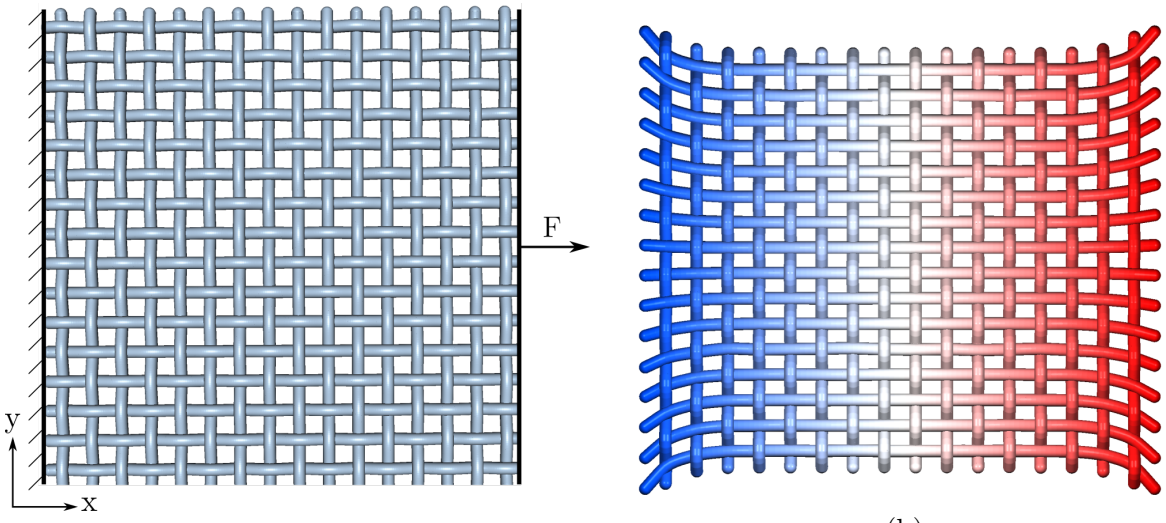


Figure 6.9: Yarn-level pull test of a plain-woven textile. (a) Reference configuration of the 16×16 grid of yarns (b) Deformed configuration due to the applied load in the x-direction. Colour distribution refers to the x-direction displacements of the yarns.

We verify the phenomenological model by closely observing the followings.

- Poisson's effects: The transverse deformation (in the y-direction) of the yarn-level model due to loading in axial (x) direction, clearly proves the existence of Poisson's effects. Reader is encouraged to see Figure 6.9 in combination with Figure 6.6.
- Decrimping of warp yarns: Due to externally applied loading in x (parallel to warp) direction, forces are predominantly transferred by the warp yarns. Thus, warp yarns decrimp until they are fully straight. This warp yarn decrimping process is clearly observed in Figure 6.5.
- Yarn contacts: We assume full friction between the warp and weft yarns similar to the works of Nadler et al. [18]. Hence the deformation in Figure 6.9 verifies the static frictional contacts between two yarns due to the observed translations of yarn contact points during deformation.
- Crimping of weft yarns: Weft yarns go through additional crimping due to Poisson's effects. This can be easily understood by comparing weft yarns at the fixed end to the ones in the middle.

We do not focus more on the yarn-level models due to higher computational cost associated with the numerical simulations [2, 17, 190]. However, compared to existing commercial finite element software, our implementations can easily vary warp and weft yarns to simulate full-scale yarn-level textiles.

Sensitivity analysis

A set of sensitivity analyses is carried out to investigate the dependency of the homogenised response on material and geometric property parameters. This sensitivity analysis is qualitatively a crucial step in building the response database discussed in Section 5.3.1. The sensitivity of each parameter is essential to investigate the parameter dependencies, redundant parameters and uncertainties of each parameter on the homogenised response of RVEs. Uncertainty quantification is one of the most sought after topics since the inception of data-driven techniques [191]. Even though we do not intend to go further into this topic, it is important to leave a remark here for completeness. For instance, one can think of a probability distribution for warp and weft yarn Young's moduli and formulate a probabilistic homogenised response. Thus, we can generate a mean and confidence interval for the homogenised response of the RVE and include it in the scale transitions to introduce a data-driven probabilistic multiscale strategy.

For brevity, only homogenised uniaxial tensile response is considered herein for the sensitivity analysis of plain-woven RVEs. As common to sensitivity analyses, we change one variable and hold others constant, unless there are any variable dependencies. A maximum 10% strain level is used in the following analyses following periodic boundary conditions to enforce the constraints.

RVE size

First, we study the sensitivity and convergence of the homogenised response on the chosen RVE size. For a general RVE, the accuracy of the homogenised response depends heavily on the selected RVE size. Geers et al. [38] and Huet [192] studied this by comparing the convergence rates of different boundary enforcement methods versus the RVE size. But as shown in Figure 6.10, all curves overlap and hence it is evident that RVE size has negligible influence on the homogenised response of the plain-woven RVE. It is mainly due to the exact definition and simplicity of the RVE compared to more complex RVEs considered by Geers et al. [38] and Huet [192] in their works. This

also proves that the selected RVE in Figure 6.2 is a good approximation to perform computational homogenisation [193].

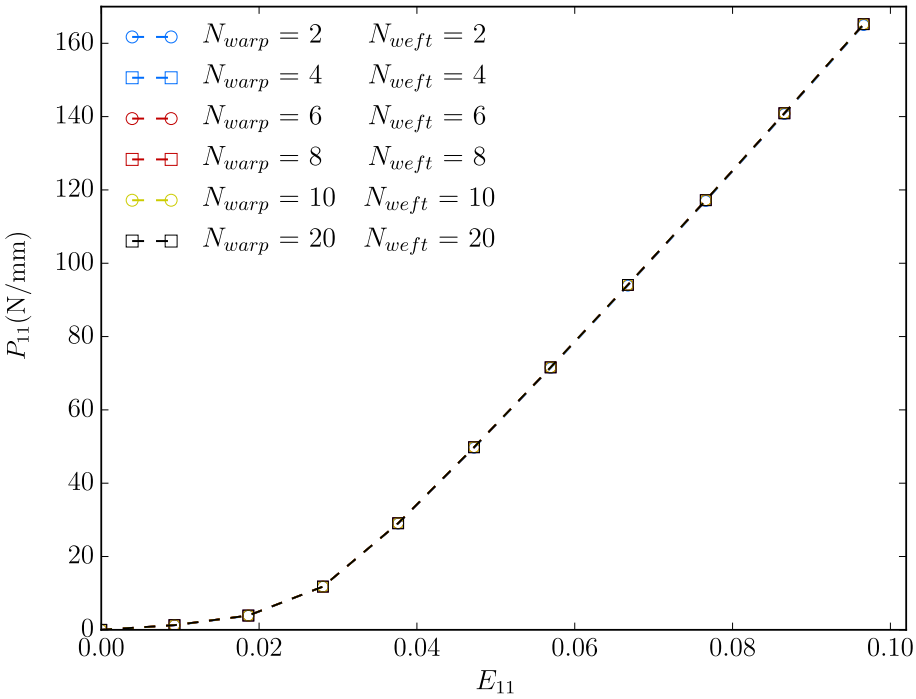


Figure 6.10: Convergence of the homogenised response for different sizes of plain-woven RVEs. Here, N_{warp} and N_{weft} refer to the number of yarns in warp and weft directions of the RVEs, respectively.

Young's moduli of warp and weft yarns

We study the homogenised response for different Young's moduli values assigned to yarns in both directions. First, we use yarns made from the same material ($E_{warp} = E_{weft}$) and investigate how the homogenised response would vary. As shown in Figure 6.11a, we can see a linear relationship between the homogenised P_{11} and the yarn Young's moduli. This is expected if one recalls that the system stiffness matrix is linearly scaled by the Young's moduli of the yarns. Equations (3.92) - (3.95) evidently prove this.

Next, we vary the Young's moduli of warp and weft yarns. In other words, we use one material to manufacture warp yarns and another to manufacture weft yarns. There are a few key observations that are interesting to discuss from Figure 6.11b.

– Young's modulus of weft yarns is mechanically effective only during the decrimping process of warp yarns. When applied strain is in the range $0 - 4\%$, we can see the

induced stress resultant P_{11} is increased with the Young's modulus of weft yarns.

– At higher strains ($> 5\%$), nearly straightened warp yarns are predominant in carrying the axial forces. Hence the effective contribution to the homogenised response from weft yarns is negligible.

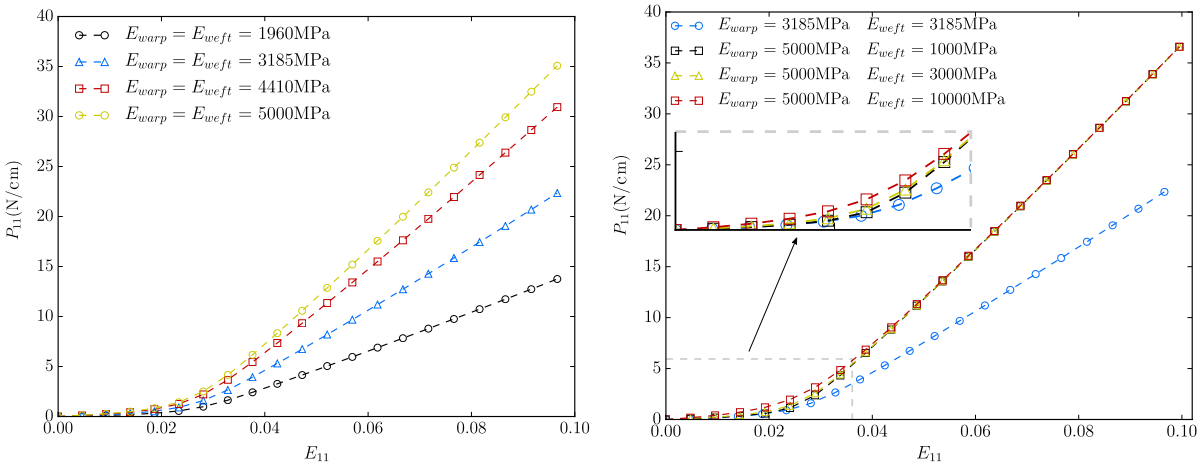


Figure 6.11: Sensitivity analysis of the Young's moduli of warp and weft yarns. (a) Same yarns in both directions. (b) Different yarns in warp and weft directions.

Radius of warp and weft yarns

Next, we vary the yarn cross-sections. We use yarn radii to define the cross-section details of warp and weft yarns uniquely. For simplicity and respecting the spatial distribution of yarns, we use the same value for both yarn radii. As expected, the homogenised response is stiffened, upon the increase of the yarn radii. This is intuitively understood because stress resultants are proportional to RVE stiffness.

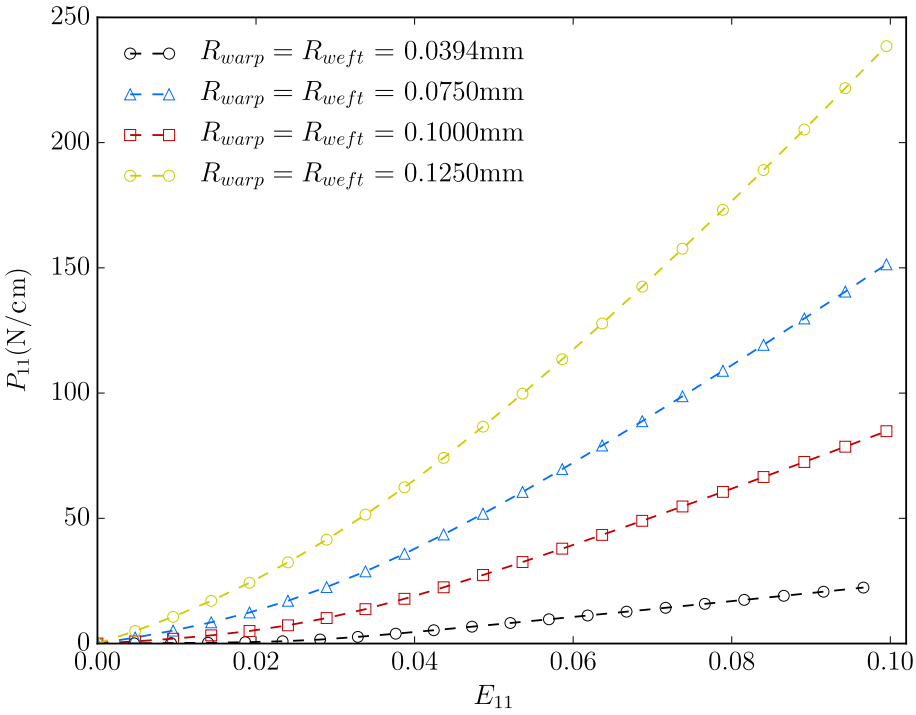


Figure 6.12: Sensitivity analysis of yarn radii of warp and weft yarns on the homogenised material response.

Next, we consider the sensitivity of the parameters that define the yarn geometry. According to Equation (6.1) we have two parameters: amplitude (H) and wavelength (w) that define the yarn geometry (see Figure 6.3). The primary motive of considering these two parameters is to understand the influence of initial crimp, decrimp phase and RVE in-plane dimensions on the overall homogenised response. Moreover, we investigate on lower and upper bounds of these parameters to analyse the change in mechanical response to get a better understanding of the influence factors mentioned earlier. Thus, some values used for H and w below, may not seem realistic in a practical sense.

Amplitude (H) of warp and weft yarns

We vary the amplitudes of both warp and weft yarns systematically to understand the influence of the initial crimp on the homogenised response.

– By setting $H_{\text{warp}} = 0.0001$mm, we nearly straighten the warp yarns. As expected, we do not observe any decrimping phase, and the warp yarns predominantly govern the homogenised response.

– By setting $H_{\text{warp}} = 0.50\text{mm}$, we introduce additional crimp to warp yarns. Now the decrimping phase is stretched from $0 - 4\%$ (see Figure 6.5) to $0 - 8\%$ (see Figure 6.13).

– Homogenised response of other combinations of yarn amplitudes can be easily deduced following a similar rationale.

Irrespective of the warp yarn amplitude, amplitude of the weft yarns has a negligible influence on the homogenised response in warp direction. This is mainly due to the predominant tensile forces transferred by the yarns in the warp direction is least affected by the weft yarn amplitudes.

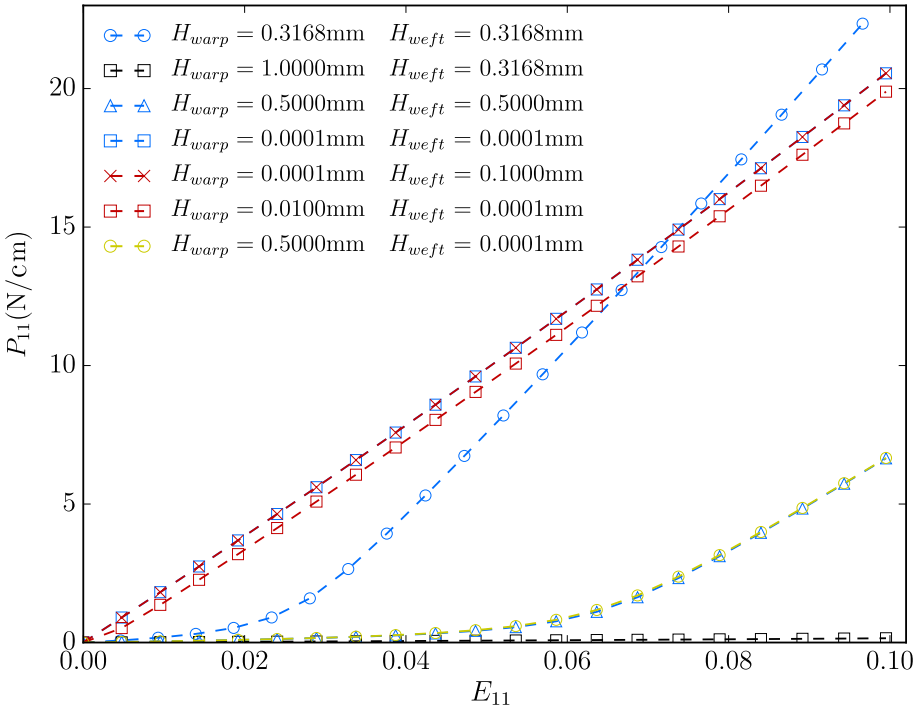


Figure 6.13: Sensitivity analysis of yarn amplitudes of warp and weft yarns on the homogenised material response.

Wavelength (w) of warp and weft yarns

For clarity, it is restated that warp, weft yarns and their spacings are defined according to Figures 6.2 and 6.3. For instance, uniaxial deformation in the warp direction is homogenised using the weft spacing (w_2). Hence, the following conclusions can be made using the wavelength sensitivity analysis in Figure 6.14.

– For constant values of w_{weft}, we can see only a small variation in the homogenised response. This small variation is due to the nonlinearity during the decrimp process of warp yarns.

– By lowering w_{weft}, we can observe a rapid increase in the homogenised stress resultants due to the increase in warp yarn density (ρ_1).

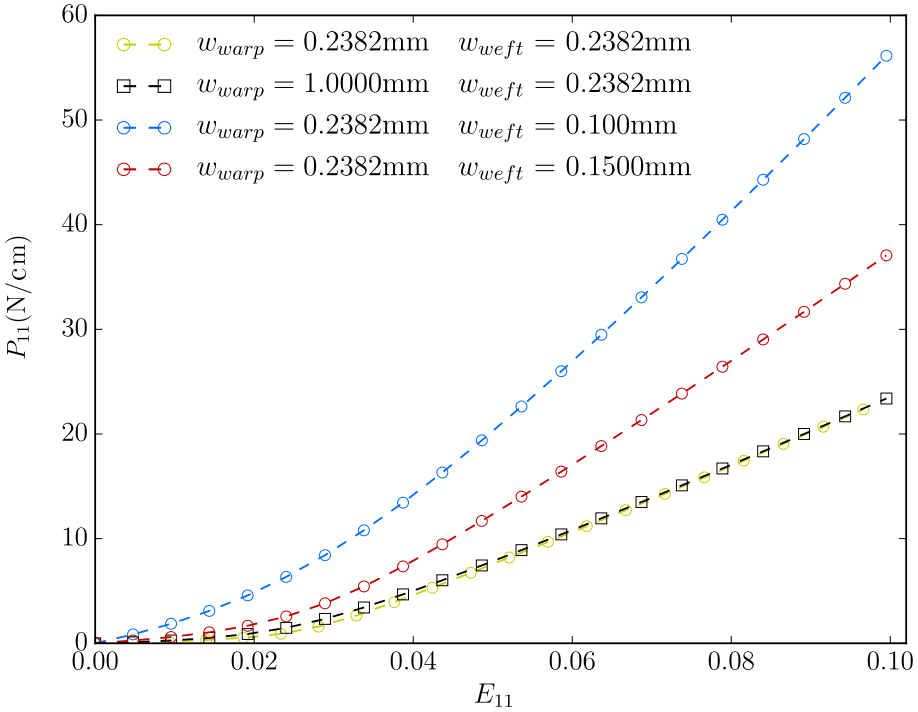


Figure 6.14: Sensitivity analysis of yarn wavelengths of warp and weft yarns on the homogenised material response.

6.1.3 Data-driven macroscale analysis

Data-driven multiscale modelling proposed in Section 5.3 is evaluated here for the computational homogenisation of plain-woven textiles. Provision of microscale model verifications, convergence study and sensitivity analyses in Section 6.1.2 constitute the applicability of machine learning for macroscale analysis.

It is not intended to repeat in-depth discussions here, because rigorous discussions on generalised formulations and implementations are provided in Section 5.2. Thus, in successive discussions, we elaborate on the results obtained for plain-woven textiles. We start building the response database using a uniform sampling technique

for strain bounds given in Equation (6.3). Response database is succinctly represented as $(E_{11}, E_{22}, E_{12}, \psi)$ where $\boldsymbol{E}$ is the Green-Lagrange strain tensor and ψ gives the area-averaged energy in mJ/mm^2.

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}^1_{\mathrm{M}} = E_{11} & E_{11} \in [-0.05, 0.22] \\ \mathcal{D}^2_{\mathrm{M}} = E_{22} & E_{22} \in [-0.05, 0.22] \\ \mathcal{D}^3_{\mathrm{M}} = E_{12} & E_{12} \in [0.0, 0.2] \end{cases} \tag{6.3}$$

Geometry ($\mathcal{G}_{\mathrm{m}}$) and material ($\mathcal{M}_{\mathrm{m}}$) parameters are held constant and take values given in Table 6.1.

Strain energy prediction and uncertainty

Using the scikit-learn Python library [175], we train a GPR model based on the response database constructed according to Equation (6.3). GPR model training, cross-validation and partial derivative formulae are detailed in Section 5.2. A small but complete subset of the full database is extracted to train a GPR model. As shown in Figure 6.15, we use 90 new data points from the full database to test the accuracy of the trained GPR model. Figure 6.15 also presents the corresponding 95% confidence interval of the predicted energy of the new data points. For this particular database, training and testing errors were recorded as 1.72×10^{-11} and 0.82×10^{-5}. Detailed analysis on these errors, optimised hyperparameters and maximum log marginal likelihood values are provided later in this chapter.

