Surrogate Constitutive Models with Multi-fidelity Gaussian Processes for Composite Micromodels

Özgür Taylan Turan

TU Delft

Surrogate Constitutive Models with Multi-fidelity Gaussian Processes for Composite Micromodels

by

Ö.T. TURAN

in partial fullfilment of the requirements for the degree of

Master of Science in Civil Engineering

at the **Delft University of Technology**,

to be defended publicly on Thursday August 20, 2020 at 9:00 AM.

Student number:	4780345	
Project duration:	February 20, 2020 – August 20, 2020	
Thesis committee:	Dr. M. A. Bessa,	TU Delft, supervisor
	Dr. ir. F. P. van der Meer,	TU Delft, supervisor
	Dr. I. B. C. M. Rocha,	TU Delft, daily supervisor

Abstract

VARIOUS engineering applications rely on efficient, high performance materials to overcome design challenges. This high performance can be achieved by engineering micro-heterogenous materials also known as composites. Since the behavior of composites relies heavily on micro-scale interactions between different components, modeling macrostructures with fully-represented microscopic geometry is needed. Thus, the standard finite element modeling approach becomes impractical. Computational homogenization, also known as concurrent finite element analysis (FE^2), is a method that is employed to model materials with distinct multi-scaled structure. FE^2 employs the concept of embedding a representative volume element (RVE), at each integration point of the macro-scale problem and obtaining the macroscopic constitutive behavior through homogenization, thus bypassing the need to develop a macro-scale constitutive model. Although it succeeds in up-scaling the microscopic material behavior accurately, this method comes with the major drawback of being computationally expensive due to its nested structure.

Developing methods to bypass the aforementioned computational bottleneck of FE^2 is an ongoing research endeavor. Employing machine learning algorithms to create surrogate constitutive models for microscopic behavior is one possible approach. However, creating surrogate models is not an easy task. A thoroughly collected training set is needed for the surrogate model to be representative. Thus, investigation of surrogate model creation strategies with machine learning while trying to reduce the computational burden of this *offline* training process is a compelling area of study.

Gaussian Process Regression (GPR) is a probabilistic machine learning model. It can be utilized to create surrogate constitutive models effectively in the aforementioned context. Moreover, the computational burden of the training procedure can be decreased by extending the conventional GPR technique into a co-kriging regression with multi-fidelity information (multiGPR). This surrogate modeling strategy reduces the need to collect high-fidelity information by collecting information from a low-fidelity model thoroughly. Thus, enabling accurate training datasets to be created from a less representative, but computationally less taxing models.

In this work, the multiGPR approach is used to construct accurate and efficient surrogates for the behavior of fiber-reinforced composite materials. Training data is obtained from selected RVE configurations that consist of linear-elastic fibers randomly embedded in a matrix that has pressure-dependent plasticity and used to train single and multi-fidelity GPR constitutive models. Both approaches are trained with various combinations of loading scenarios and their prediction capabilities are investigated to represent the training cases in addition to their prediction capabilities under unseen load cases.

Results show that multiGPR model is capable of predicting the response for unseen load cases fairly well for controlled strain paths which allow total control over the feature space, although its strong assumptions regarding the correlation between fidelities. However, when it comes to a more realistic case of a strain path that is not well under control where two different fidelities follow different strain paths multiGPR loses its effectiveness. It is shown that this problem can be solved by imposing a strain path seen by the high-fidelity to the low-fidelity. Moreover, another important issue is found to be the divergence which is surpassed by co-kriging with derivative information. A time comparison for creating a yield stress envelope showed that the multiGPR gets incredibly efficient as the number of boundary value problems to be solved increases.

This work can be used as an introduction to variations of conventional GPR by means of co-kriging with additional information available and as a path to surrogate constitutive modeling of composite micromodels via multi-fidelity information inclusion.

In loving memory of Yusuf Turan

Acknowledgements

This thesis is not the work of an individual, but the collection of people who touched my life. I would like to express my gratitude to those people...

I would like to express my deepest respect and gratitude to my committee members Frans P. van der Meer, Iuri B. C. M. Rocha, and Miguel Bessa, for not only being there for me in the hardships of life and academia but also for pushing the frontiers of human knowledge.

Special thanks to Iuri, his guidance, patience, and work ethic had a huge impact on this work, and me personally. I wish you all the best with your new position in TU Delft.

I would like to thank my dearest friends Ecem Bahcelioglu and Erhan Ersoy for making me feel home, although we were far apart. And thanks to each member of *Oldies but Goldies* for their long-lasting friendship and support throughout my life. Thanks to past and present residents of *Oude Kerkstraat 2B*, in addition to each member of *Friday Nightmares* for providing a family-like environment in a foreign country and keeping my tummy full with good food and beer.

Huge thanks to Ipek Cakir who helped me enormously in the last stretch of this work, by being there no matter the circumstance and supporting me. I appreciate your presence in my life tremendously. I feel blessed and lucky to get to know your kind heart and gentle soul.

Thanks to the Birol family for helping me with the difficulties that lie with residing in a different country. Special thanks to Nahideh Usal, Nesrin Karagulle, the Can family, and the Acar family for making me who I am today as a person.

Sincere thanks to Ecrument and Nilgun Cervatoglu for their endless support. You have inspired me to pursue this path I am on, without you paving the way I have no idea what I will be doing right now. Moreover, thanks to their daughter Idil Su Cervatoglu for giving me a taste of what it feels like to have a sister.

Special thanks to my uncle Oktay Asuman and his beautiful wife Bahar Asuman for being an essential part of my life since I can remember. Moreover, thanks for introducing my lovely cousins Mehmet Can and Ayse to my life.

Last but not least, thanks to my father Tugay and my mother Sukran for their patience, wisdom, and most importantly for their unconditional love.

Özgür Taylan Turan
Den Haag, August 2020

Contents

1 | Introduction

1.1. Background

As the complexity of the encountered problems in every field of engineering increases, the solutions get equally complex. One of these problems can be considered to be the design of high-performance materials for various applications. These types of material are obtained by mixing or combining multiple materials into one in order to enhance the desired properties. These engineered materials inherit their properties from the micro-structure interaction between various constituents. There might be a huge gap between the scale of the interest for the designer and the scale that determines the behavior. This is a major bottleneck for experimental studies. Thus, the usage of numerical modeling approaches for the understanding of material behavior is inevitable.

Modeling high-performance materials in this case composite materials numerically is an ongoing research endeavor for many years. Since the global response is derived from micro-level interactions, the computational modeling of the given material is a hassle for conventional modeling techniques. It is a big computational burden to incorporate all the microscopic interactions to the macroscopic behavior.

The concurrent finite element method (FE^2) is a multiscale approach that is commonly utilized to model these types of materials. However, computational costs associated with this multiscale approach can be exceptionally high. This computational burden is especially exacerbated if the model complexity increases by means of employing a multiphysics approach or increasing the time scale of the method (*e.g.* cyclic loading conditions).

Although there are huge advancements with the computational power increases in the world (*e.g.* quantum computing), its accessibility and applicability problems, makes the search for acceleration methods an interesting research endeavor.

1.2. Mathematical Notation

In this work only matrix notation is utilized. In the light of this information, lower-case letters represent a scalar (*e.g.* h), a bold-faced lower-case letters represents vectors (*e.g.* $\mathbf{h}$) and bold-faced upper-case letters represent matrices (*e.g.* $\mathbf{H}$). Moreover, dot product will be shown without any notation between its constituents ((*e.g.* $\mathbf{Hh} \rightarrow \mathbf{H} \cdot \mathbf{h}$)) for simplicity and the other types of products will be mentioned specifically with their respective symbols.

1.3. Scope and Outline

In this thesis, surrogate constitutive modeling options are investigated with Gaussian Process Regression (GPR) with extensions to the conventional GPR setting. The goal is not only to create surrogate constitutive models that are computationally less taxing but also representative and accurate. Here, the main tool used is co-kriging with multi-fidelity information, which incorporates the information coming from a low-fidelity source and reduces the amount of data needed in a higher-fidelity source. Since the utilized machine learning techniques require an *offline* training, for relatively simple problems the presented methods might not seem to be a computationally viable option. However, with the increasing computational complexity of the multiscale models, the training time will be insignificant. Moreover, the presented machine learning methods have their bottlenecks as well. Nevertheless, this work can be utilized as a stepping stone for investigating surrogate constitutive modeling options.

Thesis structure can be summarized as follows:

- Chapter 2 presents background information regarding the multi-scale modeling and surrogate constitutive modeling method.
- In Chapter 3, a critical study regarding the GPR and its extension with co-kriging with multi-fidelity information and co-kriging with derivative information is presented.
- In Chapter 4, the methodology is presented.
- In Chapter 5, the results of the investigation of multiGPR as a surrogate constitutive model are presented.
- In the final chapter, the main concluding remarks are made and recommendations for future research activities are summarized.

2 | Literature Review

Multiscale methods are powerful techniques to model micro-heterogenous materials through scale transitions [14]. Scale transition between the macroscopic scale to the microscopic scale enables the constitutive modeling to be simpler and incorporation of microscopic failure and degradation processes possible [44]. Multiscale methods can be summarized in two branches, namely domain decomposition, and homogenization. While the former treats the given problem in micro and macroscale in a coupled manner, the latter decouples scales by averaging the smaller scale into an equivalent homogenous medium to utilize its output on a higher scale. Among these techniques, homogenization is the more popular one. There are a variety of homogenization techniques, but due to its accuracy to upscale the microscopic behavior, the computational homogenization technique is pursued [13]. This method is also known as the concurrent finite element method (FE^2).

2.1. Concurrent Multiscale Analysis (FE^2)

This section is dedicated for the brief introduction of the multiscale equilibrium problem and creation of the link between the different scales (Figure 2.1). Interested readers can consult [13, 23, 35, 42–44, 47] for more detailed information.

2.1.1. Macroscale Problem

Let Ω represent the macroscopic domain that the problem is defined in. The displacement field $\mathbf{u}^\Omega$ is sought, resulting from a set of Neuman and Dirichlet boundary conditions applied to the surface Γ which bounds Γ. By assuming no body forces, stress equilibrium in Ω can be expressed as:

$$\mathrm{div}(\sigma^\Omega) = \mathbf{0}, \tag{2.1}$$

where $\mathrm{div}(\cdot)$ represents the divergence operator. Moreover, noting the small strain assumption the strain-displacement relation in Ω is given by:

$$\boldsymbol{\varepsilon}^\Omega = \frac{1}{2}\Big(\mathrm{grad}(\mathbf{u}^\Omega) + \mathrm{grad}(\mathbf{u}^\Omega)^T\Big), \tag{2.2}$$

where $\mathrm{grad}(\cdot)$ represents the gradient operator. Solving Equations 2.1 and 2.2 for $\mathbf{u}^\Omega$ requires a constitutive relation which relates the strains to the stresses. This constitutive relation can be expressed as:

$$\sigma^\Omega = \mathcal{M}(\boldsymbol{\varepsilon}^\Omega, \boldsymbol{\varepsilon}_h^\Omega) \tag{2.3}$$

where ε_h^Ω accounts for the possible strain path dependency. It should be noted that the constitutive relation ($\mathcal{M}$) accounts for material behavior that is coming from a smaller scale in the multiscale context. Thus, it is in a sense a homogenization operator that fuses all processes happening at lower scales [42]. It should be noted that the mentioned constitutive relation can be anything as simple as a linear relation or a complex relation that is intractable.

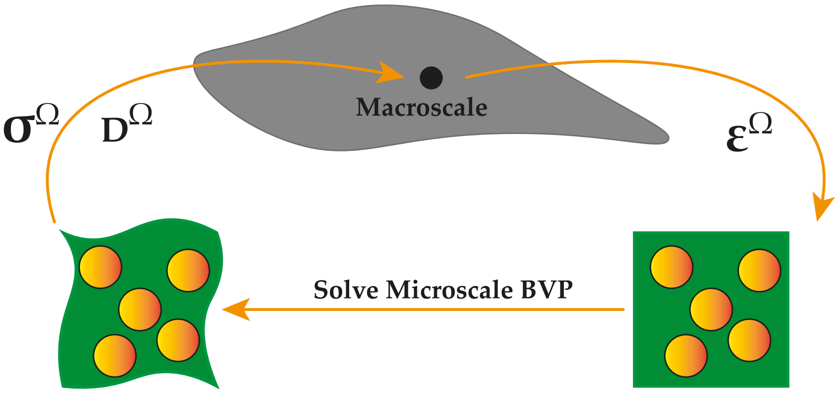

$\boldsymbol{\sigma}^\Omega$
$\mathbf{D}^\Omega$
Macroscale
$\boldsymbol{\varepsilon}^\Omega$
Solve Microscale BVP

Figure 2.1: Schematic representation of the FE^2 method.

Considering the conventional Finite Element Method (FEM) setting, the macroscopic domain Ω is discretized to N_{dof} degrees of freedom. Moreover, the equilibrium problem given in Equation 2.1 is solved by minimization of the force residual $\mathbf{r}^\Omega \in \mathbb{R}^{N_{\text{dof}}}$ as:

$$\mathbf{r}^\Omega = \mathbf{f}^\Omega - \mathbf{f}^\Gamma = \mathbf{0}, \tag{2.4}$$

where $\mathbf{f}^\Omega$ and $\mathbf{f}^\Gamma$ represent the external and internal force vectors respectively. Noting that $\mathbf{N}$ is the matrix that stores the shape functions and $\mathbf{B}$ is the matrix that stores the spatial derivatives of the shape functions, $\mathbf{f}^\Omega$ and $\mathbf{f}^\Gamma$ can be represented as:

$$\mathbf{f}^\Omega = \int_\Omega \mathbf{B}^T \sigma^\Omega d\Omega \quad \text{and} \tag{2.5}$$

$$\mathbf{f}^\Gamma = \int_{\Gamma_\text{f}} \mathbf{N}^T \mathbf{t}^\Gamma d\Gamma, \tag{2.6}$$

where $\mathbf{t}^\Gamma$ is the traction applied at the surface Γ_f. In case the equations given in 2.4 are nonlinear an iterative solution method is needed, which requires the governing equations to be linearized. Thus, the macroscale problem definition is completed by defining a tangent stiffness matrix $\mathbf{K}^\Omega \in \mathbb{R}^{N_{\text{dof}} \times N_{\text{dof}}}$, that is used to compute the displacement update as $\Delta\mathbf{u} = -(\mathbf{K}^\Omega)^{-1}\mathbf{r}^\Omega$:

$$\mathbf{K}^\Omega = \int_\Omega \mathbf{B}^T \mathbf{D}^\Omega \mathbf{B} d\Omega, \tag{2.7}$$

where $\mathbf{D}^\Omega = \partial\sigma^\Omega / \partial\varepsilon^\Omega$ is the material tangent stiffness matrix.

2.1.2. Microscale Problem

Assuming ω to define the microscopic domain of a Representative Volume Element (RVE). An RVE can be contemplated as the volume of heterogenous material that is able to statistically represent the heterogenous materials behavior. [20]. The RVE is an essential part of FE^2, since it is embedded into each and every integration point of the macroscopic model. This paves the way to numerically obtain the constitutive relation, by solving the micro-structural RVE. It should be noted that the RVE selection is a frail task. An RVE should be small enough for computational efficiency, and large enough to represent the underlying microstructure [23]. For the sake of this formulation it is assumed that an appropriate RVE is selected and the domain is bounded by γ_u and γ_f which represent the Dirchlet and Neumann surfaces on the RVE respectively. Again assuming no body forces are present and small strains, stress equilibrium and strain expressions are expressed as:

$$\operatorname{div}(\boldsymbol{\sigma}^{\omega}) = \mathbf{0} \quad \text{and} \tag{2.8}$$

$$\boldsymbol{\varepsilon}^{\omega} = \frac{1}{2}\Big(\operatorname{grad}(\mathbf{u}^{\omega}) + \operatorname{grad}(\mathbf{u}^{\omega})^{T}\Big). \tag{2.9}$$

Again a need for constitutive relation is evident to establish a relationship between $\boldsymbol{\varepsilon}^{\omega}$ and $\boldsymbol{\sigma}^{\omega}$. It should be noted that relatively simple constitutive relations can be utilized at this scale [42]. Thus, the constitutive relations will be specific to the implementation and the intended modeling endeavor.

It can be observed that the problem to be solved is the same as in the case of macroscale problem, but now the domain is changed from Ω to ω.

2.2. Coupling Macroscale and Microscale Problems

It is assumed that the microscopic length-scale is much smaller than the scale at which macroscopic loading varies in space, which means the two different length-scales are detached distinctively ($\omega \ll \Omega$) [13]. Then the two scales can be bridged with the following expression:

$$\mathbf{u}^{\omega} = \boldsymbol{\varepsilon}^{\Omega}\mathbf{x}^{\omega} + \tilde{\mathbf{u}}, \tag{2.10}$$

where $\tilde{\mathbf{u}}$ is a fluctuation term and the $\boldsymbol{\varepsilon}^{\Omega}\mathbf{x}^{\omega}$ is the term accounting for macroscopic contribution. This bridge between scales imposes the macroscopic strain to the specified corners (tying master and slave boundaries) of the embedded RVE. After, the resulting boundary value problem is solved with the information coming from the macroscale an averaging technique Hill-Mandel principle [18] is utilized to recover macroscopic stresses:

$$\boldsymbol{\sigma}^{\Omega} = \frac{1}{\omega}\int\limits_{\omega} \boldsymbol{\sigma}^{\omega} d\omega. \tag{2.11}$$

Then the tangent stiffness is obtained by utilizing the probing processes based on the microscopic stiffness matrix $\mathbf{K}^{\Omega}$ with a procedure given in section 4.2 in [35]. This probing operation is denoted as:

$$\mathbf{D}^{\Omega} = \mathcal{P}(\mathbf{K}^{\omega}), \tag{2.12}$$

which finalizes the coupling operation since both macroscopic tangent stiffness and stress are obtained. By utilizing this information an iterative solution scheme (*e.g.* Newton-Raphson method) is employed to find the solution at each time step.

2.3. Surrogate Modelling

Despite the accuracy gained by incorporating the lowest scale information, obtaining the constitutive relation from the RVE at each integration point creates a computational bottleneck for FE^2. This makes the search for alternative strategies to create constitutive models, without losing the accuracy of the FE^2 another research endeavor.

2.3.1. Mesoscale Models

Conducting the modeling on a higher scale (*e.g.* mesoscale) is a popular approach for modeling composites. The creation of mesoscale constitutive models relies on calibration with microscale simulations. This method allows the use of realistic constitutive models (*e.g.* [24, 38, 40]) for every component of the heterogeneous material, to be homogenized while incorporating that information to a mesoscale [43].

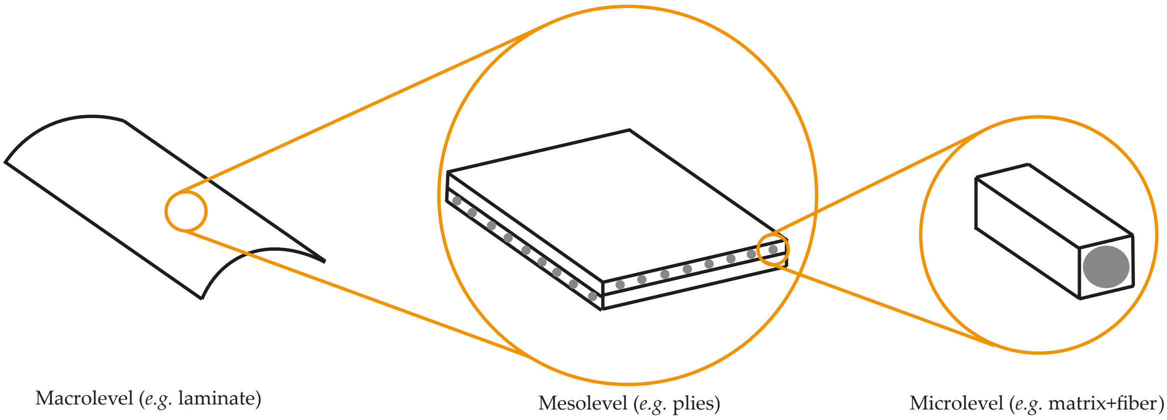

Macrolevel (*e.g.* laminate)
Mesolevel (*e.g.* plies)
Microlevel (*e.g.* matrix+fiber)

Figure 2.2: Visualization of composite laminates for different scales.

Despite its attractive features mesoscale models might fail to accurately represent under general stress state scenarios as encountered in [52]. This is because of the assumptions made during homogenization to limit the parameters to be calibrated [43].

More information regarding this modeling technique can be found in [4, 43, 50, 51, 53].

2.3.2. Reduced Order Modeling

Model order reduction (MOR) technique is another way to create surrogate models that reduces the computational complexity of the problem. This method might prove useful to circumvent the bottleneck of FE^2. The main aim of reduced order modeling is to decrease the computational burden of the existing boundary value problems. This is achieved by simply decreasing the number of degrees of freedom by projecting the equilibrium equation onto a reduced space [44]. In other words it is assumed that the solution of a given macroscopic problem can be found in a lower-dimensional manifold. Thus, shifting the computational efforts to finding the lower-dimensional manifold representative enough. This is achieved by collecting snapshots (offline training) and utilizing dimensionality reductions techniques (*e.g.* proper orthogonal decomposition (POD) or proper generalized decomposition (PGD) [5]). The reduced order solution is represented as:

$$\mathbf{\Psi} = [\psi_1 \quad \cdots \quad \psi_n], \tag{2.13}$$

where $\mathbf{\Psi} \in \mathbb{R}^{N_{dof} \times N_{red}}$ and every component of Ψ is an orthonormal basis vectors that represents the global modes of displacement. It should be noted that $N_{dof} \gg N_{red}$. Then the full order solution can be obtained by:

$$\mathbf{u}^{\omega} = \mathbf{\Psi}\boldsymbol{\alpha}, \tag{2.14}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{N_{red}}$ represent the linear combination latent variables. Then the given boundary value problem simplifies to solving for $\boldsymbol{\alpha}$. In order to achieve this residual for microscopic problem ($\mathbf{r}^{\omega} = \mathbf{f}^{\omega} - \mathbf{f}^{\gamma} = \mathbf{0}$) is constrained to be in a reduced space by Galerkin projection:

$$\mathbf{\Psi}\mathbf{r}^{\omega} = \mathbf{0}. \tag{2.15}$$

Then, the reduced internal force vector and stiffness matrix is found to be:

$$\mathbf{f}_r^{\omega} = \mathbf{\Psi}^T\mathbf{f}^{\omega} \quad \text{and} \tag{2.16}$$

$$\mathbf{K}_r^{\omega} = \mathbf{\Psi}^T\mathbf{K}^{\omega}\mathbf{\Psi}. \tag{2.17}$$

Moreover, by utilizing the Empirical Cubature Method (ECM) another level of reduction is possible. This method reduces the number of integration points that the constitutive model is computed by means of minimizing the error between original and approximated internal forces [44]. These methods perform remarkably well for linear models. However, for nonlinear time- and history-dependent models causes the reduced solution space to be extremely large to maintain a feasible level of accuracy [5, 30, 33, 45].

More information regarding these acceleration techniques can be found in [2, 15–17, 44, 45].

2.3.3. Neural Networks (NN)

Another encouraging approach to accelerate FE^2 method is to construct surrogate models to be a substitution for the micro models embedded at each integration point of the macroscale problem. To achieve this instead of a physics-based approach purely data-driven approach might be utilized.

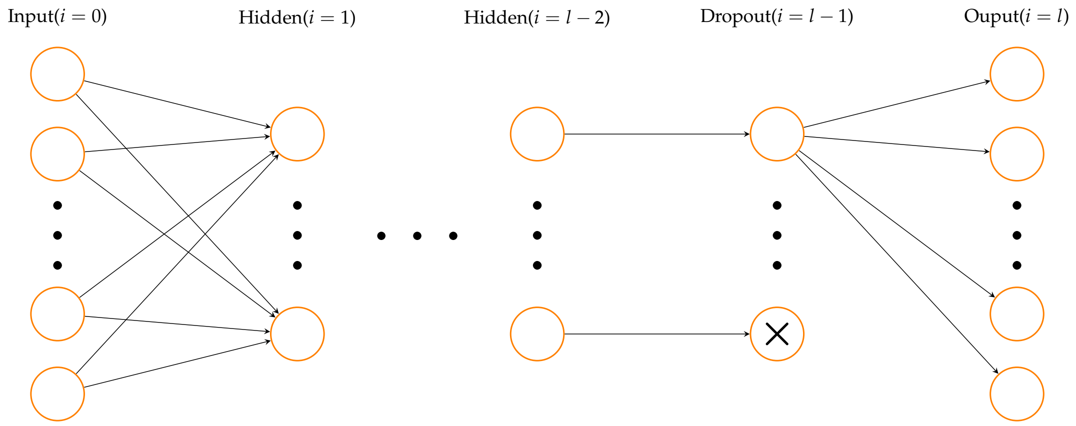


Figure 2.3: Basic architecture of a feed-forward NN with a dropout layer before the output layer (excluding the bias terms).

In order to understand how it is possible to utilize data-driven approaches as surrogate models, one has to think about what a material represents computationally. In a computational sense, a material can be defined as a black box that spits out stresses, under the imposed strains or displacements.

Then, this blackbox shown in Figure 2.4 can be in this case a NN shown in Figure 2.3 that is trained on the material behavior and can be seen in Figure 2.3. Then, noting that the neuron values in the input layer of the NN as strain values ($\mathbf{a}_0 = \varepsilon^{\Omega}$), the constitutive model can be defined as:

$$\hat{\sigma} = \mathcal{S}_{\text{NN}}(\varepsilon^{\Omega}, \mathbf{W}), \tag{2.18}$$

where the values in the output layer represents the approximated stress values ($\hat{\sigma} = \mathbf{a}_l$) and $\mathbf{W}$ represents the weights of the trained NN. Unlike other methods presented till this point $\mathbf{W}$ has no direct physical meaning, it consist of multiple weights obtained through fitting the observations obtained from the snapshots collected from the microscale model which is expressed as:

$$\mathbf{W} = \text{argmin}_{\bar{\mathbf{W}}} \sum_{i \in \boldsymbol{\lambda}} ||\hat{\sigma}_i(\varepsilon_i^{\Omega}, \bar{\mathbf{W}}) - \sigma_i^{\Omega}(\varepsilon_i^{\Omega})||^2, \tag{2.19}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{2n_{\varepsilon} \times N_{\text{iter}}}$ is the snapshot matrix that contain the solutions from N_{iter} microlevel boundary value problems and store them as σ^{Ω}-ε^{Ω} pairs.

Figure 2.4: Vague representation of a computational material model.

Then, the tangent stiffness can be found by calculating the Jacobian of the NN ($\mathbf{J} = \mathbf{D}^{\Omega}$) making the formulation ready for the NN to be utilized as a surrogate model. Because of their multi-layered structure interpretation of the results obtained by NNs might not be traced back and reasoned. Moreover, these models are blind to their errors. This makes their utilization in a robust framework risky. Furthermore, although neural networks are fast and can be retrained to incorporate new information, their extrapolation capabilities for the unseen states are limited [43]. Thus, it is interesting to investigate other data-driven methods for creating constitutive models.

More information regarding the use of NNs as surrogate constitutive models can be found in [5, 33, 43]. Detailed formulations for the NNs can be found in Appendix B.

2.3.4. Gaussian Processes

As in the case of NNs another type of machine learning technique with a Bayesian foundation, called Gaussian Processes, can be utilized to create surrogate constitutive models as well. This method relies on conditioning of the prior ($\mathcal{N}(0,k(\boldsymbol{x},\boldsymbol{x}'))$) assumptions with the input points in order to get a posterior ($\mathcal{N}(\boldsymbol{m},\boldsymbol{S})$) distribution over the functions. Noting that input values are the strains ($\boldsymbol{x} = \boldsymbol{\varepsilon}^{\Omega}$), the surrogate constitutive model can be represented as:

$$\hat{\sigma} = \mathcal{S}_{\text{GP}}(\boldsymbol{\varepsilon}^{\Omega},\boldsymbol{\eta}), \tag{2.20}$$

where the predicted values represent the stress values ($\hat{\sigma} = y_*$) and $\boldsymbol{\eta}$ represents the collection of hyper-parameters of the Gaussian Process that maximizes the marginal likelihood. This process is refered as training and expressed as:

$$\boldsymbol{\eta} = \text{argmax}_{\bar{\boldsymbol{\eta}}} p(\boldsymbol{\sigma}_{\lambda}^{\Omega}|\boldsymbol{\varepsilon}_{\lambda}^{\Omega},\bar{\boldsymbol{\eta}}), \tag{2.21}$$

where subscript λ is the indicator of the collection values obtained from the solutions of N_{iter} microlevel boundary value problems. Moreover, similar to the case of NNs, constituents of $\boldsymbol{\eta}$ has no direct physical meaning. In other words this method is data-driven as well.

