A framework for ’smart’ computational micromechanics

Master’s Thesis

by

Joshua Kiefer

to obtain the degree of
Master of Science - Aerospace Structures and Materials
at the Delft University of Technology,

to be defended publicly on Tuesday October 30st, 2018 at 14:00 AM.

Student number:	4513312	
Project duration:	January 1st, 2018 – October 30st, 2018	
Supervisors:	Dr. S. Turteltaub,	TU Delft
	Dr. C. Lopez,	IMDEA Materials

TU Delft

Acknowledgements

I would like to take this opportunity to thank those who have made this work possible. First and foremost to my supervisor Dr. Cláudio Lopes, who invited me to spend a year with his research group *Design and Simulation of Composite Structures* at *IMDEA Materials*, Madrid. It was his foresight which launched me towards Machine Learning, which proved to be an inspiring and relevant subject to explore and leverage for the work of his group.

Generously granted freedom and responsibility by Dr. Lopes, this journey turned out to be challenging, rich, and surprisingly exciting. In the course of this work, the long-standing expertise of Dr. Lopes and his group in the field of Computational Mechanics of composite materials formed the central pillar, allowing me to learn and lean on the relevant state of the art, and to get the chance to try to advance it.

Indispensable during this process, I want to acknowledge the competent guidance that I received from my supervisors Dr. Cláudio Lopes and Dr. Sergio Turteltaub. Thank you, Sergio, for your valuable input and expertise in the field of Computational Micromechanics and Continuum Mechanics, for always taking the time, and your ever-positive and humorous attitude. Thank you, Cláudio, for our regular discussions, bringing in your ideas, enthusiasm and knowledgeable perspective, which always kept me very motivated and helped shape the direction of the work, step by step and towards a meaningful result.

I am furthermore deeply thankful to Dr. Miguel Herráez, for his constant support and help in my learning to use and adapt his model and tools for my needs. I am likewise grateful for all your shared expertise on Computational Micromechanics, Abaqus, Python, and our discussions on the development, first results, and way forward of the framework. I am confident to say that without your backing this work could not have reached this far.

Last but not least, I want to thank all those who supported me personally throughout this year, my family, friends, partners, dear colleagues, Spanish teachers, and the European Union Erasmus Programme. Thank you for your support through wind and weather, certainly you played an essential role in my success and well-being during this wonderful year.

Joshua Kiefer
Merchweiler, October 2018

Contents

List of Figures

List of Tables

List of Abbreviations

AS4-CF AS4-Carbon-Fiber. xi, 8, 9, 30, 91, 94, 95

BFGS Broyden–Fletcher–Goldfarb–Shanno. 23, 24

CF Carbon Fiber. 14

CFRPC Carbon-Fiber-Reinforced-Polyer-Composite. vi, 8, 70, 73, 75, 78, 79, 82, 83, 90–92

CM Computational Micromechanics. iv, vi, vii, 1–9, 11, 12, 14, 17–19, 23, 26, 27, 35, 44, 45, 47, 50–52, 54, 56–61, 79, 81, 84–86, 91, 93, 97

CPU Central Processing Unit. 28, 31

DoE Design of Experiments. iv, 1, 3–5, 14, 17, 24, 26, 47, 50, 52–56, 58–60, 93, 97

FEA Finite Element Analysis. vi, 3, 4, 17, 23, 24, 26, 73

FEM Finite Element Model. iii, vi, 9, 23, 27, 28, 30

FRPC Fiber-Reinforced-Polymer-Composite. 2–9, 12–14, 26, 28, 32, 34, 35, 44, 47, 50, 52, 58–62, 65, 68, 69, 71, 72, 76, 79, 85–87, 90, 91, 93, 94, 96, 97

FVolF Fiber Volume Fraction. iv, vii–x, 30, 32, 34, 44, 48–50, 52, 58, 59, 62–69, 71–85, 87, 89–94, 97, 98

GUI Graphical User Interface. 28

IFS Interface Strength. iv, vii–ix, 3, 34, 45, 48, 49, 52, 61, 62, 65–73, 75–79, 91–94, 97, 98

L-BFGS Limited Broyden–Fletcher–Goldfarb–Shanno. vii, 24, 37–41, 45–49

ML Machine Learning. 18, 19, 42, 44

MLP Multilayer-Perceptron. vi, vii, 4, 17, 19–23, 25, 26, 37–47

MS Matrix Strength. iv, viii, ix, 52, 65–73, 75–79, 91, 93, 94, 97, 98

NN Neural Network. iv, vi–xi, 1, 3–7, 14, 17–19, 21–27, 35–50, 52, 54–65, 68, 69, 71–73, 76–78, 81–83, 88–90, 93, 96, 97

RBF Radial Basis-Function. 19

RFF Round Fiber Fraction. ix, 30, 79–84

RVE Representative Volume Element. 6, 32

SVE Statistical Volume Element. iii, iv, vi–ix, xiii, 1, 4–10, 13–16, 27–35, 52, 60–68, 71, 73, 77, 80–92, 96–98

SVM Support Vector Machines. 19

UD Unidirectional. vi, 2–9, 12, 14, 26, 28, 32, 35, 47, 50, 52, 58–60, 65, 68, 69, 76, 79, 83, 90–94, 96, 97

Listings

1
Summary

More than half a century after the first application of composite materials in aircraft, the accurate prediction of their failure remains a pressing and unresolved issue. Another important limitation continues to be the low transverse strength of unidirectional composite plies, which can lead to premature failure in common laminates such as the cross-ply. A tool that promises improvements on both of these fronts is computational modelling. With ever increasing computing power, today's multi-scale models are able to simulate the various failure modes across the relevant length scales, and allow a new level of optimization by accurately determining the contributing material parameters.

To fully exploit this capability, we propose a ‘smart’ framework, combining Design of Experiments, Computational modelling, and Neural Networks. It is developed with the aim to create analytical surrogate models of complex material properties, in a fully automated way, and based on a minimum amount of computer simulations. While such a framework can be universally applied, we will showcase our results for one possible application: the prediction and optimization of the transverse strength of a unidirectional composite ply.

Generating Statistical Volume Elements of the material at microscale, we use a Computational Micromechanics model to compute the ply's strength under transverse loading. By Design of Experiments principles, we then explore the interdependent influences of the main geometrical and material parameters. Finally, we obtain an analytical surrogate model of the transverse strength by employing a Neural Network.

Putting to use the developed ‘smart’ framework, the effects of the following parameters were studied: constituent strengths (matrix, fiber-matrix-interface), fiber volume fraction, shape of fiber cross-section (circular or non-circular), and fiber diameter. The results confirm and quantify the primary influence of the constituent strengths, followed by the secondary effect of the fiber volume fraction. Non-circular fiber cross-sections were found to increase transverse compressive strength, and mixing of different fiber diameters may increase both tensile and compressive strength.

In addition to presenting the resulting surrogate models, challenges in the development and automation of the framework are discussed. The underlying Computational Micromechanics model is shown to be mature and computationally cheap enough, to allow for its automatic exploitation through the developed ’smart’ framework. We further believe that the same may hold true for models at other scales of the Multi-scale Modelling approach, and make the case for its wider application in the research field of composite materials.

2
Introduction

Since their inception in the first half of the 20th century, the use of Fiber-Reinforced-Polymer-Composite (FRPC) materials has seen a sharp rise. In 2015, the market for composite materials worldwide was estimated at 66.61 billion US$, and expected to grow by a compound annual growth rate of 7.8% over the period from 2018-2023 [1]. Used originally only in military experimental aircraft, composite materials by now have found widespread use in high quantities in many commercial industries such as aviation, automotive, and sports.

At a baseline level, the specific strength and stiffness of composite materials is much higher compared to materials commonly used in these industries, e.g. aluminum. Furthermore, FRPCs are anisotropic materials which can be highly tailored to a given loading cases, for example by adapting the stacking sequence of their plies accordingly. Both aspects allow for significant weight savings *if* the composite material/structure can be properly designed for the specific use case [34].

The anisotropic nature of FRPC materials results from the heterogeneity of the material across different length scales: the interaction between single fibers and the polymer matrix at microscale, the behavior of a single ply at mesoscale, and the behavior of stacked plies at macroscale. The advantage of tailorability of composite materials however comes at the disadvantage of their many complex failure modes, across different length scales, and each based on different failure mechanisms.

The accurate prediction of these failure modes remains a challenging issue, as highlighted for example by the results of the *World-Wide Failure Exercise* [33]. It can further be summarized that reliable failure *prediction* is necessary to be able to accurately and confidently design composite structures, and thus to fully exploit their outstanding properties and weight saving potential. Furthermore, from a materials science perspective, the accurate *understanding* of their failure mechanisms is necessary to be able to design the next generation of improved FRPC materials.

A powerful methodology, which promises improvements on both fronts, is *Multi-scale Modelling* [42]. The anisotropic FRPC material is modeled at each relevant length scale, from micro- to meso- to macro-scale, including damage modelling of the constituents (fibers, matrix) and the intra- and interply interfaces. Since it is computationally not yet possible to discretize the fibers and matrix elements of all plies of a composite structure, material properties are commonly homogenized at each scale, and then passed on to the next bigger scale as homogenized properties.

Starting with the bottom-most layer, *Nanomechanics* provide the basis to experimentally characterize the material properties of the micro-constituents (fibers, matrix, and their interfaces), for example by nano-indentation. These properties can then form the input for simulations at microscale, modelling the interaction between discrete fibers and the polymer matrix in a field called Computational Micromechanics (CM). Recent advances in CM have shown that it is possible to accurately model the failure modes which have been experimentally observed in Unidirectional (UD) FRPC plies, e.g. the initiation and evolution of interfacial damage and fiber debonding under transverse loading [72][73][13][76][75][48]. Further, these CM models allow to investigate the different geometrical [30], material [27], and modelling parameters [44], which influence these failure modes, and based on this make it possible to optimize the strength of a UD FRPC ply.

Since these simulations have long been computationally expensive, the traditional approach has been to investigate the effect of each parameter individually, e.g. the effect of the fiber-matrix interface strength (IFS) *or* the effect of the fiber volume fraction on transverse failure. To minimize the number of simulations which have to be run, the *interrelated* effects however have never been explored, i.e. how does the effect of IFS on transverse strength depend on the fiber volume fraction?

With today's computing power it is now possible to explore these interrelated effects, and furthermore it can be attempted to create an accurate and comprehensive surrogate model of the failure of a composite ply. The framework that we suggest for this aim is a combination of three distinct methodologies: Design of Experiments (DoE), Computational Micromechanics, and Neural Network (NNs). With such a framework for 'smart' Computational Micromechanics, we claim to be able to address the challenges involved in creating analytical surrogate models of complex material properties, in a fully automated way, and based on a minimum amount of computer simulations.

The idea to combine DoE with computer simulations and a high-dimensional function approximation is a universal approach. With computing power on a steady rise, we have recently begun to see such method successfully applied across different fields. In the field of research on composite materials and structures, we felt particularly inspired by the early example of post-buckling optimization using a combination of a Genetic Algorithm, Finite Element Analyses (FEAs) and NNs [7], as well as the optimization of a composite airframe structure for stiffness using a combination of DoE, FEAs, and NNs [23].

Most recently, the first accounts surfaced of this method being applied in the field of CM. Using a combination of DoE, CM, and NNs, M. Bessa et al. [6] determined the constitutive laws of a nonlinear elastic heterogeneous material, derived from the results of 10377 simulations. For their generic material, elliptical particles are embedded in a matrix, allowed to overlap, and both constituents are assumed to be perfectly bonded. For the work of this master's thesis now, we propose said methodology for the realistic case of today's FRPC materials. We include a validated damage model of the constituents and their interface, necessary to represent the relevant failure modes at microscale, and base ourselves on matrix and fiber-matrix interface properties which have been measured with up-to-date in-situ techniques based on *Nanomechanics* [59][49]. Since the low transverse strength of UD FRPC plies continues to be a significant limitation for the efficient use of composite materials [34], and since transverse behavior can be conveniently represented with a 2D model, we focus on exploring the transverse strength of a UD FRPC ply.

While the proposed methodology could be applied across the different scales of the *Multi-scale Modelling* approach for composites, the computational model of the chosen scale defines the individual challenges that have to be overcome. In any case, full automation of the entire simulation process is a must, since hundreds to thousands of simulations have to be generated, executed, monitored, and post-processed automatically. For the specific case of proposed framework for 'smart' Computational Micromechanics, the damage part of the chosen CM model introduces particular challenges and modelling parameters that have to be taken into account.

The objectives of this master's thesis can be summarized in several research questions, the first one addressing the development of the proposed methodology, and the remaining ones addressing the understanding and improvement of the transverse strength of FRPC materials:

I Can we adapt the combined DoE-CM-NN approach to efficiently generate a comprehensive analytical surrogate model of a complex material property such as the transverse strength of a UD FRPC ply?

II Putting to use the developed framework, what are the main parameters determining the low transverse strength of a UD FRPC ply? Can hybridization at fiber-by-fiber level, i.e. the mixing of circular and non-circular fiber shapes, and the mixing of different fiber diameters improve transverse strength?

III Based on the potential answers found to these last questions through the proposed framework, what are the most promising directions of research to improve the transverse strength of FRPC materials?

The work of this master's thesis will try to address all of these questions, to which we shall return to during the conclusions presented in Chapter 8. Until then, the outline of this document is as follows:

Chapter 3 summarizes and explains the state of the art of the methodologies to be combined. Considering DoE to be a familiar tool, we restrict ourselves to the discussion of CM and NNs. In the first part, CM is discussed. We introduce the approach of *Multi-scale Modelling* and show where CM fit within. Then, the fundamentals of the used method of Statistical Volume Elements (SVE) and the CM model to be used are explained in detail. Lastly, we present an example from literature, showing how said model is able to represent damage up to failure within a UD FRPC ply under transverse loading.
The second part discusses NNs. First, the general field of *Machine Learning* is introduced, allowing the reader to situate NNs within this. Then, different NN architectures and alternatives are explained and our choice for the Multilayer-Perceptron (MLP) is motivated. Following that, the most fundamental working and parameters of a MLP NN are explained, such that the reader can understand the training process which is necessary to fit a NN to a given dataset. We conclude with presenting an example from literature, the optimization of a composite airframe structure for stiffness by using a combination of DoE, FEAs, and NNs.

After the discussion of the state of the art in both CM and NNs, the following two chapters discuss how both methods had to be adapted in order to fit within the proposed framework. Most importantly, every part of the framework has to run in a fully automatic way, without any user interaction. Chapter 4 details the challenges in the automation of the CM model. The simulation workflow is presented, together with rules and adaptations that had to be made to the existing set of CM model and SVE generator tool, in order to fully automate the simulation process and make it suitable for the conceived 'smart' framework. Then, Chapter 5 explains all choices that had to be made for the NN implementation. With practical examples, the effects of the most important parameters on the fitting quality of a NN are explained. Lastly, we motivate and explain the automated training procedure which was eventually chosen for the NN part of the framework: *Cross-Validation* with *repeated-5fold*.

Having explained both the CM and NN adaptations for the proposed framework, we can then proceed with presenting the latter in Chapter 6. An overview of the iterative and fully automated workflow of the framework is given: The user defines which material property relationship (s)he wants to determine, e.g. transverse tensile strength as a function of matrix strength, fiber-matrix interface strength, and fiber volume fraction. The DoE principles then define which datapoints to compute, i.e. for which combinations of input parameters the corresponding transverse tensile strength shall be simulated, in order to obtain an approximation function of said relationship. For any given number of datapoints, the NN is used to generate the surrogate model, whose accuracy depends on the number of underlying datapoints. The target accuracy set out by the user then defines the number of datapoints to be simulated, and the automated process ends once said accuracy is achieved.

After having explained the general working of the framework, Chapter 7 presents the example results generated with the framework. The first two sensitivity studies characterize the influence of two important modelling parameters: the number of fibers within the SVEs, and the minimum inter-fiber distance. Both parameters have an important influence on transverse strength results and have to be fixed a-priori for all ensuing studies. The main part then covers three different studies characterizing the transverse strength of a UD FRPC ply through our 'smart' framework. The first study determines the interrelated effect of the relevant constituent strengths (matrix, and fiber-matrix interface) and the fiber volume fraction. The last two studies fix the constituent strengths at the nominal values measured for a AS4/8552 material and explore the effect of hybridization: mixing of different fiber shapes (circular with 4-lobed) and the mixing of two different fiber diameters, for varying total fiber volume fractions.

Finally, Chapter 8 summarizes and concludes on our findings and Chapter 9 presents thoughts on how to continue this work, both short-term and applied to CM, as well as from a wider perspective.

3

State of the Art

As outlined in the introduction, the ’smart’ framework proposed in this master’s thesis is a combination of three methodologies: Design of Experiments, Computational Micromechanics, and Neural Networks. Each one of these is their own field of research, so we want to introduce their current state of the art in this chapter. Considering Design of Experiments a well-established method, we shall focus on the more recent fields of Computational Micromechanics and Neural Networks. First, we will situate each one of them within their wider context, respectively *Multi-scale Modelling* and *Machine Learning*. The ensuing discussion of each method aims at giving the reader the necessary baseline to judge the framework presented in the following chapters. Lastly, we conclude each section with an example from literature, relevant to the proposed framework, and give an outlook at some of the challenges ahead.

3.1. Computational Micromechanics

We shall begin with the discussion of Computational Micromechanics (CM). Overall, the entire framework rests on the quality of the underlying simulation data, and the ability of the model to capture the transverse failure modes of a Unidirectional Fiber-Reinforced-Polymer-Composite (UD FRPC) ply. First, we introduce the approach of *Multi-scale Modelling*, and situate Computational Micromechanics within that. Then, we explain the readily developed CM model which has to be adapted for the proposed framework. We cover the used method of Statistical Volume Elements, explain how they are generated, and by which geometrical parameters they describe a FRPC material. Following that, we discuss the details of the CM model, including all material and modelling parameters which are varied later-on, during the studies presented in Chapter 7. Finally, we introduce the research area of fibers with non-circular cross-sections, and show by way of example how said CM model has been used to simulate the transverse failure of FRPC materials.

3.1.1. Introduction: Computational Micromechanics within Multi-Scale Modelling

As mentioned in the introduction, *Multi-scale Modelling* is a state-of-the-art approach to simulate a material whose behavior is defined across multiple relevant length scales. While this approach can be employed for many different kinds of materials, in the work of this master’s thesis, we restrict ourselves to the class of composite materials. In this thesis, when writing *composite materials* we refer to FRPC materials. They commonly consist of two components, or *phases*: a high-elongation low-strength *matrix* material, and low-elongation high-strength fibers which are embedded in the matrix for reinforcement.

By adjusting the reinforcement phase of fibers, the overall material behavior of the FRPC can be tailored. The fibers can be short or long, arranged Unidirectional (UD) or randomly dispersed, and different fiber materials offer different mechanical properties for the fiber phase. Since the diameter of a single fiber generally lies in the range of 3 μm to 20 μm [79], the behavior of the FRPC material at the level of *single fibers* is referred to as the *microscopic behavior* at *microscale*, and is studied in the so-called field of *micromechanics*. Contrary, the overall material behavior at the mm-, cm-, or m-scale, is referred to as the *macroscopic* behavior at *macroscale*. Since many FRPC materials further consist of a stacking of anisotropic layers, an intermediate length scale was introduced, called *mesoscale*, which refers to the behavior of a single FRPC ply.

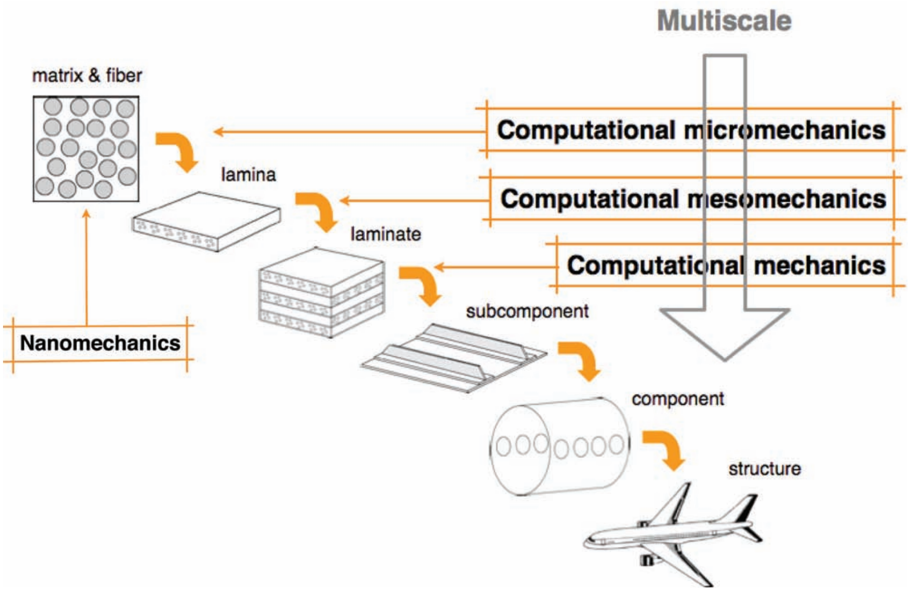


Figure 3.1: *Multi-scale Modelling* approach for the design of a composite aircraft structure. Reproduced from [42].

Since the *macroscopic* behavior of a material is determined by the interaction of its constituents at all lower-order length scales, today's state-of-the-art simulation approach for FRPC materials aims to consider all relevant material scales and is called *Multi-scale Modelling* [42]. However, with today's simulation capabilities it is yet impossible to compute a *single* model which contains details of all length scales, e.g. to simulate a multi-layer FRPC under tension with finite elements discretizing every single fiber. Instead, the approach taken today is to model the material behavior at lower level scale, and then pass volume averaged properties to the material model at the next higher scale [87].

This process is often referred to as *homogenization*, or *regularization*, and the so-obtained properties are called *effective properties*. Averaging of the material properties at each length scale is based on a *statistically representative volume* of the material, either a single large so-called *Representative Volume Element* or several smaller so-called Statistical Volume Elements (SVE) [87]. In summary, the goal of Computational Micro/Macromechanics, i.e. *Multi-scale Modelling*, can be stated as to *determine relationships between the microstructure and the macroscopic response or "structural property" of a material, using models on the microscale that are as simple as possible [87].*

Figure 3.1 [42] illustrates the concept of *Multi-scale Modelling* applied bottom-up to the design of a composite structure. First, *Nanomechanics* provide the properties of the pure constituents of the FRPC, i.e. of the fiber, matrix and interface. Then, *Computational Micromechanics* simulate the interaction between fibers and matrix in several SVEs, which is to be chosen representative of the next bigger length scale: a single UD FRPC ply/lamina. *Computational Mesomechanics* then are used to extract effective properties of a single ply passed on to simulations at the scale of a multi-ply laminate, here termed *computational mechanics*. The process further extends to the scale of subcomponent, component and eventually the final structure.

The ensuing work of this master's thesis is confined to the scale of *Computational Micromechanics*, i.e. simulations of 2D SVEs of a UD FRPC, at microscale, and under transverse loading. The following section shall now provide a description of the model to be used.

3.1.2. The Computational Micromechanics Model to be Adapted

For the work of this master's thesis, Computational Micromechanics (CM) will provide the baseline, based on which a Neural Network then creates a surrogate model of the ply property to be approximated. As every CM model requires experimental validation of its modelling assumptions for significance of its results, a validated and readily-developed model will be used for the purposes of this master's thesis.

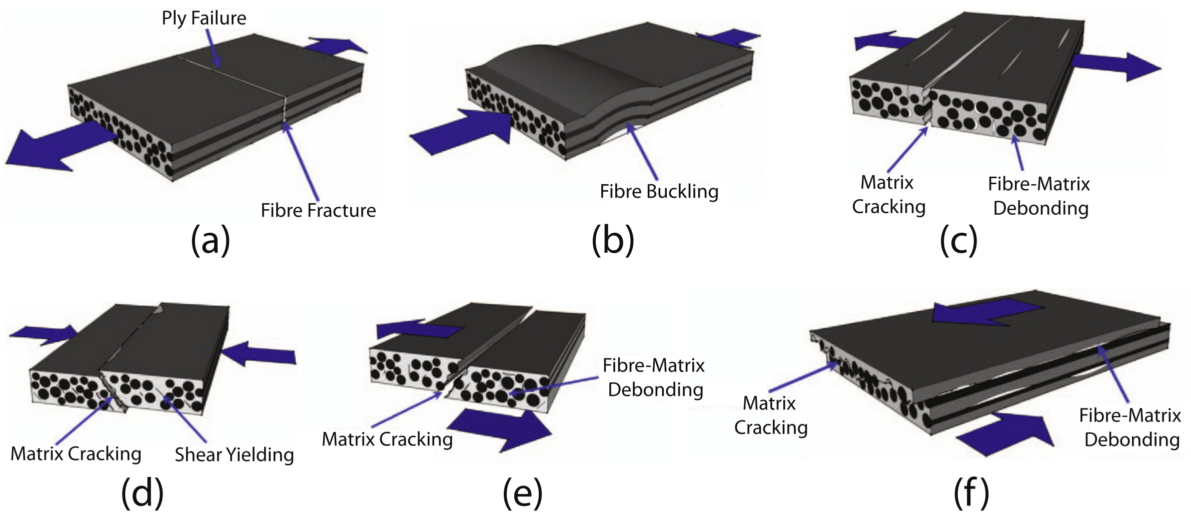


Figure 3.2: Failure modes of a unidirectional fiber-reinforce-polymer-composite ply under a) longitudinal tension, b) longitudinal compression, c) transverse tension, d) transverse compression, e) transverse shear, f) longitudinal shear. Longitudinal and transverse refer to the fiber direction. Reproduced from [28].

The CM model to be used has been developed at *IMDEA Materials*, Madrid, Spain, and builds on the heritage of pioneering work done by the institute's researchers in the field of CM using SVEs, as early as 2007 [27]. Key success was the accurate recreation of the different failure mechanisms under transverse loading, by this novel method [72][73][13][76][75][48], based on constituent properties obtained through nanomechanical in-situ measurements [28]. Most recently, the simulation capabilities at IMDEA Materials have been extended by the ability to automatically generate a wide variety of volume elements, incorporating non-circular fiber shapes [30], which shall be put to use for the work of this master's thesis.

3.1.2.1. From Ply to Statistical Volume Element

[From Ply to Statistical Volume Element (SVE)] The material to be described by the CM model is a single ply of a polymer material reinforced with continuous fibers aligned in a single direction. In short, it will be referred to as a UD FRPC ply. Figure 3.2 [28] gives an overview of all generic failure modes of a UD FRPC ply under a) tensile loading along the fiber direction, b) compressive loading along the fiber direction, c) tensile loading transverse to the fiber direction, d) compressive loading transverse to the fiber direction, e) shear loading transverse to the fiber direction, and f) shear loading along the fiber direction.

After restricting ourselves to CM, the question remains which failure modes to simulate. For practical reasons, we will restrict ourselves to the transverse behavior, i.e. the behavior under loading transverse to the fiber direction, cases c)d)e). As outlined in the introduction, the framework developed in this thesis can be easily transferred to any of the other cases, but three reasons motivate our choice for transverse behavior:

- The failure modes under transverse loading can be captured by computationally less expensive 2D SVEs, compared to the longitudinal failure modes requiring 3D SVEs. Choosing the less expensive 2D SVE models allows to compute more datapoints for the training of our NNs, and thereby to incorporate more input parameters.
- The 2D CM models capturing transverse failure are well matured and established, whereas accurate models for longitudinal failure, e.g. fiber kinking, are still under development [48].
- Low transverse tensile strength remains a significant limitation of today's FRPC materials [34], with strong motivation for improvements. Furthermore, the ply properties of transverse strength are difficult to predict analytically. Both can be addressed by CM models.

Figure 3.3 shows on the left an optical micrograph of the cross section of a UD FRPC ply. Instead of modelling the actual fiber distribution of the entire ply through a so-called *embedded cell*, the objective of the SVE methodology is to find a SVE just big enough to represent the entire ply thickness. As Figure 3.3 shows on the left, the fibers are dispersed in a random way, leaving aside some of the boundary regions, and therefore the

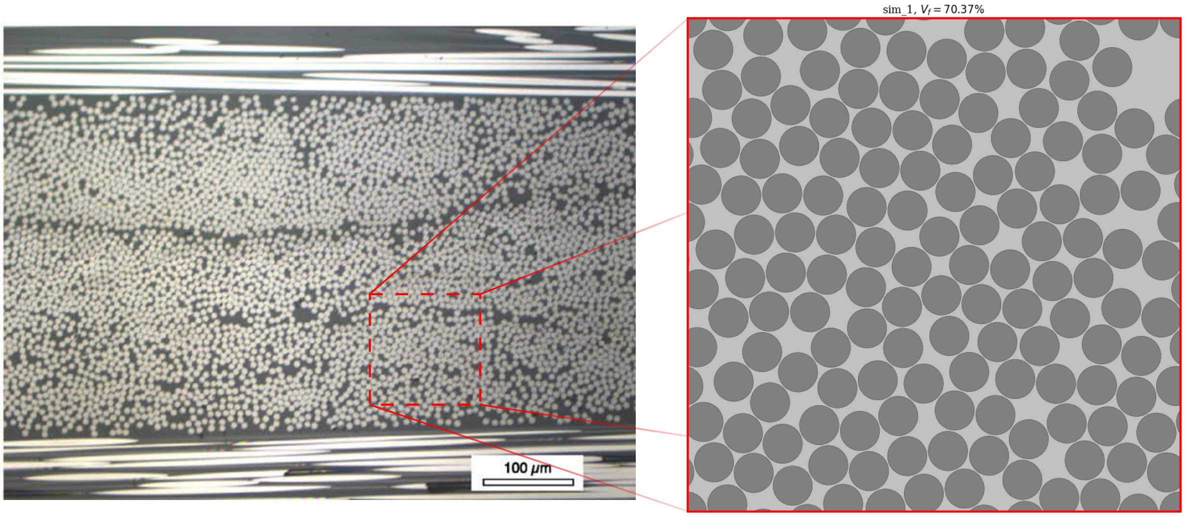


Figure 3.3: Optical micrograph of a UD CFRPC ply (left), and example SVE at 70% fiber volume fraction, generated with *Vip3r*. Micrograph reproduced from [73].

SVE has to recreate a random fiber distribution representative of the actual fiber dispersion[1]. The generation of such a volume element is far from trivial and the *Vip3r* tool to be used in conjunction with the CM model has been developed by M. Herraez [30] in the course of his PhD, building on previous developments at *IMDEA Materials*. An example volume element generated with said tool is shown in Figure 3.3 on the right. The generation is implemented in *Python* and aided by a graphical user interface. The following SVE parameters can be varied:

1. Fiber Shape: Different fiber shapes are available: circular, polygonal, lobular, elliptic. Further, their angular orientation can be chosen, either all oriented the same way, or according to a statistical distribution (log, normal, weibull, uniform).

2. Fiber Diameter: Fiber diameters can be varied according to statistical distributions (log, normal, weibull, uniform).

3. Fiber Volume Fraction: The value of the fiber volume fraction can be set.

4. Fiber Material: Different fiber materials are available, e.g. E-Glass Fiber, AS4-Carbon Fiber (AS4-CF)

   For the whole SVE, the following modelling parameters have to be set:

5. SVE Size: The dimensions of the SVE.

6. Minimum Fiber Distance: The minimum distance between two fiber edges.

The first four parameters describe the geometry and fiber material of the FRPC material that the SVE shall represent. The last two parameters pertain to the SVE methodology. Eventually, the effect of all six of them has been characterized with the proposed framework, to be presented in Chapter 7. For the interpretation of the final results, it is important to briefly summarize the current state of knowledge on the influence of the two methodological parameters *SVE size* and *minimum inter-fiber distance*:

- The SVE size has to be chosen such that the SVE contains enough fibers for a statistically representative model. A common approach in literature is to choose a square SVE with a ratio between SVE width to fiber diameter of 8 [30] [27]. However, this holds only for a certain fiber volume fraction and for the specific kind of results computed in these studies. In general, every CM model using SVEs requires its own sensitivity study to ensure the specific results generated with this model are independent of the SVE size.

[1] Research has shown that regular periodic fiber distributions are inadequate to model transverse failure since the minimum inter-fiber distance is a decisive influence [77] [25].

- The minimum inter-fiber distance (*Dmin*) has to be fixed at a certain value to ensure correct meshing of the CM model. While in reality, many fibers in the cross-section of a UD FRPC ply may touch, the CM model requires a minimum distance between fibers for a matrix element to fit in there. At the same time, it was found that *Dmin* has a significant effect on the stress field between fibers and matrix and thus strongly influences the values of transverse strength obtained by CM with SVEs [77] [25]. For results to be comparable, all SVEs that shall be compared have to maintain a constant value for *Dmin*.

3.1.2.2. From Statistical Volume Element to Finite Element Model

After generating a statistically representative SVE geometry, the latter has to be transformed into a CM model that can be simulated under transverse loading, in our case a Finite Element Model (FEM).

3.1.2.2.1 Periodicity Key requirement of the SVE method is the geometric periodicity of the model [69]. This has to be fulfilled in two aspects:

- Fiber dispersion: The SVEs shall be a small but periodic representation of the *full* volume of the material to be modeled. Accordingly, the full volume of the material could be obtained by duplicating the SVE, and translating the copies along the edges of the SVE to form a fitting puzzle. For this to be possible, the fibers on the edges of the SVE have to be sliced and positioned accordingly, see Figure 3.4, which is taken into account automatically by the generation algorithm.
- Boundary conditions: During deformation of the SVE under loading, geometric periodicity of the deformed SVE has to be ensured. In simple terms, if the deformations of the edges of the single SVE are not periodic, duplicates of the SVE would not fit together as a jigsaw puzzle anymore and thus render the single SVE invalid to represent a larger material volume. Geometric periodicity under deformation is imposed on the SVEs through periodic boundary conditions. For a SVE of depth, length, and height of $w_0 \cdot L_0 \cdot L_0$, the nodal displacements of opposite SVE edges are constrained by the following equations:

$$\vec{u}(0, X_2, X_3) - \vec{u}(w_0, X_2, X_3) = \vec{U}_1 \tag{3.1}$$

$$\vec{u}(X_1, 0, X_3) - \vec{u}(X_1, L_0, X_3) = \vec{U}_2 \tag{3.2}$$

$$\vec{u}(X_1, X_2, 0) - \vec{u}(X_1, X_2, L_0) = \vec{U}_3 \tag{3.3}$$

Herein, X_1, X_2, X_3 denote the coordinate axes with $0 < X_1 < w_0, 0 < X_2 < L_0, 0 < X_3 < L_0$, see Figure 3.4. Three master nodes are defined in the corners of the SVE: $MN_1(w_0, 0, 0), MN_2(0, L_0, 0), MN_3(0, 0, L_0)$ and their displacements are denoted by $\vec{U}_i$. Transverse tension or compression, i.e. transverse to the longitudinal fiber axes, is applied on the SVE by imposing $\vec{U}_2 = (0, \pm\delta_2, 0), \vec{U}_1 = (u_1, 0, 0), \vec{U}_3 = (0, 0, u_3)$, with δ_2 denoting the tensile/compressive displacement, and u_1, u_3 being the lateral contractions according to Poisson, obtained under the surface integral of the traction vector [30]

$$\int \vec{t} dS = \vec{0}, \text{ on } X_1 = 0, \text{ and } X_3 = 0 \tag{3.4}$$

3.1.2.2.2 Constitutive Material Models The choice of material models for the individual constituents of the UD FRPC ply is crucial for the FEM to be able to resolve the different failure mechanisms of the ply. Discretization by finite elements and analysis of the SVEs are carried out in *Abaqus/Standard* [70].

- Fibers: The fibers are discretized by 6-node fully integrated wedge isoparametric elements (C3D6). The material properties of the used AS4-CFs are shown in Table 3.1. The behavior of the AS4-CFs was modeled as elastic and transversely isotropic, and fiber fracture was not considered relevant under transverse loading, in accordance with experimental findings on AS4/8552 [27].

	E_1 (GPa)	E_2 (GPa)	ν_{12}	ν_{23}	G_{12} (GPa)	G_{23} (GPa)	α_1 ($10^{-6}K^{-1}$)	α_2 ($10^{-6}K^{-1}$)
AS4-CF	231.6	12.97	0.3	0.46	11.3	4.45	-0.9	7.2

Table 3.1: Material Properties of AS4-CF [31]

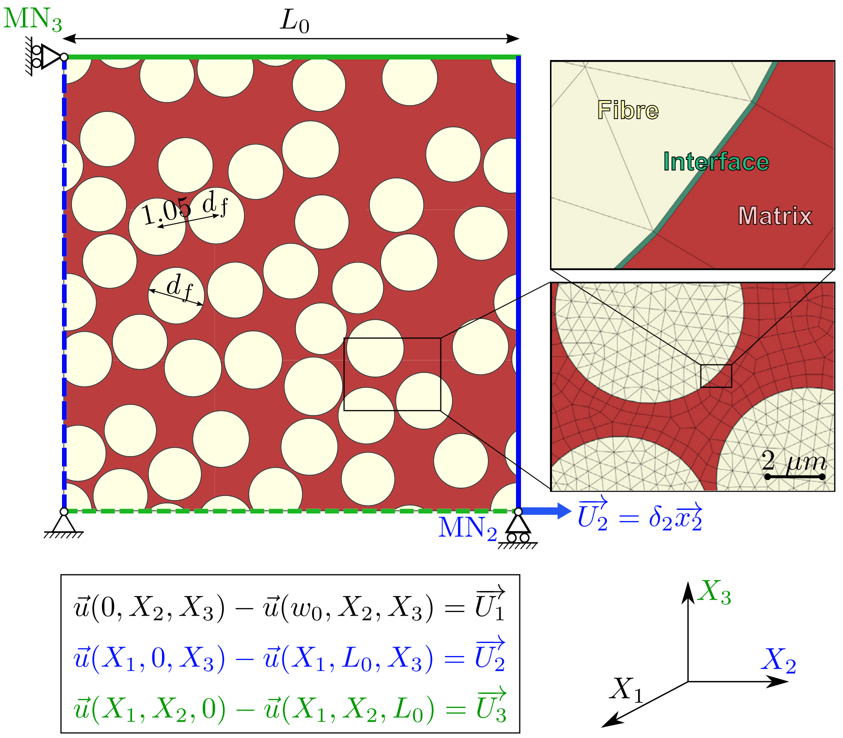

MN_3
L_0
Fibre
Interface
Matrix
$1.05\ d_f$
d_f
$2\ \mu m$
MN_2
$\vec{U_2} = \delta_2 \vec{x_2}$
$\vec{u}(0, X_2, X_3) - \vec{u}(w_0, X_2, X_3) = \vec{U_1}$
$\vec{u}(X_1, 0, X_3) - \vec{u}(X_1, L_0, X_3) = \vec{U_2}$
$\vec{u}(X_1, X_2, 0) - \vec{u}(X_1, X_2, L_0) = \vec{U_3}$
X_3
X_2
X_1

Figure 3.4: Schematic of the 2D SVE model: coordinate axes, periodic boundary conditions, master nodes, and mesh details. The case of transverse tension in X_2 direction is illustrated. Reproduced from [30].

- Matrix: The matrix is discretized by 8-node fully integrated brick isoparametric elements (C3D8). Mechanical behavior of the 8552 Epoxy polymer matrix was modelled through the Lubliner plastic-damage model provided in *Abaqus/Standard* [43]. When subjected to predominantly tensile stress, the material behaves as quasi-brittle, while under compressive loading as elasto-plastic, see Figure 3.5b. Under pure tensile stress, the matrix material thus behaves linearly with elastic modulus E_0 and Poisson ratio ν_m until tensile failure at σ_{t0}. Under further tensile loading, a quasi-brittle softening occurs with G_t denoting the matrix fracture energy. Under pure compressive stress, the matrix material also behaves linearly until the compressive yield limit σ_{c0}. Then, no brittle softening but stress hardening occurs until the ultimate compressive stress limit σ_{cu}. The properties of the plastic-damage model chosen for the model of the 8552 Epoxy polymer matrix are summarized in Table 3.2.

E_0 (GPa)	ν_m	α $(10^{-6}K^{-1})$	σ_{t0} (MPa)	G_t (J/m^2)	σ_{c0} (MPa)	σ_{cu} (MPa)
5.07	0.35	52	121	90	176	180

Table 3.2: Material Properties of Lubliner model for 8552 Epoxy polymer matrix [13]

- Fiber-Matrix-Interface: The fiber-matrix interface is discretized by 8-node cohesive isoparametric elements (COH3D8), taking into account failure of the interface through a cohesive crack approach. Full contact between fiber and matrix is assumed and the interface thickness is 1 nm. Behavior of the cohesive elements follows a bi-linear mixed-mode traction-separation law [65][12], which couples the displacement vector $\vec{\delta} = (\delta_n, \delta_s, \delta_t)$ separating the top and bottom faces of the cohesive element to the acting traction vector $\vec{t} = (t_n, t_s, t_t)$, see Figure 3.5a. The subscript n hereby denotes the normal component of the displacement/traction vector, and s, t denote its shear components. The initial linear response of the cohesive element follows the initial elastic stiffness K, which is a numerical parameter chosen just large enough to "ensure displacement continuity at the interface and to avoid any modification of the stress fields around the fibers in the absence of damage" [27]. Damage *onset* is then defined by the maximum stress criterion [50]:

$$\left(\frac{\langle t_n\rangle}{N}\right)^2+\left(\frac{t_s}{S}\right)^2+\left(\frac{t_t}{S}\right)^2=1 \tag{3.5}$$

where the McCaulay brackets $\langle t_n\rangle = max(0,t_n)$ take into account that only tensile normal stresses lead to damage initiation of the cohesive element, i.e. separation. N and S correspond to the normal and shear strength of the interface, i.e. the peak values of the stress when only acting normal to the interface or in its shear direction. For simplicity, it is assumed that the same strength S applies in both shear directions s,t.

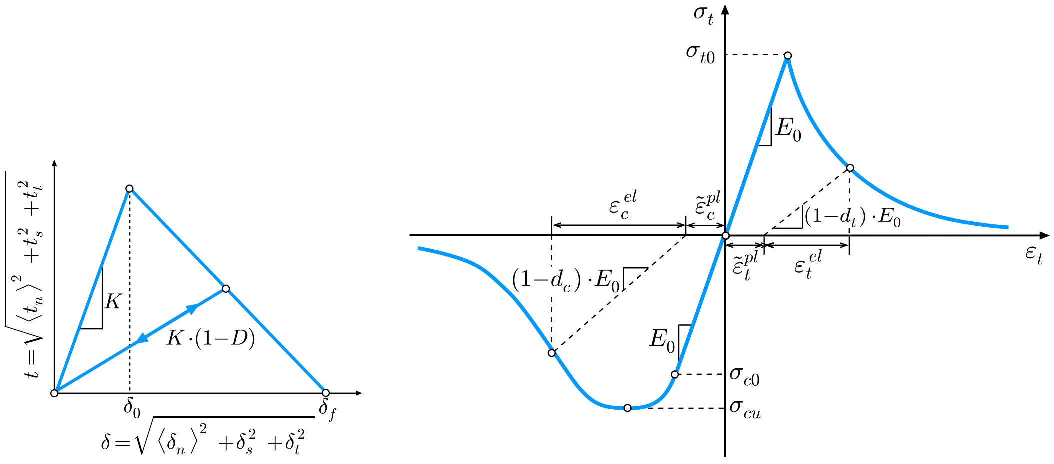
After damage initiation has occurred according to Equation 3.5, a linear softening rule for damage *evolution* applies. Referring to Figure 3.5a, which illustrates the case of a uniaxial stress, the scalar damage variable $0 \le D \le 1$ constitutes the overall damage in the element, with $D = 1$ denoting full damage. Since damage evolution may proceed under combinations of tension/compression and shear within the cohesive element, a mixed-mode definition based on fracture energies is provided. The energy dissipated within the element due to failure G^c, i.e. the area under the traction-separation curve in Figure 3.5a, is defined according to Benzeggagh-Kenane [5] as:

$$G^c = G_n^c + \left(G_s^c - G_n^c\right)\cdot\left(\frac{2G_s}{G_n+2G_s}\right)^{\eta_{BK}} \tag{3.6}$$

where G_n^c, G_s^c are the critical fracture energies in the case of pure loading normal to the element or in its shear direction. G_s, G_n is the corresponding work done by the tractions and the concomitant relative displacements, and η_{BK} the Benzeggagh–Kenane power exponent. Abaqus then ensures that the area under the mixed-mode traction-separation curve equals the specified fracture energy [70]. The values of the interface parameters used in the CM model are derived from in-situ nanoindentation measurements [59][49] and summarized in Table 3.3 and the resulting traction-separation law is visualized up to scale in Figure 3.6.

N (MPa)	S (MPa)	$K_{nn}(GPa)$	$K_{ss}(GPa)$	G_n^c (J/m^2)	G_s^c (J/m^2)	η_{BK}
42	63	100	100	2	30	1.2

Table 3.3: Properties of the fiber/matrix interface [47]



(a) Cohesive interface

(b) Matrix

Figure 3.5: Schematic of a) the traction-separation material law of the cohesive fiber-matrix interface, and b) the Lubliner plastic-damage model of the Epoxy matrix. Reproduced from [30].

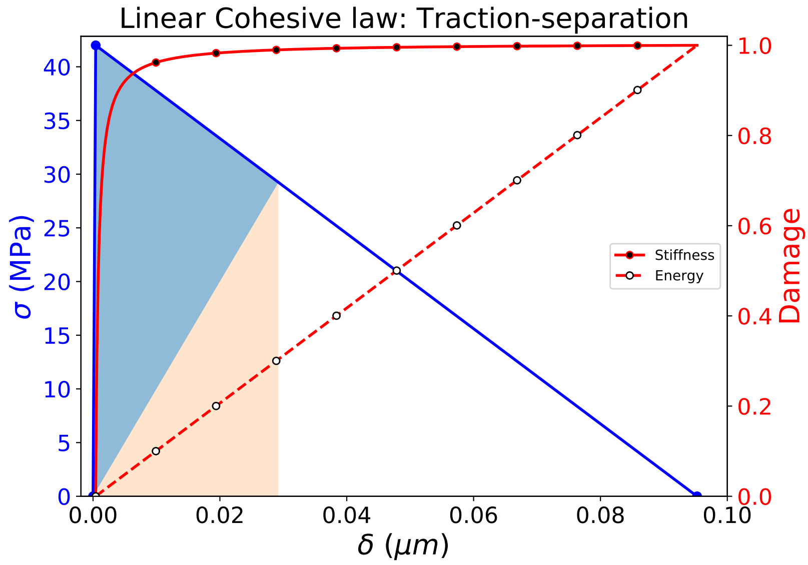


Figure 3.6: Traction-separation law of the cohesive elements used to represent the fiber-matrix-interfaces, for the case of uni-axial tension [30]. Based on the nominal input values from Table 3.3.

3.1.2.2.3 Thermal Loading Residual thermal stresses were found to significantly influence the transverse strength and strain-to-failure of UD FRPC plies [78][30][27]. They result from the curing process and the mismatch of the coefficients of thermal expansion between the fiber and matrix phase, and significantly alter the initial stress field. Accordingly, the to-be-used CM model incorporates a thermal step of $\Delta T = 160\,^{\circ}\mathrm{C}$ prior to the mechanical loading step.

3.1.3. Example Results: Transverse Failure of Composites with Non-Circular Fiber Shapes

To conclude our discussion of the CM model to be used in the proposed framework, we shall briefly discuss some example results, illustrating the model's capabilities to accurately simulate damage under transverse loading. Furthermore, we want to introduce non-circular fiber shapes as a research topic of emerging interest, thus chosen as one of the parameters to be explored in this thesis.

Since the 1980s, new manufacturing methods have been developed which allow to produce fibers of various cross-sectional shapes: Hollow [35][66][52][53], C [35][66][52][53], oval [17], peanut (two-lobular) [17][51], kidney [81][82][51][86], ribbon [20], triangular (3-polygonal) [9][18], and tri-lobular [19]. Examples are shown in Figure 3.7. The question prevails whether non-circular fibers can improve the mechanical properties of FRPC materials, particularly in transverse direction. Since experiments on transverse strength are difficult to conduct, let alone to manufacture fibers of different cross-sectional shapes but identical properties, CM offers the advantage to isolate the shape effect.