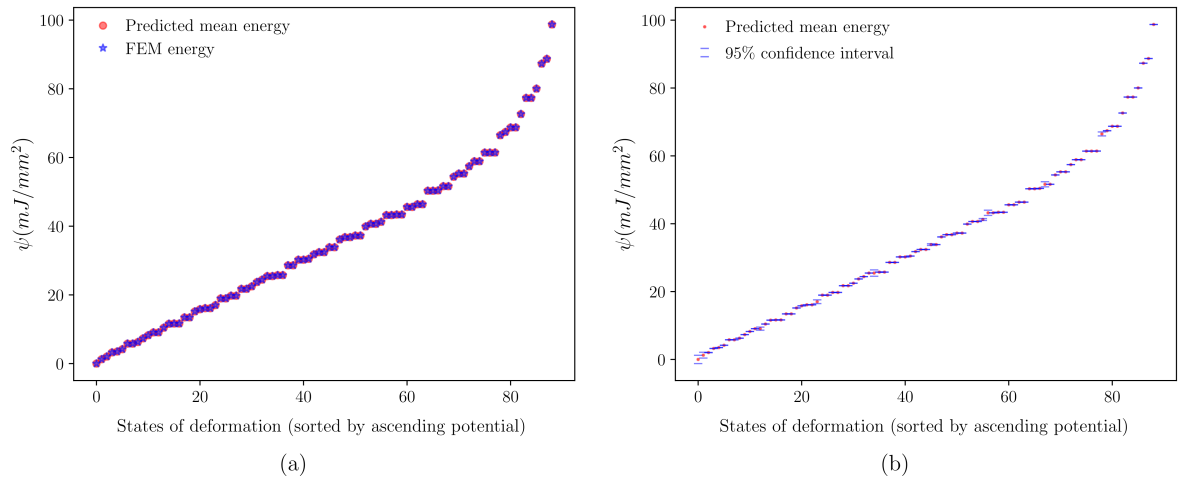
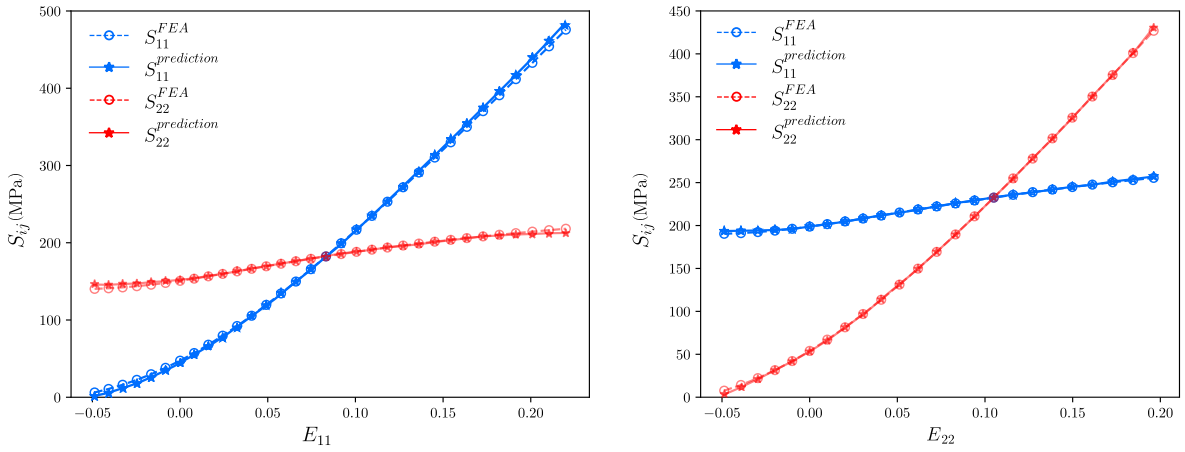


Figure 6.15: (a) Strain energy prediction for unseen data points sorted by ascending potential (b) 95% confidence interval of the predicted mean energy

Stress predictions

Accuracy test on predicted energy has to be followed by an accuracy test on the stress components. Using Equations (5.41) and (3.35), we evaluate the degree of accuracy of the predicted stresses to that of observed stresses in the database. Figure 6.16 provides two principal stress component comparisons by setting $E_{12} = 0.0$. For visualisation purposes, one strain component is held constant while the other is increased from 0.0 to 0.22. We observe nearly exact stress predictions from the trained GPR model. However, we notice slight deviations close to the bounds of strain components. Near boundary errors are common to many regression techniques, and hence additional care should be taken to avoid model overfitting while addressing such boundary errors [171]. In this thesis, boundary errors are minimised by refining the data points close to any feature bounds.



(a) (b)

Figure 6.16: Prediction and comparison of the second Piola-Kirchhoff stress components. (a) S_{11}, S_{22} for $E_{11} = [0.\%, 22\%]$ and $E_{22} = 8\%$. (b) S_{11}, S_{22} for $E_{22} = [0.\%, 22\%]$ and $E_{11} = 10\%$.

GPR model testing error quantification

It is of paramount importance to maintain algorithmic efficiency when training GPR models. Equation (5.32) indicates the requirement to invert the Gram matrix during the optimisation process which requires time $\mathcal{O}(n^3)$. Hence, in Figure 6.17, we present the variability of GPR model testing errors for energy and stress predictions. Equation (5.48) is used to compute the mean squared errors of energy predictions.

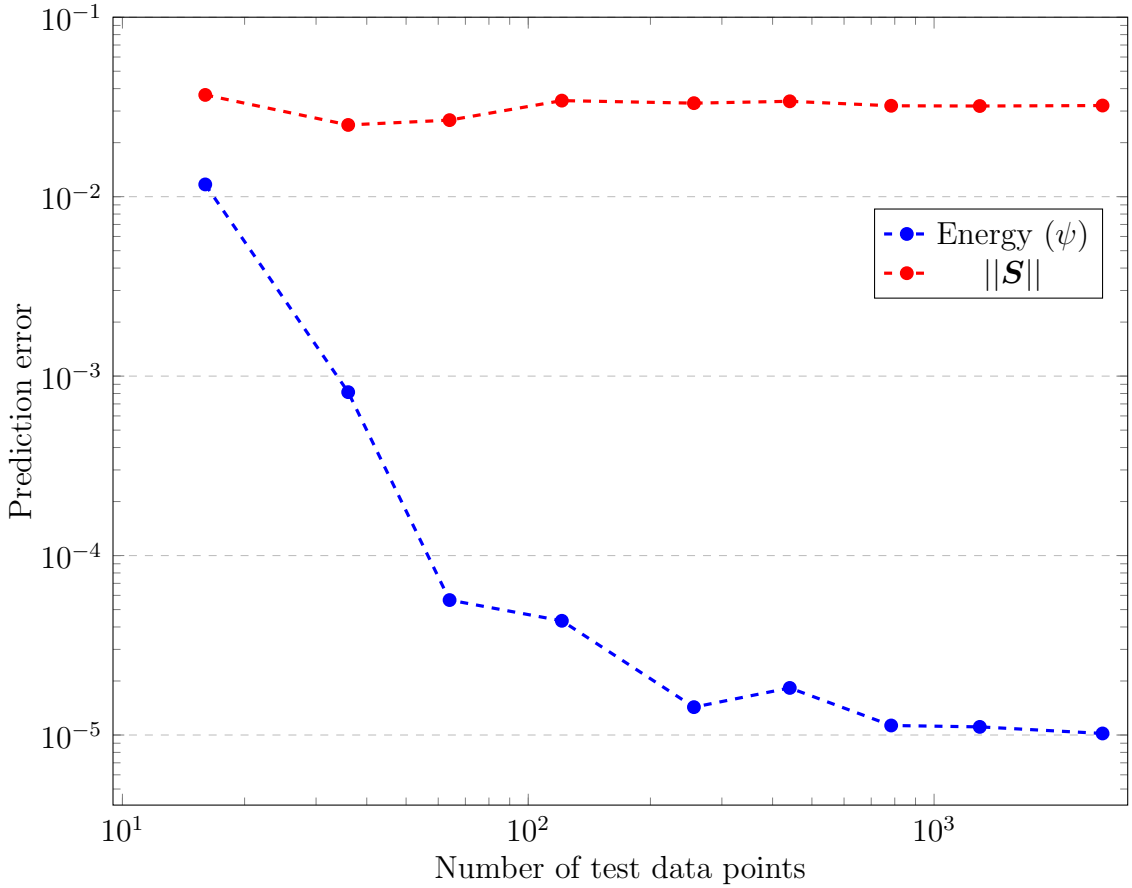


Figure 6.17: Errors of the predicted strain energy density and the norm of the vectorised second Piola-Kirchhoff stress tensor components

Errors in stress predictions are comparatively higher than the energy predictions. This observation is natural in data-driven analyses because the derivative of a data-fitted model is considered as an approximation of an approximation [61]. And this is also due to the fact that MSE is not a normalised measure of error. Hence the error convergence depends on the magnitude of the quantity being measured. Considering the error convergence and algorithmic efficiency, we use a response database with 2601 data points. Moreover, for every database, we optimise the log marginal likelihood (given in Equation (5.31)) with respect to hyperparameters and arrive at the optimum hyperparameters that yield the maximum log marginal likelihood. Table A.2 presents a concise tabulated results: error analyses R^2 and MSE; MLML and optimum hyperparameters (σ_f, ℓ).

Pull tests on data-driven plain-woven textiles

Numerically tested and verified GPR model is next applied to simulate the membrane response of pull tests along the two principal directions, namely warp-wise and weft-wise pull tests. Each membrane specimen is a square of side 10mm. These tests are designed in a way where one edge is held fixed, and the opposite edge is pulled in warp or weft directions by 1mm. The non-zero stress component distributions are shown in figures 6.18 and 6.19. It is important to emphasise on the symmetry of the stress distributions predicted by the Gaussian process model. For example, observe S_{11} in Figure 6.18 and S_{22} in Figure 6.19 for very close similarities in stress magnitudes. This is mainly due to the use of same (warp yarn) geometric and material properties for both warp and weft yarns in the microscale simulations (see Table 6.1). Symmetric results also prove the accuracy of the trained GPR model in two independent directions.

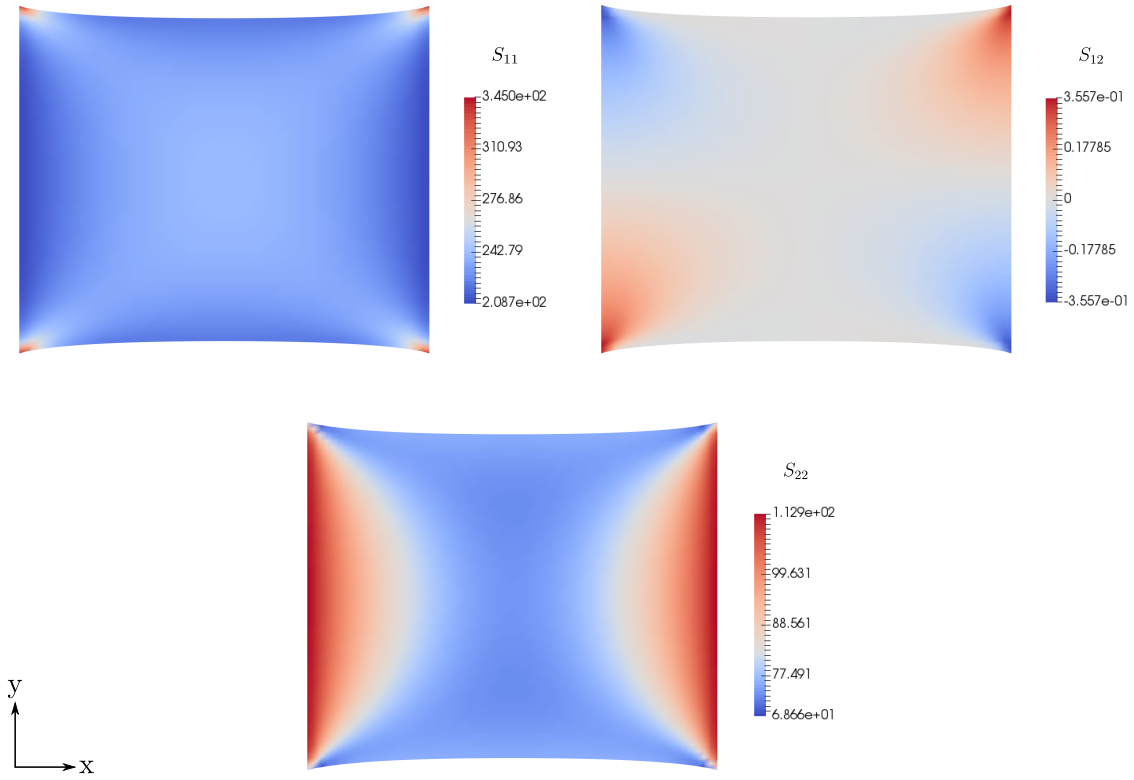


Figure 6.18: Stress distribution of a warp-wise pull test on a data-driven plain-woven membrane

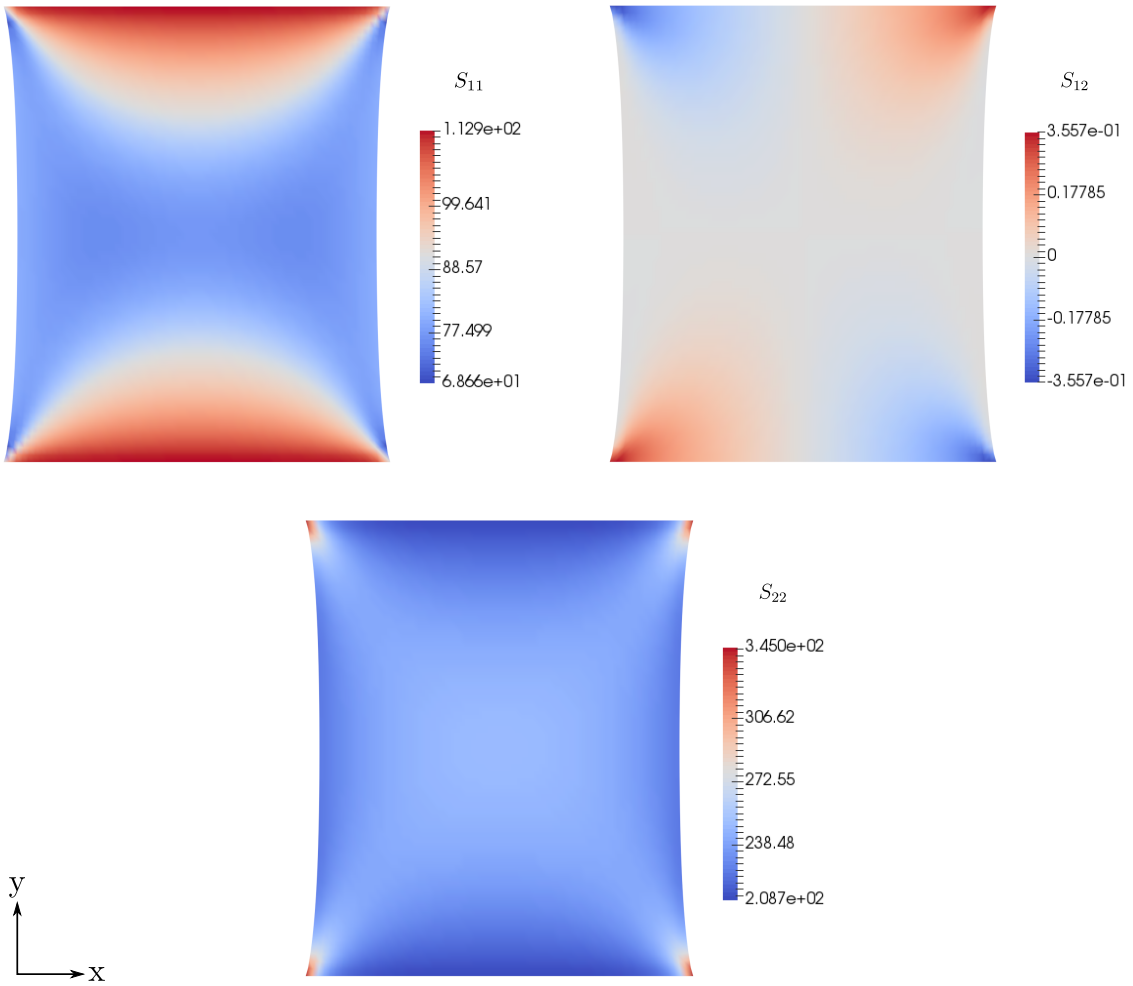


Figure 6.19: Stress distribution of a weft-wise pull test on a data-driven plain-woven membrane

Yarn-level versus data-driven homogenisation

After discussing the data-driven homogenisation of woven membranes, we present a quantitative comparison of the problem presented in Figure 6.9b. The yarn-level model comprises of 16 loops in the warp direction and 16 loops in the weft direction. Hence, we use an equal size membrane, 15.25mm long, 15.08mm wide and 0.64mm thick, for the homogenised macroscale analysis. Homogenised membrane model is analysed using a 16×16 structured mesh using the proposed Gaussian process homogenisation technique. For comparison purposes, yarn-level model in 6.20a is created by projecting it onto the xy plane using Delaunay mesh generation. We can observe a very good agreement between the two models.

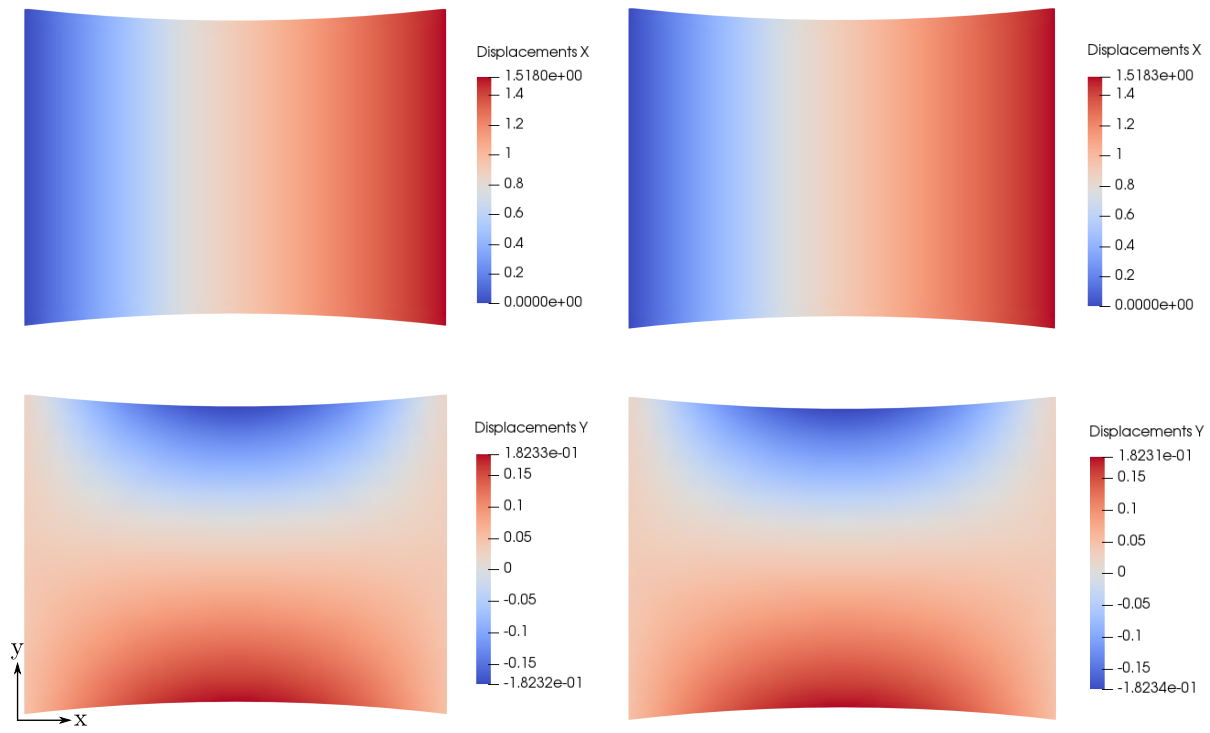


Figure 6.20: Quantitative comparison between yarn-level and homogenised models. Here, x and y displacement distributions of (a) yarn-level model and (b) homogenised woven membrane of a pull test along x-direction are illustrated.

6.1.4 Data-driven plain-woven textile design

The extension to data-driven plain-woven textile design is achieved by incorporating the design (material and geometric) parameters into the response databases. In this thesis, two examples of plain-woven textile designs have been presented. However, it is straightforward to consider desired combinations of $\mathcal{G}_{\mathrm{m}}$ and $\mathcal{M}_{\mathrm{m}}$ to construct the response databases. This essentially is an application-based decision.

Plain-woven textile design: Case 1

We first vary the Young's moduli of warp and weft yarns simultaneously and keep all geometric parameters constant, as given in Table 6.1. We define the response database entries succinctly as $(E_{11}, E_{22}, E_{12}, [\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{warp}}, [\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{weft}}, \psi)$ where $\hat{\mathrm{E}}$ takes the warp yarn

Young's modulus value used by Nadler et al. [18]. We specify the feature bounds to be used in constructing the response database as given in Equation (6.4).

$$\mathcal{M}_{\mathrm{m}} = \begin{cases} \mathcal{M}_{\mathrm{m}}^{1} = \left[\dfrac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{warp}} \quad \left[\dfrac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{warp}} \in [0.01, 2.0] \\ \mathcal{M}_{\mathrm{m}}^{2} = \left[\dfrac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{weft}} \quad \left[\dfrac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{weft}} \in [0.01, 2.0] \end{cases} \tag{6.4a}$$

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}_{\mathrm{M}}^{1} = E_{11} \quad E_{11} \in [-0.05, 0.22] \\ \mathcal{D}_{\mathrm{M}}^{2} = E_{22} \quad E_{22} \in [-0.05, 0.22] \\ \mathcal{D}_{\mathrm{M}}^{3} = E_{12} \quad E_{12} \in [0.0, 0.2] \end{cases} \tag{6.4b}$$

We take advantage of the linearity of Young's moduli to axial and bending stiffnesses of warp and weft yarns. Thus, by varying $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{warp}}$ and $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{weft}}$, we linearly vary the axial and bending stiffness given in Table 6.1.

GPR model testing error quantification

We quantify the prediction errors to measure the accuracy of the trained GPR model for plain-woven textile design. Following a similar approach considered in Section 6.1.3, we produce prediction errors for both energy and stress components, as shown in Figure 6.21. Variation and convergence of these errors follow a similar pattern to Figure 6.17. Thus, the increase in the number of features has no direct influence over the error rates. But one needs to be aware of the curse of dimensionality when increasing the dimensions of the features space [37]. Considering the error convergence and efficiency of model training, we use a response database with 6561 data points in subsequent analyses. For this specific database: recorded optimum hyperparameters were $\sigma_f^2 = 8.393^2$, $\ell = 0.0588$; maximum log marginal likelihood was 8336.441. Time taken for model training was recorded as 129 minutes and 48 seconds on an Intel Core i5-4590 CPU 3.30GHz $\times$ 4 processor.

