



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Empa

Materials Science and Technology

Surrogate modelling for multiscale thermal simulation of powder-bed additive manufacturing

Master Thesis

Fabian Keller

April 14, 2020

Advisors: Prof. Dr. E. Mazza, Dr. E. Hosseini, Dr. S. Marelli

Experimental Continuum Mechanics Lab, Empa

Department of Mechanical Engineering, ETH Zürich

Abstract

In this master thesis, a surrogate modelling strategy for temperature prediction in powder-bed additive manufacturing is presented. The goal is to reduce the computational effort needed to run a multi-scale finite element simulation for selective laser melting developed at *Empa*. The reduction is achieved by substituting part of the finite element simulations by a surrogate model that is cheap to evaluate. The proposed surrogate model uses principal component analysis to reduce the dimensions of input and output space of temperature fields and polynomial chaos expansion to model the dependence between reduced input and output data. For various different geometries, evolution of the temperatures during the printing process and peak temperatures in the build part are modeled and compared to corresponding finite element simulations for surrogate model validation.

Acknowledgements

I am very grateful that I had the opportunity to write my master thesis at the *Experimental Continuum Mechanics Lab* under supervision of Prof. Dr. Edoardo Mazza.

Dr. Ehsan Hosseini, my first supervisor, and Pooriya Ghanbari helped me a lot with everything regarding the finite element simulations and gave me plenty of good ideas on how to improve the surrogate models. I would also like to thank my second supervisor, Dr. Stefano Marelli from ETH Zurich for introducing me into the world of surrogate modelling and for sharing his incredible knowledge on dimensional reduction and surrogate modelling with me.

Finally, I would like to thank my family and Laura for all their support during these six months.

Contents

Acknowledgements	iii
Contents	v
List of Figures	vii
List of Tables	xi
1 Introduction	1
2 Theory	5
2.1 Additive manufacturing	5
2.1.1 Selective laser melting	6
2.2 Polynomial chaos expansion	7
2.2.1 Orthonormal polynomials	7
2.2.2 Truncation of infinite-dimensional basis	9
2.2.3 Computation of the coefficients	10
2.2.4 Projection method	10
2.2.5 Least squares regression method	11
2.2.6 Adaptive polynomial chaos expansion	12
2.3 Principal component analysis	13
3 Model Development	15
3.1 Multi-scale model	16
3.1.1 Global simulation	16
3.1.2 Local simulation	16
3.2 Surrogate model	17
3.2.1 Cross validation	19
3.2.2 Error measures	20
3.3 Theoretical error estimates	20
3.4 Building the principal component space	21

CONTENTS

3.5	Single model prediction	26
3.5.1	Input dimension convergence test	26
3.5.2	Output dimension convergence test	30
3.5.3	Influence of experimental design size	31
3.6	Full layer prediction	34
3.6.1	Accumulation of error within a layer	35
3.6.2	Convergence for number of training layers	36
3.6.3	Selection of training layers	42
3.7	Marginal distributions	45
3.8	Prediction of peak temperatures	47
4	Results	49
4.1	Printing a solid block	49
4.2	Printing a two-column-structure	54
4.3	Printing a T-structure	57
5	Conclusions and Outlook	61
A	Guideline for surrogate model usage	65
B	Boundary errors	67
	Bibliography	71

List of Figures

3.1	Proposed workflow of the surrogate model with dimensional reduction and reconstruction of the features in the original space.	17
3.2	Workflow for the prediction of a layer. The prediction results of a local model are interpolated to the coordinates of the next local model and used as input values to the next surrogate model. . .	18
3.3	First four elements of the principal component basis computed from normalized initial temperature data.	22
3.4	Percentage of the total variance in the initial temperature data that is explained by each principal component.	23
3.5	Normalized reconstruction error of the principal component dimension reduction for the result temperature data set of the geometry 3.7. The principal component basis is computed with 50, 100 and 150 random data points of 165 in total.	24
3.6	Normalized reconstruction error of the principal component dimension reduction for the result temperature data set of the geometry 3.7. The principal component basis is computed once with only local data and once with additional global data. . . .	25
3.7	Sketch of the square block printing geometry.	26
3.8	Normalized root mean square error plotted against the number of principal components used as input parameters for printing a solid block.	27
3.9	Sketch of the geometry for printing two equally sized blocks with a void area in between.	28
3.10	Normalized root mean square error plotted against the number of principal components used as input parameters in the setting of printing two columns (Figure 3.9).	28