The tangent stiffness matrix can be obtained easily by differentiation of the posterior of the given Gaussian Processes, which completes the surrogate model constitutive model creation.

Statistical foundations of this method enable the uncertainty bounds to be available for use, which makes it a practical method for utilization as a surrogate model. Rocha et al. utilizes this method in [42], to construct the surrogate model *online* and use uncertainty values as triggers to introduce new points from the microlevel models.

The focus of this work will be on Gaussian Process Regression (GPR) and its extensions that enhance its predictive capabilities. A detailed discussion of the GPR is presented in Chapter 3.

3 | Gaussian Process Regression

A deluge of data is present and an incredible amount of new data is generated every day. Much of this data is constantly being collected and stored. The fundamental goal of data collection and storage is to make better decisions. In general machine learning (ML) can be defined as a set of methods that can automate that decision making by pattern recognition and later on, predict future data [34].

There are mainly two types of ML, supervised and unsupervised learning. Unsupervised learning defines a set of data problems in which the input data ($\mathcal{D}_{in} = \{(\boldsymbol{x}_i)|i = 1,...,N\}$) is not related to the output data ($\mathcal{D}_{out} = \{(y_i)|i = 1,...,N\}$) or there is no output data. In this case, the main objective is to get information out of the data by finding patterns [46]. Some of the most well known unsupervised ML applications are clustering, discovering latent spaces, and matrix completion. Supervised learning on the other hand, given a training set of inputs and outputs ($\mathcal{D} = \{(\boldsymbol{x}_i, y_i)|i = 1,...,N\}$) , tries to find a mapping between inputs and outputs [46]. When the outputs are continuous this mapping procedure is called regression and classification otherwise.

Supervised learning models for regression can be classified in many ways. The most common way of classifying these types of models is through the decision to fix the number of parameters. The model has fixed parameters that do not change with the amount of training data are called parametric and ones that have an infinite number of parameters (parameters number increases with the number of training data) are called non-parametric models [34]. Furthermore, parametric models absorb information coming from the training data to its posterior parameter distribution, whereas the non-parametric models use it directly in its posterior [41]. Although parametric models are faster, it is important to note that fixing the number of parameters becomes a bottleneck for the information gained with the addition of new data. This issue does not exist in non-parametric models.

Gaussian Process (GP) is a formalism with Gaussian distribution as its core assumption [3]. GP's can be utilized as a non-parametric supervised learning model [41]. GP models are utilized for both classification and regression. In this work, only the regression part will be dealt with.

Gaussian Process Regression (GPR) is a non-parametric probabilistic model that specifies a distribution over functions [26]. This means that when function $y(\boldsymbol{x})$ is evaluated at arbitrary input points $\boldsymbol{x}_1,...,\boldsymbol{x}_N$, these evaluations jointly have a Gaussian distribution [7]. Other regression tools in ML provide only a deterministic output, but GPR's probabilistic assumptions enable this model to give reliable estimates of uncertainty [3]. This feature makes GPR a powerful tool for ML.

Computer codes are utilized in engineering and science in order to model and study physical, chemical, and biological systems in nature [27]. The aim of developing these types of codes stems from the fact that some natural systems are either too expensive or impossible to describe experimentally. The basic structure of a computer code involves running the code for predefined input values and get an output in the hopes of learning something about a system [21]. GPR can be utilized to construct a surrogate model with the outputs obtained from a specific code. In some cases, computer code might be present for varying complexity levels.[1] In the context of Finite Element Method (FEM) different sizes of mesh could represent different levels of complexity for output accuracy. Another example of that could be the complexity of the mathematical model behind the code. For instance, if the effect of water concentration is discarded in [44] the computer code generated will represent less information regarding the physical system, thus making it less accurate. Nevertheless, less accurate low complexity models can improve the computational burden of the given system, making output generation orders of magnitude faster. Thus, the utilization of these low complexity models plays an important role in the computational community.

A GPR[2] model does not have the ability to make predictions at the highest fidelity with the data coming from different levels of output. To be able to predict at the highest-fidelity with the information coming from multi-fidelity observations, multi-fidelity Gaussian Process Regression (multiGPR) can be utilized.

The content of this chapter includes information regarding the single-fidelity Gaussian Process Regression (GPR) and Multi-Fidelity Gaussian Process (multiGPR). For the sake of brevity, only the most important aspects will be discussed. Interested readers should consult [7, 34, 41] for more in-depth information regarding GPR and [21, 27] for more in-depth information regarding multiGPR.

It should be noted that because of the wide-spread use of this method in many domains of science and engineering, there are many types of notations but for the sake of simplicity the notation used by C. E. Rasmussen and Christopher K.I. Williams [41] will be adopted here.

3.1. Single Fidelity Gaussian Process Regression (GPR)

Consider the standard regression problem with Gaussian noise given by,

$$y = f(\boldsymbol{x}) + \varepsilon, \tag{3.1}$$

where $\boldsymbol{x}$ is input vector, $f(\boldsymbol{x})$ is function value and y is target value. ε is the Gaussian noise defined as $\mathcal{N}(0,\sigma_n^2)$. Given the observed data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) | i = 1,...,N\}$, the aim is to find $f(\boldsymbol{x})$ such that for any given test point $\boldsymbol{x}_*$ the corresponding y_* is predicted accurately.

In order to find $f(\boldsymbol{x})$, GPR requires defining a prior over functions [26]. Then, this prior is conditioned on the observed data to obtain the posterior over functions [34]. Although it seems challenging to define distributions over functions, it is only needed at a finite set of function values [34]. An example for prior and posterior can be seen in in Figures 3.1 and 3.2. For the given figures , the prior is defined as $\mathcal{D} = \emptyset$ (no observations) and the posterior is defined as $\mathcal{D} = (\boldsymbol{x}_i, y_i) | i = 1,...,4\}$ (four observations). Moreover, the filled yellow space represents the 95% confidence interval of the

[1] In this work complexity level and fidelity will be utilized interchangeably

[2] From this point on the GPR abbreviation will be utilized for single-fidelity regression.

prediction achieved by the GPR.

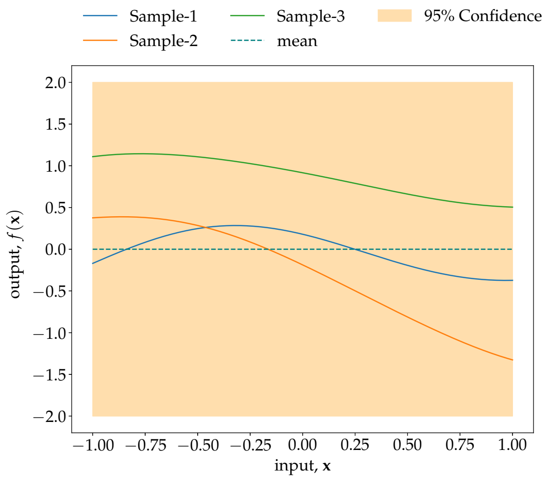


Figure 3.1: 3-Samples Drawn from GP Prior

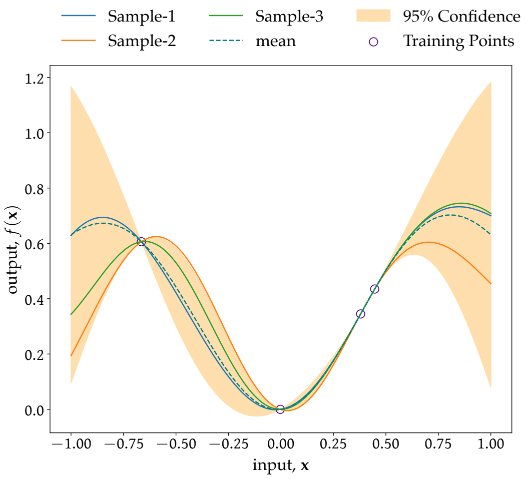


Figure 3.2: 3-Samples Drawn from GP Posterior

As in the case of Gaussian distribution, GPs can be represented with a mean and covariance, both of which are functions of the space that the process takes place (the input space) [26]. The mean function is expressed as $m(\boldsymbol{x})$ and covariance function is expressed as $k(\boldsymbol{x},\boldsymbol{x}')$. Thus, a GP $f(\boldsymbol{x})$ can be represented completely as:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}) = 0, k(\boldsymbol{x},\boldsymbol{x}')). \tag{3.2}$$

There are two important assumptions regarding GPs in this work. The first assumption is that $m(\boldsymbol{x})$ is zero without loss of generality. The second assumption is that the covariance function is generally constrained to result in positive definite matrices.[3] The zero-mean assumption on the prior can be seen in Figure 3.1. Moreover, Figure 3.2 shows how the prior dominates the prediction away from the observed data since the right end of the prediction tends to go to zero with high variance, as in the case of Figure 3.1.

3.1.1. Kernel

The aforementioned distribution over functions is achieved only with the specification of the kernel since the mean function is assumed to be zero. This makes the kernel a crucial part of the prediction. A basic assumption regarding prediction is that neighboring points are likely to have similar targets. In the GP setting, this nearness is defined by the kernel [41]. In other words, the main function of the kernel is to determine the likely functions from GP prior [10]. In this work, the widely used squared exponential (SE) kernel is adopted. It has the form:

$$k(\boldsymbol{x}_p,\boldsymbol{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}||\boldsymbol{x}_p - \boldsymbol{x}_q||^2\right), \tag{3.3}$$

[3] This feature of the covariance functions makes them valid Mercer kernels [26]. It should be noted that from now on covariance functions can be referred to as kernels.

with σ_f^2 as variance and l as length-scale terms. These parameters are called hyper-parameters. The effect of hyper-parameters can be observed in the following figures for which examples are drawn randomly from the GP prior. It is an important aspect of the SE kernel to be infinitely differentiable, this property ensures a very smooth behavior [41].

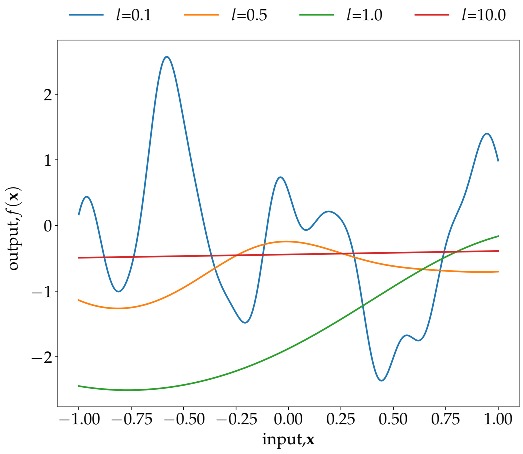


Figure 3.3: Effect of changing length-scale for $\sigma_f^2 = 1$

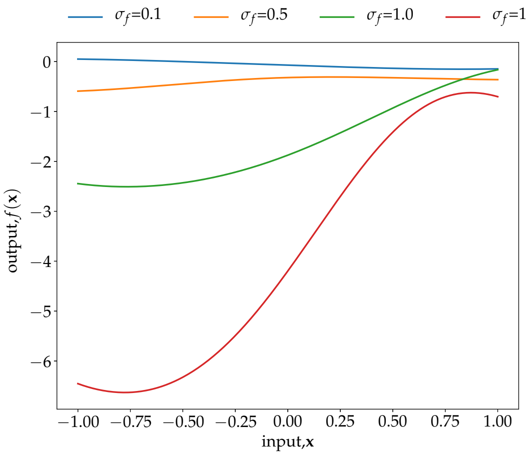


Figure 3.4: Effect of changing variance for $l = 1$

As can be seen from Figure 3.3, the length-scale determines the width of the kernel [10]. Moreover, from Figure 3.4 it can be concluded that σ_f^2 is just a scale factor that determines how much the drawn function can go away from the mean.

There are many different kernel types with different properties. Moreover, there are ways to create new kernels. Other types of kernels and ways to create new ones will not be discussed here. The interested reader can find more information regarding kernels in [10] and on Chapter 4 of [41].

3.1.2. Prediction

As can be seen from Figure 3.1, drawing functions from the prior is of little interest, since it has all the functions that are possible in the given input space. Thus, information coming from observed data, $\mathcal{D} = \{(x_i, f_i) | i = 1, ..., N\}$ should be incorporated such that test function values $\boldsymbol{y}_*$ evaluated at test locations $\boldsymbol{X}_*$ and $\mathcal{D}$ are jointly distributed. This can be expressed as:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2 \boldsymbol{I} & k(\boldsymbol{X},\boldsymbol{X}_*) \\ k(\boldsymbol{X}_*,\boldsymbol{X}) & k(\boldsymbol{X}_*,\boldsymbol{X}_*) + \sigma_n^2 \boldsymbol{I} \end{bmatrix}\right), \tag{3.4}$$

where $\boldsymbol{X}$ represents the whole training set of $\boldsymbol{x}$'s in vector form and the Gaussian noise term σ_n^2 is added as another hyper-parameter to $k(\boldsymbol{X},\boldsymbol{X})$ in order to account for the Gaussian noise of Equation 3.1 introduced to the observations and added to $k(\boldsymbol{X}_*,\boldsymbol{X}_*)$ to account for the noise in the test data. The effect of inclusion of the Gaussian noise term in the prediction can be seen in Figures 3.5 and 3.6 where observed data is collected from a noisy version of $f(\boldsymbol{x}) = \boldsymbol{x}\sin(3\boldsymbol{x})$, where $\boldsymbol{x}$ is on the interval $[-2,2]$. When GP is utilized over a noisy observation set, possible overfitting seen in Figure 3.5 is prevented with the inclusion of σ_n^2 to the same data set. In other words σ_n^2 allows the posterior functions to be disconnected from the observed data to an extent. Moreover, σ_n^2 has the role to increase the stability of the covariance matrix internally. When the points are too close to each other covariance matrices tend to lose their positive definite feature, resulting in ill-conditioned matrices. Thus, the addition of noise to the diagonal of the kernels results in better conditioned matrices.

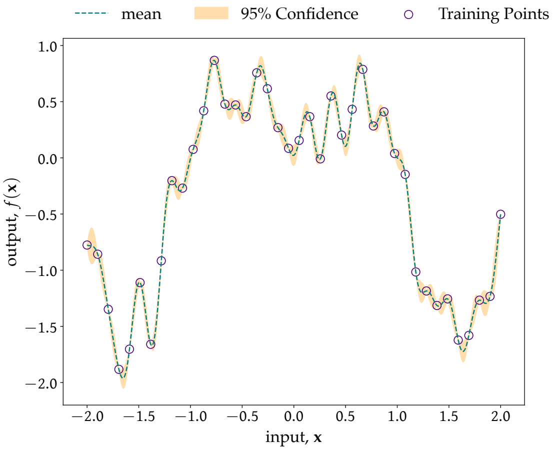


Figure 3.5: Effect of Gaussian Noise (without σ_n^2 inclusion)

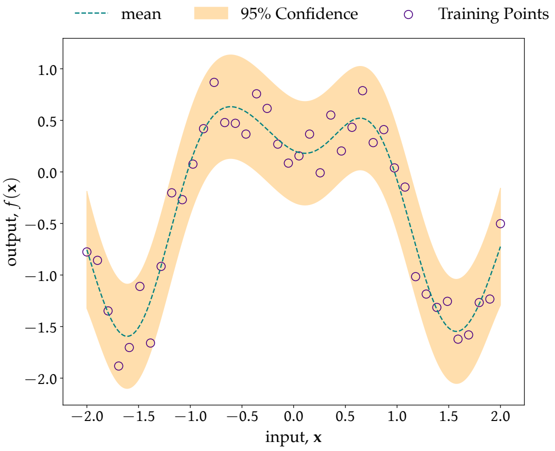


Figure 3.6: Effect of Gaussian Noise (with $\sigma_n^2 = 0.2$ inclusion)

The marginalization property of GPs enables us to marginalize the Gaussian distribution by taking the relative portion of the covariance matrix [41]. Thus, the conditioned predictive distribution of the new targets $\boldsymbol{y}_*$ is given by:

$$p(\boldsymbol{y}_*|\boldsymbol{y}) \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{S}), \tag{3.5}$$

where the mean and covariance of the posterior are given by:

$$\boldsymbol{m} = k(\boldsymbol{X}_*, \boldsymbol{X})\left(k(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}, \tag{3.6}$$

$$\boldsymbol{S} = k(\boldsymbol{X}_*, \boldsymbol{X}_*) - k(\boldsymbol{X}_*, \boldsymbol{X})\left(k(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}\right)^{-1} k(\boldsymbol{X}, \boldsymbol{X}_*) + \sigma_n^2 \boldsymbol{I}. \tag{3.7}$$

In a simpler format, the expectation and variance for a single test point $\boldsymbol{x}_*$ can be expressed as:

$$\mathbb{E}[y_*|\boldsymbol{x}_*] = \boldsymbol{k}_*^T \left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right)^{-1} \boldsymbol{y}, \tag{3.8}$$

$$\mathbb{V}[y_*|\boldsymbol{x}_*] = \boldsymbol{k}_{**} - \boldsymbol{k}_*^T \left(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right)^{-1} \boldsymbol{k}_* + \sigma_n^2, \tag{3.9}$$

where $\boldsymbol{k}_* = k(\boldsymbol{X}, \boldsymbol{x}_*) \in \mathbb{R}^{N \times 1}$, $\boldsymbol{k}_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*) \in \mathbb{R}$ and $\boldsymbol{K} = k(\boldsymbol{X}, \boldsymbol{X}) \in \mathbb{R}^{N \times N}$.

3.1.3. Model Selection

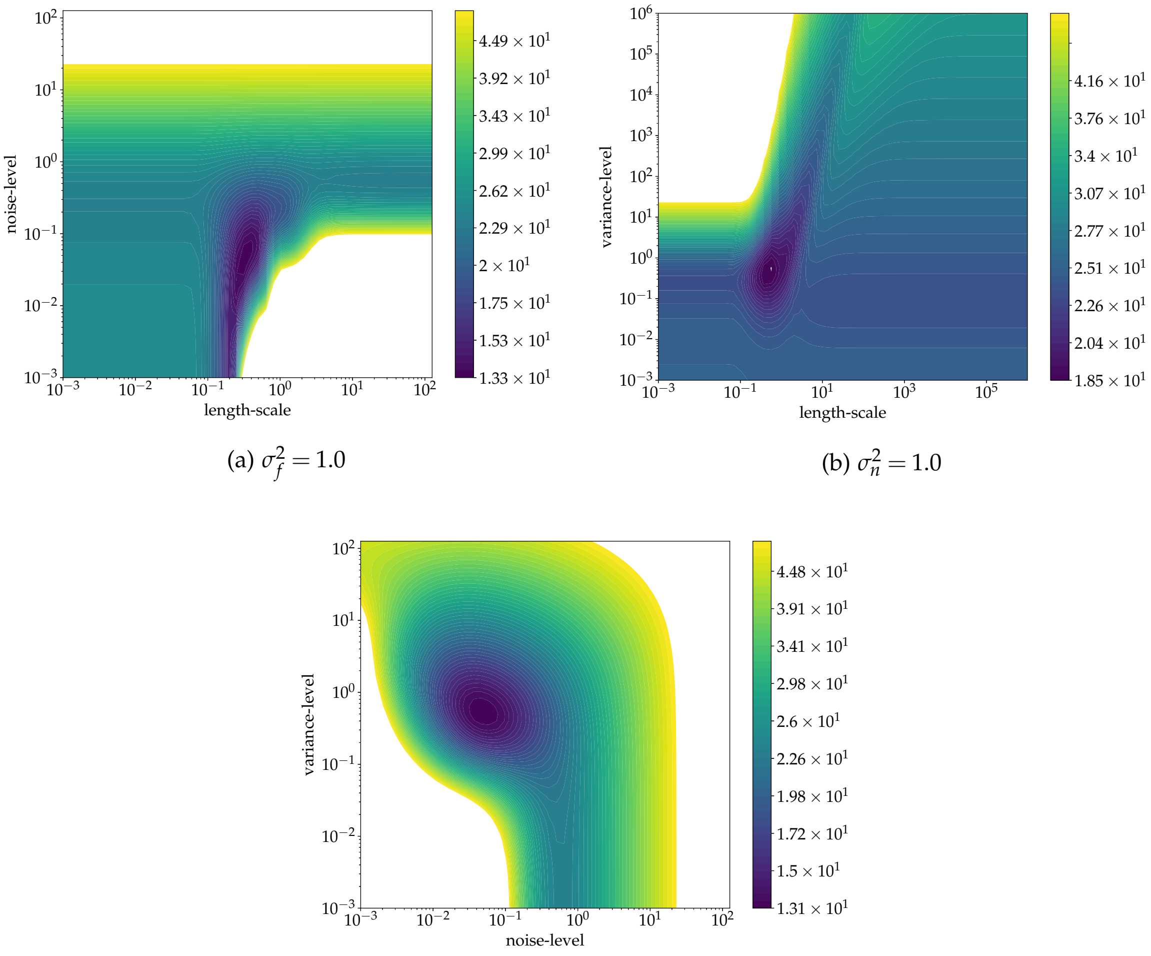
In Sections 3.1.1 and 3.1.2 a set of hyper-parameters was introduced. In this work, we refer to the determination of optimum hyper-parameter values as model selection.

Although for some easy problems it is easy to set predefined values of these parameters, in most cases this is a hard task. Thus, the need for a systematic model selection procedure is apparent

There are several methods presented in [41], Bayesian model selection, cross-validation, and marginal likelihood maximization.

Bayesian model selection may require computationally heavy simulation techniques (*eg.* Markov chain Monte Carlo), since the analytical tractability of the formulations is not guaranteed. This computational burden makes this method unattractive.

The core idea behind cross-validation is to split the training data into two disjoint sets and train the model in one set and validate it with the other set. Although the biggest drawback of this method is the usage of some portion of the data set for training, there are some ways to bypass these issues. Especially in the GPR setting, there are computational shortcuts [41].



(a) $\sigma_f^2 = 1.0$

(b) $\sigma_n^2 = 1.0$

(c) $l = 1.0$

Figure 3.7: LML Topologies for different combinations of hyper-parameters

In this work, maximization of marginal likelihood approach will be utilized. The marginal likelihood is obtained by averaging the probability of the model to reproduce $\boldsymbol{y}$ over all values of $\boldsymbol{f}$ as:

$$p(\boldsymbol{y}|\mathbf{X}) = \int p(\boldsymbol{y}|\boldsymbol{f}, X)p(\boldsymbol{f}|X)d\boldsymbol{f}. \tag{3.10}$$

It should be noted that marginal likelihood is only dependent on the hyper-parameters. Thus, by maximizing the marginal likelihood, a set of hyper-parameter values that maximizes the probability that the GP model generates the training data $\boldsymbol{y}$. In the GP setting the objective function to be maximized is the log marginal likelihood (LML) function and is expressed as:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) = \underbrace{-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{R}^{-1}\boldsymbol{y}}_{\text{data-fit}} - \underbrace{\frac{1}{2}\log|\boldsymbol{R}|}_{\text{comp. pen.}} - \underbrace{\frac{N}{2}\log 2\pi}_{\text{norm. const.}}\,, \tag{3.11}$$

where $\boldsymbol{R} = k(\boldsymbol{X},\boldsymbol{X}) + \sigma_n^2\boldsymbol{I}$ is the covariance matrix for the noisy target observations $\boldsymbol{y}$ and N is the number of observations. It should be noted that in the given LML expression terms represent, data-fit, complexity penalty (comp. pen.), and normalization constant respectively.

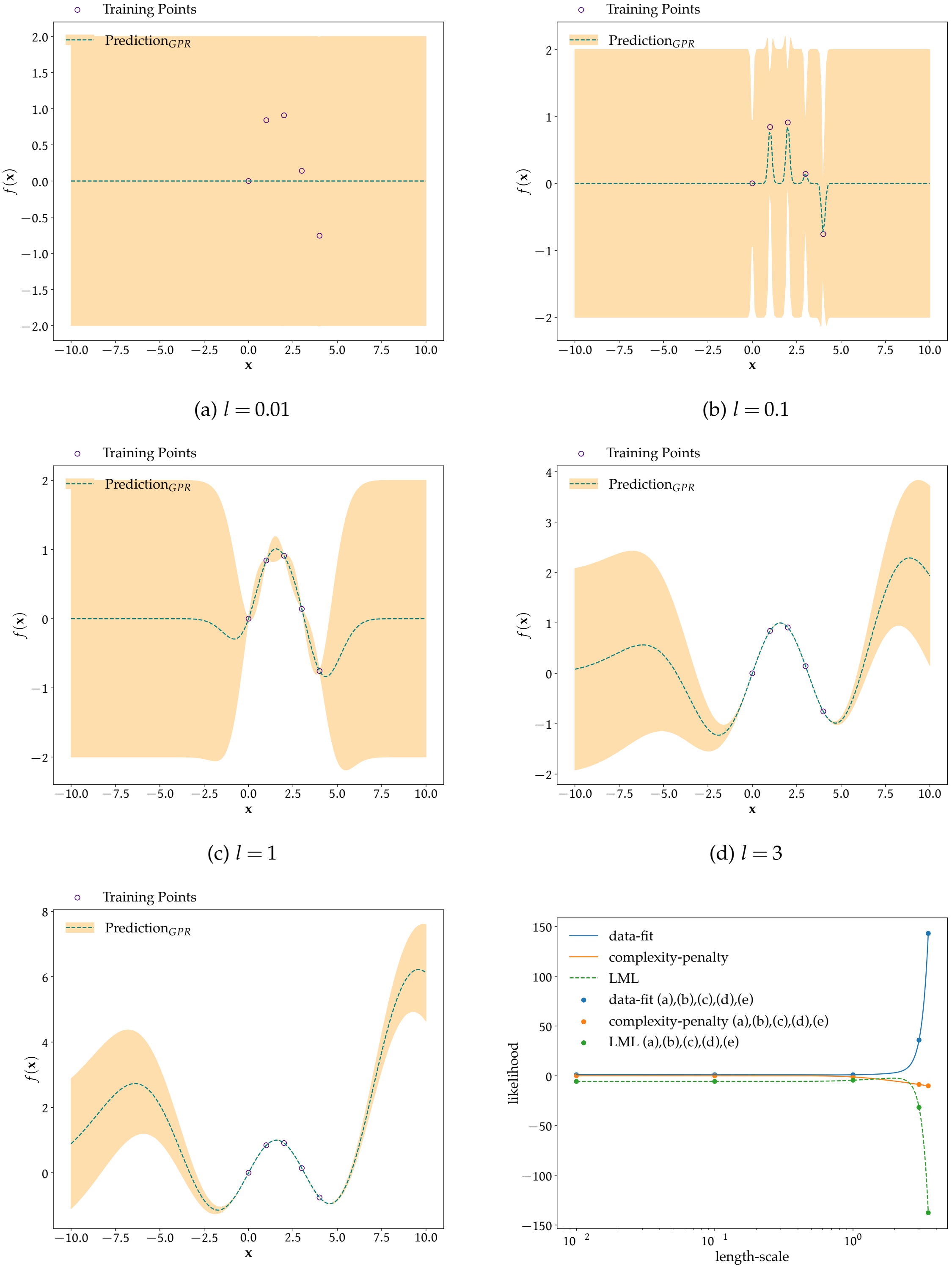
Understanding the terms in the LML is vital for the interpretation of results and understanding the behavior of LML. Thus, a series of GPR prediction results have been shown in Figure 3.8 for different length-scale values and the corresponding change with data-fit and complexity penalty terms in the LML. As can be seen from Figure 3.8f the data-fit term increases as the length-scale increases then the model gets less flexible. This can be observed in the Figures 3.8a to 3.8d since with the very small length-scale GPR prediction gives a single function valued mean with variance and as the length-scale increases data-fit term starts to dominate the behavior. It should be noted that the normalizing constant has no significant contribution to the overall behavior that is why it is left out from this discussion.

Another important aspect to be noted is the prediction results shown in Figure 3.8a. It can be easily misinterpreted as an over-fitting prediction shown in Figure 3.5, but this is not the case. The only conclusion that can be drawn is that with the given length-scale, the mean is somewhere near the of the training points value. Furthermore, the interaction of data-fit term and complexity-penalty and their effect on the LML terms can be observed in 3.8f. The maximum LML value is determined by the balance between data-fit and complexity penalty terms at a value between length-scale 2 and 3. Thus, it can be concluded that all the hyper-parameters are optimized according to the relation between data-fit and complexity-penalty terms in LML. However, when other hyper-parameters are added to the optimization mix it is not as straight forward to find the maximum of LML.

The topology of LML for multiple hyper-parameter values have been plotted in Figures 3.7a, 3.7b, and 3.7c for the previous problem shown in Figure 3.6. Specifically looking at Figures 3.7a and 3.7c results indicate that there is a possibility that the whole data can be represented as just noise for high noise levels since for high noise levels other parameters become irrelevant. This means there are multiple modes of representation for a given set of data.