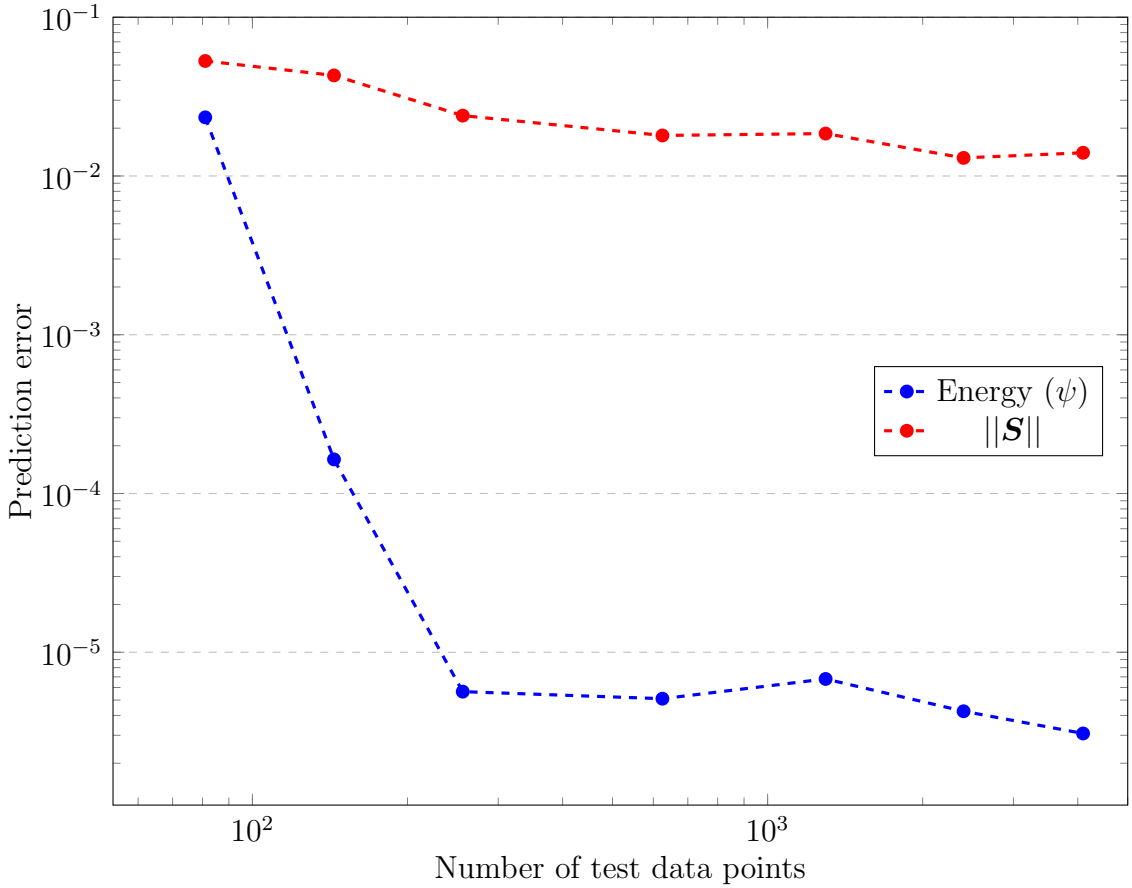
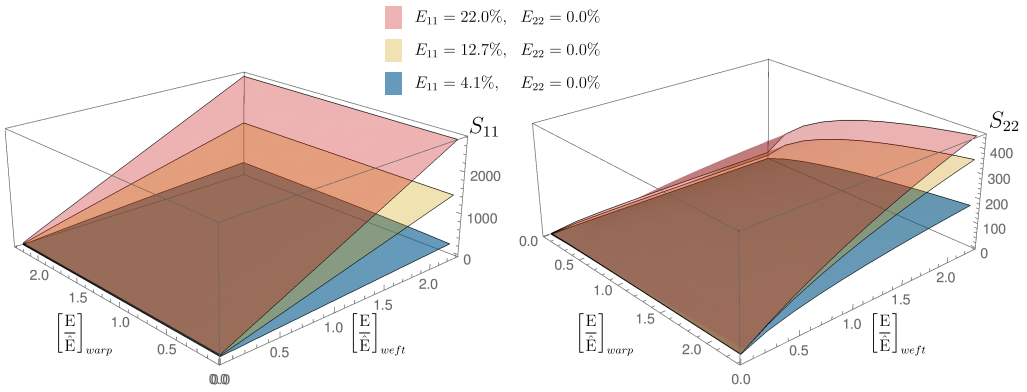


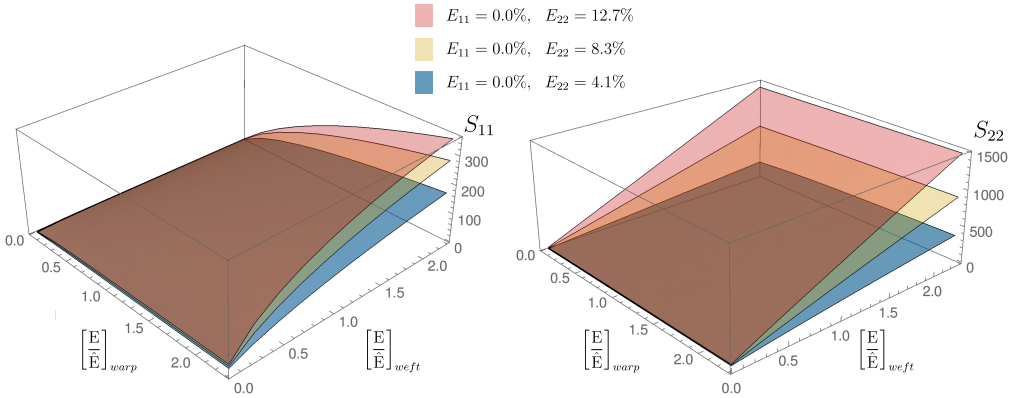
Figure 6.21: Design case 1: Error metrics of the predicted strain energy density and the norm of the vectorised second Piola-Kirchhoff stress tensor components

Now we are interested to see how the stress-strain variations would look like for a given combination of $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{warp}}$ and $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{weft}}$. We start with two strip biaxial tests, as shown in Figures 6.22a and 6.22b, where strain combinations $E_{11} : E_{22} = 1 : 0$ and $E_{11} : E_{22} = 0 : 1$ have been enforced respectively on the RVEs. Then we vary $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{warp}}$ and $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{weft}}$ on a refined grid consisting 28561 data points. This refined grid is defined in a way that only Young's moduli ratio bounds (0.01 and 2.0) are common to the response database used for model training and the refined grid. Thus, except for bounds, all other points act as unseen data points to the trained GPR model. Next, we present the stress-strain relationship of a biaxial test as shown in Figure 6.22c.

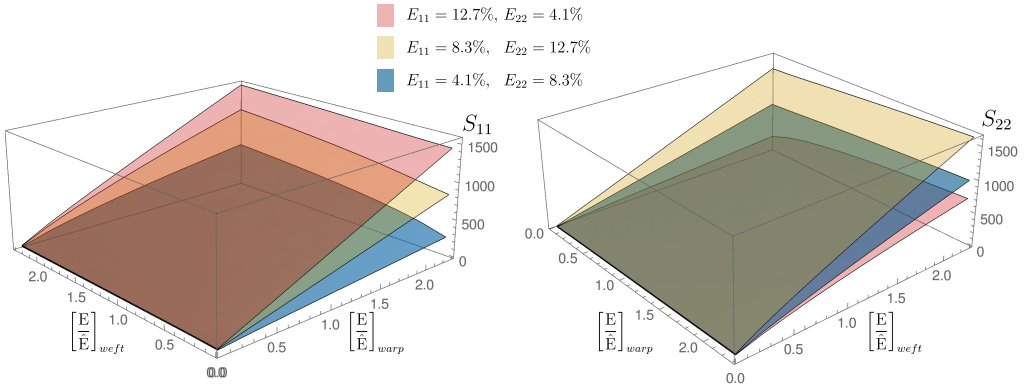
For illustration purpose, arbitrary strain values are used to represent different states of deformations. Moreover, all stress components in Figure 6.22 take the unit MPa.



(a)



(b)



(c)

Figure 6.22: Stress distributions (in MPa) for different combinations of Young's moduli of warp and weft yarns. Results of three biaxial tests are shown in each figure. (a) Strip biaxial test ($E_{11} : E_{22} = 1 : 0$). (b) Strip biaxial test ($E_{11} : E_{22} = 0 : 1$). (c) Biaxial test

Plain-woven textile design: Case 2

Second plain-woven textile design example consists of all three design variables as defined in Equation (6.5).

$$\mathcal{G}_{\mathrm{m}} = \begin{cases} \mathcal{G}^1_{\mathrm{m}} = w_1 & w_1 \in [0.762, 0.953] \\ \mathcal{G}^2_{\mathrm{m}} = w_2 & w_2 \in [0.762, 0.953] \end{cases} \tag{6.5a}$$

$$\mathcal{M}_{\mathrm{m}} = \begin{cases} \mathcal{M}^1_{\mathrm{m}} = \left[\frac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{warp}} & \left[\frac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{warp}} \in [0.5, 1.5] \\ \mathcal{M}^2_{\mathrm{m}} = \left[\frac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{weft}} & \left[\frac{\mathrm{E}}{\hat{\mathrm{E}}}\right]_{\mathrm{weft}} \in [0.5, 1.5] \end{cases} \tag{6.5b}$$

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}^1_{\mathrm{M}} = E_{11} & E_{11} \in [0.0, 0.22] \\ \mathcal{D}^2_{\mathrm{M}} = E_{22} & E_{22} \in [0.0, 0.22] \\ \mathcal{D}^3_{\mathrm{M}} = E_{12} & E_{12} \in [0.0, 0.20] \end{cases} \tag{6.5c}$$

As previously discussed, Young's modulus of yarns in the loading direction linearly scales the homogenised uniaxial response (see Figure 6.11). However, introducing warp and weft spacing as design variables, now it is not straightforward to predict the relationship between the design variables and homogenised response. Warp and weft spacings appear first in microscale RVE geometry, given by Equation (6.1) and secondly in volume averaged homogenised response, given by Equation (5.11). Hence, this complex relationship is expected to be captured by the GPR machine learning technique.

GPR model testing error quantification

We produce prediction errors for both energy and stress components, as shown in Figure 6.23. Considering the error convergence and efficiency of model training, we use a response database with 6400 data points in subsequent analyses. For this specific database: recorded optimum hyperparameters were $\sigma_f^2 = 11.832^2$, $\ell = 0.125$; maximum log marginal likelihood was 8780.429. Time taken for model training was recorded as 121 minutes and 27 seconds on an Intel Core i5-4590 CPU 3.30GHz $\times$ 4 processor.

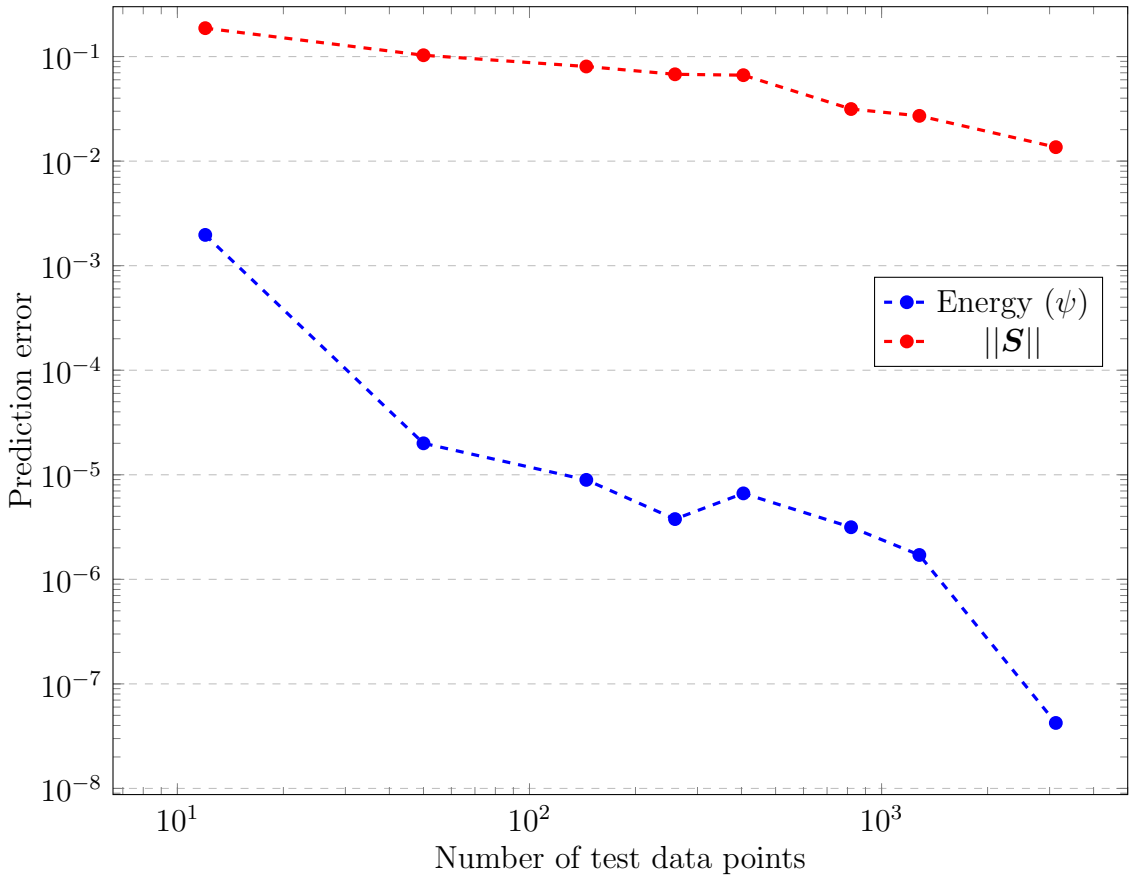
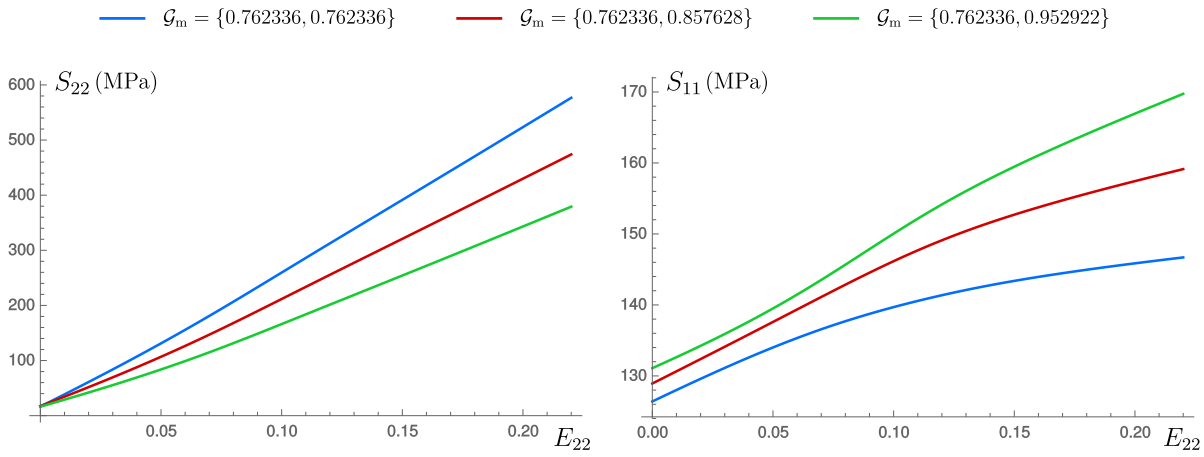
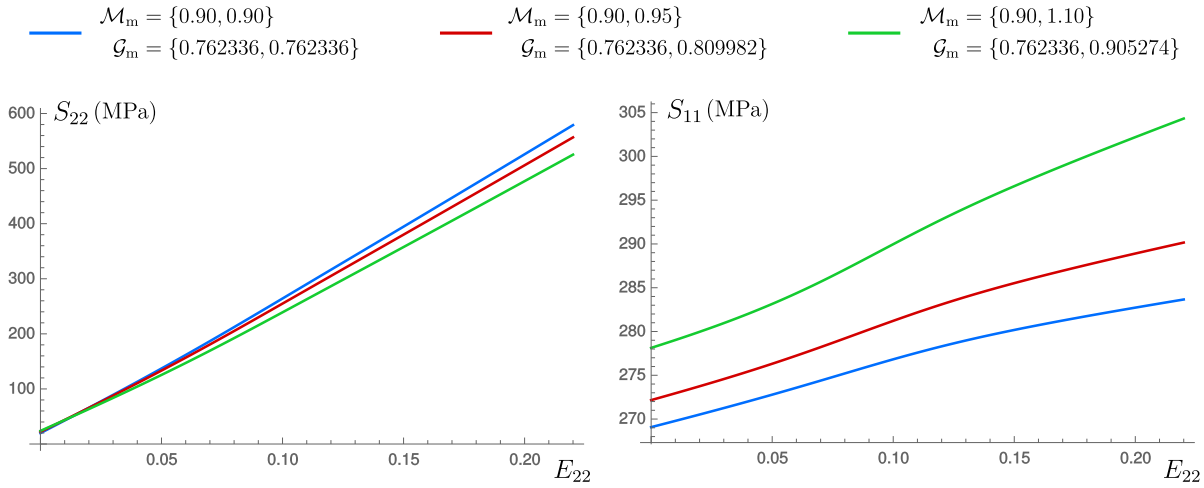


Figure 6.23: Design case 2: Error metrics of the predicted strain energy density and the norm of the vectorised second Piola-Kirchhoff stress tensor components

Now we visualise the influence of different combinations of design variables on the homogenised response of plain-woven RVEs. Figure 6.24a presents the variation of the homogenised axial stress components subject to different weft spacings. Figure 6.24b shows the homogenised axial stress component predictions for varied weft spacings and weft yarn Young's moduli. For clarity, results in Figure 6.24 consider constant E_{11} and E_{12}. These constant strain components and other fixed design variables are given in figure sub-captions. Non-design variables such as yarn radii and wave amplitudes are held constant and take values given in Table 6.1.



(a) $E_{11} = 5.125\%, E_{12} = 0.0\%, [\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{warp}} = [\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{weft}} = 0.90$



(b) $E_{11} = 10.5\%, E_{12} = 0.0\%$

Figure 6.24: Stress-strain relationships for different combinations of design variables. Homogenised responses S_{11} and S_{22} for (a) different weft spacings and (b) different weft spacings and weft yarn Young's moduli.

6.2 Weft-knitted fabrics

Knitted textiles are fundamentally different from woven textiles by the manner they are made. Knitting is the process of making textiles by interlooping a thread (yarn) where, in contrast, weaving involves interlacing of two or more yarns. On a broad picture, we identify two main types of knitted textiles: warp-knitting and weft-knitting. While weft-knitted textiles are easily handmade and commonly used, warp-knitted textiles

are popular products of knitting machines. There are other variants of knitted textiles based on stitch types and stitch patterns that we do not intend to discuss in this thesis. However, interested readers are invited to refer to Au [194] for further information. Furthermore, we restrict our focus only on the homogenisation of weft-knitted textiles as they are prevalent in various technical applications [1, 20] compared to warp-knitted textiles.

Vassiliadis et al. [114, 115] initiated geometric modelling, mechanical experiments and simulation of weft-knitted textiles for a range of stitch patterns. Weeger et al. [1] and Dinh et al. [20] continued to work on simplifying the complicated mechanical behaviour of weft-knitted textiles using computational homogenisation techniques. As common to any multiscale strategy, the first task is to identify an element to represent the macroscale continuum. Respecting the prerequisites highlighted in Section 5.1.2, we select a representative volume element as shown in Figure 6.25. The next task is to accurately model the identified analysis-suitable RVE using one of the geometric modelling techniques introduced in Chapter 2. In this section, we use quintic B-spline basis functions introduced in Section 2.1 for the isogeometric analysis of weft-knitted RVEs.

6.2.1 RVE Geometry modelling

As shown in Figure 6.25, we first define the yarn centrelines and render the full yarns by adding a radius to these centrelines. Moreover, we define the global Cartesian coordinate system in a manner where x and y axes correspond to the in-plane and z axis denotes the out-of-plane (thickness) direction of the RVE.

Weft-knitted RVE geometry is first defined in three different geometric equations for the three distinct regions, namely EM, MK and KA (see Figure 6.26). Later, they are connected smoothly using cubic B-splines at the boundary segments of the two intersections at M and K. Moreover, from a computational point of view, only yarn centerlines of the RVE are defined in the following equations. Thus, a constant diameter D renders the complete tubular geometry of the weft-knitted RVE [114, 115].

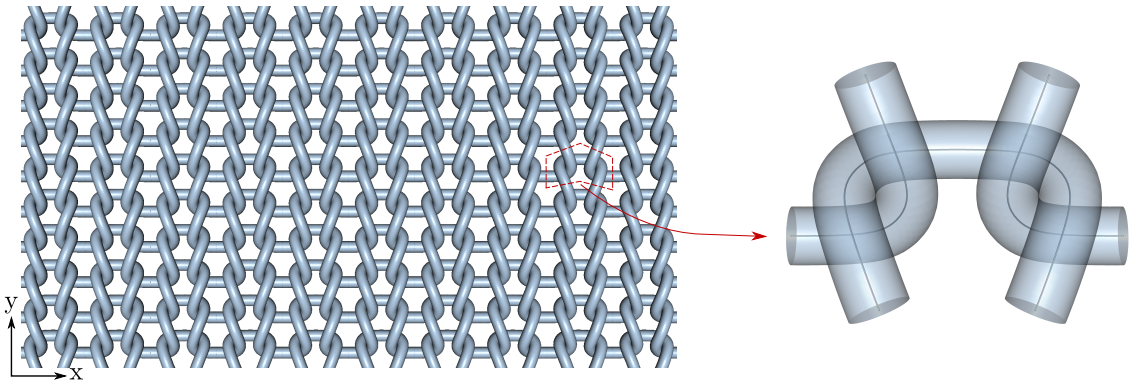
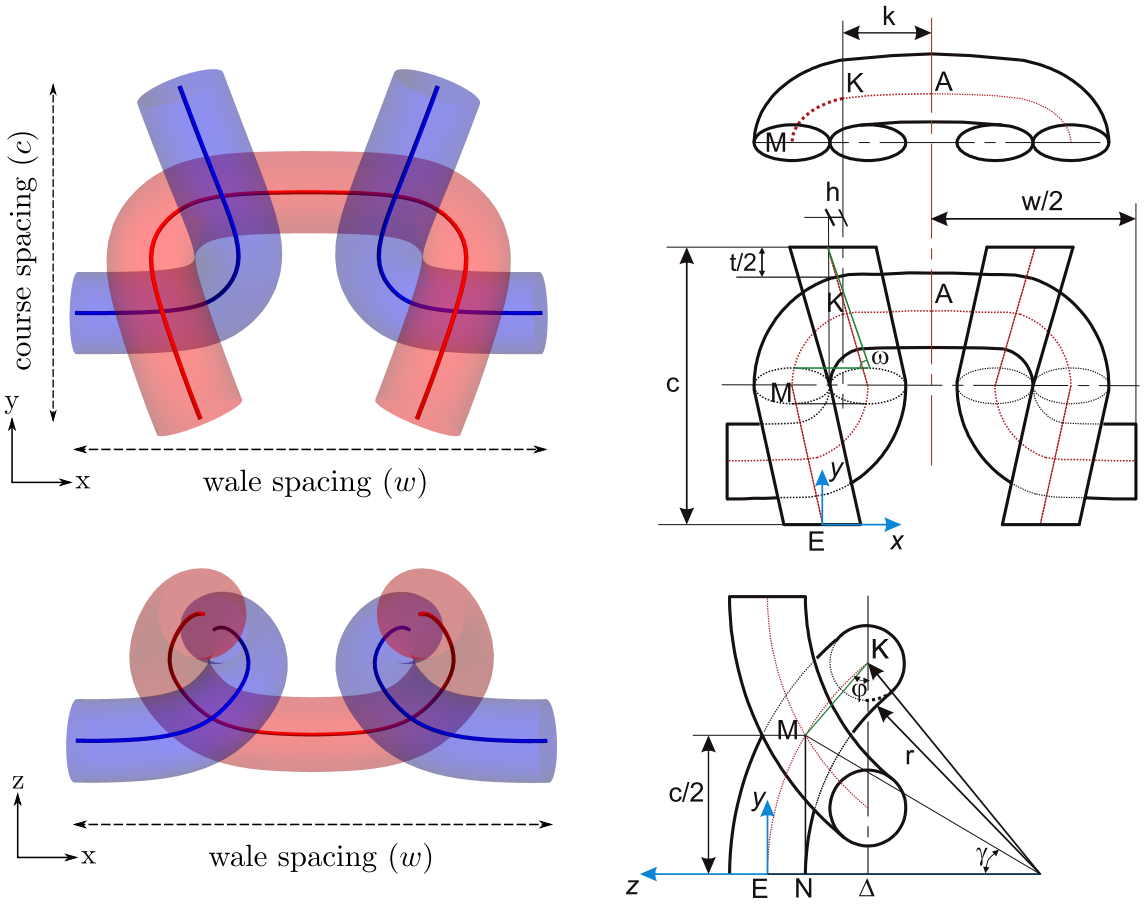


Figure 6.25: Selection of a multiscale analysis-suitable RVE of the weft-knitted textile.