3.11	Normalized root mean square error plotted against the number of principal component input parameters in the setting of printing two blocks (Figure 3.9) with two different surrogate models depending on the laser activation.	29
3.12	Normalized root mean square error plotted against the number of principal components used as output parameters in the setting of printing a solid block.	31
3.13	Normalized root mean square error plotted against the number of principal components used as output parameters in the setting of printing two columns (Figure 3.9) with two different models depending on the laser activation.	32
3.14	Normalized root mean square error plotted against the number of principal components used as input parameters for printing a solid block. For the different curves, the principal components were computed using training data sets of different sizes. In the blue curve the full, in the red curve half and in the yellow curve a quarter of the experimental design (ED) was used for building the principal component space.	33
3.15	Normalized root mean square error plotted against the size of the experimental design for training the polynomial chaos model. The solid line shows the average test error over the different random splits whereas the shaded region depicts the minimum and maximum values of the error.	34
3.16	Evolution of the normalized prediction error over one layer when using the prediction scheme for a complete layer (blue curve) and when always using the true initial temperatures from the finite element model (red curve).	35
3.17	Normalized root mean square error at the test layers plotted against the number of training layers for the polynomial chaos expansion model in the setting of a filled block (Figure 3.7). The solid line shows the average test error over the different random splits in training and test data. The shaded region depicts the minimum and maximum values of the error.	37
3.18	Normalized root mean square error at the test layers plotted against the number of training layers for the polynomial chaos expansion. For this test the geometrical setup with two separated blocks (Figure 3.9) is used and two different models are trained, for the region with activated and deactivated laser, respectively. The solid lines show the average test error over the different random splits in training and test data. The shaded regions depict the minimum and maximum values of the error.	38
3.19	Comparison of the temperature profile of a local and a surrogate model two layers above an overhang structure.	39

3.20	Normalized root mean square error at the test layers plotted against the number of training layers for the polynomial chaos expansion. For this test we take a geometrical setup with a horizontal bar. The solid lines show the average test error over the different random splits in training and test data. The shaded regions depict the minimum and maximum values of the error.	41
3.21	Normalized root mean square error of the surrogate model plotted against the number of training layers. The training layers are selected with different (non-random) strategies. For this experiment, data of the square geometry in Figure 3.7 is used.	43
3.22	Histogram of the normalized prediction error of surrogate models trained on different random selections of two training layers. The vertical red line indicates the prediction error of a model trained on two layers with equal distance to each other and to the bottom and the top of the geometry.	44
3.23	First principal component values for all the observations of the initial temperature data in the square geometry.	46
3.24	Comparison of the prediction accuracy between predicting only peak temperatures and computing the peak temperatures from the predictions of the complete temperature fields.	48
4.1	Comparison of the temperature evolution provided by the surrogate model (solid blue line) and the (exact) local model (dashed red line) at the point $(x, y) = (0.5\text{mm}, 0.5\text{mm})$.	50
4.2	Comparison of the temperature evolution provided by the surrogate model (solid blue line) and the (exact) local model (dashed red line) at the point $(x, y) = (0.5\text{mm}, 0.5\text{mm})$ for the first pass of the laser over this point.	51
4.3	Comparison of the temperature profile of a local and a surrogate model in the uppermost printing layer.	52
4.4	Comparison of the peak temperatures in the whole printing part between surrogate and finite element model for the solid block geometry.	53
4.5	Sketch of the two-column structure.	54
4.6	Comparison of the peak temperatures in the whole printing part between surrogate and finite element model for the geometry in Figure 4.5.	55
4.7	Sketch of the two-column structure with the separate overhang model indicated in red.	56
4.8	Absolute difference in the peak temperatures between the surrogate model with an additional model for the overhang region and the true values from the finite element model printing the same structure as in Figure 4.6.	57
4.9	Sketch of the T-structure.	58