The marginal likelihood maximization method for model selection relies on a numerical optimization procedure to find optimum values for the hyper-parameters. It is suggested that a gradient based optimizer should be utilized since the numerical overhead of calculating derivatives is small [41]. The needed partial derivatives of LML with respect to set of hyper-parameters ($\boldsymbol{\theta}$) are expressed as:

$$\frac{\partial}{\partial\theta_j}\log p(\boldsymbol{y}|\boldsymbol{X}) = \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \boldsymbol{R}^{-1})\frac{\partial\boldsymbol{R}}{\partial\theta_j}\right), \tag{3.12}$$



(a) $l = 0.01$

(b) $l = 0.1$

(c) $l = 1$

(d) $l = 3$

(e) $l = 3.5$

(f) Likelihood terms and LML vs length-scale

Figure 3.8: Investigating LML constituents with changing length-scale for $f(\boldsymbol{x}) = \sin(\boldsymbol{x})$ in one dimension with $\mathcal{D} = \{(\boldsymbol{x}_i, f_i) | i = 1, ..., 5\}$. Noting the fixed hyper-parameters $\sigma_f^2 = 1.0$ and $\sigma_n^2 = 0$

where $\boldsymbol{\alpha} = \boldsymbol{R}^{-1}\boldsymbol{y}$.

Even with the introduction of gradient information, it is not guaranteed that any selected optimization tool will find the optimum LML. This is due to the fact that there might be multiple local optima in the LML landscape [41]. The found local maximum might just another possible way to explain the same observations. These local optima can be avoided with multiple restarts of the numerical optimizer with different values of hyper-parameters in the given landscape to some extent. It should be noted that in this work the BFGS optimizer [11] will be utilized with multiple restarts.

3.1.4. Derivative Observations

From the fact that differentiation is a linear operator, it follows that derivative of a GP is another GP [41]. This property enables sufficiently smooth kernels to have well-defined derivative predictions as well as its function values. Thus, the posterior mean of the derivative observations can be found by differentiating Equation 3.8 with respect to $\boldsymbol{x}_*$:

$$\frac{\partial}{\partial \boldsymbol{x}_*}\mathbb{E}[y_*|x_*] = \left(\frac{\partial}{\partial \boldsymbol{x}_*}k(\boldsymbol{x}_*,\boldsymbol{X})\right)\left(\left(\boldsymbol{K}+\sigma_n^2\boldsymbol{I}\right)^{-1}\boldsymbol{y}\right). \tag{3.13}$$

The first term in Equation 3.13 for the SE kernel is expressed as:

$$\frac{\partial}{\partial \boldsymbol{x}_p}k(\boldsymbol{x}_p,\boldsymbol{x}_q) = -\frac{\sigma_f^2}{l^2}(\boldsymbol{x}_p-\boldsymbol{x}_q)k(\boldsymbol{x}_p,\boldsymbol{x}_q), \tag{3.14}$$

where $k(\boldsymbol{x}_p,\boldsymbol{x}_q)$ is given in Equation 3.3.

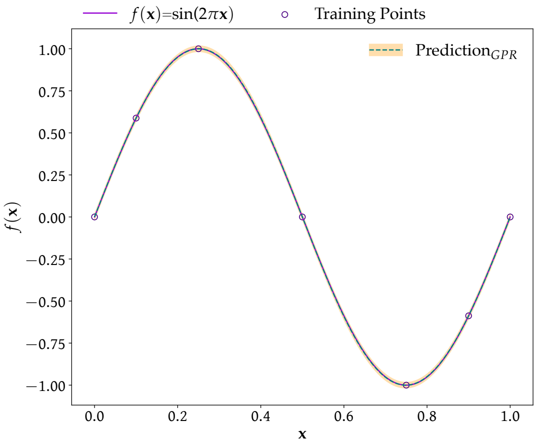


Figure 3.9: GPR prediction for $f(x) = \sin(2\pi x)$

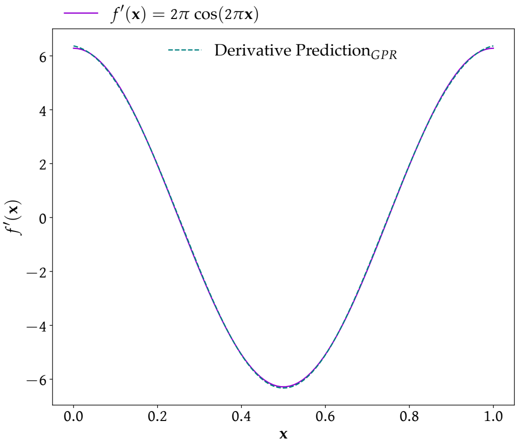


Figure 3.10: GPR prediction for $f'(x) = 2\pi\cos(2\pi x)$

In Figures 3.9 and 3.10 predictions of $f(x) = \sin(2\pi x)$ and its derivative $f'(x) = 2\pi\cos(2\pi x)$ can be observed with $\mathcal{D} = \{(x_i, y_i)|i = 1,...,4\}$. It can be seen that even without inducing derivative observations, GPR is capable of predicting its own derivatives at given points.

In some cases the derivative information of the objective function is readily available for no additional cost [48]. In those cases GPR predictions can be improved by including derivative information in the conditioning as well. In other words, now the joint probability of function values and their derivatives is sought.[4] To achieve this additional terms in the covariance matrix are needed to incorporate the correlation between function and derivative observations and among derivative observations [39]. These expressions are given by:

$$\mathrm{cov}(\boldsymbol{y}'_p, y_q) = \frac{\partial}{\partial \boldsymbol{x}_p} k(\boldsymbol{x}_p, \boldsymbol{x}_q), \tag{3.15}$$

$$\mathrm{cov}(\boldsymbol{y}'_p, \boldsymbol{y}'_q) = \frac{\partial^2}{\partial \boldsymbol{x}_p \partial \boldsymbol{x}_q} k(\boldsymbol{x}_p, \boldsymbol{x}_q). \tag{3.16}$$

For the SE kernel covariance between derivative and function values is presented in Equation 3.14. Moreover, for the same kernel the covariance between derivative observations can be expressed as:

$$\frac{\partial^2}{\partial \boldsymbol{x}_p \partial \boldsymbol{x}_q} k(\boldsymbol{x}_p, \boldsymbol{x}_q) = \frac{\sigma_f^2}{l^2} \left(\boldsymbol{I} - \frac{1}{l^2} (\boldsymbol{x}_p - \boldsymbol{x}_q)(\boldsymbol{x}_p - \boldsymbol{x}_q)^T \right) k(\boldsymbol{x}_p, \boldsymbol{x}_q). \tag{3.17}$$

The behavior of the SE kernel for Equations 3.3, 3.15 and 3.16 can be observed in Figure 3.11. As expected correlation between function values decreases monotonically as the distance increases. However, this is not the case with the correlation between function values and its derivatives as the absolute value of correlation increases to a maximum at a distance of l. This observation is fathomable as the derivative observation affects the prediction in a certain vicinity, in other words the next wiggle that can be observed which is defined by the length-scale of the kernel. Finally, the covariance among derivatives starts from a high covariance and decreases to a minimum negative value at a distance of $\sqrt{3l^2}$ [48].

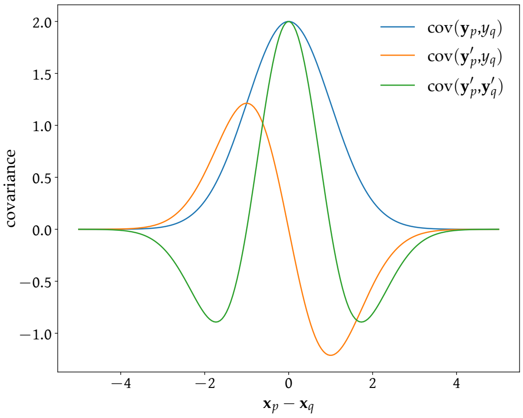


Figure 3.11: Covariance among function values, between function values and its derivatives and among derivatives with respect to distance $(\boldsymbol{x}_p - \boldsymbol{x}_q)$ for SE kernel for $\sigma_f^2 = 2.0$ and $l^2 = 1.0$

Utilizing a similar block matrix notation, as presented in [42] the joint prior distribution can be

[4] This process is also called co-kriging with derivative information. Including multi-fidelity observations in conditioning process is also a type of co-kriging. Thus, to avoid confusion, the co-kriging term will be avoided as much as possible in this work.

expressed as:

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}' \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \tilde{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K} + \sigma_n^2 & \boldsymbol{K}_{yd} \\ \boldsymbol{K}_{yd} & \boldsymbol{K}_{dd} \end{bmatrix}\right), \tag{3.18}$$

where, $\boldsymbol{K}_{yd} \in \mathbb{R}^{N \times ND}$, and $\boldsymbol{K}_{dd} \in \mathbb{R}^{ND \times ND}$ represent blocks of $\mathrm{cov}(\boldsymbol{y}'_p, y_q)$ and $\mathrm{cov}(\boldsymbol{y}'_p, \boldsymbol{y}'_q)$ respectively with D being the dimension of the input. In short, derivative observations are added to the vector of function values of observations and their correlations are added to the covariance structure.

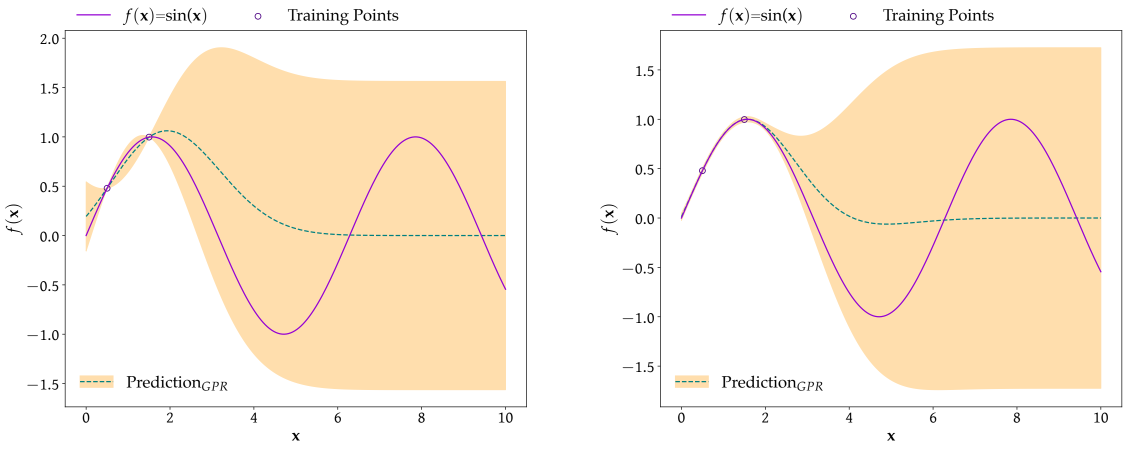
After conditioning the predictive distribution with prior information, point predictions can be expressed similar to the ones presented in Equations 3.8 and 3.40 as:

$$\mathbb{E}[y_* | \boldsymbol{x}_*] = \tilde{\boldsymbol{k}}_*^T (\tilde{\boldsymbol{K}})^{-1} \tilde{\boldsymbol{y}}, \tag{3.19}$$

$$\mathbb{V}[y_* | \boldsymbol{x}_*] = \boldsymbol{k}_{**} - \tilde{\boldsymbol{k}}_*^T (\tilde{\boldsymbol{K}})^{-1} \tilde{\boldsymbol{k}}_* + \sigma_n^2, \tag{3.20}$$

where $\tilde{\boldsymbol{k}}_* = \begin{bmatrix} \boldsymbol{k}_* & \partial_{\boldsymbol{x}_*} k(\boldsymbol{x}_1, \boldsymbol{x}_*) & \dots & \partial_{\boldsymbol{x}_*} k(\boldsymbol{x}_N, \boldsymbol{x}_*) \end{bmatrix}^T$.

The effect of derivative inclusion on the conditioning of the GPR can be observed in Figure 3.12. Derivative inclusion increases the prediction capabilities in the vicinity of training points, by eliminating the possible solutions that do not match the derivatives at the training points.



(a) GPR without derivative observations

(b) GPR with derivative observations

Figure 3.12: GPR predictions for $f(x) = \sin(x)$ on $[0,10]$ with $\mathcal{D} = \{(x_i, y_i) | i = 1,...,2\}$

Although it seems an attractive option to utilize derivative information whenever possible there are certain limitations inherent to the utilization of more information in the conditioning processes. The main computational burden of a GPR lies in the inversion of a covariance matrix (computational complexity $O(N^3)$). Upon the introduction of derivative information, the computational complexity becomes $O((ND+N)^3)$. Thus, the scalability of the GPR model takes a hit upon derivative information inclusion since not only the data amount plays an important role but also the dimension of the input space. There exist multiple ways to bypass this issue, however, it is not in the scope of this work.

3.1.5. Extrapolation

For some applications, extrapolation predictions might be of interest. For instance, the prediction of future values in a time variable dataset or predicting the values out of the feature space that GPR is conditioned. An example of extrapolation can be seen in Figure 3.12 after a certain point the posterior of the GPR tends towards zero, which is directly a result of zero mean assumption for the prior as shown in Equation 3.2. So, if the mean was included in the formulation this would have caused the posterior to tend towards the given mean. Thus, it is clear that the extrapolation capabilities of the GPR models with the zero mean assumption are limited to a certain extent due to certain assumptions present in the model creation (*eg.* kernel selection and zero mean assumptions.) and recognizing the limitations of the models that are being used is of crucial importance.

3.2. Multi-fidelity Gaussian Process Regression (multiGPR)

In section 3.1 a framework on how a surrogate model of an objective function can be created, is presented. Nonetheless, as mentioned before in practical engineering and science applications there might be instances where other approximations from different levels of complexity are present.

This section is devoted to exploiting all available, information from all levels of complexity in a structured way to give the optimal surrogate model. This is referred to as co-kriging of multi-fidelity models.[5] The first multi-fidelity models presented to be utilized in computer experiments date back to 1998 on works of Craig et. al. presented in Constructing Partial Prior Specifications for Models of Complex Physical Systems and are based on a linear regression formulation. One of the models utilized and extended significantly is the auto-regressive model presented in [21].

This section is dedicated to present an overview of the auto-regressive method [21] and recursive method recently developed by Gratiet in [27]. During this introduction to multiGPR, the reader should keep in mind the notations introduced in Section 3.1. Moreover, the zero mean GP prior assumption is still valid during this section for all fidelities.

3.2.1. Auto-Regressive Model

This model is developed for an arbitrary number of fidelities under the following general assumptions,

- Different levels of code are correlated, meaning output values of the different levels of code are in close proximity to each other,
- The response is smooth at all fidelity levels,
- Prior beliefs for all fidelities of codes can be modeled with a GP and
- All outputs are scalar.

[5] From now on the only multi-fidelity model term will be utilized since the co-kriging term is a general name for utilization of multiple information during the kriging.

It should be noted that formulations of this method will be shown only in two-fidelities (low and high) for the sake of simplicity but could be extended to more fidelities with the notation in [21]. Moreover, in the following formulations the subscripts, l, and h represent the low and high fidelities respectively.

Considering the following regression problem for two different fidelities given by:

$$\begin{aligned} y_l &= f_l(\boldsymbol{x}_l) + \varepsilon_l \\ y_h &= f_h(\boldsymbol{x}_h) + \varepsilon_h. \end{aligned} \tag{3.21}$$

$\boldsymbol{X}$ and $\boldsymbol{y}$ mentioned in section 3.1 are combined to represent all the inputs and outputs of the observed data as

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_l \\ \boldsymbol{X}_h \end{pmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_l(\boldsymbol{X}_l) \\ \boldsymbol{y}_h(\boldsymbol{X}_h) \end{pmatrix} \tag{3.22}$$

This model assumes between two successive levels of code the following:

$$\operatorname{cov}\{\boldsymbol{y}_h(\boldsymbol{x}'), \boldsymbol{y}_l(\boldsymbol{x}) | \boldsymbol{y}_l(\boldsymbol{x}')\} = 0, \quad \forall \boldsymbol{x} \neq \boldsymbol{x}'. \tag{3.23}$$

This means nothing more can be learned related to higher-fidelity data at a point, if the function value at that exact location is known, which is a kind of Markov property [12].[6] In addition to this, assuming f_l and f_h represent the local Gaussian features for the respective fidelities, the auto-regressive model reads

$$f_h(\boldsymbol{x}) = \rho f_l(\boldsymbol{x}) + \delta(\boldsymbol{x}), \tag{3.24}$$

where ρ represents both the correlation degree and the scaling between low and high fidelities and δ represents the difference between them [27]. It should be noted that normally the scaling parameter is a function of $\boldsymbol{x}$ and represented as $\rho(\boldsymbol{x})$, but for the sake of simplicity it is assumed to be a constant generally and treated as one of the hyper-parameters of the whole regression model. Then, considering the input data given in Equation 3.22, the covariance matrix is expressed as

$$\boldsymbol{C} = \begin{bmatrix} k_l(\boldsymbol{X}_l, \boldsymbol{X}_l) & \rho k_l(\boldsymbol{X}_l, \boldsymbol{X}_h) \\ \rho k_l(\boldsymbol{X}_h, \boldsymbol{X}_l) & \rho^2 k_l(\boldsymbol{X}_h, \boldsymbol{X}_h) + k_h(\boldsymbol{X}_h, \boldsymbol{X}_h) \end{bmatrix}, \tag{3.25}$$

where the subscript l or h utilized in the kernel expressions indicate which kernels hyper-parameters are used for constructing each covariance block. With this information, the conditioned predictive distribution ($p(\boldsymbol{y}_* | \boldsymbol{y}_l, \boldsymbol{y}_h) \sim \mathcal{N}(\boldsymbol{m}, \mathbf{S})$) mean and covariance is given by:

$$\boldsymbol{m} = \boldsymbol{c}^T \boldsymbol{R}^{-1} \boldsymbol{y}, \tag{3.26}$$

$$\boldsymbol{S} = \rho^2 k_l(\boldsymbol{X}_*, \boldsymbol{X}_*) + k_h(\boldsymbol{X}_*, \boldsymbol{X}_*) - \boldsymbol{c}^T \boldsymbol{R}^{-1} \boldsymbol{c} + (\sigma_n^2)_h \boldsymbol{I}. \tag{3.27}$$

It should be noted that $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ represents the addition of Gaussian noise terms and the

[6] Markov property reads, given the current state of a process the future state is independent from the past states [19].

newly introduced term $\boldsymbol{c}$ are expressed as

$$\boldsymbol{c} = \begin{bmatrix} \rho k_l(\boldsymbol{X}_l, \boldsymbol{X}_*) \\ \rho^2 k_l(\boldsymbol{X}_l, \boldsymbol{X}_*) + k_h(\boldsymbol{X}_h, \boldsymbol{X}_*) \end{bmatrix}. \tag{3.28}$$

It should be noted that model selection is done by maximizing the LML as in the case of GPR where the optimum values of hyper-parameters are found (see Equation 3.11). However, the definitions of N, $\boldsymbol{y}$ and $\boldsymbol{R}$ definitions are adjusted for the case of the auto-regressive model.

Although this model provides an easy framework for implementation of the multi-fidelity Gaussian Process Regression Model there are difficulties inherent to it. Most of them stem from the increasing amount of data that is introduced to the system. In this case, instead of 3 hyper-parameters, 7 hyper-parameters have to be optimized (3 for each of the kernels k_h and k_l and the ρ). This increase in the hyper-parameter number will make the optimization procedure harder. As mentioned before the computational burden of GPR models originate from the covariance matrix inversion and determinant calculations. Thus, constructing the kernel as given in Equation 3.25 causes the computational complexity to become $O((N_1 + \cdots + N_t)^3)$. This means that all the additional fidelity information stacked in a single covariance matrix can become a computational bottleneck. Moreover, it is known that covariance matrices suffer from numerical instabilities. A high correlation between function values will result in a nearly singular covariance matrix, making it ill-conditioned [3]. Although additional jitter terms are commonly utilized to avoid this problem, with increasing the number of fidelities the possibility of getting a strong correlation increases. All of these problems also exist in GPR, but given the construction and formulation of the auto-regressive model, these problems become more prominent as the number of fidelities increase. Moreover, when the surrogate model is built only the highest fidelity model is obtained. Thus, there are no regression models for the other fidelities utilized in constructing the highest fidelity model.

3.2.2. Recursive Model

In [27] Loic Le Gratiet attempts to address problems regarding the auto-regressive model and reformulates the multi-fidelity problem in such a way that the complexity of the method remains similar to that of a single fidelity Gaussian Process Regression. Again, it should be noted that this work also utilizes [21], thus the basic assumptions mentioned before are still valid for this method as well.

Consider the s-levels of regression problem,

$$y_t = f_t(\boldsymbol{x}_t) + \varepsilon_t, \tag{3.29}$$

where $t = 1,...,s$, with s being the highest fidelity. Then, the auto-regressive model can be expressed as:

$$f_t(\boldsymbol{x}) = \rho_{t-1}(\boldsymbol{x}) f_{t-1}(\boldsymbol{x}) + \delta_t(\boldsymbol{x}), \quad t = 2,...,s, \tag{3.30}$$

where $\delta_t(x)$ is a Gaussian field that is independent from any $\{f_{t-1},...,f_1\}$ that represents the difference between two consecutive f's. As mentioned before $\rho(\boldsymbol{x})$ is a scaling parameter that is assumed to be a scalar constant, but normally it is considered as another random field with an assigned prior distribution and calibrated with the data [37].

The main difference between the recursive model and the auto-regressive model comes from the conditioning of every fidelity data with the previous models. Thus, assuming design sets ($D_i = \{(x_j,y_j)|j = 1,...,n_t\}$) to be in a nested structure ($D_1 \subseteq ... \subseteq D_{t-1}$) allows for the decoupling of the s level auto-regressive model into s independent GPR problems [37]. In other words, it is assumed that each fidelity will only carry information coming from the lower fidelities. This assumption is reasonable since in practical applications codes are sorted in increasing fidelity [27]. Although the nested structure is not necessary to build the model, it allows for easier estimation of the model parameters [27].

Then, the predictive distribution ($p(\boldsymbol{y}_*|\boldsymbol{y}_1,...,\boldsymbol{y}_t) \sim \mathcal{N}(\boldsymbol{m}_t,\mathbf{S}_t)$) mean and covariance are given by:

$$\boldsymbol{m}_t = \rho_{t-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t^*) + k(\boldsymbol{X}_t^*,\boldsymbol{X}_t)\big(k(\boldsymbol{X}_t,\boldsymbol{X}_t) + \sigma_{n_t}^2\boldsymbol{I}\big)^{-1}(\boldsymbol{y}_t - \rho_{t-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t)), \tag{3.31}$$

$$\boldsymbol{S}_t = \rho_{t-1}^2\boldsymbol{S}_{t-1}(\boldsymbol{X}_t^*) + k(\boldsymbol{X}_t^*,\boldsymbol{X}_t^*) - k(\boldsymbol{X}_t^*,\boldsymbol{X}_t)\big(k(\boldsymbol{X}_t,\boldsymbol{X}_t) + \sigma_{n_t}^2\boldsymbol{I}\big)^{-1}k(\boldsymbol{X}_t,\boldsymbol{X}_t^*) + \sigma_{n_t}^2\boldsymbol{I}, \tag{3.32}$$

where, $\boldsymbol{m}_{t-1}(\boldsymbol{x}_t^*)$ and $\mathbf{S}_{t-1}(\boldsymbol{x}_t^*)$ represents the mean and covariance blocks of the previous fidelities prediction mean and covariance matrix evaluated at the corresponding input values $\boldsymbol{X}_t$ of the fidelity that is in process of being predicted. It can be seen that for $t = 1$ Equations 3.31 and 3.32 takes the form of GPR versions given in Equations 3.6 and 3.7. For a single prediction point $\boldsymbol{x}_*$ expectation and variance can be expressed in a simpler format as:

$$\mathbb{E}_t[y_*|\boldsymbol{x}_*] = \rho_{t-1}\mathbb{E}_{t-1}[y_*|\boldsymbol{x}_*] + \boldsymbol{k}_*^T\big(\boldsymbol{K}_t + \sigma_{n_t}^2\boldsymbol{I}\big)^{-1}(\boldsymbol{y}_t - \rho_{t-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t)), \tag{3.33}$$

$$\mathbb{V}_t[y_*|\boldsymbol{x}_*] = \rho_{t-1}^2\mathbb{V}_{t-1}[y_*|\boldsymbol{x}_*] + \boldsymbol{k}_{**} - \boldsymbol{k}_*^T\big(\boldsymbol{K}_t + \sigma_{n_t}^2\boldsymbol{I}\big)^{-1}\boldsymbol{k}_* + \sigma_{n_t}^2, \tag{3.34}$$

where $\boldsymbol{k}_* = k(\boldsymbol{X}_t,\boldsymbol{x}_*) \in \mathbb{R}^{N_t\times 1}$, $\boldsymbol{k}_{**} = k(\boldsymbol{x}_*,\boldsymbol{x}_*) \in \mathbb{R}$ and $\boldsymbol{K} = k(\boldsymbol{X}_t,\boldsymbol{X}_t) \in \mathbb{R}^{N_t\times N_t}$.

As stated with all the other methods, the optimum values of hyper-parameters can be found by maximizing the LML expressed as:

$$\log p(\boldsymbol{y}_t|\boldsymbol{X}_t,f_{t-1}) = -\frac{1}{2}(\boldsymbol{y} - \rho_{t-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t))^T\boldsymbol{R}^{-1}(\boldsymbol{y} - \rho_{t-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t)) - \frac{1}{2}\log|\boldsymbol{R}| - \frac{N_t}{2}\log 2\pi, \tag{3.35}$$

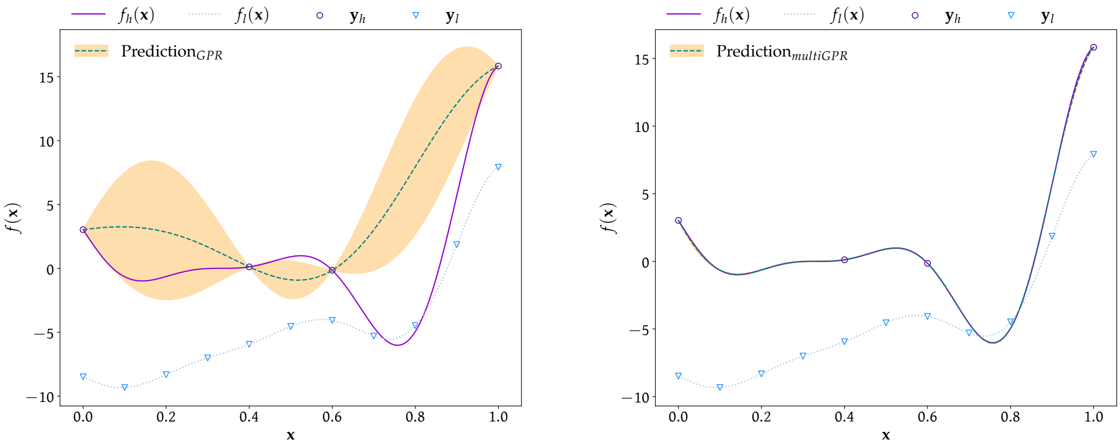
where $\boldsymbol{R} = k(\boldsymbol{X}_t,\boldsymbol{X}_t) + \sigma_{n_t}^2\boldsymbol{I}$. Normally, the LML of a multiGPR model should be optimized for the hyper-parameters of the kernel ($\sigma_{f_t}^2$, l_t and $\sigma_{n_t}^2$) in combination with the assumed to be scalar scaling parameter ρ_{t-1}. However, to simplify the optimization procedure and ease up the implementation of this method to existing frameworks, the fact that ρ_{t-1} can be decoupled from the remaining hyper-parameters is exploited by doing:

$$\nabla_{\rho_{t-1}}\log\Big(p(\boldsymbol{y}_t|\boldsymbol{X}_t,f_{t-1})\Big) = 0, \tag{3.36}$$

which leads to and optimum value for ρ_{t-1}:

$$\hat{\rho}_{t-1} = \big(\boldsymbol{m}_{t-1}(\boldsymbol{X}_t^*)^T\boldsymbol{R}^{-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t^*)\big)^{-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t)\boldsymbol{R}^{-1}\boldsymbol{y}_t. \tag{3.37}$$

It is obvious that the optimum value of the ρ_{t-1} depends on the remaning hyper-parameters, thus it is only computed after optimizing for $\sigma_{f_t}^2$, l_t and $\sigma_{n_t}^2$ using BFGS [11].



(a) GPR predictions with only $\mathcal{D}_h$.

(b) multiGPR predictions with $\mathcal{D}_h$ and $\mathcal{D}_l$.