(a) (b)

Figure 6.26: Geometry modelling of a weft-knitted RVE (a) Different views of a RVE of a weft-knitted textile (b) Introducing the geometric parameters that define the geometric description of the selected RVE [114]

.

The following geometric equations are extracted from Vassiliadis et al. [115] and smoothened at C^1 locations to recover the real smooth geometry of the RVE.

- Part EM $(0 < y < c/2)$:

$$x(y) = -\frac{D}{c}y$$
$$z(y) = \sqrt{\left(r + \frac{D}{2}\right)^2 - y^2} - \left(r + \frac{D}{2}\right)$$

  where $r = \dfrac{\left[\left(c - \frac{D}{2} - \frac{t}{2}\right)^2 - \left(\frac{D}{2} + \frac{t}{2}\right)^2\right]^2}{2D}$

- Part MK $(c/2 < y < c/2 + R)$:

$$x(y) = h - a\sqrt{1 - \left(\frac{y - c/2}{b}\right)}$$
$$z(y) = \sqrt{\left(r + \frac{D}{2}\right)^2 - y^2} - \left(r + \frac{D}{2}\right)$$

  where $h = \left(\frac{c}{2} - R\right)\tan\left(\frac{\pi}{2} - \omega\right)$, $\quad R = \frac{c}{2} - \frac{t}{2} - \frac{D}{2}$, $\quad \omega = \arctan\left(\frac{c - D\sin\gamma}{D}\right)$ and $\gamma = \arcsin\left(\frac{c/2}{r + D/2}\right)$

- Part KA $(x(y = c/2 + R) < x < w/4)$:

$$y(z) = \sqrt{\left(r + \frac{D}{2}\right)^2 - \left(z + r + \frac{D}{2}\right)^2}$$
$$z(x) = OZ - \sqrt{A^2 - (x - OX)^2}$$

  where $OX = w/4$, $\quad OZ = \dfrac{(x_2 - OX)^2 - (x_1 - OX)^2 + z_2^2 - z_1^2}{2(z_2 - z_1)}$ and $A = \sqrt{(x_1 - OX)^2 + (z_1 - OZ)^2}$ Additionally, (x_1, z_1) and (x_2, z_2) are the coordinates of two points in part MK with $y_1 = c/2 + R - 0.001$ and $y_2 = c/2 + R$

- A cubic B-spline curve was fitted at C^0 continuous locations namely, M and K. Moreover, this smooth curve EMKA is mirrored on the vertical axis passing at point A to obtain the complete geometry of the red yarn shown in Figure 6.26.

- Lastly, one smooth EMKA is rotated and translated appropriately to create the left blue yarn of Figure 6.26 and mirrored subsequently, to complete the RVE geometry.

For model verification and homogenised RVE response validation, we use the problem parameters as given in Table 6.2, which have been extracted from the works of Weeger et al. [1].

Table 6.2: Weft-knitted RVE yarn geometric and material parameters

D(mm)	c(mm)	w(mm)	t(mm)	Young's modulus(N/mm^2)	Poisson's ratio
0.1845	0.4857	0.8327	0.072	800.0	0.0

6.2.2 Microscale boundary value problems

Similar to plain-woven textiles in Section 6.1, finite element analysis is used for RVE simulations subjected to periodic boundary conditions. Here, four main analyses have been considered for the simulations namely: uniaxial tension in course direction (course-wise uniaxial tension); uniaxial tension in wale direction (wale-wise uniaxial tension); biaxial tension and pure shear. The boundary value problems for the aforementioned four analyses are illustrated in Figure 6.27.

Due to the complicated load transferring mechanism and fuzziness of the fibrous microstructure, we make the following assumptions to idealise the kinematics of real weft-knitted textiles [1, 114].

- We penalise yarn bending stiffnesses by a factor of 0.5 to account for the fuzziness of the knitting yarns. Thus, the Young's modulus used to compute axial stiffness (EA), is halved to compute the bending stiffnesses ($0.5EI_2$ and $0.5EI_3$). See Equation (3.59b).
- For numerical stability and consistency, initial frictionless point-to-point yarn contacts are enforced by introducing an artificial penetration depth of magnitude 10^{-5}mm.

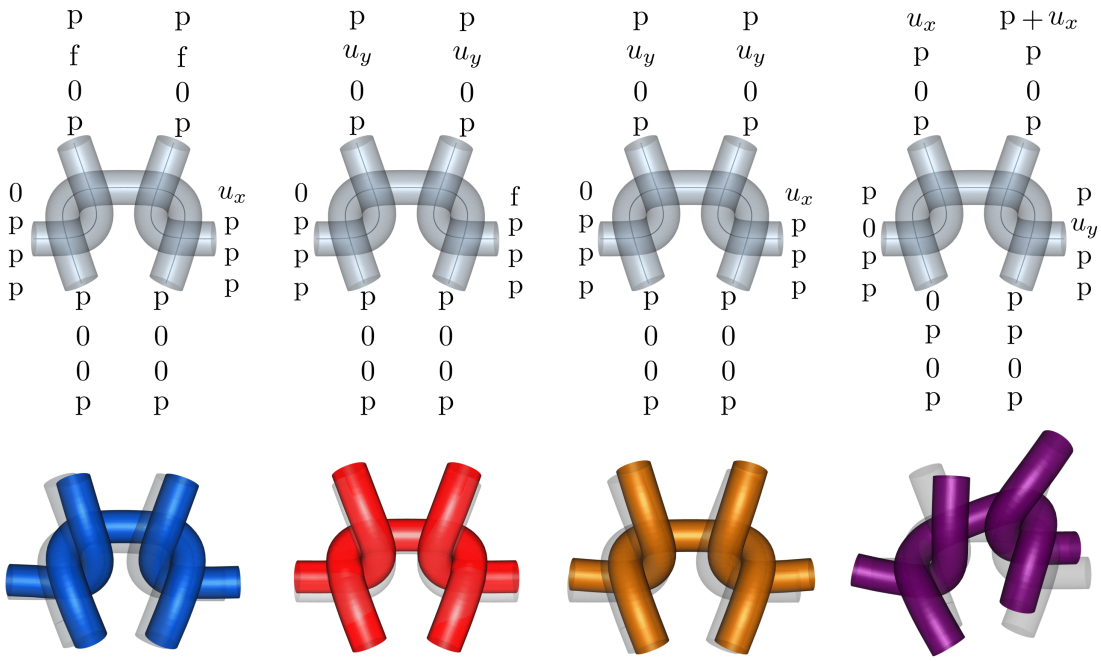


Figure 6.27: Periodic boundary condition enforcement (top row) and respective deformed RVEs (bottom row). At every boundary node, four characters denote the constraint enforcement to each degree of freedom at the respective boundary node. u_i refers to the applied displacements in the i direction. Moreover, $p, f, 0$ refer to *p*eriodic, *f*ree and *zero*-displacement constraints respectively.

Verification of the numerical model

We use the experimental and numerical results obtained by Vassiliadis et al. [114], Weeger et al. [1] and Dinh et al. [20] to compare the homogenised responses of weft-knitted RVEs. These results correspond to two strip biaxial tests and one pure shear test. Two strip biaxial tests refer to $E_{11} : E_{22} = 1 : 0$ (course-wise) and $E_{11} : E_{22} = 0 : 1$ (wale-wise). Moreover, for comparison purpose, we use a unit conversion 1 N/mm = 1019.716 gf/cm.

Result comparison is presented in Figure 6.28 and we conclude our microscale RVE simulations approximate the experimental results very well, and follow a better trend than that of the numerical results from Weeger et al. [1]. It is evident that the homogenised responses of strip biaxial tests are nonlinear and RVEs are stiffened (increasing gradients) at higher strains. Moreover, the shear response is observed to be nearly linear.

Thus, shear stiffnesses at each strain level can be considered constant and comparatively lower. We can safely assume there are no axial-shear coupling effects due to yarn alignments in Figure 6.26. Therefore, we use a constant shear modulus in macroscale simulations to relate shear strains to shear stresses.

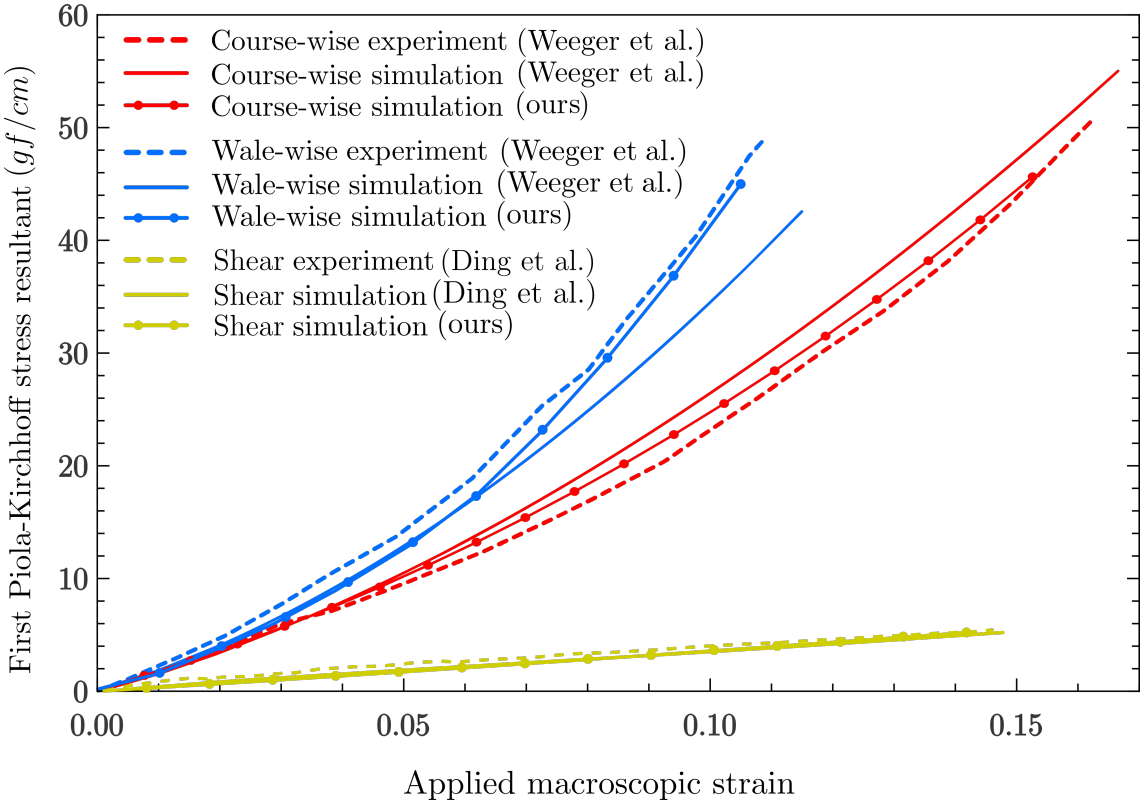


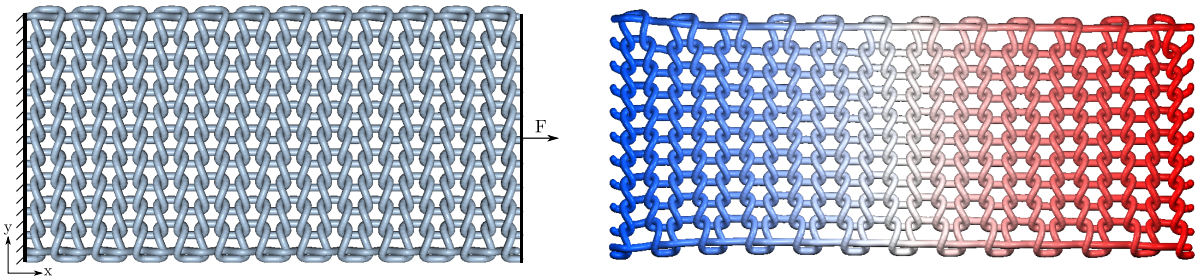
Figure 6.28: Comparison of simulation results for course-wise and wale-wise strip biaxial tension and pure shear test against the experimental results obtained in Vassiliadis et al. [114] and numerical simulations by Weeger et al. [1], Dinh et al. [20].

Yarn-level simulation of a weft-knitted fabric

Similar to plain-woven textiles (see Figure 6.29), we develop a yarn-level phenomenological model of a weft-knitted fabric to visually verify the deformations of a pull test. A specimen weft-knitted textile made of 12×10 (course loops$\times$wale loops) loops is loaded in the x-direction by a 10% strain level (see Figure 6.29a). Figure 6.29 illustrates the deformation of the weft-knitted specimen held in place at the top and bottom edges by two bounding yarns. Some of the key observations can be discussed not only to

visually verify the phenomenological model but also to verify the accuracy of the finite deformable rod model introduced in Section 3.

- Poisson's effects: The transverse deformation (in the y-direction) of the yarn-level model due to loading in axial (x) direction, clearly highlights the Poisson's effects in weft-knitted textiles.
- Deformed loops: Loops are elongated predominantly in the x-direction by maintaining stable contacts at penetration-prone locations.
- Decrimping of bounding yarns: Two bounding yarns are used to avoid the unravelling of knitted loops. As expected, they have been decrimped (nearly straightened) during the deformation process.
- Asymmetry about mid-y-plane: Reference geometry in Figure 6.29a is not symmetric about mid-y-plane. This is evident because RVE in Figure 6.26 is also asymmetric about mid-y-plane. Thus we observe distinct deformations in the top and bottom edges of the fabric. However, in reality, these are not observed due to tightly restrained edges by applying selvedge stitches [194].



(a) (b)

Figure 6.29: Yarn-level pull test of a weft-knitted fabric. (a) Problem description of the reference configuration. (b) Deformed configuration due to the applied load in the x-direction. Colour distribution refers to the x-direction displacements of the yarns.

Sensitivity analysis

A sensitivity analysis is carried out to investigate the dependency of the homogenised response on material and geometric property parameters. This is qualitatively a vital step in building the response database discussed in Section 5.3.1. The sensitivity of each

parameter is important to investigate the parameter dependencies, redundant parameters and uncertainties of each parameter on the homogenised response of RVEs. For brevity, only homogenised tensile response is considered herein for the sensitivity analysis of weft-knitted RVEs. As common to sensitivity analyses, we change one variable and hold others constant, unless there are any variable dependencies.

RVE size

It is a common practice in computational homogenisation communities to study the convergence of the homogenised response for different RVE sizes. Figure 6.30 shows the convergence results for different sizes of weft-knitted RVEs. For this study, we use the same number of loops in both course and wale directions. From a theoretical standpoint, however, it is possible to consider different combinations of course and wale loops that can represent the weft-knitted textile in the macroscale. Moreover, for notational convenience, we use $n_c \times n_w$ to classify RVEs by their number of course loops (n_c) and wale loops (n_w).

When the RVE size is increased from 1×1 to 6×6, as shown in Figure 6.30, we can clearly observe the convergence of the homogenised response P_{11}. This RVE size convergence is important to establish the principle of separation of scales for using hierarchical computational homogenisation. Unlike in woven textiles, there are some variations in this convergence study due to the frictionless contacts and the comparative complexity of the RVE model in terms of arbitrary initial curvatures and out-of-plane yarn deformation. However, since we do not observe significant changes in the homogenised response, and also considering the numerical efficiency of the database construction, 1×1 RVEs are used henceforth.

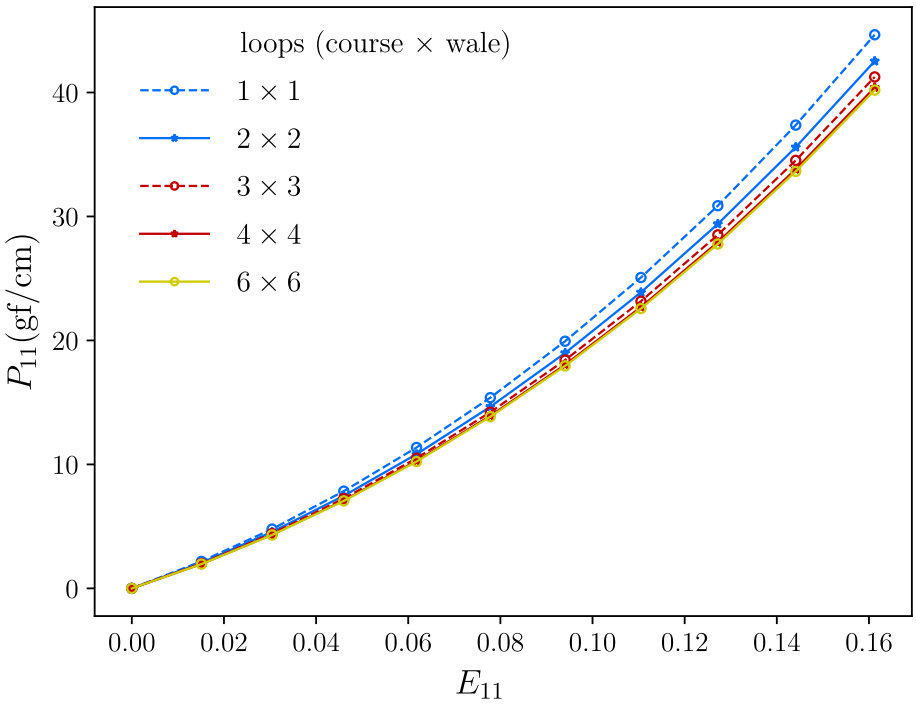


Figure 6.30: Convergence of the homogenised response for different sizes of weft-knitted RVEs. Equal number of loops in course and wale directions have been considered here.

Young's modulus of yarns

Figure 6.31 presents the sensitivity of yarn Young's modulus on the homogenised axial response of the weft-knitted RVE. As expected, we can observe a linear increase of the stress resultant P_{11}. This is obvious as we are using a linear isotropic material to analyse the RVEs. $\hat{\text{E}}$ in Figure 6.31 refers to the original value of Young's modulus used to verify the microscale simulations in Section 6.2.2 (see Table 6.2).

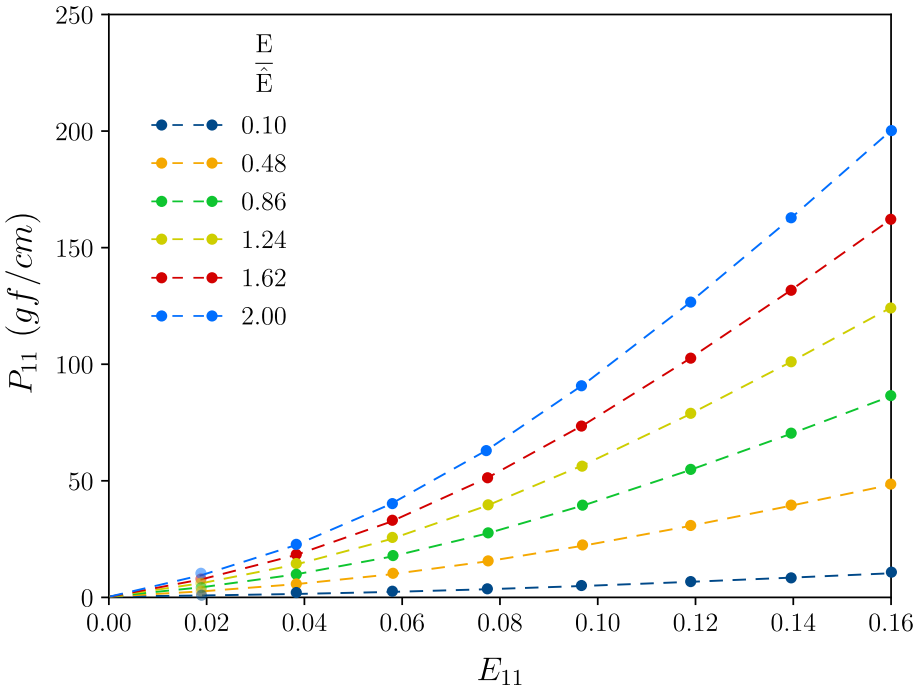


Figure 6.31: Sensitivity analysis of the Young's modulus of yarns on the homogenised response of weft-knitted RVEs.

6.2.3 Data-driven macroscale analysis

Data-driven multiscale modelling proposed in Section 5.3 is evaluated here for the computational homogenisation of weft-knitted textiles. Provision of microscale model verifications, convergence study and sensitivity analyses in Section 6.2.2 constitute the applicability of machine learning for macroscale analysis.