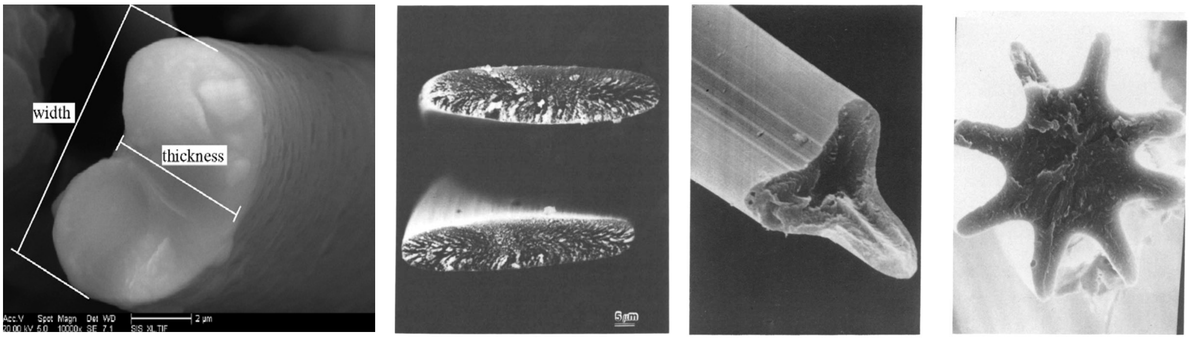
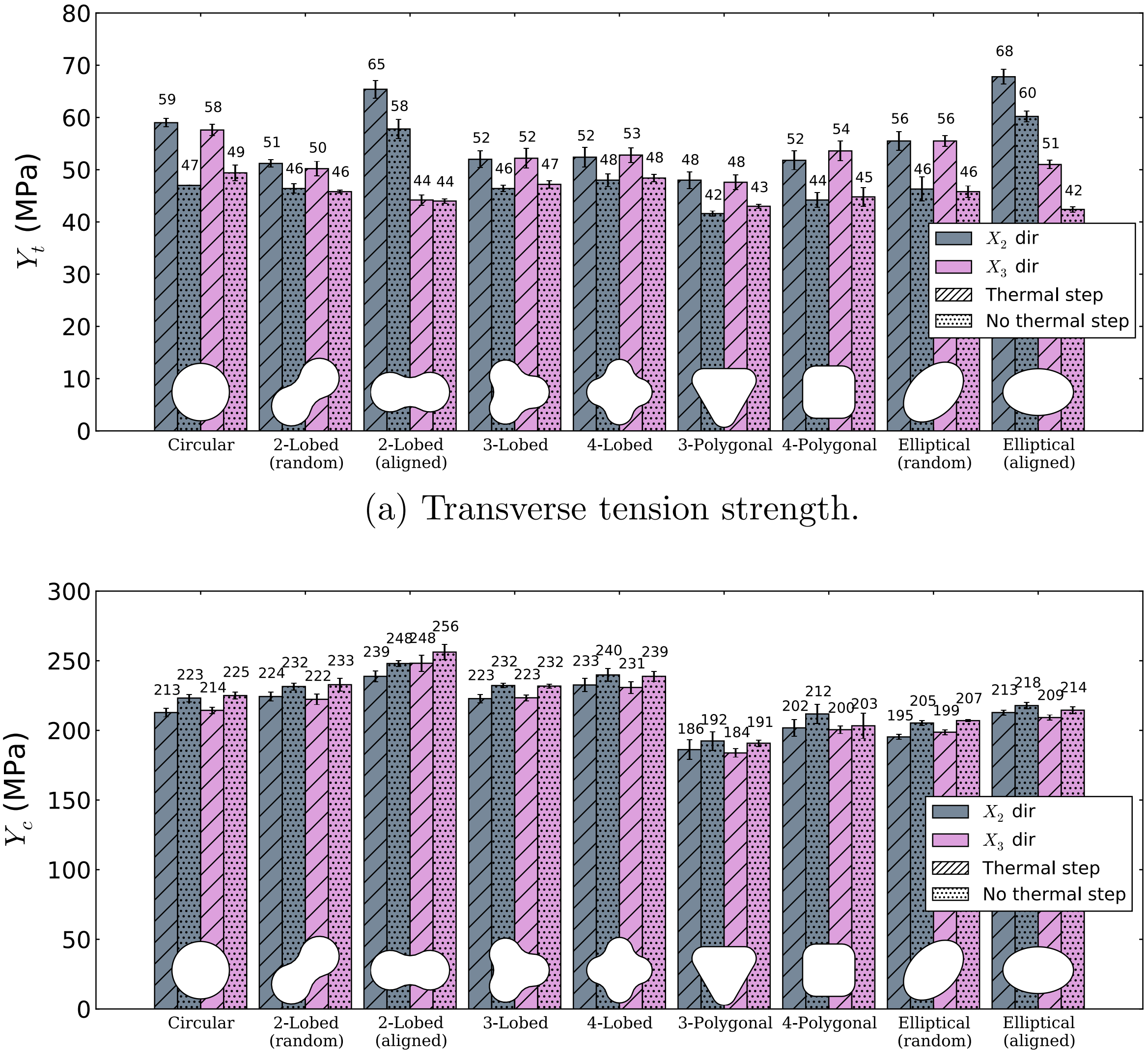


Figure 3.7: Examples of manufactured fibers with non-circular cross-section, from left to right: kidney, elliptical, tri-lobular, octa-lobular. Reproduced from [82][20][19].

The most thorough micromechanics investigation into the effect of non-circular fiber cross-sections on transverse strength of a FRPC is provided by [30], which is also the model to be adopted for the proposed framework. Seven different fiber shapes were compared: circular, 2-lobed, 3-lobed, 4-lobed, 3-polygonal, 4-polygonal, and elliptical; also varying the alignment for the 2-lobed and elliptical shapes. Tensile strength higher than for the circular fiber was only reported for the 2-lobed and elliptical fibers when *aligned*, i.e. when the tensile load was applied along the major axis of the aligned fiber cross-sections. However, the proportional *de*crease in tensile strength aligned with the *minor* axis, as predicted by Tsai et al. 1966 [74], was confirmed as well. Higher compressive strength than for circular fibers was reported for all lobular cross-sections, again more pronounced for aligned fiber distributions. Figure 3.8 summarizes the found results, including the effect of a thermal step of $\delta_T = 160\,°\mathrm{C}$ prior to mechanical loading. X_2, X_3 refer to the orthogonal axes who span the area of the 2D SVEs.

In the course of this thesis, we now want to look one step further, and explore the effect of mixing different fiber shapes. Again, the same aforepresented computational micromechanics model is to be used for that, and we shall present one example result for a single hybrid SVE under each transverse tensile and compressive loading[2]. We will use this to illustrate the capability of the model to represent damage evolution up to failure.



(a) Transverse tension strength.

(b) Transverse compression strength.

Figure 3.8: Transverse strength results for simulations of 2D SVEs of different fiber-cross-section-shapes, with and without thermal step of $\delta_T = 160\,°\mathrm{C}$: a) transverse tension, b) transverse compression. Reproduced from [30].

[2] Comprehensive results for several SVEs will be shown later on in Section 4.4.1

Figure 3.9 shows a meshed SVE of the cross-section of a UD ply of intermingled circular and 4-lobed AS4 CFs, embedded in a 8552 matrix. Figure 3.10 shows the stress-strain curve for the same SVE subjected to transverse tension. The deformed SVE after a 10% load drop from maximum stress is depicted in Figure 3.11 (deformation scale factor = 5), i.e. at a strain level of 0.9%. As in the case for circular fibers, final transverse failure under tension constitutes of a joined band of interface failures. Figure 3.11 highlights the latter by plotting the *overall scalar stiffness degradation* SDEG of the cohesive elements representing the fiber-matrix-interfaces. As can be readily seen, the horizontally applied transverse tension has eventually created a transverse crack through the entire height of the SVE, consisting of joined interface failures around single fibers.

Figure 3.12 show the stress-strain curve of the same SVE now failing under transverse compression. In correspondence with results from literature on circular fibers [27], final transverse failure constitutes of one or several joined shear bands within the matrix, triggered initially by interface failure. Figure 3.13 shows the plastic strain distribution within the now deformed SVE of Figure 3.9 at the point of maximum stress (deformation scale factor = 1). As can be seen, several shear bands with peaks of plastic strain of up to 44% have formed throughout the volume of the SVE, leading to a final loss of load-carrying capability.

3.1.4. Summary

We now have introduced the Computational Micromechanics (CM) model to be used as a baseline for the framework proposed in this thesis. The Design of Experiments approach will eventually be used to automatically command the simulation of varying parameter combinations of said CM model. Finally, a Neural Network will be used to create a coherent surrogate model, based on the available database of simulated parameter combinations. In this section, we have introduced and described all geometrical, material, and modelling parameters of the CM model, which will be varied later on in Chapter 7. Lastly, with an example from literature we illustrated the capability of the CM model to simulate the two dominant failure modes in a UD FRPC ply under transverse loading: fiber-matrix interface failure, and shear band formation within the polymer matrix.

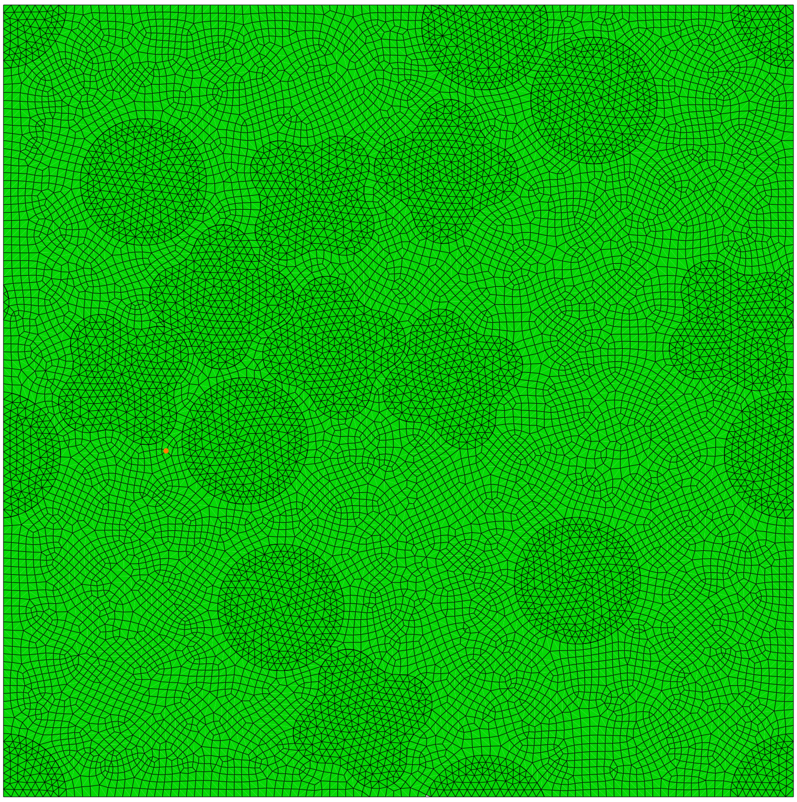


Figure 3.9: Meshed 2D SVE of 50µm x 50µm, mixing circular and 4-lobed fibers of each 15% volume fraction and 8µm diameter. Mesh size is 0.5µm, minimum inter-fiber distance is 0.2µm.

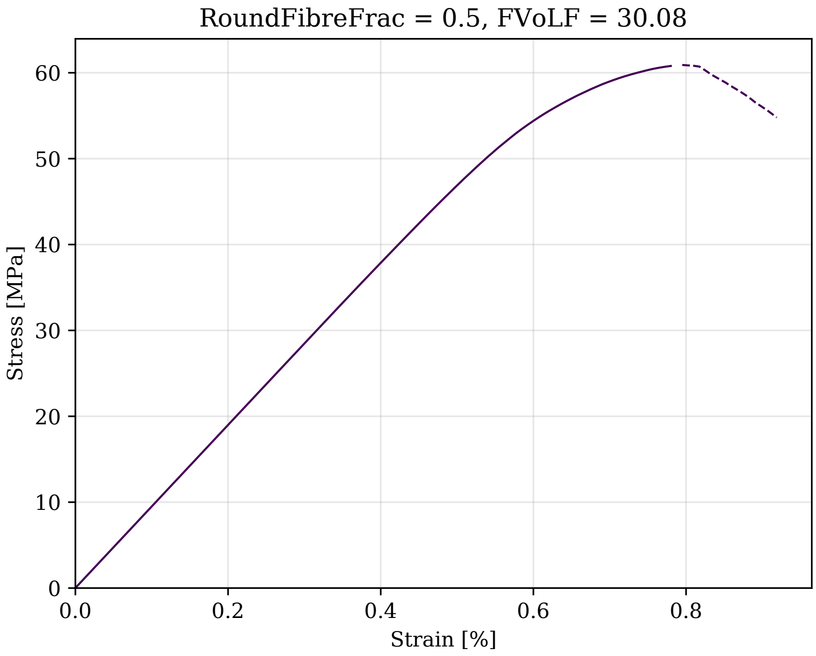


Figure 3.10: Stress-strain curve for the SVE from Figure 3.9 subjected to transverse tension. Dashed line after the maximum stress value.

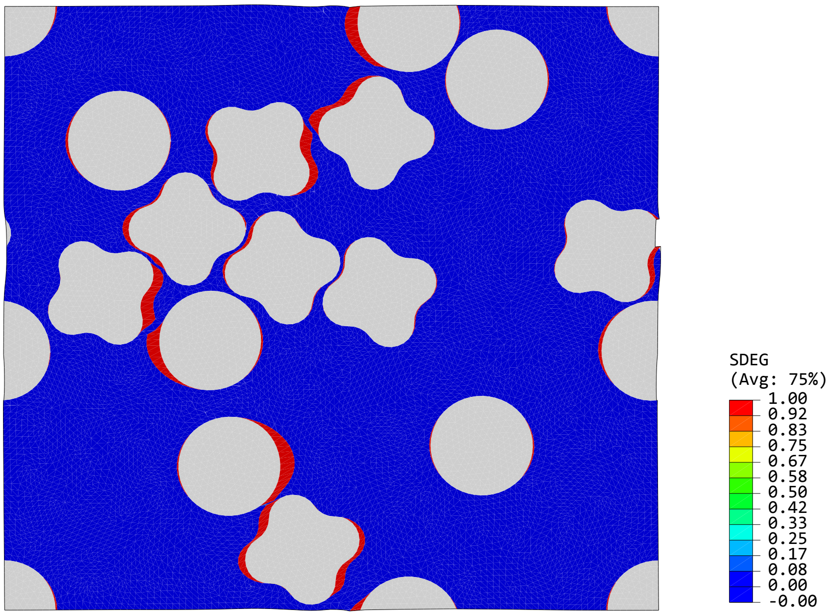


Figure 3.11: Deformation of the SVE from Figure 3.9 subjected to transverse tension and shown at the final strain value computed at 0.9%. The plotted variable is the *overall scalar stiffness degradation* SDEG of the fiber-matrix interfaces. Deformation is shown with a scaling factor of 5.

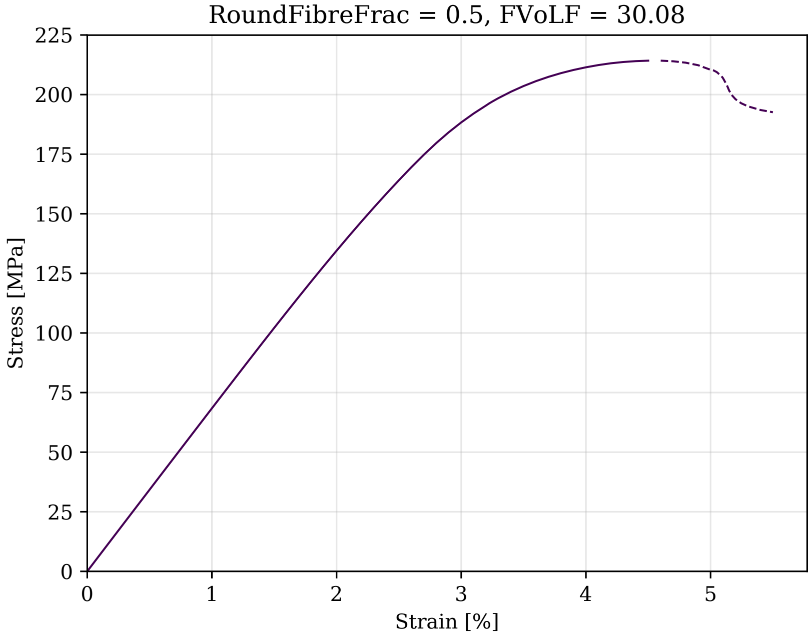


Figure 3.12: Stress-strain curve for the SVE from Figure 3.9 subjected to transverse compression. Dashed line after the maximum stress value.

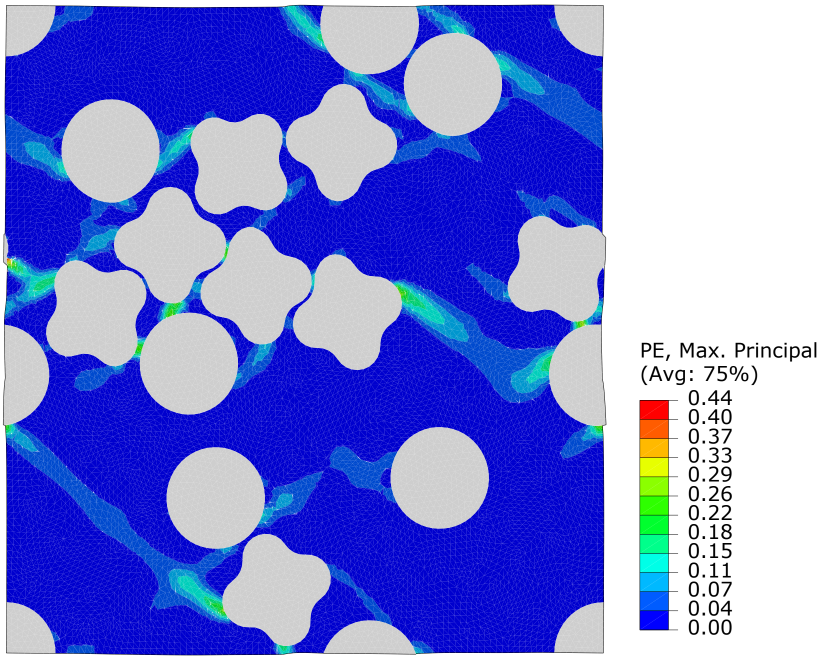


Figure 3.13: Deformation of the SVE from Figure 3.9 at maximum load-carrying capability, subjected to transverse compression. The plotted variable is plastic strain PE within the matrix, highlighting the formation of several shear bands. Deformation is shown unscaled.

3.2. Neural Networks

The second methodology to be leveraged in the proposed 'smart' framework is Neural Networks (NNs). Within the framework, NNs are used to create a surrogate model, based on the results from the Computational Micromechanics simulations. While the Design of Experiments (DoE) defines which combinations of parameters are simulated, the NNs take the role of creating an analytical fit to the resulting high-dimensional datapoints. In the terminology of *Machine Learning*, this is referred to as *Non-linear Regression*, borrowed from statistics. To be able to understand the adaptation and implementation of NNs for the proposed framework, we shall explain them in the following sections.

First, we introduce and define the term of *Neural Networks*, and situate them within the larger field of *Machine Learning*. The distinction between algorithms for so-called *supervised* and *unsupervised* learning is explained, and why our framework requires an algorithm of the former kind. After a brief overview of various NN architectures and alternatives, we can motivate our choice for the popular Multilayer-Perceptron (MLP), and begin with the explanation of its fundamental yet simple processing unit, the *Perceptron*. Based on this, we can derive architecture of the MLP and briefly summarize its so-called *training process*. The latter is the key part of *any* NN: the fitting of the NN to a given dataset in order to create a smooth approximation function. At this point, we have covered the key NN parameters: **hidden_layer_sizes**, **activation**, **solver**, and **tol**. These are later discussed in Chapter 5, when dealing with the challenge of automation and implementation of the NNs into our framework.

Finally, we conclude the section with a relevant example case from literature: a NN is trained on data from Finite Element Analysiss (FEAs) of different variants of an aircraft structure, and thereby allows to optimize the structure for stiffness. While the example uses the same idea of a combined DoE-FEA-NN methodology, we highlight and focus here on the NN implementation and its challenges.

3.2.1. Introduction and Terminology

In the 1900s, the working of the brain was found to revolve around a fundamental building block, called the *neuron* [11]. Estimated to occur around 86 Billion times in the human brain [80], each neuron is a simple unit able to receive, process, and transmit information in an all-or-nothing way to other neurons. The ensemble of interconnections of neurons in the brain can be referred to as a *Biological Neural Network*, and by fact allows us to learn from experience in order to perform a multitude of tasks.

With the advent of the first computer hardware in the 1940s, the idea to mimic the structure of the brain as a "highly complex, non-linear, and parallel computer" [29] was born and the first theoretical models of an *Artificial Neural Network* were developed and computationally implemented [61]. Today, the following definition has been proposed by [29]:

> "A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural prospensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:
>
> 1. Knowledge is acquired by the network from its environment through a learning process.
> 2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.
>
> The function used to perform the learning process is called a *learning algorithm*, the function of which is to modify the synaptic weights of the network in an orderly fashion to attain a desired design objective."

In the following and throughout this master's thesis, the term *Neural Network* will be used referring to an *Artificial Neural Network*. Furthermore, NNs are grouped under a broader field within the computer sciences, termed *Machine Learning* and just like all *Machine Learning* algorithms they can be subdivided according to two simple questions:

- *How* do they learn from experience? With or without supervision?
- *What* do they learn to do with the input they are given? To classify it in predefined categories? For example to judge whether an image shows a cat or a dog. Or to learn to approximate the non-linear functional relationship between the given inputs and outputs? For example, how the buckling load of a composite panel depends on the number and wall thickness of the stiffeners.

The following section shall briefly define the term of *Machine Learning*, and allow the reader to understand where NNs fit within. Furthermore, the afordescribed distinction between classification and functional approximation will be explained in more detail. This will then allow us to understand how a NN can be put to use with Computational Micromechanics: for *Non-Linear Regression* through *Supervised Learning*.

3.2.2. Neural Networks within Machine Learning

Who defined Machine Learning, and as what?
The term *Machine Learning* first appeared scholarly in 1959, in A.L. Samuel's paper titled *Some studies in machine learning using the game of checkers* [63]. While drawing on previous research from the 1950's on the same problematic, he was the first to cast the basic idea of "programming computers to learn from experience" [63] into the governing term still used today: *Machine Learning* (ML).

Since 1959, significant advances have been made in this field. A popular example for impactful application of ML algorithms is self-driving cars: based only on the visual input of cameras [8], NNs have been successfully trained to drive a car without human interaction. Further ML breakthroughs were achieved in object and speech recognition [37] [32].

For this brief introduction to ML algorithms, it suffices to see them as the mathematical functions that they are. Given a set of independent input parameters, they return a set of dependent output values. A ML algorithm is not useful when the exact mathematical function to be represented is known, instead their field of application is for problems where this is difficult or impossible. Recalling the task of driving a car in traffic, the number of input parameters to consider is extensive, and the relation between these input parameters and the final output, i.e. the steering commands, today is impossible to find as an exact mathematical function.

ML algorithms don't aim to find the exact formulation, but to *approximate* the underlying input-output relationship with sufficient accuracy. To achieve this, and contrary to a traditional programming approach, the ML algorithm shall not have to contain *explicit* rules such as

```
if obstacle:
    break
```

Instead, ML algorithms are designed to be able to *learn* all necessary rules from experience, as independent as possible from the programmer. *How* the algorithm is learning from experience provides a useful division into two major classes: *supervised learning*, and *unsupervised learning*.

What is supervised learning, and how is it different than unsupervised learning?
The process of an algorithm learning to approximate a sought-after input-output relationship is classified as one of the following two [46]:

- *Supervised learning*: In this case, the sought-after relationship is only known through a limited amount of discrete datapoints, i.e. known pairs of matching input-output data. Analogous to polynominal interpolation, these datapoints could be called the interpolation points. In the terminology of ML, they are called *training data*. For the example of self-driving cars, the training data would consist of input data, e.g. speed, position, obstacles, as well as the matching output data, i.e. the corresponding correct steering commands as given by a human.

  During supervised learning, the ML algorithm adapts its intrinsic parameters, based on mathematical methods, in order to reproduce the shown training data as accurately as possible. Lastly, the so-trained algorithm can be used to predict the output values for input data which it has not been trained with. Two major use cases are distinguished:

  – The allowable output values are discrete, e.g. the self-driving car algorithm has to predict whether the detected object in its way is a child or an ape. This problem is called *Classification*.

  – The allowable output values are continuous, e.g. the algorithm has to predict what the correct steering rates are in order to swerve around the obstacle. Analogous to statistics, this is called *Regression*. For a non-linear input-output relationship, it is called *Non-linear Regression*.

- *Unsupervised learning*: In this case, the sought-after relationship is completely unknown, i.e. no training data is available. These methods are usually referred to as *data mining*, as they focus on extracting useful information from the input data itself. Two major cases are distinguished:

  - Based on a user-specified *similarity measure*, the algorithm attempts to classify the data points into different groups, i.e. clusters. This is called *Cluster analysis*.
  - Based on prior approximate knowledge of what the most important features within the input data are, the algorithm attempts to reduce the dimension of the input data. This is called *Dimensionality Reduction*.

Which kind of learning process and use case is relevant for the 'smart' framework to be developed in this thesis?
NNs can be used both with unsupervised and supervised learning, for both *Classification* and *Non-Linear Regression*. The most prominent success cases for NNs, self-driving cars [8], object recognition from images [37], and speech recognition [32], are largely based on NNs trained through *supervised learning* and performing *Classification* tasks.

In the case of the proposed framework, we want to predict the mechanical properties of a composite material, based on training data generated by a Computational Micromechanics model. Accordingly, a *supervised learning* approach has to be used, and since mechanical properties are continuous, the necessary way of predicting them is by *Non-Linear Regression*. To achieve this with a NN, the network first has to be trained with pairs of matching input-output data, before it is able to make predictions on input data that it has not been trained with, a process called *generalization*.

Before explaining the simple mathematical foundations of NNs, we shall briefly briefly compare two popular NN architectures with an alternative ML algorithm, and motivate our choice of the so-called *Mulitlayer-Perceptron* architecture.

3.2.3. Overview of Neural Network Architectures and Alternatives

Based on the literature review, three common methods for *Non-Linear Regression* were identified:

- *Multilayer-Perceptron (MLP)*
- *Radial Basis-Function (RBF)*
- *Support Vector Machines (SVM)*

The former two are both NN architectures with inherent differences in: mathematical model of the simple processing units; allowed arrangement of these units (e.g. number of *hidden layers*); allowed interaction between the units (e.g. linear or non-linear); possible training algorithms. Among these three methods, the MLP has established itself as a reference, being a versatile "global approximator" [29] both for *Classification* and *Non-linear Regression*. More specifically, for the case at hand of predicting the mechanical properties of a composite material through a NN, with almost no exception all reviewed cases in literature have used the said MLP architecture.

Why use a Multilayer-Perceptron, and not a different architecture or method?
As aforementioned, the MLP is a standard ML algorithm which has achieved sufficient prediction accuracy in similar cases. The work of this master's thesis now aims at *extending* the use of ML within the field of Computational Micromechanics, and *not* to try to introduce a new kind of algorithm. Advantages of the MLP over the other methods were reported as the following:

- *MLP vs. RBF*: [29] summarizes that for "approximation of a nonlinear input-output mapping, the MLP may require a smaller number of parameters than the RBF network for the same degree of accuracy".
- *MLP vs. SVM*: [29] summarizes that "SVM are currently slower than other neural networks (e.g. multilayer perceptrons trained with back-propagation algorithm) for a similar generalization performance."

3.2.4. The Single Perceptron

In order to understand how a Multilayer-Perceptron can be trained to approximate a non-linear function, we first have to explain the basic processing unit it consists of: the *neuron*, a terminology borrowed from *Biological Neural Networks*. In the case of the MLP, the mathematical model of each neuron is the *Perceptron*. First proposed in 1958 as a "probabilistic model for information storage and organization in the brain" [61], its architecture is illustrated as a block-diagram in Figure 3.14 [29]. One or several inputs $x_1, x_2, ...x_m$ on the left hand side are processed through three simple units into a *single* output y:

- The so-called *synapses*, again in biological analogy, are the connections from the inputs $x_1, x_2, ...x_m$ to the next unit, the *summing junction*. Each synapse takes its input value, multiplies it with the so-called *synaptic weight* w, and passes it further to the summing junction.

- The *summing junction* first sums up all inputs coming from the synapses, each with their synaptic weight, and then adds a value b, called *bias*. Its output v is called *local field*.

$$v = \sum_{j=1}^{m} (w_j x_j) + b \tag{3.7}$$

- The *activation function*, $\varphi(v)$ takes as input the local field and limits the amplitude of the output of the neuron y. Different types of activation functions can be chosen: threshold, piecewise-linear, or sigmoid. A key requirement of the MLP learning algorithm is that the activation function must be differentiable. A common choice is therefore the logistic function of the sigmoid class, with $a > 0$ denoting its *slope parameter*, and so limiting y to the range of $0 \le y \le 1$. [3]

$$\varphi(a, v) = \frac{1}{1 + e^{-av}} \tag{3.8}$$

Putting together these three basic units, we obtain the single formula describing the perceptron:

$$y(x_0 = 1, x_1, x_2, ..., x_m, w_0, w_1, w_2, ..., w_m) = \varphi(\sum_{j=0}^{m} (w_j x_j)) \tag{3.9}$$

For simpler annotation, we have recast the bias b as the synaptic weight w_0 by defining $x_0 = 1$, as shown in Figure 3.14. Since the number of inputs $x_1, x_2, ...x_m$ and the single output y are defined by the training data, the only free parameters of the perceptron are the *synaptic weights* $w_0, w_1, w_2, ..., w_m$. Thus, the only way the perceptron can be fitted to a set of training data is by adjusting these *synaptic weights*, much like a polynomial is fitted to a curve by adjusting the polynomial coefficients.

Contrary to a polynomial though, the perceptron is very limited in the kind of non-linear functions it can fit: Only functions which resemble the simple *activation function* can be fitted by a single perceptron. The common choices for the activation function like threshold or sigmoid illustrate that the perceptron was primarily conceived for *Classification* tasks. For *Non-linear Regression*, it performs very poor.

However, it was found that when connecting *several* perceptrons in a certain way, the resulting Multilayer-Perceptron can indeed fit almost any non-linear function. This leads straight to the next question: *How* are these perceptrons arranged to form a Multilayer-Perceptron?

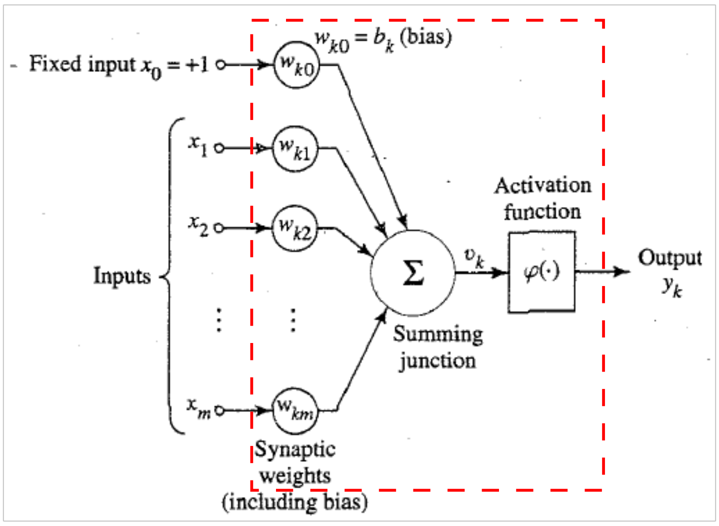


Figure 3.14: Example architecture of a Perceptron, with its overall boundary demarcated in red. Reproduced from [29].

[3] As to be discussed later-on, this range determines the scaling of the training data.

3.2.5. From Single Perceptron to Multilayer-Perceptron

Every neuron in a MLP is an identical copy of the same basic processing unit, the percepton of Figure 3.14. These identical neurons are then arranged in *layers* to form a MLP, as shown in Figure 3.15. The most simple network, a *single-layer*, consists only of the inputs, and the output layer. More layers can be added in between of the two, called *hidden layers*, with so-called *hidden neurons*. The resulting networks are called *multilayer* networks, with Figure 3.15 being an example with two hidden layers.

The number of neurons in the input and output layer is equal to the number of input/output parameters of the function to approximate, but the number of hidden neurons in each hidden layer is a parameter to be determined by the user. In general, the more complex the input-output relationship to be approximated, the more hidden layers and hidden neurons are necessary [15].

The propagation of the inputs through the network is generally from left to right, from layer to layer, which is called a *feed-forward* network. This means that the inputs pass initially to the neurons of the first hidden layer, then the second hidden layer, and so forth until they reach the neurons of the output layer. [4]

Concerning the interconnection between neurons, this means that neurons within the *same layer* are not connected with each other, as shown in Figure 3.15. Instead, each neuron of a given layer receives as input only the outputs from the preceding layer. For the example of Figure 3.15, any hidden neuron in the first layer receives all inputs of the input layer. The same holds true for all neurons in the network: Each neuron receives *all* the outputs from the preceding layer, the network is so said to be *fully-connected*. The opposite case, when some of these connections are omitted, is called *partially connected*. Further, this is a way to incorporate prior information, about the function to be approximated, into the neural network [39].

For brevity, the number of layers and neurons in a NN can be summarized with the following annotation: $i - h_1 - h_2 - ... - o$, where i is the number of input nodes, h_1 the number of hidden neurons in the first hidden layer, h_2 the number of hidden neurons in the second hidden layer and so forth, and o is the number of output neurons. Accordingly, the network shown in Figure 3.15 is a fully-connected, feed-forward, MLP, 4-4-4-3 network. Having discussed the possible architectures of a MLP network, it is now pertinent to examine the main question:

How can a MLP network store experiential knowledge?
Recalling the definition of a NN [29] in section 3.2.1, it is by modifying the synaptic weights of all the interneural connections within the network, that it learns to approximate the sought-after input-output relationship. This process is called *learning* or *training*, and is automated by a so-called *learning/training algorithm*. For the case of the single perceptron, the number of synaptic weights equals the number of inputs. When connecting several perceptrons to a MLP, the training effort increases sharply, since every perceptron now has its own set of synaptic weights which shall be modified.

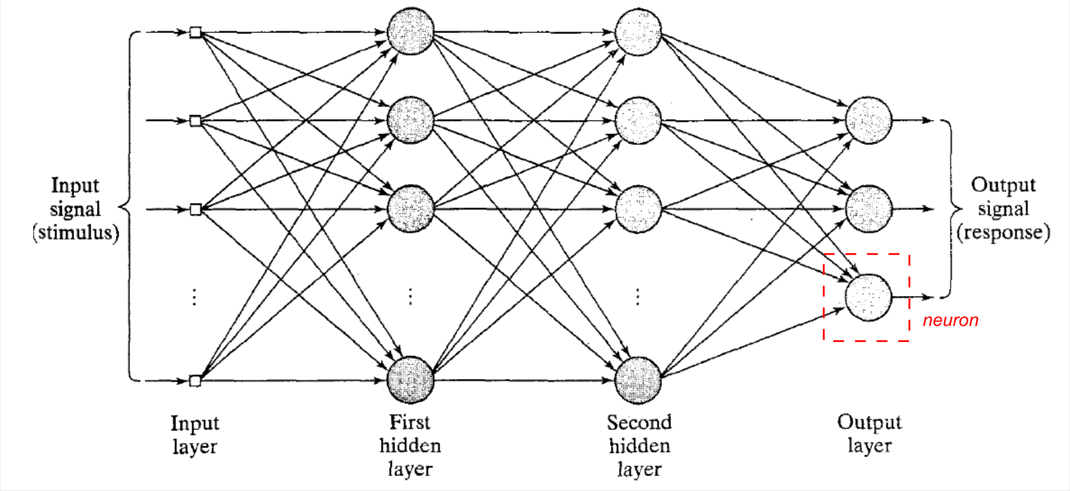


Figure 3.15: Example architecture of a Multilayer-Perceptron. A single neuron, i.e. Perceptron, is demarcated in dashed red lines, corresponding to Figure 3.14. Reproduced from [29].

[4] In contrast, networks where the output of a layer is fed back to a *preceding* layer, i.e. a feedback loop, are called *recurrent networks*.

How to choose the number of hidden layers/neurons?
It has to be pointed out that the learning process modifies only the synaptic weights, not the number of hidden layers/neurons of the network. The architecture of the NN therefore has to be pre-defined by the user before starting the learning process. How to decide on the optimal number of hidden layers/neurons does not follow exact rules unfortunately, but generally requires "an exhaustive experimental study" [29]. A few general trends however can be summarized:

- As aforementioned, more neurons achieve better approximation [15], up to a certain point of convergence that depends on the complexity of the input-output relationship. Said example of a NN trained to steer a car only based on visual input, required 27 Million connections and 250000 parameters [8]. Contrary, for the reviewed cases where NNs were used to predict the mechanical properties of a composite material, e.g. buckling load, usually not more than two layers with each less than 10 neurons, or a single layer of up to 50 neurons were necessary to capture the material behavior [7]. The concomitant drawback of a higher number of hidden layers and neurons is a steep increase in computation time required for training.

- Inputs which are more important for the input-output relationship, relative to the other inputs, require a larger number of neurons for their representation in the network [3].

After having chosen the number of hidden layers and neurons, their synaptic weights have to be modified during the so-called training process whose basics are explained in the following section.

3.2.6. The Multilayer-Perceptron Training

For the prediction of mechanical properties of composite materials, the approach of *supervised learning* will be used to train the MLP network (as stipulated in section 3.2.2). First, the architecture of the MLP network has to be fixed: number of hidden layers, number of hidden neurons. Likewise, initial values for the synaptic weights and biases of the network have to be chosen, as well as the type of activation functions for the perceptrons. Introducing already the terminology to be used in Chapter 5, these are the parameters **hidden_layer_sizes** and **activation**. Then, the first set of the training data, i.e. input data for which the corresponding output values are known, is passed through the network. The input data is propagated from the input layer through the hidden layers to the output layer, a process referred to as the *forward-pass*. Given the initial values for synaptic weights and biases, the network's output y can be readily calculated. Then, this result is compared to the target output values of the fed training data. The difference between the two defines the so-called *total error energy E* of the network. Then, the minimization of this term up to a certain tolerance **tol** becomes the main objective of the training.

How can the total error energy of a MLP network be minimized?
Accordingly, this can be seen as a continuous optimization problem with the objective being to minimize a *loss function*, in this case the total error energy E of the network. Since the number of hidden layers and neurons, as well as the type of activation function are fixed prior to the training, the loss function at this point only depends on one set of continuous scalar parameters: the synaptic weights of all neurons in the MLP network. In more simple terms, the *training* of a MLP network boils down to the iterative adaption of its synaptic weights w, in order to minimize the difference between target output (from training data) and current output calculated by the network. For the minimization of E, the standard choice are gradient-based optimization algorithms. This is not a coincidence, but taking advantage of the MLP architecture.

MLP networks are mathematical functions $y = MLP(x, w)$ which can be classified as a *directed graph.* A peculiarity of these is that *gradients* of the loss function with respect to its parameter w can be easily calculated through application of the chain rule. This approach was conceived and first proposed in 1986 [62], termed the *error back-propagation algorithm*, which today has become the most popular training algorithm for MLP networks [29], foremost due to its computational efficiency.

When depicting a MLP network as a directed graph, the application of the chain rule can be visualized as a *back-propagation* of the error signal through the network, starting from the output layer and moving backwards layer-by-layer. To clarify the terminology, *back-propagation* is therefore an efficient *method of computation of gradients* in directed graphs, such as MLP networks. However, back-propagation does not say anything about the *rule* by which the parameters of the network shall be updated, in order to minimize the output error. In the general field of mathematics, such rule is referred to as an *optimization method*, while in the field of Neural Networks, this has been termed the *learning rule*, or **solver**. Popular gradient-based learning rules implemented for MLP training are *Gradient Descent* and *Stochastic Gradient Descent*, both using first-order gradients, and the *Broyden–Fletcher–Goldfarb–Shanno* (BFGS) algorithm [22] using second-order derivatives of the loss function.

For reasons of brevity we have to omit a detailed mathematical derivation of the MLP training at this point, and refer the interested reader to the literature review. On a concluding note, it has to be remarked that the methodology of training and testing a MLP network is not subject to exact rules. Many different approaches have been developed, e.g. depending on the number of known input/output data, the nature of the data (experimental with scatter, or simulation results without scatter), etc. More details of this will be discussed in Chapter 5, when explaining the training process of the NN implemented for the proposed 'smart' framework.

After having explained the fundamentals of NNs, their architecture and training, we shall conclude this chapter with an applied example from research on composite materials, using NNs in combination with Finite Element Analyses (FEA) to predict the stiffness of a composite structure.

3.2.7. Case Study: Material Property Prediction through a trained Multilayer-Perceptron

The literature review summarizes the rather recent history of NNs applied within the field of composite materials. The necessary foundation was laid in 1986 with the successful development of the error backpropagation algorithm for MLP networks [62], yet it took until the late 1990s for the first publications on NNs to appear in the field of composite materials. Due to very limited computational capabilities, they were restricted to train the NNs on experimental data, often containing too few datapoints to obtain a reliable generalization of the mechanical properties they were trying to predict [40]. With computational power and interest in *Machine Learning* on the rise, the past few years have shown the first success cases of NNs trained on datapoints generated by simulation. Hereby, it was possible to predict various complex mechanical properties of composite materials such as their damage behavior [55] or their ballistic limit [4].

This kind of approach has been the inspiration for the work of this master's thesis, as we propose to apply the same combined methodology to the field of Computational Micromechanics, specifically including damage modelling. To illustrate the state of the art found in literature, we shall briefly present one recent success case: the optimization of a thin-walled composite structure through a NN trained on FEA data [23].

- **What is the application?**

  A thin-walled composite structure, used as part of the elevator in an aircraft, shall be designed for maximum stiffness and minimum weight. Possible design choices, i.e. input parameters, are the geometrical features of the structure. Figure 3.16 shows the Finite Element Model of the structure.

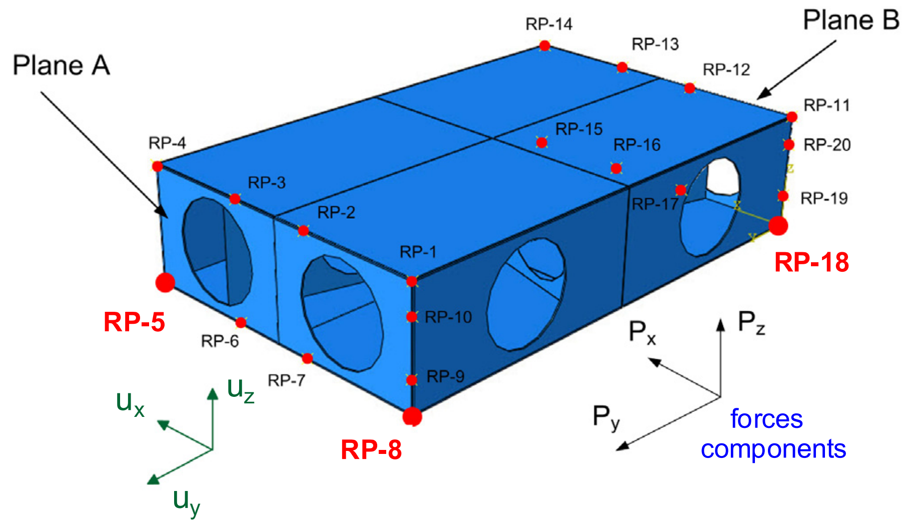


Figure 3.16: FEM of the composite structure. Reproduced from [23].

- **What was predicted?**

  A single continuous output was predicted: The stiffness of the structure, defined as the displacement u_{max} at a specific point of the structure under the applied loading.

- **What was the available data for training/testing of the NN?**

  The NN was trained/tested with FEA data, consisting of 160 simulations with different combinations of the input parameters, generated according to Design of Experiments (DoE) principles.

- **What was chosen as the inputs and outputs of the NN?**

  The extracted output from the simulations was the stiffness of the structure, as well as its weight as a secondary result. Four geometrical input parameters were chosen, as illustrated in Figure 3.17, and with ranges as summarized in Table 3.4.

Input Parameter	Range
Rib Wall Thickness	0.2...2 mm
Rib Angle α	0°...82.5°
Side Hole Diameter	0...25 mm
Rid Hole Diameter	0...25 mm

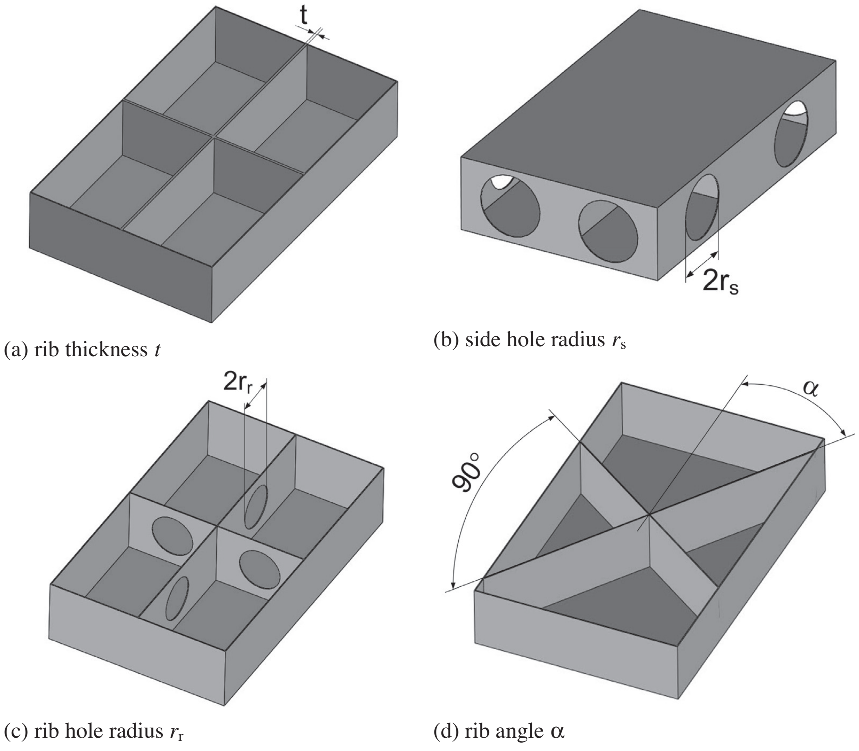
Table 3.4: Input parameters and ranges of the dataset to which the NN is fitted [23]

- **What was the chosen NN architecture?**

  A fully connected, feed-forward multilayer-perceptron was chosen as NN architecture.

- **Which kind of learning rule was used?**

  Only one kind of learning rule was investigated, Limited Broyden–Fletcher–Goldfarb–Shanno [41], an implementation of the BFGS algorithm that minimizes memory usage.



(a) rib thickness t

(b) side hole radius r_s

(c) rib hole radius r_r

(d) rib angle α

Figure 3.17: Geometric input parameters of the FEA/NN. Reproduced from [23].

- **How many hidden layers and neurons were chosen?**

  The number of hidden layers was kept constant to one. The number of neurons in the single hidden layer was varied from three to ten, as well as different choices for the activation function in the hidden and output layer were tried. Table 3.5 summarizes the results with $B_{testing}$ denoting the coefficient of determination familiar from statistics. A maximum value of $B_{testing} = 1$ is equal to a perfect fit to the testing data. Comparing the different parameter choices, no main trend can be observed, but it can be noted that a simple 4-3-1 MLP network was able to achieve the best fit.

Network	$B_{testing}$	Activation (hidden)	Activation (output)
4-5-1	0.953	Logistic	Logistic
4-3-1	0.94	Logistic	Exponential
4-10-1	0.94	Exponential	Tanh
4-8-1	0.95	Logistic	Exponential
4-3-1	0.96	Logistic	Exponential

Table 3.5: Comparison of the quality of the NN fit, for varying number of hidden neurons, and different activation functions. $B_{testing}$ denotes the coefficient of determination familiar from statistics, with a maximum value of $B_{testing} = 1$ equal to a perfect fit to the testing data. Reproduced from [23].

- **What was the final predicted result?**

  The final result is the trained NN, which is an analytical approximation of the stiffness of the structure, as a function of all five input parameters as listed in Table 3.4. Since it is difficult to plot 5-dimensional data, Figure 3.18 shows by way of example the determined maximum displacement u_{max}, plotted over rib hole radius and rib angle α, just two of the five input parameters. The remaining parameters, side hole diameter and rib wall thickness, were fixed at certain values. The same plot can be created for any other combination of the input parameters, thereby allowing to understand the underlying relationship between the input parameters and the output. Lastly, the obtained analytical continuous function allows any kind of common mathematical operation on it, with finding the global maximum/minimum just being the most simple one.