Figure 3.13: Comparison of GPR and multiGPR for $f_h(x) = (6x-2)^2 \sin(12x-4)$ and $f_l(x) = Af_h + B(x-0.5) + C$ in one dimension with $A = 0.5$, $B = 10$ and $C = -5$ with $\mathcal{D}_h = \{(x_i, y_i) | i = 1, ..., 4\}$ and $\mathcal{D}_l = \{(x_i, y_i) | i = 1, ..., 11\}$.

It can be observed that compared to the auto-regressive model, the recursive method resolves the bottleneck of computational complexity and numerical instability by handling one fidelity at a time with the information coming from the fidelity at one level lower. As it can be observed from Equation 3.25 computational complexity for auto-regressive method is $O((N_1 + \cdots + N_t)^3)$ whereas for recursive model it is $O(N_1^3 + \cdots + N_t^3)$ for the bi-fidelity regression. Moreover, the recursive approach presents regression predictions for all fidelities, which are readily available for possible exploitation or investigation. However, with the auto-regressive approach, only the highest-fidelity predictions are available. Furthermore, Gratiet shows that the posterior distributions give identical results [27]. Thus, it can be concluded that the recursive formulation of the multi-fidelity regression model is superior to the auto-regressive model. It should be noted that from now on only the recursive model will be considered in this work and the term multiGPR will only refer to the recursive model.

An example of multiGPR and its comparison with GPR prediction results can be seen in Figure 3.13 for widely utilized test functions in multiGPR literature [9, 12, 28, 32]. It is clear that GPR predictions are poor due to $\mathcal{D}_h$ having a small size. However, this poor prediction should not be perceived as a bad prediction since the information provided by $\mathcal{D}_h$ results in a posterior that maximizes its own likelihood, which means that with the given information this is one of many modes of predictions that maximize the LML. On the other hand, low-fidelity information inclusion in the regression process provides a substantial amount of new information for the posterior, as the prediction of multiGPR fits almost perfectly to the exact function values.

3.2.3. Derivative Observations

As in the case of GPR, posterior mean of derivative observations can be found by:

$$\frac{\partial}{\partial \boldsymbol{x}_*}\mathbb{E}_t[y_*|x_*] = \rho_{t-1}\frac{\partial}{\partial \boldsymbol{x}_*}\mathbb{E}_{t-1}[y_*|\boldsymbol{x}_*] + \Big(\frac{\partial}{\partial \boldsymbol{x}_*}k(\boldsymbol{x}_*,\boldsymbol{X}_t)\Big)\Big(\big(\boldsymbol{K}_t + \sigma_n^2\boldsymbol{I}\big)^{-1}(\boldsymbol{y}_t - \rho_{t-1}\boldsymbol{m}_{t-1}(\boldsymbol{X}_t))\Big), \quad (3.38)$$

where $\partial_{\boldsymbol{x}_*}\mathbb{E}_1[y_*|\boldsymbol{x}_*]$ is given in Equation 3.13 and $\partial_{\boldsymbol{x}_*}k(\boldsymbol{x}_*,\boldsymbol{X}_t)$ is given by the expression Equation 3.14.

In cases where derivative information is available, utilizing the same additional terms presented in Equations 3.15 and 3.16 for covariance matrix $\tilde{\boldsymbol{K}}_t$ creation at each level of fidelity t, prediction results can be expressed as:

$$\mathbb{E}_t[y_*|\boldsymbol{x}_*] = \rho_{t-1}\mathbb{E}_{t-1}[y_*|\boldsymbol{x}_*] + \tilde{\boldsymbol{k}}_*^T\big(\tilde{\boldsymbol{K}} + \sigma_{n_t}^2\boldsymbol{I}\big)^{-1}(\tilde{\boldsymbol{y}}_t - \rho_{t-1}\tilde{\boldsymbol{m}}_{t-1}(\boldsymbol{X}_t)), \quad (3.39)$$

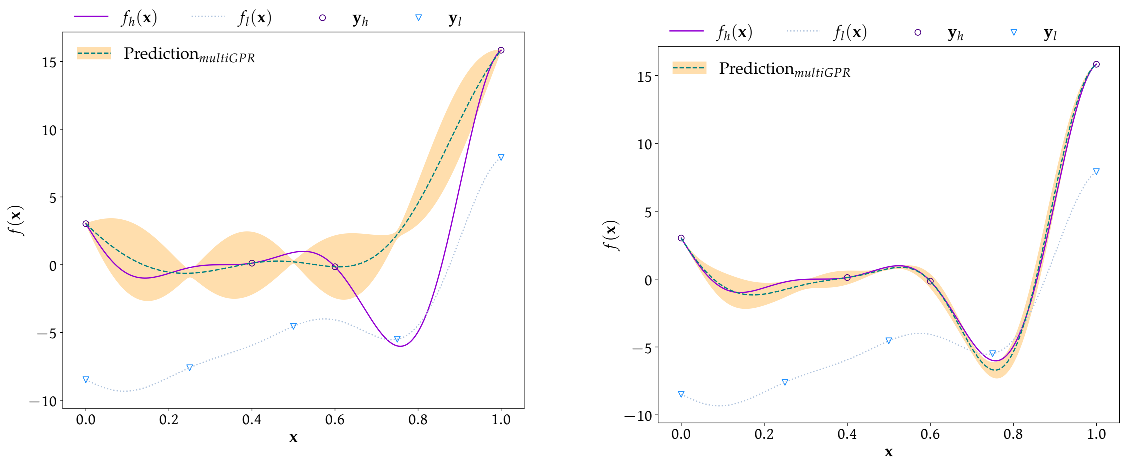
$$\mathbb{V}_t[y_*|\boldsymbol{x}_*] = \rho_{t-1}^2\mathbb{V}_{t-1}[y_*|\boldsymbol{x}_*] + \boldsymbol{k}_{**} - \tilde{\boldsymbol{k}}_*^T\big(\tilde{\boldsymbol{K}} + \sigma_{n_t}^2\boldsymbol{I}\big)^{-1}\tilde{\boldsymbol{k}}_* + \sigma_{n_t}^2, \quad (3.40)$$

where, $\tilde{\boldsymbol{k}}_* = \begin{bmatrix}\boldsymbol{k}_* & \partial_{\boldsymbol{x}_*}k(\boldsymbol{x}_{t1},\boldsymbol{x}_*) & \dots & \partial_{\boldsymbol{x}_*}k(\boldsymbol{x}_{tN},\boldsymbol{x}_*)\end{bmatrix}^T$ and $\tilde{\boldsymbol{m}}_{t-1}(\boldsymbol{X}_t)$ is the lower-fidelity regression results at the points of higher-fidelity input points with derivative predictions stacked together in a single vector.

Again, increased prediction capabilities in the case of derivative observation inclusion for multiGPR can be observed in Figure 3.14. An important difference with the example shown in Figure 3.13 is the fact that $\mathcal{D}_l = \{(x_i,y_i)|i = 1,...,5\}$ meaning only 5 observations have been utilized from lower-fidelity instead of 11. The motivation behind this choice was to decrease the fit of the function such that it is possible to see the difference between the prediction capabilities of multiGPR. As expected, as the number of lower-fidelity training points decreases, model uncertainty increases and the multiGP gives poor prediction results. Another important observation is the decreased uncertainty bounds for the higher-fidelity predictions at the places where no training point exists, as it can be seen in Figure 3.14a. This overconfidence is originating from the lower-fidelity since low-fidelity variances are added to the high-fidelity. This overconfidence issue for the regions where there is no information from the high-fidelity should be treated with care. As in the case of co-kriging with derivative information the matching slopes of objective function and its surrogate can be observed in Figure 3.14.

3.2.4. Extrapolation

Although investigating the extrapolation predictions is interesting, the term extrapolation in a multiGPR setting is not as clear as it is for GPR since the information source is variable. Thus, capabilities of multiGPR is investigated in Figure 3.15 for two different types of setup. In Figure 3.15a the case where thoroughly sampled low-fidelity information available in the prediction domain is investigated, making it a not pure extrapolation. Moreover, in Figure 3.15b the case where there is neither information coming from the low-fidelity nor from the high-fidelity is investigated, making it a pure extrapolation. It can be seen that by collecting lower-fidelity abundantly a good prediction of the

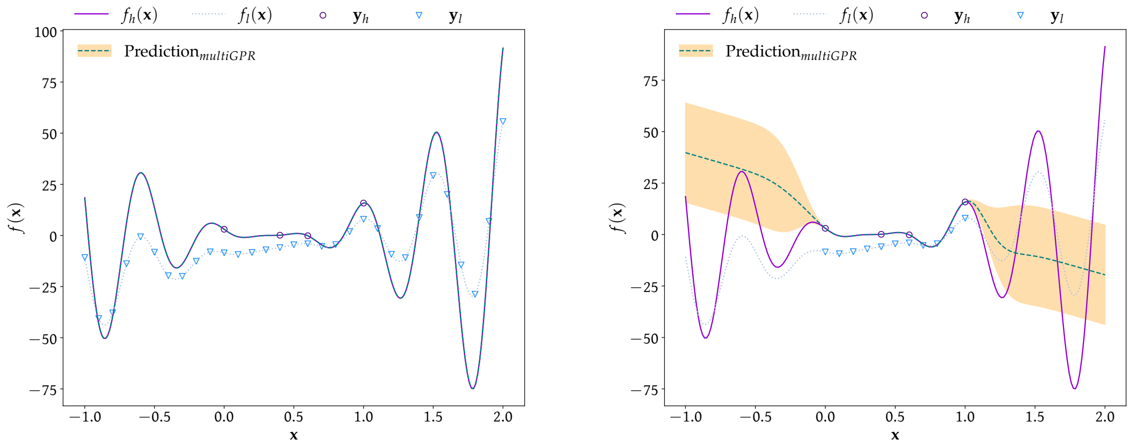


(a) multiGPR predictions without derivative observations.

(b) multiGPR predictions with derivative observations

Figure 3.14: Investigating the effect of derivative observation for $f_h(x) = (6x-2)^2\sin(12x-4)$ and $f_l(x) = Af_h + B(x-0.5) + C$ in one dimension with $A = 0.5$, $B = 10$ and $C = -5$ with $\mathcal{D}_h = \{(x_i, y_i)|i = 1,...,4\}$ and $\mathcal{D}_l = \{(x_i, y_i)|i = 1,...,5\}$.

objective function is obtained. However, when there is little to no information available the predictions are very poor for extrapolation. The same phenomenon was observed by Gratiet and concluded that multiGPR is not suitable for extrapolation outside of the design set utilized for training [27]. Although it is not shown in 3.15b multiGPR extrapolation goes to its zero mean prior with a high variance similar to GPR extrapolation. However, inclusion of a mean-like term coming from the lower-fidelity scaled with ρ results in the prediction to go to its zero mean prior at a much larger distance compared to GPR.



(a) $\mathcal{D}_h = \{(x_i, y_i)|i = 1,...,4\}$ and $\mathcal{D}_l = \{(x_i, y_i)|i = 1,...,31\}$.

(b) $\mathcal{D}_h = \{(x_i, y_i)|i = 1,...,4\}$ and $\mathcal{D}_l = \{(x_i, y_i)|i = 1,...,11\}$.

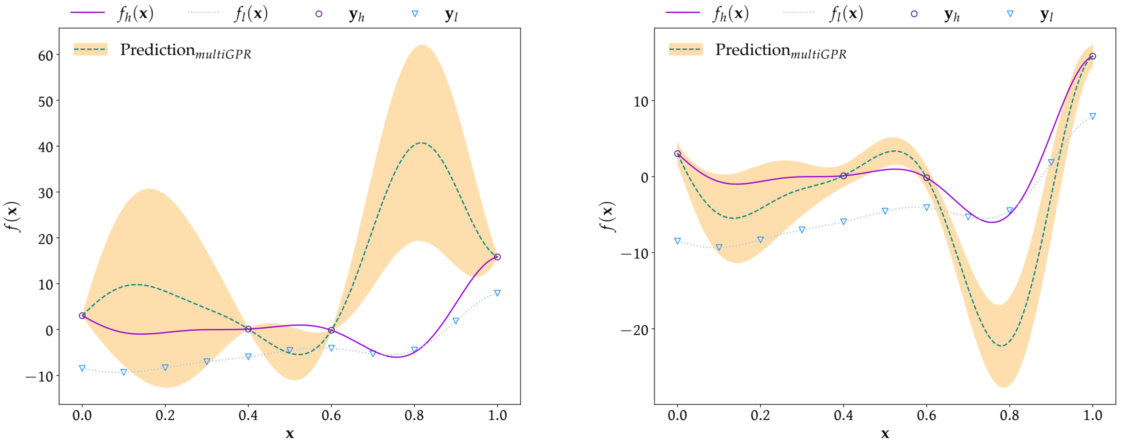
Figure 3.15: multiGPR extrapolation for $f_h(x) = (6x-2)^2\sin(12x-4)$ and $f_l(x) = Af_h + B(x-0.5) + C$ in one dimension with $A = 0.5$, $B = 10$ and $C = -5$

Although the extrapolation prediction capabilities of multiGPR seem to predict almost exactly the objective function, it should be noted that the prediction results are highly affected by the training points and the type of problem being approximated.

3.2.5. Understanding the hyper-parameter ρ

As mentioned before ρ is a hyper-parameter that scales the correlation between two successive fidelities [8, 27]. It can be observed from Equation 3.31, that the terms with the ρ can be interpreted as a mean function that is dependent on the lower fidelity which is scaled by the correlation between the successive fidelities. Thus, it has a similar behavior as a mean function as well.

Effect of ρ for changing values can be observed in Figure 3.16 where instead of optimizing, ρ is fixed to a scalar value. It is clear to see that the negative ρ results in a negative correlation with the low-fidelity function that is utilized in Figure 3.16a. Moreover from Figure 3.16b, it can be observed that increasing the optimum value of $\rho = 2.0$ (see Figure 3.13b) to $\rho = 5.0$ results in the scaling of the correlation to a larger value away from the training points.



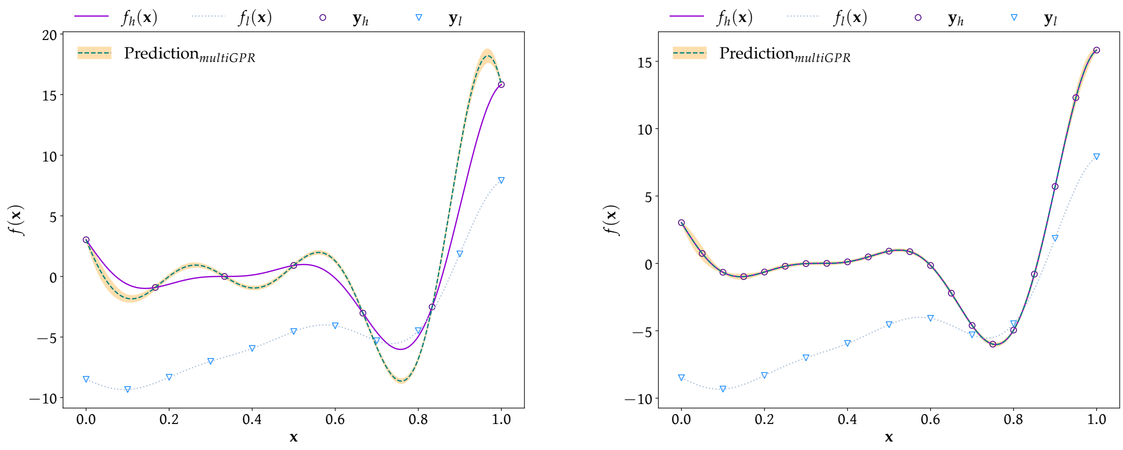
(a) multiGPR predictions for $\rho = -5.0$

(b) multiGPR predictions for $\rho = 5$

Figure 3.16: Comparison of different ρ values for $f_h(x) = (6x-2)^2\sin(12x-4)$ and $f_l(x) = Af_h + B(x-0.5) + C$ in one dimension with $A = 0.5$, $B = 10$ and $C = -5$ with $\mathcal{D}_h = \{(x_i, y_i)|i = 1, ..., 4\}$ and $\mathcal{D}_l = \{(x_i, y_i)|i = 1, ..., 11\}$.

Aforementioned effect of ρ away from the training points can be better visualized in Figure 3.17 when the same problem is further investigated by increasing the training points in the high-fidelity. In order to investigate this effect ρ is kept constant and the number of training points for high-fidelity is increased in the given domain. It is clear to see that as the training points are sampled more densely, the effect of ρ on the prediction results decreases dramatically on the mean predictions. This observation is especially important when the intended use of multiGPR framework is extrapolation in the high-fidelity with the information coming from the low-fidelity. In other words, when there is less training points from the high-fidelity the low-fidelity predictions scaled with the ρ dominate the high-fidelity prediction.

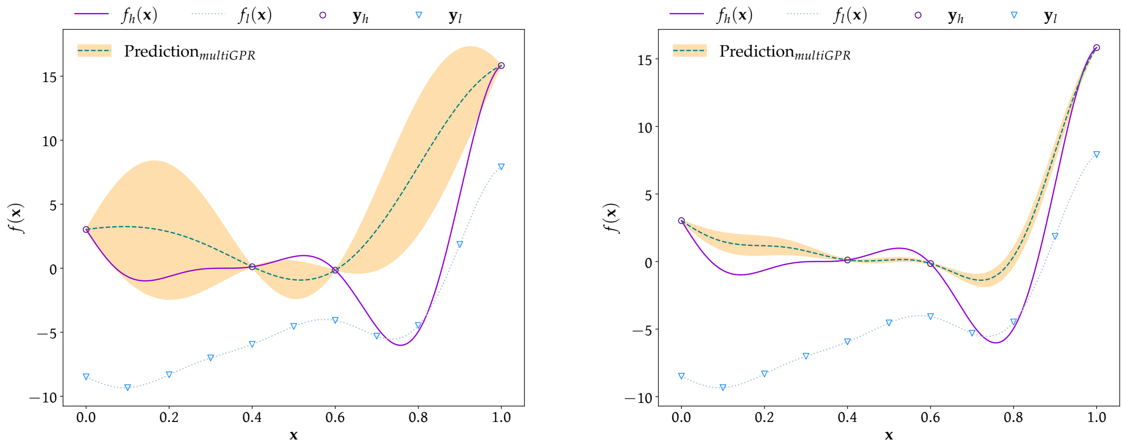
Two important limit cases are visualized for $\rho = 1$ and $\rho = 0$ in Figure 3.18. As expected the $\rho = 0$ shown in Figure 3.18a results in GPR predictions that is observed in Figure 3.13a ($f_t(x) = \delta_t(x)$). Moreover, the $\rho = 1$ case presented in 3.18b results in predictions very similar to the lower fidelity as expected, whereas it is not exactly the same due to the difference parameter adjusting the predictions



(a) multiGPR predictions for $\mathcal{D}_h = \{(x_i, y_i) | i = 1, ..., 7\}$ (b) multiGPR predictions for $\mathcal{D}_h = \{(x_i, y_i) | i = 1, ..., 21\}$

Figure 3.17: Comparison of a constant ρ for increasing training point density for $f_h(x) = (6x-2)^2 \sin(12x-4)$ and $f_l(x) = Af_h + B(x-0.5) + C$ in one dimension with $A = 0.5$, $B = 10$ and $C = -5$ with $\mathcal{D}_l = \{(x_i, y_i) | i = 1, ..., 11\}$.

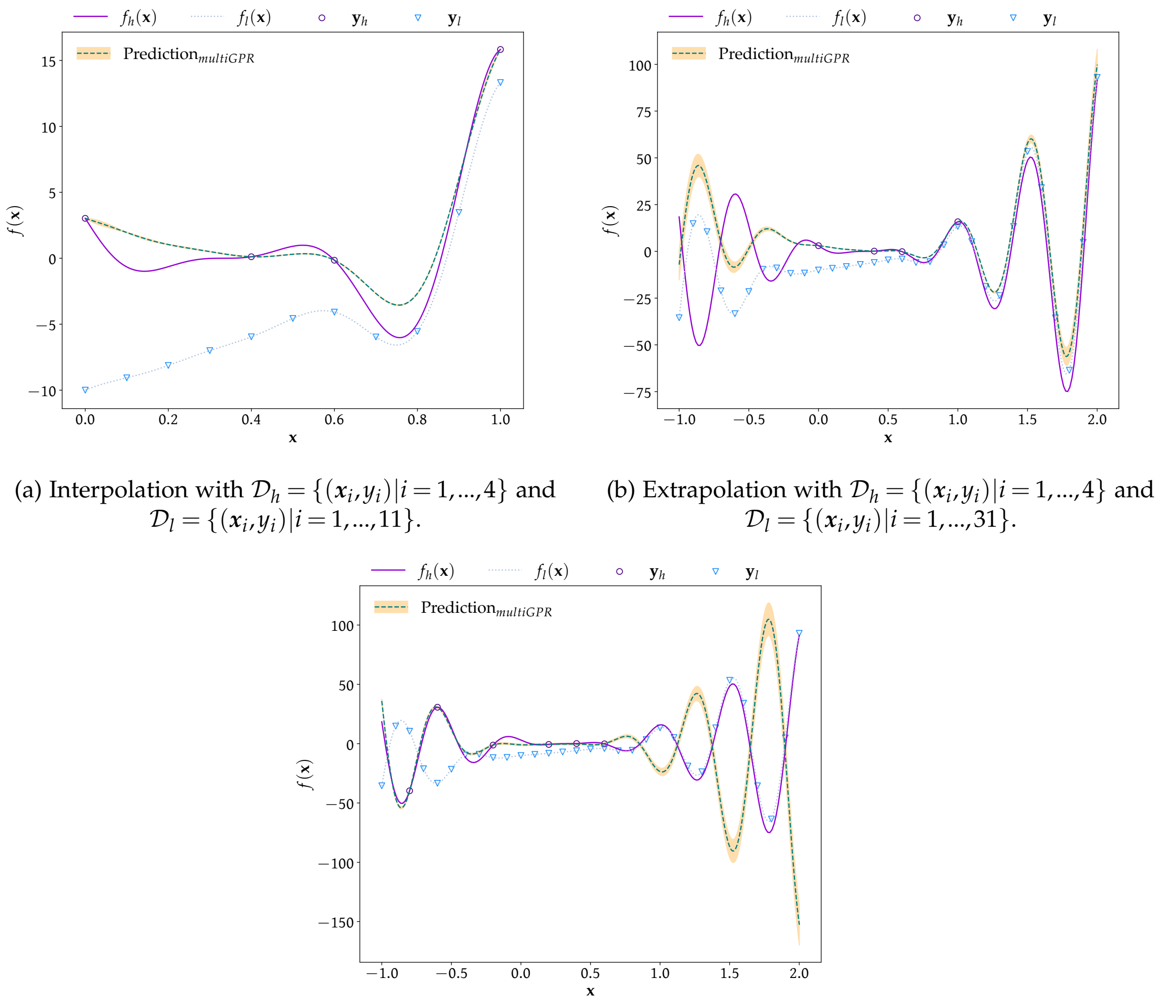
caused by the discrepancy between low- and high-fidelity originating from $B(x-0.5)+C$ term in the example function.



(a) multiGPR predictions for $\rho = 0.0$ (b) multiGPR predictions for $\rho = 1.0$

Figure 3.18: Investigating the important cases of ρ for $f_h(x) = (6x-2)^2 \sin(12x-4)$ and $f_l(x) = Af_h + B(x-0.5) + C$ in one dimension with $A = 0.5$, $B = 10$ and $C = -5$ with $\mathcal{D}_h = \{(x_i, y_i) | i = 1, ..., 4\}$ and $\mathcal{D}_l = \{(x_i, y_i) | i = 1, ..., 11\}$.

It is clear to see that the auto-regressive method presented in Equation 3.30 is only capable of fully capturing the linear correlations between two successive fidelities. This can be seen in Figure 3.19 where the correlation between fidelities is nonlinear. Figure 3.19a illustrates the interpolation case, where the nonlinear relation between the low- and high-fidelity is not captured well enough. In addition, there is a visible problem with overconfidence which takes away one of the most powerful features of GP's, which is integrated uncertainty bounds. Moreover, ρ has a huge impact on the extrapolation capabilities as well for the given nonlinear correlation case. This problem is visualized in Figure 3.19b, where training points are selected from the region where the correlation of lower and higher fidelity is positive, and the extrapolation is conducted in a region that involves both negative and positive correlations. As can be seen from Figure 3.19b, the positive correlation case is represented fairly well, whereas the negative correlation region is not captured at all when the training points are selected from the positively correlated region. However, when training points from both positive and negative regions are utilized (Figure 3.19c) negative correlation is found to be favored for the given example which results in unrepresentative prediction results on the positively correlated region. Thus, it can be concluded that a scalar ρ for the whole domain might not be representative of some cases with variable correlations. However, it might be possible to use this oversimplified version for the cases with little variations of correlations in the domain since the positive correlation parts are represented fairly well in Figure 3.19b.



(a) Interpolation with $\mathcal{D}_h = \{(x_i, y_i)|i = 1, ..., 4\}$ and $\mathcal{D}_l = \{(x_i, y_i)|i = 1, ..., 11\}$.

(b) Extrapolation with $\mathcal{D}_h = \{(x_i, y_i)|i = 1, ..., 4\}$ and $\mathcal{D}_l = \{(x_i, y_i)|i = 1, ..., 31\}$.

(c) Extrapolation with $\mathcal{D}_h = \{(x_i, y_i)|i = 1, ..., 6\}$ and $\mathcal{D}_l = \{(x_i, y_i)|i = 1, ..., 31\}$

Figure 3.19: Investigating the effect of ρ in multiGPR formulation for nonlinear correlation between two successive fidelities for $f_h(x) = (6x-2)^2 \sin(12x-4)$ and $f_l(x) = Af_h + B(x-0.5) + C$ in one dimension with $A = \sin(x)$, $B = 10$ and $C = -5$

This work acknowledges the limitations of the assumptions that are made by utilizing a single scalar ρ and utilizes this assumption for the sake of simplicity as there are a variety of successful applications with the same assumption presented in [12, 21, 27, 37] . Information regarding a scaling parameter that is dependent on x is provided in [27]. Moreover, the limitations caused by the formulation given in Equation 3.30 can be tackled in different ways, interested readers can find the related information in [29, 32, 36, 54].

3.3. Discussion on Utilization of GPs as Surrogate Models

Throughout Chapter 3, the creation of surrogate models from GP's is explained in detail. It is evident that there are a number of assumptions for surrogate model creation starting from assuming a Gaussian distribution until the very end where the successive GP models are scaled and correlated with a single parameter. It is important to know the assumptions of the created models to be able to understand what the output of the model is. That is why most of the assumption was investigated in detail.

The reader may argue against any of the assumptions lying under the hood of a GPR or multiGPR and wonder the reason to not use other methods. Although it is certain that GP has some serious assumptions that may hinder its usage, there are benefits of using a GP, which include being able to have uncertainty information for the predictions and adding prior knowledge to your model via kernel selection. Although there are known bottlenecks for GPs like not being able to represent singularities or dealing with a huge amount of data. There are certain techniques developed to tackle most of the bottlenecks as well. Moreover, it is true that the world around us is not always Gaussian, but other processes might not be mathematically tractable, thus require computationally burdensome approximation techniques. Since there should be a balance between a models' tractability and applicability, GPs provide a good basis for building surrogates.

In short, GP's should be utilized with great care due to its underlying assumptions, but most of those assumptions are needed to express a model in a simple enough mathematical form. As long as the assumptions are understood properly, it is fairly straight forward to build a model with GPs and try to improve its certain aspects as one moves along, especially with the abundance of online tools that are available in a variety of platforms.

3.4. Gaussian Processes as a Constitutive Model

Unlike the Neural Networks utilizing GPs as a constitutive model is not as straight forward. As it can be observed, this chapter presented prediction for single dimension output y from a multidimensional $\mathbf{x}$, but in order to be utilized as a constitutive model, the output dimension should match the input dimensions since for every strain input there is a stress output.

There are multiple ways to tackle this problem. The first one is treating each output independently. Another way is to correlate the processes through a correlated noise process, which will induce correlations between different outputs in the posterior. Finally, if the prior structure is known covariance function selection can be done in a way to include the correlations between different channels [41].

This work utilizes the simplest approach which is to create independent GPs for each output. In other words, in 2 dimensional setting input $\boldsymbol{\varepsilon} = [\varepsilon_{xx} \quad \varepsilon_{yy} \quad \gamma_{xy}]$ provided to every GP instance with

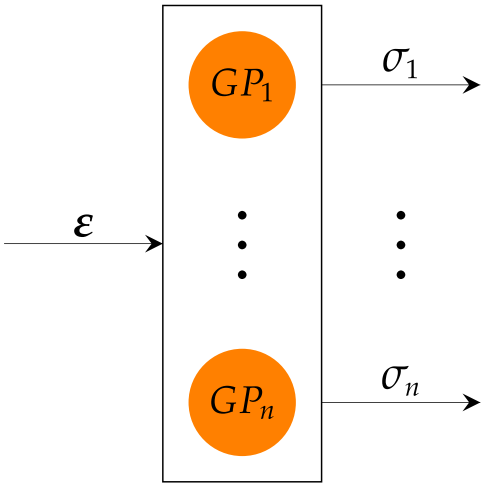


Figure 3.20: Visualization of multiple GPs as a constitutive model.

their respective output stresses σ_{xx}, σ_{yy} and σ_{xy}. Although it is easy to implement, the assumption of independent output values might result in information loss [41].