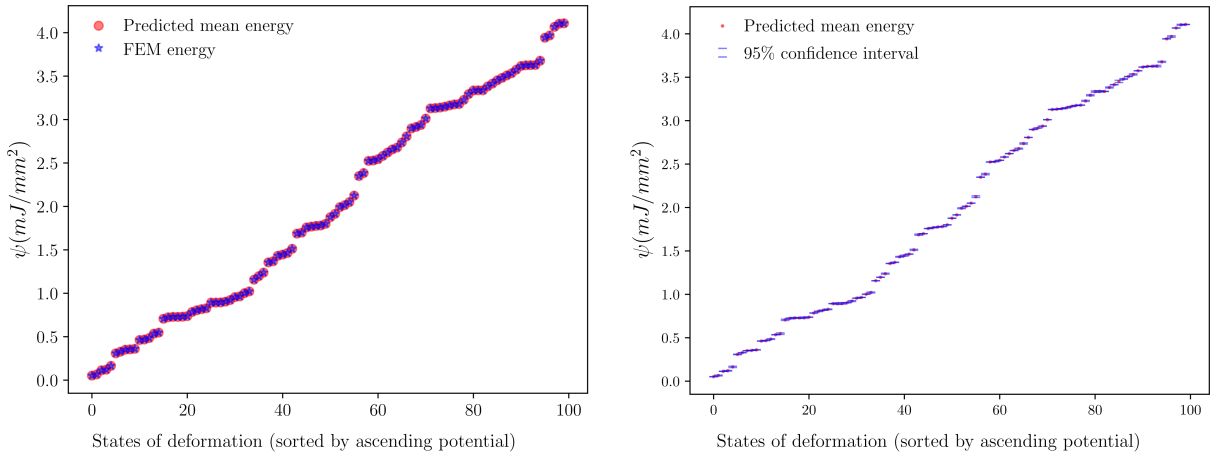
Similar to data-driven plain-woven macroscale analysis, we represent the response database succinctly as $(E_{11}, E_{22}, E_{12}, \psi)$ where $\boldsymbol{E}$ is the Green-Lagrange strain tensor and ψ gives the area-averaged energy in mJ/mm^2.

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}_{\mathrm{M}}^{1} = E_{11} & E_{11} \in [-0.05, 0.15] \\ \mathcal{D}_{\mathrm{M}}^{2} = E_{22} & E_{22} \in [-0.05, 0.15] \\ \mathcal{D}_{\mathrm{M}}^{3} = E_{12} & E_{12} \in [0.0, 0.15] \end{cases} \tag{6.6}$$

Geometric ($\mathcal{G}_{\mathrm{m}}$) and material ($\mathcal{M}_{\mathrm{m}}$) parameters are held constant and take the values given in Table 6.2.

Strain energy prediction and uncertainty

Similar to plain-woven textiles, we train a GPR model using the constructed response database as given in Equation (6.6). A small but complete subset of the full database is extracted to train a GPR model. As shown in Figure 6.32, we use 90 new data points from the database to test the accuracy of the trained GPR model. Figure 6.32 also presents the corresponding 95% confidence interval of the predicted energy of the new data points. For this particular database, training and testing errors were recorded as 6.52×10^{-12} and 3.48×10^{-6}. Detailed analysis on these errors, optimised hyperparameters and maximum log marginal likelihood values are provided later in this chapter.

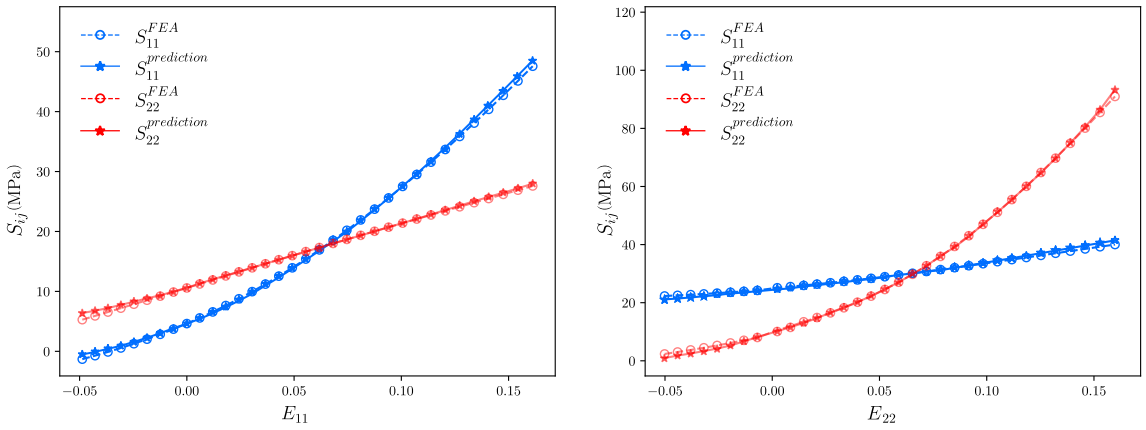


(a) (b)

Figure 6.32: (a) Strain energy prediction for unseen data points sorted by ascending potentials (b) 95% confidence interval of the predicted mean energy.

Stress predictions

Accuracy test on predicted energy has to be followed by an accuracy test on the stress components. Using Equations (5.41) and (3.35) we evaluate the degree of accuracy of the predicted stresses to that of observed stresses in the database. Figure 6.33 provides two principal stress component comparisons thus setting $E_{12} = 0.0$. For visualisation purpose, one strain component is held constant while the other is increased from 0.0 to 0.16. We can observe nearly exact stress predictions from the trained GPR model. Moreover, we notice slight deviations close to the bounds of strain components which is treated by refining the data points close to these bounds.



(a) (b)

Figure 6.33: Prediction and comparison of the second Piola-Kirchhoff stress components. (a) S_{11}, S_{22} for $E_{11} = [0.\%, 22\%]$ and $E_{22} = 8\%$. (b) S_{11}, S_{22} for $E_{22} = [0.\%, 22\%]$ and $E_{11} = 10\%$.

GPR model testing error quantification

Here, similar to plain-woven textiles, we present the variability of GPR model testing errors for energy and stress predictions. This is a vital step to investigate the error convergence and algorithmic efficiency of the model training and testing phase. Equation (5.48) is used to compute the mean squared errors (MSE) of energy predictions.

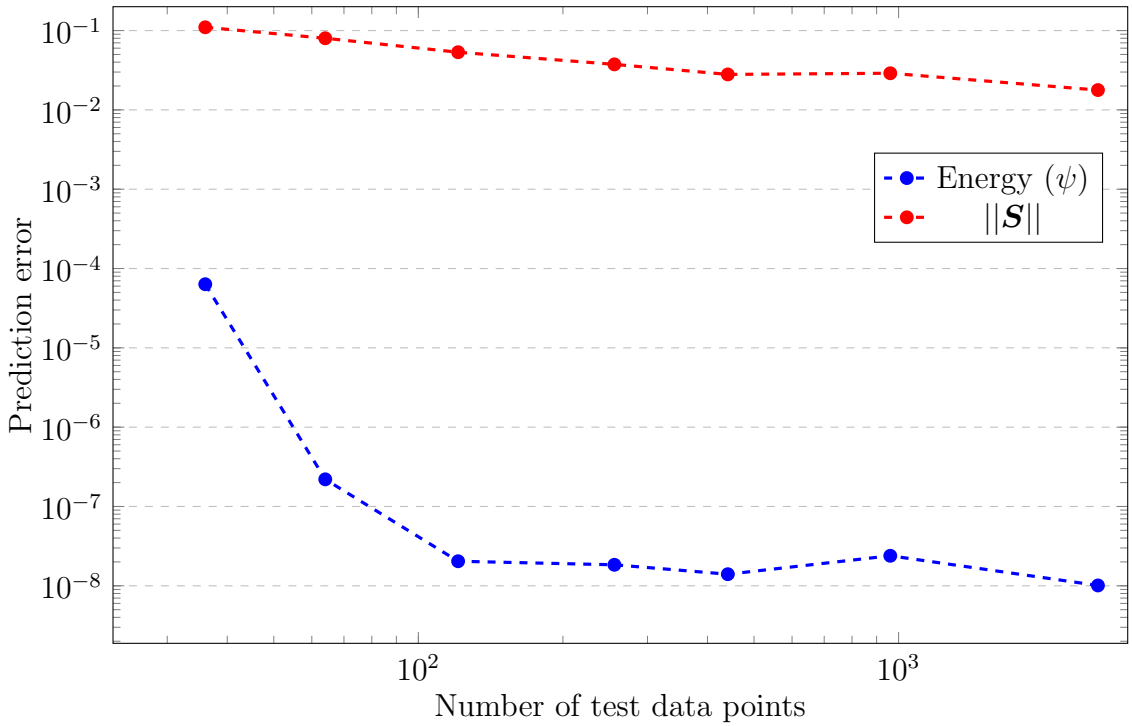


Figure 6.34: Error metrics of the predicted strain energy density and the norm of the vectorised second Piola-Kirchhoff stress tensor components

As expected, errors in stress predictions are comparatively higher than the energy predictions. Considering the error convergence and algorithmic efficiency, we use a response database with 2601 test data points. Moreover, for every database, we optimise the log marginal likelihood (given in Equation (5.31)) with respect to hyperparameters and arrive at the optimum hyperparameters that yield the maximum log marginal likelihood. Table A.3 presents a concise tabulated results of the error analyses (R^2, MSE), maximum log marginal likelihood (MLML) and optimum hyperparameters (σ_f, ℓ).

Pull tests on data-driven weft-knitted textiles

Numerically tested and verified GPR model is next applied to simulate the membrane response of pull tests along the two principal directions, namely course-wise and wale-wise pull tests. Each membrane specimen is a square of side 10mm. These tests are designed in a way where one edge is held fixed and the opposite edge is pulled in the course or wale directions by 1mm. The non-zero stress component distributions are shown in figures 6.35 and 6.36.

It is important to emphasise on the asymmetry of the stress distributions predicted by the Gaussian process model where S_{11} in Figure 6.35 is not equal to S_{22} in Figure 6.36. This is mainly due to the anisotropy of the mechanical response of RVEs in the course and wale directions. This is also observed in the numerical strip biaxial tests used in microscale model verification (see Figure 6.28). In Figure 6.28, for a constant macroscopic strain of 10% (E_{11} or E_{22}), we observe the corresponding stress resultants taking different values, $S_{11} = 24.53$ gf/cm and $S_{22} = 44.27$ gf/cm, respectively.

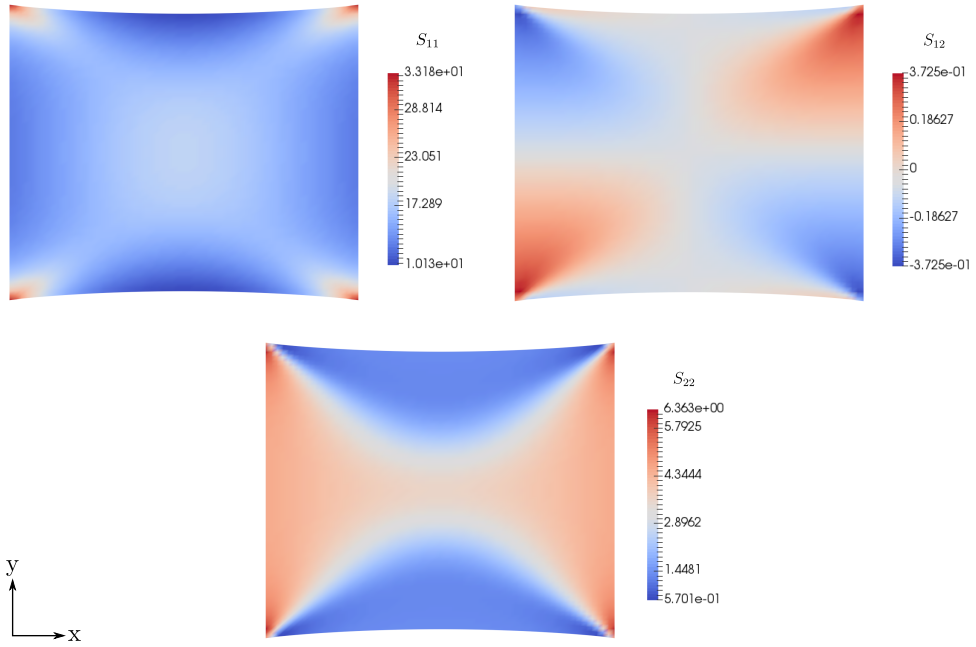


Figure 6.35: Stress distribution of a course-wise pull test on a data-driven weft-knitted textile

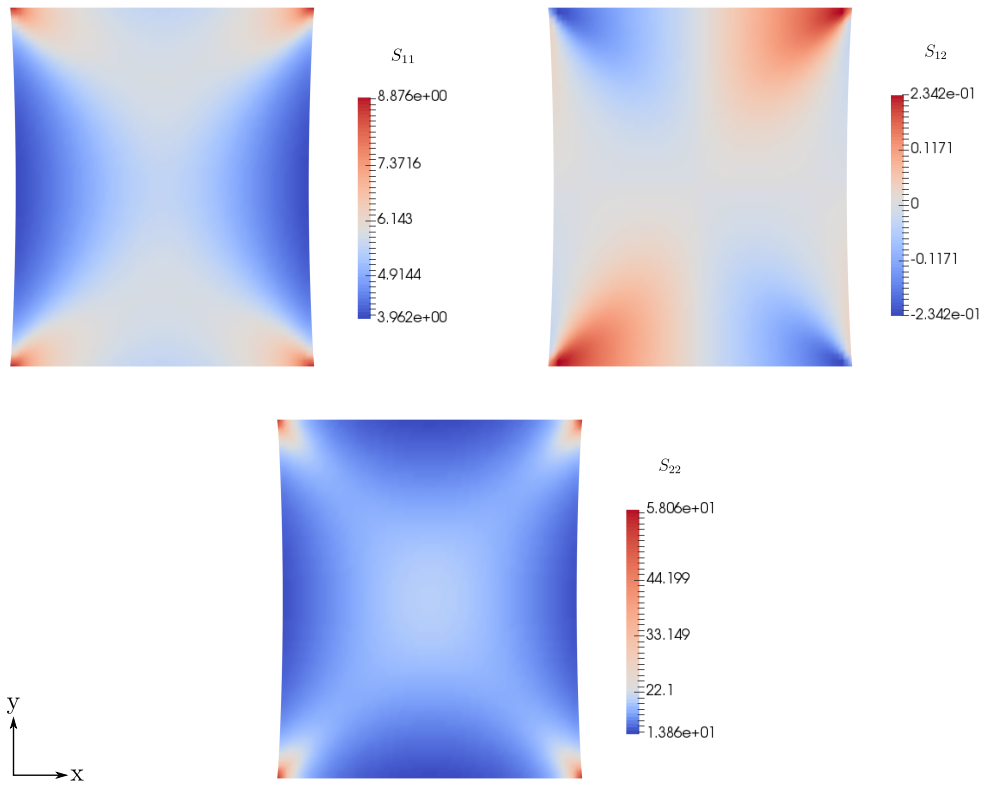
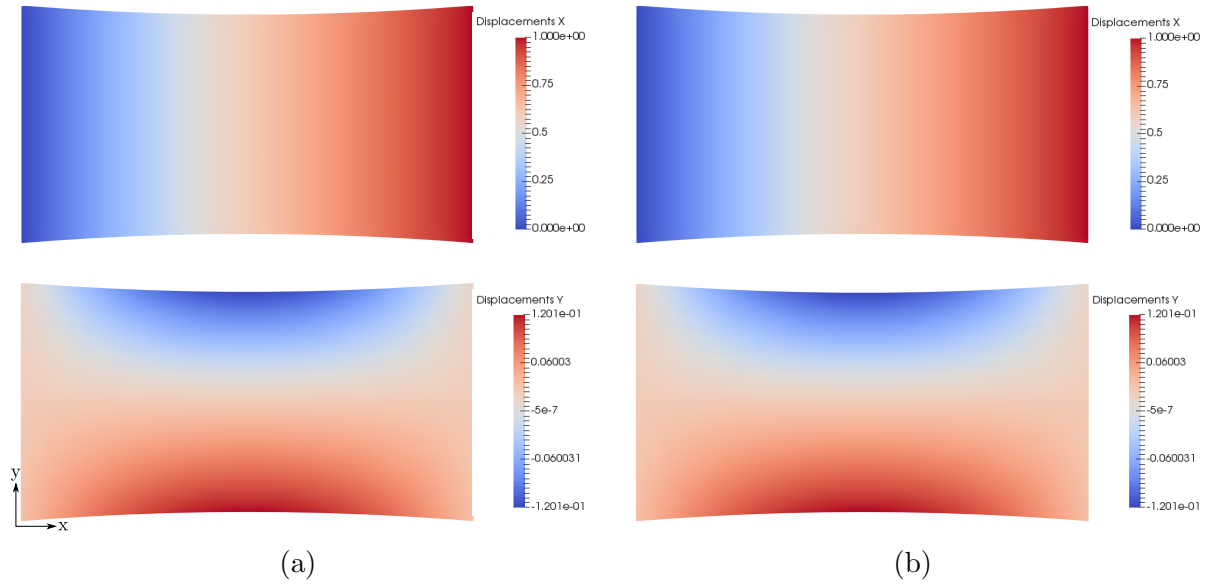


Figure 6.36: Stress distribution of a wale-wise pull test on a data-driven weft-knitted textile

Yarn-level versus data-driven homogenisation

After discussing the data-driven homogenisation of knitted membranes, we present a quantitative comparison of the problem presented in Figure 6.29. The yarn-level model comprises of 12 loops in the course direction and 12 loops in the wale direction. Hence, we use an equal size membrane, 10.00mm long, 5.85mm wide and 0.2356mm thick, for the homogenised macroscale analysis. Homogenised membrane model is analysed using a 12×12 structured mesh using the proposed Gaussian process homogenisation technique. For comparison purposes, yarn-level model in 6.37a is created by projecting it onto the xy plane using Delaunay mesh generation. We can observe a very good agreement between the two models. However, y-direction displacement distribution of the yarn model in Figure 6.37a is slightly different to that of the homogenised membrane in Figure 6.37b. This difference results mainly from two reasons. First, zero out-of-plane displacement boundary conditions on the top and bottom edges of the yarn-level model directly contribute to the observed difference. These boundary conditions are applied to

simulate selvage stitches that restrict the top and bottom yarns being freely straightened during deformation. Secondly, the geometric asymmetry of the yarn-level model about the mid-horizontal plane has an influence on the overall response due to the relatively small number of loops used in both course and wale directions.



(a) (b)

Figure 6.37: Quantitative comparison between yarn-level and homogenised models. Here, x and y displacement distributions of (a) yarn-level model and (b) homogenised knitted membrane of a pull test along x-direction are illustrated.

6.2.4 Data-driven weft-knitted textile design

The extension to data-driven weft-knitted textile design is achieved by incorporating the design (material and geometric) parameters into the response databases. We simultaneously vary the Young's moduli and radius of yarns constrained by the prerequisites discussed in Section 5.4.1. We define the design response database entries succinctly as $(E_{11}, E_{22}, E_{12}, \mathrm{E}/\hat{\mathrm{E}}, \mathrm{R}/\hat{\mathrm{R}}, \psi)$. We specify the feature bounds to be used in constructing

the response database as given in Equation (6.7).

$$\mathcal{G}_{\mathrm{m}} = \left\{ \mathcal{G}_{\mathrm{m}}^{1} = \frac{\mathrm{R}}{\hat{\mathrm{R}}} \quad \frac{\mathrm{R}}{\hat{\mathrm{R}}} \in [0.8, 1.2] \right. \tag{6.7a}$$

$$\mathcal{M}_{\mathrm{m}} = \left\{ \mathcal{M}_{\mathrm{m}}^{1} = \frac{\mathrm{E}}{\hat{\mathrm{E}}} \quad \frac{\mathrm{E}}{\hat{\mathrm{E}}} \in [0.01, 3.0] \right. \tag{6.7b}$$

$$\mathcal{D}_{\mathrm{M}} = \begin{cases} \mathcal{D}_{\mathrm{M}}^{1} = E_{11} & E_{11} \in [-0.05, 0.15] \\ \mathcal{D}_{\mathrm{M}}^{2} = E_{22} & E_{22} \in [-0.05, 0.15] \\ \mathcal{D}_{\mathrm{M}}^{3} = E_{12} & E_{12} \in [0.0, 0.15] \end{cases} \tag{6.7c}$$

GPR model testing error quantification

We quantify the prediction errors to measure the accuracy of the trained GPR model for plain-woven textile design. Following a similar approach considered in Section 6.2.3, we produce prediction errors for both energy and stress components, as shown in Figure 6.38. Variation and convergence of these errors follow a similar pattern to Figure 6.34. Considering the error convergence and efficiency of model training, we use a response database with 4096 data points. For this specific database: recorded optimum hyperparameters were $\sigma_f^2 = 6.721^2$, $\ell = 0.112$; maximum log marginal likelihood was 15890.915. Time taken for model training was recorded as 33 minutes and 28 seconds on an Intel Core i5-4590 CPU 3.30GHz $\times$ 4 processor.