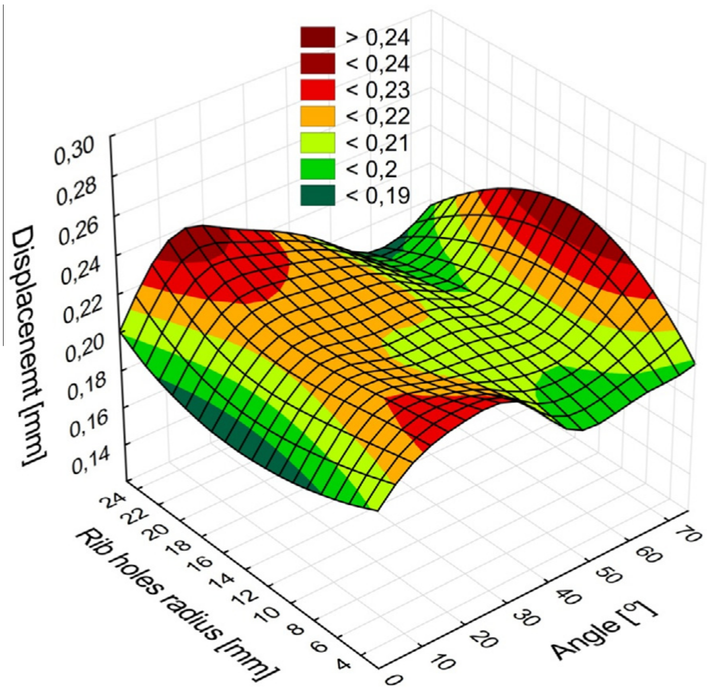


Figure 3.18: NN approximation of the maximum displacement u_{max} of the composite structure, depending on the rib hole radius and rib angle α. Reproduced from [23].

- **What were the main conclusions?**

  The main goal of predicting the overall response of the composite structure through a synergy of DoE, FEAs and NNs was achieved. Trained with the results of 160 FEAs, the MLP network was used to create a surrogate model which relates the stiffness of the structure with the relevant five geometric input parameters through an analytical function. The latter can then easily be exploited for optimization of the structure through common mathematical operations. What was not discussed in this paper was the accuracy of the surrogate model. It is missing a discussion on how the number of necessary datapoints was determined, and by which measure the combinations of input parameters were generated, i.e. a regular grid filling the parameter space, or a random distribution. Overall however, the outlined case served as an inspiring example of the kind of automated and comprehensive surrogate model that can be achieved by the used methodology combining Design of Experiments, Computational Modelling, and Neural Networks.

3.2.8. Summary

In this second part of the chapter on the state of the art, we have introduced Neural Networks (NNs) as the other key methodology to be put to use in the proposed 'smart' framework. The mathematical fundamentals of the chosen Multilayer-Perceptron architecture have been laid out together with its training process. This will serve as the basis for Chapter 5, discussing the practical challenges of implementing and automating the NN training for the framework. Lastly, we presented a recent example from literature which served as an inspiration for the work of this master's thesis. Herein, a NN was trained on 160 datapoints generated by Finite Element Analyses to create a 5D surrogate model of the stiffness of a composite structure.

3.3. Summary

Hereby, we conclude the chapter on the state of the art. We have explained in all necessary detail two of the three methodologies to be combined for the framework of this thesis: Computational Micromechanics (CM), and Neural Networks (NNs). The third methodology, Design of Experiments is considered well-established and touched upon later in Chapter 6. In the first part, we explained the CM model that is the fundamental element of the proposed framework. It is used to generate the datapoints describing the transverse strength of the modeled material, and it is exactly these datapoints to which the NN will be fitted. In the second part, we explained the theory and practice of NNs, as necessary to understand by which principles they can approximate a non-linear relationship such as the transverse strength of a UD FRPC ply.

The following two chapters shall now discuss the challenges and implications of adapting these methodologies, CM and NN, to the framework proposed in this thesis. Foremost, the entire framework has to run automatically without any user interaction required. This is due to the fact that an accurate surrogate model of a complex material property, such as transverse strength, requires many datapoints to rest upon. As to be discussed in the following chapter, every datapoint further requires several simulations to obtain a statistically significant result. In consequence, thousands of simulations have to be generated, run and post-processed automatically. Lastly, also the NN training has to be fully automated since the NN approximation is used to judge whether more datapoints are necessary to improve the quality of the surrogate model.

After having addressed the automation of the CM and NN part in Chapter 4 and Chapter 5, we can then proceed with the presentation of the workflow and implementation of the entire framework in Chapter 6. Lastly, we may present five example studies performed with the 'smart' framework in Chapter 7.

4
Automation of Computational Micromechanics

This chapter shall detail the automation of the Computational Micromechanics (CM) simulations, necessary to generate the fundamental database of the proposed framework. Based on this database, the final surrogate model can then be constructed through a Neural Network. The underlying CM model discussed in this chapter has been introduced in Chapter 3, Section 3.1. First, an overview of the simulation workflow is given, listing its four main steps: 1) Generation of a Statistical Volume Element (SVE), 2) Definition of the Finite Element Model in *Abaqus*, 3) Running of the simulation, 4) Post-Processing of the simulation results.

For each step, we discuss on a very practical level the used tools and data formats, as well as encountered challenges when adapting the simulation workflow to full automation. Furthermore, we point out the parameter choices which the user has to make at each step. Eventually, it is exactly these parameters which we later vary and study in Chapter 7 with the proposed framework. Therefore, we also discuss and justify the nominal values which are assumed for the baseline CM model, such as mesh size, SVE size, and minimum inter-fiber distance.

Lastly, the damage modelling introduces significant numerical complexity into the CM model, resulting in a statistical spread of the results depending on the fiber distribution in each SVE. In consequence, simulations for a single combination of model parameters have to be repeated several times for different SVE realizations in order to obtain a statistically significant result. The implications of this on the post-processing strategy are explained with the aid of some example simulation results in the last section of this chapter.

4.1. Overview of the Simulation Workflow

Figure 4.1 illustrates the different steps of the workflow of the CM simulations:

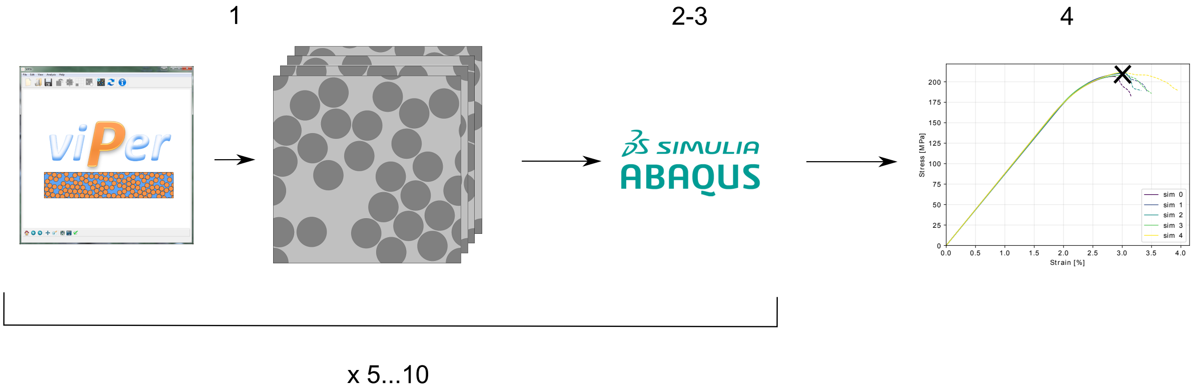


Figure 4.1: Workflow of the Computational Micromechanics simulations. Step 1: Generation of a SVE through *Viper*. Step 2 & 3: Definition of the FEM in *Abaqus* and execution of the simulation. Step 4: Post-Processing of the simulation results. Steps 1-4 are repeated until 5 valid SVE results have been obtained, but no more than 10 attempts are made.

1. Generation of a SVE of the UD FRPC ply in question: This is automated through the tool *Vip3r* [30], developed by M. Herraez for *IMDEA Materials*, and implemented in *Python*. The result of this step is a **.txt* file describing the geometry of the SVE and listing the fiber materials.

2. Definition of the FEM in *Abaqus*:

   - Generation of mesh based on SVE geometry
   - Association of constituents with their material properties and failure models
   - Generation of periodic boundary conditions
   - Definition of thermal step prior to mechanical loading
   - Definition of transverse mechanical loading step: tension or compression

   This process is also automated through a *Python* script developed at *IMDEA Materials*. The result of this step is a **.cae* and **.inp* file defining the *Abaqus* simulation to be run, and a **.txt* file defining the periodic boundary conditions to be applied.

3. Running of the simulation: The simulation is run in parallel on 6 Central Processing Units (CPUs) and with a *Fortran* subroutine to stop the simulation as soon as the load drops by more than 10%, with respect to the maximum load attained.

4. Post-Processing of the simulation results: A multitude of results are stored for each simulation, but for now the final data of interest is the maximum load obtained during the transverse loading, i.e. the maximum transverse tensile/compressive strength of the UD FRPC ply in question.

This procedure for a *single* SVE finally has to be repeated for 5 to 10 different SVE realizations, each of a statistically different fiber distribution and in consequence resulting in a slightly different transverse strength value. Details of this will be discussed in Section 4.4.

4.2. Step 1: Generation of a Statistical Volume Element

First, a SVE of the UD FRPC ply to be modelled has to be generated. For this, the *Vip3r* tool from M. Herraez [30] has been used, offering a wide array of geometrical input parameters to be varied, see Section 3.1.2.1.

4.2.1. SVE Generation through *Vip3r*

For the proposed framework, thousands of SVEs have to be generated automatically, so the *Python* script behind *Vip3r* had to be used instead of the GUI. To automate the creation of different kinds of SVEs, the *RVE.py* file was modified, with code listing 4.1 showing the part added to the script. Line 2 displays the geometrical input parameters which can be varied with this specific implementation:

- *Dmin*: minimum inter-fiber distance
- *FVolF*: total fiber volume fraction
- *RFiberFrac*: fraction of round fibers (the remainder of fibers is considered lobular)
- *LobularDiam*: diameter of the lobular fibers (round fibers are kept at a nominal diameter of 8 µm)
- *Height*, *Width*: height and width of the SVE

The remainder of the code listing illustrates how *Vip3er* can be utilized to automatically generate SVEs based on these input parameters. *Vip3r* is able to generate a wide array of SVE geometries, for different fiber shapes, and up to very high fiber volume fractions of >70%. This is achieved by various algorithms implemented in the tool, including compaction and stirring steps. The iterative fiber placement algorithm further has to find a SVE geometry which satisfies opposing conditions, like high fiber volume fraction and minimum inter-fiber distance. As a result, the final SVE proposed by *Vip3r* may in some cases have overwritten one of the conditions, therefore requiring some additional verification measures as to be explained in the following section.

4.2.2. Measures of Valid SVE Generation

Three errors have to be caught in order to automatically generate valid SVEs with *Vip3r*:

- First, it has to be verified that *Vip3r* actually complies with the minimum inter-fiber distance as defined through variable *Dmin*, see line 35. While this generally goes well for circular fibers, it was found that for mixing of different fiber shapes, the condition might never be met. Specifying initially $Dmin$ = 0.35µm constantly resulted in a final $Dmin \leq 0.2$µm, in the case of mixing round and lobular fibers. As mentioned before in Chapter 3, Section 3.1.2.1, it is important to maintain a constant value for *Dmin* in order for transverse strength results to be comparable. Further details of this will be discussed in Chapter 7, Section 7.3.2.

- Second, fibers may overlap after the iterative fiber placement process has come to an end. This condition is checked in line 40.

- Third, in rare cases the final SVE may miss some of the periodic fibers on its edges. This can be verified by calculating the nearest neighbour for every fiber, see line 46, which results in an error if periodic fibers are missing.

These three errors occur at different frequencies, and were only discovered iteratively during the work of this master's thesis. The 'smart' framework proposed in this thesis requires a fully automated simulation scheme which can generate thousands of valid SVEs without user interaction. For that, a very necessary requirement is to detect and circumvent all of the errors described above, and furthermore to do so in an automated way. Fortunately, the implementation of these countermeasures proved to be simple yet effective.

```
1  ...
2  def main(Dmin,FVolF,RFiberFrac,LobularDiam,Height,Width,directory,filename):
3  
4      L = Width   # 8.15
5      H = Height   # 8.15
6      rve_size = (L, H)
7      RFiberFrac = np.around(RFiberFrac,decimals=1)
8  
9      # Define fibersets
10     fibersets = [{'Geometry': CIRCULAR, 'Parameters': [], 'Vf': FVolF * RFiberFrac,
11             'Phi': Statistical(UNIFORM, (0., 180.)),
12             'df': Statistical(NORMAL, (8.0, 0), not_negative=True),
13             'Material': Material.List[1]},   # CF AS-4
14 
15            {'Geometry': LOBULAR, 'Parameters': [4], 'Vf': FVolF * (1 - RFiberFrac),
16             'Phi': Statistical(UNIFORM, (0., 180)),
17             'df': Statistical(NORMAL, (LobularDiam, 0), not_negative=True),
18             'Material': Material.List[1]}]
19 
20 
21     tinit = time.time()
22     dn1 = [0.00001]
23     while not Dmin <= min(dn1) <= Dmin*1.1:
24 
25         # initialize myMicrostructure.valid()
26         myMicrostructure = Microstructure(rve_size=rve_size, fibersets=fibersets, gen_algorithm=(DYNAMIC,)
    ,                                     tolerance=Dmin, optPeriod=FULL)
27 
28         # while fibers are overlapping: regenerate SVE
29         while not myMicrostructure.valid():
30             myMicrostructure = Microstructure(rve_size=rve_size, fibersets=fibersets, gen_algorithm=(
                                               DYNAMIC,), tolerance=Dmin, optPeriod=FULL)
31 
32         # now fibers are NOT overlapping: analyze dn1
33         # BUT! SVE can in rare cases be non-periodic (periodic fibers missing!)
34         try:
35             dn1 = myMicrostructure.analyzeNearestNeighbour(neighbour=1, show_plot=False)
36         except IndexError, message: # if NOT periodic
37             print 'ERROR: SVE not periodic; regenerated'
38             dn1=[0.00001]
39         print 'Dmin: ' + str(min(dn1))
40         ...
```

Listing 4.1: Modification of *RVE.py* script to automate and parameterize the SVE generation

4.2.3. Final Result

The final result of Step 1 of the simulation workflow is a **.txt* file describing the SVE geometry and denoting the material of each fiber. An example **.txt* file is shown in listing 4.2. Most parameters are self-explanatory. *PHI* denotes the rotation of the nominal fiber cross-section shape, thus only relevant for non-circular fibers. In the case of lobular fibers, *PARAMETERS* describes the number of lobes. *PERIOD* and *Nf* are descriptors of the periodic fibers on the edge of the 2D SVE, which are split up such that the SVE can be duplicated and put together like a jig-saw puzzle. By the end of Step 1, the user has made choices for the following input parameters:

- Geometrical Parameters:
  - minimum inter-fiber distance *Dmin*; nominally **0.35** µm
  - total Fiber Volume Fraction (FVolF); nominally **50%**
  - Mixing of fibers: either of different diameters, shapes, or materials; defined by mixing ratio, i.e. Round Fiber Fraction (RFF); nominally **1** (only round fibers)
  - SVE dimensions; nominally **50** µm x **50** µm
- Material Parameters:
  - Choice of materials for the fibers: **AS4-CF**

```
#  INDEX      MATERIAL     SIZE       PHI        x0        y0  PERIOD  Nf      SHAPE  PARAMETERS
       1        CF–AS4    8.000  +157.359    36.325    13.650       0   1   CIRCULAR
       2        CF–AS4    8.000  +158.160    30.469    48.640       2   1   CIRCULAR
       3        CF–AS4    8.000  +158.160    30.469    −1.360       4   0    LOBULAR       4.000
       ...
```

Listing 4.2: Final **.txt* file describing an example SVE with circular and 4-lobed fibers (shortened)

4.3. Step 2: Definition of the Finite Element Model in *Abaqus*

The objective of Step 2 of the simulation workflow is

- first, the translation of the SVE geometry and fiber materials defined in the **.txt* file of Step 1 into a Finite Element Model in *Abaqus*, and
- second, the definition of the loading which the aforedefined SVE geometry shall undergo, all the while respecting the periodic boundary conditions as described in Section 3.1.2.2.1.

Both steps are automated through a *Python* script developed at *IMDEA Materials*, which had to be adapted to the specific needs of this work. The following input parameters were implemented for this step:

1. Mesh size of the SVE; nominally **0.5** µm
2. Material properties of the matrix and the fiber-matrix interfaces; nominally as described in Tables 3.2 and 3.3
3. Magnitude and sign of the displacement-controlled mechanical loading step, imposed on the SVE during the simulation

4.3.1. Mesh Size

The mesh size was chosen such that nominally exactly one mesh element is placed between two fibers, at the point of minimum inter-fiber distance (nominally 0.35µm). This is in accordance with literature, and results in about 100 elements around the circumference of each fiber (nominal diameter 8µm), see Figure 3.9 for example. As discussed in Section 3.1.3, transverse failure is always triggered by interfacial debonding, which is why the mesh size around the interfaces, and at the point of minimum inter-fiber distance may influence the result for overall transverse strength of the SVE. As part of the work of this thesis, a mesh sensitivity study was performed. When only looking at maximum stress, its value depends much less than expected on the mesh size around and in between fibers. Even for a very coarse mesh, with only 6 mesh elements around the circumference of the fibers, the value for maximum stress did not change significantly.

4.3.2. Scaling of Matrix and Interface Strength

Some of the performed studies to be discussed in Chapter 7, involve scaling of the strength of the matrix and the fiber-matrix interfaces. The nominal properties of both are stated in Tables 3.2 and 3.3. The aim of the scaling is to increase or decrease the strength by a linear scaling factor.

For the cohesive elements of the fiber-matrix-interfaces, linear scaling is easily implemented. The normal and shear strengths N, S are just multiplied with the scaling factor, and the critical fracture energies G_N^c, G_S^c are multiplied with the square of the scaling factor in order to preserve the *shape* of the traction-separation curve. The initial elastic stiffness K is kept unchanged. Please refer to Section 3.1.2.2.2 and Figure 3.6 for further illustration.

The same concept has been applied to the material model of the matrix described in Section 3.1.2.2.2. All features defining the strength of the matrix were scaled such that the relative damage behavior remains the same, i.e. ratio between shear and normal strength, and only the overall strength is scaled linearly. By default, the material model of the matrix is more complicated, but can be scaled by applying the same principles than for the cohesive elements: linear scaling in the stress values defining the yield/compressive limit and hardening, and scaling to the power of 2 for the fracture energies.

In Chapter 7, when referring to the strengths of the matrix and fiber-matrix interfaces, only the scaling factor will be denoted. A scaling factor of 1 is equal to the material models defined in Section 3.1.2.2.2 and with values according to Tables 3.2 and 3.3. All other scaling factors not equal to 1 are calculated relative to these baseline values, according to the aforedescribed rationale.

4.3.3. Definition of Thermal and Mechanical Loading

As motivated in Section 3.1.2.2.3, a nominal thermal step prior to the mechanical transverse loading was implemented for all simulations. With a temperature differential of $\Delta T = 160\,°\mathrm{C}$, this induces the important thermal residual stress field generally found after the curing process of composites. The final mechanical loading step is subsequently applied displacement-controlled. In the nominal case, 1.5% of tensile strain or 7% of compressive strain were applied. When scaling the strengths of the matrix and interface, these values have to be adapted of course. To minimize computation time, all simulations were run together with a *Fortran* subroutine, which cuts the simulation as soon as the load drops by 10% relative to the maximum load obtained.

4.3.4. Final Result

The final result of Step 2 of the simulation workflow is a **.cae* and **.inp* file defining the *Abaqus* simulation to be run, and a **.txt* file defining the periodic boundary conditions to be applied. By the end of Step 2, the user has made additional choices for the following input parameters:

- Geometrical Parameters:
  - Mesh size; nominally **0.5** µm
- Material Parameters:
  - Strength of the matrix and fiber-matrix interfaces; nominally as listed in Tables 3.2 and 3.3
- Simulation Parameters:
  - Displacement of mechanical loading; nominally **1.5%** under tension or **7%** under compression

4.4. Step 3 & 4: Running and Post-Processing of the Simulation

The simulations are run on a desktop computer with 6 physical and 12 virtual CPUs. At every time, two simulations using each 6 virtual CPUs were run in parallel. For this configuration, the computation time for a single SVE with nominal input parameters (particularly SVE and mesh size) falls between a range of 10 to 30 minutes, with simulations of transverse *compressive* loading on the upper end of this scale. An example result, i.e. stress-strain curve, is presented in Figure 3.10. The effect of the *Fortran* subroutine can be readily seen, cutting of the simulation as soon as the load drops by 10% relative to the maximum load obtained.

4.4.1. Stochastic Effects of Fiber Distribution on Damage Modelling

At this point, the stochastic nature of the fiber distribution within a SVE has to be reminded. As touched upon in Section 3.1.2.1, a random distribution of fibers within the SVE is necessary in order to obtain statistically significant results comparable with experiments of UD FRPC plies. Figure 4.2 shows an example of six different *realizations* of a SVE with the same geometrical parameters [1]:

- *Dmin*: 0.2 µm
- *FVolF*: 30%
- *RFF*: 0.5 (thus equal fractions of round and 4-lobed fibers)
- *LobularDiam*: 8 µm (same for circular fibers)
- *Height, Width*: 50 µm x 50 µm

The top left SVE of *sim_1* is the one familiar from Figure 3.9, while the remaining five are different realizations for the same geometrical SVE parameters. Since the fibers are placed randomly by the automatic SVE generation tool *Vip3r* [30], every *realization* of a SVE has a different fiber distribution, particularly noticeable in the case of low fiber volume fraction, as can be readily observed in Figure 4.2.

A general approach when aiming to simulate a *single* Representative Volume Element (RVE) is to choose the number of geometric elements within the RVE sufficiently high, such that two realizations only lead to insignificantly different results. Hereby it is up to the user to define and decide the accuracy which is needed. For a fixed fiber size and fiber volume fraction the only parameter left to tune is the RVE size. When the results of interest are for example the elastic properties of the FRPC ply represented through the RVE, it was found that this approach is indeed feasible. After a sensitivity study, the size of the RVE is fixed and a *single* RVE may from now on be considered sufficient to determine the elastic behavior of the represented FRPC material [6].

In our case however, we model the damage and failure behavior of each constituent of the FRPC material, which has been shown to strongly depend on the fiber distribution, see Section 3.1.2.1. In consequence, a very large RVE would be necessary to reduce the statistical spread of transverse strength results to a minimum. For an example case of circular fibers at $FVolF$ = 50% under transverse tension, this was achieved only for a RVE size greater than 200 µm x 200 µm, at 16 times higher computational cost than for the nominal size later to be chosen as 50 µm x 50 µm. In such a case that the computation of a single RVE is computationally too expensive, an alternative approach may be used. In order to obtain statistically significant results with a "small" volume element, *several* realizations may be generated and simulated. To be precise, the RVEs in this case must be referred to as Statistical Volume Elements. Then, the results from all SVEs are averaged, with the standard deviation of the results indicating the accuracy of the obtained mean value.

To illustrate this further, Figure 4.3 shows the stress-strain curves of the six different SVE realizations of Figure 4.2 subjected to transverse tension. The stress-strain curve of *sim_1* is the one familiar from Figure 3.10. As noted in the legend of the figure, the stress-strain curve of *sim_5* was disregarded since it has a local maximum before the global stress maximum, a rule to be explained later-on in Section 4.4.3. The average maximum stress value can now be calculated from *sim_1..4* & *sim_6* as 57.76 MPa, with a standard deviation of 2.96 MPa, i.e. 5.1%. The average strain at maximum stress is 0.72% with a standard deviation of 0.076, i.e. 10.6%.

4.4.2. SVE Parameter Choices

When using the approach of SVEs, the user has to decide on two parameters: the size of the SVE, and the number of realizations to evaluate. The former defines the range of the statistical spread of results, and the latter the accuracy and confidence up to which the spread can be determined. Based on a performed sensitivity study, presented in Chapter 7, Section 7.1.2, and in accordance with literature [27][30][55], the nominal SVE size for the simulations of this master's thesis was set to 50 µm x 50 µm. For the nominal fiber diameter of 8 µm, this results in 15, 25, and 35 fibers for a FVolFs of 30%, 50% and 70%. Furthermore, 5 *good* SVE realizations were evaluated for each UD FRPC ply in question, based on findings in literature [55][27][57].

The adequacy of these assumptions is defined by the type of result which shall be computed, and the desired accuracy. Generally, it was found that the relative deviation of the strain results is much larger than those of the stress results. Furthermore, non-circular fiber shapes introduce a new geometrical feature into the SVE:

[1] For explanation of the parameters, see Section 4.2.

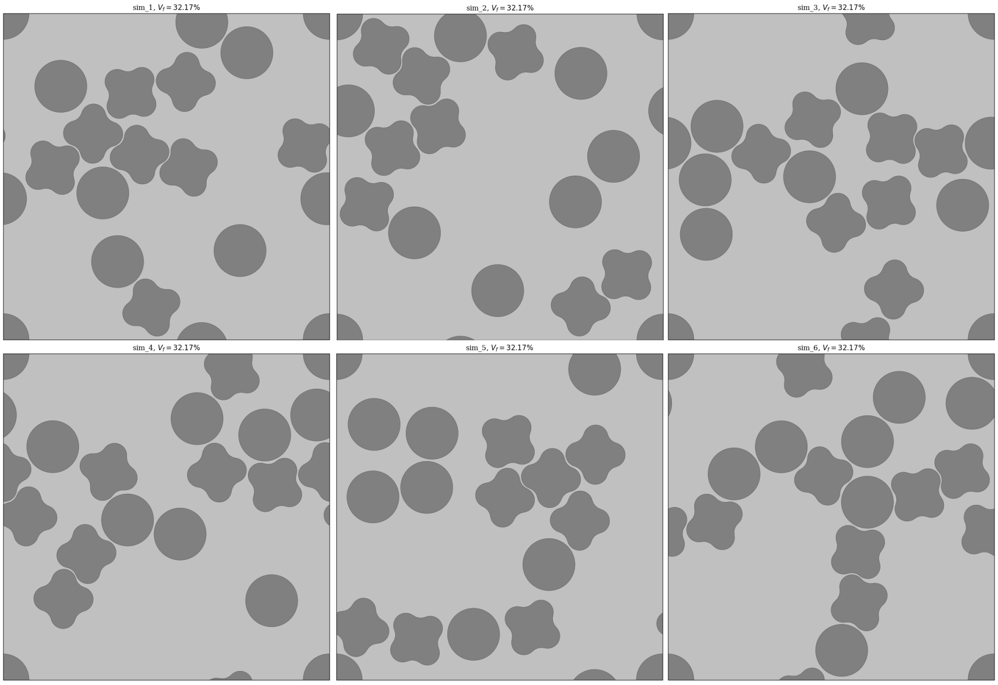


Figure 4.2: Six different SVE realizations for the same geometrical parameters: *Dmin*=0.2 µm; *FVolF*=30%; *RFiberFraction*=0.5; *LobularDiam*=8 µm (same for circular fibers); *Height, Width*=50 µm x 50 µm.

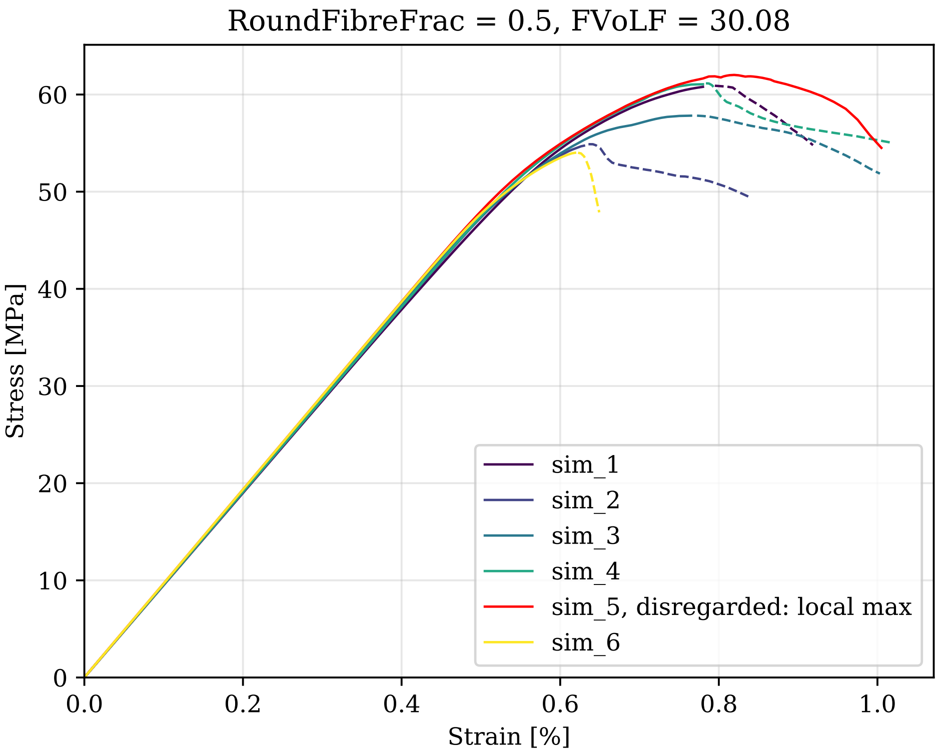


Figure 4.3: Stress-strain curves for the six SVE realizations of Figure 4.2. *Sim_5* has been disregarded due to its local maximum before its global maximum. Of the remaining five simulations, the average maximum stress value is 57.76 MPa, with a standard deviation of 2.96 MPa. The average strain at maximum stress is 0.72% with a standard deviation of 0.076.

the angular orientation of the cross-section. When representing a FRPC material with non-circular fibers, any SVE has to be large enough to capture the statistical variations in *both* fiber position and orientation. Last but not least, for a fixed SVE size, the observed spread of the transverse strength results depends on the Fiber Volume Fraction, i.e. defining the number of fibers in the SVE. The validity of our SVE size of 50 μm x 50 μm for FVolFs as low as 30% is discussed in general terms in Section 7.1.2. Furthermore, for *every* performed parametric study, the standard deviation of the five strength results is recorded, see Listing 4.3, and used to judge the validity of our assumptions after each study. For example:

- The SVEs of Figure 4.2 sustain under transverse tension a maximum macroscopic stress of 57.76 MPa ± 5.12% (standard deviation), see Figure 4.3. As to be shown in Chapter 7, in one study we evaluate what kind of effect the scaling of the IFS has on the maximum stress value. It was found that scaling the IFS can easily double the overall strength of the SVEs. In this case, a standard deviation of 5% for a single datapoint is insignificant.

- In another case, we evaluate what kind of effect the FVolF has on the maximum stress value. In this case, the value of the maximum stress changes to much lower extent compared to scaling of the IFS, in the order of only 15%. Accordingly, a standard deviation of 5% for a single datapoint becomes significant.

At this point, it has to be remarked that the example results shown in Figure 4.3 correspond to the "worst" case, of lowest FVolF and non-circular fiber cross-sections. Based on the findings of the sensitivity study in Section 7.1.2, results for fiber volume fractions lower than 40% were generally disregarded. In most other cases, the obtained standard deviations for transverse strength with the nominal SVE size of 50 μm x 50 μm are well below 2%. Details will be discussed in Chapter 7, specific to each study.

4.4.3. Post-Processing Rules

As mentioned in the previous section, five *good* SVE realizations are required for every combination of input parameters to be simulated. *Good* realizations are considered those which produce results which are in line with the other realizations based on the same parameters. Figure 4.3 shows the stress-strain curves of six different SVE realizations subjected to transverse tension. Every stress-strain curve starts with same slope i.e. Young's Modulus up to the point of failure initiation, which can be at slightly lower or higher strain levels, depending on the stochastic distribution of fibers within the SVE. After the non-linear part of the stress-strain curve, the point of maximum stress is reached, followed by a more or less pronounced load-drop. Especially for the shown case of low fiber volumes and non-circular fibers, the stress-strain curves show a noticeable spread in their non-linear parts up to maximum stress, which can be reduced by increasing the SVE size, albeit at increased computational cost.

Throughout the thousands of simulations of this thesis, it was observed that in some cases, the stress-strain curves show local maxima before the global maximum, which generally leads to a maximum stress and strain higher than for all other SVEs of the same batch. In the example of Figure 4.3, this is the case for *sim_5*, although more pronounced deviations have been observed in other cases. In particular, this was usually only observed for low FVolFs, and could further be attributed to a limitation of our SVE generation tool *Vip3r*.

This tool was developed to create SVEs of FRPC materials with *high* FVolFs, as they are most common in engineering applications. To position the fibers in a random way within the SVE, *Vip3r* starts by putting each one fiber in each corner of the SVE, as can be seen as a commonality of all the SVEs shown in Figure 4.2. For SVEs of high FVolF, this is not noticeable and furthermore does not alter the simulations under transverse loading. For $FVolF < 40\%$ however, it is visually noticeable as a regular feature of the otherwise random fiber distributions. More significantly, in some cases it was observed that transverse failure may concentrates around the cornering fibers, which is not judged appropriate. These simulations usually result in a first stress maxima around the mean value of the other SVE realizations of the same batch, but then reach a second stress maxima at a much higher value.

In order to automatically avoid such alterations of the averaged results, the post-processing scripts were simply adapted such that simulations with local stress maxima before their global stress maximum are automatically discarded. Lastly, in single cases it was observed that failures in the automatic mesh generation may result in stress-strain curves with wrong Young's Modulus and final failure occurring much earlier than for *good* SVE realizations. In order to automatically disregard such *outliers*, the ensemble of final results were cleared from single maximum stress values off by more than 40% from the average.

4.4.4. Final Result

The final result of Step 4 of the simulation workflow is a single line **.txt* file, containing the synthesized results for the simulated *datapoint*. This term is introduced to describe the combination of the input parameters of the modeled UD FRPC ply, plus the corresponding transverse strength results which were obtained. Exactly these readily computed datapoints then later form the *database* to which the NN is fitted, relating the input parameters to the transverse strength with the sought-after surrogate model.

Listing 4.3 shows the *datapoint*, i.e. results file for the SVEs of Figure 4.2, having been subjected to transverse tension. The first three lines describe the relevant geometrical input parameters of the SVEs of this batch: the fraction of round fibers *RFiberFrac* (= 1 - fraction of 4-lobed fibers), the total fiber volume fraction *FVolF*, and the minimum inter-fiber distance *Dmin*. All other parameter were kept at their nominal values, as specified throughout this chapter. Lines 4 to 7 then note the averaged results for maximum stress and strain obtained for this combination of input parameters, as well as the corresponding standard deviations.

Lines 8 to 13 are additional outputs for automatic diagnostics. For the computation of any datapoint, five *good* SVE realizations are required, and a maximum of 10 attempts are allowed as explained before. If after simulating 10 SVEs, no 5 acceptable stress-strain curves are obtained, the computation of this datapoint is aborted and *Flag_abortion* set to 1. The total number of SVE realizations generated for this datapoint is outputted in line 9 as *nSVEs*. Line 10 denotes the number of SVEs whose simulations achieved full strain. This was sometimes observed in the compression case for up-scaled matrix strength, with the load often not dropping significantly after the point of maximum stress. Line 11 denotes how many SVE results were disregarded because of local maxima before the global maximum. Lines 12 and 13 denote the number of SVEs whose results were disregarded for being an outlier, as defined in the previous section.

```
 1  RFiberFrac                 0.4970
 2  FVolF                      30.0821
 3  Dmin                       0.2112
 4  Max_stress_avg             57.7575
 5  Max_stress_std             2.9575
 6  Strain_at_max_stress_avg   0.7219
 7  Strain_at_max_stress_avg   0.0762
 8  Flag_abortion              0.0000
 9  nSVEs                      6.0000
10  nSVEs_full_strain          0.0000
11  nSVEs_local_max            1.0000
12  nSVEs_outlier_stress       0.0000
13  nSVEs_outlier_strain       0.0000
```

Listing 4.3: Final results file of a single datapoint for the SVEs of Figure 4.2 subjected to transverse tension

4.5. Summary

Hereby, we conclude the chapter on the automation and adaptation of the Computational Micromechanics (CM) simulations for the proposed framework. The simulation workflow has been explained step by step, including the used tools, nominal choice of parameters, and the challenges encountered on the way towards full automation. Regarding the latter, the damage modelling part of the used CM model introduces the biggest challenges. Even more so, because the model has to perform correctly over a wide range of parameter values, which are later-on varied for the parametric studies presented in Chapter 7. Necessary post-processing rules have been introduced, which for example automatically disregard stress-strain curves with a local stress maximum before the global stress maximum. Furthermore, in order to obtain statistically significant results, five Statistical Volume Elements are generated and simulated for every set of input parameters. The obtained standard deviation of the transverse stress results can then be used to monitor and judge the accuracy of the CM model, which will be discussed again in Chapter 7 for every study.

After having explained the adaptations of the CM model, the next chapter shall now discuss the Neural Network (NN) implementation for the ’smart’ framework. Again, the main challenge is to fully automate the process, which on the NN side amounts to the choice of the right parameters and training procedure. Eventually, whenever a new datapoint has been simulated, by the simulation workflow explained in this chapter, the NN shall automatically be fitted to the database. A side-result from its training process of *Cross-Validation* can then serve as a measure whether more datapoints have to be computed.

5 Implementation of Neural Networks

This chapter shall detail the implementation and automation of Neural Networks (NNs) for the proposed 'smart' framework. Within the framework, the NN takes the role of a universal approximation function, which must be able to approximate the unknown, non-linear, and high-dimensional transverse strength relationship, based on a very limited amount of datapoints. In order to do so without any user interaction required, the definition of the NN architecture and the training process must be both robust and automatic. A particular challenge in this is to avoid both *underfitting* and *overfitting* of the NN, at the same time.

Building on the theory of NNs introduced in Section 3.2, this chapter explains with very practical examples all choices which have to be made by the user to correctly implement a NN. First of all, our choice for the open-source distribution *scikit-learn* is motivated. Then, all parameters of the used *MLPRegressor* class are explained, together with minimal examples illustrating their influence on the quality of the NN fit. The main challenge consists of defining an automated training process, which can correctly fit the NN to any kind of dataset. To understand the challenges involved, we explain the training process in more detail, define what *underfitting* and *overfitting* means, and why the available database has to be split into so-called *training* and *testing* data. Then, we compare three popular training methods to be able to motivate our final choice for *Cross-Validation* with *repeated-5Fold*. After a detailed example illustrating how said method can train a NN in a fully automated way, we conclude with another important step of our training process: the scaling of the training data.

5.1. Choosing a Neural Network Library and Programming Language

The theory of NNs, as outlined in Section 3.2, has been readily implemented in programming language. Both commercial as well as open-source libraries exist. A popular example of the former is the *Matlab Neural Network Toolbox* [16], commercially available since the 1990s, which provides a conveniently complete set of NN functionalities: pre- and post-processing of data, training of shallow and deep networks, choice of different training algorithms, etc. However, forefront research on NNs today is often being shared through open-source distributions, with *Google*'s *TensorFlow* [2] being a popular variant made available in 2015 and focusing on *deep learning*, i.e. NNs with hidden layers in the numbers of hundreds. *Keras* [14] and *scikit-learn* [54] are two more examples of widespread open-source NN libraries, both of them written in *Python*.

The problem at hand in this master's thesis requires a NN for function approximation, i.e. non-linear regression. As mentioned in Section 3.2.5, not more than one or two hidden layers were necessary to accurately approximate any of the reviewed cases where relationships between material properties were approximated. Further, no advanced NN architectures such as *convolutional* or *recurrent* NNs were necessary to achieve sufficient approximation accuracy. Since the implementation of a shallow, fully-connected, feed-forward NN can be found in several open-source NN libraries, the work of this master's thesis was confined to the use of open-source code. Both libraries *TensorFlow* and *Keras* focus on *deep learning*, therefore *scikit-learn* and accordingly *Python* were chosen as the NN library and programming language for this master's thesis.

The use of an open-source software brings about the attractive advantage of being able to debug down to the bottom-most layer of the source-code, thus the opportunity to fully understand the computations behind (in contrast to commercial codes like *Matlab Neural Network Toolbox*, or even *Abaqus*). On the downside, bugs may arise and there is not more support other than the bug reporting option of *scikit-learn* on the open software repository *github*, and the list of developers and supporters continuously working on fixing them without being paid for it. Limitations of the *scikit-learn* library for the specific use case of this master's thesis were discovered, as well as relevant bugs in the current version of the library. Both however could be circumvented, or did not have such a significant effect that the use of another library would have been necessary. Details are discussed in the following sections.

5.2. Choosing the Right Neural Network Parameters

As mentioned in Section 3.2.3, the Multilayer-Perceptron (MLP) has become a standard NN architecture, both for *Classification* and *Non-linear Regression*. Dealing with the latter problem in the course of this master's thesis, the *scikit-learn* library offers a readily implemented *MLPRegressor* class. In general terms, it allows to define a NN architecture and training process. The latter in mathematical terms amounts to the minimization of the *total error energy E* of the NN for a given set of training data, as touched upon in Section 3.2.6 and explained in more detail in the literature review.

```
class sklearn.neural_network.MLPRegressor(hidden_layer_sizes=(100,), activation=relu, solver=adam, alpha
    =0.0001, batch_size=auto, learning_rate=constant, learning_rate_init=0.001, power_t=0.5, max_iter
    =200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9,
    nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999,
    epsilon=1e-08)
```

Listing 5.1: scikit-learn MLPRegressor class

As shown in Listing 5.1, the *MLPRegressor* class consists of many parameters for which values have to be chosen by the user. Only for the right choice of parameters will the NN fit the training data in the desired way. Figure 5.1 illustrates the fitting of a 1-15-1 MLP NN to the datapoints marked as blue dots. The orange line marks the reference function $f(x) = \cos - \frac{(10 \cdot x)^2}{9}$, which in this case has been perfectly approximated by the NN. A total of 918 iterations, i.e. forward and backward passes of the training data were necessary for convergence towards a minimum of the *total error energy E*. The corresponding parameter choices in this case are

- **hidden_layer_sizes** = 15, i.e. a single hidden layer with 15 neurons
- **activation** = logistic sigmoid function
- **solver** = L-BFGS
- **alpha** = 0.0001
- **tol** = 0.0001

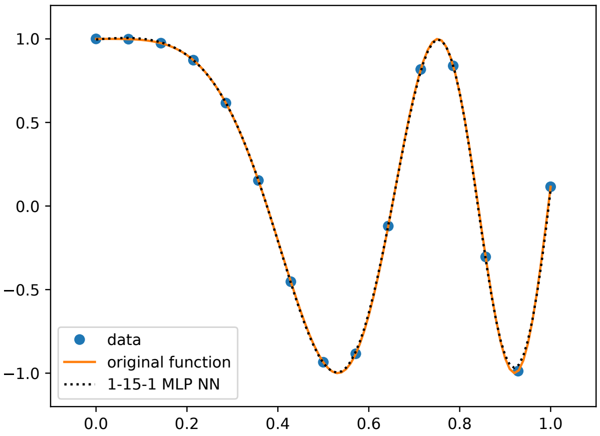


Figure 5.1: Fitting of a 1-15-1 MLP NN to the datapoints marked as blue dots. Reference function $f(x) = \cos - \frac{(10 \cdot x)^2}{9}$ drawn as orange line. Parameters of the *MLPRegressor* are: **hidden_layer_sizes** = 15; **activation** = logistic sigmoid function; **solver** = L-BFGS; **alpha** = 0.0001; **tol** = 0.0001

In the following, the many parameter choices to be made for the *MLPRegressor* class shall be explained, and their severe influence on the quality of the NN approximation be highlighted. To understand where these parameters fit within the concept of the MLP NN, please refer to Section 3.2. For more detailed mathematical derivations, please refer to the literature review.

5.2.1. hidden_layer_sizes

The number of hidden layers, as well as the number of neurons per hidden layer, is denoted by the parameter **hidden_layer_sizes**. Using the annotation introduced in Section 3.2.5, a 3-4-1 network only has a single hidden layer of four neurons, while the number of neurons in the input and output layer is determined by the training data. In this case, three neurons in the input layer for three inputs, and one neuron in the output layer for a single output. The corresponding choice for the **hidden_layer_sizes** parameter would be *(4)*.

As highlighted in Section 3.2.5, the number of hidden layers and hidden neurons is decisive for the ability of the NN to approximate the function underlying the training data. As mentioned in the previous Section 5.1, our case of *Non-linear Regression* does not require a NN architecture with more than a single hidden layer. Conveniently, this leaves as a single free parameter the number of neurons in the hidden layer.

The results of the literature review with regard to this latter choice were experimentally confirmed in this work: Too few neurons may result in a NN which is unable to match a complicated function, independent of all other parameter choices, which is referred to as *underfitting*. On the other extreme, once a NN contains more neurons than at least necessary to fit a certain function, then the risk of *overfitting* prevails. In the context of NNs, overfitting refers to the NN assuming a more complex fit to the training data than necessary. Both underfitting and overfitting are illustrated in Figure 5.2, on the left and right respectively, through a 1-1-1 and 1-200-1 NN trained on the same dataset as shown in Figure 5.1.

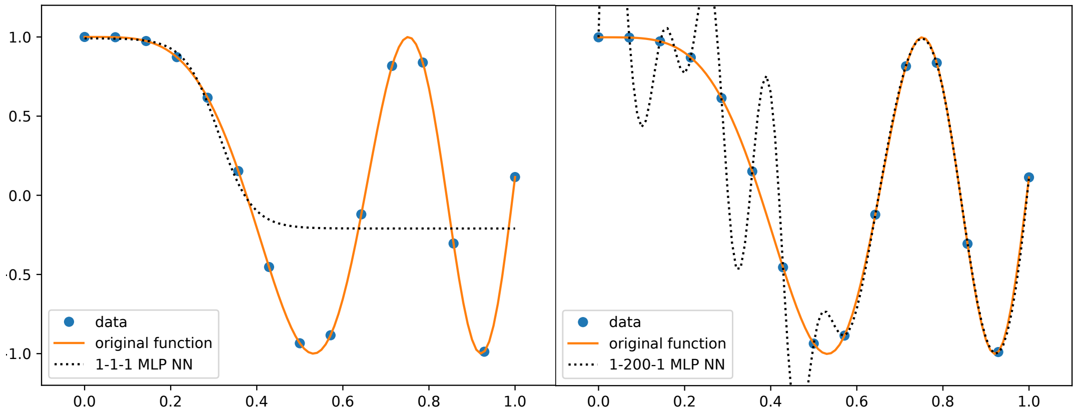


Figure 5.2: Underfitting of a 1-1-1 MLP NN (left), and overfitting of a 1-200-1 MLP NN (right) to the datapoints marked as blue dots. Reference function $f(x) = \cos - \frac{(10 \cdot x)^2}{9}$ drawn as orange line. Other parameters of the *MLPRegressor* are kept constant: **activation** = logistic sigmoid function; **solver** = L-BFGS; **alpha** = 0.0001; **tol** = 0.0001

Two remarks are due at this point:

- Prior to the training of the NN, it is not possible to deduce the number of neurons necessary to adequately fit the function underlying the training data. Instead, an experimental approach is always required, which underlines the need for an automated process which is able to judge the quality of the NN approximation. This quality has to be measured twofold, how well the NN matches the training data in the points of the training dataset, i.e. avoiding *underfitting*, and whether *overfitting* occurs *in between* the points of the training data.
- The results shown in Figure 5.2 are strongly depending on the value chosen for **alpha**, the *L2 regularization* term to be discussed later-on.

5.2.2. activation

The activation function, see Section 3.2.4. The *MLPRegressor* class offers the following function choices:

- identity function

- logistic sigmoid function
- hyperbolic tan function
- rectified linear unit function

The use of the identity function and rectified linear unit function is limited to the input and output layers, or when deviating from the classical feed-forward NN architecture. Therefore, in our use case, only the logistic sigmoid function and hyberbolic tan function are relevant. The same activation function is chosen for all hidden neurons. For the simple example shown in Figure 5.1, the choice of activation function for the hidden layer does not produce a visible change. For larger datasets and fitting in more than one variable, e.g. (x,y,z) data, slight differences can be observed between NNs using tanh or logistic activation functions in their hidden layer. Throughout this thesis, it was generally observed that using the logistic function results in a slightly better approximation.

In general, the choice of tanh or sigmoid activation function has a much less significant effect on the quality of the NN fit than the number of hidden neurons, or the choice of the *L2 regularization* term. In the experience of this work, tanh vs. sigmoid may slightly alter the shape or smoothness of the NN fit, but does not lead to severe over- or underfitting rendering the results useless. Attention has to be paid since the input range of the activation function determines the later-on scaling of the inputs of the training data. For both tanh and sigmoid function, the input range with high gradients lies centered around $x = 0$. Details of the scaling of the input data, and why it is necessary are explained later-on in Section 5.3.4.

5.2.3. solver

The *learning rule*, i.e. gradient-based optimization algorithm, as described in Section 3.2.6, is denoted by the parameter **solver**. The *MLPRegressor* class offers implementations of the following algorithms:

- Limited Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [22][41][85]
- Stochastic Gradient Descent (SGD) [29][56]
- ADAM [36], a popular variant of SGD

Aside from the NN architecture, the choice of the learning rule i.e. optimization algorithm is central to the design of a NN and exerts a major impact on the results. The *MLPRegressor* class of *scikit-learn* offers most refined capabilities, such as *early stopping* and *batch-learning* [1], only for the solvers SGD and ADAM. However, as the first attempts of data fitting with the *MLPRegressor* and SGD/ADAM solver revealed in this work, both solvers fail to converge for small datasets. Figure 5.3 illustrates the fit of the same 1-15-1 MLP NN as described in Figure 5.1, only changing the solver to SGD (left) and ADAM (right). In both cases, the NN is not able to fit the training data, and the training process aborts after 3/412 iterations for SGD/ADAM.