4 | Methodology

As mentioned before the main aim of this thesis is to investigate the capabilities of the multiGPR framework as a constitutive model. This thesis focuses on a two-dimensional problem setting in order to have a better understanding of the fundamental aspects of the multiGPR while the employed model input and output are small enough whereas the problem is complex enough.

Creating surrogate models by means of GPs as constitutive models requires an *offline* training process, which means all the GPs are conditioned with the snapshots obtained from the FE^2 results obtained from the selected type of problems. During this training, hyper-parameters are optimized and the most probable constitutive relations are established to be later on employed as a material model. A material model is an indication of employing an online solver in this case. In other words, the surrogate material model is also responsible for the creation of a stiffness matrix and provides that information to the selected solver during the simulation for every iteration, then when the criteria of convergence are met the solution is accepted.

The multiGPR procedure aims to decrease the number of points needed in the *offline* training stage. This is achieved by including information coming from a very cheap and unrepresentative version of the simulation that is desired. For the purpose of understanding the capabilities of multiGPR, only bi-fidelity information is considered which allows an easy traceback in the case of unexpected results.

It should be noted that this work is implemented on an open-source numerical library provided by the Dynaflow Research Group, which is called Jem/Jive and written in C++.

4.1. Creating Fidelities

There are different ways to create fidelities as shown in Figure 4.1. The first one is decreasing the number of *degrees of freedom* in the given RVE, which allows decreased computational burden with a less representative result. Another one is decreasing the representativeness of the micromodel. This is achieved by decreasing the number of fibers included in the RVE. Finally, the results coming from another surrogate model is employed for a fast, but unrepresentative information source. Although there are many ways to create fidelities this thesis uses a homogenous material solution as low-fidelity and a 1-fiber RVE as high-fidelity.

4.2. Analysis

Two different types of analysis are considered in this thesis. The first type involves constraining all the strains (controlled strain paths) of the RVE so that total control over the feature space is maintained

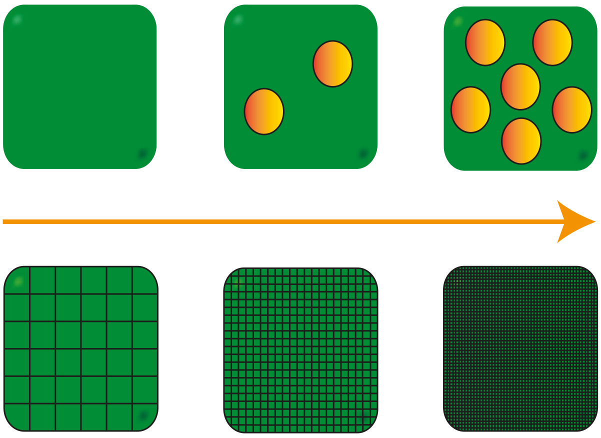
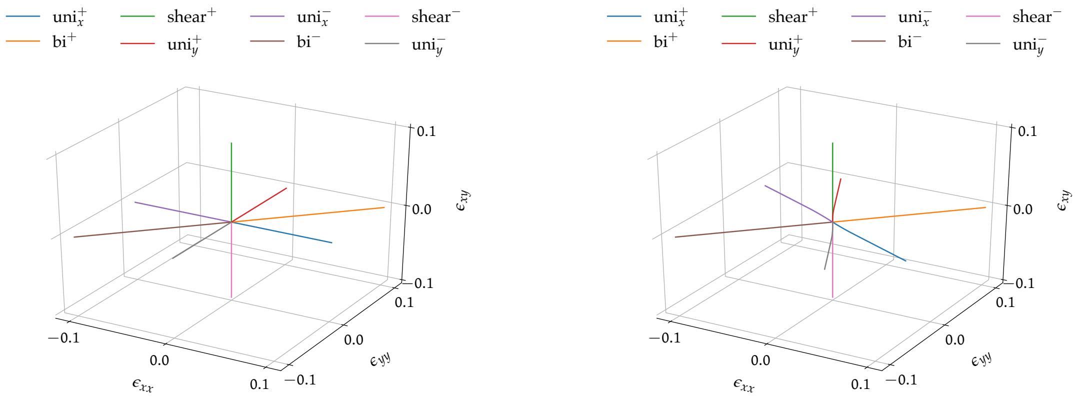


Figure 4.1: Increasing fidelities and computational burden.

during the analysis. In other words strain paths for simulations are not affected by the Poisson effect (uncontrolled strain paths). However, this does not mean the Poisson effect is not present. In fact, it is present and is embedded in the stress outputs. Another type relies on prescribing only the relevant strains and finding the other(s) by solving the global system of equations for the given time step. Sample strain paths for both types of analysis with common loading scenarios utilized in this work can be seen in Figure 4.2. Thus, from now on an analysis with the first type of analysis is referred to as controlled analysis (Figure 4.2a) and the second type of analysis is referred to as an uncontrolled case (Figure 4.2b).



(a) Controlled

(b) Uncontrolled

Figure 4.2: Strain paths for common load cases utilized for both types of analysis.

Creating surrogate constitutive models requires snapshots from the FE^2 method. Thus, another important measure is to decide until which point the full order solution is collected. The snapshots are created with 100-time steps where the stepping size is 0.001 for all the analyses conducted. For an ideal case, this results in a 10% strain level at the end, which ensures the plastic response to be well established. However, for cases where the convergence is not achieved, an adaptive stepping scheme is put into use. This scheme decreases the step size previously defined as 0.001 until the convergence is achieved. Because of this reason the snapshot collections might involve uneven step sizes.

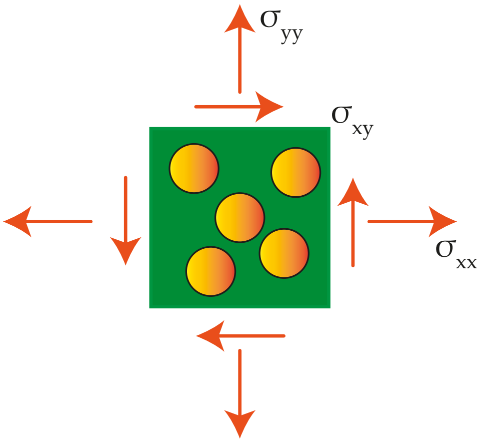


Figure 4.3: Sign convention

The sign convention that is adopted is presented in Figure 4.3. Moreover, as mentioned before this work focuses on two-dimensional analysis with plane strain assumption.

Finally, it should be noted that unless mentioned otherwise the above values are valid for all the simulations performed in this thesis.

4.3. Material and RVE Information

In this thesis, a micro-heterogenous material consisting of fibers embedded inside the matrix is considered. Fibers are purely elastic with $E = 74000$ MPa (Young's modulus) and $\nu = 0.2$ (Poissons's ratio). Moreover, for the matrix a pressure-dependent elastoplastic model proposed by Melro et al. [31] is used with $E = 3130$ MPa, $\nu = 0.37$, $\nu_p = 0.32$ (plastic Poissons's ratio) and hardening behavior given by;

$$\sigma_t = 64.80 - 33.6e^{-\epsilon_{eq}^p/0.003407} - 10.21e^{-\epsilon_{eq}^p/0.006493}, \tag{4.1}$$

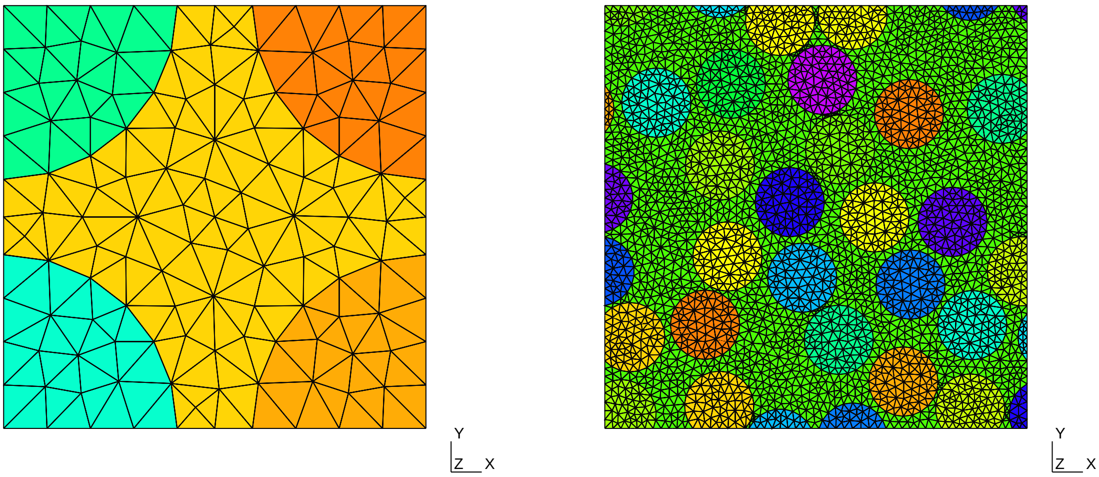
$$\sigma_c = 81.00 - 42.0e^{-\epsilon_{eq}^p/0.003407} - 12.77e^{-\epsilon_{eq}^p/0.006493}, \tag{4.2}$$

where ϵ_{eq}^p represents the equivalent plastic strain [42].

4.4. Surrogate Model Creation

This work includes multiple versions of surrogate constitutive models created with Gaussian Processes regression. Thus, it is important to have conventions to indicate which method is being employed. Expressions that can be found in the legends are as follows:

- GPR: Gaussian Process Regression without derivative information,
- multiGPR: Multi-fidelity Gaussian Process Regression without derivative information,
- GPR$^+$: Gaussian Process Regression with derivative information, and



(a) 1-fiber RVE

(b) 25-fiber RVE

Figure 4.4: Examples of RVEs.

- multiGPR$^+$: Multi-fidelity Gaussian Process Regression with derivative information.

In addition to the given abbreviations, the subscripts l and h represent the low and high fidelities, respectively.

4.5. Training Points for Surrogate Model Creation

In the case of creating a surrogate constitutive model, training points are collected from the snapshots obtained from FE^2 simulations conducted before training. Noting the evident scalability problem of the GP methods, training points should be limited. Because if a method to circumvent the computational bottleneck caused by the inversion of the covariance matrix is not employed, a dramatic increase in the training time is inevitable. Thus, training point selection is done in a consistent manner, for which the domain is divided into equal parts by means of the number of snapshots collected. For example, if 100 snapshots are available from a loading case and a training point stepping of 6 is selected the number of training points is 17. Due to the adaptive stepping scheme and possible non-converging solutions, the selected training points might not be equally spaced in the feature space. However, from the perspective of available snapshots, they are equally spaced.

It should be noted that all the loading cases and selected number of training points are indicated explicitly with the results, when applicable.

Another important aspect is the presentation of the training points and the exact full-order solutions in two-dimensional plots. It is important to note that representing the given multi-dimensional problem in two-dimensional plots brings forth some problems when the strain path is not controlled. Since the solution is allowed to move anywhere in strain space, visualizing the exact solution and training points might be misleading. That is the reason for excluding the training points from the plots of the analysis with uncontrolled strains. Although again with the possibility of being misleading, each micromodel solution is presented in the same plot, but the reader should keep in mind that the shown two-dimensional cuts for the exact solution and surrogate model predictions might not lie on the same plane.

4.6. Yield Stress Envelopes

Creating yield stress envelopes is a common design practice for materials involving plasticity. Since the problem investigated in this thesis involves plasticity as well, it is useful to create these envelopes under certain scenarios with surrogate modeling techniques. In this work, only the positive quadrant of the yield stress envelope is investigated, as shown in Figure 4.5.

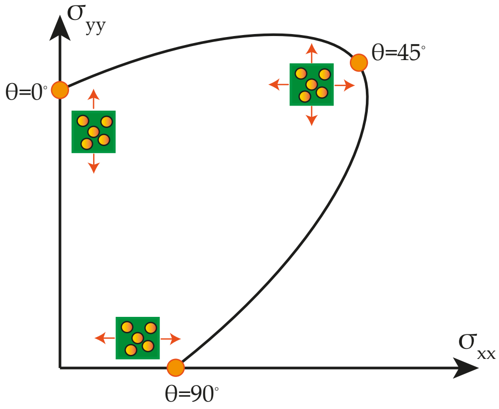


Figure 4.5: Illustration of the positive quadrant of the

It should be noted that the $(\sigma_{xx}, \sigma_{yy})$ tuples are obtained from the strain level of $\sqrt{\varepsilon_{xx}^2 + \varepsilon_{yy}^2} = 0.04$. Moreover, θ in Figure 4.5 is given by,

$$\theta = \arctan\left(\frac{\sigma_{xx}}{\sigma_{yy}}\right). \tag{4.3}$$

5 | Results and Discussion

This chapter is dedicated to present the results obtained from the implementation of various surrogate model creation techniques provided in the previous chapters. As mentioned before, since the input and output space is three-dimensional, visual results are provided only with one of the relevant input and output components in two-dimensional plots.

The first section is dedicated to the results of the simulations with controlled strain paths (for a given loading scenario strain paths are same for both fidelities), whereas the second section is dedicated to the results from a much more realistic analysis in which the solutions are allowed to move more freely in the strain space (for a given loading scenario strain paths might not be same for both fidelities), given that certain global constraints are applied to the problem being solved. In the final section, the time gain with the utilization of this method is investigated.

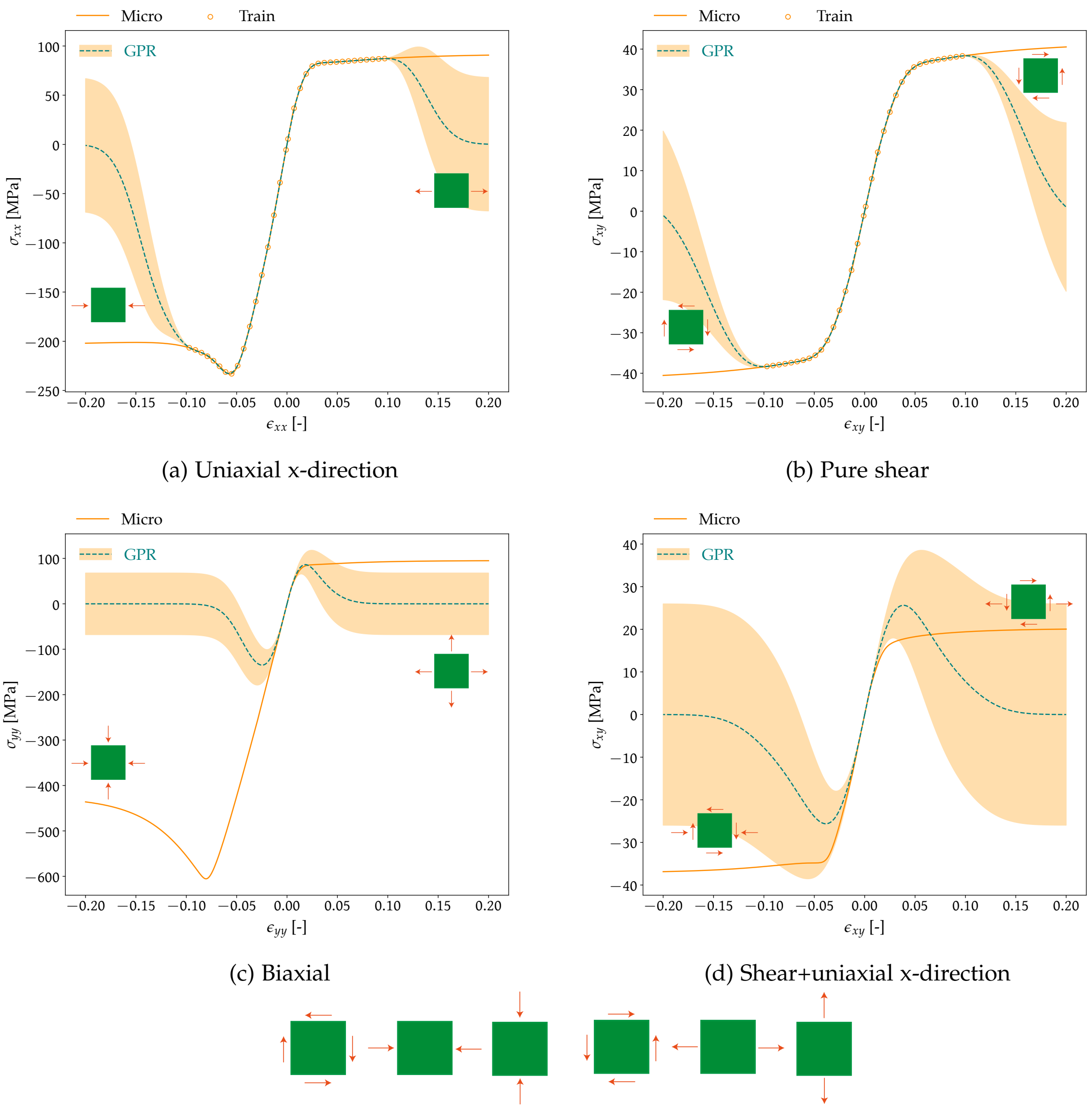
5.1. Controlled Strain Paths

As mentioned before this type of analysis will give way to total control over the feature space for the GP models and strain paths for some cases are the same for both fidelities (Figure 4.2a). Although it is not realistic to have that much control over the strain path, for investigating the capabilities of a surrogate model in a controlled environment, it is reasonable to do so.

5.1.1. Investigating the Capabilities of GPR

Before going into the multiGPR setting, understanding of the GPR model is crucial. This subsection is dedicated to understanding the behavior of the GPR surrogate model for various training scenarios and the predictions for various loading cases.

A surrogate constitutive model for the homogenous Melro material created with GPR trained with unidirectional loading cases is shown in Figure 5.1. It is evident that predictions for the trained cases given in the training interval are accurate. However, for simulations outside of the training domain (*e.g.* unseen strain levels for trained loading cases and untrained loading cases) the predictions are not good. When simulations for the surrogate model are extended beyond the training domain intentionally, predictions tend toward the prior of the GPR model, which is zero in this case as can be seen on the intervals [0.1,0.2] and [-0.2,-0.1] in Figures 5.1a and 5.1b. Another observation can be made from Figures 5.1c and 5.1d. It is clear to see that where there is no training data available from the loading cases that the surrogate model is utilized, the GPR model gives poor predictions. Moreover, in this case the tendency of the GPR model to go to its prior can be observed more clearly.



(a) Uniaxial x-direction

(b) Pure shear

(c) Biaxial

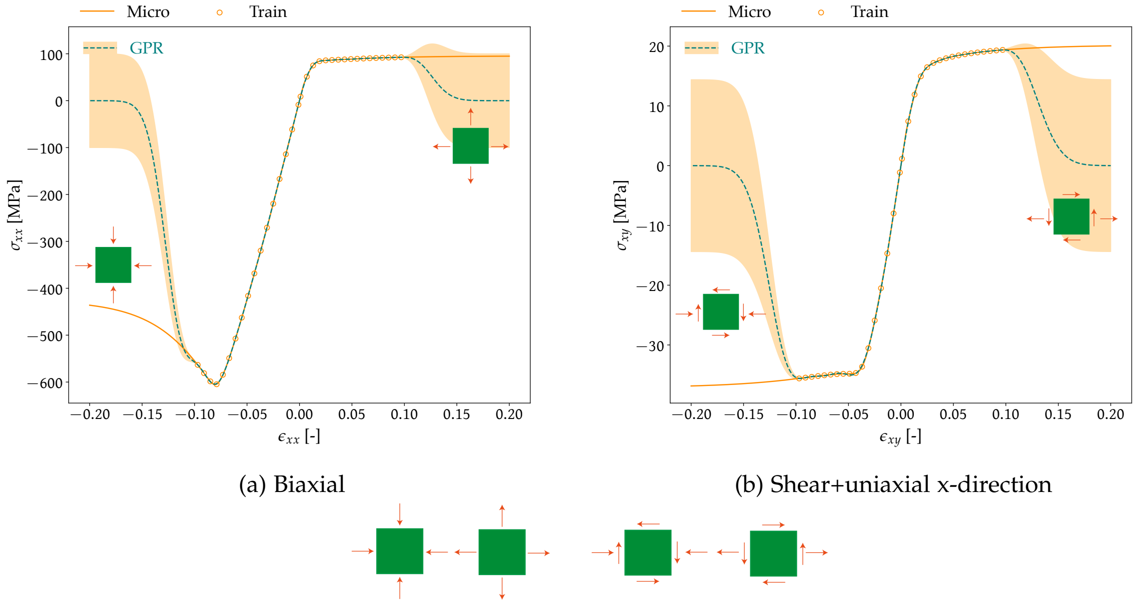
(d) Shear+uniaxial x-direction

(e) Training cases ($\mathcal{D} = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

Figure 5.1: Various load case predictions for surrogate model created with GPR trained with homogenous Melro material results obtained from only unidirectional load cases.

After additional loading cases are added to the training set an increased improvement in prediction capabilities can be observed by comparing the Figures 5.1c and 5.1d with Figures 5.2a and 5.2b.

Upon observing Figures 5.1 and 5.2, it can be concluded that for the GPR model to be an effective tool, training data should be collected thoroughly from various loading cases. Moreover, it is impossible to obtain good predictions outside the training interval for a given loading case, due to zero mean assumption made in the GPR and the purely data-driven nature of the model. This deficiency of GPR models makes utilization of the multiGPR model an interesting option.



(a) Biaxial

(b) Shear+uniaxial x-direction

(c) Additional training cases to Figure 5.4g ($\mathcal{D}_{add} = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

Figure 5.2: GPR model predictions with biaxial tension and compression loading cases added to the training set of example in Figure 5.1.

5.1.2. Investigating the Capabilities of multiGPR

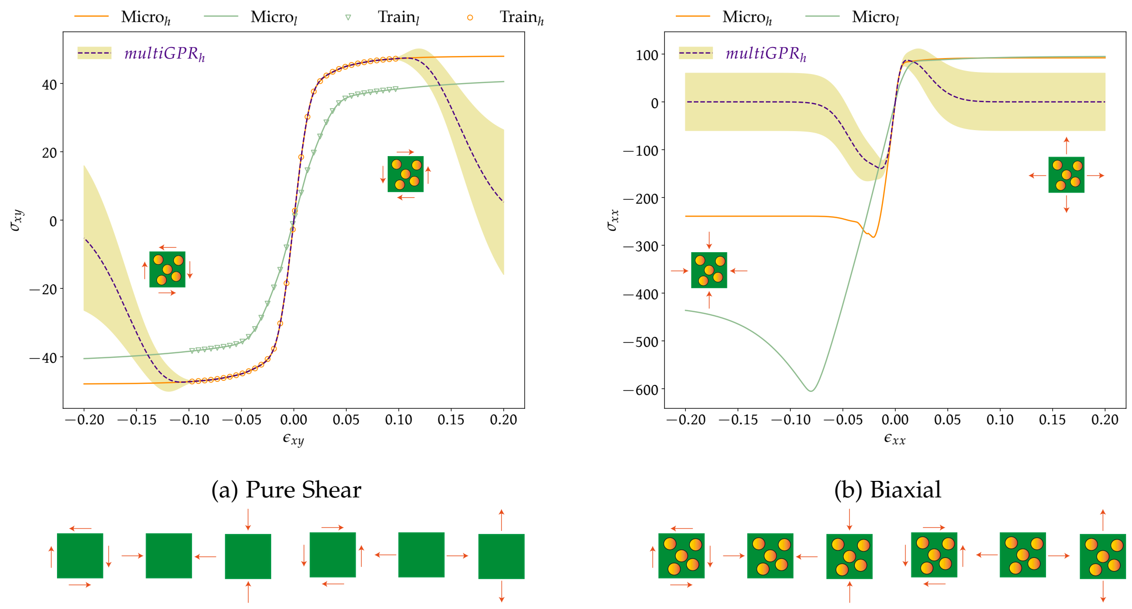
Although finding the full-order solution for a homogenous material is fairly trivial, it takes a significant amount of computational effort for a micro-heterogenous material. Thus, understanding the capabilities of multiGPR is important to utilize the scarce information obtained with high-fidelity solutions. Thus, this section is dedicated to understanding the limitations of the multiGPR method for the aforementioned fidelity structure (homogenous material as low-fidelity and 1-fiber RVE as high-fidelity) presented in the previous chapter by collecting the low-fidelity information thoroughly and utilizing them in combination with few high-fidelity solutions as a multiGPR model.

Training both fidelities of the multiGPR model with the same load cases on the same regions:

First, a similar case introduced in the previous section, where only unidirectional training data is provided in the [0,1] interval is investigated. Figure 5.3 shows the prediction capabilities of the multiGPR model for both seen and unseen cases. As can be seen from Figure 5.3a the prediction capabilities outside away from the training points multiGPR is as poor as the GPR method since the predictions tend towards the prior as in the case of GPR model. Another similarity of multiGPR and the GPR can be seen in Figure 5.3b where for an untrained loading scenario similar, with predictions going towards the prior.

Training the multiGPR model with additional information coming only from the low-fidelity:

It should be noted that the aforementioned similarity between GPR and multiGPR is an expected result of the given training set. Since there is no information available outside the training region, all the predictions are expected to return to a zero prior after a certain point. This can be avoided by providing lower-fidelity information as shown in Chapter 3.



(a) Pure Shear

(b) Biaxial

(c) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(d) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

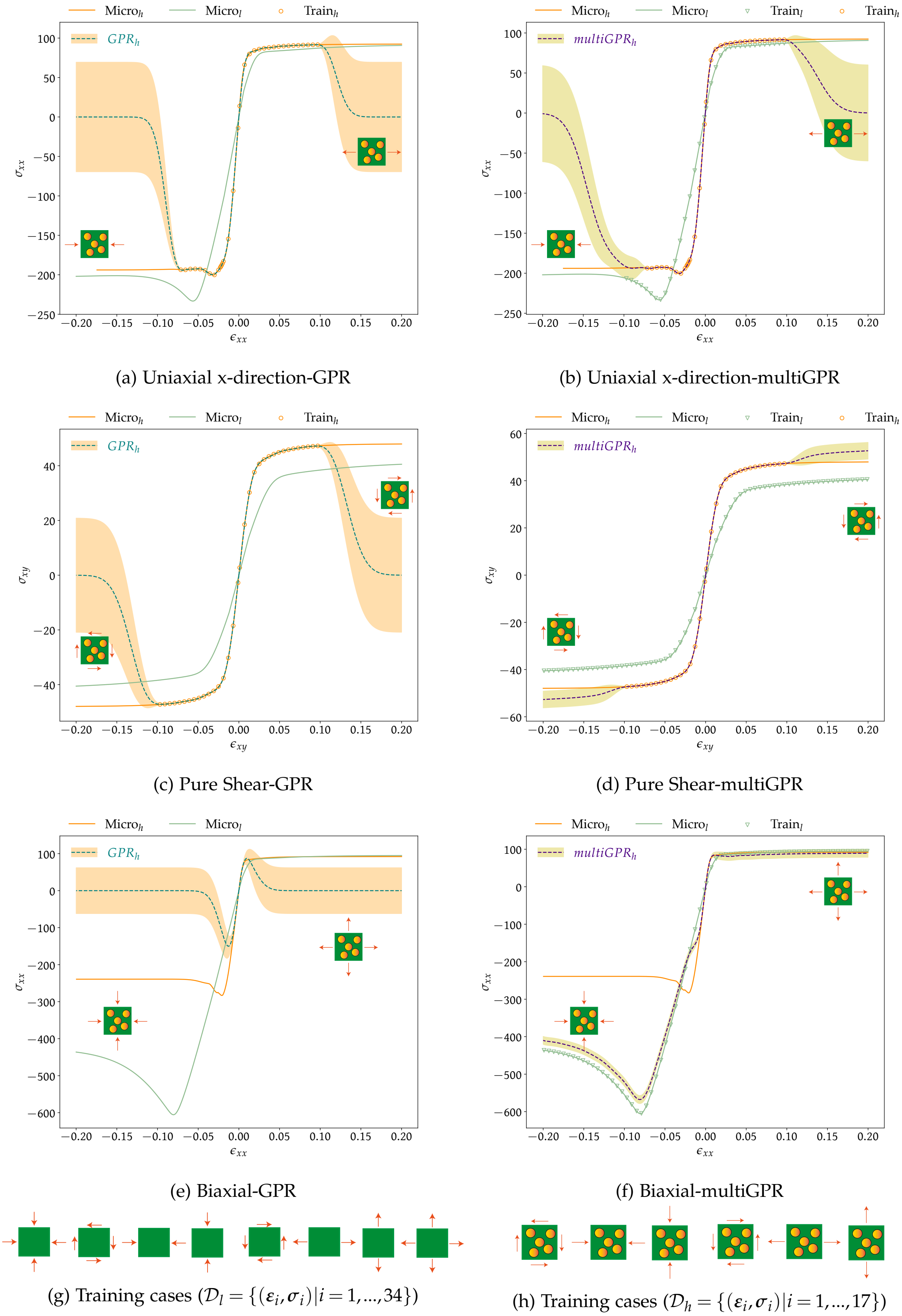
Figure 5.3: multiGPR model predictions for a trained and untrained load case for the given training scenario.