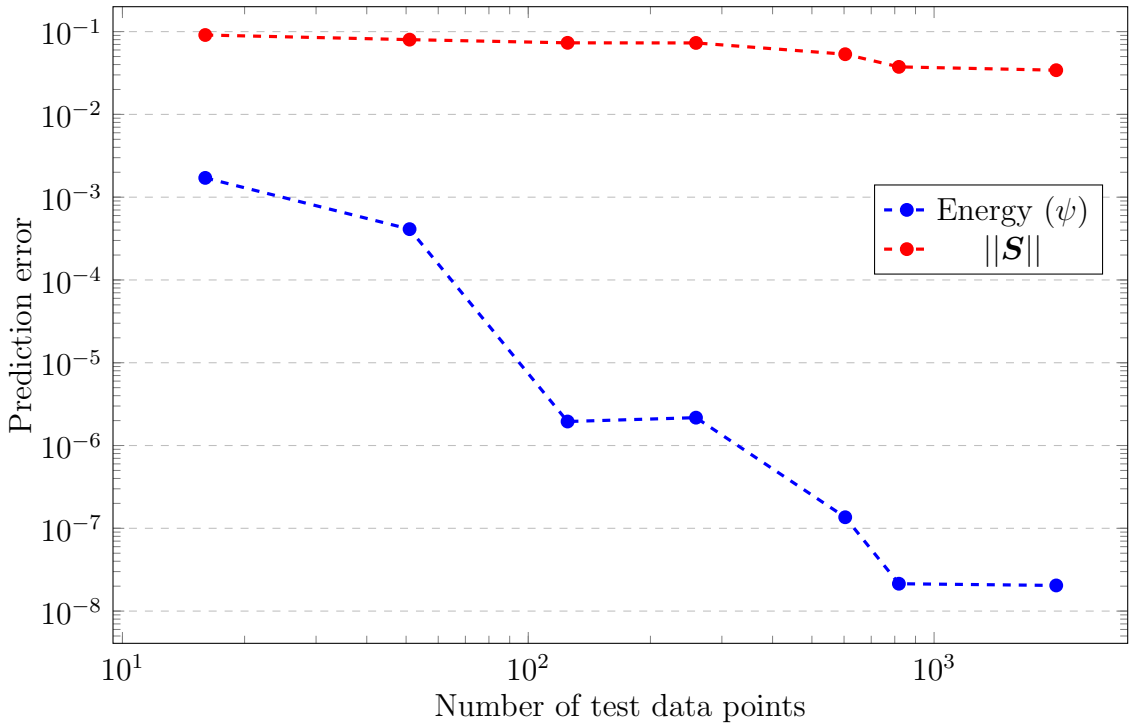
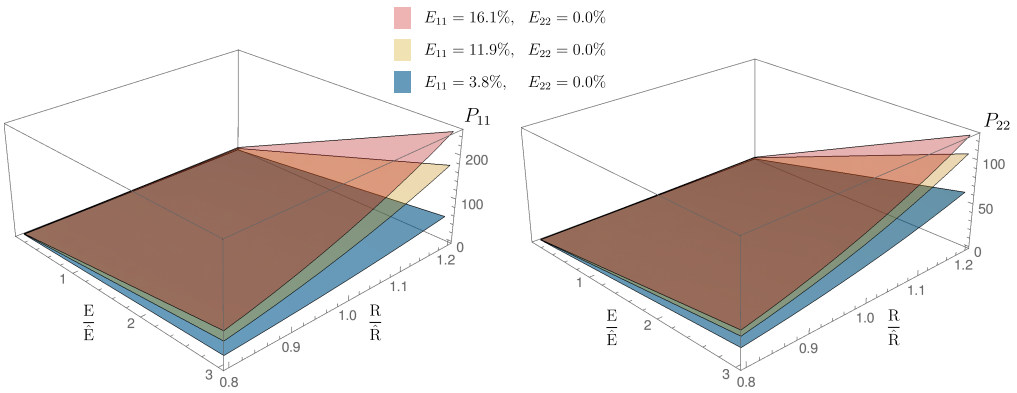


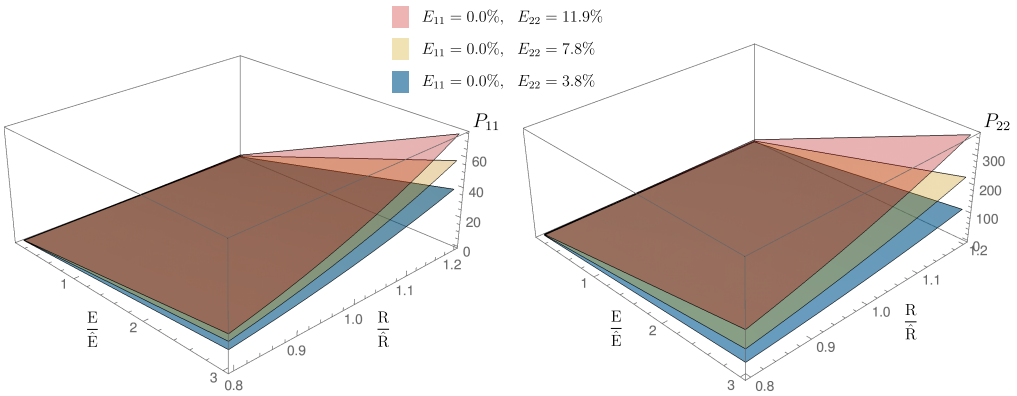
Figure 6.38: Error metrics of the predicted strain energy density and the norm of the vectorised second Piola-Kirchhoff stress tensor components

Weft-knitted textile design examples

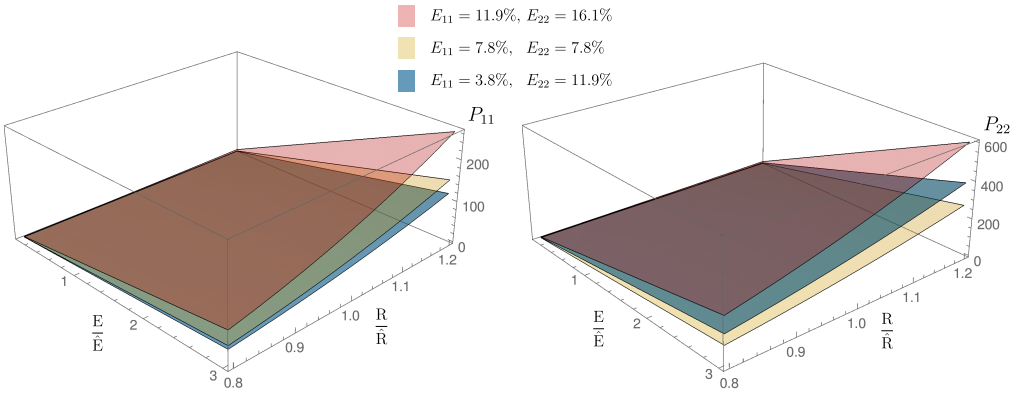
Now we are interested to see how the stress-strain variations would look like for a given combination of $\mathrm{E}/\hat{\mathrm{E}}$ and $\mathrm{R}/\hat{\mathrm{R}}$. We start with two strip biaxial tests, as shown in Figures 6.39a and 6.39b, where strain combinations $E_{11} : E_{22} = 1 : 0$ and $E_{11} : E_{22} = 0 : 1$ have been enforced respectively on the RVEs. Then we vary $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{warp}}$ and $[\mathrm{E}/\hat{\mathrm{E}}]_{\mathrm{weft}}$ on a refined grid consisting 20736 data points. This refined grid is defined in a way that only Young's modulus ratio bounds (0.01 and 3.0) and radius ratio bounds (0.8 and 1.2) are common to the response database used for model training and the refined grid. Thus, except for bounds, all other points act as unseen data points to the trained GPR model. Next, we present the stress-strain relationship of a biaxial test in Figure 6.39c. For illustration purpose, arbitrary strain values are used to represent different states of deformations.



(a)



(b)



(c)

Figure 6.39: Stress distributions for different combinations of yarn Young's modulus and radius. Results of three biaxial tests are shown in each figure. (a) Strip biaxial test ($E_{11} : E_{22} = 1 : 0$). (b) Strip biaxial test ($E_{11} : E_{22} = 0 : 1$). (c) Biaxial test

7 Conclusions and Future Research

7.1 Conclusions

The primary aim of this thesis is to introduce and detail unified data-driven frameworks for nonlinear computational homogenisation and material design of textiles. On the journey to the successful completion of the primary aim, there were secondary but significant objectives achieved that are regarded as contributions towards respective research fields. At this juncture, all these contributions are summarised in the appreciation of this research work.

- Non-smooth manifold-based basis functions: A novel set of univariate basis functions were introduced specifically for modelling non-smooth rod structures. Frames, lattices and trusses usually have one or more $\mathbb{C}^1$-smooth locations within the structure. Using non-smooth manifold basis functions we have proven that such geometries can be represented exactly. Moreover, the existence of these non-smooth features as singularities of smooth curves can also be easily incorporated using manifold basis functions. This is straightforward as one needs to choose between smooth or non-smooth local polynomials accordingly during the mapping from parametric to physical domains.

In Chapter 3, we detailed rigorous formulations and implementation aspects of a finite deformable spatial rod model. Some key contributions deserve to be mentioned.

- Analytical expressions for initially curved rods: Using Jacobian expansions we derived analytical expressions for initially curved rods. Bauer et al. [52] approximated stretching internal force contributions from initial curvatures. However, our formulations not only deduced exact internal force contributions (both stretching and bending) but also attained the Hessian contributions.

- Non-conservative loading simulations: Another demerit of many beam formulations is the inability to handle non-conservative (follower) loading. Under the domain of differential geometry, we have successfully formulated and verified the applicability of follower loading in rod structures. Moreover, in a more general context, our formulations can handle external stiffness matrices arising from non-zero external loading tangents.

- Isogeometric analysis of hinged beams: Isogeometric analysis of hinged beams using the introduced non-smooth manifold basis functions was implemented. One example was used to verify the results and the same example was used to study the error convergence rate of using manifold basis functions in isogeometric analysis.

- Anisotropic thin-shell simulator: Based on a differential geometric approach, implementation of an anisotropic thin-shell simulator has many advantages. In this context, the most important advantage would be the ability to implement linear computational homogenisation of anisotropic shells. Accuracy of the implementations was measured against a well-established example from the literature and the optimal convergence was also observed.

- Data-driven computational homogenisation of textiles: In-depth discussions on machine learning, specifically on Gaussian Process Regression technique, have been thoroughly explored in Chapter 5 to introduce machine learning in the context of computational homogenisation. Extensive formulations of Gaussian Process Regression were implemented within a finite element environment to initiate a data-driven nonlinear computational homogenisation scheme for technical textiles. Appreciation of the merits of using machine learning in a FE^2 framework was elaborated in Chapter 6 using two types of textiles namely, plain-woven and weft-knitted.

- Data-driven textile design: Lastly, another advantage of using machine learning is exploited to incorporate material design in computational homogenisation. Several prerequisites, concerning geometric constraints and numerical stability, were established to perform data-driven material designs of textiles. Each application in Chapter 6 was extended to include material designs and some examples were included to prove the robustness of simulations.

7.2 Future work

The presented research work in this thesis can be extended in many ways.

- Though, the attention was focussed only on plain-woven and weft-knitted fabrics, the presented data-driven computational homogenisation strategy can be applied to other stitch and weave pattern types. In woven textiles, this extension can be used to homogenise satin, twill, basketweave, dobby, cross and jacquard weave types. Different stitch types in weft-knitted and warp-knitted can also be homogenised using the proposed data-driven framework.
- Data-driven material design framework can be further extended by adding more material and geometric parameters. Thus, not only the design framework becomes more general and versatile but also gives more handles to control the homogenised response of plain-woven and weft-knitted fabrics.
- Similar to computational homogenisation of other weave and stitch types of woven and knitted textiles, we can use the data-driven material design framework to design for other stitch and weave types.
- Departing from textiles, aforementioned data-driven techniques can be directly applied in homogenisation and design of fibre composites, lattice structures and other 3D printable periodic structures. This extension should be equipped with bending deformations which are dominant in non-textile applications.
- Spatial rod model can be further extended to include material anisotropy in both linear and nonlinear regimes. Moreover, warping effects can be included to simulate warping torsion effects in beams. Extension of the rod model to include linear material anisotropy would not take much efforts as a similar implementation has been carried out for thin-shells.
- Another extension would be to try different regression machine learning techniques to assess the accuracy of predictions, algorithmic efficiency and overall computational budget. This could be interesting if one uses complex material models with finite deformations and rotations to perform computational homogenisation.

- A direct extension would be to include plasticity and peridynamics into the computational homogenisation framework. This would be crucial in analysing the failure criterion for different textiles.

- Departing from computational homogenisation, we can use the spatial rod model to arrive at Bayesian displacement models. This can be accomplished in many ways. One straightforward technique would be to add probability distributions to applied loading (or displacements) so the model results not only a mean displacement distribution, but also predicts its variations given a confidence interval. Another way would be to depart from the material models. For instance, one would attach a probability distribution to the Young's modulus of a linear isotropic material and arrive at a similar statistical displacement field. In a broad picture, this can be identified as uncertainty quantification in the context of continuum mechanics.

- On a slightly different research focus, we can use data-driven techniques in optimising structures. For instance, consider optimising a steel structure. Prevalent topology, shape and size optimisation schemes has a barrier to entry into the industrial applications due to manufacturing difficulties of complicated optimum shapes and topologies. However, we can specify the existing standardised steel cross-sections (e.g. Universal Beams, Universal Columns, Channel and Angle cross-sections etc.,) when constructing the response databases, and the optimisation schemes are set to a discrete form than the traditional continuous forms. Continuous forms lead to better structures in an optimisation sense, but require advanced additive manufacturing technologies to manufacture the optimal shapes which is not yet feasible for medium-to-large scale constructions.

A Appendix

A.1 Finite deformable rod - Detailed formulations

A.1.1 Rotation matrix gradients

Gradient of $\mathbf{\Lambda}_1, \mathbf{\Lambda}_{1,1}$ with respect to axial torsional degrees of freedoms,

$$\frac{\partial^2 \mathbf{\Lambda}_1}{\partial \vartheta_I \partial \vartheta_J} = (-\sin\vartheta \boldsymbol{K}(\boldsymbol{A}_{1u}) + \cos\vartheta \boldsymbol{K}(\boldsymbol{A}_{1u})\boldsymbol{K}(\boldsymbol{A}_{1u}) N^I N^J \tag{A.1}$$

$$\begin{aligned}\frac{\partial \mathbf{\Lambda}_{1,1}}{\partial \vartheta_I} &= \cos\vartheta \boldsymbol{K}_{,1} N^I - \sin\vartheta \boldsymbol{K}\vartheta_{,1} N^I + \cos\vartheta \boldsymbol{K} N^I_{,1} + \sin\vartheta(\boldsymbol{K}\boldsymbol{K}_{,1} + \boldsymbol{K}_{,1}\boldsymbol{K}) N^I \\ &+ \cos\vartheta \boldsymbol{K}\boldsymbol{K}\vartheta_{,1} N^I + \sin\vartheta \boldsymbol{K}\boldsymbol{K}\vartheta_{,1} N^I_{,1}\end{aligned} \tag{A.2}$$

$$\begin{aligned}\frac{\partial^2 \mathbf{\Lambda}_{1,1}}{\partial \vartheta_I \partial \vartheta_J} &= -\sin\vartheta \boldsymbol{K}_{,1} N^I N^J - \cos\vartheta \boldsymbol{K}\vartheta_{,1} N^I N^J \\ &- \sin\vartheta \boldsymbol{K}\vartheta_{,1} N^I N^J_{,1} + (\boldsymbol{K}\boldsymbol{K}_{,1} + \boldsymbol{K}_{,1}\boldsymbol{K}) \cos\vartheta N^I N^J \\ &- \sin\vartheta \boldsymbol{K}\boldsymbol{K}\vartheta_{,1} N^I N^J + \cos\vartheta \boldsymbol{K}\boldsymbol{K} N^I N^J_{,1} + \cos\vartheta \boldsymbol{K}\boldsymbol{K} N^I_{,1} N^J \ .\end{aligned} \tag{A.3}$$

Gradient of $\mathbf{\Lambda}_2$ with respect to displacement degrees of freedom $\boldsymbol{u}_I$,

$$\frac{\partial \mathbf{\Lambda}_2}{\partial \boldsymbol{u}_I} = \frac{\partial \mathbf{\Lambda}_2}{\partial \boldsymbol{a}_{1u}} \frac{\partial \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}_I} \tag{A.4}$$

$$\frac{\partial \mathbf{\Lambda}_2}{\partial \boldsymbol{a}_{1u}} = -\frac{(\boldsymbol{A}_{1u} + \boldsymbol{a}_{1u}) \otimes \mathbf{I}}{(1 + \boldsymbol{a}_{1u} \cdot \boldsymbol{A}_{1u})} - \frac{\mathbf{I} \otimes \boldsymbol{a}_{1u}}{(1 + \boldsymbol{a}_{1u} \cdot \boldsymbol{A}_{1u})} + \frac{((\boldsymbol{A}_{1u} + \boldsymbol{a}_{1u}) \otimes \boldsymbol{A}_{1u}) \otimes \boldsymbol{a}_{1u}}{(1 + \boldsymbol{a}_{1u} \cdot \boldsymbol{A}_{1u})^2} \tag{A.5}$$

$$\frac{\partial \boldsymbol{a}_{1u}}{\partial \boldsymbol{u}_I} = \left(\frac{\mathbf{I}}{||\boldsymbol{a}_1||} - \frac{\boldsymbol{a}_1 \otimes \boldsymbol{a}_1}{||\boldsymbol{a}_1||^3} \right) N^I_{,1} \tag{A.6}$$

A.1.2 Hessian terms

Second variations of strain components as required by Equations (3.92), (3.93), (3.94), (3.95) to compute the stiffness matrix. First taking the second variants with respect to $\boldsymbol{u}$

$$\begin{aligned}
\frac{\partial^2 \alpha}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} &= \frac{\partial}{\partial \boldsymbol{u}_J}\left(\boldsymbol{a}_1 \cdot \frac{\partial \boldsymbol{a}_1}{\partial \boldsymbol{u}_I}\right) \\
&= \frac{\partial \boldsymbol{a}_1}{\partial \boldsymbol{u}_J} \cdot \frac{\partial \boldsymbol{a}_1}{\partial \boldsymbol{u}_I} && \text{(A.7)} \\
\frac{\partial^2 \beta_j}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} &= -\frac{\partial}{\partial \boldsymbol{u}_J}\left(\frac{\partial \beta_j}{\partial \boldsymbol{u}_I}\right) = -\frac{\partial}{\partial \boldsymbol{u}_J}\left(\frac{\partial \boldsymbol{a}_j}{\partial \boldsymbol{u}_I} \cdot \boldsymbol{a}_{1,1} + \boldsymbol{a}_j \cdot \frac{\partial \boldsymbol{a}_{1,1}}{\partial \boldsymbol{u}_I}\right) \\
&= -\frac{\partial \boldsymbol{a}_j}{\partial \boldsymbol{u}_I} \cdot \frac{\partial \boldsymbol{a}_{1,1}}{\partial \boldsymbol{u}_J} - \frac{\partial^2 \boldsymbol{a}_j}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} \cdot \boldsymbol{a}_{1,1} - \frac{\partial \boldsymbol{a}_j}{\partial \boldsymbol{u}_J} \cdot \frac{\partial \boldsymbol{a}_{1,1}}{\partial \boldsymbol{u}_I} \qquad j = 2,3 && \text{(A.8)} \\
\frac{\partial^2 \boldsymbol{a}_j}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} &= \frac{\partial}{\partial \boldsymbol{u}_J}\left(\frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_j\right) \\
&= \frac{\partial^2 \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J}\boldsymbol{\Lambda}_1 \boldsymbol{A}_j \qquad j = 2,3 && \text{(A.9)} \\
\frac{\partial^2 \gamma}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{u}_J}\left(\frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{u}_I} \cdot \boldsymbol{a}_{3,1} + \boldsymbol{a}_2 \cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I}\right) \\
&= \frac{1}{2}\left(\frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{u}_I} \cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_J} + \frac{\partial^2 \boldsymbol{a}_2}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} \cdot \boldsymbol{a}_{3,1} + \frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{u}_J} \cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I} + \boldsymbol{a}_2 \cdot \frac{\partial^2 \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J}\right) && \text{(A.10)} \\
\frac{\partial^2 \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J} &= \frac{\partial}{\partial \boldsymbol{u}_J}\left(\frac{\partial \boldsymbol{\Lambda}_{2,1}}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_3 + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_{1,1} \boldsymbol{A}_3 + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_{3,1}\right) \\
&= \frac{\partial^2 \boldsymbol{\Lambda}_{2,1}}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J}\boldsymbol{\Lambda}_1 \boldsymbol{A}_3 + \frac{\partial^2 \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J}\boldsymbol{\Lambda}_{1,1} \boldsymbol{A}_3 + \frac{\partial^2 \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I \partial \boldsymbol{u}_J}\boldsymbol{\Lambda}_1 \boldsymbol{A}_{3,1} && \text{(A.11)}
\end{aligned}$$

$\boldsymbol{u}$ and ϑ coupled second variants

$$\begin{aligned}
\frac{\partial^2 \beta_j}{\partial \boldsymbol{u}_I \partial \vartheta_J} &= -\frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \beta_j}{\partial \boldsymbol{u}_I}\right) = -\frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \boldsymbol{a}_j}{\partial \boldsymbol{u}_I}\cdot \boldsymbol{a}_{1,1} + \boldsymbol{a}_j \cdot \frac{\partial \boldsymbol{a}_{1,1}}{\partial \boldsymbol{u}_I}\right) \\
&= -\frac{\partial^2 \boldsymbol{a}_j}{\partial \boldsymbol{u}_I \partial \vartheta_J}\cdot \boldsymbol{a}_{1,1} - \frac{\partial \boldsymbol{a}_j}{\partial \vartheta_J}\cdot \frac{\partial \boldsymbol{a}_{1,1}}{\partial \boldsymbol{u}_I} \qquad j = 2,3 \qquad \text{(A.12)} \\
\frac{\partial^2 \boldsymbol{a}_j}{\partial \boldsymbol{u}_I \partial \vartheta_J} &= \frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_j\right) \\
&= \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_J}\boldsymbol{A}_j \qquad j = 2,3 \qquad \text{(A.13)} \\
\frac{\partial^2 \gamma}{\partial \boldsymbol{u}_I \partial \vartheta_J} &= \frac{1}{2}\frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{u}_I}\cdot \boldsymbol{a}_{3,1} + \boldsymbol{a}_2 \cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I}\right) \\
&= \frac{1}{2}\left(\frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{u}_I}\cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \vartheta_J} + \frac{\partial^2 \boldsymbol{a}_2}{\partial \boldsymbol{u}_I \partial \vartheta_J}\cdot \boldsymbol{a}_{3,1} + \frac{\partial \boldsymbol{a}_2}{\partial \vartheta_J}\cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I} + \boldsymbol{a}_2 \cdot \frac{\partial^2 \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I \partial \vartheta_J}\right) \qquad \text{(A.14)} \\
\frac{\partial^2 \boldsymbol{a}_{3,1}}{\partial \boldsymbol{u}_I \partial \vartheta_J} &= \frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \boldsymbol{\Lambda}_{2,1}}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_3 + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_{1,1}\boldsymbol{A}_3 + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\boldsymbol{\Lambda}_1 \boldsymbol{A}_{3,1}\right) \\
&= \frac{\partial \boldsymbol{\Lambda}_{2,1}}{\partial \boldsymbol{u}_I}\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_J}\boldsymbol{A}_3 + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\frac{\partial \boldsymbol{\Lambda}_{1,1}}{\partial \vartheta_J}\boldsymbol{A}_3 + \frac{\partial \boldsymbol{\Lambda}_2}{\partial \boldsymbol{u}_I}\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_J}\boldsymbol{A}_{3,1} \qquad \text{(A.15)}
\end{aligned}$$

second variants with respect to ϑ

$$\begin{aligned}
\frac{\partial^2 \beta_j}{\partial \vartheta_I \partial \vartheta_J} &= -\frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \beta_j}{\partial \vartheta_I}\right) = -\frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \boldsymbol{a}_j}{\partial \vartheta_I}\cdot \boldsymbol{a}_{1,1}\right) \\
&= -\frac{\partial^2 \boldsymbol{a}_j}{\partial \vartheta_I \partial \vartheta_J}\cdot \boldsymbol{a}_{1,1} \qquad \text{(A.16)} \\
\frac{\partial^2 \boldsymbol{a}_j}{\partial \vartheta_I \partial \vartheta_J} &= \frac{\partial}{\partial \vartheta_J}\left(\boldsymbol{\Lambda}_2 \frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_I}\boldsymbol{A}_j\right) = \boldsymbol{\Lambda}_2 \frac{\partial^2 \boldsymbol{\Lambda}_1}{\partial \vartheta_I \partial \vartheta_J}\boldsymbol{A}_j \qquad j = 2,3 \\
\frac{\partial^2 \gamma}{\partial \vartheta_I \partial \vartheta_J} &= \frac{1}{2}\frac{\partial}{\partial \vartheta_J}\left(\frac{\partial \boldsymbol{a}_2}{\partial \vartheta_I}\cdot \boldsymbol{a}_{3,1} + \boldsymbol{a}_2 \cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \vartheta_I}\right) \\
&= \frac{1}{2}\left(\frac{\partial \boldsymbol{a}_2}{\partial \vartheta_I}\cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \vartheta_J} + \frac{\partial^2 \boldsymbol{a}_2}{\partial \vartheta_I \partial \vartheta_J}\cdot \boldsymbol{a}_{3,1} + \frac{\partial \boldsymbol{a}_2}{\partial \vartheta_J}\cdot \frac{\partial \boldsymbol{a}_{3,1}}{\partial \vartheta_I} + \boldsymbol{a}_2 \cdot \frac{\partial^2 \boldsymbol{a}_{3,1}}{\partial \vartheta_I \partial \vartheta_J}\right) \qquad \text{(A.17)} \\
\frac{\partial^2 \boldsymbol{a}_{3,1}}{\partial \vartheta_I \partial \vartheta_J} &= \frac{\partial}{\partial \vartheta_J}\left(\boldsymbol{\Lambda}_{2,1}\frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_I}\boldsymbol{A}_3 + \boldsymbol{\Lambda}_2 \frac{\partial \boldsymbol{\Lambda}_{1,1}}{\partial \vartheta_I}\boldsymbol{A}_3 + \boldsymbol{\Lambda}_2 \frac{\partial \boldsymbol{\Lambda}_1}{\partial \vartheta_I}\boldsymbol{A}_{3,1}\right) \\
&= \boldsymbol{\Lambda}_{2,1}\frac{\partial^2 \boldsymbol{\Lambda}_1}{\partial \vartheta_I \partial \vartheta_J}\boldsymbol{A}_3 + \boldsymbol{\Lambda}_2 \frac{\partial^2 \boldsymbol{\Lambda}_{1,1}}{\partial \vartheta_I \partial \vartheta_J}\boldsymbol{A}_3 + \boldsymbol{\Lambda}_2 \frac{\partial^2 \boldsymbol{\Lambda}_1}{\partial \vartheta_I \partial \vartheta_J}\boldsymbol{A}_{3,1} \qquad \text{(A.18)}
\end{aligned}$$

A.2 GPR model training and testing results

Presented here are the tabulated results of training and testing of the GPR models for plain woven and weft-knitted fabrics. We restate the following abbreviations.