[1] See literature review Section 2.4 for more details.

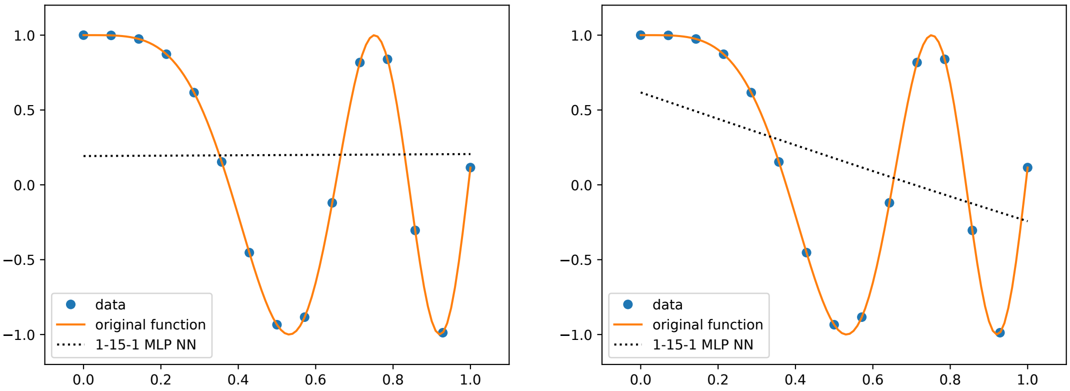


Figure 5.3: Misfitting of a 1-15-1 MLP NN to the datapoints marked in blue through the solver SGD (left), and ADAM (right). Reference function $f(x) = \cos - \frac{(10\cdot x)^2}{9}$ drawn as orange line. Other parameters of the *MLPRegressor* are kept constant: **activation** = logistic sigmoid function; **alpha** = 0.0001; **tol** = 0.0001

Since the size of the training set remains below 150 in the course of the work of this master's thesis, the use of SGD/ADAM solver was thus barred. Being bound to the use of the L-BFGS solver restricts the number of parameters of the *MLPRegressor* class for which values have to be chosen by the user. On a positive side, this simplifies i.e. accelerates the search for an optimal NN architecture, which in practice requires "an exhaustive experimental study" [29], as to be seen later-on. On a negative note, the pre-implemented *early stopping* method for SGC/ADAM, a common technique to avoid *overfitting*, is not available when using L-BFGS.

It was attempted to implement an *early stopping* rule for the L-BFGS solver by defining a custom stopping criterion. This requires incremental training of the NN, verifying the stopping criterion at every step, and then either resuming or stopping the training process. Unfortunately, a bug prevailed in the *MLPRegressor* class when using the necessary functionality *warmstart* in combination with the L-BFGS solver [2], thus rendering it impossible to implement and use the method of *early stopping*. Instead, an alternative method to prevent overfitting was implemented, called *Cross-Validation*, described later-on in Section 5.3.2.2.

5.2.4. alpha

The scaling factor of the *L2 regularization* term, which is added to the total error energy E of the NN. At this point it shall be recalled that the *learning* of the NN boils down to the optimization algorithm trying to minimize the total error energy E of the NN, for a given set of training data. An error energy term of zero means that the NN matches exactly the training data. However, this does not say anything about the quality of the approximation *in between* points of the training data, i.e. *overfitting* may well be prevalent. Now, the *L2 regularization* term $L2$ of a vector of synaptic weights $\boldsymbol{w} = [b, w_1, w_2, ..., w_m]^T$ of a NN is defined as

$$L2(\boldsymbol{w}) = w_1^2 + w_2^2 + ... + w_m^2 \tag{5.1}$$

By adding the $L2$ term to the total error energy E, the solver will accordingly try to minimize both. Large values for the synaptic weights will thus be penalized, which is equal to penalizing the NN when taking the shape of a more complex function. The **alpha** term of the *MLPRegressor* class is simply the scaling factor of the $L2$ term. The choice of **alpha** further has a major influence on overfitting, being illustrated in Figure 5.4. The fitting of three different MLP NNs to the datapoints marked in blue is shown. The NNs only differ among each other in the number of neurons in their hidden layer. Compared to the NN shown in Figure 5.1 however, the *L2 regularization* factor **alpha** was set to zero, thus not penalizing large weights. When comparing Figure 5.1 with Figure 5.4, it can be noticed that the latter shows only slight indices of overfitting which are shown more pronounced for the 1-47-1 NN. By further adding just one more hidden neuron, the overfitting becomes predominant and renders the NN approximation useless. Two concluding remarks can be made:

- The extent of overfitting of an unregularized MLP NN does not scale linearly with the number of hidden neurons. Increasing the latter from 47 to 48 in the example of Figure 5.4 suddenly increases overfitting.
- With all other parameters chosen reasonably, the factor of the regularization term **alpha** and the number of hidden neurons are the two main parameters influencing the quality of the NN fit to a dataset.

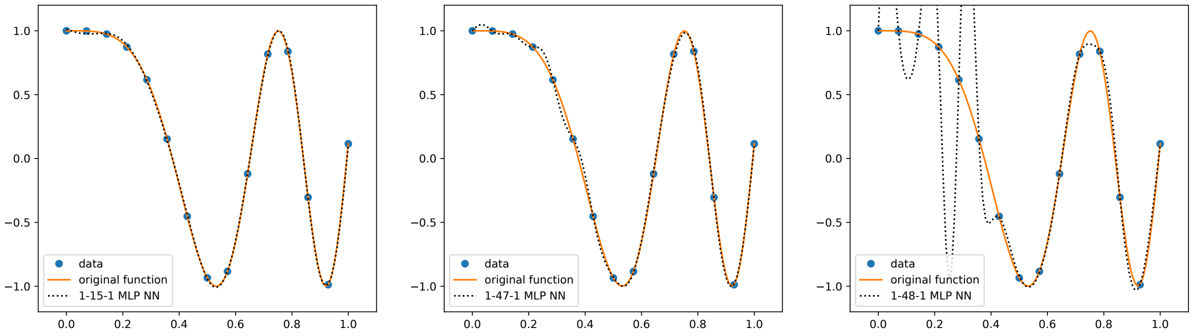


Figure 5.4: Overfitting of a 1-15-1, 1-47-1, and 1-48-1 MLP NN, from left to right, and from slight to moderate to severe. Reference function $f(x) = \cos - \frac{(10\cdot x)^2}{9}$ drawn as orange line. No *L2 regularization* was used, i.e. **alpha = 0**. The other parameters of the *MLPRegressor* are kept constant: **activation** = logistic sigmoid function; **solver** = L-BFGS; **tol** = 0.0001

[2] scikit-learn v0.19, April 2018

5.2.5. tol

The stopping i.e. convergence criterion for the solver. As soon as the total error energy E does not reduce by the value of **tol** from one interation to the next one, the solver stops and assumes convergence. In the course of the work of this thesis, this parameter was kept at its default value of 0.0001.

5.2.6. random_state

Generally, the initial values of the synaptic weights and biases of a NN are assigned at random. This means that two different initializations of the same NN will lead to different initial values of the synaptic weights and biases, and thus to a different progression of the learning process, i.e. the adaptation of the synaptic weights and biases through the optimization algorithm. In order to be able to reproduce and compare results, the **random_state** parameter allows to initialize the weights and biases with *the same* random draw of numbers.

5.2.7. Summary

The above discussed all necessary parameters which have to be set by the user in order to define a NN through the *MLPRegressor* class. All other parameters shown in Listing 5.1 are either not applicable to / not available for the L-BFGS solver, or insignificant for this discussion. In summary, for our case of training a MLP NN for a *Non-linear Regression*, it suffices to determine the values of only two NN parameters:

- **hidden_layer_sizes**
- **alpha**

5.3. Defining and Automating the Training Process of a Neural Network

The previous section explained all basic parameters which have to be set for the *MLPRegressor* class, in order to define the architecture of the MLP NN. Aside from that, the training process of the NN has to be defined. The detailed mathematical derivations of the training process can be found in the literature review Section 2.4. In short, the training process of a MLP NN follows four steps:

1. Choose the basic parameters for the *MLPRegressor* class: All parameters described in the previous section have be chosen by the user, a-priori to the training.

2. Present the training data to the NN: The aforedefined NN is presented with the training data. Either the NN is fitted to *all* available datapoints, or the available database is split into *training data* and *testing data*. In all preceding examples, e.g. Figure 5.1, the NN was trained with *all* available datapoints. However, in most practical NN applications with larger datasets and more than a single variable, the available database is split such that *overfitting* can be evaluated automatically, see next step.

3. Verify the quality of the NN fit to the available data: After the NN has been trained, i.e. fitted to the training data, the quality of the fit has to be evaluated. The quality of a NN approximation to a dataset has to be judged by answering two questions:

   - How well does the NN match the datapoints that it has been trained with, at the *exact* points of the training data? (underfitting)
   - How well does the NN approximate *in between* the datapoints that it has been trained with? (overfitting)

   For the previously shown simple example cases, e.g. Figure 5.1, the quality of the NN approximation could be verified visually through plotting. However, for problems with larger datasets and more than a single variable, this kind of manual verification is not feasible anymore. Instead, automated methods have to be used, which requires to hold-out part of the available dataset from training/fitting, as mentioned in the previous step.

4. Restart the NN training process if the quality of the NN approximation is inadequate: Judging by quality measures introduced to answer the two questions of the previous step, the training process may have to be repeated with a different choice for the basic parameters for the *MLPRegressor* class. In this case, back to the start of the training process.

As can be seen, the training process revolves around the challenge to find the ideal combination of *MLPRegressor* parameters (step 1) and training method (step 2/3), in order to avoid both over- and underfitting. Since there are no fixed rules for either one, the training process in practice is an iterative process, during which a range of parameter combinations are tried before settling with the best one. In order to explain the final training process that has been implemented for the work of this master's thesis, the most common strategies for avoiding underfitting and overfitting must be briefly explained.

5.3.1. Avoiding Underfitting

Underfitting of a NN trained with a certain dataset can be easily measured and thus prevented. The difference between the points of the *training* data and the approximation of the NN at the same points can be calculated and compared for different NNs. Measures of the goodness of the fit are manifold, for example the *coefficient of determination B* [84], which briefly surfaced in Section 3.2.7. By comparing different NNs based on their goodness of fit, a NN can be designed with *MLPRegressor* parameters such that underfitting is prevented.

5.3.2. Avoiding Overfitting

The same aforedescribed method to prevent underfitting does not work for the opposite case. As illustrated in the example of Figure 5.4, when overfitting occurs all datapoints of the training dataset are perfectly matched ($B_{training} \approx 1$), while at the same time the approximation of the NN *in between* the datapoints starts to diverge. Several techniques have been developed to detect or avoid this, three popular variants being:

- Early Stopping
- Cross-Validation
- Bayesian Regularization

The first two methods, *Early Stopping* and *Cross-Validation*, both require to hold-out part of the available dataset for the purpose of verifying that no overfitting is occurring. *Bayesian Regularization* on the other hand is able to circumvent overfitting without such a division. Therefore, it is most valuable for problems with small datasets, and where new datapoints are unavailable or expensive to obtain. The kind of parametric studies performed in the course of this master's thesis stay below 150 datapoints, so certainly a small dataset compared with classical ML problems (which are often the *Classification* type, not *Non-Linear Regression*). Therefore, *Bayesian Regularization* was the first-choice candidate to be implemented in the framework of this thesis to avoid overfitting.

> The concept of Bayesian Regularization avoids the necessity of computationally expensive *Cross-Validation*, by converting the *Non-linear Regression* into a "well-posed statistical problem in the manner of a ridge regression" [10]. MLP NNs designed and trained by Bayesian Regularization are claimed to be more robust in terms of overtraining or overfitting [10], and example cases reviewed in the course of the literature review confirmed that they achieved among the best results [84].
>
> However, while this method is readily implemented in commercial packages such as the *Matlab Neural Network Toolbox* [16], it is missing for *scikit-learn* [54] and all other reviewed open-source library except *auto-sklearn* [21], which is in development status and only available for *Linux*. Aside from that, open-source libraries of *Bayesian Optimization* algorithms are available, but would have to be tied into the framework of *scikit-learn*.
>
> Since the next best method *Cross-Validation* yielded good results throughout the whole span of work of this master's thesis, the effort of switching to *Linux*, or manually implementing a Bayesian Optimization algorithm into *scikit-learn* was deemed disproportional to the certain gain in computation speed and the uncertain gain in quality of the NN approximations. Therefore, the method of *Cross-Validation* was eventually implemented for the framework developed in this master's thesis. In order to motivate this choice over *Early-Stopping*, we want to briefly outline and discuss the commonalities and differences of their working.

The procedure of holding out part of the available data is fundamental to the learning process of NNs when using *Early Stopping* or *Cross-Validation*: The available dataset is first randomly split into *training data* and *testing data*, commonly by a ratio of 70/30% or 80/20% [84]. Then, the NN is fitted to the *training data*, while the quality of the fit is measured afterwards by comparing the withheld *testing data* with the trained NN approximation at the same points.

The rationale for detecting overfitting is that an overfitted NN will predict very poorly on unseen testing data, which lies *in between* the training data. For the example of Figure 5.4, the NN was fitted to all datapoints. Imagine another point of the reference function, in between the blue dots of training data, and that now the quality of the approximation of the trained NN would be tested at this point. As can be seen, the NN function would be far off the reference function at this intermediate point, thus indicating overfitting. The training procedure of splitting up the available dataset into training and testing data is the same for both *Early Stopping* and *Cross-Validation*. The differences come in the following steps:

- *Early Stopping* tries to *avoid* overfitting by stopping the iterative fitting of the NN to the *training* data, as soon as it detects the beginning of overfitting on the *testing* data.
- On the other hand, *Cross-Validation* does not stop the fitting process, it only is a tool to *detect* overfitting once the fit to the training data is completed. When trying different combinations of parameters for the *MLPRegressor* class, *Cross-Validation* can be used to measure and compare the *extent* of overfitting for each case. This eventually allows to choose the *MLPRegressor* which results in the least overfitting.

So far, both methods seem equally adequate to circumvent overfitting. The following two sections now shall explain the two methods in more detail, such that our preference for *Cross-Validation* can be motivated.

5.3.2.1. Early Stopping

For *early stopping*, the iterative training of the NN is stopped as soon as the accuracy of the NN predictions with respect to the *testing* data *decreases*, for example measured by a decrease of the value of the *coefficient of determination B*. To remind, a perfect fit is equal to a B value of 1, lower values (negative possible) indicate worse fit. Generally, any $B < 0.9$ is a very poor fit.

To illustrate the practical complications of the *Early-Stopping* method, the following example is made: Figure 5.5 shows the familiar case of Figure 5.1 repeated, now dividing the available dataset into training and testing data, and stopping at different increments of the iterative fit of the NN to the training data[3]. The points of testing data, shown as 'X', are omitted from the fitting of the NN. At any given increment of the NN fit to the training data, the current quality of approximation of the NN can be measured, both with respect to the training data (B_{train}), and to the unseen testing data (B_{test}). Final convergence of the fit to the training data is achieved when the total error energy E of the NN does not change by more than the value of **tol** from one iteration to the next one, in this case after 975 iterations.

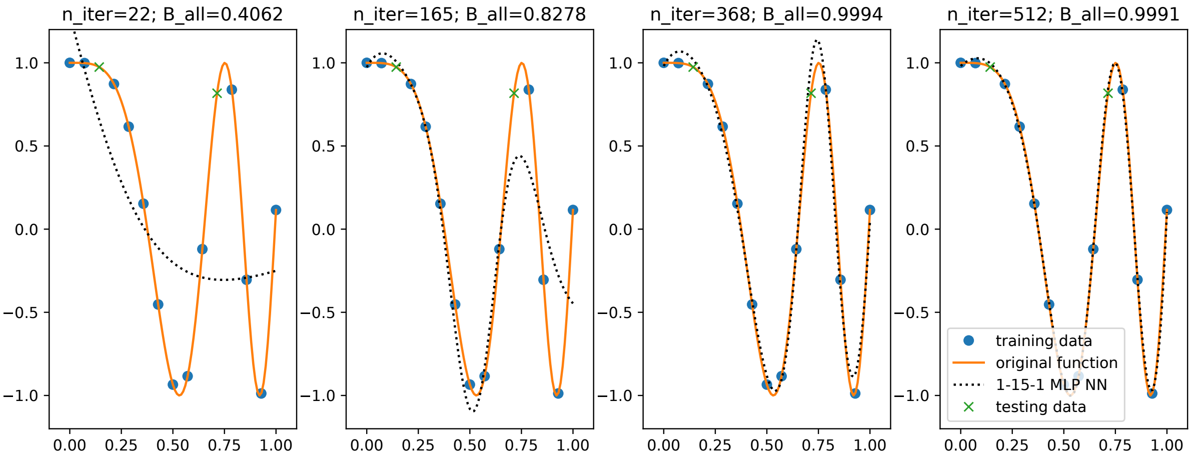


Figure 5.5: Iterative training of the 1-15-1 MLP NN of Figure 5.1, now with two datapoints held out for testing. The final NN approximation is plotted at four different increments of the iterative training process. The annotated B_{all} values refer to the case if all datapoints would have been used for training.

Now the method of *Early Stopping* assumes that B_{train} steadily converges towards a value of 1, and that B_{test} initially increases steadily, and then at some point starts to decrease again. This first divergence is thought to occur when the NN starts to overfit, and therefore the training process should be stopped at this point. That in practice this point is not easy to make out is illustrated by Figure 5.6. It shows the convergence behavior of the

[3] For details on the iterative training process, please refer to Section 2.4 of the literature review.

$B_{training}$ and $B_{testing}$ values, recorded throughout the iterative training process. Both eventually converge towards a value of $B = 1$, which indicates that no underfitting, and no significant overfitting is present, as can also be deduced visually from Figure 5.1.

Figure 5.6 further shows that the gradient of $B_{testing}$ changes several times, which makes it difficult to define a simple stopping criterion. Stopping the training process at any of the first four local maxima of $B_{testing}$ at $n_{iter} = 22, 165, 368, 512$, is visualized in Figure 5.5. The final NN approximation one would end up with ranges from very poor to acceptable. In practice, in order to avoid stopping the iterative fit to the training data too early, a third dataset is held-out from the training process, called the *validation data*. In summary, the available dataset is split into training, testing, and validation data, commonly by 60/20/20% [23]. Fitting of the NN to the training data is stopped as soon as $B_{testing}$ decreases. Then, to verify that the fitting was not stopped too early, $B_{validation}$ is computed.

Overall, the *Early Stopping* method requires to hold out a large share, e.g. 40% of the available data from the training of the NN. Therefore, it is more advantageous to use this method when a large database is available, or when the generation of new datapoints is cheap. Alternatives which require to hold-out a smaller share of the available data to circumvent overfitting are *Cross-Validation* and *Bayesian Regularization*, described hereafter.

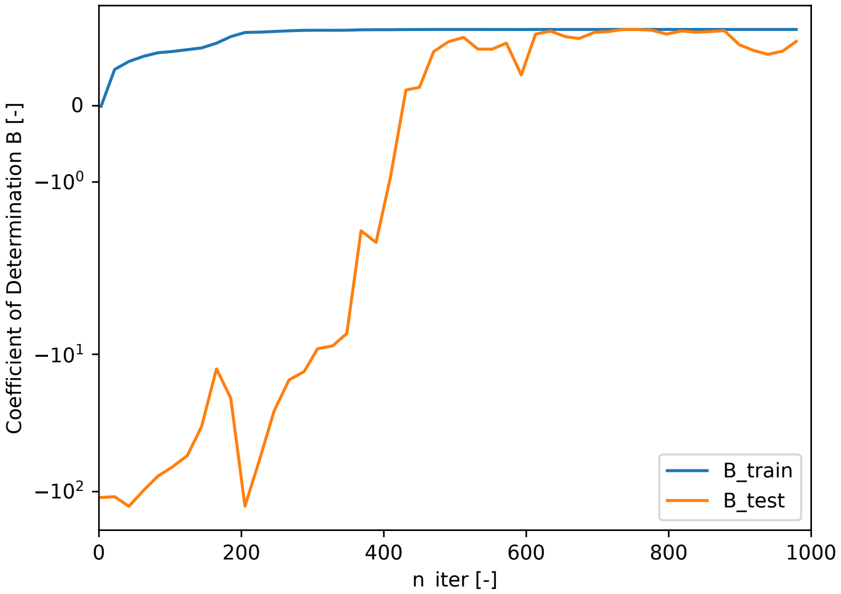


Figure 5.6: Values of the coefficient of determination B plotted over the number of iterations in the training process of the 1-15-1 MLP NN of Figure 5.1. Two datapoints were held out for testing data see Figure 5.5, i.e. calculation of B_{test}, the remaining were used for training of the NN and calculation of B_{train}. The training converged after 975 iterations according to the convergence criterion defined by the *MLPRegressor* parameter **tol**.

5.3.2.2. Cross-Validation

For *cross-validation*, the fitting of the NN to the training data is completed as usual until convergence set out by the parameter **tol**. The held-out *testing* data is then used to calculate $B_{testing}$, or an equivalent quality measure. Its value indicates overfitting of the trained NN approximation, for example $B < 0.9$.

At this point, it has to be reminded that the splitting up of the available data into training and testing data is generally done at random. Therefore, the values of $B_{testing}$ are of stochastic nature as well. To obtain meaningful results, the aforedescribed procedure is repeated several times for different random splits of the available database. Then, the mean of the quality measure is taken as the final result. A minimal example shall now be presented to illustrate the methodology of *Cross-Validation*.

Different than the example of Figure 5.1, we will now use five datapoints obtained with our baseline Computational Micromechanics model, with two instead of one input parameter. In the field of ML, the input parameters are called *features* and the output values *targets*. Each one of the five datapoints thus consists of two *features* and one *target*. The first feature is the Fiber Volume Fraction of the modeled FRPC material and is varied from 30% to 70%. The second feature is the Interface Strength, i.e. the factor by which the strength of the fiber-matrix interfaces is scaled linearly, varying from 1 to 10. The target is the transverse tensile strength, as computed with our CM model described in Section 3.1.2. Details of the computation are not relevant for this discussion, only that overall, every datapoint eventually has the format $\sigma_{transverse,tensile} = f(FVolF, IFS)$.

A MLP NN shall now be fit to this database of five points, in order to produce a continuous function as an approximation of the simulated material behavior. To evaluate overfitting through *Cross-Validation*, the available dataset of five points is randomly split by a ratio of 80/20% into training data (four datapoints) and testing data (one datapoint). After fitting the NN to the four datapoints of the training data, the quality of the fit *in between* the training data is evaluated with the testing datapoint: The difference between the reference value of $\sigma_{transverse,tensile}$ (target of testing datapoint) and the prediction of the trained NN at the same values of the $FVolF, IFS$ features is calculated.

Then, this training process is repeated for a different random split of the available database. Borrowed from statistics, this is called *repeated-kFold*, as the database is split into *k folds*, in our case 5 folds, and this process is repeated 5 times. By this, it is ensured that every *fold*, i.e. group of datapoints of the database, has once served as testing data. This repetition is also the reason for the name of the method, *Cross-Validation*.

Figure 5.8 finally shows the said method applied to the example set out above. The five different *folds* of the available database result in five different trained NNs. For each case, Figure 5.8 indicates the four datapoints of the training data as blue dots, and the testing datapoint as a red *X*. The quality of the trained NN is evaluated as aforedescribed. Since the *coefficient of determination B* is only defined for datasets of two or more datapoints, a different quality measure was used. The *relative error* E_{rel} is defined as

$$E_{rel} = \frac{|\sigma_{prediction} - \sigma_{reference}|}{\sigma_{reference}} \tag{5.2}$$

and simply puts in relation the difference between predicted and actual stress value (or any other entity), to the actual value. Every subplot of Figure 5.8 shows the relative error E_{rel} in its title. As can be seen, their values vary significantly, from 0.09 to 1.7, depending on which datapoint was used for testing. Their mean value, i.e. the *mean relative error* in this case is 0.53, so on average the NN prediction at the testing datapoint is 53% off its reference value. So far, so good, but the question remains:

How does this now help with the issue of choosing the right MLPRegressor parameters?
First, it has to be reminded that the NN approximations shown in Figure 5.8 are based on an a-priori fixed set of parameters for the *MLPRegressor*. In this case: **hidden_layer_sizes**=100, **alpha**=0.0001, **activation**=tanh, **solver**=L-BFGS, **tol**=0.00001. Any change in the choice of parameters may result in a significantly worse or better NN approximation to the same five datapoints. What is achieved by the presented *Cross-Validation* method is that it gives a quality measure at hand to quantify such a change of the quality of the approximation: A higher *mean relative error* indicates a worse fit, i.e. overfitting.

For example, Figure 5.9 shows the same procedure of Figure 5.8 repeated for different, more overfitting prone parameters for the *MLPRegressor*: **hidden_layer_sizes**=(100,10), **alpha**=0. Using the *Cross-Validation* method with *repeated-5fold* as before, the NNs with two hidden layers display a mean relative error of 0.57, compared to the 0.53 of the previous case. According to the rationale introduced before, the parameters of the second NNs achieve overall a worse fit to the available database than the first NNs with 100 hidden neurons in a single layer. Comparing the subplots of both examples visually, it can be noticed that the NNs with two hidden layers show pronounced overfitting: they match the training datapoints exactly, but fail to interpolate in between these datapoints with a simple surface.

With the shown method of *Cross-Validation*, it is thus possible to evaluate the quality of the MLP NN approximations resulting from different combinations of *MLPregressor* parameters. It remains up to the user to predefine the grid of parameter combinations to be evaluated through the *Cross-Validation* method. As has been concluded in Section 5.2.7, for our case of *Non-linear Regression*, the array of parameters for the *MLPRegressor* shown in 5.1 can be broken down to just two parameters whose values have to be chosen:

- **hidden_layer_sizes**
- **alpha**

Choosing for example five values for each parameter results in 25 different NN configurations to be evaluated through *Cross-Validation*. For a *repeated-5fold* procedure, this further amounts to $25 \cdot 5 = 125$ runs of training a NN with the 4 datapoints of the training data. While for such few training points, this does still not amount to much computation time, it was found that for later cases with more than 50 datapoints, the computation time increased significantly. Using parallel computation on 6 CPU's, it may take up to half an hour to evaluate all 25 possible NN parameter combinations.

To showcase the utility of this method with the aforedescribed example, the following parameter grid was evaluated:

- **hidden_layer_sizes**: (10),(30),(40),(60),(80),(100)
- **alpha**: 0.00001, 0.0001, 0.001, 0.01, 0.1, 0.3, 0.5, 0.7, 0.8

Within a computation time of 126 seconds, the *Cross-Validation* method with *repeated-5fold* procedure determined the best parameter combination to be **alpha**=0.00001, **hidden_layer_sizes**=30. This achieved a mean relative error of **0.48**, yet lower than the 0.53 achieved through manual optimization based on visual inspection of the NN fitting.

5.3.3. Summary

This lengthy discussion of methods for MLP NN training and optimization was necessary to coherently motivate why *Cross-Validation* eventually is the method of choice for the work of this master's thesis. As outlined above, it allows to automatically optimize the NN approximation to a given dataset, by determining the necessary *MLPRegressor* parameters for that. As will be discussed in Chapter 6, Section 6.5, the results of *Cross-Validation* further can be used to determine how many datapoints are necessary to approximate an unknown function up to a predefined accuracy.

To conclude this section on the training and optimization process of a NN, a last important issue shall be highlighted: scaling of training data. As has been touched upon before, when explaining the effect of the activation function on the NN approximation quality, *scaling* of the training data is an important pre-processing step in the learning process of a MLP NN.

5.3.4. Scaling of Training Data

Figure 5.7 shows two NNs fitting the same five datapoints familiar from Figure 5.8. Both NNs share the same parameters for the *MLPRegressor* class: **hidden_layer_sizes** = 10; **alpha** = 0.1; **activation** = logistic sigmoid function; **solver** = L-BFGS; **tol** = 0.00001. The only difference lies in the pre-processing of the training data: Both NNs are fitted to all five available datapoints, only for the NN shown on the left of Figure 5.7, the features of the training data were linearly scaled prior to the training process, to fall between a range of -1 and 1.

The differences in the quality of the NN approximations for scaled and unscaled training data can be readily observed in Figure 5.7. Both achieved a very close fit to the training data, with $B_{training,scaled} = 0.9990$ and $B_{training,unscaled} = 0.9987$, although the former required 90 iterations, and the latter 2777 iterations until convergence. This difference indicates the underlying issue of unscaled training data.

The basic mathematical derivations of the fitting of a MLP NN to a set of datapoints have to be recalled at this point, see Section 3.2.4. The features of the training data, in this case $FVolF = 30...70$ and $IFS = 1...10$ are passed through the NN feed-forward. After summing all weighted inputs of a neuron, they are passed through the *activation function*, see Figure 3.14. In this case this is the logistic sigmoid function, whose sensitive region lies between $-4 \leq x \leq 4$. Accordingly, any input higher or lower than this will result in a saturated response of the activation function. In order for the NN to *learn* to fit the training data, it has to adapt the synaptic weights such that each neuron gives the right output. In the case of unscaled features of the training data, whose values lie outside of the range of $-4 \leq x \leq 4$ as in the example of Figure 5.7, many iterations and extremely large values for the weights are necessary to bring the input of the neurons into the sensitive area of the activation function.

In consequence, the iterative training process may take much longer than necessary, and the final NN approximation may be of very poor quality as is the case in the example of Figure 5.7. For the ensuing work of this master's thesis, the scaling of the features of the training data was therefore implemented as a mandatory pre-processing step.

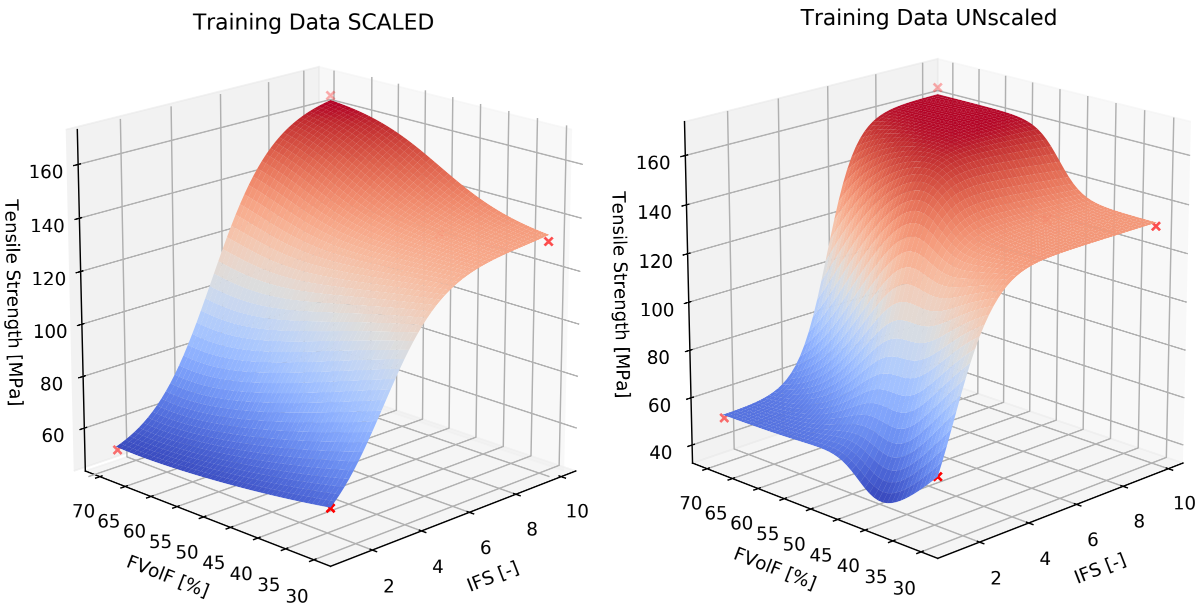


Figure 5.7: Illustration of the effect of scaling of the training data. Both subplots show the same 2-10-1 MLP NN, fitted to the five datapoints of the preceding example in Figure 5.8. The common parameters for the *MLPRegressor* class are: **hidden_layer_sizes** = 10; **alpha** = 0.1; **activation** = logistic sigmoid function; **solver** = L-BFGS; **tol** = 0.00001. The NN on the left was fitted to training data whose features $FVolF$ and IFS were linearly scaled to fall in between -1 and 1. In consequence, the left NN only needs 90 iterations for a good fit, compared to the 2777 iterations of the NN on the right for a much worse fit.

5.4. Summary

Hereby, we conclude the chapter on the implementation and automation of Multilayer-Perceptron Neural Networks (MLP NNs) for the proposed 'smart' framework. We have discussed in its necessary detail all choices which had to be made for setting up the required robust and automated NN architecture and training process. First off, we motivated our choice for the open-source NN library *scikit-learn* and explained, how with some simple rules the large number of parameters of the *MLPRegressor* class can be boiled down to just two: **alpha** and **hidden_layer_sizes**. With some basic assumptions about the possible ranges of these two parameters, we can design a fully automated training process, based on the method of *Cross-Validation* and the procedure of *repeated-5fold*. Eventually, this allows us to implement our MLP NN into the proposed 'smart' framework, where it has to approximate the unknown, non-linear, and high-dimensional relationship between transverse strength and its dependant input parameters. Our NN implementation is equipped to do so in an automated way, and based on a very limited amount of datapoints, circumnavigating both *underfitting* and *overfitting*.

After having explained both the adaptations of the Computational Micromechanics model in Chapter 4 and of the Neural Networks part in this chapter, we can now finally proceed with the most exciting part of this work. In the following chapter, we show how these two methodologies are combined and joined with the Design of Experiments approach, to form the proposed framework for 'smart' Computational Micromechanics. Eventually, in Chapter 7 then we can show how the framework was put to use: to perform automated parametric studies of the transverse strength of a UD FRPC ply, and to determine simple but comprehensive analytical surrogate models of the underlying relationships.

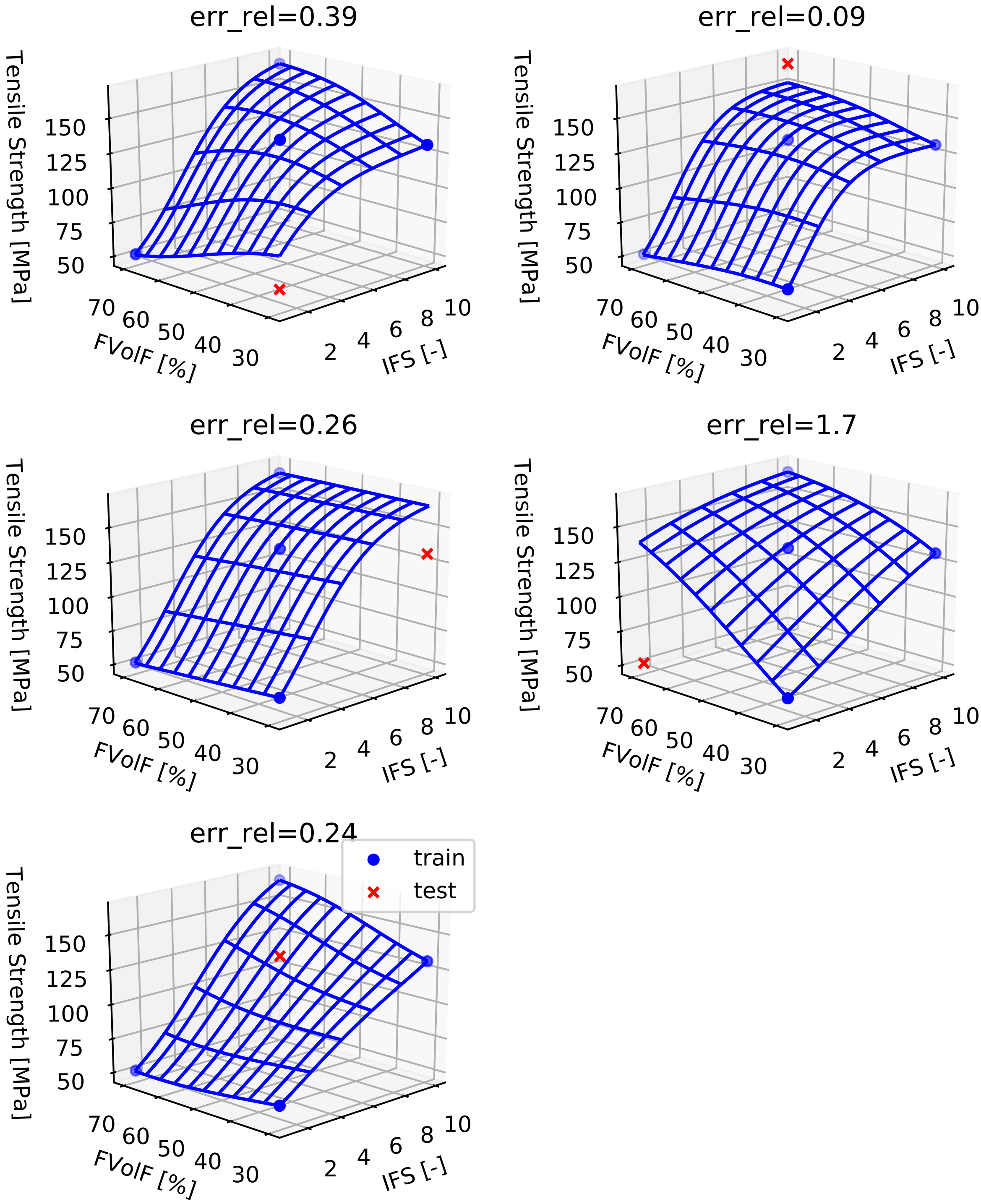


Figure 5.8: Illustration of the *Cross-Validation* method with *5fold* procedure. The available database consists of five points: two features FVolF & IFS, and one target *Transverse Tensile Strength*. Training data is shown as blue dots, testing data as red *X*. The parameters of the *MLPRegressor* are: **hidden_layer_sizes**=100, **alpha**=0.0001, **activation**=tanh, **solver**=L-BFGS, **tol**=0.00001. The relative error of the trained NN at the position of the testing datapoint is annotated above each plot. The mean relative error is 0.53.

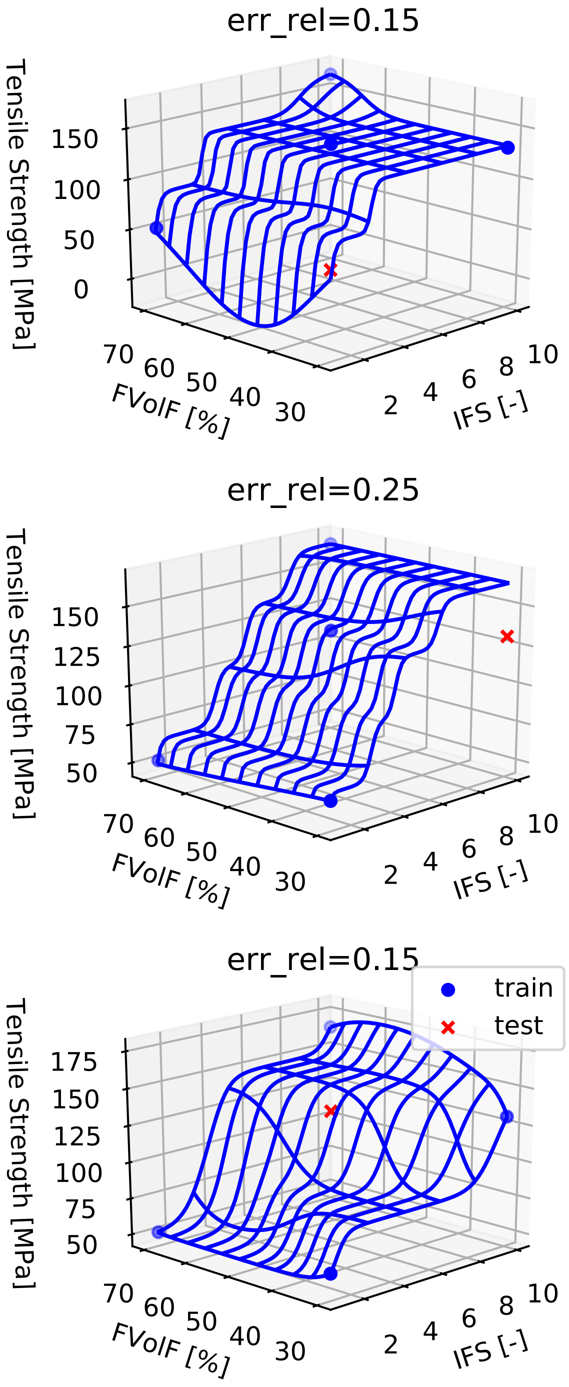


Figure 5.9: Illustration of the *Cross-Validation* method with *5fold* procedure, detecting overfitting. The available database consists of five points: two features FVolF & IFS, and one target *Transverse Tensile Strength*. Training data is shown as blue dots, testing data as red *X*. The parameters of the *MLPRegressor* are: **hidden_layer_sizes**=(100,10), **alpha**=0, **activation**=tanh, **solver**=L-BFGS, **tol**=0.00001. The relative error of the trained NN at the position of the testing datapoint is annotated above each plot. The mean relative error is 0.57.

6

Creating a Framework for ‘Smart’ Computational Micromechanics: Synergy of DoE, CM and NNs

This chapter shall detail the proposed ’smart’ framework, combining a Design of Experiments (DoE) approach with the modelling capabilities of Computational Micromechanics (CM), and the universal approximation capability of Neural Networks (NNs). Based on a user-specified number of input parameters to vary, the DoE method fills the parameter space iteratively with datapoints which shall be computed, i.e. combinations of input parameters for which the corresponding transverse strength shall be simulated. These simulations are then carried out with the CM model, and the NN is used to create a comprehensive surrogate model of the underlying functional relationship.

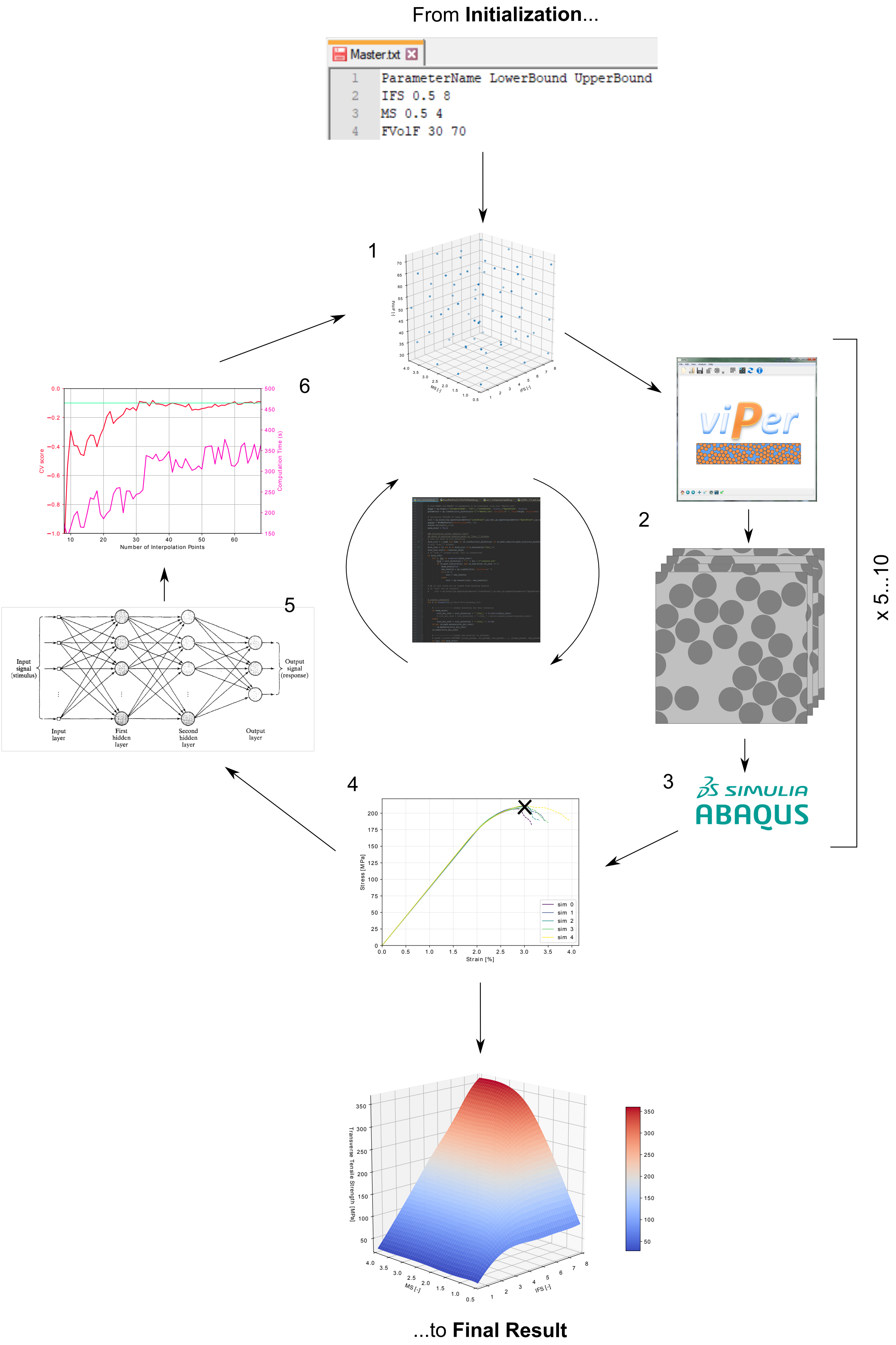
First, we present an overview of the workflow of the entire framework, which allows us to precisely situate the familiar CM and NN pieces, presented in Chapter 4 and Chapter 5, within the framework. Then, we explain the remaining new pieces of the framework, focusing on the DoE implementation and its interplay with the other two methodologies. Based on a practical example of the NN fitting to a given dataset of 10 datapoints, we can now show how the *mean relative error*, calculated during the NN training=fitting process, can serve as a measure to decide how many datapoints we need, in order to obtain a sufficiently accurate surrogate model. After discussing some of the challenges towards automating this decision, we conclude with presenting an example of a final result of the framework: the surrogate model of the transverse tensile strength of a UD FRPC ply, as a function of the constituent strengths and the Fiber Volume Fraction.

6.1. Overview of the Framework

Figure 6.1 shows a schematic overview of the entire work-flow of the framework. Starting with the user-defined parameters to be varied and their corresponding value ranges, the DoE centered process computes datapoint by datapoint in an iterative manner, until the NN fit to the ensemble of datapoints reaches a certain accuracy. The final result is then this sufficiently accurate NN fit, i.e. the final surrogate model.

The simulation workflow of the CM model, as presented in Chapter 4, is now shown embedded into the framework on the right of Figure 6.1, Steps 2-4. The readily-simulated datapoints are straight fed into the NN, presented in Chapter 4 and now implemented as Step 5 within the framework, see left side of Figure 6.1. From top to bottom the entire framework can be divided into the following steps:

- **Initialization**: Definition of input parameters and their value ranges: The user defines in a simple *Master.txt* file the input parameters to be investigated in this study, for example the strengths of the constituents (matrix, fiber-matrix interface) and the Fiber Volume Fraction, including their value ranges. The final result of this study will then be an analytical surrogate model, relating these input parameters to the transverse strength of a UD FRPC ply.



```
ParameterName LowerBound UpperBound
IFS 0.5 8
MS 0.5 4
FVolF 30 70
```

Figure 6.1: Workflow of the proposed framework for ’smart’ Computational Micromechanics.

- **Step 1**: Determination of the next datapoint to be simulated: Based on common DoE principles, our so-called *point-generator* determines the next combination of input parameters, i.e. the next datapoint to be simulated.
- **Steps 2 to 4**: Generation of SVE's, Simulation, and Post-Processing: Based on the specific values of the input parameters for this datapoint, the SVEs are generated, simulated under transverse tension or compression, and once fice valid results have been generated, a single final averaged strength value is calculated. These steps were detailed in Chapter 4.
- **Step 5**: NN fit to all computed datapoints: As soon as a minimum number of datapoints has been simulated, they are automatically fitted to a NN. The optimal values for the parameters of the NN are automatically determined through *Cross-Validation*, as discussed in Section 5.3.2.2 and to be illustrated later-on in Section 6.4. The result of Step 5 is a NN approximation of the input-output relationship between the input parameters specified initially, and the transverse strength of the UD FRPC ply.
- **Step 6**: Accuracy calculation of the NN approximation: The accuracy of the NN fit to the given number of datapoints can judged based on the *mean relative error*, calculated during the NN training by *Cross-Validation*. If the accuracy is considered insufficient, i.e. a too high mean relative error, another datapoint is commanded to be simulated, and the process restarts with Step 1. If sufficient, i.e. below a user-defined threshold, the final result is achieved.
- **Final Result**: Once the user-defined accuracy threshold is achieved by the NN approximation in Step 6, the final result has been achieved: a sufficiently accurate analytical approximation of the input-output relationship between the input parameters specified initially, and the transverse strength of the UD FRPC ply. A convenient visualization of such a function can be a 3D plot.