Introducing additional lower fidelity information to the already existing training interval with an additional loading scenario of biaxial loading results in the prediction results shown in Figure 5.4. On the same figure a comparison of the GPR model and the multiGPR model is made for the same loading case. As can be observed by comparing Figures 5.4c and 5.4d there is a visible improvement when utilizing multiGPR when extra low-fidelity information is included in the part of the training domain where there is no information coming from the higher fidelity. However, as the results start to deviate from the micromodel solution the variance bands increase. This increase indicates the awareness of the GP method about its error.

Moreover, comparing Figures 5.4e and 5.4f it can be seen that prediction results of the multiGPR model for the positive biaxial loading case are good compared to GPR model, whereas the negative biaxial loading results in poor predictions compared to actual values. It should be noted that although prediction results are poor with respect to micromodel solutions, it is still much more informative than the GPR model for the same case. Instead of the predictions going to a prior assumed before conducting any simulations, now the predictions follow a path that is similar to the low-fidelity information. This representative difference is coming from the fact that the low-fidelity is not representative of the high-fidelity model selected for the given case. In other words, if the correlation between the low- fidelity and high-fidelity is poor, one should not expect a good prediction result for all cases by just including lower fidelity training points.

Yield stress envelopes:

Prediction capabilities can be investigated further with the predictions for more complex loading scenarios by creating yield stress envelopes. It should be noted that only the positive quadrant of the yield surface envelope shown in Figure 4.5 is simulated and investigated for different training scenarios.



(a) Uniaxial x-direction-GPR

(b) Uniaxial x-direction-multiGPR

(c) Pure Shear-GPR

(d) Pure Shear-multiGPR

(e) Biaxial-GPR

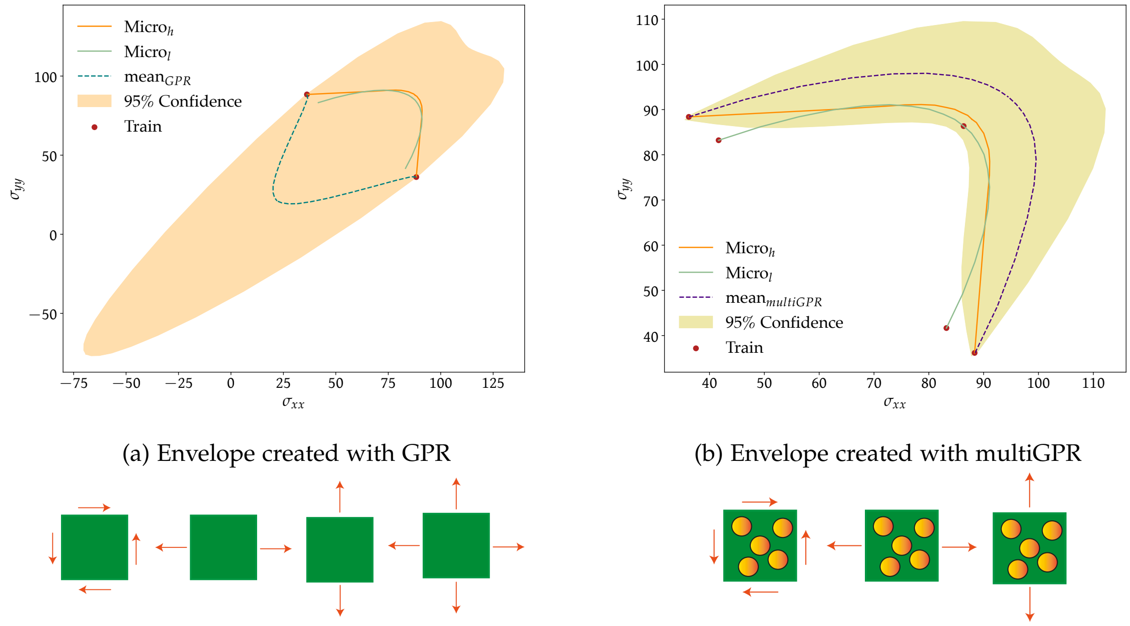
(f) Biaxial-multiGPR

(g) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 34\}$)

(h) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

Figure 5.4: multiGPR model predictions for a trained and untrained load case for the given training scenario.

A case where only positive loading scenarios are involved in both training and prediction is investigated in Figure 5.5. In the given envelopes it is important to note that training is not done on the envelope values themselves. In other words, the envelopes are created utilizing the predictions of trained models with the given training load cases. So, every point seen on the envelopes is coming from a prediction corresponding to the predetermined strain level of $\sqrt{\varepsilon_{xx}^2 + \varepsilon_{yy}^2} = 0.04$. Red dots in the plots represent only the training load cases. That is the reason for a different color scheme other than high and low-fidelity exact solutions to be utilized. It is exciting to observe the overall improvement by adding 4 more load cases to the training set from the low-fidelity which is obtained from a homogenous material model with little to no computational cost. Even with the introduction of 4 loading cases from the lower fidelity, the predicted envelope is promising. In addition, the upper and lower limits introduced with the uncertainty bands provide a safety net for the predicted values to be utilized as always.



(a) Envelope created with GPR

(b) Envelope created with multiGPR

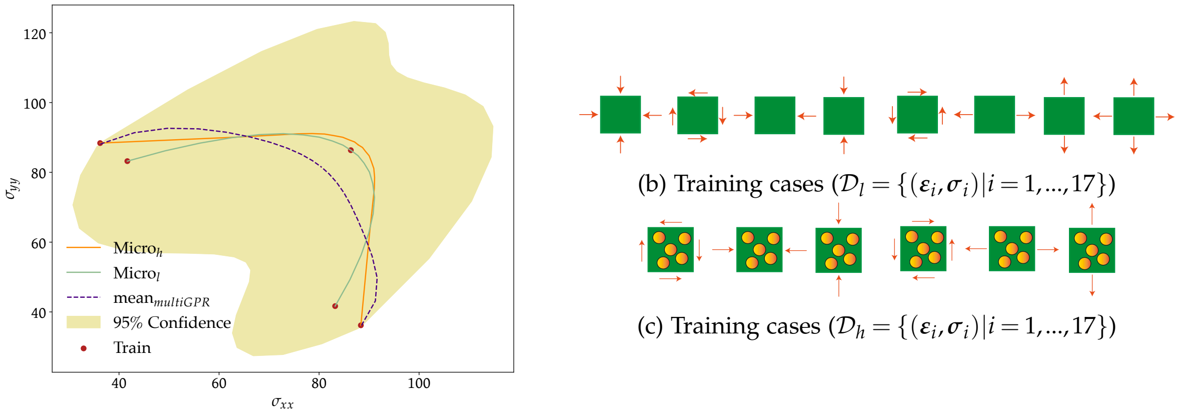
(c) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(d) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

Figure 5.5: Yield stress envelopes for GPR and multiGPR models for a training scenario involving only positive load cases.

After observing the promising results for a model trained with only the positive region of the envelope. it is interesting to look at what happens when more load cases outside of the prediction region are added to the training set. This case is visualized in Figure 5.6. Noting that a thorough training procedure is employed for obtaining both envelopes shown in Figures 5.5b and 5.6a (Maximization of evidence is done with 50 random restarts and training phase is repeated 10 times to make sure the maximized LML obtained from the optimizer is not on a local maximum.). Thus, the only viable explanation for the observed difference between the predictions caused by two different training sets is the inability of the multiGPR method to produce a good prediction in an extrapolation case with varying correlations between load cases. This correlation difference between various loading scenarios can be observed clearly in Figure 5.4. This variation of results is expected due to the discussion presented in Chapter 3 regarding ρ.

Although results seem to vary based on the training cases included in both fidelities from which ρ is learned, it is evident that predictions are still more promising than the ones generated with GPR in Figure 5.5a. Thus, it can be deduced that it is still quite an achievement that by providing a limited amount of high-fidelity cases a reasonably good prediction is obtained.



(a) Envelope created with multiGPR

(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

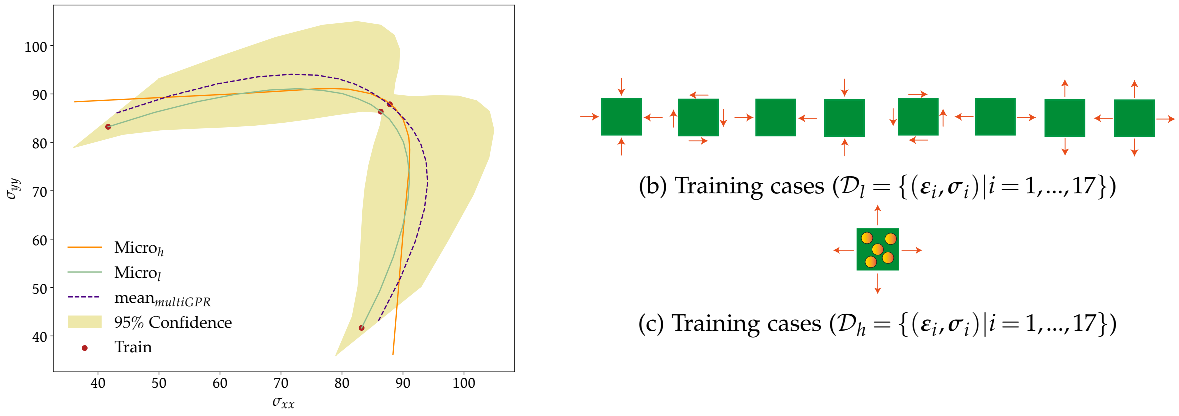
Figure 5.6: Yield stress envelope for multiGPR model for a training scenario involving both positive and negative load cases.

After acknowledging the dependency of the predictions to the training set that is utilized, it is interesting to investigate the behavior of the predictions by collecting the high-fidelity locally. As mentioned before there is a widely recognized problem of curse of dimensionality associated with the GPR methods. In other words, GP models are limited in their ability to incorporate huge datasets without utilizing a method to circumvent this bottleneck. Thus, the investigation of the prediction capabilities with local information inclusion is an exciting case and is especially appealing in the setting presented in [42], where constitutive models are created on the fly. Here, we assume there is low-fidelity information already present and the constitutive relation is sought in the near vicinity of $\theta = 45°$. Then, including just biaxial results in the high-fidelity results in significant information gain. Looking at Figure 5.7 it is clear to see that even with only one loading case from the high-fidelity solution is enough to create a representative envelope. As expected the accuracy of the predictions is higher near the vicinity of the added load case of $\theta = 45°$.

Moreover, as can be seen from Figure 5.8 the response can be improved by including additional low and high-fidelity loading scenarios at $\theta = 26.5°$ and $\theta = 63.4°$. It can be seen that the addition of the low-fidelity and high-fidelity near the desired loading case improves the solution tremendously for a large region of the yield stress envelope.

In light of this observation, in a case where the low-fidelity information is readily available in a surrogate format (as a trained low-fidelity GPR) substantial information can be gained by just conducting a multiGPR model with limited amount of load cases from the high-fidelity. Since the low-fidelity surrogate is already available, the only cost of obtaining the high-fidelity surrogate lies in volume of training points coming from the high-fidelity data.

Then, another important limit case can be investigated where the low-fidelity information is ex-

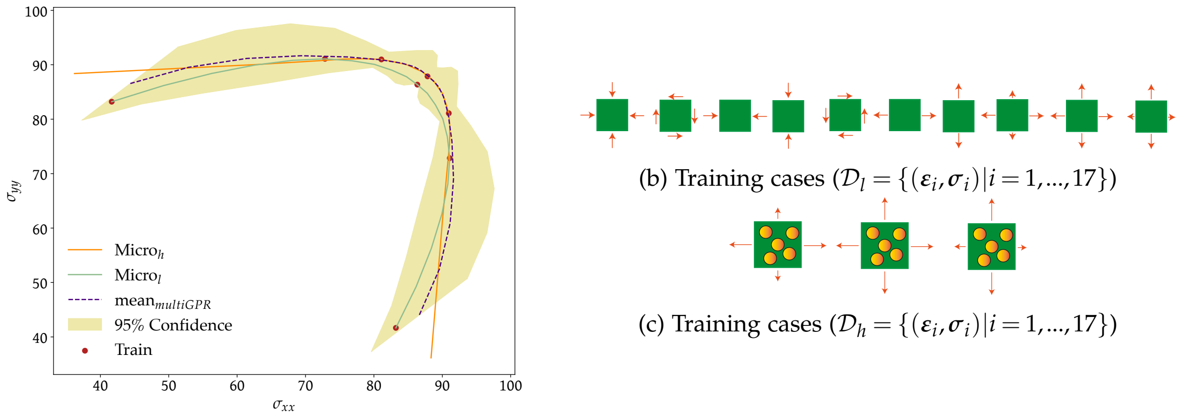


(a) Envelope created with multiGPR

(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

Figure 5.7: Yield stress envelope for multiGPR model with local training scenario involving both positive and negative load cases in low-fideilty.



(a) Envelope created with multiGPR

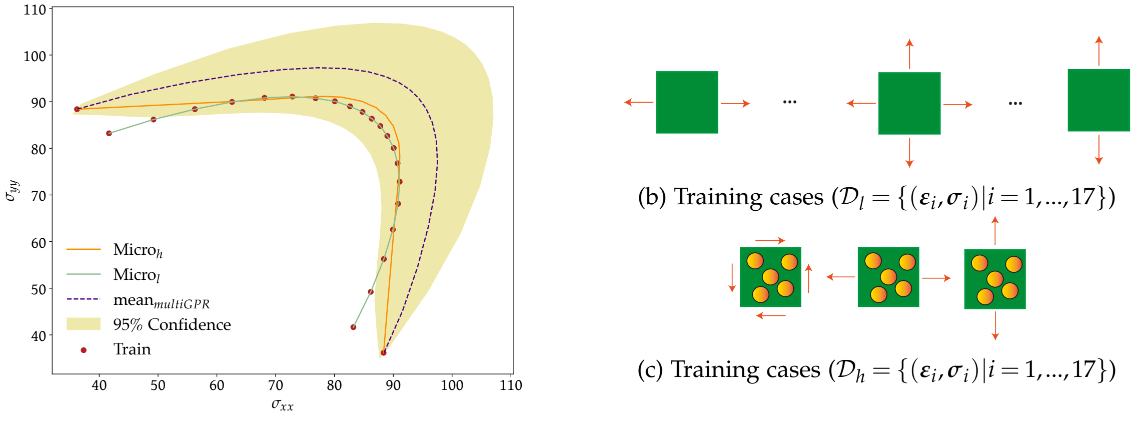
(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

Figure 5.8: Yield stress envelope for multiGPR model with dense local training scenario involving both positive and negative load cases in low-fideilty.

haustively collected and the high-fidelity information is scarce. This case can be seen in Figure 5.9 and it can be compared with the predicted envelope in Figure 5.5, since both predictions are made with training cases only made in the positive quadrant. By comparing both of the mentioned figures it can be seen that the increased density of low-fidelity training load cases has a little observable effect on the predictions with slightly improved confidence bounds.

The case where densely collected low-fidelity has no major impact can be observed by comparing the predictions shown in Figures 5.10 and 5.7. Even when there is only one local training load case present the effect of low-fidelity information inclusion has little effect on the bounds of the prediction. Thus, it can be concluded that when the low-fidelity is representative enough the additional data does not have a big influence on the prediction results. This is another promising result since there is no need to collect multiple loading cases even from a computationally less taxing low-fidelity model. This might prove to be useful, especially if the data collection and training is conducted in an online

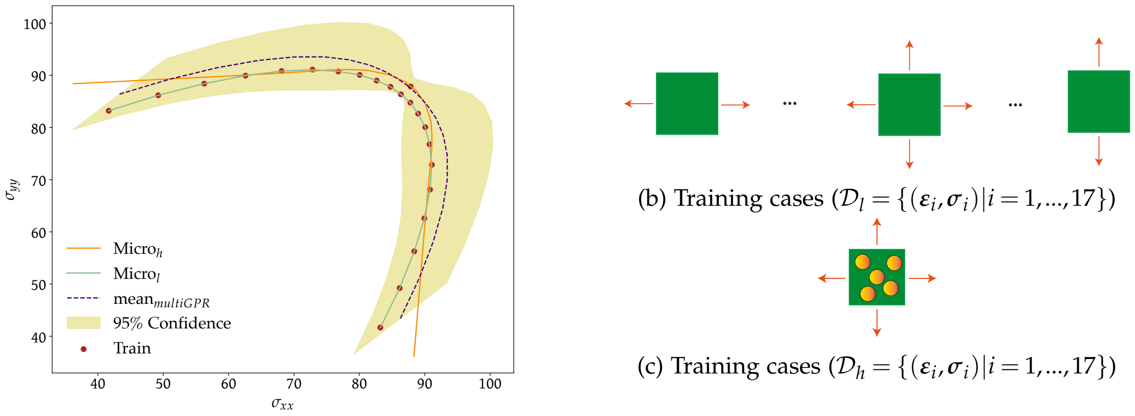


(a) Envelope created with multiGPR

(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

Figure 5.9: Yield stress envelope for multiGPR model with training scenario involving a densely collected low-fidelity on the positive quadrant.



(a) Envelope created with multiGPR

(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

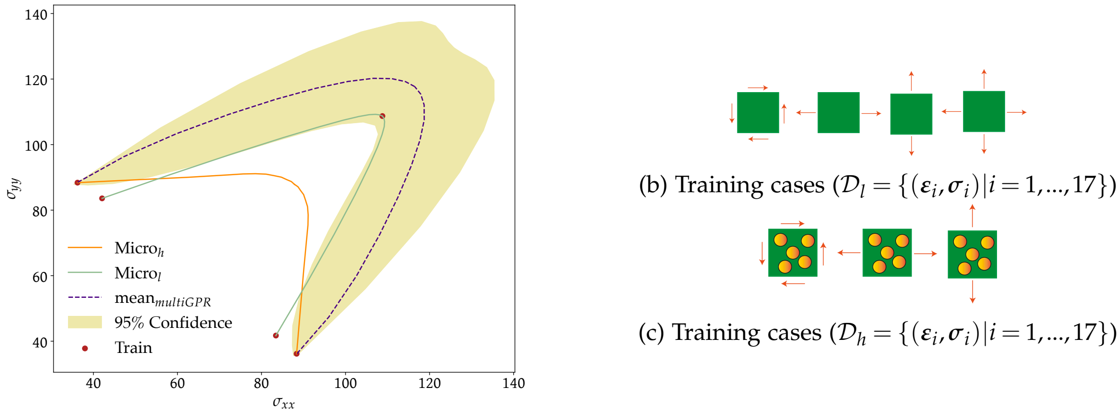
Figure 5.10: Yield stress envelope for multiGPR model with training scenario involving a densely collected low-fidelity on the positive quadrant.

framework.

5.1.3. Importance of the Low-fidelity Information Accuracy

While trying to get a cheap low-fidelity other surrogate modeling techniques mentioned in Chapter 2 can be employed in order to get computationally cheap but unrepresentative training points. For the sake of this discussion, let's assume that even the low-fidelity information can only be obtained for a high computational cost but there is a badly trained neural network readily available. Information regarding the Neural Network that is used to generate this envelope is given in Appendix B.

When utilized in a region where there is no information coming from the high-fidelity it is highly risky to use this information as can be observed in Figure 5.11. Looking at selected training load cases the observed case is expected, since the correlation is established by only unidirectional cases, and with the additional biaxial load case the low-fidelity is utilized to create a poorly predicted envelope.

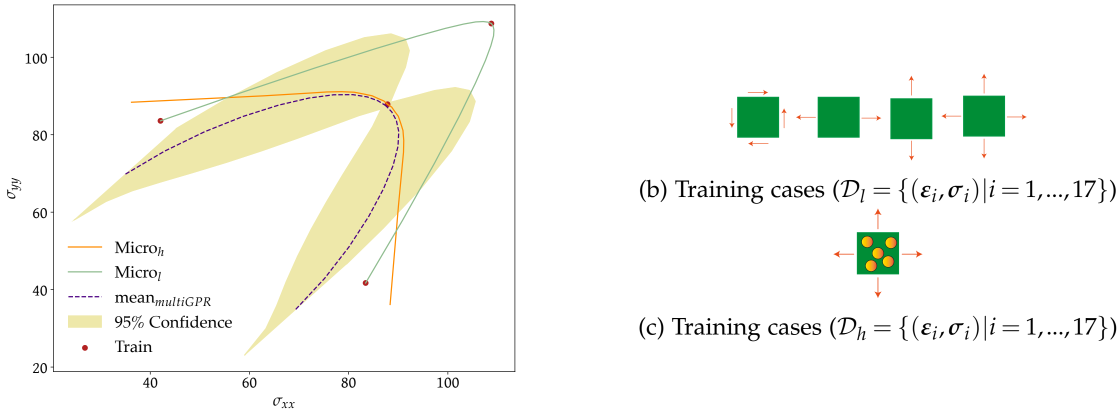


(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1,...,17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1,...,17\}$)

(a) Envelope created with multiGPR

Figure 5.11: Yield stress envelope for multiGPR model with training scenario involving a densely collected low-fidelity on the positive quadrant.



(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1,...,17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1,...,17\}$)

(a) Envelope created with multiGPR

Figure 5.12: Yield stress envelope for multiGPR model with training scenario involving a densely collected low-fidelity on the positive quadrant.

However, a more interesting finding is that due to Markov like property of the multiGPR method, which implies that there is no information to be gained from the low-fidelity when information is available from the high-fidelity. This behavior can be observed in Figure 5.12. As can be seen, although the low-fidelity is unrepresentative its effect does not affect predictions with the multiGPR model. Moreover, the near vicinity of training scenarios is predicted reasonably well. As expected the predictions lose their representativeness further away from the loading scenarios that are used for training.

As a result of this discussion, it is clear to see that even a poor low-fidelity in certain regions can be compensated by selecting the high-fidelity sampling regions wisely. This makes the utilization of this method with various other surrogate modeling techniques possible given that the deficiencies of the other surrogate methods are understood properly prior to utilization.

5.2. Uncontrolled Strain Paths

After seeing the performance of the multiGPR model in a setting where all the feature space variables are controlled, a more realistic case where the material is allowed to deform freely is investigated. It should be noted that because of this path dependency of the solution it is not reasonable to show the training points on the plots anymore, that is why in this section training points are not shown explicitly. However, the number of training points taken will be mentioned with the related training load cases. Even showing the exact solution on the same plot is not relevant, but in order to create a frame of reference, it is included. Moreover, this time extrapolation capabilities in the same loading scenario will not be investigated, so the training and predictions are constrained in the strain intervals of [0,1]

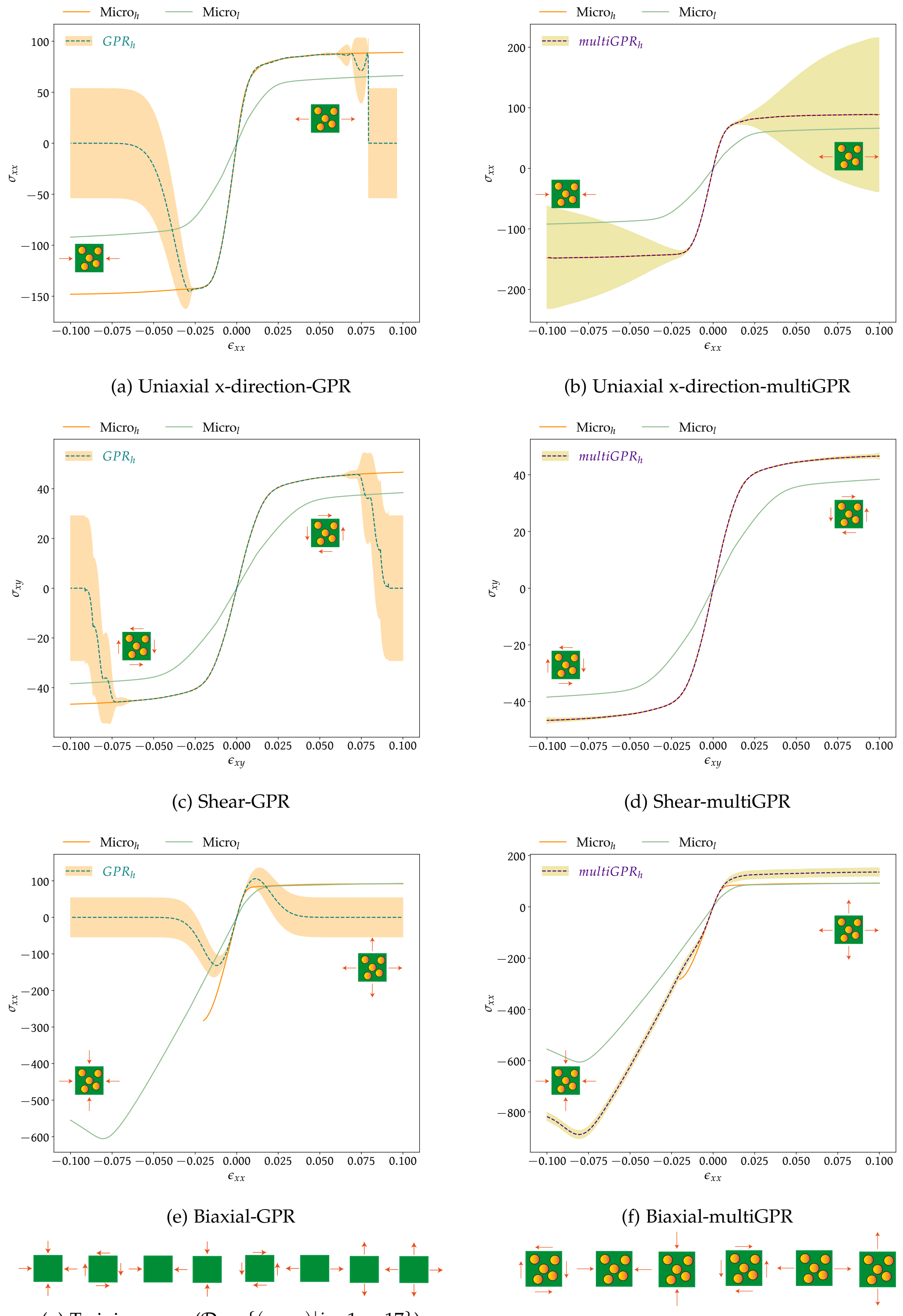
Here, we consider the same training and prediction cases given in Figure 5.4 for the uncontrolled strain paths. From Figure 5.13 by just comparing the GPR model to the multiGPR model the increased prediction performance of the multiGPR model is evident for the uncontrolled strain path case as well. However, close observation shows that there is an increasing variance that can be observed clearly in Figure 5.13b. Moreover, there is a divergence observed near the 0.1 strain level. Furthermore, the same problem is observed in Figure 5.13a around the same region as well. Furthermore, there is clear evidence that the GPR solutions even in the trained cases have bad predictions suddenly after certain strain levels. Although some problems are present for trained loading scenarios multiGPR model have similar and expected predictions for shear and biaxial loading to the controlled strain path case introduced in Figure 5.3. In other words, observed problems are not evident in shear and biaxial loading cases. This is because the shear is not affected by the Poisson effect and the biaxial loading has a constrained strain path inherently.

The overprediction that can be seen in Figure 5.16f by quite a margin in biaxial loading scenario compared to a controlled strain case should be highlighted. It can be seen that introducing more degrees of freedom to the system causes the correlation between multiple loading cases to change which results in a different correlation scaling parameter. This inevitably affects the predictions in the untrained region. However, although it is overpredictive the results of the multiGPR are still favorable compared to the ones given by the GPR model.

Due to similar natures of both controlled and uncontrolled strain paths, from now on the discussion will focus on the problems encountered, namely increasing error bounds and divergence problems.