R^2 score - Coefficient of determination
MSE - Mean Squared Error
MLML - Maximum Log Marginal Likelihood

A.2.1 Influence of covariance functions on predictions

Table A.1: GPR model training and testing results for different covariance functions

Covariance function	Data points	Model training		Model testing (on unseen data)			
		R^2	MSE ($\times 10^{-10}$)	R^2	MSE ($\times 10^{-8}$)	Optimised $\boldsymbol{\Theta}$ $\{\Theta_1, \Theta_2, \cdots\}$	MLML
RQ	36	1.0	114.38	1.0	5030.0	37.13, 0.0177, 0.989	525.801
	121	1.0	1.971	1.0	3.472	23.63, 0.00121, 1.262	2022.679
	441	1.0	5.082	1.0	4.455	8.82, 0.00255, 0.556	4715.243
	961	1.0	5.655	1.0	7.899	11.16, 0.00024, 0.053	13784.370
	2601	1.0	15.238	1.0	0.163	8.77, 0.00012, 0.658	57351.711
SE	36	1.0	33.352	1.0	5826.5	7.87, 0.0785	433.224
	121	1.0	0.639	1.0	3.698	2.17, 0.0288	1344.696
	441	1.0	0.567	1.0	8.708	1.51, 0.0192	3315.804
	961	1.0	0.958	1.0	3.340	0.162, 0.0545	-8920.712
	2601	1.0	0.492	1.0	0.144	0.161, 0.0545	12646.968
Exp-Sine	36	1.0	792.12	1.0	5936.7	8.41, 0.612, 1.0	452.758
	121	1.0	2.069	1.0	8.631	1.65, 0.138, 1.0	1377.407
	441	1.0	1.069	1.0	3.512	4.89, 0.323, 1.0	3034.238
	961	1.0	11.468	1.0	5.987	19.35, 0.144, 2.91	6822.816
	2601	1.0	14.481	1.0	1.693	6.84, 0.124, 3.08	29254.113
Matérn	36	1.0	139.26	1.0	16211.8	119.34, 3.37, 2.50	51.883
	121	1.0	13.188	1.0	6.577	107.45, 2.43, 2.50	2652.891
	441	1.0	0.238	1.0	0.397	95.72, 2.14, 2.50	6519.493
	961	1.0	0.416	1.0	0.570	41.36, 1.01, 2.50	18264.605
	2601	1.0	0.271	1.0	0.303	11.12, 0.303, 2.50	72549.535

where $\boldsymbol{\Theta}_{\text{RQ}} = \{\sigma_f, \alpha, \ell\}$, $\boldsymbol{\Theta}_{\text{SE}} = \{\sigma_f, \ell\}$, $\boldsymbol{\Theta}_{\text{Exp-Sine}} = \{\sigma_f, \ell, \text{periodicity}\}$, $\boldsymbol{\Theta}_{\text{Matérn}} = \{\sigma_f, \ell, \nu = 2.5\}$.

A.2.2 Plain woven fabrics

Table A.2: $5-fold$ data training and testing results for multiscale modelling of plane-woven fabrics using Gaussian Process Regression.

Data points	Trial #	Model training		Model testing (on unseen data)				
		R^2	MSE ($\times 10^{-11}$)	R^2	MSE ($\times 10^{-5}$)	Optimised $\boldsymbol{\Theta}$		MLML
						σ_f	ℓ	
36	1	1.0	0.032	1.0	81.42	62.34	0.172	-27.265
	2	1.0	0.007	1.0	67.23	59.21	0.171	-29.481
	3	1.0	0.216	1.0	84.29	68.13	0.173	-24.147
	4	1.0	0.167	1.0	74.17	65.36	0.173	-25.471
	5	1.0	0.037	1.0	99.34	57.98	0.168	-34.725
121	1	1.0	0.544	1.0	5.23	23.64	0.0677	72.986
	2	1.0	0.463	1.0	8.62	27.41	0.0684	81.326
	3	1.0	0.753	1.0	2.81	19.36	0.0648	68.264
	4	1.0	0.549	1.0	3.45	26.11	0.0680	78.399
	5	1.0	0.742	1.0	1.53	25.26	0.0681	76.341
441	1	1.0	5.541	1.0	1.44	12.52	0.0298	465.997
	2	1.0	5.302	1.0	3.56	14.92	0.0304	513.362
	3	1.0	7.461	1.0	2.72	11.32	0.0296	431.732
	4	1.0	2.661	1.0	0.99	12.53	0.0298	469.873
	5	1.0	3.767	1.0	0.48	8.23	0.0284	389.232
1296	1	1.0	4.442	1.0	1.21	8.96	0.0148	1488.52
	2	1.0	3.093	1.0	2.47	7.45	0.0142	1387.37
	3	1.0	2.893	1.0	0.71	7.12	0.0140	1299.15
	4	1.0	1.251	1.0	0.81	8.08	0.0144	1411.57
	5	1.0	1.892	1.0	0.35	8.38	0.0146	1432.59
2601	1	1.0	2.493	1.0	1.15	8.12	0.0105	3011.58
	2	1.0	5.614	1.0	1.03	7.96	0.0101	3110.44
	3	1.0	4.633	1.0	1.85	8.05	0.0103	3063.82
	4	1.0	2.122	1.0	0.62	7.92	0.0099	3102.93
	5	1.0	2.452	1.0	0.46	8.21	0.0110	3183.52

A.2.3 Weft-knitted fabrics

Table A.3: $5-fold$ data training and testing results for multiscale modelling of weft-knitted fabrics using Gaussian process regression.

Data points	Trial #	Model training		Model testing (on unseen data)				
		R^2	MSE ($\times 10^{-10}$)	R^2	MSE ($\times 10^{-8}$)	Optimised $\boldsymbol{\Theta}$		MLML
						σ_f	ℓ	
36	1	1.0	0.202	1.0	6532.24	19.23	0.172	37.173
	2	1.0	0.223	1.0	6485.43	19.21	0.171	36.481
	3	1.0	0.256	1.0	6783.94	18.59	0.169	34.271
	4	1.0	0.197	1.0	5768.74	20.06	0.174	39.761
	5	1.0	0.237	1.0	6138.49	17.98	0.166	31.946
121	1	1.0	0.594	1.0	2.15	6.99	0.0723	344.74
	2	1.0	0.783	1.0	2.27	7.09	0.0725	356.12
	3	1.0	0.713	1.0	2.06	6.89	0.0722	339.82
	4	1.0	0.635	1.0	1.92	6.71	0.0716	321.86
	5	1.0	0.689	1.0	1.83	7.12	0.0725	362.83
441	1	1.0	4.501	1.0	1.48	2.17	0.0292	1322.73
	2	1.0	4.682	1.0	1.45	2.09	0.0289	1284.82
	3	1.0	4.418	1.0	1.35	2.13	0.0290	1298.72
	4	1.0	4.531	1.0	1.39	2.31	0.0295	1361.82
	5	1.0	4.877	1.0	1.34	2.23	0.0293	1357.22
961	1	1.0	1.542	1.0	2.41	1.28	0.0142	2899.49
	2	1.0	1.354	1.0	2.31	1.25	0.0142	2868.41
	3	1.0	1.683	1.0	2.38	1.16	0.0139	2793.85
	4	1.0	1.461	1.0	2.45	1.31	0.0143	2904.62
	5	1.0	1.498	1.0	2.40	1.39	0.0146	2984.91
2601	1	1.0	2.193	1.0	0.96	1.05	0.00899	9407.88
	2	1.0	1.756	1.0	1.07	1.13	0.00901	9531.86
	3	1.0	1.493	1.0	1.12	1.25	0.00904	9594.98
	4	1.0	1.892	1.0	0.92	0.92	0.00889	9291.49
	5	1.0	1.782	1.0	0.98	1.19	0.00902	9552.66

Bibliography

[1] Oliver Weeger, Amir Hosein Sakhaei, Ying Yi Tan, Yu Han Quek, Tat Lin Lee, Sai-Kit Yeung, Sawako Kaijima, and Martin L Dunn. Nonlinear Multi-Scale Modelling, Simulation and Validation of 3D Knitted Textiles. *Applied Composite Materials*, pages 1–14, may 2018. 1, 2, 13, 38, 45, 108, 137, 166, 169, 170, 171

[2] Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A Otaduy. Yarn-level Simulation of Woven Cloth. *ACM Trans. Graph.*, 33(6):207:1–207:11, 2014. 1, 2, 4, 147

[3] Gaurav Nilakantan, Michael Keefe, Travis A Bogetti, Rob Adkinson, and John W Gillespie. On the finite element analysis of woven fabric impact using multiscale modeling techniques. *International Journal of Solids and Structures*, 47(17):2300–2315, 2010. 1

[4] M J King, P Jearanaisilawong, and S Socrate. A continuum constitutive model for the mechanical behavior of woven fabrics. *International Journal of Solids and Structures*, 42(13):3867–3896, 2005. 1

[5] Cuong Ha-Minh, Toufik Kanit, François Boussu, and Abdellatif Imad. Numerical multi-scale modeling for textile woven fabric against ballistic impact. *Computational Materials Science*, 50(7):2172–2184, 2011. 1

[6] David Clyde, Joseph Teran, and Rasmus Tamstorf. Modeling and data-driven parameter estimation for woven fabrics. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation - SCA '17*, pages 1–11, New York, New York, USA, 2017. ACM Press. 1

[7] T C Lim, S Ramakrishna, and H M Shang. Axisymmetric sheet forming of knitted fabric composite by combined stretch forming and deep drawing. *Composites Part B: Engineering*, 30(5), 1999. 1

[8] Hossein Hasani, Sanaz Hassanzadeh, Mohammad Javad Abghary, and Elahe Omrani. Biaxial weft-knitted fabrics as composite reinforcements: A review. *Journal of Industrial Textiles*, 46(7):1439–1473, 2017. 1

[9] Mars Exploration Rovers NASA Parachutes. `https://mars.nasa.gov/resources/8833/mars-exploration-rovers-parachute-test/`. 1, 3

[10] Thin-Shell Concrete fabric molds. `http://fabwiki.fabric-formedconcrete.com/lib/exe/fetch.php?media=chile:thin_shell_panels.pdf`. 2, 3

[11] Digital Smart textiles - Apparel industry. `https://www.dezeen.com/2014/08/29/movie-francesca-rosella-cutecircuit-digital-fashion-smart-textiles`. 2, 3

[12] Aran biomedical - Implantable medical devices. `https://www.aranbiomedical.com/medical-textiles/#%20braided_Medical_Fabrics`. 2, 3

[13] Ruwan Sumanasinghe and Martin King. New trends in biotextiles - the challenge of tissue engineering. *J Text Apparel Technol Manage*, 3, 01 2003. 2, 3

[14] Continental Protective Equipment Materials. `https://www.contitech.de/en-gl/Solutions/Surface-Materials/Protection-Health/Protective-Equipment-Materials`. 2, 3

[15] Ultra Reliable Bullet Proof Vest Protection. `https://www.safeguardclothing.com/15-bullet-proof-vests`. 2, 3

[16] Snarkitecture created pop-up clouded-fabric for Valextra's store in Milan. `https://worldarchitecture.org/architecture-news/cvcze/snarkitecture_created_popup_cloudedfabric_for_valextra_s_store_in_milan.html`. 2, 3

[17] Jonathan M Kaldor, Doug L James, Steve Marschner, Jonathan M Kaldor, Doug L James, and Steve Marschner. Simulating knitted cloth at the yarn level. In *ACM SIGGRAPH 2008 papers on - SIGGRAPH '08*, volume 27, page 1, New York, New York, USA, 2008. ACM Press. 2, 45, 147

[18] Ben Nadler, Panayiotis Papadopoulos, and David J. Steigmann. Multiscale constitutive modeling and numerical simulation of fabric material. *International Journal of Solids and Structures*, 43(2):206–221, jan 2006. 2, 37, 38, 45, 105, 107, 108, 132, 137, 139, 140, 141, 144, 146, 160

[19] Ben Nadler and David J Steigmann. A model for frictional slip in woven fabrics. *Elsevier SAS*, 331:797–804, 2003. 2, 45

[20] T D Dinh, O Weeger, S Kaijima, and S K Yeung. Prediction of mechanical properties of knitted fabrics under tensile and shear loading: Mesoscale analysis using representative unit cells and its validation. *Composites Part B: Engineering*, 148:81–92, sep 2018. 2, 38, 45, 108, 137, 166, 170, 171

[21] E Miguel, D Bradley, B Thomaszewski, B Bickel, W Matusik, M A Otaduy, and S. Marschner. Data-Driven Estimation of Cloth Simulation Models. *Computer Graphics Forum*, 31(2pt2):519–528, may 2012. 2, 4, 108

[22] Bengt Behre. Mechanical properties of textile fabrics part i: Shearing. *Textile Research Journal*, 31(2):87–93, 1961. 4

[23] P Grosberg and S Kedia. The mechanical properties of woven fabrics: Part i: The initial load extension modulus of woven fabrics. *Textile Research Journal*, 36(1):71–79, 1966. 4

[24] J W S Hearle, P Grosberg, and S Backer. Structural mechanics of fibers, yarns, and fabrics, 1969. 4

[25] John WS Hearle. Mechanics of flexible fibre assemblies. *Proceedings of the NATO Advanced Study Institute on Mechanics of Flexible Fibre Assemblies*, 1980. 4

[26] Nhan G. Ly. A model for fabric buckling in shear. *Textile Research Journal*, 55(12):744–749, 1985. 4

[27] Ron Postle, Garth Alan Carnaby, and Stuart De Jong. The mechanics of wool structures. 1988. 4

[28] Simona Jevšnik, Fatma Kalaoğlu, Sena Terliksiz, and Jure Purgaj. Review of Computer Models for Fabric Simulation. 2014. 4

[29] Warren S Seames. *Computer numerical control: concepts and programming*. Cengage Learning, 2001. 4

[30] Mark Beecroft. Digital interlooping: 3d printing of weft-knitted textile-based tubular structures using selective laser sintering of nylon powder. *International Journal of Fashion Design, Technology and Education*, 12(2):218–224, 2019. 5, 6

[31] Shajay Bhooshan and Mostafa El Sayed. Sub-division surfaces in architectural form finding and fabric forming. In *Proceedings of the Second International Conference of Flexible Formwork (ICFF 2012), BRE CICM, University of Bath, Claverton Down*, 2012. 6

[32] Felecia Davis. 3D Printed Textiles from Textile Code: Structural Form and Material Operations. *SIGraDi 2012 [Proceedings of the 16th Iberoamerican Congress of Digital Graphics] Brasil - Fortaleza 13-16 November 2012, pp. 327-331*, 2012. 6

[33] G A Bingham, R J M Hague, C J Tuck, A C Long, J J Crookston, and M N Sherburn. Rapid manufactured textiles. *International Journal of Computer Integrated Manufacturing*, 20(1):96–105, 2007. 6

[34] A R Damanpack, M Bodaghi, and W H Liao. Experimentally validated multi-scale modeling of 3D printed hyper-elastic lattices. *International Journal of Non-Linear Mechanics*, 108:87–110, jan 2019. 6, 107

[35] Tobias Maconachie, Martin Leary, Bill Lozanovski, Xuezhe Zhang, Ma Qian, Omar Faruque, and Milan Brandt. Slm lattice structures: Properties, performance, applications and challenges. *Materials and Design*, 183:108137, 2019. 6

[36] 3D printed lattice structures. `https://www.sculpteo.com/blog/2018/04/18/3d-printing-lattices-find-the-best-lattice-generation-tools/`. 7

[37] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006. 7, 8, 110, 112, 120, 160

[38] M G D Geers, V G Kouznetsova, and W A M Brekelmans. Computational homogenization. pages 327–394. Springer, Vienna, 2010. 8, 100, 102, 104, 105, 106, 107, 141, 147

[39] Marc G D Geers, Varvara G Kouznetsova, Karel Matouš, and Julien Yvonnet. Homogenization Methods and Multiscale Modeling: Nonlinear Problems. In *Encyclopedia of Computational Mechanics Second Edition*, pages 1–34. John Wiley & Sons, Ltd, Chichester, UK, dec 2017. 2, 8, 100, 101, 102, 103, 105, 106, 107, 108

[40] Bassam El Said, Federica Daghia, Dmitry Ivanov, and Stephen R. Hallett. An iterative multiscale modelling approach for nonlinear analysis of 3d composites. *International Journal of Solids and Structures*, 132-133:42 – 58, 2018. 8

[41] S Rogers and M Girolami. *A First Course in Machine Learning*. Chapman & Hall. Taylor & Francis, 2011. 8, 110, 128

[42] Philippe G Ciarlet. *An Introduction to Differential Geometry with Applications to Elasticity*. City University of Hong Kong, 2005. 8, 50, 78, 81, 86

[43] M P do Carmo. *Differential geometry of curves and surfaces*. Prentice-Hall, 1976. 8

[44] M Majeed and F Cirak. Isogeometric analysis using manifold-based smooth basis functions. *Computer Methods in Applied Mechanics and Engineering*, accepted, 2016. 8, 13, 24, 32

[45] Musabir Majeed. *Isogeometric analysis using manifolds and immersed method*. PhD thesis, Cambridge, 2016. 8, 13, 19, 23, 24, 25, 26, 27, 32, 78, 88

[46] D Millán, A Rosolen, and M Arroyo. Nonlinear manifold learning for meshfree finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*, 93(7):685–713, 2012. 8

[47] Qiaoling Zhang and Fehmi Cirak. Manifold-based isogeometric analysis basis functions with prescribed sharp features, 2019. 8

[48] Xizhi Sun and Kai Cheng. Chapter 2 - micro-/nano-machining through mechanical cutting. In Yi Qin, editor, *Micromanufacturing Engineering and Technology (Second Edition)*, Micro and Nano Technologies, pages 35 – 59. William Andrew Publishing, Boston, second edition edition, 2015.

[49] Tzeng Yih-fong. Parameter design optimisation of computerised numerical control turning tool steels for high dimensional precision and accuracy. *Materials & Design*, 27(8):665 – 675, 2006.