The entire process is automated through a central custom made *Python* script, depicted in black in the center of Figure 6.1. This script is commands the different steps of the whole process, calls the other *Python* scripts necessary for that. Steps 2 to 4 were explained in detail in Chapter 4, while Step 5 was explained in Chapter 5. The remaining new steps shown in Figure 6.1 are specific to the iterative DoE process and shall be explained hereafter.

6.2. Initialization

The starting point of every parametric study is the definition of the input parameters which shall be varied, and in between which ranges of values. For the example of Figure 6.1, the corresponding *Master.txt* files is shown in Listing 6.1. Each line defines the name of the parameter to be varied, together with its lower and upper bound. For this example study, three input variables have been chosen:

- Geometrical Parameters:
  - FVolF: Total fiber volume fraction, with $30\% \leq FVolF \leq 70\%$
- Material Parameters:
  - IFS: Scaling factor of the strength of the fiber-matrix interfaces, with $0.75 \leq IFS \leq 8$
  - MS: Scaling factor of the strength of the matrix, with $0.5 \leq MS \leq 4$

All other parameters of the SVE generation and the CM model have been fixed at their nominal values, as listed in Chapter 4. For this particular study, transverse *tension* was the chosen loadcase. Accordingly, the function to be approximated becomes

$$\sigma_{transverse,tension} = f(FVolF, IFS, MS) \tag{6.1}$$

with $\sigma_{transverse,tension}$ denoting the transverse strength of the UD FRPC ply, and $FVolF$, IFS, and MS denoting the input parameters as defined with their value ranges above.

```
ParameterName LowerBound UpperBound
IFS 0.75 8
MS 0.5 4
FVoLF 30.0 70.0
```

Listing 6.1: *Master.txt* file of the example shown in Figure 6.1 - defining the input parameters to be varied and their value ranges

Following the definition of the input parameters and the loading case, two pivotal questions arise:

1. **How many** datapoints are necessary to approximate Equation 6.1 with sufficient accuracy?
2. **Which** datapoints, i.e. interpolation points, are necessary for that?

The former question is addressed in Section 6.5 - Step 6, but first we shall discuss the latter question.

6.3. Step 1: Determination of the Next Datapoint

In the case of the parametric studies performed for this master's thesis, the function to be approximated is unknown and involves several inputs which have to be related to a single output variable. Since we are dealing with relations between material properties and geometrical parameters, no higher order functions are expected. Since we further can simulate any kind of combinations of input parameters, we are free to choose *which* datapoints to compute within the bounds of the parameter space, as defined in the file *Master.txt*, and *how many* of them.

Assuming no prior knowledge about the function to be approximated, several so-called *space-filling* design methods have been developed [67], filling every region of the parameter space equally with sample points. The most simple way to do this is with an evenly-spaced grid. For high-dimensional functions, different measures exist to define the distance between points, with a common choice being the Euclidian distance. The reviewed cases of DoEs applied in the field of composite structures used such a regular grid to sample the parameter space [38]. An alternative and more complex approach is to introduce a random component to the regular grid [64], with example methods being *Latin Hypercube Sampling* [45] and *Sobol Sequence* [68].

Since the author was not aware of these latter methods until the very end of the master's thesis, and since the regular grid seemed to achieve good results in the reviewed literature [38], such a regular space-filling method was chosen for the proposed 'smart' framework. Since the author was not able to find a readily available open-source implementation for the problem of equally spacing points in n-dimensions in an iterative manner, a simple *point generator algorithm* was developed in *Python*. By way of example, Figure 6.2 shows a so-created distribution of 69 sample points, within the parameter space defined in Listing 6.1. To briefly explain the working of the developed algorithm, we will refer to minimal examples in 2D and 3D in the following section.

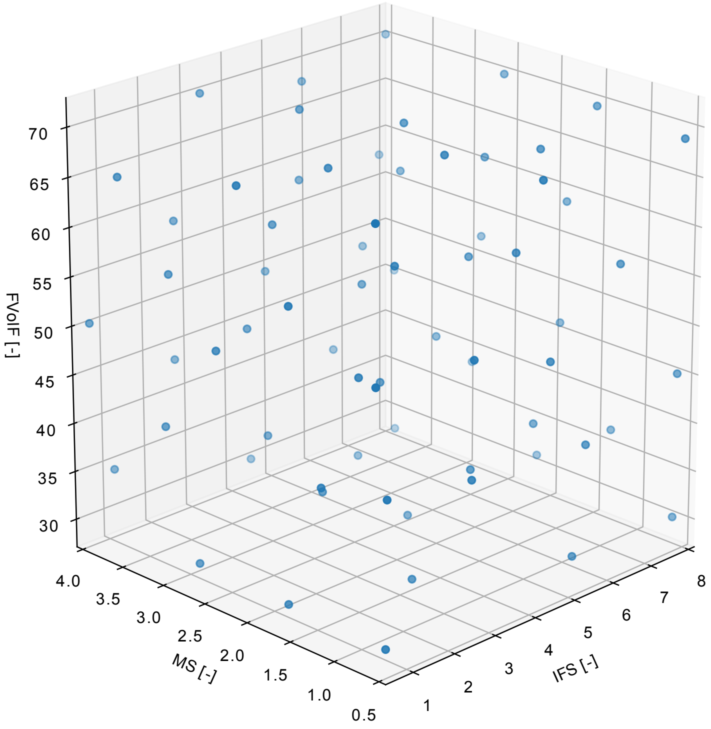


Figure 6.2: Equally spaced distribution of 69 datapoints within the input parameter space defined in Listing 6.1.

6.3.1. Point Generator Algorithm

For simplicity, a brute force approach was chosen, positioning at random 50000 points within the parameter space, calculating the distances between these new points and the existing points, and then settling with the best match. Distances between two points in n-dimensions were calculated as the euclidian distance, and best match was defined as the point which is as far away as possible from all existing points, i.e. maximizing the minimum Euclidian distance. The reviewed example from [38] used a seemingly complex "sequential multi-objective optimisation based on a goal attainment method scheme" [38][24], and did not disclose his code publicly. Since the brute force approach yielded both accurate and fast results, no further attempts were made to develop an exact analytical method. For an extension of the proposed framework to higher numbers of input parameters (>6), the computation time of the algorithm may become limiting, and a faster/exact algorithm may have to be implemented.

Figure 6.3 illustrates the iterative working of the algorithm. On the left side, a single point $(x1, y1) = (0, 0)$ was given as a baseline. Then, the algorithm added iteratively 9 more points, each time at the location furthest away from all other points. On the right hand side of Figure 6.3, the same is repeated in three dimensions, i.e. for three input parameters. Given two points $(x1, y1, z1) = (0, 0, 0), (x2, y2, z2) = (1, 1, 1)$, the algorithm determines four more points, each at maximum equal distance to all preceding points. In the 3D example, it shall be noted that for two baseline points in two opposing corners of the cube, none of the following points are positioned exactly in one of the other 6 corners.

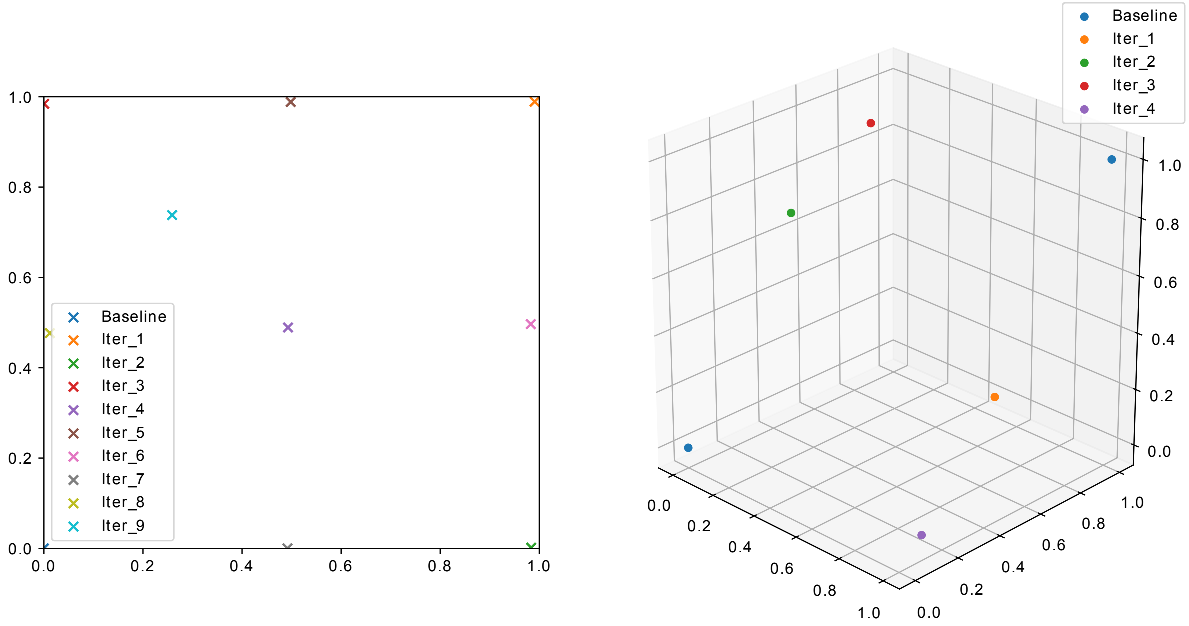


Figure 6.3: Left: Results from point generator algorithm in 2D: Baseline of $(x1, y1) = (0, 0)$, followed by 9 more iterations. Right: Results in 3D: Baseline of $(x1, y1, z1) = (0, 0, 0), (x2, y2, z2) = (1, 1, 1)$ followed by 4 more iterations.

To summarize from an outside perspective, Step 1 of the proposed framework takes as input all datapoints which already have been computed, and returns a single new datapoint, i.e. combination of input parameters, to be simulated in the coming iteration. For the very first iteration, two datapoints are defined as the baseline. For the example outlined in Listing 6.1, these would be $(0.75, 0.5, 30.0)$ and $(8, 4, 70.0)$, analog to Figure 6.3 on the right hand side. In general, the first baseline datapoint takes as coordinates all lower bounds of the parameter space, and the second datapoint all upper bounds.

6.4. Step 5: Neural Network Fit to all Computed Datapoints

The final result of Step 4 is the averaged strength value of the last simulated combination of input parameters. This combination of input parameters and the resulting strength value forms a complete datapoint. Together with all previously computed datapoints, they constitute the currently available *database* of the parametric study, which is the input for Step 5. The output of Step 5 is then a NN fitted to this database, i.e. the current NN approximation of the sought-after input-output relationship, in this example Equation 6.1. In the following step, Step 6, the accuracy of the current NN approximation is then used to judge whether more datapoints have to be computed, in order to improve this accuracy. Since the chosen measure of accuracy in Step 6 is a side-result of the training process in Step 5, we shall briefly illustrate the latter with a practical example.

Details of the NN implementation for the framework are explained and motivated extensively in Chapter 5. In brief, the input parameters of the database are first linearly scaled from their original ranges to fall in between a range of -1 and 1, see Section 5.3.4. Then, before any training of the NN may begin, the NN architecture has to be defined, i.e. the parameters of the *MLPRegressor* have to be chosen. As outlined in Section 5.3.2.2, only two parameters have to be considered: **hidden_layer_sizes**, and **alpha**. For a detailed explanation of these parameters, please refer to Section 5.2 and Section 5.3.2.2. Conveniently, the chosen training method of *Cross-Validation* now does not require to know a-priori the exact values to choose for these parameters, which is also impossible to know. Instead, a range of values are considered for each parameters, and so a wide array of NN architectures will be evaluated and compared. For the example presented throughout this chapter, the following parameter ranges are evaluated:

- **hidden_layer_sizes**: (10), (20), (40), (60), (80), (100)
- **alpha**: 0.00001, 0.0001, 0.001, 0.01, 0.1 ,0.2, 0.3, 0.5, 0.7, 0.8

Now, according to the procedure of *repeated-5Fold*, the database is split into training and testing data by a ratio of 80/20%. The first NN, i.e. **hidden_layer_sizes**=10, **alpha**=0.00001, is fitted to the training data, and its approximation accuracy is calculated with the testing data. To illustrate this with some real numbers, we shall take an example database of 10 datapoints: the first 10 sample points within the parameter space shown in Figure 6.2, based on the input parameters defined in Listing 6.1.

For this database, the aforedescribed first NN achieves a mean relative error of 44.9%[1]. Then, this process is repeated four more times for different splits of the database into training and testing data. Said NN so achieves mean relative errors of 41.9%, 4.1%, 9.2%, and 133.1%. Eventually, the mean relative errors of all splits are averaged and form a single value describing the approximation accuracy of the first NN, in this case an average mean relative error of **46.6%**.

However, this was only the evaluation of the *first* NN, i.e. of the first combination of *MLPRegressor* parameters as listed above. This whole training process is now repeated for all other NN architectures which were set out initially. For each one, the average mean relative error is calculated, describing the overall accuracy of its fit to the database. This allows finally to choose the single set of *MLPRegressor* parameters which results in the best NN approximation of Equation 6.1, *for the given amount of datapoints.*

Figure 6.4 shows visually the average mean relative error plotted over all NN architectures which were evaluated through *Cross-Validation* in this example. In line with the findings of the literature review [29], no general rule can be deduced which could tell the user a-priori which kind of parameters may result in the best trained NN. Throughout the work of this thesis, it was noticed that generally two kinds of combinations of the two parameters result in the best NN approximations: high number of neurons plus high alpha, or the opposite case of only a few neurons and a small alpha.

For the parameter grid of this example, the best NN so found has 40 hidden neurons and a scaling factor of the *L2 regularization term* of 0.00001, i.e. **hidden_layer_sizes**=40, **alpha**=0.00001. This NN architecture achieves an average mean relative error of **33%**. However, it has to be noted that this result strongly depends on the size of the database. The above results were calculated for a small database of 10 datapoints. For increasing size of the database, the mean relative error generally decreases as expected: An infinite number of interpolation points would leave little to do for the surrogate model, and little to gain from the whole framework. As discussed in the following section on Step 6, this can be used to answer the second pivotal question of the DoE part of the 'smart' framework: *How many* datapoints are necessary to approximate the sought-after input-output relations, in this example Equation 6.1, with sufficient accuracy?

6.5. Step 6: Accuracy Calculation of the Neural Network Approximation

The average mean relative error calculated in Step 5 was for a database of 10 datapoints. As mentioned, its value generally decreases for increasing size of the database. This is a useful property, which we shall use to determine the minimum number of datapoints, necessary to approximate Equation 6.1 with a certain

[1] Throughout this master's thesis, the measure of *mean relative error* based on Equation 5.2 was used for comparing the quality of a fit.

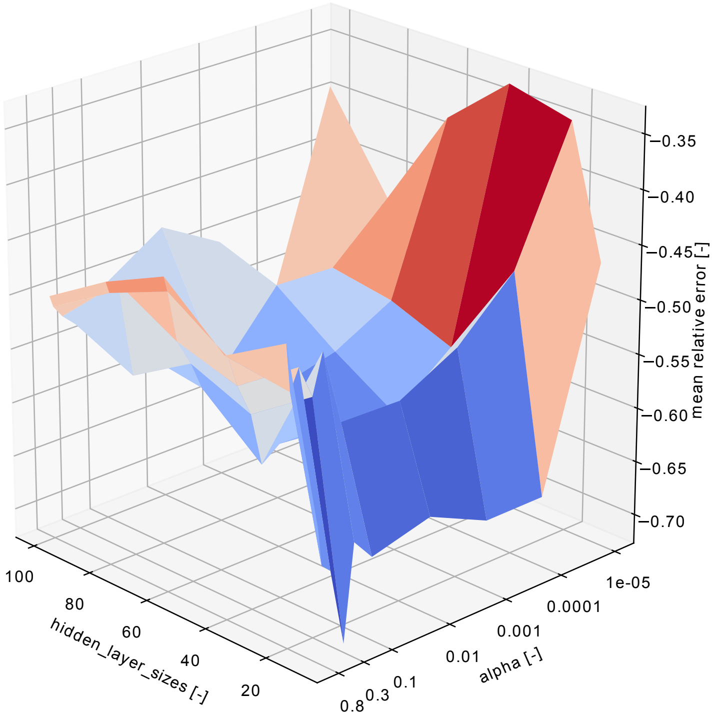


Figure 6.4: Cross-Validation results for training of various NNs on a database of 10 points in 4D. The measure of accuracy of the NN fit to the database, i.e. the average mean relative error, is plotted over the two parameters of the *MLPRegressor* class which were evaluated: **hidden_layer_sizes** and **alpha**. Every combination of parameters corresponds to a different NN architecture. Best results are achieved for 40 hidden neurons and a scaling factor of the *L2 regularization term* of 0.00001, resulting in an average mean relative error of 33%.

accuracy. To illustrate this, Figure 6.5 shows the average mean relative error of Step 5, but now for a database between 8 and 69 datapoints. On the x-axis, the number of datapoints used for the NN fitting is written. In total, 69 datapoints were simulated, corresponding to the sample points shown in Figure 6.2. On the y-axis, two measures are written.

The main result is the *average mean relative error* of the *Cross-Validation* method, as summarized in the previous section and in more detail in Section 5.3.2.2. For every iteration, i.e. every number of available datapoints, the average mean relative error of the *best* NN approximation is plotted in Figure 6.5 in red. Looking at the example of the previous section, for a database of 10 datapoints, the best combination of *MLPRegressor* parameters yielded a mean relative error of 33%. It is now up to the user to define the accuracy up to which she seeks to approximate Equation 6.1 with this parametric study. For the example of Figure 6.5, a target accuracy of 10% was set, and in total 69 datapoints were simulated.

The second result plotted in Figure 6.5 is the computation time of the NN training process. *Cross-Validation* is a computationally expensive method, especially for large datasets and NNs with many hidden layers and hidden neurons. For reference, the computation time of the NN training has been recorded and is drawn in pink over the number of datapoints in Figure 6.5.

6.5.1. Challenges Towards Full Automation

An explicit objective for the proposed ’smart’ framework of this master’s thesis is full automation. Ideally, the user only has to specify the function that she is interested in, i.e. inputs & output, and the target accuracy of the approximation that she seeks. Then, the process automatically determines how many datapoints are necessary to achieve this accuracy, and runs by itself until the approximation function has been determined. All this is possible with the proposed framework, but Figure 6.5 suggests that the question of *how many datapoints?* may need a more complex criterion. It can be seen that the average mean relative error does converge to a minimum value, but not steadily. The set out accuracy of 10% mean relative error is first reached for 31 datapoints, but then diverges again up to 14% until "final" convergence after 63 datapoints.

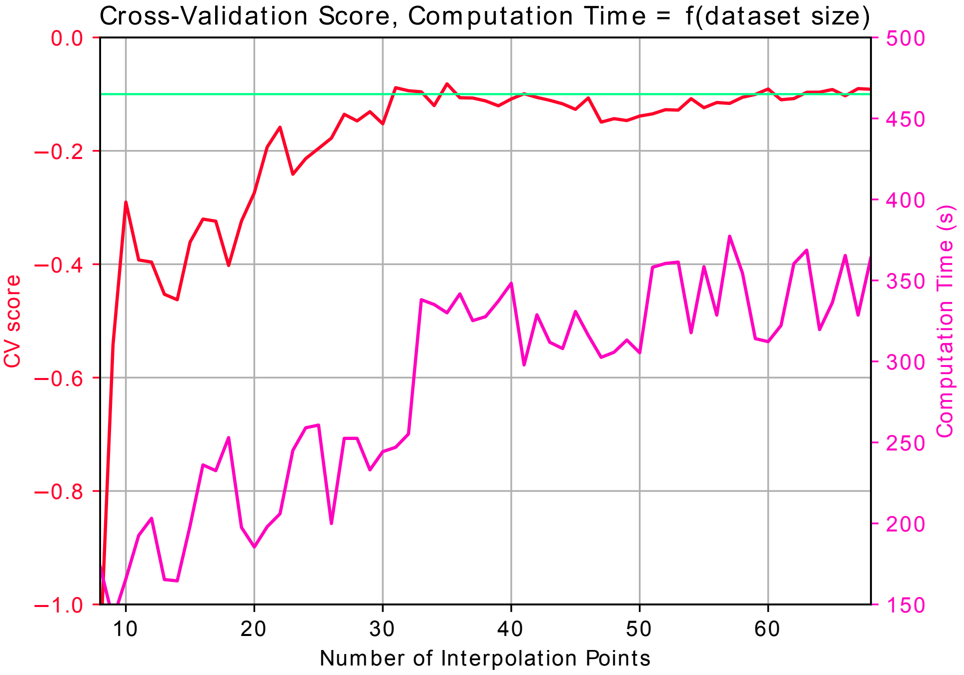


Figure 6.5: Cross-Validation score, i.e. average mean relative error, and NN computation time plotted over the number of datapoints used for NN fitting of Equation 6.1. The vertical green line is drawn at a value of 10%.

As can be seen, the definition of an exact measure of accuracy is challenging and up for discussion. During the presentation of results in the following chapter, the plot of Figure 6.5 will be shown for more studies, which lays a better foundation for subsequent discussion. The presentation of results in Section 7.2 will further show that the *minimum* value for the average mean relative error, as possible to be achieved with a reasonable amount of datapoints, differs from one to another study. Significant influence is ascribed to the load-case, tension or compression, and the choice of input parameters to vary. This leads to the fact that it may not be straightforward for the user to define a-priori a target accuracy, as this value may not be possible to achieve.

Lastly, the target accuracy has to be chosen relative to the extent of the strength effect to be approximated. In the example of Equation 6.1, the transverse tensile strength eventually varies between 33 MPa and 361 MPa, i.e. by a factor of > 10 over the specified ranges of the input parameters, as to be seen in Figure 6.6. Therefore, a target accuracy of 10% is very low compared to the overall extent of the strength variations. If the user is only interested in the overall effect of IFS and MS on the transverse tensile strength, a target accuracy of 30% may well be enough. Referring to Figure 6.5, this would reduce the necessary number of datapoints from 31 to 20, thus a reduction in datapoints and furthermore in overall computation time of about one third [2].

With regard to the objective of implementing a *fully* automated process for parametric studies, the target accuracy could also be automatically determined. A fixed *relative target accuracy* could be defined, e.g. fixed at a value of 5%. This would mean that the parametric study is stopped as soon as the average mean relative error is smaller than 5% of the entire strength variation over the simulated input parameter space.

For the example of Equation 6.1, the absolute strength variation is 328 MPa, thus relative to the minimum strength value of 33 MPa, this amounts to a change of 994%. Multiplied with the *relative target accuracy* of 5%, this would result in a *mean relative error* of 49.7%. Referring to Figure 6.5, the parametric study may in this case already be stopped after 10 datapoints. Whether such a coarse approximation of Equation 6.1 is sufficient is up to the user to decide. It depends on whether a coarse but global approximation of the function is sought-for, or a finer approximation up to local details.

[2] The computation time of the CM model is much higher than the one of the NN training.

6.6. Final Result

The final objective of the proposed framework is to obtain a global approximation function of the input-output relationship, based on a minimum amount of datapoints. For the example used throughout this chapter, we obtain as a final result a trained NN, representing the function

$$\sigma_{transverse,tension} = \tilde{f}(MS, IFS, FVolF) \tag{6.2}$$

Herein, $\tilde{f}(...)$ now denotes the found *approximation* function, describing the material property relationship of the simulated UD FRPC plies. The trained NN is an analytical function, albeit a very complex one, and results are thus best visualized through a plot. Since Equation 7.3 has three input variables and one output variables, it is difficult to visualize all input variables at the same time. Figure 6.6 shows Equation 7.3 for a fixed FVolF of 70%. The contour lines of the surface are plotted on the respective planes.

Recalling the discussion of the previous section, the NN approximation depends strongly on the size of the database. The plot of Figure 6.6 corresponds to a database of 69 points, i.e. the maximum number of points that have been simulated for this parametric study. Figure 6.7 now shows the same results for a smaller database, i.e. if the study would have been stopped earlier. Results are shown for a database of 10, 21, 32, and 69 datapoints. These values have been chosen with reference to Figure 6.5 and the discussion of the preceding section.

The question had been formulated whether 10 datapoints are sufficient to approximate Equation 6.1. As Figure 6.7 shows, 10 datapoints are sufficient to capture the global minimum and maximum of the function. To be fair, this is not a general rule, but due to the fact that the first two datapoints to compute, $MS, IFS, FVolF = (0.5, 0.75, 30.0)$ and $(4.0, 8.0, 70.0)$, happened to result in the global minimum and maximum tensile strength. At the same time, the NN approximation with 10 datapoints fails to show the linear slopes as seen for 69 datapoints, as well as the asymptotic behavior for low values of MS and increasing IFS. These differences can be explained with the distribution of integration points by equal spacing. For only 10 datapoints, none may have been positioned in the corner of $MS, IFS = (0.5, 5...8)$ yet, thus the NN obviously can only extrapolate in this area.

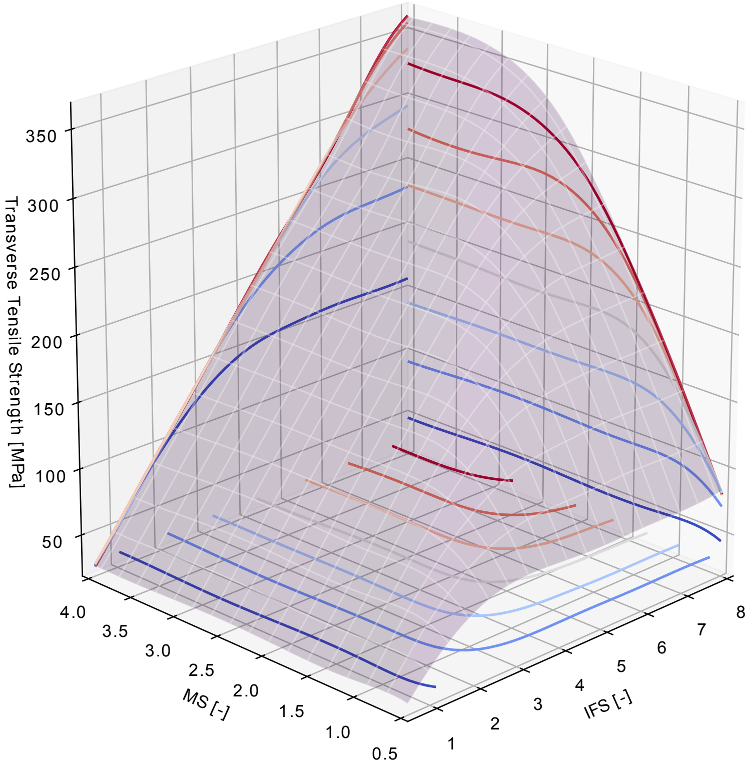


Figure 6.6: Plot of the NN approximation of Equation 6.1, for a fixed FVolF of 70%, and based on 69 datapoints. The contour lines of the 3D surface are projected onto the planes of the plot.

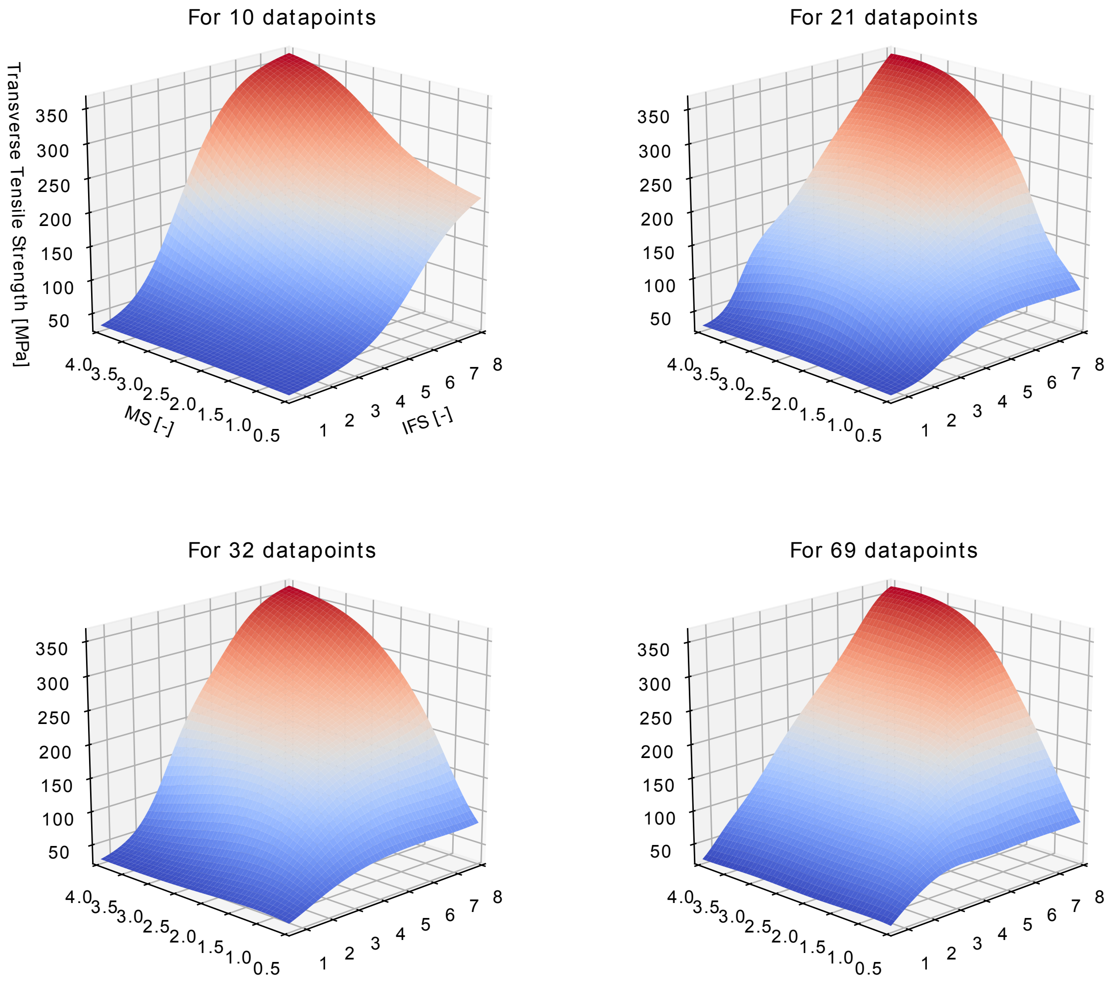


Figure 6.7: Plots of the NN approximation of Equation 6.1, for a fixed FVolF of 70%, and based on 10, 21, 32, and 69 datapoints.

On a last note, and as mentioned before in Section 6.3, parametric studies with very computationally expensive datapoints may benefit from an improved scheme to determine the best positions of the interpolation points. Such a scheme might take into account the local gradient of the approximation function, and so achieve a better approximation for the same number of datapoints. For the parametric studies performed within this master's thesis however, limited to a maximum of four input parameters, the computation of the necessary amounts of datapoints was yet cheap enough to not require such a rule.

6.7. Summary

Hereby, we conclude the chapter presenting the proposed framework for 'smart' Computational Micromechanics. We have unveiled the overall workflow of the framework, combining a Design of Experiments (DoE) approach with a Computational Micromechanics (CM) model and a Neural Network (NN), in order to obtain a surrogate model of the transverse strength of a UD FRPC ply. Further, we could show where the CM and NN method fit within the framework, and how they interact with the DoE method. The latter has to answer two important questions: *Which* datapoints shall be computed, and *how many*? We address the first question with a regular grid sampling of the parameter space, for which we developed a simple algorithm in *Python*. The second question is more ambiguous. We showed how to use the average *mean relative error*, a result of the training process of the NN, to judge the convergence of the surrogate model. Still, it remains up to the user to define the required accuracy, and so the necessary number of datapoints which have to be simulated.

After having presented the proposed 'smart' framework, both as a whole as well as its three underlying methodologies, we can now finally present the five performed example studies in the following chapter.

7

’Smart’ Prediction of Composite Ply Properties: Transverse Strength

The proposed ’smart’ framework, as presented in the previous chapter, is universal in the sense that its Design of Experiments (DoE) and Neural Network (NN) parts can operate on any kind of data. By having implemented our Computational Micromechanics (CM) model, we restrict ourselves to its specific predictive capabilities: the behavior of a UD FRPC ply under transverse loading up to failure. Still, the CM model generates a wealth of data for every simulation, with the stress-strain curve only being a very simple and descriptive example. In order to implement, test and demonstrate the capabilities of the proposed framework within the given timeframe, we had to limit ourselves to a single value of the stress-strain curve, eventually choosing maximum stress, i.e. the transverse strength of the simulated UD FRPC ply, for reasons outlined in Chapter 2.

However, with the same already generated data, it would be equally possible to exploit other output variables of interest, thinking for example about strain, strain energy, or to let the NN approximate the entire stress-strain curve [7]. Choices beyond that are left to the imagination of the reader, within the bounds of the used CM model, or even further, by replacing it with any other computer model. More ideas for future work, short and longterm will be discussed in Chapter 9. In the ensuing chapter now we shall present the five example studies which we performed with the framework, relating the transverse strength to all geometrical, material, and modelling parameters which we considered most relevant.

First, two sensitivity studies are presented which were necessary to characterize the used methodology of Statistical Volume Element (SVEs) for our specific use case. The influence of the SVE size, i.e. the number of fibers within the SVE, as well as the influence of the minimum inter-fiber distance *Dmin* on the transverse tensile strength were investigated. Based on these results, appropriate nominal values were chosen for these two parameters for the following studies.

Then, three studies were performed to characterize the transverse strength of a UD FRPC ply, and to explore possible ways of improving today’s FRPC materials in this regard. First and foremost, the dominant parameters determining transverse strength are the constituent strengths, i.e the strengths of the matrix and the fiber-matrix-interface. Accordingly, the first study explores the effect of these two parameters. After determining the surrogate model of these primary effects, we explored the influence of fiber-by-fiber hybridization. The second study mixes fibers with circular and 4-lobed cross-sections within the same SVE, and the last study mixes two different fiber diameters. To obtain a complete picture, all studies varied fiber volume fraction as an additional parameter and were performed for both transverse tensile and compressive loading.

7.1. Methodological Findings

Before being able to characterize transverse strength with the developed framework, we shall use it to characterize the influence of two important modelling parameters, common to all simulations to be performed:

- minimum inter-fiber distance, and
- number of fibers within the SVE, which for a fixed fiber diameter value boils down to the size of the SVE

As revealed in the literature review and summarized in Section 3.1.2.1, both have a significant influence on the obtained transverse strength values for the used method of SVEs at microscale. The size of the SVE is an important parameter for the SVE methodology in general, whereas the minimum inter-fiber distance is a parameter specific to the modelling of FRPC materials at microscale. The following two sections shall describe the studies which were performed to investigate the influence of these two parameters.

7.1.1. Transverse Strength as a Function of Minimum Inter-fiber Distance and IFS

One of the findings of the literature review is the significant effect of the minimum inter-fiber distance on results generated by CM using SVEs. In order to confirm this effect and measure its magnitude for our model, a parametric study was performed varying the following two input parameters and ranges:

- Geometrical Parameters:
  - *Dmin*: minimum inter-fiber distance: 0.1 µm to 0.75 µm
- Material Parameters:
  - IFS: fiber-matrix interface strength: 0.5 to 8

The IFS was added as a second parameter, in order to evaluate the effect of *Dmin* over the different possible failure regions to be explored in another study, see Section 7.2.1. Figure 7.1 shows two example SVEs generated for this study now. On the left, for the lower limit of $Dmin = 0.1\,\mu\text{m}$, and on the right for the upper limit $Dmin = 0.75\,\mu\text{m}$. The final approximation function to be found becomes

$$\sigma_{transverse,tension} = \widetilde{f}(Dmin, IFS). \tag{7.1}$$

A total of 26 datapoints were computed and none of them aborted. Since we are only varying two input parameters, no pre-defined stopping criterion was used to determine the necessary number of datapoints, but this was checked visually based on Figure 7.2. The latter shows the spatial distribution of the 26 datapoints, as well as the final NN approximation of Equation 7.1. Since the magnitude of the IFS effect is of much greater magnitude than the *Dmin* effect, Figure 7.2 shows on its right side a slice of the surface on the left, at a fixed IFS value of 1. Several observations can be readily made:

- As expected, the magnitude of the IFS effect on transverse tensile strength is much greater than the effect of *Dmin*. This makes it difficult to judge the latter, but Figure 7.2 shows on its right side that the minimum inter-fiber distances changes the transverse tensile strength by 5 MPa for the chosen parameter ranges. Relative to the nominal strength obtained at $IFS = 1$, this is about 10%. For our later studies minor effects like the mixing of circular and 4-lobed fibers under tension, see Section 7.3.3, this is very significant. For the study of the primary effects of the constituent strengths however, a change of 5 MPa is much less significant, as to be seen in Section 7.2.

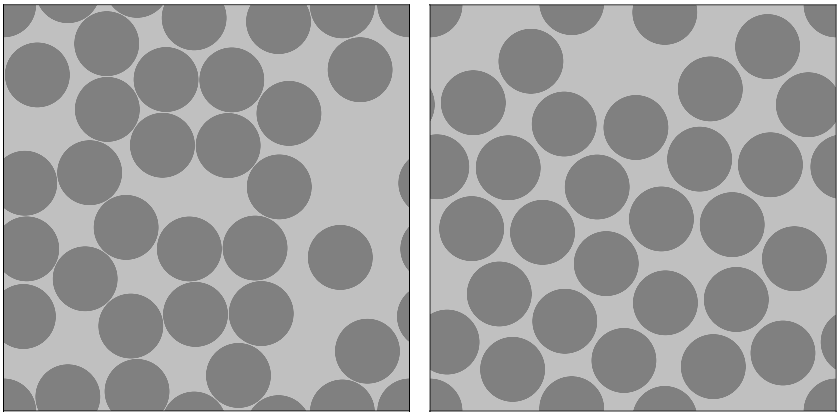


Figure 7.1: Two SVEs generated through *Vip3r* [30], with minimum inter-fiber distance of 0.1 µm on the left, and 0.75 µm on the right.

- For increasing IFS, the total magnitude of the effect first remains constant at about 5MPa, thus diminishing in comparison with the overall strength increase. For the upper limit case of $IFS = 8$, i.e. the matrix-dominated failure region see Section 7.2.1, the effect of *Dmin* disappears. Apparently, the minimum inter-fiber distance affects only the failure of the interfaces. When these are assumed to be perfectly bonded, the minimum inter-fiber distance has no effect on the failure of the matrix ligaments between the fibers.

- Lastly, it has to be noted that some datapoints are unusually close to each other, see Figure 7.2. This was the result of an issue with an earlier version of the point generator algorithm, used in this study performed early on. For subsequent studies, this was corrected. However, it does not change the quality of the presented results.

Concluding, all following studies shall be performed at the same fixed value for the minimum inter-fiber distance. Thereby, we can eliminate the shown influence of *Dmin*, and ensure that results are comparable. As mentioned before, we will use a nominal value of $Dmin = 0.35$µm, identical to the original model of M. Herraez [30]. The studies on *hybrid* FRPC materials will further show that it is still important to be aware of the magnitude of the *Dmin* effect. For the study on mixing circular and 4-lobed fibers in Section 7.3, the SVE generator *Vip3r* consistently deviated from the specified value of $Dmin = 0.35$µm, such that the final results are obtained at a lower value of $Dmin = 0.2$µm. The study on mixing two different fiber diameters in Section 7.4 also highlights that *Dmin* is an important influence, when explaining why the SVEs of mixed fiber diameters generally fail around the larger fibers.

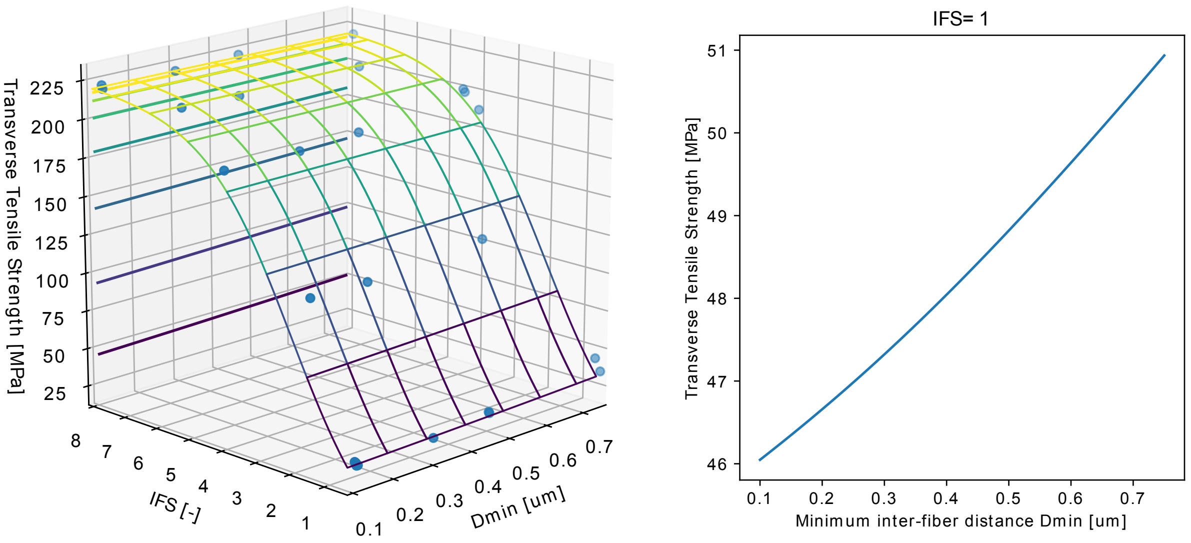


Figure 7.2: NN approximation of Equation 7.1 for a total of 26 datapoints. The output variable transverse tensile strength is plotted over the two input variables minimum inter-fiber distance *Dmin* and fiber-matrix interface strength IFS. On the right hand side, a slice of the NN calculated surface is presented at $IFS = 1$, for better visualization of the effect of *Dmin*.

7.1.2. Transverse Strength as a Function of SVE Size and Fiber Volume Fraction

As previously discussed in Section 4.4.2 and Section 4.4, another important parameter for all simulations based on the method of SVE is the number of fibers within the volume element. For a fixed fiber diameter, in our case 8µm, the number of fibers within the SVE is determined by the size of the SVE, in our case nominally 50µm x 50µm. To validate this choice over the whole range of investigated FVolFs, a study was performed with the following two input parameters and ranges:

- Geometrical Parameters:
  - SVE size: 30µm to 100µm
  - FVolF: 30% to 70%
- Material Parameters: none

For this sensitivity study we are interested in two output variables: First, we want to know how the transverse tensile strength depends on the SVE size, such that we can choose a SVE big enough for this dependence to be sufficiently small. Further, we want to make sure that our chosen SVE size is valid over the whole range of FVolFs simulated in the course of the following studies. As to be seen, for low FVolFs, we will run into limitations of the SVE generator *Vip3r*, which was not primarily designed for such applications.

Second, we are interested in the *relative standard deviation* of the transverse strength results, calculated for the each five SVEs computed for each datapoint. As discussed in Section 4.4.2, this deviation should generally converge towards a minimum when increasing the number of fibers within the SVE. Accordingly, we can use its value to monitor the quality of our results. In this study for example, it helps us to illustrate the inadequacy of *Vip3r* for low FVolFs in Figure 7.4 on the right. For all following studies, we will keep plotting this deviation to have a visual check of the quality of the results, over the entire parameter space.

For illustration, Figure 7.3 shows two example SVEs of 100µm x 100µm generated for this study. On the left, for $FVolF = 30\%$, and on the right for $FVoLF = 70\%$. The minimum SVE size of 30µm is indicated in red, and the eventually chosen nominal size of 50µm in green. Summarizing, the final approximation function to be found becomes

$$\left(\sigma_{transverse,tension}, \frac{S_{\sigma transverse,tension}}{\sigma_{transverse,tension}}\right) = \tilde{f}(SVE_{size}, FVolF) \tag{7.2}$$

Contrary to the previous studies, the function to be approximated now has two output variables: the familiar transverse tensile strength $\sigma_{transverse,tension}$, plus the *relative standard deviation*, with S denoting the standard deviation of the mean transverse tensile strength values, from a sample size of 5. NNs are able to approximate any kind of function with n input variables and m output variables. On a practical level, it was however found that it is easier to just split the multi-output NN function into each one function for each output variable, in this case two individual NN functions [7].

Eventually, a total of 17 datapoints were computed and none of them aborted. Just like in the previous studies, the number of necessary datapoints was determined visually by monitoring the shape of the 3D surface together with the positions of the datapoints. Figure 7.4 shows the two surfaces approximating the two output variables of Equation 7.2, as well as the spatial position of the datapoints. On the left, the transverse tensile strength $\sigma_{transverse,tension}$ is drawn, and on the right the relative standard deviation of the latter. For better readability, both plots show the contour lines at fixed values for the FVolF, i.e. $FVolF = 35, 40, 45, 50, 55, 60, 65\%$, projected onto the vertical plane.

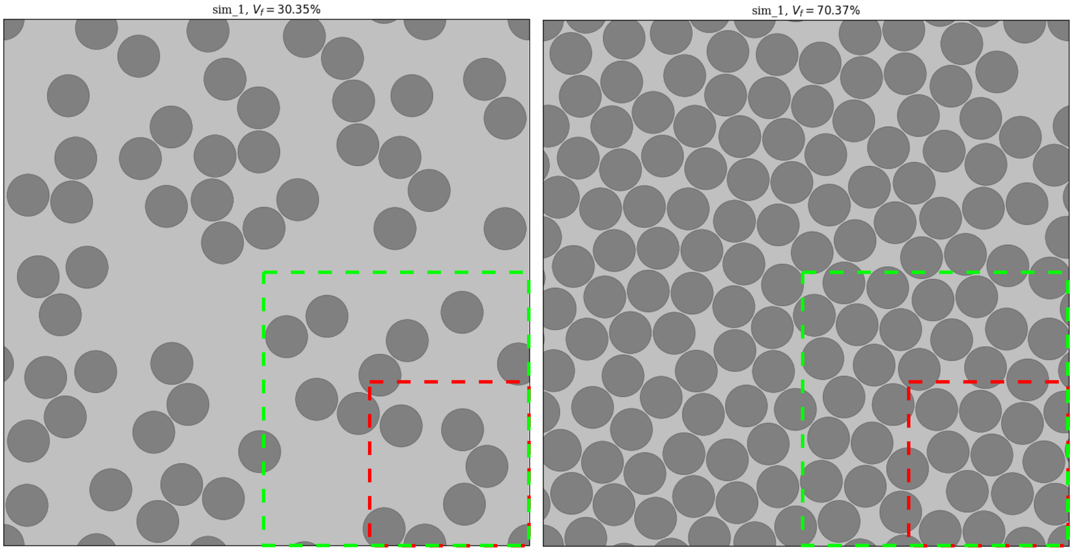


Figure 7.3: Two SVEs of 100µm x 100µm generated through *Vip3r* [30]. On the left for $FVolF = 30\%$, on the right for $FVolF = 70\%$. Marked in red is the lower end SVE size used in this study, i.e. 30µm. Marked in green is the nominal SVE size of 50µm used throughout this master's thesis.

The effect of FVolF on transverse tensile strength will be discussed later-on in Section 7.2.1. Of primary interest now is the convergence of the strength values for fixed FVolF, when varying the SVE size: The contour lines of the left plot of Figure 7.4 show that generally the transverse tensile strength values do not change by more than 2 MPa for varying SVE size. For higher FVolFs, the mean values converge steadily when increasing the SVE size, e.g the three bottom-most contour lines of the left plot for $FVolF = 65, 60, 55\%$. This changes for lower FVolFs, such that no converged value can be deduced, e.g. for the uppermost contour line at $FVolF = 35\%$.

The same trends are repeated in the right plot of Figure 7.4. As expected, when fixing the FVolF, increasing the SVE size decreases the relative standard deviation. Likewise, when fixing the SVE size, decreasing the FVolF decreases the number of fibers in the SVE and the deviation increases. However, the plot shows one exception to these general rules of the SVE methodology: For low $FVolF \leq 40\%$, increasing the SVE size first results in a *de*crease of the deviation, but then its value *in*creases again. This unexpected diverging trend towards $FVolF = 30\%, SVE_{size} = 100\,\mu m$ is attributed to the limitations of *Vip3r* for low FVolFs, as previously discussed in Section 4.4.3. The algorithm to position the fibers at random within the SVE always starts with putting one fiber in each corner. While for high FVolFs, this does not significantly distort the random fiber distribution, see Figure 7.3 on the right, it is visually noticeable for low FVolFs, see Figure 7.3 on the left or Figure 4.2. Based on the right plot of Figure 7.4, we deduce that the results for low $FVolF \leq 40\%$ shall be considered as non-converged for all following studies.