5.2.1. Divergence and Poor Predictions

Since the stiffness matrix is also obtained from the GP models, upon facing divergence or any kind of unexpected behavior from the solver the first reason that comes to mind is problems related to the creation of the stiffness matrix. It can be observed in Figure 5.14 that when predictions are made



(a) Uniaxial x-direction-GPR

(b) Uniaxial x-direction-multiGPR

(c) Shear-GPR

(d) Shear-multiGPR

(e) Biaxial-GPR

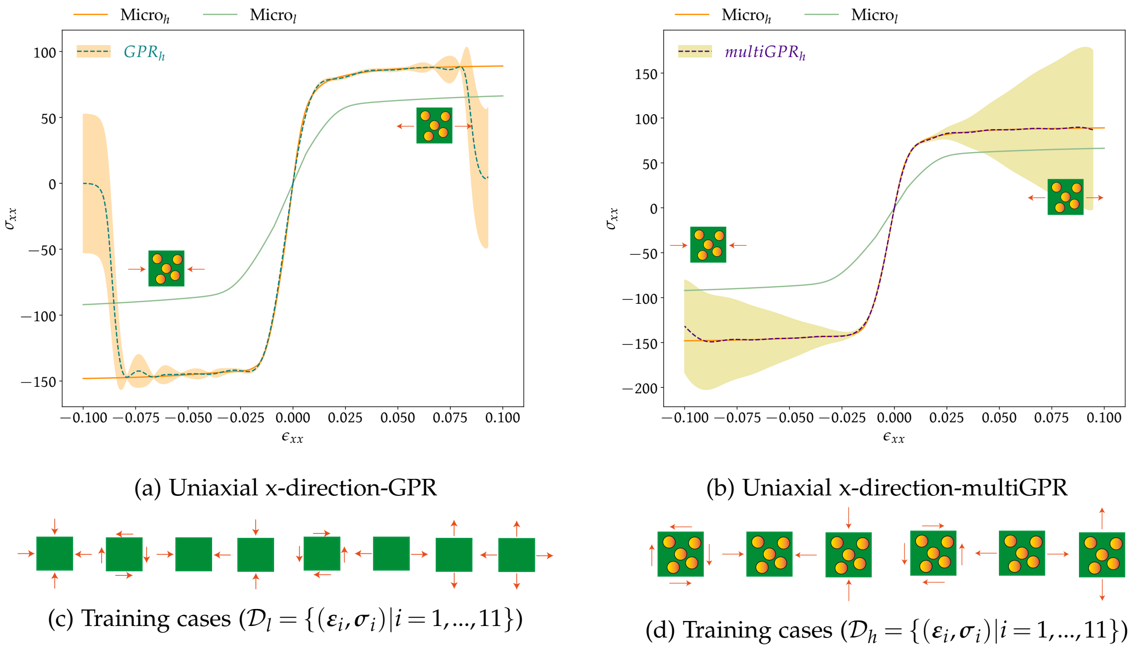
(f) Biaxial-multiGPR

(g) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(h) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

Figure 5.13: Comparison of GPR and multiGPR model predictions for trained and untrained load cases for uncontrolled strain paths.

poorer compared to the ones shown in Figure 5.13 by decreasing the number of training points then the divergence problem starts at an earlier stage. A similar trend is observed for unrepresentative predictions as well. As expected since more data is utilized during co-kriging with multi-fidelity information the changes in the multiGPR model are not as drastic as in the case of the GPR models.



(a) Uniaxial x-direction-GPR

(b) Uniaxial x-direction-multiGPR

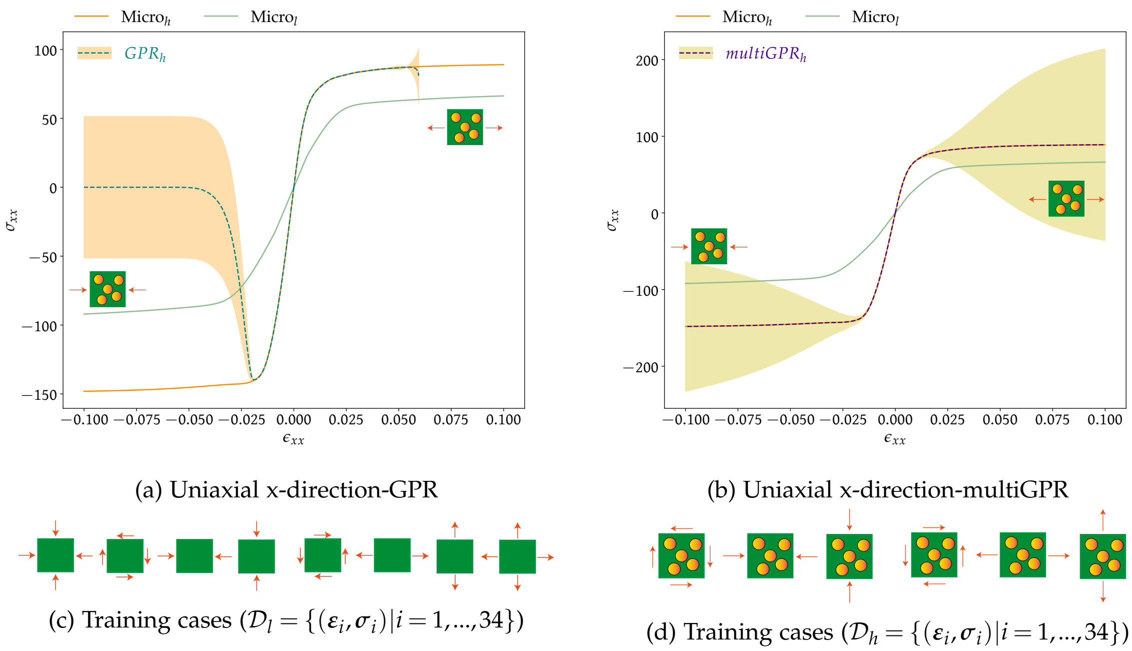
(c) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 11\}$)

(d) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 11\}$)

Figure 5.14: Comparison of GPR and multiGPR model predictions for trained load cases for uncontrolled strain paths with poorer surrogate model.

The predictions get poorer when the number of training points is decreased, then it might be automatically assumed that increasing the number of points utilized in the GP conditioning can give much better predictions. However, this is not necessarily the case as it can be seen in Figure 5.15, where the divergence occurs at a much earlier stage of the GPR model and the divergence problem in the multiGPR is still there near the end of the simulation around 0.1 strain level.

Thus, upon an investigation of the poor surrogate model and the increased deviations and fluctuations observed for GPR model predictions, the derivative inclusion in the surrogate modeling is employed by means of co-kriging. Using derivative information is especially advantageous since the stiffness matrix is readily available from the micromodels. Prediction results of co-kriging with derivative information can be seen in Figure 5.16. It can be concluded that with the utilization of this technique the divergence problems for the trained loading cases seemed to be bypassed for both of the problematic cases, namely uniaxial loading in the x-direction and pure shear. This can be explained by considering the information that derivative inclusion gives to the surrogate models, where an abundance of information is provided to the model in all directions in the near vicinity of the training points which not only increases the quality of the predictions in a certain direction but also in other directions. In other words, additional points only give information in a certain direction whereas derivative inclusion introduces more spatial awareness to the model. However, although this method circumvents the divergence problem, there is a clear bottleneck of the computational cost associated with it as mentioned in Chapter 3.



(a) Uniaxial x-direction-GPR

(b) Uniaxial x-direction-multiGPR

(c) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 34\}$)

(d) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 34\}$)

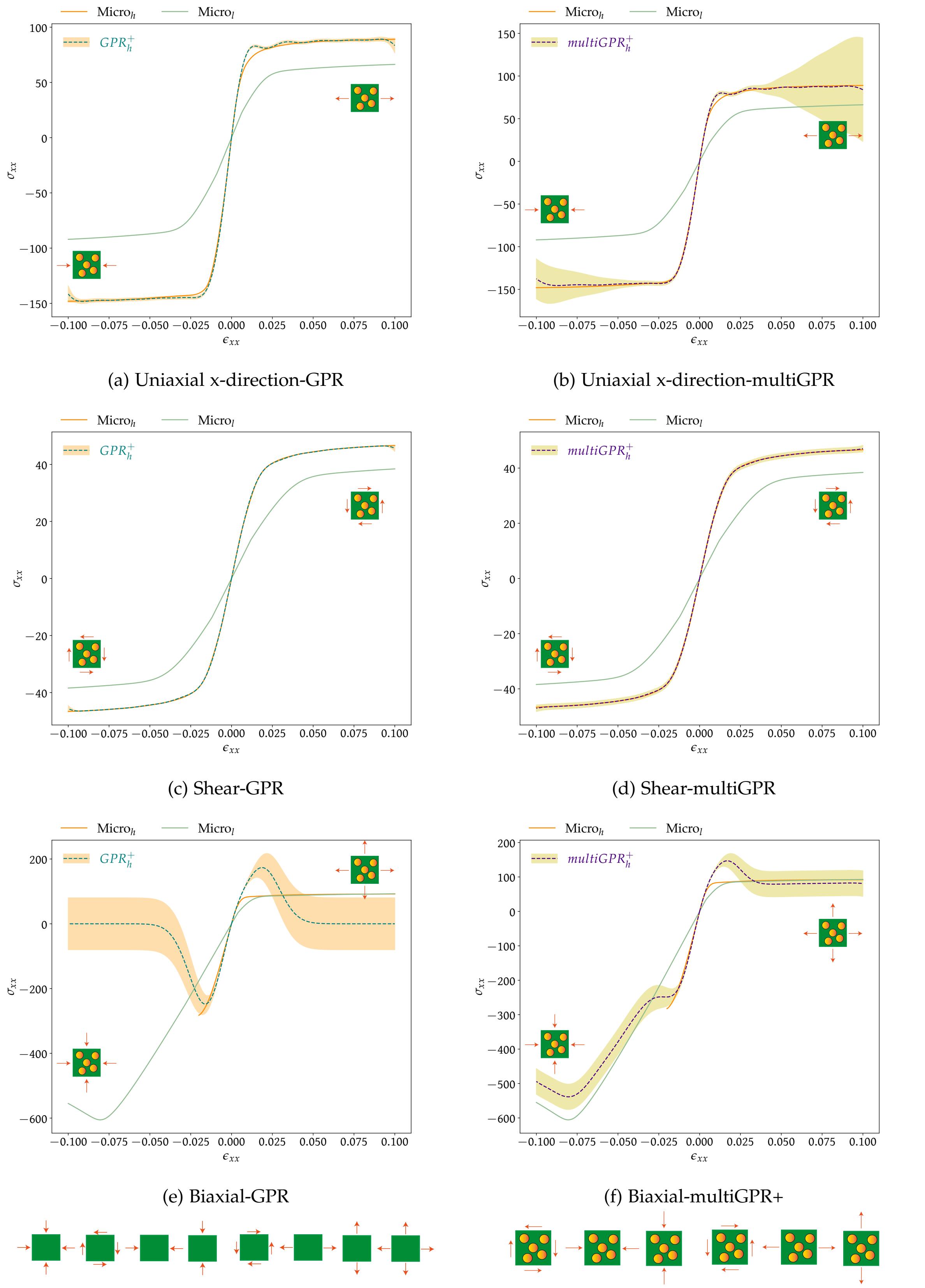
Figure 5.15: Comparison of GPR and multiGPR model predictions for trained load cases for uncontrolled strain paths with poorer surrogate model.

Another interesting observation that can be made in Figure 5.16 is slight over predictions near the start of plasticity for every surrogate model, this problem is solved by adaptively selecting the training points in Appendix C. Moreover, the biaxial loading scenario where there is no information from the high fidelity is seemingly much better compared to GPR and multiGPR results, since the over prediction is not that much although there is quite a big jump near the points where the plasticity starts, due to lack of information originating from the training point selection procedure.

5.2.2. Increasing Uncertainty Bounds

Increasing error bounds are an indication that the solver is taking the model to a place in the feature space away from the training points. Moreover, this observation is supported by the fact that increasing error bounds are only observed in uniaxial loading makes it clear that due to Poisson effect solutions at some point deviate from the strain path that the model is trained for. Upon investigation, it is found that this is exactly the case and this stems from the fact that the high and low-fidelity results are not on the same strain path as shown in Figure 5.17. This indicates that this variance is intrinsically builds up each step due to the utilization of low-fidelity approximations in the high-fidelity predictions. This can be clearly seen formulations given in Equations 3.31 and 3.32. Thus, by simply inducing the same strain path that the high-fidelity sees to the low-fidelity as shown in Figure 5.18. The training of both fidelities can be done with data taken from the same strain path.

The results for the case for which the training data is provided from the same strain path for both fidelities is shown in Figure 5.19 avoiding the GPR counterparts of the predictions since there is no change in the GPR model. This is simply an artifact caused by the fact that different fidelities can have different Poisson ratios. Looking at these improved solutions it can be concluded that inherent



(a) Uniaxial x-direction-GPR

(b) Uniaxial x-direction-multiGPR

(c) Shear-GPR

(d) Shear-multiGPR

(e) Biaxial-GPR

(f) Biaxial-multiGPR+

(g) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(h) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

Figure 5.16: Comparison of GPR$^+$ and multiGPR$^+$ model predictions for trained and untrained load cases for uncontrolled strain paths.

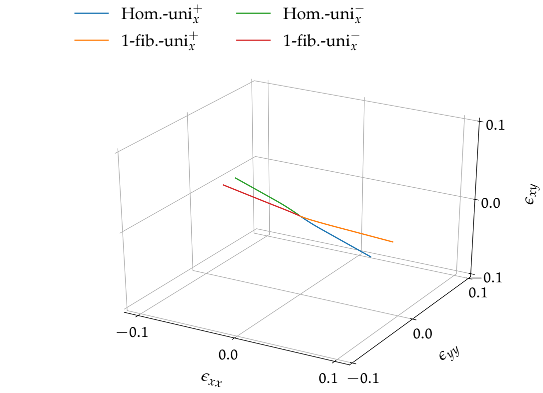


Figure 5.17: Uniaxial strain paths for low and high-fidelity with uncontrolled strain case.

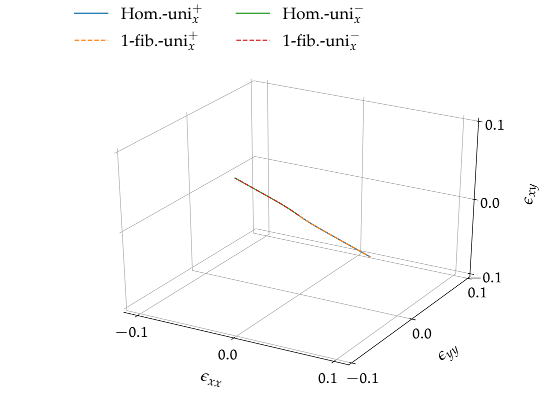
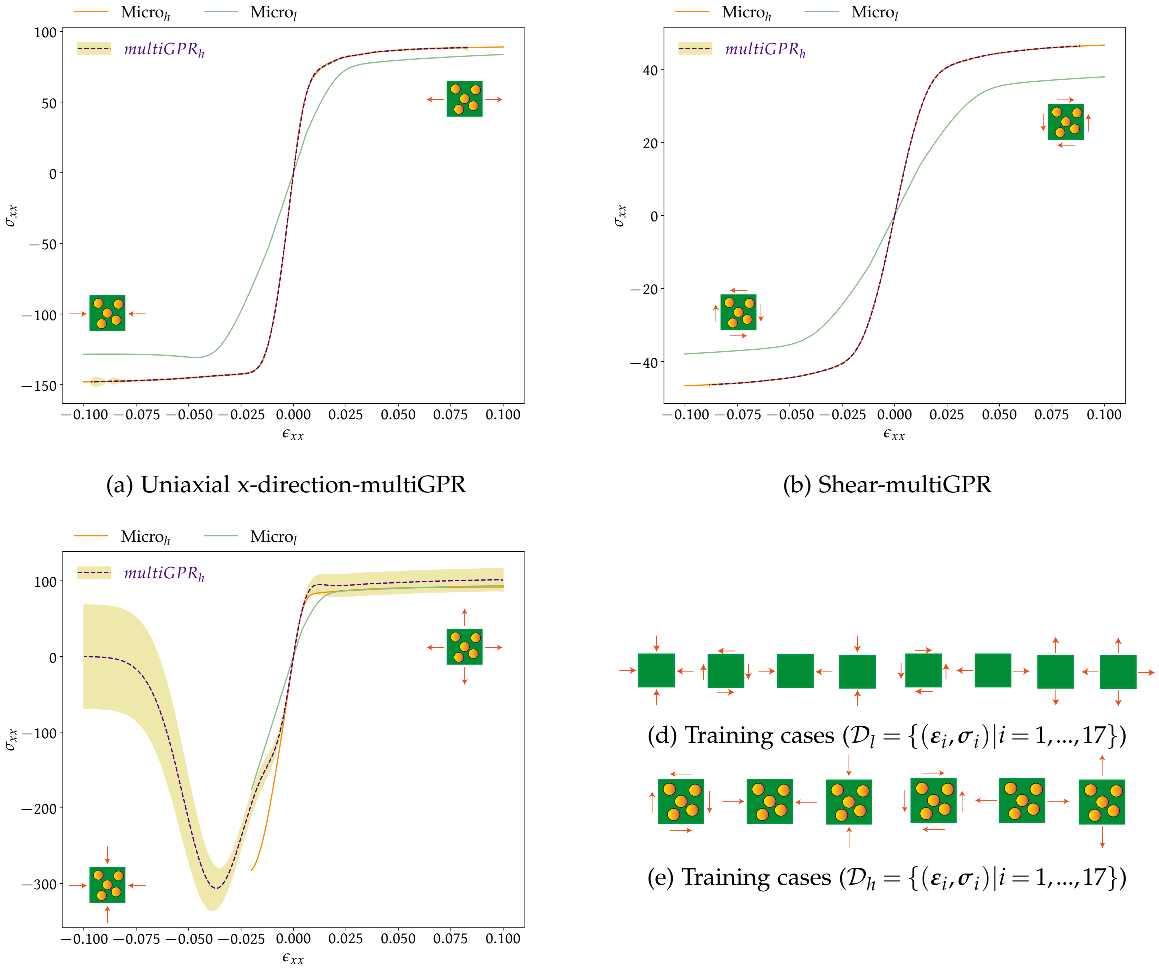


Figure 5.18: Imposing the strain path of high-fidelity to the low-fidelity.



(a) Uniaxial x-direction-multiGPR

(b) Shear-multiGPR

(d) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(e) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i) | i = 1, ..., 17\}$)

(c) Biaxial-multiGPR

Figure 5.19: multiGPR model predictions trained with matching strain paths of low and high-fidelity micromodels.

problems in the uniaxial direction regarding the increasing uncertainty bounds are prevented by collecting the low-fidelity training data by imposing the exact strain path seen by the high fidelity. Moreover, other loading scenarios do not show a significant difference other than the aforementioned and tackled problem of divergence since the Poisson's effect does not play a role in their behavior. Finally, the strain path that can be obtained from the high fidelity of biaxial loading in compression shown in Figure 5.19c is limited, in other words, low-fidelity information is limited to the point of the end of the high fidelity this time. That is the reason for observing a prediction going to the prior.

5.3. Performance of multiGPR

Prediction capabilities of multiGPR are investigated in the previous sections. Although there are some inherent problems, why it is useful to utilize this surrogate model is one of the questions that this work is trying to answer. As mentioned before FE^2 suffers from a computational bottleneck and this can be avoided by surrogate modeling techniques. As a surrogate modeling technique multiGPR has three major parts on the way to get results, namely snapshot collection, offline training, and predictions. This section investigates the time difference that is needed to create some of the envelopes shown in Section 5.1.2. Considering the two envelopes for the 1-fiber RVE as shown in Figure 5.6a, which is trained by a comprehensive dataset presented (multiGPR$_h$-general), and Figure 5.7a which is trained with a comprehensive low-fidelity, but the high fidelity involves a single training case (multiGPR$_h$-local). Surrogate modeling time comparison can be seen in Figure 5.20. It is evident from the big head start of a thoroughly trained multiGPR model that obtaining snapshots and training processes can lead to significant time consumption. It can be concluded that for the given case the micromodel solution is much faster than collecting and training a surrogate GP. However, although training and snapshot collection takes a considerable amount of time it can be seen that predictions obtained for the envelope with multiGPR have constant time for each step whereas the convergence issues hinder the micromodel considerably at some θ values. This is a good indication of the model's

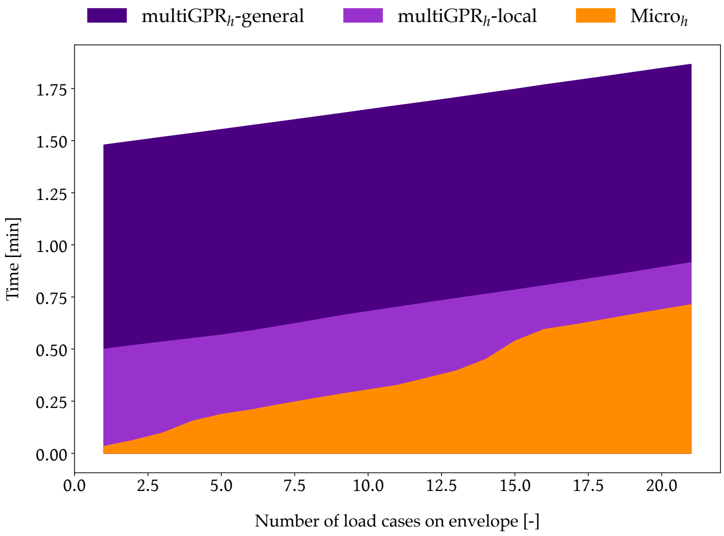


Figure 5.20: Cumulative time plot for 1-fiber RVE to create the envelopes shown in Figure 5.6a and 5.7a.

capability in other scenarios. Moreover, it is important to see that investigating the high-fidelity locally with minimal load cases is highly advantageous since the additional load case increases the training and snapshot collection by a factor of 3 in this case.

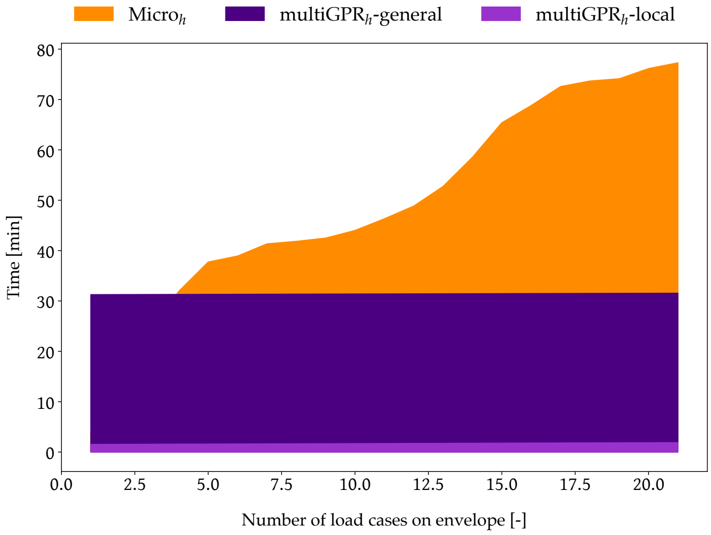


Figure 5.21: Cumulative time plot for 25-fiber RVE to create the envelopes shown in Appendix C.

In order to show the effectiveness of the surrogate model a more representative and computationally taxing RVE that includes 25-fibers embedded in the matrix is employed to conduct the same time analysis. Envelopes that are created can be seen in Figures C.1a and C.2a located in Appendix C. It can be seen from Figure 5.21 that the biggest computational cost goes to obtaining the snapshots since the number of training points included in the envelope creations is the same for both Figures 5.20 and 5.21. It can be observed that after the 4^{th} micromodel simulation the initial computational time spent for training and snapshot collection of the multiGPR model is exceeded. Furthermore, compared to the initial time consumption, predictions are obtained for almost no cost for the surrogate model. For the 25-fiber case, when a single loading case at the high-fidelity is used the whole training and snapshot collection process gets much more efficient. This can be seen as the multiGPR$_h$-the general case is 20 times slower compared to multiGPR$_h$-local.

Thus, it can be concluded that as the computational burden increases and the number of models needed to be run increases using multiGPR becomes advantageous. Especially if multiGPR is utilized locally meaning that only a limited number of load cases from high-fidelity are provided. However, as it can be seen from the related envelopes a price has to be paid in terms of accuracy with every single assumption that has been made until the prediction initiation.

6 | Conclusions and Future Work

6.1. Concluding Remarks

The main goal of this work was to investigate the capabilities of multiGPR as a surrogate constitutive model for composite micromodels. In order to do that multiple scenarios have been presented with a homogenous material as low-fidelity and a 1-fiber RVE as high-fidelity. The investigation started with a version of the problem where all the feature space was under control and complex loading scenario predictions were investigated. Then, the investigation moved on to a more realistic case where the feature space was free to follow paths determined by the global Newton-Raphson solution of a 1-element FE model. Finally, the time gain in a more computationally demanding fidelity structure has been investigated. The findings from this research endeavor are summarized below.

Controlled Strain Paths

- The multiGPR model is able to predict unseen loading scenarios at the high-fidelity fairly well compared to GPR, which is unable to do so. This is due to increased extrapolation capabilities of the multiGPR with the information coming from the low-fidelity.
- The assumption constant ρ has a notable effect on the results as the correlation between the load cases change with respect to feature space, the found overall correlation coefficient ρ loses its effectiveness resulting in the predictions to lose representativeness for some loading scenarios where the correlation is not representative. It is observed that the local utilization of the multiGPR is significantly more effective with the given assumption since the correlation between successive fidelities is achieved by just one loading scenario and due to its Markov like property the accuracy near the vicinity of the trained loading case predictions are quite similar as a result of similar correlations being present between low and high-fidelity.
- The effect of a thorough collection of low-fidelity data is found out to be insignificant, and combined with the local high-fidelity collection resulting in good predictions which makes multiGPR an efficient method in an online framework, where all the data is collected during simulation.
- Even an unrepresentative low-fidelity created by neural networks is found to be usable if the investigation is constrained to be in a local region, where high-fidelity information is present.

Uncontrolled Strain Paths

- Findings from controlled strain path simulations still hold their validity (*e.g.* consequences of constant ρ assumption, thorough collection of low-fidelity not affecting the predictions in terms of loading scenarios too much.). Moreover, some of them have more significant consequences due to the correlations changing for most of the load cases.

- Divergence and inability to follow the strain path are observed. Although one might assume that the inclusion of more data points might improve the performance, only providing information on the strain path is as effective as providing information in every direction. Thus, the divergence problems are solvable by means of co-kriging with derivative information at every point. However, the increased computational complexity should be noted due to the increased size of the covariance matrix.
- Observed increasing variance bands are caused by the difference between low and high-fidelity snapshots being collected from different strain paths due to the possibility of different fidelities having different Poisson's ratios. The cause of this problem can be seen clearly in the posterior of the recursive model, where the low-fidelity predictions are utilized at the training points observed in on the high-fidelity. Although not realistic, collecting the low-fidelity snapshots using strains coming directly from the high fidelity solutions solves this problem.

Performance

- It is found from the performance review that for the introduced fidelity structures, obtaining and training the multiGPR model takes longer compared to running a 1-fiber RVE model.
- By increasing the computational complexity by means of more fiber inclusions, it is found that in the case of an increasing number of computations even for a single RVE multiGPR can outperform the micromodel. However, there is a price to pay in terms of accuracy.
- It is found that employing the multiGPR model locally decreases the time consumed for both obtaining the snapshots and training, making it a feasible tool to utilize online.

Overall, it is found that multiGPR is a promising tool that can be employed as a surrogate model for composite models. However, due to strong underlying assumptions, there are some inherent problems that hinder its general use. This work can be used as a stepping stone towards multi-fidelity information inclusion while constructing surrogate constitutive models for composite micromodels.

6.2. Recommendations for Future Work

This section is dedicated to the recommendations for future work that might improve the given scheme as a surrogate constitutive model.

- In order to be utilized as an offline-trained surrogate model that can be used without loss of generality, the scalability problem inherent to GP models, in general, has to be addressed. Examples of this can be found in [6], by means of employing Sparse Gaussian Processes and a futuristic version of this is employed in [25] where the scalability issue of GPs is tackled by harnessing the power of quantum computing. Then more interesting cases can be investigated with offline training.

- The assumed constant and scalar correlation parameter ρ introduces problems regarding the modeling of correlations that are not linear. In order to get better interpolation capabilities employing methods to overcome this known issue is highly suggested. This problem is addressed in [29, 32, 36, 54] with mathematical improvements to the existing framework for multi-fidelity information utilization.

- This work has focused on a very simplistic scenario, in other words, a simple 1-fiber RVE in a two-dimensional setting where there are three stress-strain pairs. More representative cases should be investigated as in the case of [43, 52] where more problematic load cases are investigated in a three-dimensional setting.

- Due to the single output nature of the GP models, three independent GP's are used for creating a surrogate model. This case can be further improved by means of including the cross-correlations between the stress components as deemed possible in [41].