[50] J C Simo and L Vu-Quoc. A three-dimensional finite-strain rod model. part II: Computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 58(1):79–116, oct 1986. 9, 35, 37, 43, 65, 66, 70

[51] Christoph Meier, Alexander Popp, and Wolfgang A Wall. An objective 3D large deformation finite element formulation for geometrically exact curved Kirchhoff rods. *Computer Methods in Applied Mechanics and Engineering*, 278:445–478, aug 2014. 9, 35, 36, 37, 44, 45, 65

[52] A M Bauer, M Breitenberger, B Philipp, R. Wüchner, and K U Bletzinger. Nonlinear isogeometric spatial Bernoulli beam. *Computer Methods in Applied Mechanics and Engineering*, 303:101–127, may 2016. 9, 35, 36, 37, 38, 45, 49, 65, 66, 67, 68, 69, 185

[53] F Cirak, M Ortiz, and P Schroder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47(12):2039–2072, 2000. 78, 87

[54] Fehmi Cirak, Michael J Scott, Erik K Antonsson, Michael Ortiz, and Peter Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34(2):137–148, 2002. 13

[55] Fehmi Cirak and Michael Ortiz. Fully C1-conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*, 51(7):813–833, 2001. 86, 87

[56] Quan Long, P Burkhard Bornemann, and Fehmi Cirak. Shear-flexible subdivision shells. *International Journal for Numerical Methods in Engineering*, 90(13):1549–1577, jun 2012. 77, 78, 87

[57] H M Yasitha Chinthaka Mallikarachchi. *Thin-walled composite deployable booms with tape-spring hinges*. Thesis, University of Cambridge, jul 2011. 9, 107, 108

[58] Ahmad Kueh and Sergio Pellegrino. ABD Matrix of Single-Ply Triaxial Weave Fabric Composites. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Reston, Virigina, apr 2007. American Institute of Aeronautics and Astronautics. 9, 107, 108, 138

[59] Herath Sumudu and Mallikarachchi Chinthaka. Modified Ply Thickness for Classical Lamination Theory for Thin Woven Fibre Composites — Request PDF. 9, 107, 108

[60] 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, 2015. 9

[61] M A Bessa, R Bostanabad, Z Liu, A Hu, Daniel W Apley, C Brinson, W Chen, and Wing Kam 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, jun 2017. 9, 108, 112, 122, 125, 156

[62] T J R Hughes, J A Cottrell, and Y Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.*, 194:4135–4195, 2005. 13, 19

[63] Kosala Bandara and Fehmi Cirak. Isogeometric shape optimisation of shell structures using multiresolution subdivision surfaces. *CAD Computer Aided Design*, 95, 2018. 13

[64] Q Long. *Subdivision finite elements for geometrically complex thin and thick shells.* PhD thesis, 2009. 13, 24, 54, 77, 88

[65] K Hollig and J Horner. *Approximation and Modeling with B-Splines.* Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2015. 13, 14, 15

[66] G Farin, J Hoschek, and M S Kim. *Handbook of computer aided geometric design.* Elsevier, 2002. 13, 15, 16, 18, 87

[67] W Böhm, G Farin, and J Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1(1):1–60, 1984. 13

[68] L E Andersson and N F Stewart. *Introduction to the mathematics of subdivision surfaces.* SIAM, 2010. 14, 15

[69] L Ramshaw. Blossoms are polar forms. *Computer Aided Geometric Design*, 6(4):323–358, 1989. 15

[70] P Bornemann and F Cirak. A subdivision-based implementation of the hierarchical b-spline finite element method. *Computer Methods in Applied Mechanics and Engineering*, 253:584–598, 2013. 15, 19

[71] J N Reddy. *An introduction to the finite element method.* McGraw-Hill, Department of Mechanical Engineering, Texas A&M University, College Station, Texas, USA, third edition edition, 2005. 15, 27, 36, 84, 132

[72] K Höllig. *Finite element methods with b-splines.* SIAM, 2003. 15

[73] J A Cottrell, T J R Hughes, and Y Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009. 16, 19, 24, 38

[74] R Sanches, P Bornemann, and F Cirak. Immersed b-spline (i-spline) finite element method for geometrically complex domains. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1432–1445, 2011. 18

[75] D Zorin and P Schröder. Subdivision for modeling and animation. *ACM SIGGRAPH Course Notes*, 12, 1998. 19

[76] E Catmull and J Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978. 19

[77] D Doo and M Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, 1978. 19

[78] C Giannelli, B Jüttler, and H Speleers. THB-splines: the truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012. 19

[79] T Rüberg and F Cirak. Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209:266–283, 2012. 19

[80] F P Miller, A F Vandome, and J McBrewster. *Frenet-Serret Formulas: Vector Calculus, Curve, Derivative, Euclidean Space, Kinematics, Darboux Frame, Differential Geometry of Curves, Affine Geometry of Curves*. Alphascript Publishing, 2010. 20, 21, 45

[81] Christoph Meier, Alexander Popp, and Wolfgang A Wall. A Finite Element Approach for the Line-to-Line Contact Interaction of Thin Beams with Arbitrary Orientation. mar 2016. 21, 35, 44, 45, 59, 62, 75

[82] C Grimm. *Modeling surfaces of arbitrary topology using manifolds*. PhD thesis, Brown University, 1996. 24

[83] Cindy Grimm and Denis Zorin. Surface modeling and parameterization with manifolds: Siggraph 2006 course notesauthor presenation videos are available from the citation page. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, pages 1–81, New York, NY, USA, 2006. ACM. 25

[84] Origami-inspired sheltering structures. https://motivistjapan.com/2015/12/origami-and-architecture/. 29

[85] Three-hinged bridge strcutures. https://structurae.net/en/photos/147828-salginatobel-bridge. 29

[86] Architectural forms of pavilion. https://www.archdaily.com/781664/origami-pavilion-creates-shelter-with-just-3-folded-aluminum-sheets/. 29

[87] Q Z Xiao and B L Karihaloo. Improving the accuracy of XFEM crack tip fields using higher order quadrature and statically admissible stress recovery. *International Journal for Numerical Methods in Engineering*, 66(9):1378–1410, May 2006. 29

[88] Erasmo Carrera, Gaetano Giunta, and Marco Petrolo. *Beam structures : classical and advanced theories*. Wiley, 2011. 33, 36

[89] J C Simo. A finite strain beam formulation. the three-dimensional dynamic problem. part i. *Computer Methods in Applied Mechanics and Engineering*, 49(1):55 – 70, 1985. 35, 36, 43

[90] J.C. Simo and L. Vu-Quoc. On the dynamics in space of rods undergoing large motions — a geometrically exact approach. *Computer Methods in Applied Mechanics and Engineering*, 66(2):125 – 161, 1988. 35

[91] A Cardona and M Geradin. A beam finite element non-linear theory with finite rotations. *International Journal for Numerical Methods in Engineering*, 26(11):2403–2438, 1988. 35, 43

[92] K Kondoh, K Tanaka, and S N Atluri. An explicit expression for the tangent-stiffness of a finitely deformed 3-d beam and its use in the analysis of space frames. *Computers & Structures*, 24(2):253 – 271, 1986. 35

[93] Eduardo N Dvorkin, Eugenio Onte, and Javier Oliver. On a non-linear formulation for curved timoshenko beam elements considering large displacement/rotation increments. *International Journal for Numerical Methods in Engineering*, 26(7):1597–1613, 1988. 35

[94] G Jelenić and M A Crisfield. Interpolation of rotational variables in nonlinear dynamics of 3d beams. *International Journal for Numerical Methods in Engineering*, 43(7):1193–1222, 1998. 35

[95] Christoph Meier, Maximilian J Grill, Wolfgang A Wall, and Alexander Popp. Geometrically exact beam elements and smooth contact schemes for the modeling of fiber-based materials and structures. nov 2016. 35, 45

[96] Christoph Meier, Alexander Popp, and Wolfgang A Wall. Geometrically Exact Finite Element Formulations for Slender Beams: Kirchhoff–Love Theory Versus Simo–Reissner Theory. *Archives of Computational Methods in Engineering*, pages 1–81, jul 2017. 35

[97] Vikram S Deshpande. *Module 3D7 : Finite Element Methods Part I : One-dimensional formulation*. Number January. 2014. 36

[98] A F Bower. *Applied mechanics of solids*. CRC press, 2009. 36

[99] Oliver Weeger, Utz Wever, and Bernd Simeon. Isogeometric analysis of nonlinear Euler–Bernoulli beam vibrations. *Nonlinear Dynamics*, 72(4):813–835, jun 2013. 36

[100] S.H. Lo. Geometrically nonlinear formulation of 3d finite strain beam element with large rotations. *Computers & Structures*, 44(1):147 – 157, 1992. Special Issue: WCCM II. 36, 66, 69

[101] A locking-free finite element formulation and reduced models for geometrically exact Kirchhoff rods. *Computer Methods in Applied Mechanics and Engineering*, 290:314–341, jun 2015. 37

[102] I. Romero, M. Urrecha, and C.J. Cyron. A torsion-free non-linear beam model. *International Journal of Non-Linear Mechanics*, 58:1 – 10, 2014. 37

[103] William E Warren. The elastic properties of woven polymeric fabric. *Polymer Engineering & Science*, 30(20):1309–1313, 1990. 37, 139, 144

[104] Grzegorz Orzechowski and Janusz Fraczek. Nearly incompressible nonlinear material models in the large deformation analysis of beams using ancf. *Nonlinear Dynamics*, 82(1):451–464, Oct 2015. 37

[105] *Use of General Nonlinear Material Models in Beam Problems: Application to Belt and Rubber Chains*, volume Volume 4: 7th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B and C of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2009. 37

[106] R A Hawileh, A Rahman, and H Tabatabai. Nonlinear finite element analysis and modeling of a precast hybrid beam–column connection subjected to cyclic loads. *Applied Mathematical Modelling*, 34(9):2562 – 2583, 2010. 37

[107] Oliver Weeger, Bharath Narayanan, Laura De Lorenzis, Josef Kiendl, and Martin L Dunn. An isogeometric collocation method for frictionless contact of Cosserat rods. *Computer Methods in Applied Mechanics and Engineering*, 321:361–382, jul 2017. 38, 59, 73

[108] John Argyris. An excursion into large rotations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):85 – 155, 1982. 43

[109] M Ritto-Correa and D Camotim. On the differentiation of the Rodrigues formula and its significance for the vector-like parameterization of Reissner–Simo beam theory. *International Journal for Numerical Methods in Engineering*, 55(9):1005–1032, 2002. 43

[110] M A Crisfield. *Non-linear finite element analysis of solids and structures*. Number v. 2 in Non-linear Finite Element Analysis of Solids and Structures. Wiley, 1997. 44

[111] Christoph Meier, Wolfgang A Wall, and Alexander Popp. A unified approach for beam-to-beam contact. *Computer Methods in Applied Mechanics and Engineering*, 315:972–1010, 2017. 45, 59, 61

[112] Dani Liu, Daniel Christe, Bahareh Shakibajahromi, Chelsea Knittel, Nestor Castaneda, David Breen, Genevieve Dion, and Antonios Kontsos. On the role of material architecture in the mechanical behavior of knitted textiles. *International Journal of Solids and Structures*, 109:101–111, mar 2017. 45

[113] Mark S Yeoman, Daya Reddy, Hellmut C Bowles, Deon Bezuidenhout, Peter Zilla, and Thomas Franz. A constitutive model for the warp-weft coupled non-linear

behavior of knitted biomedical textiles. *Biomaterials*, 31(32):8484–8493, nov 2010. 45

[114] Savvas G Vassiliadis, Argyro E Kallivretaki, and Christopher G Provatidis. Mechanical simulation of the plain weft knitted fabrics. *International Journal of Clothing Science and Technology*, 19(2):109–130, mar 2007. 45, 108, 166, 167, 169, 170, 171

[115] Savvas Vassiliadis, Argyro Kallivretaki, and Christopher Provatidis. Geometrical modelling of plain weft knitted fabrics. *Indian Journal of Fibre and Textile Research*, 32:62–71, mar 2007. 45, 166, 168

[116] Gerhard A Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. John Wiley & Sons, 2000. 47, 82, 103

[117] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006. 48, 56

[118] S Boyd and L Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. 48, 56

[119] P Wriggers. *Nonlinear finite element methods*. Springer Science & Business Media, 2008. 50, 78

[120] Bonet Javier and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, second edition edition, 1997. 55, 103

[121] P Wriggers and G Zavarise. On contact between three-dimensional beams undergoing large deflections. *Communications in Numerical Methods in Engineering*, 13(6):429–438, jun 1997. 59

[122] G Zavarise and P Wriggers. Contact with friction between beams in 3-D space. *International Journal for Numerical Methods in Engineering*, 49(8):977–1006, nov 2000. 59

[123] Fu-Jun Wang, Li-Ping Wang, Jian-Gang Cheng, and Zhen-Han Yao. Contact force algorithm in explicit transient analysis using finite-element method. *Finite Elements in Analysis and Design*, 43(6-7):580–587, apr 2007. 59

[124] Fehmi Cirak and Matthew West. Decomposition contact response (DCR) for explicit finite element dynamics. *International Journal for Numerical Methods in Engineering*, 64(8):1078–1110, oct 2005. 59

[125] Martin W Heinstein, Frank J Mello, Stephen W Attaway, and Tod A Laursen. Contact—impact modeling in explicit transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, 187(3-4):621–640, jul 2000. 59

[126] Damien Durville. Contact-friction modeling within elastic beam assemblies: an application to knot tightening. *Computational Mechanics*, 49(6):687–707, jun 2012. 59

[127] Ahmed K Noor and Jeanne M Peters. Mixed models and reduced/selective integration displacement models for nonlinear analysis of curved beams. *International Journal for Numerical Methods in Engineering*, 17(4):615–631, 1981. 65, 70

[128] D A DaDeppo and R Schmidt. Instability of Clamped-Hinged Circular Arches Subjected to a Point Load. *Journal of Applied Mechanics*, 42(4):894–896, 12 1975. 65

[129] Klaus-Jürgen Bathe and Said Bolourchi. *Large Displacement Analysis of three-dimensional beam structures*. 1979. 66, 69

[130] Ignacio Romero. A comparison of finite elements for nonlinear beams: the absolute nodal coordinate and geometrically exact formulations. *Multibody System Dynamics*, 20(1):51–68, Aug 2008. 66

[131] Steen Krenk. *Non-linear Modeling and Analysis of Solids and Structures*. Cambridge University Press, 2009. 67

[132] Chapter 6 - rate of convergence of the finite element method. In A K Aziz, editor, *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, pages 185 – 241. Academic Press, 1972. 72

[133] J N Reddy. *Theory and Analysis of Elastic Plates and Shells*. Series in Systems and Control. CRC Press, 2006. 77, 78

[134] S P Timoshenko and S Woinowsky-Krieger. *Theory of plates and shells*. McGraw-Hill, 1993. 77, 78

[135] Gilbert Strang. *Introduction to Linear Algebra*. SIAM, 2003. 85

[136] R M Jones. *Mechanics Of Composite Materials*. Materials Science and Engineering Series. Taylor & Francis, 1998. 84, 86, 90, 92

[137] M Itskov. *Tensor Algebra and Tensor Analysis for Engineers: With Applications to Continuum Mechanics*. Springer Berlin Heidelberg, 2009. 86, 89

[138] F Cirak and Q Long. Subdivision shells with exact boundary control and non-manifold geometry. *International Journal for Numerical Methods in Engineering*, 88(9):897–923, 2011. 87

[139] Mikhail Itskov and Nuri Aksel. A class of orthotropic and transversely isotropic hyperelastic constitutive models based on a polyconvex strain energy function. *International Journal of Solids and Structures*, 41(14):3833–3848, jul 2004. 89

[140] Y Başar, M Itskov, and A Eckstein. Composite laminates: nonlinear interlaminar stress analysis by multi-layer shell elements. *Computer Methods in Applied Mechanics and Engineering*, 185(2):367–397, 2000. 89

[141] A J M Spencer. *Continuum Theory of the Mechanics of Fibre-Reinforced Composites*. CISM International Centre for Mechanical Sciences. Springer Vienna, 2014. 91

[142] A L Dobyns. Analysis of Simply-Supported Orthotropic Plates Subject to Static and Dynamic Loads. *AIAA Journal*, 19(5):642–650, may 1981. 92, 93

[143] J M Whitney and N J Pagano. Shear Deformation in Heterogeneous Anisotropic Plates. *Journal of Applied Mechanics*, 37(4):1031–1036, 12 1970. 93

[144] V G Kouznetsova, M G D Geers, and W A M Brekelmans. *Computational homogenisation for nonlinear heterogeneous solids*, pages 1–42. 100

[145] Claude Le Bris and Tony Lelièvre. *Multiscale Modelling of Complex Fluids: A Mathematical Initiation*, pages 49–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 100

[146] Guoqing Hu and Dongqing Li. *Multiscale Modeling and Numerical Simulations*, pages 1402–1407. Springer US, Boston, MA, 2008. 100

[147] Gustavo Roberto Ramos and Rodrigo Rossi. A novel computational multiscale approach to model thermochemical coupled problems in heterogeneous solids: Application to the determination of the "state of cure" in filled elastomers. *Computer Methods in Applied Mechanics and Engineering*, 351:694 – 717, 2019. 100

[148] T Gu, J R Medy, F Volpi, O Castelnau, S Forest, E Hervé-Luanco, F Lecouturier, H Proudhon, P O Renault, and L Thilly. Multiscale modeling of the anisotropic electrical conductivity of architectured and nanostructured cu-nb composite wires and experimental comparison. *Acta Materialia*, 141:131 – 141, 2017. 100

[149] M Stepniewski, M Breit, M Hoffer, and G Queisser. Neurobox: computational mathematics in multiscale neuroscience. *Computing and Visualization in Science*, Jun 2019. 100

[150] 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, aug 2010. 100, 102

[151] 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*, 11(2):127 – 140, 1963. 100

[152] Sebastian Fillep, Julia Orlik, Zoufine Bare, and Paul Steinmann. Homogenization in periodically heterogeneous elastic bodies with multiple micro-contact. *Mathematics and Mechanics of Solids*, 19(8):1011–1021, 2014. 100

[153] Katinka Ridderbos. The coarse-graining approach to statistical mechanics: How blissful is our ignorance? *Studies in History and Philosophy of Science Part B: Studies in History and Philosophy of Modern Physics*, 33(1):65–77, 2002. 100

[154] André Zaoui and Renaud Masson. Micromechanics-based modeling of plastic polycrystals: an affine formulation. *Materials Science and Engineering: A*, 285(1):418 – 424, 2000. 101

[155] Jacob Fish. Bridging the scales in nano engineering and science. *Journal of Nanoparticle Research*, 8(5):577–594, Oct 2006. 101

[156] M G D Geers, V G Kouznetsova, and W A M Brekelmans. Gradient-enhanced computational homogenization for the micro-macro scale transition. *Journal de Physique IV, Colloque*, 11(5):145–152, 2001. 102, 107

[157] R Hill. On Constitutive Macro-Variables for Heterogeneous Solids at Finite Strain. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 326(1565):131–147, jan 1972. 106

[158] H M Y C Mallikarachchi and S Pellegrino. Quasi-static folding and deployment of ultrathin composite tape-spring hinges. *Journal of Spacecraft and Rockets*, 48(1):187–198, 2011. 107, 108

[159] Christian Miehe. Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. *Computer Methods in Applied Mechanics and Engineering*, 134(3):223 – 240, 1996. 107, 108

[160] V 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, Jan 2001. 107

[161] Karel Matouš, Marc G D Geers, Varvara G Kouznetsova, and Andrew Gillman. A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *Journal of Computational Physics*, 330:192 – 220, 2017. 107

[162] Markus J. Buehler. Multiscale mechanics of biological and biologically inspired materials and structures. *Acta Mechanica Solida Sinica*, 23(6):471 – 483, 2010. 107

[163] Eujin Pei, Jinsong Shen, and Jennifer Watling. Direct 3d printing of polymers onto textiles: experimental studies and applications. *Rapid Prototyping Journal*, 21(5):556–571, 2015. Copyright - © Emerald Group Publishing Limited 2015; Last updated - 2015-09-10; CODEN - RPJOFC. 107

[164] Evangelos Ptochos and George Labeas. Elastic modulus and poisson's ratio determination of micro-lattice cellular structures by analytical, numerical and homogenisation methods. *Journal of Sandwich Structures & Materials*, 14(5):597–626, 2012. 107

[165] S M Ahmadi, G Campoli, S Amin Yavari, B Sajadi, R Wauthle, J Schrooten, H Weinans, and A A Zadpoor. Mechanical behavior of regular open-cell porous biomaterials made of diamond lattice unit cells. *Journal of the Mechanical Behavior of Biomedical Materials*, 34:106 – 115, 2014. 108

[166] Gaurav Nilakantan, Michael Keefe, Travis A Bogetti, and John W Gillespie. Multiscale modeling of the impact of textile fabrics based on hybrid element analysis. *International Journal of Impact Engineering*, 37(10):1056 – 1071, 2010. 108

[167] Miguel A Bessa and Miguel A. Data-driven Multi-scale Analyses of Materials and Structures. *ProQuest Dissertations And Theses; Thesis (Ph.D.)–Northwestern University, 2016.; Publication Number: AAT 10160508; ISBN: 9781369154153; Source: Dissertation Abstracts International, Volume: 78-02(E), Section: B.; 110 p.*, 2016. 108, 112, 122

[168] Zeliang Liu, M A Bessa, and Wing Kam Liu. Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 306:319–341, jul 2016. 108

[169] A L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1.2):206–226, Jan 2000. 109

[170] Thomas M Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. 110

[171] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. 110, 111, 112, 115, 128, 154

[172] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Application of deep learning for object detection. *Procedia Computer Science*, 132:1706 – 1717, 2018. International Conference on Computational Intelligence and Data Science. 111

[173] ElMouatez Billah Karbab and Mourad Debbabi. Maldy: Portable, data-driven malware detection using natural language processing and machine learning techniques on behavioral analysis reports. *Digital Investigation*, 28:S77 – S87, 2019. 111

[174] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*. MIT Press, 2006. 112, 113, 114, 117, 118, 119

[175] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine learning in Python.

Journal of Machine Learning Research, 12:2825–2830, 2011. 112, 119, 127, 134, 153

[176] R A FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. 112

[177] G M Laslett. Kriging and splines: An empirical comparison of their predictive performance in some applications. *Journal of the American Statistical Association*, 89(426):391–400, 1994. cited By 182. 112

[178] E T Jaynes, E T J Jaynes, G L Bretthorst, and Cambridge University Press. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003. 115

[179] Andrew McHutchon. Differentiating gaussian processes. 2013. 121

[180] C K I Williams. *Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond*, pages 599–621. Springer Netherlands, Dordrecht, 1998. 122

[181] Kristian Kersting, Christian Plagemann, Patrick Pfaff, and Wolfram Burgard. Most likely heteroscedastic gaussian process regression. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 393–400, New York, NY, USA, 2007. ACM. 122

[182] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. 127

[183] Stephen Joe and Frances Y Kuo. Remark on algorithm 659: Implementing sobol's quasirandom sequence generator. *ACM Trans. Math. Softw.*, 29(1):49–57, March 2003. 129

[184] Ramaseshan Kannan, Stephen Hendry, Nicholas J. Higham, and Françoise Tisseur. Detecting the causes of ill-conditioning in structural finite element models. *Computers & Structures*, 133:79 – 89, 2014. 131

[185] Ian Jolliffe. *Principal Component Analysis*, pages 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 132

[186] Dmitriy Fradkin and David Madigan. Experiments with random projections for machine learning. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 517–522, New York, NY, USA, 2003. ACM. 132

[187] Types of Woven Fabrics - universally used fabric names - Textile School. 138

[188] M Haghi Kashani, A Rashidi, B J Crawford, and A S Milani. Analysis of a two-way tension-shear coupling in woven fabrics under combined loading tests: Global to local transformation of non-orthogonal normalized forces and displacements. *Composites Part A: Applied Science and Manufacturing*, 88:272 – 285, 2016. 138

[189] Different weave patterns of Woven Fabrics. 138

[190] Jonathan M Kaldor, Doug L James, and Steve Marschner. Efficient yarn-based cloth with adaptive contact linearization. *ACM Trans. Graph.*, 29(4):105:1–105:10, July 2010. 147

[191] Roger Ghanem, David Higdon, and Houman Owhadi. *Handbook of uncertainty quantification*. 147

[192] Christian Huet. Coupled size and boundary-condition effects in viscoelastic heterogeneous and composite bodies. *Mechanics of Materials*, 31(12):787 – 829, 1999. 147

[193] Sebastian Fillep, Julia Mergheim, and Paul Steinmann. Computational homogenization of rope-like technical textiles. *Computational Mechanics*, 55(3):577–590, Mar 2015. 148

[194] K F Au. *Advances in Knitting Technology*. Woodhead Publishing Series in Textiles. Elsevier Science, 2011. 166, 172