On an ending note, the magnitude of the shown effects shall be put into perspective. Both the change in strength for varying SVE size, as well as the relative standard deviations of the strength values are lower than 4%. This value has to be judged individually for each one of the following studies. When studying the effect of scaled constituent strengths on transverse strength in Section 7.2, the overall magnitude of this effect is >100%, thus rendering deviations of 4% insignificant. However, for the studies characterizing the much less pronounced hybrid effect in Section 7.3 and Section 7.4, the results of Figure 7.4 have to be well considered. The developed 'smart' framework in its proposed first version requires a fixed SVE size. As a compromise between computation time and accuracy, we chose a nominal value of 50 µm x 50 µm for all following studies. Referring to Figure 7.4, the so-generated results can be considered well-converged for all cases except low $FVolF < 40\%$. An evolution of the proposed framework may consider to automatically adapt the SVE size based on the FVolF. A simple rule could be to keep the number of fibers within the SVE constant. A more complex rule could be to maintain a constant standard deviation of the strength results.

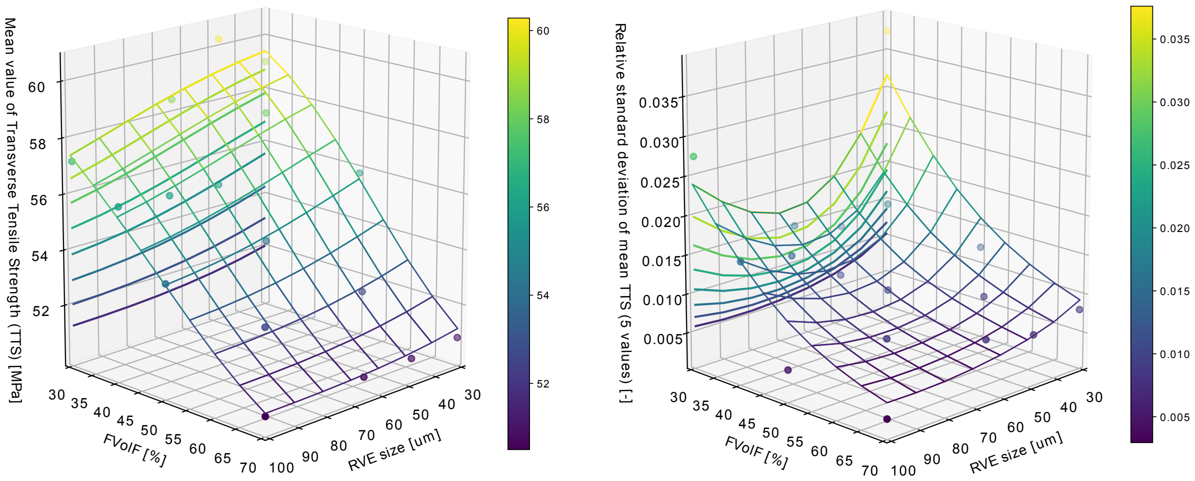


Figure 7.4: NN approximation of Equation 7.2 for a total of 17 datapoints. On the left, the mean value of the transverse tensile strength is plotted over the two input variables FVolF and SVE size. On the right, the relative standard deviation of the datapoints is plotted over the same two input variables. For better readability, the contour lines at fixed values for the FVolF, i.e. $FVolF = 35, 40, 45, 50, 55, 60, 65\%$, are projected onto the vertical plane.

7.1.3. Summary

Hereby, we conclude the presentation of the two sensitivity studies which were performed to characterize the used SVE methodology for the following studies. The effect of the two most important modelling parameters was quantified, i.e. minimum inter-fiber distance *Dmin* and SVE size. We could show that both can influence transverse tensile strength results by <5%. To minimize their effect, we maintain a nominal *Dmin* of 0.35 µm throughout the following studies, and fix the SVE size at 50 µm x 50 µm, which gives well converged results for $FVolF > 40\%$. Finally, we could illustrate an apparent limitation of *Vip3r* to generate valid SVEs with fully random fiber distributions for very low Fiber Volume Fractions.

After having determined the nominal values for these two important parameters of the used SVE method, we can now proceed with using the developed framework for its main task: to characterize the transverse strength of a UD FRPC ply.

7.2. Transverse Strength as a Function of MS, IFS, and FVolF

The first study investigates the main parameters which determine the transverse strength of a UD FRPC ply: the strengths of its constituents. Within our SVEs, these are the fibers, matrix, and the fiber-matrix interface. For today's FRPC materials, the strength of the matrix and the interface is much lower than the transverse strength of the fibers, thus failure of the latter is not considered relevant. This leaves two constituent strengths left for our investigation: MS and IFS. The final result of a study exploring the effect of these two input parameters on transverse tensile strength would be datapoints with three coordinates (MS, IFS, $\sigma_{transverse,tensile}$). Many mathematical methods exist to interpolate such a 3D surface, but few to interpolate higher-dimensional data. Therefore, a third input variable was added to be able to showcase the potential of NNs for global approximation of n-dimensional data, in our case with $n = 4$. The next best input parameter known to have a significant effect on transverse strength is FVolF, i.e. the total fiber volume fraction of the SVE. Summa sumarum, the three input parameters of the first parametric study are:

- Geometrical Parameters:
  - FVolF: Total fiber volume fraction
- Material Parameters:
  - IFS: Scaling factor of the strength of the fiber-matrix interfaces
  - MS: Scaling factor of the strength of the matrix

All other parameters are fixed at their nominal values, e.g. circular fibers of 8 µm diameter, as outlined in Chapter 4. The next hurdle is to choose the ranges of each input parameter to be varied. For FVolF, the *maximum* may theoretically be 90.8% for a perfect hexagonal pattern. The latter is never achieved in practice, and a maximum value of 70% is more realistic from a manufacturing perspective [26]. To recreate such dense but random fiber packing within a SVE is far from easy, but the *Vip3r* tool developed by M. Herraez is able to do so [30]. Regarding the *minimum* FVolF, a minimum value of $FVolF = 30\%$ was chosen after pre-studies showed numerical issues for values lower than that. Still, results in the range of $FVolF \leq 40\%$ have to be interpreted cautiously, as discussed in detail in Section 7.1.2.

For scaling factors of IFS and MS, initial results yielded that the ratio between the two determines the failure behavior. This again also depends on the loadcase, either tension or compression. In the tension case, final failure is dominated either by interface failure, or by failure of the matrix in direct vicinity of the interface. For both tension and compression it was noticed that extreme combinations, e.g. very low IFS together with high MS, particularly at low FVolFs, result in numerical instabilites and convergence issues. Figure 7.5 shows the stress-strain curves of each 5 SVEs for such extreme combinations, and subject to tensile loading. All three cases have a very high ratio of MS to IFS, with increasing FVolF. Both the plots for 30% and 50% FVolF show numerical issues, and an absence of the required 10% load drop after the maximum load. Only for the case of 66% FVolF do we finally obtain valid stress-strain curves with the required load drop of 10% to stop the simulation.

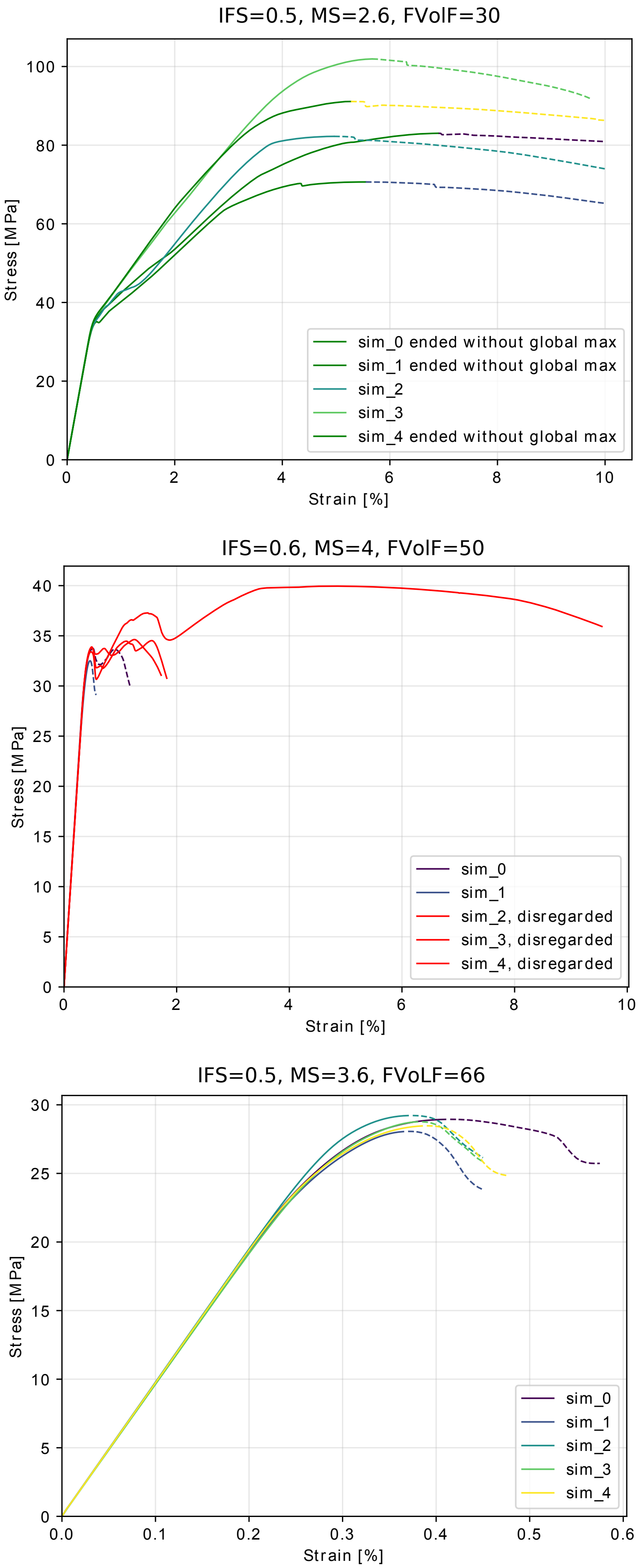


Figure 7.5: Stress-strain curves for each 5 SVEs simulated under tensile loading, but for varying input parameters IFS, MS, and FVolF. All three cases have a very high MS to IFS ratio. From top to bottom, the FVolF is increased, and only for 66% FVolF are satisfactory results achieved.

In order to automatically detect such issues, provisions to the automated implementation of the parametric studies had to be made, as discussed before in Section 4.4:

- The central *Python* script which commands the entire parametric study was extended with an abort criterion, for the case that a certain combination of input parameters would not converge. As explained in Section 4.4, nominally 5 valid results are required, and if more than 10 attempts have been made, the computation of this datapoint is aborted.
- As shown in Listing 4.3, the final strength result stored for each datapoint was extended with additional results for automatic diagnostics. This includes for example the number of SVEs which had to be generated and simulated until 5 valid stress-strain curves were obtained. These results allow the user to judge easily for which combinations of input parameters the simulations tend to fail, and thus to adjust their ranges accordingly if necessary.

As can be seen, the definition of the ranges of the input parameters is a critical step and it requires some thought on the side of the user. The implementation of the 'smart' framework has been automated to catch any simulation difficulties automatically. Lastly, the post-processed data directly indicates to the user whether his chosen ranges have to be adjusted, and if so in which areas.

7.2.1. Tension

With regard to the previous discussion on numerical issues for certain combinations of input parameters, the following ranges were chosen:

- FVolF: 30% to 70%
- IFS: 0.5 to 8
- MS: 0.5 to 4

The final approximation function to be found thus becomes

$$\sigma_{transverse,tension} = \widetilde{f}(MS, IFS, FVolF), \tag{7.3}$$

analogous to the example presented in Section 6.2 of the previous chapter. A total of 69 datapoints were simulated, out of which 5 aborted due to local maxima before the global stress maximum. Figure 7.6 shows the spatial distribution of the 69 datapoints. The aborted datapoints are marked as "x", for all other data-

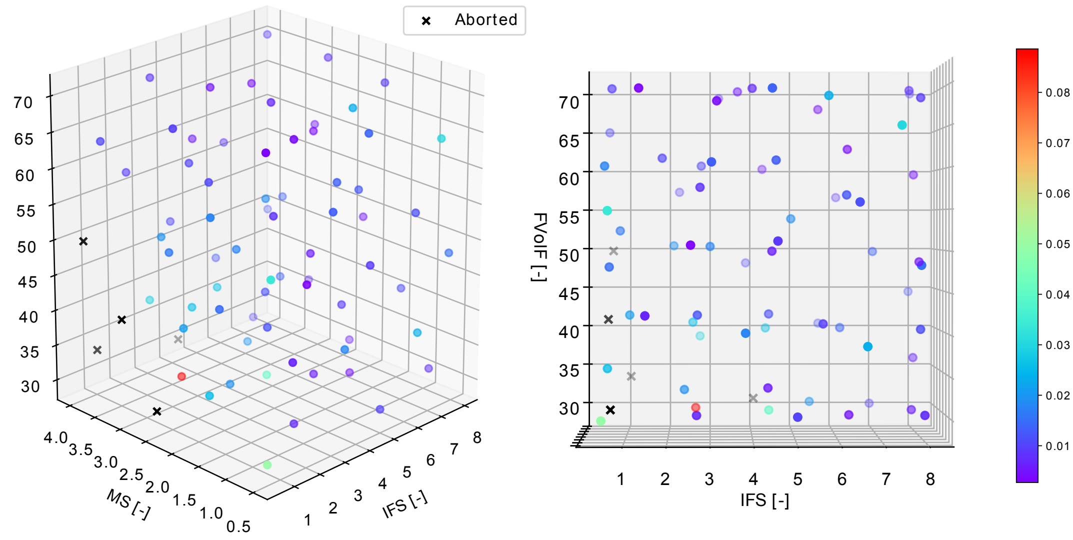


Figure 7.6: Equally spaced distribution of the 69 datapoints to approximate Equation 7.4. The 5 aborted datapoints are marked as "x", i.e. 10 unsuccessful attempts were made to obtain 5 valid results. The color of the points corresponds with the relative standard deviation, i.e. standard deviation of the five stress values calculated at this combination of input parameters, divided by their mean stress value.

points their color indicates the relative standard deviation of the stress result at this datapoint. It is calculated as the standard deviation divided by the mean value of the five transverse tensile strength values at this datapoint. As can be seen, the relative standard deviation is generally below 3%, and only once peaks at 8%. Four out of five aborted datapoints occured in the same region of low IFS and high MS, particularly at low FVolF<40%. In the ensuing discussion, results will be disregarded in this specific region due to a lack of datapoints for the NN approximation.

Regarding the total number of datapoints, since this was the first full scale study to be performed, no predefined stopping criterion was employed. The plot of the *Cross-Validation* score over the number of datapoints has already been presented and discussed in Section 6.5, Figure 6.5.

7.2.1.1. Primary Effect of Matrix Strength and Interface Strength

Figure 7.7 shows the final NN function for 69 datapoints, and with the FVolF fixed at 70%. As can be seen, the surface shows three different zones: A linear slope on the left hand side, a linear slope on the right hand side, and a transitional zone in the center. These three zones correspond to three different final failure modes of the simulated UD FRPC ply under transverse tension:

- Interface failure: For today's FRPCs, final failure under tension is dictated by the failure of the fiber-matrix interfaces. Figure 7.7 illustrates this readily through the gradient at point $MS = IFS = 1$: Only an increase in IFS can increase the overall tensile strength for this combination of MS and IFS.
- Matrix failure: For the theoretical case of very high IFS while $MS = 1$, final failure of the SVE becomes dominated by failure of the matrix material near the fiber-matrix interfaces. This is in correspondence with analytical calculations by Tirosh et al. [71]. They calculated the stress concentration around a perfectly-bonded ciruclar fiber embedded in a matrix material, and found them at values between 1.53 and 1.74, at a distance of $1.2 \cdot r$, with r denoting the fiber radius, for resins with Poisson's ratio of 0.25 to 0.5. When taking into account several fibers, the maximum tensile stress concentrations were calculated at values around 2 for $FVolF = 50\%$, and to occur midway between fibers.

  An example point of the matrix failure regime in Figure 7.7 would be $(MS, IFS) = (1, 8)$. As can be seen, only an increase in matrix strength may increase the overall tensile strength at this combination.
- Mixed failure: A third case is defined by the transition between these two failure modes. In Figure 7.7, this corresponds to the center area of changing gradients, between the two asymptotic slopes on the left and right side. For these combinations of MS and IFS, the overall tensile strength of the material can be increased by increasing either one of the two constituent strengths.

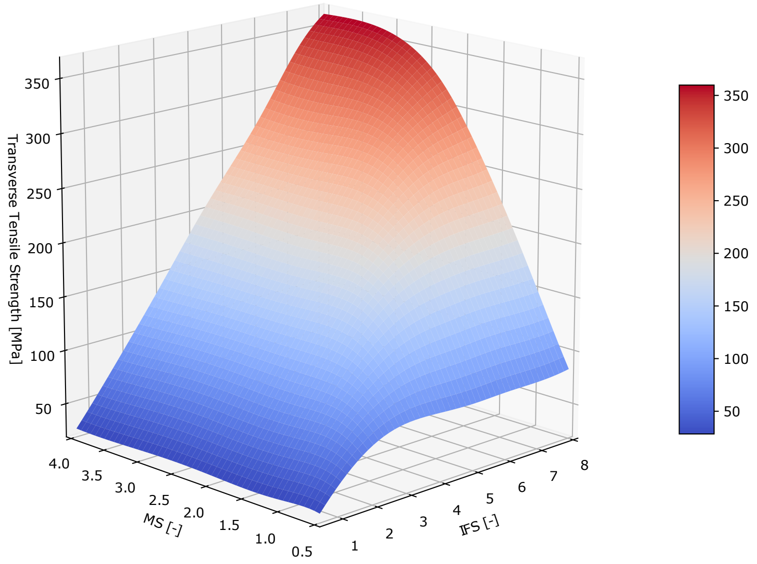


Figure 7.7: NN approximation of Equation 7.3 when trained with 69 datapoints. The input variable FVolF is fixed at 70%. The output variable transverse tensile strength is plotted over the two input variables matrix strength MS and fiber-matrix interface strength IFS.

A gradient plot provides an alternative way to easily distinguish the different failure zones. Since the trained NN is an analytical function, albeit very complex when written out, a gradient map can be readily calculated. Figure 7.8 shows the gradient plot of Equation 7.3 for a FVolF fixed at 70%. First of all, it illustrates nicely the separation between the different failure zones. Second, the arrows indicate, for each combination of constituent strengths, the direction which results in the highest gain in transverse tensile strength per increase in constituent strength. For the nominal example of $MS = IFS = 1$, marked as a black 'x', the transverse tensile strength can only be increased by increasing the interface strength. At $MS = 1$, only for $IFS > 2$ does an increase of the matrix strength make sense. To add another experimentally measured datapoint, the red 'x' illustrates the same AS4/8552 material but after having been exposed to a wet environment until saturation and under elevated temperatures. Its degraded properties have been measured experimentally and in-situ as $IFS = 0.71$ and $MS = 0.86$ [49][60][59]. As can be seen, environmental degradation of both interface and matrix strength does not change the conclusion that the transverse *tensile* strength of today's FRPC materials can primarily only be increased by improved interface strength.

Another result of interest could be a plot which easily shows the necessary direction of research in order to increase the transverse strength of today's FRPC materials. The shown calculation of gradients assumes that an increase in matrix strength is as easily achievable as an increase in interface strength. In reality however, the latter is much more difficult. In order to incorporate that, the gradients could be weighted and so produce a more realistic indication for future research.

As outlined in the literature review Section 3.3, Figure 3.9, previous research has been investigating the effect of IFS on overall tensile strength [78]. Vaughan et al. however did only slightly vary the IFS, for a fixed MS. In comparison with Figure 7.7, his work corresponds to a single vertical slice at a fixed MS, near $MS = 1$, and for $0.35 < IFS < 2.14$ [1]. As can be seen, the research of Vaughan et al. was confined to the area of fiber-matrix interface failure. In contrast, Figure 7.7 allows to situate his work in a greater frame, and to show the global function governing the transverse tensile strength of a UD FRPC ply.

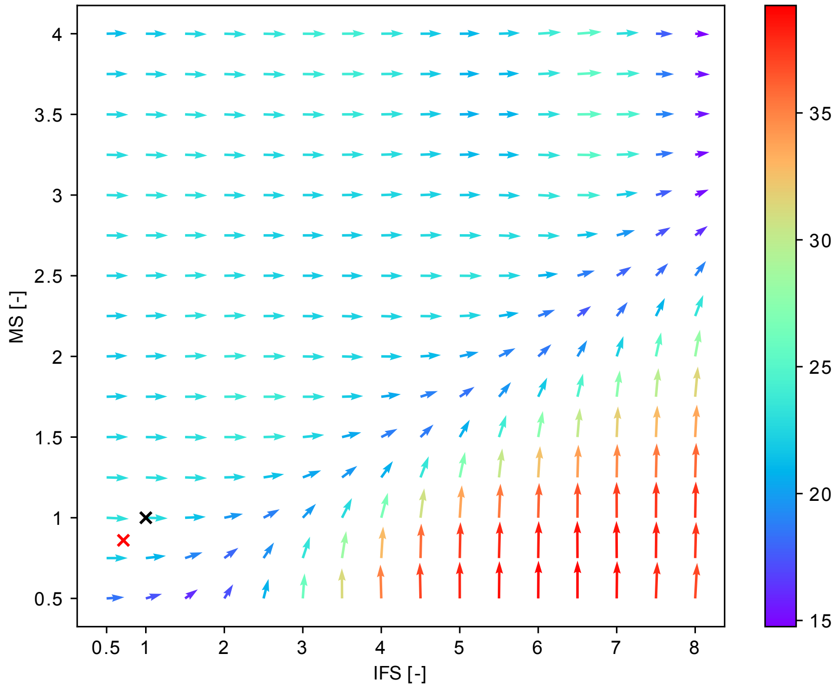


Figure 7.8: Plot of the gradients of the NN function shown in Figure 7.7. The coloring of the arrows is mapped to the modulus of each gradient vector, with the colorbar displayed in units of MPa. Marked with a black 'x' is the nominal AS4/8552 material with $IFS = MS = 1$, for a dry environment at room temperature. Marked with a red 'x' is the same material after exposure to a wet environment until saturation and under elevated temperatures, with degraded properties $IFS = 0.71$ and $MS = 0.86$, as measured experimentally in-situ by [49][60][59].

[1] Vaughan et al. assumed a fiber-matrix interface with the same value for normal and shear strength N, S. The model of this master's thesis assumes two different values, see Table 3.3. The comparative range of $0.35 < IFS < 2.14$ was calculated relative to the lower value of normal strength $N = 42\,\text{MPa}$ used in this thesis.

The plot of Figure 7.7 further showcases the capability of the proposed framework to produce a global analytical approximation based on a minimum amount of simulation. This can be put to use for the search of new materials. Requiring little a-priori knowledge about the unknown material property relationship, the developed framework can automatically explore the given parameter space until it has found a sufficiently accurate global approximation.

For the ensuing studies, the first idea was to just keep adding input parameters in order to showcase the full potential of the framework for larger parameter spaces. However, as Figure 7.7 showed, all possible combinations of matrix and interface strengths can be reduced to the three cases outlined above. Therefore, ensuing studies do not have to consider MS and IFS as varying input parameters, but can be performed for fixed values of these two variables. The most relevant case is of course $MS = IFS = 1$, i.e. the strength ratio of today's CFRPC materials, which we later explore in our studies on fiber-by-fiber hybridization.

Lastly, it has to be noted that Figure 7.7 shows only a single extracted value from each one of the 69 computed datapoints, i.e. the maximum transverse tensile stress endured. To give a better idea of the results generated, Figure 7.9 shows the stress-strain curves computed for a single slice of the grid of input parameters. For $FVolF = 70\%$ and $MS = 1$, the interface strengths were varied between 1 and 10 [2]. As discussed before, this slice of results is sufficient to illustrate all three failure zones. This can readily be seen in Figure 7.8, and shall now briefly be illustrated with the stress-strain curves of Figure 7.9:

- For $IFS = 1$, we are within the zone of interface failure. Interface failure triggers initial damage, and after a short non-linear part of the stress-strain curve due to matrix plasticity, the maximum tensile load is obtained. After the peak load, the load carrying capability drops rapidly and the simulation is stopped.

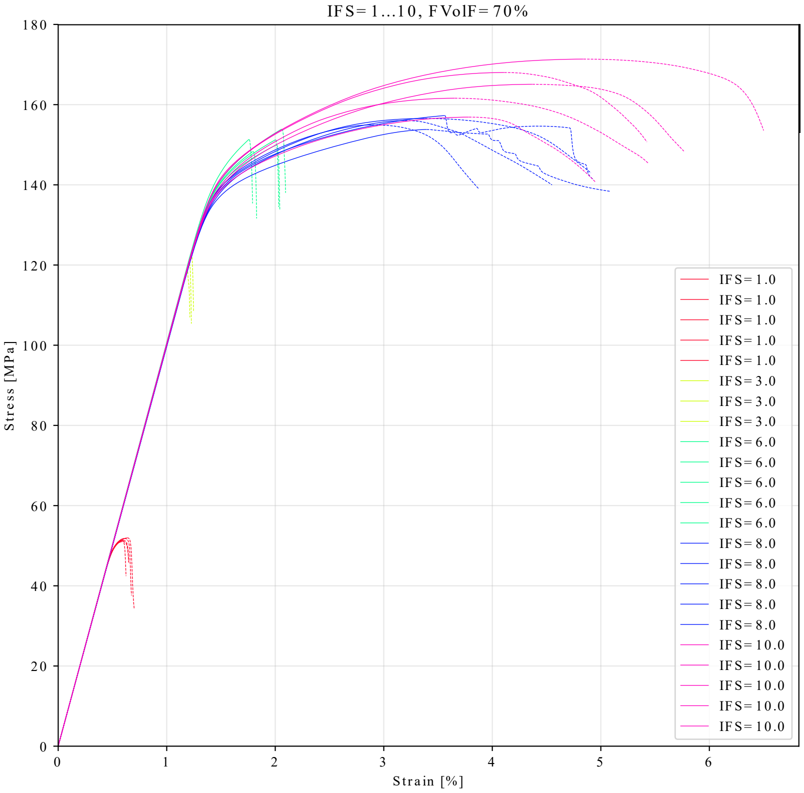


Figure 7.9: Plot of the stress-strain curves for 5 datapoints subject to transverse tension. Two input parameters were fixed, $FVoLF = 70\%$, $MS = 1$, while varying IFS between 1 and 10. Solid lines are used for the curves up to the maximum load peak, dashed liens until the cutoff criterion of 10% load drop. For high IFS values, the variance between the curves increases significantly, both in maximum stress and strain-to-failure.

[2] Additional simulations were performed to cover $IFS = 8...10$.

- For $IFS = 3$, we are within the transition zone. For this combination of constituent strengths, a higher transverse tensile strength of the SVEs may be obtained by increasing either one of the constituent strengths. Indeed, Figure 7.8 suggests that at this ratio, matrix failure and interface failure are equally contributing to the overall strength. The stress-strain curves for $IFS = 3$ illustrate this by sudden failure without any non-linear part due to matrix plasticity.

- For $IFS > 6$, we are within the zone of matrix failure. For this combination of constituent strengths, a higher transverse tensile strength of the SVEs may only be obtained by increasing the matrix strength. The stress-strain curves for $IFS = 6$ show a significant non-linear part before reaching the peak load, as resulting from the plastic deformation of the matrix between closely neighboring fibers. For $IFS > 6$, the maximum peak load does not increase much further, only the strain-to-failure and the variances between the curves. The latter indicates that for this extreme combination of constituent strengths, the simulations are increasingly experiencing numerical difficulties due to large deformations of the single elements between fibers. Results for these border cases thus have to be interpreted with caution.

7.2.1.2. Secondary Effect of Fiber Volume Fraction

It is evident that the primary parameters determining the transverse strength of any FRPC material will be the constituent strengths. Following that, geometrical effects are playing the next biggest role. Figure 7.10 shows the final trained NN approximating Equation 6.1, with overlaid plots for distinct values of the FVolF ($FVolF = 30, 40, 50, 60, 70\%$). As discussed earlier in this section, low FVolFs lead to numerical instabilities and outlier results. Figure 7.6 had already indicated that this still holds true for $FVolF \leq 40\%$ and Figure 7.10 now reconfirms this visually. It can be seen that for high matrix strengths combined with low interface strengths, the overall tensile strength results at low FVolF do not fit with the remainder of the data. Therefore, we will restrict the following discussion to the FVolF range between 40% and 70%.

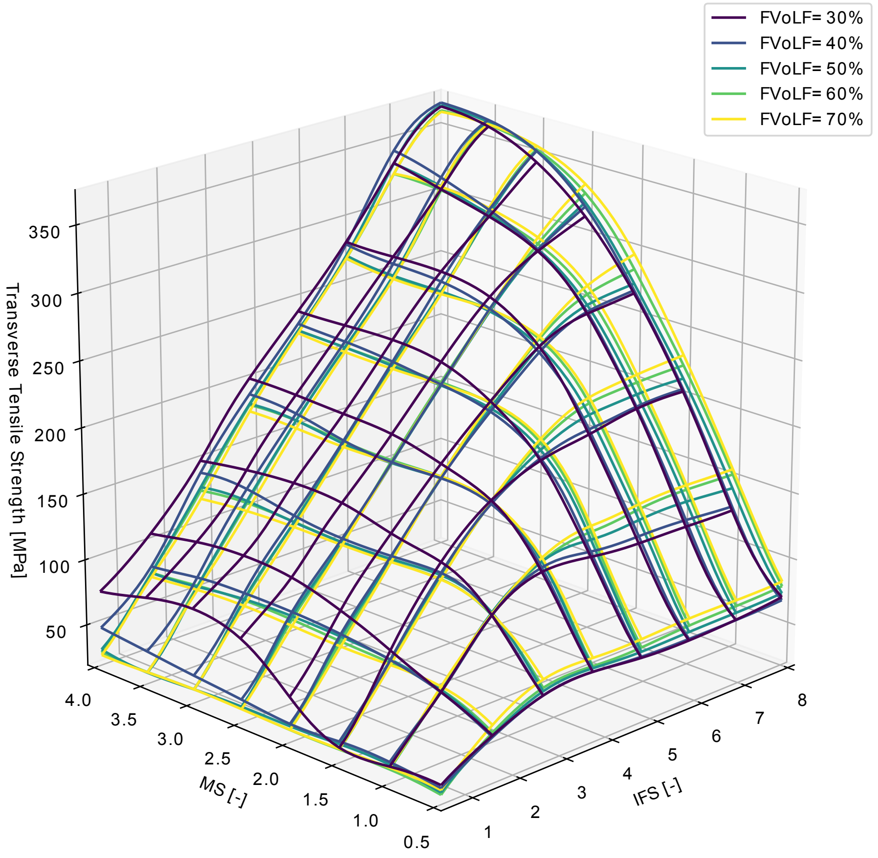


Figure 7.10: NN approximation of Equation 7.3 when trained with 69 datapoints. The input variable FVolF is varied between $FVolF = 30...70\%$. The output variable transverse tensile strength is plotted over the two input variables matrix strength MS and fiber-matrix interface strength IFS for each value of FVolF. As can be seen, the surface for $FVolF = 30\%$ does not align with the results for higher FVolFs in the region of high matrix strength and very low interface strength.

The overall effect of FVolF on the transverse tensile strength surface is best visualized through contour plots. Figure 7.11 shows the contour plots in direction of both MS and IFS, for $FVolF = 40\%$ and $FVolF = 70\%$. On the left, curves are plotted for fixed IFS, at values $IFS = 1, 2, 3, 4, 5, 6, 7$. On the right, curves are plotted for fixed MS, at values $MS = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5$. The following can be observed:

- For fixed IFS: As long as increasing the matrix strength yields a higher overall tensile strength, a higher FVolF *increases* also the transverse tensile strength.
- For fixed MS: As long as increasing the interface strength is the *only* means to increase overall tensile strength, i.e. before entering the transition area of combined failure, until then a higher FVolF *decreases* the transverse tensile strength.

Together, the same general trend can be summarized as: In the zones of matrix failure and mixed failure, higher FVolF results in higher transverse tensile strength. In the zone of interface failure, higher FVolF results in lower transverse tensile strength. For the two edge cases, failure zones of weak interfaces or weak matrix, it seems obvious that *reducing* the volume fraction of the *weaker* constituent increases the transverse tensile strength. For the transition zone of mixed failure however, it is surprising that the FVolF effect in this area is entirely dominated by the same trend than shown in the zone of matrix failure. It is neither the case, that the FVolF effect is reversed at some medium point within the transition zone, nor at the edge between the transition zone and the matrix dominated failure zone.

To judge the *magnitude* of the overall effect of FVolF on transverse tensile strength, Figure 7.12 lastly shows the difference between the two surfaces for 70% and 40% FVolF, divided by the tensile strength obtained for 40% FVolF. Compared with the contour plot of Figure 7.11, the plot has been rotated around the z-axis for better readability. Marked in black is the contour line which separates the region of tensile strength *increase* through *higher* FVolF, i.e. positive values, from the opposite region.

First, it has to be noted that for extreme values of either high IFS and low MS, or the opposite case, the simulated results start to diverge significantly. This is due to numerical issues for these extreme combinations of constituent strengths, as has been illustrated before in Figure 7.5, and will be reiterated for this specific case in Figure 7.14. Ignoring these peak values, it can be seen that varying the FVolF between 70% and 40% can change the transverse tensile strength by +20% and -15%. Today's FRPC materials, i.e. $MS = IFS = 1$, lie in the region of interface failure and thus tensile strength decrease for higher FVolF. As can be seen in Figure 7.12 however, the potential to reverse this effect exists, but only for higher values of IFS. If such would be achieved however, the magnitude of +20% possible gain in transverse strength is significant.

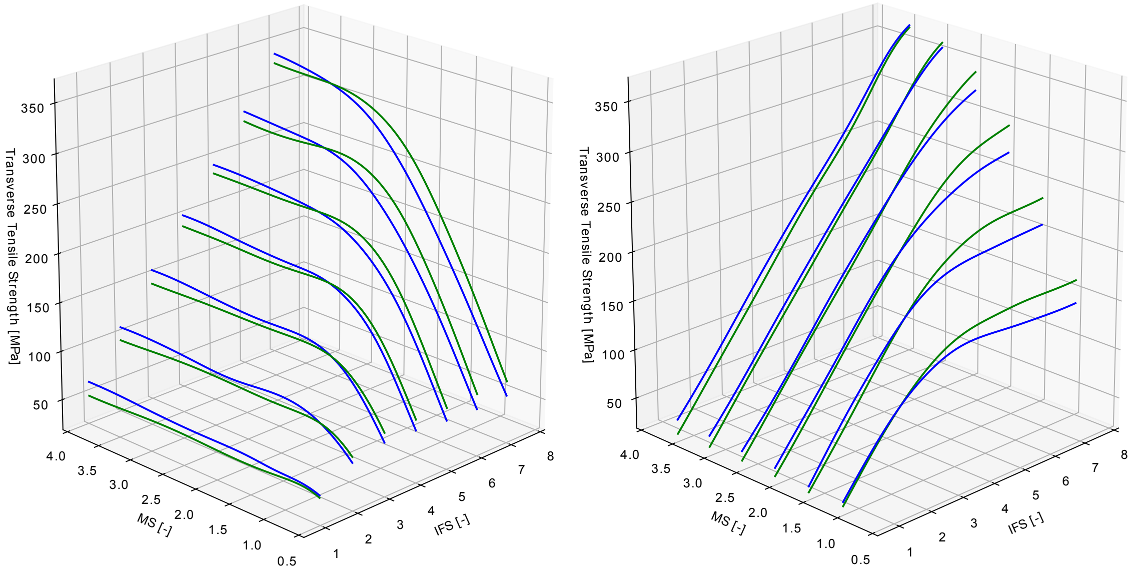


Figure 7.11: Contour plots of the NN function of Figure 7.10, in direction of both MS and IFS. The FVolF is fixed at 40% and 70% . On the left, the curves of transverse tensile strength are plotted for fixed IFS, at values $IFS = 1, 2, 3, 4, 5, 6, 7$. On the right, curves are plotted for fixed MS, at values $MS = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5$.

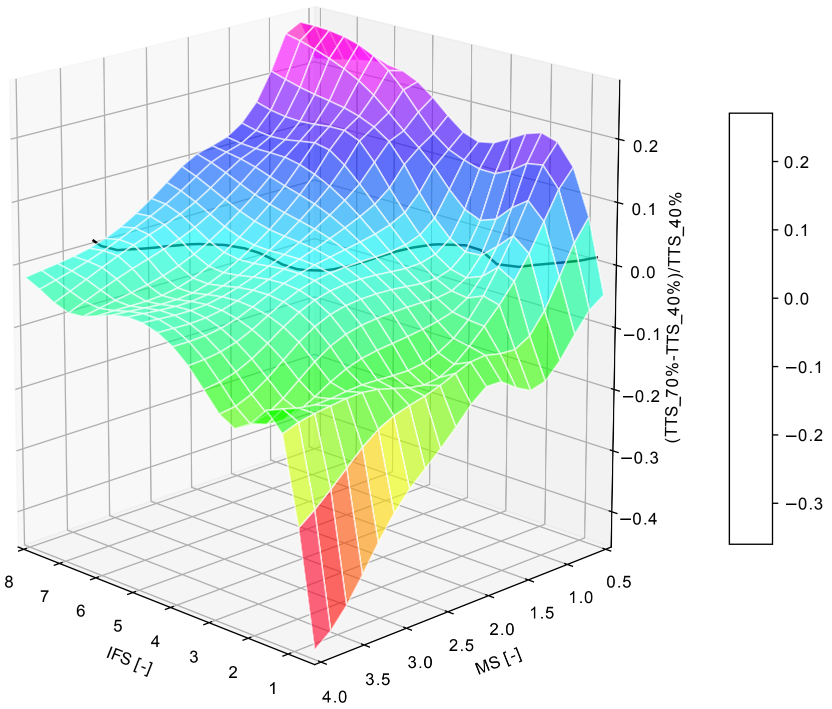


Figure 7.12: Plot of the magnitude of the effect of FVolF on transverse tensile strength (TTS). On x- and y-axis, the matrix strength MS and interface strength IFS are drawn. On the z-axis, the difference between the surfaces from Figure 7.10 for $FVolF = 70\%$ and $FVolF = 40\%$ is plotted, normalized by the transverse tensile strength values for $FVolF = 40\%$. The contour line splitting positive from negative z-values is drawn in black. As can be seen, for extreme combinations of high IFS and low MS, or the opposite case, the numerical results start to diverge from the rest of the results.

On a last note on Figure 7.12, the magnitude of the +20% and -15% FVolF effect on transverse tensile strength is much smaller compared to the primary effect of IFS and MS, as shown in Figure 7.7. The smoothness of Figure 7.12 could be improved, but this would require to simulate more datapoints for the NN training, or to manually force a NN approximation function of lower order, for example by increasing **alpha**. For the present discussion on the secondary effect of FVolF, the results were considered accurate enough.

Putting these results into perspective, today's CFRPC materials lie well within the interface failure zone, and thus a higher FVolF decreases transverse tensile strength. Figure 7.13 shows the stress-strain curves up to maximum stress for 24 SVE FEAs. The constituent strengths are kept at their nominal values, i.e. $MS = IFS = 1$, while varying the FVolF between 30% and 70%. Note that the 30% results at this combination of MS and IFS could still be considered valid, as to be judged from Figure 7.10 and Figure 7.4. Figure 7.13 now shows said effect of higher FVolF decreasing transverse tensile strength in the interface failure zone, but in more detail. The changing stiffness due to varying FVolF can be observed as well as the decrease of the strain-to-failure at maximum stress level. Lastly, this overall trend is in line with analytical equations developed to describe transverse tensile strength as a function of FVolF [34].

Figure 7.14 now shows the same simulations repeated for the matrix failure zone, i.e. $MS = 1$, $IFS = 10$ [3]. Cross-referencing with Figure 7.10, the results for $FVolF = 30\%$ are judged to be lower than they should, compared with the remaining results for $FVolF > 40\%$. It also has to be noted that the variation in both maximum stress and strain-to-failure increases significantly for higher FVolF. While it is acknowledged that this may influence the *extent* of the observed FVolF effect, it is not considered to change the overall direction of the effect. If all simulations were to be stopped right after damage initiation, e.g. a 10% deviation from their initial stiffness, then we could still observe the same effect: higher FVolFs *increase* transverse tensile strength in the matrix failure zone.

In conclusion, if future research may succeed in increasing the IFS to such an extent that matrix failure becomes a relevant damage mechanism for transverse tensile failure, only then will the observed negative effect of FVolF on transverse tensile strength be reversed.

[3] Additional simulations were performed to cover $IFS = 8...10$.

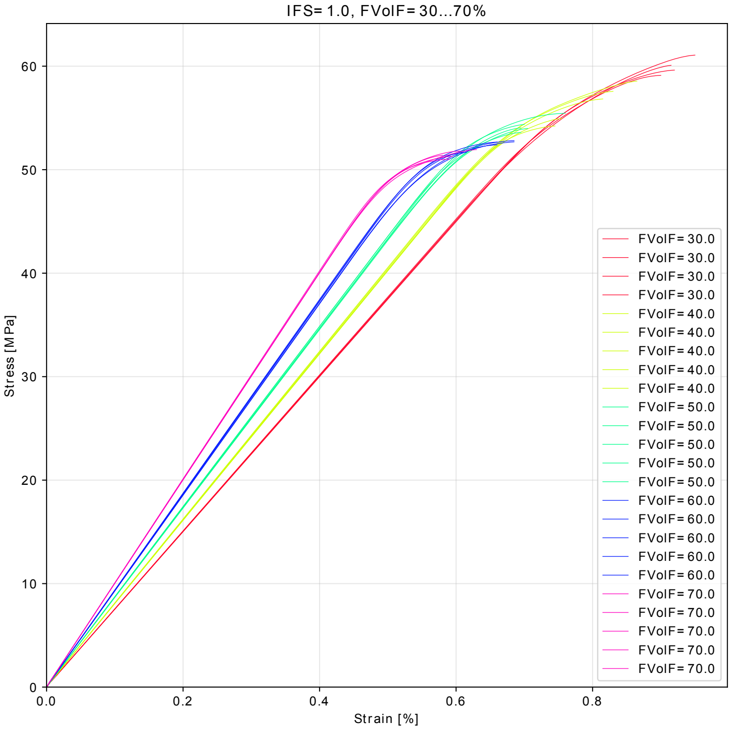


Figure 7.13: Plot of the stress-strain curves for 5 datapoints subject to transverse tension, and failing by interface failure. Two input parameters were fixed, $MS = IFS = 1$, while varying the FVolF between 30% and 70%. The stress-strain curves are plotted up to the maximum load peak, and variance between the curves belonging to the same datapoint can be noted as very low.

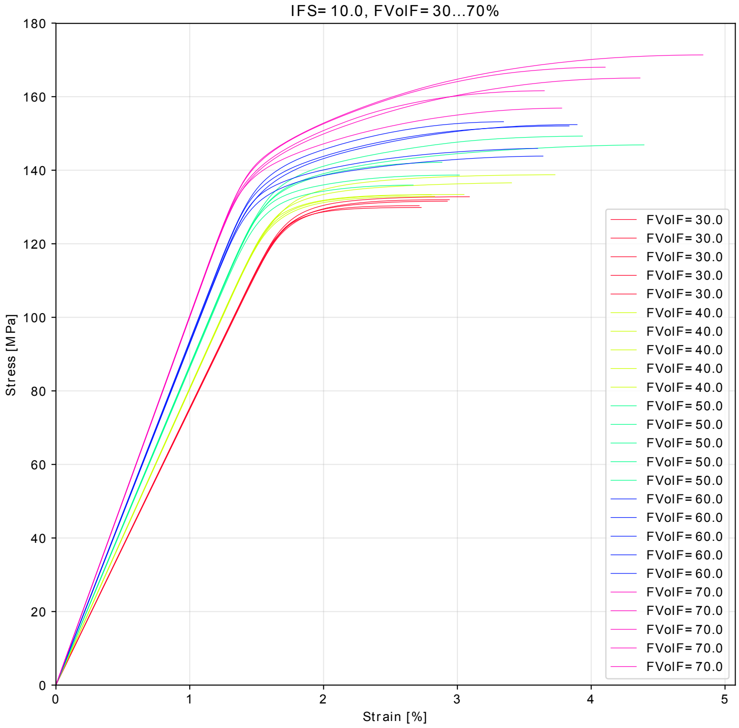


Figure 7.14: Plot of the stress-strain curves for 5 datapoints subject to transverse tension, and failing by matrix failure. Two input parameters were fixed, $MS = 1$, $IFS = 10$, while varying the FVolF between 30% and 70%. The stress-strain curves are plotted up to the maximum load peak, and the variance between the curves of the same datapoint is generally higher compared with the $MS = IFS = 1$ case of Figure 7.13. For increasing FVolF, the variance in maximum stress and strain-to-failure increases further.

7.2.1.3. Summary

Hereby, we conclude the presentation of the results for the study of transverse *tensile* strength, as a function of the constituent strengths and the Fiber Volume Fraction. The developed framework was used to determine the surrogate model which describes this relationship, in an automated way and based on 69 simulated datapoints.

The surrogate model shows that the primary influences on transverse tensile strength are the constituent strengths. The ratio of matrix strength to fiber-matrix interface strength was found to define the overall failure mode of the ply: either by interface failure, matrix failure, or a mixed mode. Today's CFRPC materials fall well within the region of dominant interface failure. Accordingly, their transverse tensile strength can only be increased by improving the strength of the fiber-matrix interfaces. For doubled IFS, the material would enter a region where increases in matrix strength also start to show a positive effect.

Furthermore, the effect of Fiber Volume Fraction was shown to be of secondary order. Depending on the values for the constituent strengths, an increase in FVolF can change transverse tensile strength by +20% or -15%. In general, increasing the volume fraction of the weaker constituent decreases the strength. Accordingly, today's CFRPC materials fall well within the region where higher FVolF leads to a strength decrease. For doubled IFS, this effect could be reversed.

After having covered the tensile loading case, we shall now present the results for the compression case.

7.2.2. Compression

With regard to the preceding discussion on numerical issues for certain combinations of input parameters, the following ranges were chosen:

- FVolF: 30% to 70%
- IFS: 0.75 to 5
- MS: 0.5 to 5

The final approximation function to be found thus becomes

$$\sigma_{transverse,compression} = \widetilde{f}(MS, IFS, FVolF), \tag{7.4}$$

A total of 115 datapoints were simulated, out of which 12 aborted. Of these 12 aborted datapoints, all failed due to local maxima before the global maximum. Figure 7.15 shows their spatial distribution. As before, the

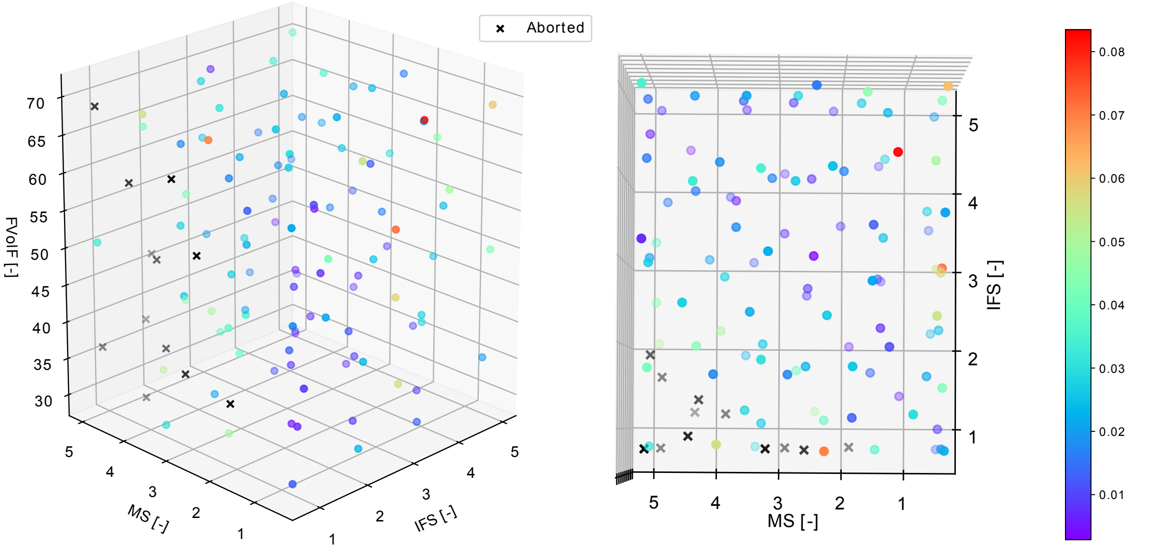


Figure 7.15: Equally spaced distribution of the 115 datapoints to approximate Equation 7.4. The 12 aborted datapoints are marked as "x", i.e. 10 unsuccessful attempts were made to obtain 5 good results. The color of the points corresponds with the relative standard deviation, i.e. standard deviation of the five stress values calculated at this combination of input parameters, divided by their mean.

aborted datapoints are marked as "x", for all other datapoints their color indicates the relative standard variation of the stress results at this datapoint. As can be seen, the deviation is generally below 5%, with higher values up to a maximum of 8% only obtained for extreme combinations of MS and IFS. This illustrates how this deviation can readily serve as a measure of the quality of the obtained results, without having to dive into the deformation plots of the simulations. It shall also be noted *which* combinations of MS and IFS lead to abortion, as readily illustrated on the right side of Figure 7.15. Numerical difficulties are only encountered for low IFS values, in combination with increasingly high MS values. The correlation between numerical difficulties and low FVolF is less significant than in the tensile case, see Figure 7.6.