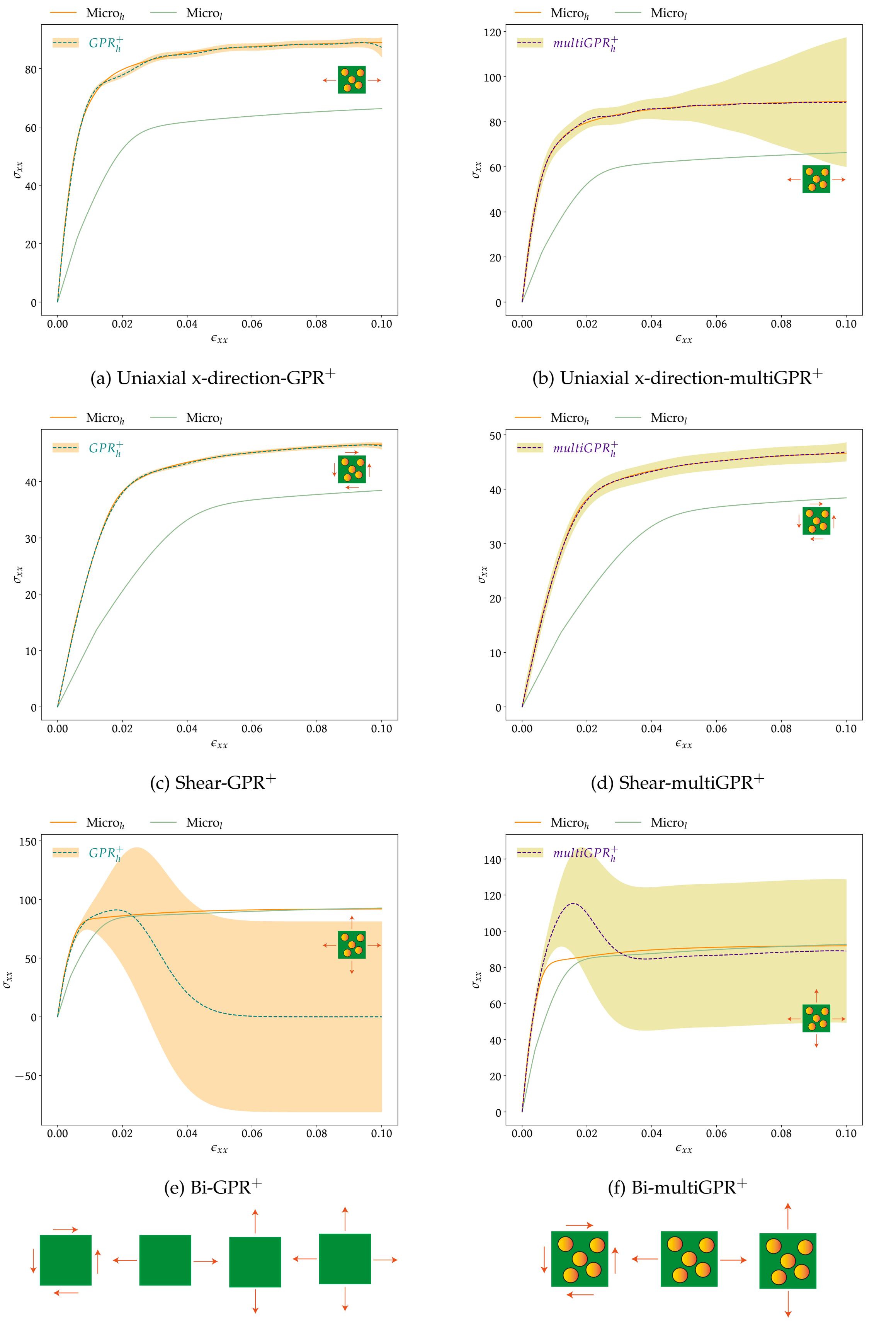
- This work focused on only bi-fidelity investigation for the sake of simplicity. This fidelity structure can be increased to s levels is straightforward thanks to the recursive formulation presented in [27].

- Online applications similar to the ones that are presented in [43] can be interesting to look at. This might be a better use case this method by collecting local information from multiple fidelities benefit the overall solution or the computational performance of the multiGPR method.

- The performance comparison in this thesis was meant to show the potential of the method and is not comprehensive. Thus, more detailed benchmarking should be done in order to fully deem this method as a viable option for creating surrogate models.

A | GPR$^+$ and multiGPR$^+$

This section includes extra results to tackle with overprediction problem seen with the GPR$^+$ and multiGPR$^+$ seen in Figure 5.16. In order to circumvent this problem, the training domain can be separated into two regions. The first region extends until a point beyond the start of the plasticity and other region covers the rest. By collecting the first region denser than the second region. As it can be seen from Figure A.1. It should be noted that only positive cases are utilized to train the surrogate models and only the positive loading cases are investigated. As can be seen from the given figure, increasing the density of training points through the sudden changing nature of the given problem helps to overcome the issue of overprediction near the start of plasticity. Moreover, again the divergence problem is avoided as in the case presented in Figure 5.16. Thus, it can be concluded that the density of the training points has an influence on the given solution as well, especially when the feature space is roamed freely.



(a) Uniaxial x-direction-GPR$^+$

(b) Uniaxial x-direction-multiGPR$^+$

(c) Shear-GPR$^+$

(d) Shear-multiGPR$^+$

(e) Bi-GPR$^+$

(f) Bi-multiGPR$^+$

(g) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(h) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

Figure A.1: Predictions for varying training point densities to solve overprediction near the start of plasticity for GPR$^+$ and multiGPR$^+$ models with a different sampling scheme.

B | Neural Networks

This appendix chapter is dedicated to the brief introduction to the concepts of neural networks utilized in this thesis. The interested reader can consult to [1, 7, 43] for more detailed information. Moreover, it should be noted that this thesis utilizes a multilayer perceptron, which also known as feed-forward Neural Network (NN) and the focus will be solely on this method. As shown in Figure 2.3 the NN architecture consists of input layer ($i = 0$), multiple fully connected multiple hidden layers ($l, ..., l-2$), dropout layer ($l-1$) and the output layer (l). Considering the parametric regression model:

$$y(\boldsymbol{x}, \mathbf{w}) = \sum_{j}^{M} w_j \phi_j(\boldsymbol{x}). \tag{B.1}$$

Neural networks use the same form, where every basis function is a nonlinear function of linear combinations of the inputs [7]. Moreover, the coefficients w_i are the adaptive parameters. This allows the neural network model to be represented as a series of functional transformations.

Hidden Layers

Every hidden layers feeds another layer with the neuron states (**a**) from the previous layer:

$$\mathbf{v}_i = \mathbf{W}_i \mathbf{a}_{i-1} + \mathbf{b}_i, \tag{B.2}$$

where $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_{i-1}}$ is the weight matrix, $\mathbf{b}_i \in \mathbb{R}^{n_i}$ is the bias vector n_i is the neuron number of the i^{th} layer.During the propagation $\mathbf{v}_i$ is passed through a nonlinear activation function $\phi(\cdot)$ to introduce nonlinearity to the output.

$$\mathbf{a}_i = \Phi(\mathbf{v}_i) \tag{B.3}$$

In this works sigmoid function is chosen to be the activation function. It is widely utilized for regression problems and expressed as:

$$\Phi(v) = \frac{e^v}{e^v + 1}. \tag{B.4}$$

Although increasing n_i leads to a better representation as it increases the number of tunable parameters it can cause overfitting [43]. This means the increased ability to represent the training data set perfectly, but fail to predict unseen data accurately. Multiple regularization methods can be utilized to circumvent this problem [1]. In this thesis, a dropout layer after the last hidden layer will be utilized to prevent overfitting.

Dropout Layer

The dropout layer stochastically deactivates some of the nodes coming from the previous layer in order to prevent overfitting.

$$\mathbf{a}_{l-1} = \frac{1}{1-r_d}(\mathbf{r} \odot \mathbf{a}_{l-2}) \tag{B.5}$$

where $\odot$ is the Hadamard product and $\mathbf{r}$ is a vector consisting of independent Bernoulli random variables [49]. Scaling factor $1 - r_d$ ensures the average of neuron values are unchanged. With r_d being the dropout rate between 0 and 1. This method prevents the NN to depend on a single neuron by eliminating some of the nodes at each time there is output resulting from the feed-forward process [43].

Training Process

Main purpose of the training is to minimize the error (loss) function given by:

$$L = \frac{1}{p}\sum_{j=1}^{p}\frac{1}{2}||y(\boldsymbol{x},\mathbf{w}) - \hat{y}||^2, \tag{B.6}$$

where the $\hat{y}$ is the predicted value p is the number of points utilized in training set. It should be noted that the loss function selected be mean squared error multiplied with $1/2$ in order to keep the derivatives simpler. [43]. During the training process 20% of the randomly selected inputs will be preserved, to be later on utilized as a validation set.

Loss function is minimized using the Stochastic Gradient Descent (SGD) to update the tunable prametres $\mathbf{W}$ and $\mathbf{b}$ given in Equation B.2. Noting that 0 indicates the current values and n indicates the updated values updated values are expressed as:

$$\mathbf{W}^n = \mathbf{W}^0 - \mathcal{A}\left(\frac{1}{B}\sum_{j}^{B}\frac{\partial L_j}{\partial \mathbf{W}}\right) \tag{B.7}$$

$$\mathbf{b}^n = \mathbf{b}^0 - \mathcal{A}\left(\frac{1}{B}\sum_{j}^{B}\frac{\partial L_j}{\partial \mathbf{b}}\right) \tag{B.8}$$

where B is the size of the mini-batch. Mini-batches are utilized in order to have a balance relationship between the speed of convergence and the gradient variance. Moreover, operator $\mathcal{A}$ is solver dependent and in this work *Adam* solver will be utilized [22].

As it can be seen in Equations B.7 and B.8, for the updated values of weights and biases, the derivative of the loss function with respect to both of them is needed. In a feed-forward network, this can be achieved by a local message-passing scheme, also called *error backpropagation* [7]. The chain rule is utilized to pass on the derivative of the loss function from the output layer to the first hidden layer. Noting that the error terms for each layer are defined by $\mathbf{d}_i \in \mathbb{R}^{n_i}$ at the output layer (l) it can be found to be:

$$\mathbf{d}_l = \frac{\partial L}{\partial \mathbf{a}_l} = \hat{y} - y. \tag{B.9}$$

Then, the effect of activation function is imposed on the errors:

$$\bar{\mathbf{d}}_i = \mathbf{d} \odot \frac{\partial \Phi}{\partial v}(\mathbf{v}_i). \tag{B.10}$$

Now, the gradients of the loss function with respect to parameters $\mathbf{W}$ and $\mathbf{b}$ can be calculated as:

$$\frac{\partial L}{\partial \mathbf{W}_i} = \bar{\mathbf{d}}_i \mathbf{a}_i^T \tag{B.11}$$

$$\frac{\partial L}{\partial \mathbf{b}_i} = \bar{\mathbf{d}}_i \tag{B.12}$$

Thus, the errors in the previous layer can be obtained by:

$$\mathbf{d}_{i-1} = \mathbf{W}_i^T \bar{\mathbf{d}}_i. \tag{B.13}$$

It should be noted that since dropout layers has no trainable parameters, only the effect of the dropout layer is backpropagated as:

$$\mathbf{d}_{i-1} = \bar{\mathbf{d}}_i = \frac{1}{1-r_d}(\mathbf{r} \odot \mathbf{d}_i). \tag{B.14}$$

Derivative Predictions

In some cases the prediction of the derivative is as important as the predicted value. In this case the Jacobian of the can be calculated as:

$$\mathbf{J} = \frac{\partial \mathbf{a_l}}{\partial \mathbf{v}_0}. \tag{B.15}$$

which can be obtained by backward pass as:

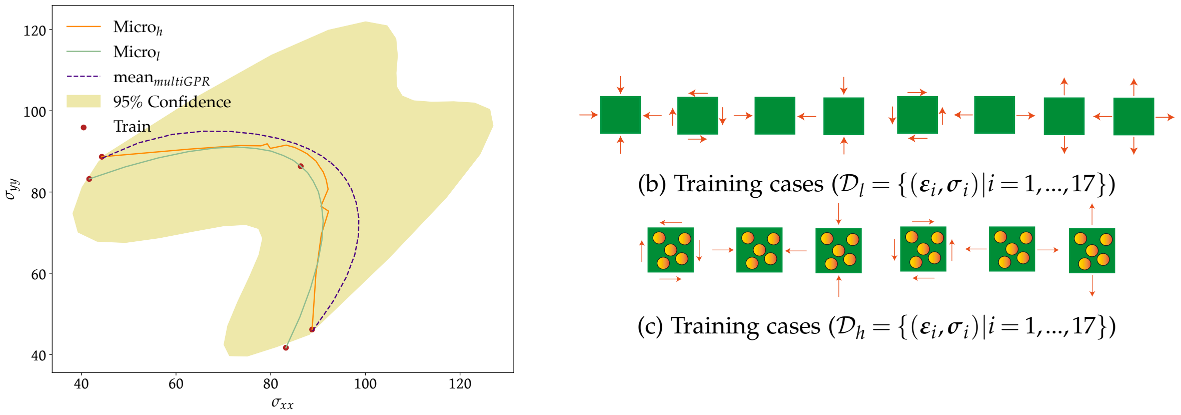
$$\mathbf{J}_i = \mathbf{J}_{i+1} \mathbf{I}_i^{\Phi'} \mathbf{W}_i. \tag{B.16}$$

It should be noted that $\mathbf{J}_{l+1} = \mathbf{I}$ and $\mathbf{I}_i^{\Phi'}$ is expressed as:

$$\mathbf{I}_i^{\Phi'} = \operatorname{diag}\left(\frac{\partial \Phi}{\partial v}(\mathbf{v}_i)\right) \tag{B.17}$$

C | 25-fiber RVE Envelopes

This chapter is dedicated to showing the 25-fiber general and local envelopes created with the multi-GPR model. It should be noted that the kinks in the envelopes are due to the diverging solutions obtained. Moreover, the low-fidelity utilized is still a homogenous material.

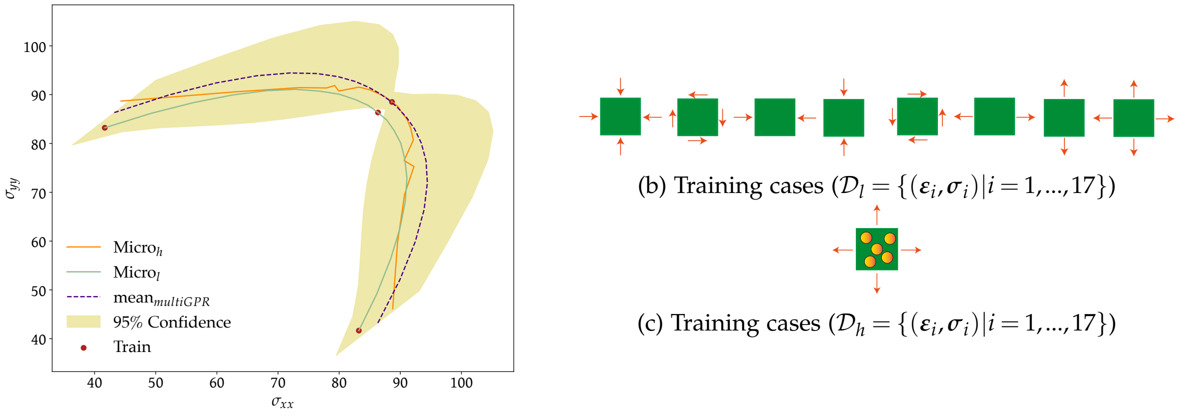


(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(a) Envelope created with multiGPR

Figure C.1: Yield stress envelope for multiGPR model for a training scenario involving both positive and negative load cases.



(b) Training cases ($\mathcal{D}_l = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(c) Training cases ($\mathcal{D}_h = \{(\varepsilon_i, \sigma_i)|i = 1,...,17\}$)

(a) Envelope created with multiGPR

Figure C.2: Yield stress envelope for multiGPR model for a training scenario involving both positive and negative load cases.

Bibliography

[1] Charu C. Aggarwal. *Neural Networks and Deep Learning*. 2018. ISBN 9783319944623. doi: 10.1201/b22400-15.

[2] D. Amsallem and A. Paul-Dubois-Taine. An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models. *International*, 102(September):1262–1292, 2015. doi: 10.1002/nme.4759. URL `http://onlinelibrary.wiley.com/doi/10.1002/nme.3279/full`.

[3] Francesco Archetti and Antonio Candelieri. *Bayesian Optimization and Data Science*. 2019. ISBN 9783030244934. doi: 10.1007/978-3-030-24494-1.

[4] Andrew C. Bergan, Miguel Herráez, Carlos González, and Claudio S. Lopes. Development of a mesoscale finite element constitutive model for fiber kinking. *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, (210049):1–23, 2018. doi: 10.2514/6.2018-1221.

[5] M. A. Bessa, R. Bostanabad, Z. Liu, A. Hu, Daniel W. Apley, C. Brinson, W. Chen, and Wing Kam Liu. A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. *Computer Methods in Applied Mechanics and Engineering*, 320:633–667, 2017. ISSN 00457825. doi: 10.1016/j.cma.2017.03.037.

[6] Miguel A. Bessa, Piotr Glowacki, and Michael Houlder. Bayesian Machine Learning in Metamaterial Design: Fragile Becomes Supercompressible. *Advanced Materials*, 31(48), 2019. ISSN 15214095. doi: 10.1002/adma.201904845.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning*, volume 27. 2006. ISBN 9780387310732.

[8] Luca Bonfiglio, Paris Perdikaris, Stefano Brizzolara, and George Em Karniadaki. A multi-fidelity framework for investigating the performance of super-cavitating hydrofoils under uncertain flow conditions. *19th AIAA Non-Deterministic Approaches Conference, 2017*, (January):1–13, 2017. doi: 10.2514/6.2017-1328.

[9] Kurt Cutajar, Mark Pullin, Andreas Damianou, Neil Lawrence, and Javier González. Deep Gaussian Processes for Multi-fidelity Modeling. 1(NeurIPS):1–6, 2019. URL `http://arxiv.org/abs/1903.07320`.

[10] David Kristjanson Duvenaud. Automatic Model Construction with Gaussian Processes. 2014. URL `https://www.cs.toronto.edu/{~}duvenaud/thesis.pdf`.

[11] R. Fletcher. Practical Methods of Optimization, 2000.

[12] Alexander I.J. Forrester, András Sóbester, and Andy J. Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269, 2007. ISSN 14712946. doi: 10.1098/rspa.2007.1900.

[13] M. G.D. Geers, V. G. Kouznetsova, and W. A.M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics*, 234(7): 2175–2182, 2010. ISSN 03770427. doi: 10.1016/j.cam.2009.08.077. URL `http://dx.doi.org/10.1016/j.cam.2009.08.077`.

[14] Marc G. D. Geers, Varvara G. Kouznetsova, Karel Matouš, and Julien Yvonnet. *Homogenization Methods and Multiscale Modeling: Nonlinear Problems*. 2017. ISBN 9781119176817. doi: 10.1002/9781119176817.ecm2107.

[15] Olivier Goury, David Amsallem, Stéphane Pierre Alain Bordas, Wing Kam Liu, and Pierre Kerfriden. Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to Bayesian optimization. *Computational Mechanics*, 58(2):213–234, 2016. ISSN 01787675. doi: 10.1007/s00466-016-1290-2.

[16] J. A. Hernández, J. Oliver, A. E. Huespe, M. A. Caicedo, and J. C. Cante. High-performance model reduction techniques in computational multiscale homogenization. *Computer Methods in Applied Mechanics and Engineering*, 276:149–189, 2014. ISSN 00457825. doi: 10.1016/j.cma.2014.03.011.

[17] J. A. Hernández, M. A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer Methods in Applied Mechanics and Engineering*, 313:687–722, 2017. ISSN 00457825. doi: 10.1016/j.cma.2016.10.022.

[18] R. Hill. A self-consistent mechanics of composite materials. *Journal of the Mechanics and Physics of Solids*, 13(4):213–222, 1965. ISSN 00225096. doi: 10.1016/0022-5096(65)90010-4.

[19] Oliver C. Ibe. Introduction to Markov Processes. *Markov Processes for Stochastic Modeling*, pages 49–57, 2013. doi: 10.1016/b978-0-12-407795-9.00003-7.

[20] T. Kanit, S. Forest, I. Galliet, V. Mounoury, and D. Jeulin. Determination of the size of the representative volume element for random composites: Statistical and numerical approach. *International Journal of Solids and Structures*, 40(13-14):3647–3679, 2003. ISSN 00207683. doi: 10.1016/S0020-7683(03)00143-4.

[21] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000. ISSN 00063444. doi: 10.1093/biomet/87.1.1.

[22] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[23] V.G. Kouznetsova. *Computational homogenization for the multi-scale analysis of multi-phase materials*. Number 2002. 2002. ISBN 9038627343. doi: 10.6100/IR560009. URL `http://www.narcis.nl/publication/RecordID/oai:library.tue.nl:560009`.

[24] Anouar Krairi and Issam Doghri. A thermodynamically-based constitutive model for thermoplastic polymers coupling viscoelasticity, viscoplasticity and ductile damage. *International Journal of Plasticity*, 60:163–181, 2014. ISSN 07496419. doi: 10.1016/j.ijplas.2014.04.010. URL `http://dx.doi.org/10.1016/j.ijplas.2014.04.010`.

[25] Gawel Kus. *Quantum gaussian processes for data-driven design of metamaterials*. 2019. URL https://repository.tudelft.nl/islandora/object/uuid{%}3Aa94b7431-22d2-4d95-bfd1-986519857ba1?collection=education.

[26] Neil Lawrence. Design of a Novel Piezoelectric Stick-Slip Driving Nanopositioning Stage and Power Supply Circuit. *MARSS 2018 - International Conference on Manipulation, Automation and Robotics at Small Scales*, 6:1783–1816, 5. doi: 10.1109/MARSS.2018.8481225.

[27] Loic Le Gratiet. Multi-fidelity Gaussian process regression for computer experiments Thèse dirigée par Josselin Garnier Thèse rapportée par. 2013.

[28] Loic Le Gratiet. Bayesian Analysis of Hierarchical Multifidelity Codes. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):244–269, 2013. ISSN 2166-2525. doi: 10.1137/120884122.

[29] Haitao Liu, Yew Soon Ong, Jianfei Cai, and Yi Wang. Cope with diverse data structures in multi-fidelity modeling: A Gaussian process method. *Engineering Applications of Artificial Intelligence*, 67(November 2017):211–225, 2018. ISSN 09521976. doi: 10.1016/j.engappai.2017.10.008. URL https://doi.org/10.1016/j.engappai.2017.10.008.

[30] Zeliang Liu, M. A. Bessa, and Wing Kam Liu. Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 306:319–341, 2016. ISSN 00457825. doi: 10.1016/j.cma.2016.04.004. URL http://dx.doi.org/10.1016/j.cma.2016.04.004.

[31] A. R. Melro, P. P. Camanho, F. M. Andrade Pires, and S. T. Pinho. Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part I-Constitutive modelling. *International Journal of Solids and Structures*, 50(11-12):1897–1905, 2013. ISSN 00207683. doi: 10.1016/j.ijsolstr.2013.02.009. URL http://dx.doi.org/10.1016/j.ijsolstr.2013.02.009.

[32] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *Journal of Computational Physics*, 401:109020, 2020. ISSN 10902716. doi: 10.1016/j.jcp.2019.109020. URL https://doi.org/10.1016/j.jcp.2019.109020.

[33] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, Jian Cao, and M. A. Bessa. Deep learning predicts path-dependent plasticity. *Proceedings of the National Academy of Sciences of the United States of America*, 116(52):26414–26420, 2019. ISSN 10916490. doi: 10.1073/pnas.1911815116.

[34] Kevin P. Murphy. *Machine Learning A Probabilistic Perspective*. 2012. ISBN 9780262018029. doi: 10.1007/978-94-011-3532-0_2.

[35] Vinh Phu Nguyen, Oriol Lloberas-Valls, Martijn Stroeven, and L.J. Sluys. Computational homogenization for multiscale crack modeling. Implementational. *International*, (February):1102–1119, 2012. doi: 10.1002/nme.3237. URL http://onlinelibrary.wiley.com/doi/10.1002/nme.3279/full.

[36] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198), 2017. ISSN 14712946. doi: 10.1098/rspa.2016.0751.

[37] Paris Perdikaris and George Em Karniadakis. Model inversion via multi-fidelity Bayesian optimization: A new paradigm for parameter estimation in haemodynamics, and beyond. *Journal of the Royal Society Interface*, 13(118), 2016. ISSN 17425662. doi: 10.1098/rsif.2015.1107.

[38] S. Pimenta, R. Gutkin, S. T. Pinho, and P. Robinson. A micromechanical model for kink-band formation: Part I - Experimental study and numerical modelling. *Composites Science and Technology*, 69(7-8):948–955, 2009. ISSN 02663538. doi: 10.1016/j.compscitech.2009.02.010. URL `http://dx.doi.org/10.1016/j.compscitech.2009.02.010`.

[39] Jakub Pruher and Simo Sarkka. On the use of gradient information in Gaussian process quadratures. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2016-Novem, 2016. ISSN 21610371. doi: 10.1109/MLSP.2016.7738903.

[40] C. Qian, T. Westphal, and R. P.L. Nijssen. Micro-mechanical fatigue modelling of unidirectional glass fibre reinforced polymer composites. *Computational Materials Science*, 69:62–72, 2013. ISSN 09270256. doi: 10.1016/j.commatsci.2012.10.015. URL `http://dx.doi.org/10.1016/j.commatsci.2012.10.015`.

[41] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*. 2006. ISBN 026218253X. doi: 10.1142/S0129065704001899.

[42] I B C M Rocha, P Kerfriden, and F P Van Der Meer. On-the-fly construction of surrogate constitutive models for concurrent multiscale analysis through probabilistic learning. *In the process of Review*, 2020.

[43] I. B.C.M. Rocha, P. Kerfriden, and F. P. van der Meer. Micromechanics-based surrogate models for the response of composites: A critical comparison between a classical mesoscale constitutive model, hyper-reduction and neural networks. *European Journal of Mechanics, A/Solids*, 82:103995, 2020. ISSN 09977538. doi: 10.1016/j.euromechsol.2020.103995. URL `https://doi.org/10.1016/j.euromechsol.2020.103995`.

[44] I.B.C.M. Rocha. *Numerical and Experimental Investigation of Hygothermal Aging in Laminated Composites*. 2018. ISBN 9789462339583. doi: 10.4233/uuid.

[45] I.B.C.M. Rocha, F.P. van der Meer, and L.J. Sluys. An adaptive domain-based POD/ECM hyper-reduced modeling framework without offline training. *Computer Methods in Applied Mechanics and Engineering*, 358:112650, 2020. ISSN 00457825. doi: 10.1016/j.cma.2019.112650. URL `https://doi.org/10.1016/j.cma.2019.112650`.

[46] Stuart Russell and Peter Norvig. *Artifical Intelegence A Modern Approach*. 2016. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004.

[47] Jörg Schröder and Klaus Hackl. *Plasticity and Beyond*. Springer, 2014. ISBN 9783709116241. URL `www.springer.com/series/76`.

[48] E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen. Derivative observations in Gaussian Process Models of Dynamic Systems. *Advances in Neural Information Processing Systems*, 2003. ISSN 10495258.

[49] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting N. *Journal of Machine Learning Research*, 20143. ISSN 03702693. doi: 10.1016/0370-2693(93)90272-J.

[50] F. P. Van Der Meer and L. J. Sluys. Continuum models for the analysis of progressive failure in composite laminates. *Journal of Composite Materials*, 43(20):2131–2156, 2009. ISSN 00219983. doi: 10.1177/0021998309343054.

[51] F.P. van der Meer. *Computational Modeling of Failure in Composite Laminates*, volume PhD. 2010. ISBN 9789090256979.

[52] Frans P. Van Der Meer. Micromechanical validation of a mesomodel for plasticity in composites. *European Journal of Mechanics, A/Solids*, 60:58–69, 2016. ISSN 09977538. doi: 10.1016/j.euromechsol.2016.06.008. URL `http://dx.doi.org/10.1016/j.euromechsol.2016.06.008`.

[53] M. Vogler, R. Rolfes, and P. P. Camanho. Modeling the inelastic deformation and fracture of polymer composites-Part I: Plasticity model. *Mechanics of Materials*, 59:50–64, 2013. ISSN 01676636. doi: 10.1016/j.mechmat.2012.12.002. URL `http://dx.doi.org/10.1016/j.mechmat.2012.12.002`.

[54] Shuhan Zhang, Wenlong Lyu, Fan Yang, Changhao Yan, Dian Zhou, Xuan Zeng, and Xiangdong Hu. An efficient multi-fidelity Bayesian optimization approach for analog circuit synthesis. *Proceedings - Design Automation Conference*, pages 0–5, 2019. ISSN 0738100X. doi: 10.1145/3316781.3317765.