Figure 7.16 shows the *Cross-Validation* score plotted over the number of datapoints. No pre-defined stopping criterion was employed, but this study was used to investigate the convergence behavior for many datapoints. As can be seen, a final minimum value of 5% average mean relative error can be achieved when training the NN with more than 80 datapoints. In the tensile case, see Figure 6.5, convergence occurred much faster, already starting from 30 datapoints on, but the eventually obtained minimum remained at a value of 10%.

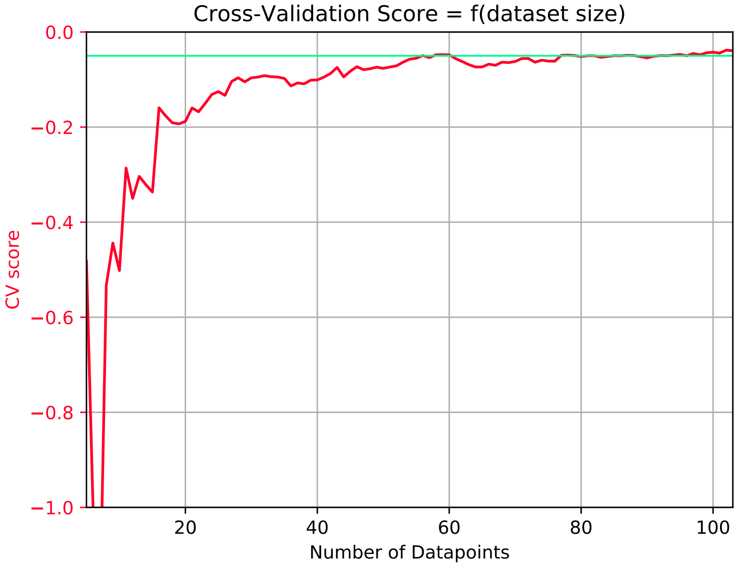


Figure 7.16: Cross-Validation score, i.e. average mean relative error, plotted over the number of datapoints used for NN fitting of Equation 7.4. The vertical green line is drawn at a mean relative error of 5%.

7.2.2.1. Primary Effect of Matrix Strength and Interface Strength

Figure 7.17 shows the final NN approximation of Equation 7.4 for 115 (103) datapoints, plotted with the FVolF fixed at 70%. The surface resembles the plot of the tension case, see Figure 7.7 and Figure 7.8, in that it can be divided in the same three zones: interface-dominated failure, matrix-dominated failure, and mixed failure. However, compared to the tension case, the surface is now shifted: Whereas for the tension case, the nominal parameter combination $MS = IFS = 1$ lies well within the interface-dominated failure region, in the compression case it lies in the *mixed* failure region, see gradient plot in Figure 7.17. This is in line with findings from literature [27]. For today's FRPC materials, transverse tensile failure of a UD ply is dominated by interface failure. In the compression case however, matrix and interface failure are both contributing. The gradient plot in Figure 7.17 can now quantify this: At point $MS = IFS = 1$ both failure mechanisms are about equally contributing to the transverse compressive strength of the ply. A few general comments shall be made:

- A first observation is that the surface of Figure 7.17 is less smooth than the in the tension case in two specific regions:
  - The region of low IFS - high MS is not in line with the remainder of results: Consulting Figure 7.15 this is exactly the region where most simulations abort due to numerical difficulties, and thus where valid interpolation points are scarce. For $FVolF = 70\%$, the datapoint in this region did

abort see Figure 7.15 and thus the NN has to approximate the transverse compressive strength in this region based on interpolation points which are much further away. Since the region of low IFS - high MS generally did not produce many valid datapoints, the NN approximations in this region should be disregarded.

- The surface in the opposite region of high IFS - low MS shows some "waviness" compared to the nearby regions: Consulting Figure 7.15 again, this is a less critical region compared to the opposite case, i.e. the simulations do not abort. However, it has been noted in Figure 7.15 that the relative standard deviation of the stress results increases in this region, indicating increasing numerical difficulties and in consequence less accurate results.

- A second observation is the generally less regular shape of the surface compared to the tensile case. This can be attributed to the different failure behavior of the SVEs under compression than under tension. In the tensile case we saw that the SVEs may either fail through

  - linked interface-failures, which are governed by the rather simple material model of cohesive elements (no compressive behavior), or
  - matrix failure in the single elements between closely neighboring fibers, thus very localized, or
  - mixed failure, combining both failure modes.

  In the compressive case now, failure nominally constitutes of one or several shear bands within the matrix, see Figure 3.13, triggered by interface failures. A general difference to the tensile case is that many matrix elements are now involved in contributing to final failure. The used material model for the 8552 Epoxy matrix is much more complex than for the cohesive elements of the interfaces, see Section 3.1.2.2.2. Increase of strength under confining hydrostatic pressure is one of its features which is attributed to the more irregular shape of the surface of Figure 7.17.

- A third remark concerns fiber failure: As can be seen in Figure 7.17, the maximum transverse compressive strength computed is greater than 1000 MPa, which renders our initial assumption of neglecting fiber failure invalid. Keeping this in mind when interpreting the results, this does not render the parametric study invalid for other combinations of input parameters, or assuming that fiber strength could be increased as well. After all, this study aims at characterizing the general theoretical influence of MS and IFS on transverse compressive strength.

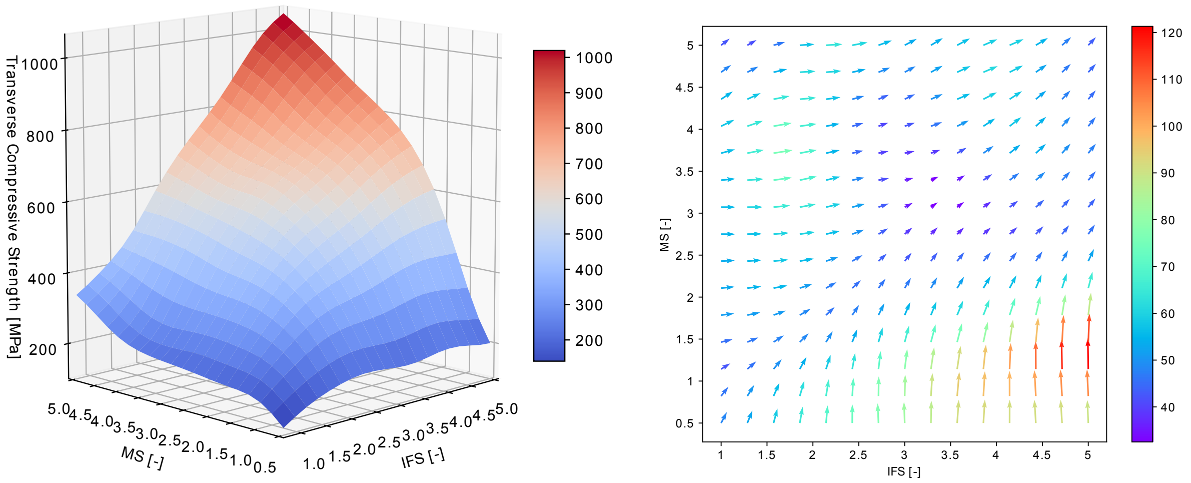


Figure 7.17: Left: NN approximation of Equation 7.4 when trained with 115 (103) datapoints. The input variable FVolF is fixed at $FVolF = 70\%$. The output variable transverse compressive strength is plotted over the two input variables matrix strength MS and fiber-matrix interface strength IFS. Right: Plot of the gradients of the NN function shown on the left. The coloring of the arrows is mapped to the modulus of each gradient vector, with the colorbar displayed in units of MPa.

7.2.2.2. Secondary Effect of Fiber Volume Fraction

Figure 7.18 shows the final trained NN approximating Equation 7.4, with overlaid plots for distinct values of the FVolF ($FVolF = 30, 40, 50, 60, 70\%$). Compared to the tensile case of Figure 7.11, the effect of the FVolF is identical in both its sign and regions:

- In the region of matrix-dominated failure, i.e. high IFS and low MS, an increase in FVolF leads to an increase in transverse compressive strength.
- In the region of interface-dominated failure, i.e. low IFS and high MS, an increase in FVolF leads to a decrease in transverse compressive strength.

What is different however is the *magnitude* of the effect of FVolF, particularly in the region of interface-dominated failure. As discussed before, Figure 7.18 has to be interpreted with care in this region of low IFS combined with high MS, since very few valid datapoints are available for the NN approximation see Figure 7.15 on the right. Still, the effect of FVolF on transverse compressive strength is well pronounced, even if only judging for values of $IFS > 2$.

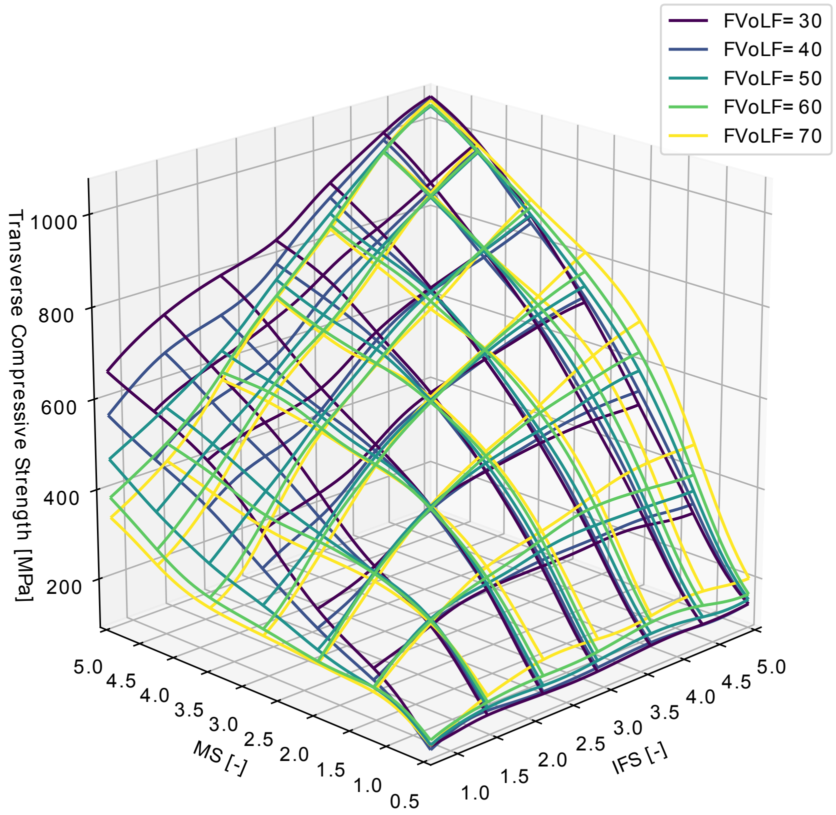


Figure 7.18: NN approximation of Equation 7.4 when trained with all 103 datapoints. The input variable FVolF is varied between $FVolF = 30...70\%$. The output variable transverse tensile strength is plotted over the two input variables matrix strength MS and fiber-matrix interface strength IFS for each value of FVolF.

7.2.2.3. Summary

Hereby, we conclude the presentation of the results for the study of transverse *compressive* strength, as a function of the constituent strengths and the Fiber Volume Fraction. The developed framework was used to determine the surrogate model which describes this relationship, in an automated way and based on 103 datapoints.

The surrogate model shows that the primary influences on transverse compressive strength are again the constituent strengths. As in the tensile case, the ratio of matrix strength to fiber-matrix interface strength was found to define the overall failure mode of the ply: dominated by interface failure, matrix failure, or a mixed both. Contrary to the *tensile* case, today's CFRPC materials fall within the mixed mode region. In line with experimental findings, they fail under a combination of interface failure and shear band formation within the matrix, the latter being illustrated in Figure 3.13. The surrogate model can now quantify their contributions, suggesting that matrix and interface contribute equally to the transverse compressive strength. Accordingly, the latter can be increased by improving the strength of either one of the constituents.

Furthermore, the effect of FVolF was shown to be of secondary order. Depending on the combination of constituent strengths, an increase in FVolF can change the transverse compressive strength of the ply by ±30%. In general, as in the tensile case, increasing the volume fraction of the weaker constituent decreases the strength. Today's CFRPC materials fall within the border of the region where increased FVolF still leads to a strength decrease. For a slightly increased IFS, the surrogate model suggests that this effect could be reversed. For completeness, it has to be remarked that the used CM model does not include voids, which in practice can have a significant influence on experimental results.

After having covered both the tensile and compressive loading case, we can summarize our findings of this first study.

7.2.3. Summary

Hereby, we conclude the presentation of results for the study of transverse strength, under both tension and compression, as a function of the constituent strengths and the Fiber Volume Fraction. The developed framework was used to determine the corresponding surrogate models which describe this relationship, in an automated way, and based on a non-optimized amount of 69 and 103 datapoints respectively.

The surrogate models suggest that for both tensile and compressive loading, the primary variables which determine transverse strength are the constituent strengths. The ratio between the two defines the dominant failure mode, either interface or matrix failure, or a combined mode. The transverse failure of today's CFRPC materials is interface dominated in the tensile case, and of combined mode in the transverse strength. The surrogate model now quantifies these contributions and can readily suggest the most effective ways to increase this strength in the near future. Improving the tensile strength is only possible by increasing the Interface Strength. Improving the compressive strength can be equally achieved by increasing IFS of MS. To reverse the strength-decreasing effect of high FVolFs, the only way to do so is by improving the IFS.

After having characterized these primary variables defining the transverse strength of a UD FRPC ply, we now turn to investigate the unknown effect of fiber-by-fiber hybridization. We are most interested in the potential of this effect on today's CFRPC materials, therefore we fix the constituent strengths at their nominal values, and maintain the FVolF as an input parameter. The following two studies now explore the effect of mixing two different fiber cross-sectional shapes (circular and 4-lobed), and the effect of mixing two different fiber diameters.

7.3. Transverse Strength as a Function of Fiber Shape and FVolF

In order to characterize the effect of fiber-by-fiber hybridization, in this case mixing fibers of different cross-sectional shapes, the following two input parameters are chosen:

- Geometrical Parameters:
  - FVolF: Total fiber volume fraction
  - RFF: Round Fiber Fraction, i.e. the fraction of the FVolF with circular cross-section
- Material Parameters: none

No material parameters were chosen, the constituent strengths were instead fixed at their nominal values $MS = IFS = 1$ in order to keep the study applicable to today's CFRPC materials. The fiber shape to mix with round fibers was eventually chosen as 4-lobed. This was based on the results from M. Herraez [30], discussed in Section 3.1.3 and summarized in Figure 3.8.

7.3.1. Choice of Non-circular Fiber Shape

Since we are most interested in *increasing* the transverse tensile strength, the only non-circular fiber shape found by M. Herraez to achieve that under both tensile and compressive loading is 2-lobed, aligned. His simulations were performed at a fixed FVolF of 50%. In our study however we want to vary the FVolF over a wide range, and up to 70% as is usually aimed for in industry applications of composite materials. When generating SVEs with high FVolFs and *aligned* non-circular fibers, this latter condition is however hard to fulfill. The algorithms of *Vip3r* usually succeed in generating a valid SVE, but the alignment condition is violated for an increasing amount of fibers when going up to 70% FVolF. Figure 3.8 further illustrates that 2-lobed fibers only increase tensile strength *if* properly aligned. In the unaligned case, the effect reverses and tensile strength even *decreases* compared to the case of circular fibers.

In conclusion, to avoid a distortion of results by poorly controlled fiber alignments, it was chosen to use unaligned 4-lobed fibers instead of the aligned 2-lobed ones. The former was found as the best non-aligned fiber shape to increase transverse strength under compression, and ranks second-"best" among all non-aligned non-circular fiber shapes under tension, although they all decrease transverse tensile strength slightly, see Figure 3.8.

7.3.2. Controlling Minimum Inter-fiber Distance

During a first round of simulations, it was further observed that by introducing non-circular fiber-cross-sections, the fiber placement algorithms of *Vip3r* are unable to maintain the user-defined minimum inter-fiber distance *Dmin*, see Section 4.2.1. When specifying the nominal value of $Dmin = 0.35\,\mu\text{m}$, *Vip3r* could only generate SVEs with a minimum inter-fiber distance of about 0.2 µm. As has been highlighted in Section 7.1.1, the minimum inter-fiber distance has a significant effect on transverse strength, therefore its value has to be constant among all SVEs which are to be compared within a study. To enforce this for this study, the *RVE.py* file had to adapted accordingly, eventually achieving a constant range of $0.2\,\mu\text{m} < Dmin < 0.22\,\mu\text{m}$. Figure 7.19 shows three example SVEs generated for the purposes of this study. From left to right, the fraction of round fibers is decreased from 1 to 0.5 to 0, while the FVolF stays fixed at 70%. Comparing with Figure 4.2 at $FVolF = 30\%$, the much denser packing has to be noted, while both SVEs maintain the same *Dmin* of about 0.2 µm.

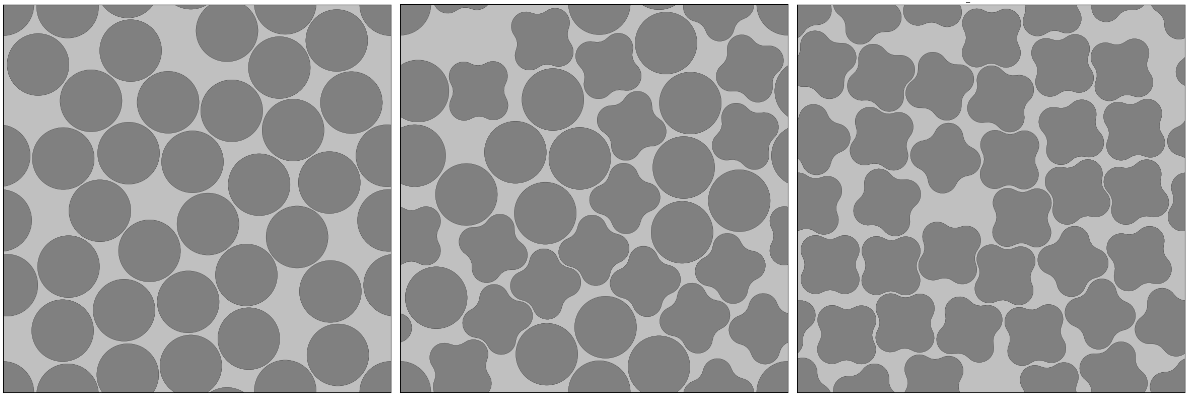


Figure 7.19: Three SVEs generated through *Vip3r* [30], with round fiber fraction decreasing from 1 to 0.5 to 0 from left to right. Fiber volume fraction is fixed at 70%, minimum inter fiber distance at 0.2 µm.

7.3.3. Tension

The following ranges were chosen for the two input parameters:

- FVolF: 30% to 70%
- RFF: 0 to 1, with 0 denoting only round fibers, and 1 only 4-lobed fibers

Regarding the FVolF range, the validity of SVEs generated with only 30% FVolF remains questionable, as discussed in Section 7.2. To remind the reader of the issue of the cornering fibers, Figure 4.2 shows six example SVE generations for $RFF = 0.5$ and $FVolF = 30\%$. The fiber placement algorithms of *Vip3r* always start with positioning one circular fibers in each corner, thereby distorting the *random* distribution of fibers within the SVEa for low FVolFs. Keeping this in mind for the interpretation of the final results, the approximation function to be found becomes

$$\sigma_{transverse,tension} = \widetilde{f}(RFF, FVolF). \tag{7.5}$$

A total of 25 datapoints were simulated, and none of them aborted. To determine the number of datapoints, no pre-defined stopping criterion was employed. Figure 7.20 shows the spatial distribution of the 25 datapoints, as well as the final NN approximation of Equation 7.5.

The first observation which has to be pointed out is the scale of the observed effects. As can be seen, the effect of fiber volume fraction on transverse tensile strength is about 12 MPa. In contrast, the effect of the round fiber fraction, i.e. fiber-by-fiber hybridization, is only about 2 MPa. Both are relatively small compared to the average transverse strength of about 50 MPa, particularly the effect of hybridization which we are primarily interested in. As a consequence, the relative standard deviation of the 17 datapoints of up to 5% suddenly becomes significant. This peak value of 5% is reached only for simulations with $FVolF \leq 40\%$, which is in line with the results of our sensitivity study on the effect of the SVE size for circular fibers, see Figure 7.4. Surprisingly though, in this region of low FVolF, only the *mixed* SVEs showed such a comparably high variation, while the SVEs of either only circular or only 4-lobed fibers stayed below 2.5%, just like all other datapoints.

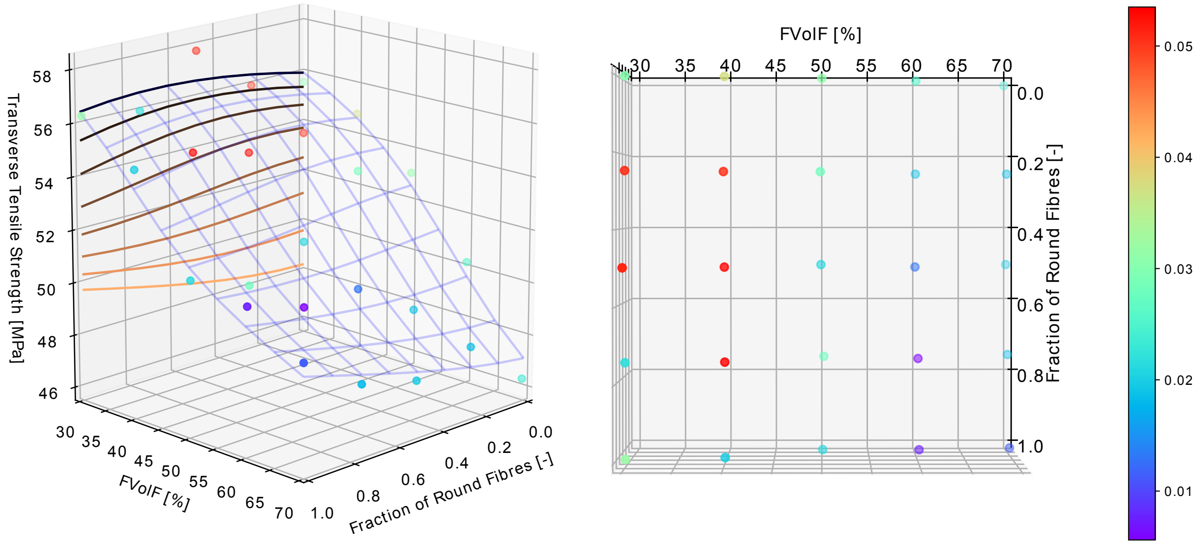


Figure 7.20: NN approximation of Equation 7.5 for a total of 25 datapoints. The output variable transverse tensile strength is plotted over the two input variables fiber volume fraction FVolF and round fiber fraction RFF. The color of the datapoints corresponds with the relative standard deviation, i.e. standard deviation of the five stress values calculated at this combination of input parameters, divided by their mean stress value.

A second important observation are the unexpected variations of the mean strength values of the datapoints, for example when fixing $FVolF = 50\%$ and increasing the fraction of 4-lobed fibers. Based on the results of M. Herraez in Figure 3.8, the transverse tensile strength is expected to decrease slightly but constantly. Instead, we can a) see an oscillating behavior between only-circular and only-4-lobed fibers, and b) notice that both only-circular and only-4-lobed fibers resulted in the *same* strength value of 52 MPa. A more detailed discussion of these surprising effects is due:

(a) The unexpected variations of trend a) are not specific to $FVolF = 50\%$, but seem to occur throughout the range of fiber volume fractions. This is believed to result from a new variable which remained unnoticed but has been introduced together with non-circular fibers into the SVE geometry: fiber orientation. For circular fibers, this variable does not exist, but one of the findings of M. Herraez [30] was that the alignment of non-circular fibers has a significant effect on the transverse strength. The unexpected strength oscillations observed in Figure 7.20 are believed to result from the distinct distribution of fiber orientations, which is different for every choice of the two geometrical parameters FVolF or RFF. Figure 7.19 illustrates this readily. On the right, for only 4-lobed fibers at $FVolF = 70\%$, most fibers are aligned like a horizontal square, whereas in the 50/50 mixed case with circular fibers in the center, the 4-lobed fibers are mostly oriented like a diamond. Implicitly, we assumed that the distribution function of the fiber orientations would resemble a standard deviation around a mean of 0, but we can readily observe that this is not the case.

(b) On the CM side, thisstudy uses exactly the same modelling approach than the one developed by M.

Herraez [30]. His results at $FVolF = 50\%$ are that the transverse tensile strength decreases from 59 MPa to 52 MPa when replacing all circular fibers with all 4-lobed fibers, see Figure 3.8. This is in direct contradiction with the findings of this study, which shows *no change* in transverse tensile strength between the two. Since the same tool *Vip3r* was used to generate the SVEs, it is assumed that the former study of M. Herraez did not notice that *Vip3r* unnoticeable starts to violate the user-specified minimum inter-fiber distance *Dmin* when changing from circular to 4-lobed fibers. Since the minimum inter-fiber distance has a strong effect on the transverse strength, see the sensitivity study presented in Section 7.1.1, this correlates well with the previously reported strength decrease from 59 MPa to 52 MPa.

Summarizing, in order to correctly compare circular with 4-lobed fibers, and to judge the effect of mixing on transverse tensile strength, *Dmin* has to be well controlled, as was the case for this study. However, the magnitude of the effect was determined as insignificantly small, always less than ±1 MPa. Compared to the original findings of M. Herraez [30] however, this is a positive result, since the reported strength *decrease* for 4-lobed fibers may now be revoked based on the found results.

For a more accurate evaluation of the small scale effect, it might be necessary to attempt to reduce the observed oscillations of the datapoints. For further studies, we therefore propose to increase the size of the SVEs, in order to obtain a smoother distribution function of the fiber orientations. Still, it is possible that at certain FVolFs, and for certain non-circular fiber-shapes, the fibers within the SVEs may favor a certain non-random distribution, see Figure 7.19 on the right. To test this hypothesis, it would be required to increase the number of SVEs computed per datapoints, e.g. from now 5 to 10. Further, the distribution functions of the fiber orientations, i.e. angle over occurrence, should be analyzed to find out whether they change according to a deterministic pattern, depending on fiber volume fraction and the mixing ratio RFF. Lastly, this could be repeated for the various non-circular fiber shapes proposed in [30], in order to complete the picture of the effect of fiber alignment.

For the practical purposes of transverse tensile strength increase through fiber hybridization, this effort is deemed over-proportional compared to the meager amount of strength gain which could be achieved. On the other hand, this finding has implications for any kind of simulations with SVEs and non-circular fiber shapes: The fiber alignment is an important parameter which has to be properly controlled and might exert a significant influence on your results.

7.3.4. Compression

The same ranges of the input parameters for the tension case were kept for the simulations of the compression case. The final approximation function to be found becomes

$$\sigma_{transverse,compression} = \tilde{f}(RFF, FVolF). \tag{7.6}$$

A total of 20 datapoints were simulated, and none of them aborted. To determine the number of datapoints, no pre-defined stopping criterion was employed. Figure 7.21 shows the spatial distribution of the 20 datapoints as well as the final NN approximation of Equation 7.6. Several observations have to be pointed out:

- First, the magnitude of the hybrid effect is now larger than the effect of fiber volume fraction. This is opposite to the tension results, where FVolF exerted a larger influence on transverse strength than the mixing of fiber shapes.

- Furthermore, replacing circular with 4-lobed fibers now consistently results in a strength *in*crease. This effect increases towards higher FVolFs, with the maximum increase of +27 MPa, i.e. +13% obtained for $FVolF = 70\%$. This boost of the hybrid effect for increasing FVolF could be attributed to the interlocking of the 4-lobed fibers. They are able to confine the inter-fiber matrix ligaments much more than circular fibers, and even more so for higher FVolFs. Since the compressive strength of the matrix material increases under pressure, the final failure of the SVEs through matrix shear bands may be delayed.

- Interestingly, when using primarily 4-lobed fibers, the usual negative effect of FVolF on strength is reversed. Today's CFRPC materials of circular fibers generally *de*crease in transverse compressive strength for *in*creasing FVolF. Figure 7.21 shows this for $RFF = 1$, which is consistent with our previous results shown in Figure 7.18 at point $MS = IFS = 1$. Figure 7.21 now also shows that for the case of $RFF = 0$, i.e. only 4-lobed fibers, an *in*crease in FVolF leads to an *in*crease in transvserse compressive strength.

- The results at $FVolF = 50\%$ are now in line with those of M. Herraez [30], suggesting that the influence of *Dmin* on transverse strength is less significant in the compression case than in the tension case.
- As in the tensile case, the effect of the fiber orientations of mixed SVEs persists. In comparison with the magnitude of the FVolF and hybridization effect which is much larger than in the tensile case, the effect of fiber orientation becomes *in*significantly small. At the same time, also the relative standard deviation of the datapoints is generally small, less than 3%, and low FVolFs do not to cause less accurate results in the compression case.

The methodological conclusions of the tensile case also apply to the compression case: To evaluate the hybrid effect of mixing circular with 4-lobed fibers under compression, the fiber orientation and minimum interfiber distance have to be well controlled. In contrast to the tension case however, the effect of both parameters is much less pronounced.

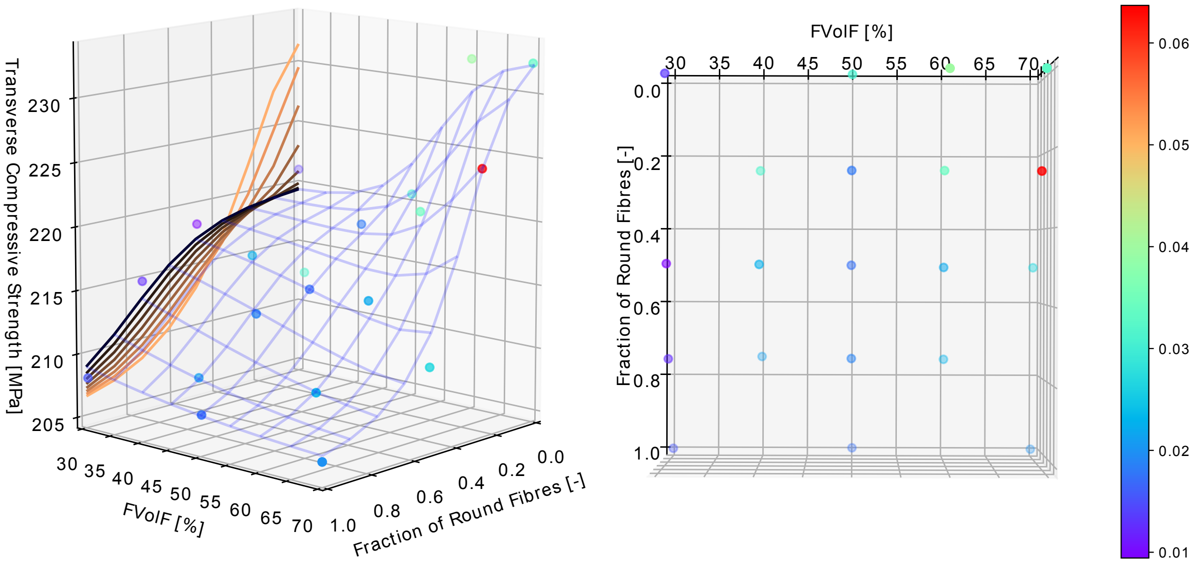


Figure 7.21: NN approximation of Equation 7.5 for a total of 20 datapoints. The output variable transverse compressive strength is plotted over the two input variables fiber volume fraction FVolF and round fiber fraction RFF. The color of the datapoints corresponds with the relative standard deviation, i.e. standard deviation of the five stress values calculated at this combination of input parameters, divided by their mean stress value.

7.3.5. Summary

Hereby, we conclude the presentation of results for the study of the transverse tensile/compressive strength of a UD CFRPC ply, as a function of the total fiber volume fraction and the fraction of circular versus 4-lobed fibers. The developed framework was used to determine the corresponding surrogate models which describe this relationship, in an automated way, and based on a non-optimized amount of 25 and 20 datapoints respectively.

The surrogate models suggest that transverse *tensile* strength does not change when replacing or mixing circular fibers with 4-lobed ones, independent of the value for the total fiber volume fraction. For transverse *compressive* loading however, the strength increases significantly the more circular fibers are replaced with 4-lobed ones. The higher the total fiber volume fraction, the stronger this increases in transverse compressive strength. Interestingly, 4-lobed fibers *reverse* the usual negative effect that high fiber volume fractions have on the compressive strength of a UD CFRPC ply of circular fibers. For 4-lobed fibers, *higher* FVolF leads to an *in*crease in compressive strength.

The generated results complete the findings reported in literature [30]. Instead of a fixed FVolF and only the edge cases of *only* circular or *only* 4-lobed fibers, the determined surrogate models now describe the global behavior of the effect of 4-lobed fiber shapes, alone or in mixture with circular fibers. A side result of this study is the awareness of an important parameter of SVEs with non-circular fiber cross-sections: the orientation of the cross-sections. As was reported in literature, if all fibers are aligned in the same way, this can have a significant impact on transverse strength [30]. The performed study now indicates that the distribution of fiber orientations may further depend on the FVolF, as well as the mixing ratio. The 4-lobed fibers tended to arrange and interlock in regular ways, resulting in unusually "oscillating" strength results. Further studies involving non-circular fiber shapes, especially thos which are highly anisotropic, therefore need to carefully consider and monitor the distribution of fiber orientations.

After having characterized the effect of fiber-by-fiber hybridization for the case of mixing two different fiber shapes, we now turn to the last study, exploring the effect of mixing two different fiber diameters.

7.4. Transverse Strength as a Function of Fiber Diameter and FVoLF

The previous study explored the mixing of different fiber *shapes*, and its effect on transverse strength. The question remains what the effect is of mixing fibers of different *diameters*. The literature review showed that experiments have been performed with mixing glass and carbon fibers of different diameters in a single ply [83]. Most results however focus on longitudinal tension, not the hybrid effect under *transverse* tension [**?**]. Further, most results were produced experimentally, and particularly for transverse tensile testing, significant scatter is reported which hinders to obtain a reproducible trend [58]. This makes it difficult to come to a consensus conclusion on the hybrid effect in general, but particularly in the transverse tension case.

Exactly at this point, the advantages of Computational Micromechanics shall be leveraged. This methodology allows to break down all contributing factors of the hybrid effect, and investigate their influences individually. For the example of mixing glass and carbon fibers, the final ply will have two different kinds of fiber materials and diameters, and two different kinds of interfaces between each fiber and the common matrix material. The effect of mixing different fiber materials, i.e. different Young's modulus and coefficients of expansion, has been briefly examined during the pre-studies of this master's thesis, but no influence on transverse strength was found. Therefore, as a first step, the ensuing study shall investigate the effect of mixing different fiber *diameters* only. Future work could then extend the model by attributing different interfaces to each fiber diameter.

For the present case of mixing circular fibers of *two* different cross-sectional diameters, the following input parameters and ranges are chosen:

- Geometrical Parameters:
  - FVolF: Total fiber volume fraction, from 30% to 50% [4]
  - RFF: 8 mm Fiber Fraction, i.e. the fraction of the FVolF with circular fibers of 8 mm diameter, varied between 0 to 1, with 0 denoting only fibers of 2 mm diameter
- Material Parameters: none

Just like in the preceding study, no material parameters were chosen and the constituent strengths were fixed at their nominal values of $IFS = MS = 1$. The question remains *which* fiber diameter to mix with our nominal fibers of 8 µm. During preliminary studies we observed the greatest increase in transverse strength for lower fiber diameters. Therefore we chose a diameter of 2 µm for the second set of fibers. Figure 7.22 shows three example SVEs generated for the purpose of this study. From left to right, the fraction of 8 µm fibers (8FF) is decreased from 1 to 0.5 to 0, while the FVolF stays fixed at 50%. At the same time, the size of the SVEs remains also fixed at 50 µm x 50 µm. What seems trivial at first, turns out to touch on the fundamental limitations of the SVE methodology, when trying to model size-dependent effects. The SVEs for $8FF = 1$ and $8FF = 0$, shown in Figure 7.22 on the left and right, shall be adduced to explain these limitations in the following section.

7.4.1. Modelling Size-Dependent Effects with Statistical Volume Elements

At this point we have to discuss a fundamental point of the used methodology of SVEs when simulating the behavior of a certain material. A seemingly similar yet different approach than what we use in our framework are the so-called *embedded cells*. Take for example the case of modelling a cross-ply laminate of three stacked layers. The outer layers are modeled with only a few elements and homogenized material properties, whereas the middle layer is modelled in high detail as a large cell containing the real fiber distribution, thus *embedded* within the larger model [31]. At first sight, the embedded cell may seem like a very large SVE, embedded with

[4] Higher FVolF than 50% are generally not possible when decreasing fiber diameter while maintaining the minimum inter-fiber distance at its nominal value.

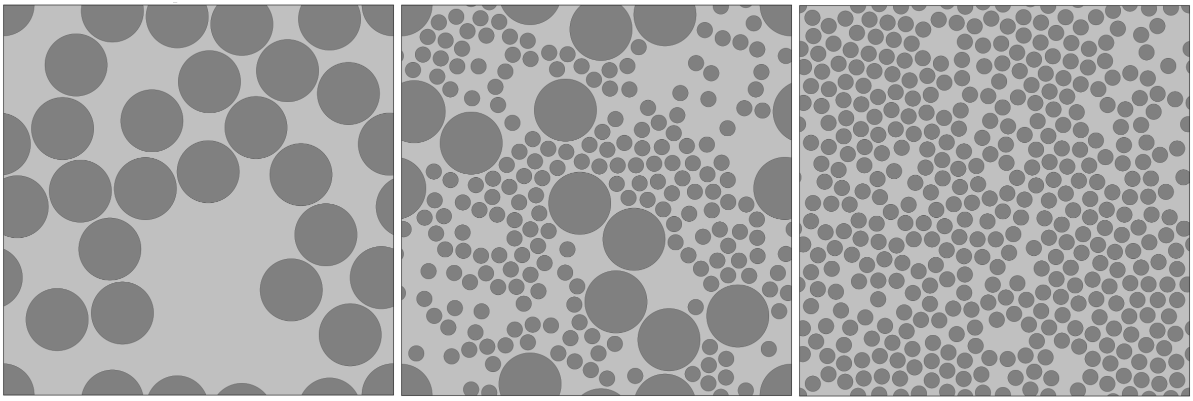


Figure 7.22: Three SVEs generated through *Vip3r* [30], with 8 mm fiber fraction decreasing from 1 to 0.5 to 0 from left to right. Fiber volume fraction is fixed at 50%, minimum inter fiber distance at 0.35 µm, and SVE size at 50 µm x 50 µm.

non-periodic boundary conditions into a larger model. But what distinguishes such *embedded cells* from the SVEs used by our CM model are their length scales and boundary conditions:

- For *embedded cells*, the user inputs the *real* values of the material properties for the model, i.e. fiber diameter, fracture energies, Young's moduli, and various failure strengths [5]. The boundary conditions for the embedded cell are chosen according to the surrounding elements. In the above example they are dictated by the adjacent plies, and according to the prescribed loading case.
- For SVEs, it is assumed that the SVEs are a *length-scale independent* representation of the material. The material is not discretized as a whole, but a volume just big enough is cut out and taken as representative of the whole material. For this to work out, the boundary conditions have to be *periodic*. This ensures that the deformations of the SVEs are periodic and that the SVEs could be duplicated and put together like a jig-saw puzzle, indeed representing a larger volume of the material. When looking at the fiber distribution of any of the SVEs throughout this thesis, there is no information about absolute values of the sizes, like the fiber diameter or the overall SVE size. This is the case because a fundamental characteristic of the SVE approach is that the *absolute* length values do not influence the results, only the *ratio* of all length dependent input parameters.

To lay the necessary baseline for the discussion of the results of the ensuing study of mixing different fiber diameters, we want to take the time now to illustrate the issue of modelling such size-dependent effects with the used SVE method. Apart from the random distribution of fibers, any of the used SVEs is defined by only three *size-dependent* geometrical parameters:

- Fiber diameter
- Inter-fiber distance *Dmin*
- SVE size

The SVE methodology holds that our transverse strength results shall be *independent* of the *absolute* values for each one of these three geometrical parameters. Instead they only depend on the *ratio* between these three values. For example, the transverse strength obtained will be the same for 8 µm fibers in a 50 µm x 50 µm SVE with $Dmin$ = 0.35 µm, than for 2 µm fibers in a 12.5 µm x 12.5 µm SVE with $Dmin$ = 0.0875 µm. In fact, both represent the same material, defined by the ratio of their size-dependent geometrical parameters. So the question arises, how can we model and distinguish two FRPC materials with different fiber diameters when using SVEs? The answer necessarily has to be that these two different materials must be characterized by different ratios of the three size-dependent geometrical parameters of their SVEs. For the case of the ensuing study, a first intuitive choice could be:

- Material 1: 8 µm fibers in a 50 µm x 50 µm SVE with $Dmin$ = 0.35 µm for the nominal FRPC material

[5] The latter two are generally size independent, all the others are not.

- Material 2: 2µm fibers in a 50µm x 50µm SVE with $Dmin = 0.35$µm for the material of smaller fibers

These two materials correspond with the SVEs shown in Figure 7.22: on the left material 1, on the right material 2. This time we changed the fiber diameter, but kept *Dmin* constant [6]. In consequence, we have two different materials and the transverse strength results for these two materials will be different. However, at the same time we have to remind ourselves that the material 2 is not a material of 2µm fiber diameter, but only a material with a different fiber diameter to $Dmin$ ratio. Material 2 can be equally seen as a material of 8µm fibers and a minimum inter-fiber distance of 1.4µm.

In a previous study we highlighted that increasing $Dmin$ significantly increases transverse tensile strength, see Figure 7.2, and accordingly material 2 shows a higher transverse tensile strength than material 1, as our following results discussion will show. As discussed before, this strength increase has no physical equivalent in reality. When manufacturing a FRPC ply and observing the fiber distribution of a cross-section, it can be noted that something like a minimum inter-fiber distance does not exist. In reality, many fibers appear to be touching, or are very close to touching. A limitation of the CM model used in the proposed framework is that we can not implement touching fibers, but a small minimum inter-fiber distance $Dmin$ has to be ensured in order to mesh the matrix elements in between the fibers, always at least one element.

Summarizing, $Dmin$ is a *necessary* geometrical parameter for our model, and it has to be chosen as a *fixed* value such that our results are comparable and coherent. Therefore, two SVEs which only differ in their $Dmin$ value are **not** adequate to model two different materials, since we have seen that $Dmin$ does not exist in reality and has to be kept equal if results between different models shall be compared.

At this point, we have to clarify a previous statement. The *geometry* of a SVE is indeed fully defined by the three size-dependent geometrical parameters listed above, i.e. fiber diameter, inter-fiber distance, and SVE size. However, the FRPC *material* represented by the SVEs is defined by the material parameter of its three constituents (fiber, matrix, interface). We must distinguish between:

- Size-*in*dependent material properties
  - Strength
  - Stiffness
- Size-dependent material properties
  - Fracture energies, thus indirectly: max. crack-opening displacement of matrix and interface elements.

Therefore, in order to fully define and model a material through SVEs and the CM model implemented in the framework [30], the defining size-dependent parameters are [7]:

- Fiber diameter
- Inter-fiber distance
- Fracture energies of matrix and interface elements

In the case of transverse tension, final failure and overall strength are determined by failure of the interfaces. For such a study, the following two SVEs, i.e. materials, would yield identical results:

- 8µm fibers with $Dmin = 0.35$µm and $G_n^c = 2\,\mathrm{J/m^2}$, and
- 2µm fibers with $Dmin = 0.0875$µm and $G_n^c = 0.5\,\mathrm{J/m^2}$.

[6] The fact that we also kept the size of the SVEs constant shall not concern us at this point. A SVE of 12.5µm x 12.5µm would have been adequate if we want to preserve also the ratio with the SVE size, but larger SVEs have the only effect that the obtained strength result is more converged, i.e. the standard deviation of the 5 strength results decreases, see Figure 7.4.

[7] Omitting SVE size at this point in line with the previously presented reasoning.

Herein, G_n^c denotes the critical fracture energy in the case of pure loading normal to the cohesive elements of the fiber-matrix interface, as explained in Section 3.1.2.2.2. Only by intentionally *not* scaling some of the listed values, can one model a different material. Analog to the example presented above, the SVE on the right of Figure 7.22 corresponds to the following parameters: 2µm fibers with $Dmin = 0.35$µm and $G_n^c = 2\,\mathrm{J/m^2}$. Again, this is identical to a material/SVEs with 8µm fibers, $Dmin = 1.4$µm and $G_n^c = 8\,\mathrm{J/m^2}$. Comparing with the nominal SVEs used in the ensuing study, this is equal to same fiber diameter, but $Dmin$ and G_n^c scaled up by a factor of 4. The undesired but inevitable effect of *Dmin* has been discussed already in the previous paragraph. A similar conclusion has to be drawn for the fracture energy of the fiber-matrix interface G_n^c: when scaled up by a factor of 4, G_n^c results in an unrealistic failure behavior of the SVEs. We shall briefly discuss the influence of G_n^c on the results presented in this chapter:

> Figure 7.14 shows the stress-strain curves for tensile loading and the *nominal* set of input parameters used throughout this master's thesis. With regard to this discussion, they are 8µm fibers, at $Dmin = 0.35$µm and with $G_n^c = 2\,\mathrm{J/m^2}$. All stress-strain curves in Figure 7.14 show the same trend: After a constant slope, damage starts to initiate at the fiber-matrix interfaces and the slope changes up to the point of maximum stress, after which the stress generally drops suddenly and drastically.
>
> The value of G_n^c now determines how soon the SVEs fail after damage initiation, i.e. how brittle or ductile the material behaves. The nominal value of $G_n^c = 2\,\mathrm{J/m^2}$ is relatively high, such that the stress-strain curve has a noticeable *knee*, i.e. after the first interfaces have failed, they still bear enough strain-energy to be dissipated before the stress-strain curve smoothly reaches the point of maximum stress. This is advantageous if one wants to show the progression of the crack growth throughout a SVE, like in Figure 3.11, but the non-linear part of the stress-strain curve is generally not observed when subjecting a FRPC ply to transverse tension in a real-world experiment. In this scenario, they generally fail brittle, without a noticeable non-linear part of the stress-strain curve [34]. Such behavior can also be represented with the used model, just by decreasing the value of G_n^c. Lastly, it has to be remarked that it is extremely challenging to determine the exact value of G_n^c through experiments on real AS4/8552 fiber-matrix interfaces. In consequence, when it comes to choosing a value for G_n^c for your model, it is usually guesstimated/determined such that the resulting failure behavior of the SVEs correlates with experiments.
>
> For the presented studies performed with the proposed framework, the exact value of G_n^c does not play a significant role because the overall obtained *trends* are considered to be less influenced by the value of G_n^c. For the first study, the range of strength values obtained is far greater than the effect that G_n^c could exert, see Figures 7.13 and 7.14. For the second study of mixing different fiber shapes, the absolute value of the obtained strength is less important than being able to compare results for different fiber geometries.

Eventually, we now have to put all these effect into the perspective of the current study, trying to model different fiber diameters through SVEs. The SVE/material on the right of Figure 7.22 is identical to the SVE/material on the left of Figure 7.22, only that the minimum inter-fiber distance $Dmin$ as well as the fracture energies of the matrix and interface elements, most notably G_n^c, are all scaled by a factor of 4. That increasing $Dmin$ significantly increases transverse tensile strength has been discussed before, and the same has now been pointed out for G_n^c. The result of both effects combined is now finally shown in Figure 7.23. Each 5 stress-strain curves are shown for the three SVEs/materials presented in Figure 7.22, subjected to transverse tension.

The purple curves are the results of the nominal SVEs/material 1 used throughout this master's thesis. In yellow are drawn the stress-strain curves of the aforediscussed material 2, which eventually is nothing else than material 1 with $Dmin$ and G_n^c scaled by a factor of 4. As can be seen in Figure 7.23, this results in unrealistic stress-strain curves. Damage initiation occurs at a stress 10 MPa higher than for material 1, an effect primarily attributed to the increased $Dmin$. More notably, final failure only occurs after a very elongated section of the curve. Such behavior can not be expected in reality, and is only due to the dissipation of the large fracture energy value of the fiber-matrix interfaces of $G_n^c = 8\,\mathrm{J/m^2}$ for material 2.

This lengthy discussion, deemed necessary for the ensuing results presentation, can finally be concluded with the following:

- The used methodology of SVEs is inadequate to model FRPC material cross-sections of *each* different fiber diameters, either *only* small fibers, or only large fibers. This is due to the inherent length-scale independent nature of the SVE methodology. An alternative approach would be to use an embedded cell, if the effect of fiber diameter shall be assessed, *not* mixing two different fiber diameters.

- However, what is indeed possible with the used SVE methodology is the modelling of materials *mixing* two different fiber diameters, as is the final goal of this study. Figure 7.23 shows that the stress-strain curves of the SVEs with mixed fiber diameters do agree with behavior observed in experiment. On the other hand, the gain in transverse strength is of course less pronounced than for the unrealistic case of $8FF = 0$, i.e. material 2. In this case of mixing two fiber diameters, the results have to be interpreted keeping in mind the discussed effect of the minimum inter-fiber distance and the fracture energies of the constituents. Both are size-dependent input parameters of the SVEs and material model with significant influence on the transverse strength, but difficult to choose appropriately. With this in mind, the results of the study can now be presented.

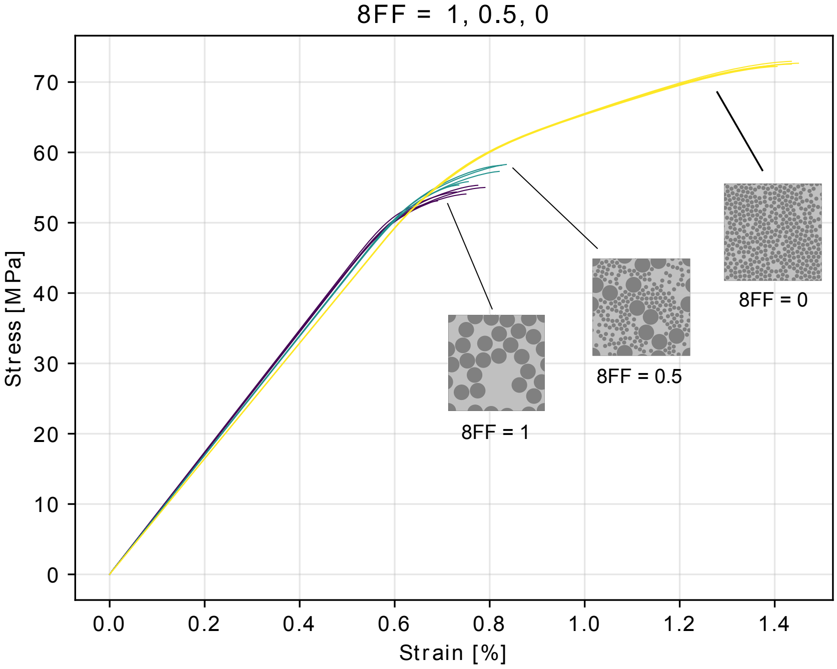


Figure 7.23: Plot of the stress-strain curves for 8µm fiber fractions of $RFF = 0, 0.5, 1$, from left to right, and for each 5 SVE realizations. The stress-strain curves are plotted up to the point of maximum stress. As can be seen, the SVEs for $RFF = 0$, with minimum inter-fiber distance $Dmin$ and interface fracture energy G_n^c scaled by a factor of 4, result in stress-strain curves with an unrealistic non-linear part up to failure.

7.4.2. Tension

For the choice of input parameters and their ranges as outlined above in Section 7.4, the approximation function to be found becomes

$$\sigma_{transverse,tension} = \widetilde{f}(8FF, FVolF). \tag{7.7}$$

A total of 17 datapoints were simulated and none of them aborted. As in the preceding study, the number of datapoints was visually determined. Figure 7.24 shows the spatial distribution of the 17 datapoints as well as the final NN approximation of 7.7.

The first observation which has to be pointed out is the trend of the observed effects. As can be seen, the familiar effect of fiber volume fraction remains the same for all simulated materials. Since we fixed $IFS = MS = 1$, we are in the region of interface-dominated final failure, and thus having less fibers results in a slightly higher transverse tensile strength. The effect that we are most interested in however is the one of mixing the two different fiber diameters. The discussion of the previous section concluded that the results for the lower limit $8FF = 0$, i.e. material 2, have to be discarded. Figure 7.24 illustrates this readily, by the suddenly peaking strength values when approaching $8FF = 0$. The divergence of the surface towards this limit value can be taken as a practical measure for deciding up to which mixing ratio $8FF$ the results are valid. In this case, we might draw this line between $8FF = 0.5$ and $8FF = 0.25$.

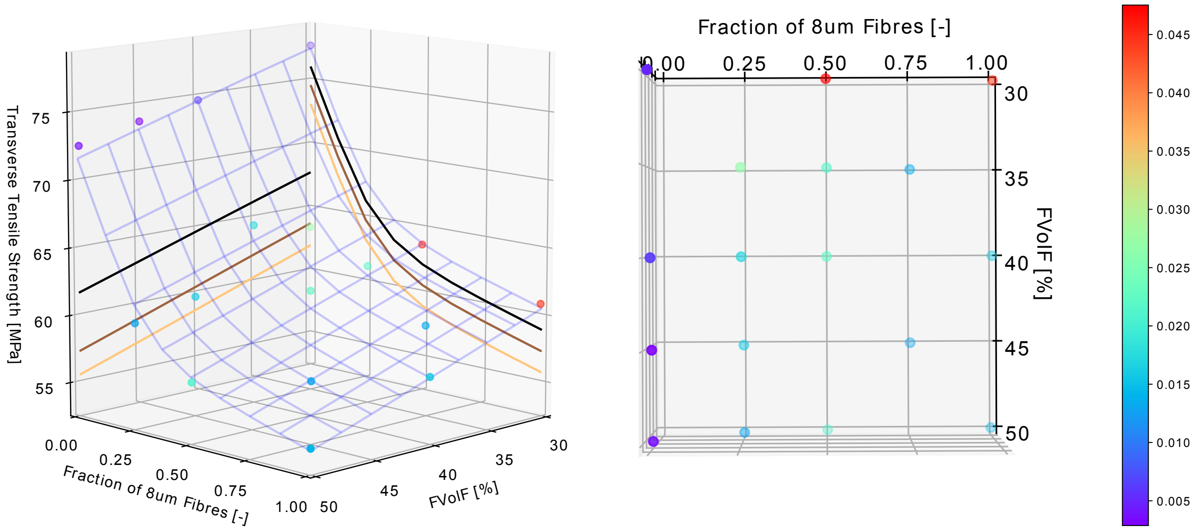


Figure 7.24: NN approximation of Equation 7.7 for a total of 17 datapoints. The output variable transverse tensile strength is plotted over the two input variables fiber volume fraction FVolF and 8μm fiber fraction (8FF). The color of the datapoints corresponds with the relative standard deviation, i.e. standard deviation of the five stress values calculated at this combination of input parameters, divided by their mean stress value. For better readability, the contour lines at fixed values for the FVolF, i.e. $FVolF = 35, 40, 45\%$ and $8FF = 0.25, 0.5, 0.75$ are projected onto the vertical plane.

Up to this value, we observe a steady increase in transverse strength between 4...8 MPa. The comparison of stress-strain curves in Figure 7.23 provides a more detailed explanation. For equal mixing of both fiber diameters, i.e. the curves drawn in green, we can observe that damage initiation occurs slightly later than for the nominal curves in purple. Second, the softening of the stress-strain curve, up to the point of final failure, occurs more gradual. When interpreting the former effect, we must take into account the inter-fiber distance, which is constant throughout the SVEs. As outlined above, what is important for the development of stress concentrations is the *ratio* of inter-fiber distance to fiber diameter. Obviously, for the smaller fibers, this ratio is lower compared to the bigger fibers, and the same may hold for the stress concentrations developed between closely neighboring fibers.

Another effect of a constant value for $Dmin$ is that the distributions of the smaller fibers are more regular. The total volume fraction which is taken up by the sum of the artificial disks of $Dmin$ around each fiber is larger for the smaller fibers, as can be readily observed in Figure 7.22 comparing the left and right SVEs. In consequence, the smaller fibers and constant $Dmin$ don't allow for pronounced resin pockets to form, as is the case for SVEs of larger fibers. Since we are applying a thermal step, significant residual thermal stresses are created within the SVEs, particularly where regions of resin pockets are next to closely neighboring fibers. Damage initiation then generally starts in these regions. In consequence, this is one of the contributing effects to the observation that SVEs of mixed fiber diameters first develop damage around larger fibers, and fail eventually along their lines as well.

When interpreting the more gradual development of damage up to final failure, we must further take into account the fracture energy of the interfaces. Replacing a certain volume fraction of larger fibers with smaller fibers increases the amount of interfaces within a SVE. Scaling the fiber diameter by 4 results in a 4 times higher sum of the circumferences of all fibers, and accordingly 4 times more interfaces within the SVE. For the extreme case of $8FF = 0$, there are four times more interfaces in the SVE. However, final failure occurs generally within a vertical band through the SVE, therefore with only 2 times more interfaces participating. This increased number of interfaces is able to dissipate more fracture energy than in the nominal case. At this point, we have to acknowledge that the model assumes the same fracture energy G_n^c for all interfaces. An important question is whether the interfaces of smaller fibers in reality would have the same G_n^c value. For future work, we therefore recommend to repeat this study for different values of G_n^c for *all* interfaces, or to assign smaller values only to the smaller fibers. Both might change the magnitude or trend of the strength increase as it is shown in Figure 7.24.

7.4.3. Compression

The same SVEs were subjected to transverse compression, with the approximation function to be found being

$$\sigma_{transverse,compression} = \widetilde{f}(8FF, FVolF). \tag{7.8}$$

A total of 29 datapoints were simulated and none of them aborted. Figure 7.25 shows the spatial distribution of the 29 datapoints as well as the final NN approximation of 7.8.

As in the tensile case, increasing the amount of smaller fibers increases the overall strength, for all fiber volume fractions. Contrary to the tensile case, the hybrid effect under compression is stronger than the FVolF effect, with strength increases between 17...25 MPa for mixing in up to 75% of the smaller fibers. Relative to the mean strength values, the strength increase is between 10...15%, just like in the tensile case.

Different than in the tensile case, the results for only smaller fibers, i.e. $8FF = 0$, now fit better with the rest of the surface. This should not be seen as a positive sign, but more as a sign of caution when interpreting the remaining data. Transverse damage starts with interface debonding, which then triggers damage of the matrix under shear. Final failure consists of a linked shear band within the matrix, all through the SVEs. The value of the interface fracture energies G_n^c and G_s^c plays a significant role in determining how soon and how severe the interface failure triggers damage within the matrix, which eventually causes final failure. Lastly, the effect of the minimum inter-fiber distance $Dmin$ should be considered, although the preceding study suggested that its influence in the case of compressive loading is less significant, see Section 7.3.4. For future studies, further investigations into both of these aspects are recommended, in order to substantiate the significant strength increase found in this study.

Lastly, an interesting observation can be made: For the nominal material used throughout this master's thesis, i.e. $8FF = 1$, representing today's FRPC materials, increasing the FVolF decreases the transverse compressive strength. When adding smaller fibers to the SVEs, this effect steadily reverses, up to the point of $8FF = 0$ where an increased fiber volume fraction increases the compressive strength. As discussed before, these last results represent a different material, far from today's possibilities. Still, the trend shows that increasing the fracture toughness of the fiber-matrix interfaces would greatly benefit the transverse compressive strength of UD CFRPC plies. Up to the point, that high fiber volume fractions would increase the strength, as in the case of replacing circular with 4-lobed fibers see Section 7.3.4. Since high FVolF are generally also desirable for best longitudinal properties, this would be especially advantageous.

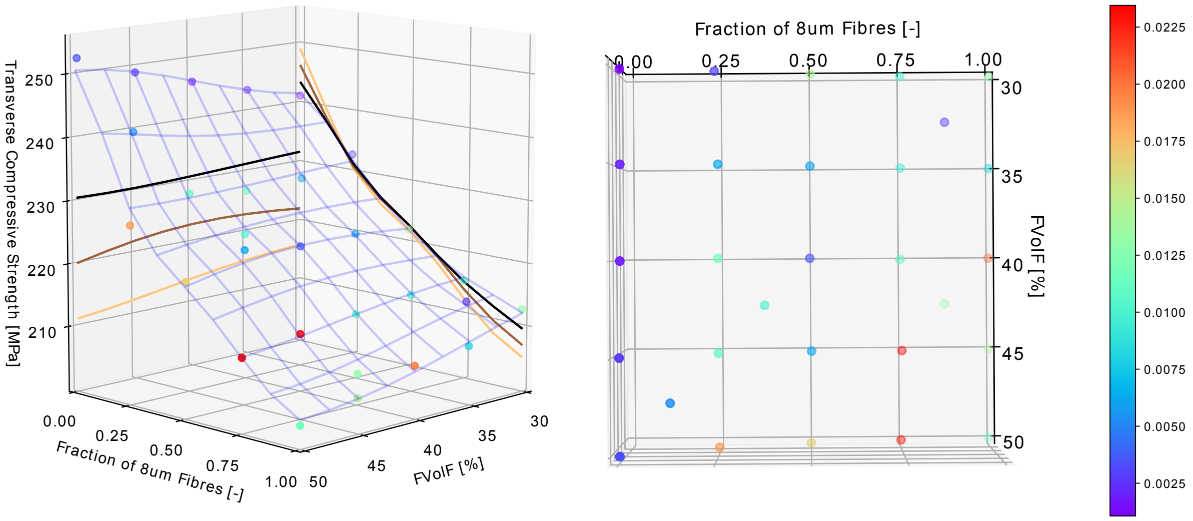


Figure 7.25: NN approximation of Equation 7.7 for a total of 29 datapoints. The output variable transverse compressive strength is plotted over the two input variables fiber volume fraction FVolF and 8µm fiber fraction (8FF). The color of the datapoints corresponds with the relative standard deviation, i.e. standard deviation of the five stress values calculated at this combination of input parameters, divided by their mean stress value. For better readability, the contour lines at fixed values for the FVolF, i.e. $FVolF = 35, 40, 45\%$ and $8FF = 0.25, 0.5, 0.75$ are projected onto the vertical plane.

7.4.4. Summary

Hereby, we conclude the presentation of results for the study of the transverse tensile/compressive strength of a UD CFRPC ply, as a function of the total fiber volume fraction and the fraction of bigger to smaller fibers within the sves. The developed framework was used to determine the corresponding surrogate models which describe this relationship, in an automated way, and based on a non-optimized amount of 17 and 29 datapoints respectively.

The surrogate models suggest that both transverse tensile and compressive strength could be increased by up to 10...15 % for mixing smaller fibers with our nominal AS4-CF. However, this rests on the assumption that the interfaces of the smaller fibers would have the same fracture energies than their larger counterparts. Furthermore, the influence of the minimum inter-fiber distance *Dmin* has to be weighed in. Throughout each SVE, *Dmin* was kept constant meaning that stress concentrations will focus around the larger fibers. Since *Dmin* is a non-physical modelling parameter which does not exist in the reality of actual FRPC plies, its contribution to the shown hybrid effect should be further assessed in future studies, in particular for tensile loading. The same holds for the influence of the value of the fracture energies of the smaller fibers.

While the generated results provide a possible first insight into the transverse strength of a hybrid UD CFRPC ply, they also highlighted the challenges of modelling the effect of size-dependent parameters with the SVE methodology. The effect of all size-dependent modelling parameters must be well known and considered, and the minimum inter-fiber distance introduces another undesired influence. A CM model which can consider SVEs with touching fibers, and adequately model the fiber-fiber interface, would be advantageous but remains for now on the further horizon.

Having presented the results of the last example study, we may now summarize the results of all studies.

7.5. Summary

Hereby, we conclude the chapter presenting the five example studies performed with the proposed framework. For varying input parameters, we determined their functional relationship with the transverse tensile and compressive strength of a UD FRPC ply. This was achieved by the proposed 'smart' framework for Computational Micromechanics in an automated way, and based on a user-defined minimum number of simulated datapoints. As was shown in this chapter, the developed framework can be used to perform a wide variety of studies in a conveniently automated way. Since it is not limited in the number of parameters to be explored simultaneously, it can be used to determine *global* surrogate models, relating for example a complex ply property such as transverse strength to all dependant parameters deemed relevant by the user.

In the first two studies, we characterized two important modelling parameters of the used Computational Micromechanics (CM) model and methodology of Statistical Volume Elements (SVEs): minimum inter-fiber distance *Dmin* and SVE size. The former was confirmed to have a significant influence on transverse tensile strength, as reported in literature. In order for results to be comparable, we fixed *Dmin* at a nominal value of 0.35µm. The latter, i.e. SVE size, was shown to depend on the Fiber Volume Fraction (FVolF) as expected. The framework in its current status required a fixed SVE size for all studies to be performed, varying FVolF over the wide range of $30\% \leq FVoLF \leq 70\%$. Based on the results of the second study, we chose a nominal SVE size of 50µm x 50µm as a compromise between convergence of the results and required computation time. For $FVoLF \geq 40\%$, the nominal SVE size results in well converged strength results. For an evolution of the developed framework, it could be envisaged to implement a variable SVE size. It could be adapting to the FVolF of the datapoint to be computed, or be increased automatically until every datapoints achieves the same accuracy.

After having adequately fixed these modelling parameters, three studies were performed with the developed framework, in order to globally and quantitatively characterize the transverse strength of a UD FRPC ply.

First, the dominant influence of the constituent strengths was confirmed. The ratio of Matrix Strength (MS) and Interface Strength (IFS) defines the overall failure mode of the ply, either dominated by interface or matrix failure, or a mixture of both. For the tension case, today's CFRPC materials lie well within the interface-dominated region. For compressive loading, our results suggest that both constituent strengths are equally contributing. Recalling the objective to improve the transverse strength of today's CFRPC materials, for a steepest increase under *both* loading conditions, research needs to focus on improving the strength of the fiber-matrix interface. What might be easier to achieve today is to improve only the matrix strength. Our results suggest that this would improve the transverse compressive strength of the ply, while keeping the tensile strength unchanged. Lastly, the effect of the FVolF was determined. It can be generalized as the following: Increasing the volume fraction of the failure-dominating constituent decreases the overall transverse strength. For today's CFRPC materials, higher FVolF decreases the overall transverse strength of the ply. Our results confirm that in the compression case the magnitude of this effect is smaller. In general, the effect could be reversed for an increased IFS.

The last two example studies were dedicated to exploring the effect of fiber-by-fiber hybridization of a UD ply of today's CFRPC materials. To do so, the constituent strengths were fixed at their nominal values $MS = IFS = 1$ and new geometrical parameters were introduced.

In a first case, fibers of circular cross-sections were mixed with 4-lobed ones, for varying FVolFs. No significant change in transverse *tensile* strength was found. Compared to results from literature [30], this is a positive outcome, as it revokes a reported tensile strength *de*crease for 4-lobed fibers. For compressive loading, the case changes. A significant strength increase of up to +13% was found for increasing fractions of 4-lobed fibers, most amplified for high FVolFs. For only 4-lobed fibers, the common strength-decreasing effect of high FVolFs is even reversed. Aside from these findings, it was highlighted that non-circular fiber shapes may introduce another influential geometrical parameter: the orientation of the cross-section. The results in the tension case suggested that the latter further depends on the FVolF, as certain fiber shapes like to arrange in distinct non-random patterns for a given FVolF. That such an alignment may have a significant effect on transverse strength has been reported in [30] and is re-suggested by our results.

In the last study finally, our nominal circular fibers were mixed with smaller fibers, also circular, again for varying FVolFs. Significant increases in both transverse tensile and compressive strength were found, for increasing fractions of smaller fibers. At the same time, these results were obtained for the assumption that both the nominal and the smaller fibers have the same IFS, including fracture energies, i.e. crack-opening displacement. In particular the latter part is questionable, that the crack-opening displacement be independent of the fiber diameter. Further studies are recommended to investigate a possible influence, assigning a different fracture energy to the smaller fibers. Lastly, the effect of the minimum inter-fiber distance *Dmin* was pointed out. In SVEs of mixed fiber diameters, only a single value of *Dmin* can be imposed. In consequence, stress concentrations focused around those fibers with smaller *Dmin*-to-diameter ratio, i.e. the larger fibers. In summary, while promising results were achieved, we also had to take note of some of the challenges when trying to model size-dependent effects with the used SVE methodology. In particular, the size-dependent parameters *Dmin* and the interface fracture energies have to be well controlled.

8

Conclusions

We now have presented all major parts of the work of this master's thesis. In order to draw conclusions, we shall turn towards the research questions that we initially set out with in the introduction:

I Can we adapt the combined DoE-CM-NN approach to efficiently generate a comprehensive analytical surrogate model of a complex material property such as the transverse strength of a UD FRPC ply?

II Putting to use the developed framework, what are the main parameters determining the low transverse strength of a UD FRPC ply? Can hybridization at fiber-by-fiber level, i.e. the mixing of circular and non-circular fiber shapes, and the mixing of different fiber diameters improve transverse strength?

III Based on the potential answers found to these last questions through the proposed framework, what are the most promising directions of research to improve the transverse strength of FRPC materials?

The first question, about the methodological proof-of-concept of the framework, can be answered positively:

- The baseline of the work of this thesis is the approach of combining Design of Experiments (DoE), Computational Micromechanics (CM), and Neural Networks (NNs). While this idea has been reported on in literature [7][23][6], it had not yet been applied to the complex problem of failure prediction of a composite material at ply level. In this work we were now able to show that the CM model required for this problem is both mature and computationally cheap enough to allow for its automatic exploitation through such a combined approach. Moreover, the developed framework for 'smart' Computational Micromechanics allowed to determine a comprehensive analytical surrogate model of the transverse strength of a Unidirectional Fiber-Reinforced-Polymer-Composite (UD FRPC) ply.
- Previously, studies of such failure predictions by CM have only explored the effect of *one* parameter at a time [30][27][44]. In contrast, the proposed 'smart' framework now allows to characterize the *interrelated* influence of several parameters, at the same time and in a fully automated way. In one of the performed studies, we so determined the surrogate model that relates the overall transverse strength of the ply with its main dependencies: Matrix Strength (MS), fiber-matrix Interface Strength (Interface Strength), and Fiber Volume Fraction (FVolF).
- The question may be raised whether this surrogate model is truly 'comprehensive', or whether more input parameters would need to be considered for that. The latter is not a problem for the developed framework, as it is generally not limited in the number of input parameters to vary. More modelling parameters may be readily included. In another study not reported herein we added the minimum inter-fiber distance *Dmin* to the three parameters mentioned above, a further extension with the fiber-matrix fracture energies may be interesting to consider as well. The main challenge for increasing numbers of parameters is expected to be the computation time. For the presented studies in Chapter 7, this was not an issue yet. In fact, we did not have to care about optimizing the number of datapoints to simulate, or the computation time of each simulation, or to move our computations to a cluster. For more than five input parameters however, increasing computation time and these three mitigation strategies are expected to become more relevant.

The second research question, about insights gained into the transverse strength of a UD FRPC ply, can be answered with the following:

- The first surrogate model generated with the framework indicates that the main parameters determining the transverse strength of a UD FRPC ply are the constituent strengths, i.e. MS and IFS. Their *ratio* determines which constituent is the dominant contribution to the final failure mode, and this ratio depends on the loading case. Under tensile loading, the found surrogate model shows that the ratio of their tensile strength components indicates 1:1 which constituent fails primarily. For today's AS4-CF/8552 material for example, the much weaker interface fails first. Accordingly, the only way to increase transverse tensile strength is by increasing the Interface Strength. If the IFS could be increased up to the value of the MS, the overall failure mode would become an equal mix of matrix and interface failure.

  The situation changes under compressive loading. Again, the ratio of the constituent strengths is decisive, but its value is different than in the tensile case. The determined surrogate model quantifies that for today's AS4-CF/8552 material, both constituent strengths are equally contributing to overall failure. This is surprising because the IFS is only about one third of the value of the matrix strength[1]. The explanation is the mixed failure observed in experiments of UD FRPC plies under transverse compression. Final failure occurs as a combination of interface debonding and the development of shear bands within the matrix. The results generated by the framework now quantify the contributions of each failure mode, which turn out to be about equal for today's AS4-CF/8552 material. In consequence, this suggests that the transverse compressive strength of these materials could be readily improved by increasing any of the two, either MS or IFS.

  Aside from the primary effect of the constituent strengths, a secondary effect on transverse strength was attributed to the Fiber Volume Fraction (FVolF). For both tensile and compressive loading, it can be generalized that increasing the volume fraction of the weaker constituent decreases the strength. Accordingly, today's AS4-CF/8552 material falls well within the region where higher FVolF leads to a strength decrease. For increased IFS, this effect could be reversed.

- The second and third surrogate models generated with the framework explored the effect of hybridization on the transverse strength of a UD FRPC ply. For this, the constituent strengths were fixed at the nominal values measured for AS4-CF/8552, i.e. $MS = IFS = 1$. First, fibers of circular and 4-lobed cross-sections were mixed, at varying total FVolFs. The results indicate no change of transverse *tensile* strength. Under compressive loading however, the overall strength increased by up to +13% for increasing fractions of 4-lobed fibers, an effect showing most amplified for high FVolFs.

  In the second part, the nominal AS4-CF were mixed with their counterparts of one fourth the nominal diameter. The results show notable increases in both transverse tensile and compressive strength, for increasing fractions of the smaller fibers. In the tension case, this effect is independent of the FVolF, in the compression case it is intensified for higher FVolF. It has to be remarked that these results were obtained for the assumption that both the nominal and the smaller fibers have the same IFS, including fracture energies, i.e. crack-opening displacement. Further studies are recommended to substantiate this assumption, or to investigate to which extent the observed effect changes if a different fracture energy is assigned to the smaller fibers.

The third research question asks for suggestions to improve the low transverse strength of today's FRPC materials. Based on the obtained results, the following can be summarized:

- The performed studies confirm that the steepest increase in transverse strength can only be achieved by directly increasing the strength of the matrix or the fiber-matrix interface. For the tensile loading case, only an increase in IFS can lead to improvements for today's AS4-CF/8552 material. For the compressive loading case and the same material, both an increase in IFS or MS would equally increase the transverse compressive strength of the ply.

  An improvement of secondary order could be achieved by fiber-by-fiber hybridization. Replacing circular AS4-CF with 4-lobed fibers of identical properties would have no effect on transverse tensile strength, but increase the compressive strength of the ply by up to +13%, as suggested by our results.

[1] Comparing their tensile components.

Mixing in smaller fibers is indicated to lead to improvements of both tensile and compressive strength of a AS4-CF/8552 ply. For increased confidence in these latter results however, we recommend an extension of the performed study, to determine the dependence of these results on the interface fracture energies of the smaller fibers.

9 Future Work

Based on the findings of this master's thesis, we want to recommend possible ways to continue this work. We shall distinguish three categories. First, we recommend ways to further exploit the already generated data. Second, we lay out options how the performed studies could be extended in promising ways. Third, we suggest improvements and additions to the developed 'smart' framework.

Without requiring any extra analyses, only by exploiting the already generated data, we propose the following:

- Exploit strain: As mentioned before, the shown results in Chapter 7 only used a single point of the stress-strain curve which has been generated for every simulation: the maximum stress. Without requiring any additional analyses, surrogate models could be generated for the strain at failure, for example. This property of a Unidirectional Fiber-Reinforced-Polymer-Composite (UD FRPC) ply is equally important to consider in layups like the cross-ply, where the transversely loaded ply may fail early on due to its low strain-to-failure. As we have observed and discussed before, the observed variations in strain-to-failure are higher than for the maximum stress. Therefore, a more precise determination of the surrogate models for strain-to-failure may require a larger Statistical Volume Element (SVE), and/or a fine-tuning of the chosen values for the interface fracture energies.

- Exploit strain-energy-at-failure: On the same note, another ply property of interest could be the macroscopic strain energy dissipated until failure. This value has been recorded for all performed simulations and the corresponding surrogate models could be readily generated.

- Approximate the full stress-strain curve: In order to exploit the valuable generated data as much as possible, another possibility would be to let the Neural Network (NN) approximate the entire stress-strain curve. An example case has been reported in [7], although the SVE approach used in this thesis would require some adaptations to this. While a NN generated stress-strain curve would be both impressive and very visual, it would be harder to compare the full stress-strain curvers in higher-dimensional plots like the ones we showed in Chapter 7. At the same time, the question has to be raised how much additional information does the full curve provide, which we can not extract in a more distilled way otherwise? The main point of interest could be the non-linear part of the curve, albeit it generally being very small for today's FRPC materials under transverse loading. Still, the full stress-strain curve may aid in the development of a constitutive description of the material until failure.

- Combine plots: Chapter 7 illustrated the analytical surrogate model obtained for each study and each loading case. The advantages of having an analytical description of a complex ply property were touched upon with the gradient plots, but could be explored further. In an example case not reported in this document, we combined the plots for transverse tension and compression of the first study on the effect of the constituent strengths. The combined plot can then be a summation of both, such that the user can easily and visually optimize for a common maximum of both transverse tensile and compressive strength. More complex design rules could be imagined, for example to specify a minimum transverse tensile strength and strain-energy-at-failure, and then to optimize for transverse compression in the remaining area of possible input parameter combinations.

- Determine simple constitutive equations: The found surrogate models are already analytical functions describing the material property relationship that we are interested in. Although they describe a function which is not too complex, a NN of even just a single layer and ten neurons is already too complicated to write out. Since the underlying function is generally not very complex, it should be attempted to describe it with a simple, readable and easily understandable equation. Eventually, this could be a logical next step in the data-driven development of constitutive equations describing frpc materials up to failure.

With the possibility at hand to extend the performed studies, we propose the following:

- Regarding the first study, the development of a single comprehensive surrogate model characterizing the transverse strength of a UD FRPC ply:

  In the first study, as presented in Section 7.2, we extended the limits of Interface Strength (IFS) and Matrix Strength (IFS) unrealistically far. This was done since we wanted to determine the overall and general relationship between the constituent strengths and macroscopic transverse strength. Unsure how far we would have to extend the values of IFS and MS for that, we ended up with the presented results. For a follow-up study however, we can now significantly reduce their ranges, covering matrix or interface degradation in relevant use cases such as HOT/WET, and instead introduce the simulated loading case as a continuous parameter to vary. A model of longitudinal and in-plane shear [73] as well as for mixed loading cases [72] has been readily developed at *IMDEA Materials* and could be implemented. The final goal would be to obtain a single surrogate model describing all failure modes of a UD FRPC ply which can be modelled with the given 2D or shallow 3D SVE representation.

  Lastly, the inclusion of void modelling could improve the accuracy and relevance of our surrogate models. A relevant modelling strategy has been readily developed at *IMDEA Materials* [75], and could be implemented into the used Computational Micromechanics (CM) model.

- Regarding the second study, the mixing of circular and 4-lobed fibers:

  In the tensile case, the strength values showed surprising oscillations at small magnitude for fixed Fiber Volume Fraction (FVolF) but varying mixing ratios. A suspected reason for that is the packing of the 4-lobed fibers, which might depend on the FVolF. To clarify this effect, we suggest to repeat this study for larger and more SVEs per datapoint, and to record the angular orientation of all fibers within the SVEs. It is suspected that the distribution of the latter is not uniformly random, but would change in a deterministic way, depending on the FVolF and the mixing ratio. For the case of 4-lobed fibers the effect on transverse strength may be small, but that could change for more anisotropic fiber shapes, such as 2-lobed. As shown in [30], a uniform alignment of such non-circular fibers can have a significant effect on transverse strength of a UD FRPC ply.

- Regarding the third study, the mixing of nominal and smaller fibers:

  As mentioned in the conclusions of the previous chapter, the results of this study depend on the value chosen for the fracture energies of the smaller fibers. In this case, the same value was chosen for both the nominal and the smaller fibers. It is recommended to assign different values for the fracture energies of the interfaces of the smaller fibers, e.g. as an additional input parameter to vary. Hereby, the robustness of the shown results for these modelling assumptions could be quantified, and so the reported promising hybrid effect could be judged more precisely.

A main goal of this master's thesis was the development of the proposed framework for 'smart' Computational Micromechanics. While the general idea of combining Design of Experiments, Computational Modelling, and Neural Networks is not new, its specific adaptation to our CM model, including failure modelling of the constituents, posed a set of unique challenges. To further improve on the developed framework, we propose the following:

- Implement other space-filling algorithm: As suggested by [64], a space-filling algorithm which combines regular and random features for filling the parameter space may improve the results and convergence speed of the DoE approach. Suggested methods by [6] are *Latin Hypercube Sampling* [45] and *Sobol Sequence* [68].

- Automatically adapt the SVE size: As discussed in Section 4.4.2 and Section 7.1.2, the size of the used SVEs determines the accuracy of each datapoint. In the current version of the framework, a fixed value has been chosen for the size of the SVEs. Since we are varying the FVolF in many studies as one of the input parameters, it would be beneficial to scale the SVE size according to the FVolF, such that the number of fibers within the SVE is maintained constant. A more complex rule could be to choose the SVE size for every datapoint to compute such that the variation of the SVE results reaches a set minimum value.

- Consolidate the criterion to determine the optimum number of datapoints: As discussed in Section 6.5.1, it is difficult to specify a general rule or threshold accuracy, by which the framework can automatically determine how many datapoints it has to compute until the results are sufficiently converged. The maximum accuracy which can be reached within a reasonable computation time may vary between studies, loading cases and the number of input parameters. To find a suitable general criterion would be advantageous, although the framework in its current form is not significantly hindered by not having such a criterion. Visual check of the convergence plot by the user is equally sufficient for now.

- Allow irregular parameter grids: The current version of the developed framework can only generate "regular" parameter spaces, combining all input parameters in all combinations throughout their value ranges. However, it may be advantageous to be able to specify a-priori to exclude certain areas of the parameter space. In the case of the first study presented in Section 7.2, it would have been beneficial to not simulate any combinations of low IFS and high MS, particularly for low FVolFs. The implementation of a space-filling algorithm which takes such boundary conditions into account is complex, but has been reported on [38].

- Implement tools for efficient evaluation of high-dimensional data: The example studies performed with the proposed framework were limited to a maximum of three input parameters and a single output parameter. When further extending the number of input parameters, as proposed above, the visualization and representation of the resulting high-dimensional data will become a challenge. In order to cope with this, appropriate data-visualization and analysis tools should be implemented into the framework.

Bibliography

[1] Composites market size, share trends analysis report by product (carbon, glass), by resin (thermosetting, thermoplastic), by manufacturing process, by application, by end use, and segment forecasts, 2018 - 2024, July 2018.

[2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[3] James A Anderson, Edward Rosenfeld, and Andras Pellionisz. *Neurocomputing*. MIT press, 1990.

[4] JA Artero-Guerrero, J Pernas-Sanchez, J Martin-Montal, D Varas, and J López-Puente. The influence of laminate stacking sequence on ballistic limit using a combined experimental/fem/artificial neural networks (ann) methodology. *Composite Structures*, 183:299–308, 2018.

[5] ML Benzeggagh and M Kenane. Measurement of mixed-mode delamination fracture toughness of unidirectional glass/epoxy composites with mixed-mode bending apparatus. *Composites science and technology*, 56(4):439–449, 1996.

[6] MA 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, 2017.

[7] C Bisagni and L Lanzi. Post-buckling optimisation of composite stiffened panels using neural networks. *Composite Structures*, 58(2):237–247, 2002.

[8] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[9] Ian Bond, Martyn Hucker, Paul Weaver, Stephen Bleay, and Sajad Haq. Mechanical behaviour of circular and triangular glass fibres and their composites. *Composites Science and Technology*, 62(7-8):1051–1061, 2002.

[10] Frank Burden and Dave Winkler. Bayesian regularization of neural networks. In *Artificial neural networks*, pages 23–42. Springer, 2008.

[11] S Ramon Cajal. The structure and connections of neurons. *Nobel Lecture December 12, 1906*, 1906.

[12] Pedro P Camanho and Carlos G Dávila. Mixed-mode decohesion finite elements for the simulation of delamination in composite materials. 2002.

[13] Luis Pablo Canal, Carlos González, Javier Segurado, and Javier LLorca. Intraply fracture of fiber-reinforced composites: microscopic mechanisms and modeling. *Composites science and technology*, 72(11):1223–1232, 2012.

[14] François Chollet et al. Keras, 2015.

[15] PS Churchland and TJ Sejnowski. Computational neuroscience. the computational brain, 1992.

[16] Howard Demuth and Mark Beale. Neural network toolbox. 1992.

[17] Shiqiang Deng, Lin Ye, and Yiu-Wing Mai. Influence of fibre cross-sectional aspect ratio on mechanical properties of glass fibre/epoxy composites i. tensile and flexure behaviour. *Composites science and technology*, 59(9):1331–1339, 1999.

[18] DD Edie and MG Dunham. Melt spinning pitch-based carbon fibers. *Carbon*, 27(5):647–655, 1989.

[19] DD Edie, NK Fox, BC Barnett, and CC Fain. Melt-spun non-circular carbon fibers. *Carbon*, 24(4):477–482, 1986.

[20] DD Edie, CC Fain, KE Robinson, AM Harper, and DK Rogers. Ribbon-shape carbon fibers for thermal management. *Carbon*, 31(6):941–949, 1993.

[21] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2015.

[22] Roger Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[23] Jakub Gajewski, Przemysław Golewski, and Tomasz Sadowski. Geometry optimization of a thin-walled element for an air structure using hybrid system integrating artificial neural network and finite element method. *Composite Structures*, 159:589–599, 2017.

[24] FW Gembicki. Vector optimization for control with performance and parameter sensitivity indices. *Ph. D. thesis, Case Western Reserve Univ.*, 1974.

[25] Hossein Ghayoor, Suong V Hoa, and Catharine C Marsden. A micromechanical study of stress concentrations in composites. *Composites Part B: Engineering*, 132:115–124, 2018.

[26] Ronald F Gibson. *Principles of composite material mechanics*. CRC press, 2011.

[27] Carlos González and Javier LLorca. Mechanical behavior of unidirectional fiber-reinforced polymers under transverse compression: microscopic mechanisms and modeling. *Composites Science and Technology*, 67(13):2795–2806, 2007.

[28] Mark Hardiman, Ted Joseph Vaughan, and Conor T McCarthy. A review of key developments and pertinent issues in nanoindentation testing of fibre reinforced plastic microstructures. *Composite Structures*, 180:782–798, 2017.

[29] Simon Haykin. Neural networks: a comprehensive foundation. *Mc Millan, New Jersey*, 2004.

[30] M Herráez, C González, CS Lopes, R Guzmán de Villoria, J LLorca, T Varela, and J Sánchez. Computational micromechanics evaluation of the effect of fibre shape on the transverse strength of unidirectional composites: an approach to virtual materials design. *Composites Part A: Applied Science and Manufacturing*, 91:484–492, 2016.

[31] Miguel Herráez, Diego Mora, Fernando Naya, Claudio S Lopes, Carlos González, and Javier LLorca. Transverse cracking of cross-ply laminates: A computational micromechanics perspective. *Composites Science and Technology*, 110:196–204, 2015.

[32] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[33] Mike J Hinton, A Sam Kaddour, and Peter D Soden. *Failure criteria in fibre reinforced polymer composites: the world-wide failure exercise*. Elsevier, 2004.

[34] Derek Hull and TW Clyne. *An introduction to composite materials*. Cambridge university press, 1996.

[35] T-J Kim and C-K Park. Flexural and tensile strength developments of various shape carbon fiber-reinforced lightweight cementitious composites. *Cement and concrete research*, 28(7):955–960, 1998.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[38] L Lanzi. *Optimisation of composite stiffened panels under post-buckling constraints.* PhD thesis, Dept. of Ingegneria Aerospaziale Politecnico di Milano, 2004, 2004.

[39] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

[40] JA Lee, DP Almond, and B Harris. The use of neural networks for the prediction of fatigue lives of composite materials. *Composites Part A: Applied Science and Manufacturing*, 30(10):1159–1169, 1999.

[41] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[42] Javier LLorca, Carlos González, Jon M Molina-Aldareguía, Javier Segurado, Rocio Seltzer, Federico Sket, Marcos Rodríguez, Sergio Sádaba, Raul Muñoz, and Luis Pablo Canal. Multiscale modeling of composite materials: a roadmap towards virtual testing. *Advanced Materials*, 23(44):5130–5147, 2011.

[43] J Lubliner, J Oliver, Sand Oller, and E Onate. A plastic-damage model for concrete. *International Journal of solids and structures*, 25(3):299–326, 1989.

[44] AR Maligno, NA Warrior, and AC Long. Effects of inter-fibre spacing on damage evolution in unidirectional (ud) fibre-reinforced composites. *European Journal of Mechanics-A/Solids*, 28(4):768–776, 2009.

[45] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1): 55–61, 2000.

[46] Tim Mueller, Aaron Gilad Kusne, and Rampi Ramprasad. Machine learning in materials science: Recent progress and emerging applications. *Rev. Comput. Chem*, 2015.

[47] F Naya, C González, CS Lopes, S Van der Veen, and J LLorca. Combined multi-scale simulations in fiber-reinforced composites. In *ECCM16-16th European Conference on Composite Materials*, 2014.

[48] F Naya, C González, CS Lopes, S Van der Veen, and F Pons. Computational micromechanics of the transverse and shear behavior of unidirectional fiber reinforced polymers including environmental effects. *Composites Part A: Applied Science and Manufacturing*, 92:146–157, 2017.

[49] F Naya, JM Molina-Aldareguia, CS Lopes, C González, and J LLorca. Interface characterization in fiber-reinforced polymer–matrix composites. *JOM*, 69(1):13–21, 2017.

[50] Shinji Ogihara and Jun Koyanagi. Investigation of combined stress state failure criterion for glass fiber/epoxy interface by the cruciform specimen method. *Composites Science and Technology*, 70(1):143–150, 2010.

[51] HR Pakravan, M Jamshidi, M Latif, and F Pacheco-Torgal. Influence of acrylic fibers geometry on the mechanical performance of fiber-cement composites. *Journal of Applied Polymer Science*, 125(4):3050–3057, 2012.

[52] S-J Park, M-K Seo, and H-B Shim. Effect of fiber shapes on physical characteristics of non-circular carbon fibers-reinforced composites. *Materials Science and Engineering: A*, 352(1-2):34–39, 2003.

[53] Soo-Jin Park, Min-Kang Seo, Hwan-Boh Shim, and Kyong-Yop Rhee. Effect of different cross-section types on mechanical properties of carbon fibers-reinforced cement composites. *Materials Science and Engineering: A*, 366(2):348–355, 2004.

[54] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[55] Ditho Pulungan, Gilles Lubineau, Arief Yudhanto, Recep Yaldiz, and Warden Schijve. Identifying design parameters controlling damage behaviors of continuous fiber-reinforced thermoplastic composites using micromechanics as a virtual testing tool. *International Journal of Solids and Structures*, 117:177–190, 2017.

[56] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1): 145–151, 1999.

[57] Lina Riaño, Lenaïk Belec, and Yoann Joliff. Validation of a representative volume element for unidirectional fiber-reinforced composites: Case of a monotonic traction in its cross section. *Composite Structures*, 154:11–16, 2016.

[58] Maximilian James Ripepi. Transverse mechanical properties of unidirectionally reinforced hybrid fiber composites. 2013.

[59] M Rodríguez, Jon M Molina-Aldareguía, C González, and J LLorca. Determination of the mechanical properties of amorphous materials through instrumented nanoindentation. *Acta Materialia*, 60(9): 3953–3964, 2012.

[60] M Rodríguez, Jon M Molina-Aldareguía, C González, and J LLorca. A methodology to measure the interface shear strength by means of the fiber push-in test. *Composites Science and Technology*, 72(15): 1924–1932, 2012.

[61] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[62] David E Rumelhart, James L McClelland, PDP Research Group, et al. *Parallel distributed processing*, volume 1. MIT press Cambridge, MA, 1987.

[63] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3, 1959.

[64] J Santiago, M Claeys-Bruno, and M Sergent. Construction of space-filling designs using wsp algorithm for high dimensional spaces. *Chemometrics and Intelligent Laboratory Systems*, 113:26–31, 2012.

[65] Ingo Scheider. Cohesive model for crack propagation analyses of structures with elastic–plastic material behavior foundations and implementation. *GKSS research center, Geesthacht*, 2001.

[66] Hwan-Boh Shim, Min-Kang Seo, and Soo-Jin Park. Thermal conductivity and mechanical properties of various cross-section types carbon fiber-reinforced composites. *Journal of materials science*, 37(9): 1881–1885, 2002.

[67] Timothy W Simpson, Dennis KJ Lin, and Wei Chen. Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications*, 2(3):209–240, 2001.

[68] Ilya M Sobol. Uniformly distributed sequences with an additional uniform property. *USSR Computational Mathematics and Mathematical Physics*, 16(5):236–242, 1976.

[69] CT Sun and RS Vaidya. Prediction of composite properties from a representative volume element. *Composites Science and Technology*, 56(2):171–179, 1996.

[70] Dassault Systèmes. Abaqus analysis user's manual. *Simulia Corp. Providence, RI, USA*, 2007.

[71] Jehuda Tirosh, E Katz, G Lifschuetz, and AS Tetelman. The role of fibrous reinforcements well bonded or partially debonded on the transverse strength of composite materials. *Engineering Fracture Mechanics*, 12(2):267–277, 1979.

[72] Essam Totry, Carlos González, and Javier LLorca. Prediction of the failure locus of c/peek composites under transverse compression and longitudinal shear through computational micromechanics. *Composites Science and Technology*, 68(15-16):3128–3136, 2008.

[73] Essam Totry, Jon M Molina-Aldareguía, Carlos González, and Javier LLorca. Effect of fiber, matrix and interface properties on the in-plane shear deformation of carbon-fiber reinforced composites. *Composites Science and Technology*, 70(6):970–980, 2010.

[74] Stephen W Tsai, Donald F Adams, and Douglas R Doner. Analyses of composite structures. Technical report, PHILCO CORP NEWPORT BEACH CA, 1966.

[75] Danial Ashouri Vajari. A micromechanical study of porous composites under longitudinal shear and transverse normal loading. *Composite Structures*, 125:266–276, 2015.

[76] Danial Ashouri Vajari, Carlos González, Javier Llorca, and Brian Nyvang Legarth. A numerical study of the influence of microvoids in the transverse mechanical response of unidirectional composites. *Composites Science and Technology*, 97:46–54, 2014.

[77] TJ Vaughan and CT McCarthy. A combined experimental–numerical approach for generating statistically equivalent fibre distributions for high strength laminated composite materials. *Composites Science and Technology*, 70(2):291–297, 2010.

[78] TJ Vaughan and CT McCarthy. Micromechanical modelling of the transverse damage behaviour in fibre reinforced composites. *Composites Science and Technology*, 71(3):388–396, 2011.

[79] Frederick T Wallenberger, James C Watson, and Hong Li. Glass fibers. *Materials Park, OH: ASM International, 2001.*, pages 27–34, 2001.

[80] Robert W Williams and Karl Herrup. The control of neuron number. *Annual review of neuroscience*, 11 (1):423–453, 1988.

[81] Zhiwei Xu, Yudong Huang, Li Liu, Chunhua Zhang, Jun Long, Jinmei He, and Lu Shao. Surface characteristics of kidney and circular section carbon fibers and mechanical behavior of composites. *Materials Chemistry and Physics*, 106(1):16–21, 2007.

[82] Zhiwei Xu, Jialu Li, Xiaoqing Wu, Yudong Huang, Li Chen, and Guoli Zhang. Effect of kidney-type and circular cross sections on carbon fiber surface and composite interface. *Composites Part A: Applied Science and Manufacturing*, 39(2):301–307, 2008.

[83] HaNa Yu, Marco L Longana, Meisam Jalalvand, Michael R Wisnom, and Kevin D Potter. Pseudo-ductility in intermingled carbon/glass hybrid composites with highly aligned discontinuous fibres. *Composites Part A: Applied Science and Manufacturing*, 73:35–44, 2015.

[84] Z Zhang, K Friedrich, and K Velten. Prediction on tribological properties of short fibre composites using artificial neural networks. *Wear*, 252(7-8):668–675, 2002.

[85] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[86] YT Zhu, JA Valdez, IJ Beyerlein, SJ Zhou, C Liu, MG Stout, DP Butt, and TC Lowe. Mechanical properties of bone-shaped-short-fiber reinforced composites. *Acta materialia*, 47(6):1767–1781, 1999.

[87] Tarek I Zohdi and Peter Wriggers. *An introduction to computational micromechanics*. Springer Science & Business Media, 2008.