LIST OF FIGURES

4.10 Comparison of the peak temperatures in the whole printing part between surrogate and finite element model for the geometry in Figure 4.9.	59
B.1 Mean square error of all the local surrogate models on the geometry 4.5.	68

List of Tables

3.1	Test errors for different combinations of marginal distribution and copula type for the square geometry (Figure 3.7) and the geometry with two columns (Figure 4.5). Marginal types are uniform, inferred from the data by statistical methods and kernel smoothing (KS). The dash indicates that it is not possible to fit a model.	47
4.1	Time needed for running the multiscale simulation for the solid block geometry.	49
4.2	Time needed for training and running the surrogate model for the solid block geometry.	49

Chapter 1

Introduction

Additive manufacturing has been introduced as a manufacturing technique about 30 years ago. In its early days it has mostly been used for (rapid) prototyping. More advanced techniques and processes allow its usage in productive environments as well, especially in areas where conventional manufacturing techniques are faced with challenges, such as freedom of design [5].

Selective laser melting (SLM) is a metal additive manufacturing technique with wide-spread use. It builds up parts by melting and fusing layers of metal powder together. SLM provides highly accurate building components and few restrictions on the part design. However, the mechanical properties of the printed parts show deficiencies in terms of high porosity, distortions or high residual stresses. Thus, for industrial applications, it is very important to find good process parameters in order to improve the build quality. Optimization of process parameters can be achieved with physical simulations of the build process.

Many different simulations for selective laser melting focusing on different characteristics and following various approaches have been developed. Due to the high power of the laser, large temperature gradients appear around the melt pool. In simulations, these gradients need to be captured accurately. To keep the computational costs small, different simplifications are applied. Often, the metal powder is modeled as a continuum and the dynamics regarding the granular form of the powder and the fluid inside the melt pool are neglected.

To reduce the computational costs even more, a multi-scale model is being developed at *Empa* by Pooriya Ghanbari. This model uses a very coarse mesh to simulate the whole printing part and many fine mesh models that successively cover the area around the melt pool where high temperature gradients occur. Finally, the results of both simulation types are combined.

This approach is computationally efficient because the simulations with the fine mesh only cover a small fraction of the complete component. Yet, for large printing parts, many thousand fine mesh models must be solved. This results in a large computational burden. The goal of this project is to substitute these fine models in the multi-scale approach with a data-driven polynomial chaos expansion (PCE) surrogate model.

Meta modelling techniques like surrogate models are used to replace expensive simulations or experiments with cheap to evaluate computational models. Typically, surrogate models treat the computational algorithms they aim to replace as black boxes and do not require knowledge about their internals. Instead, they work in a data-driven way, i.e. they only use pairs of input and output data points of the simulation. The goal of surrogate modelling is to use very few runs of the expensive simulation (or experiment) at wisely chosen sample points to construct a model that emulates the behavior of the original simulation as good as possible.

Examples of surrogate models include Kriging [15] or polynomial chaos expansion, which is used in this project. Polynomial chaos expansion (PCE) [22] was developed to determine the evolution of uncertainties in systems with probabilistic input. It tries to emulate the behavior of the original simulation with a set of orthogonal polynomial functions. Moreover, it is specially suited for situations where only few data is available. In these situations other methods like artificial neural networks tend to perform poorly due to gaps between data points in small data sets [11].

Surrogate modelling in the context of multi-scale simulation for selective laser melting processes used in this project is faced with additional challenges. One difficulty lies in the very high dimensionality of the input and output data of the finite element simulations. Sparse polynomial chaos expansion [2] can deal with reasonably high dimensional inputs. Yet, the dimensionality of the input space in this project is far too large and would result in a computationally expensive surrogate model. Thus, dimensionality reduction in form of principal component analysis (PCA) [6] is applied both to the input and the output space. Principal component dimension reduction and polynomial chaos surrogate models have been successfully combined in [3] or in [14]. In both of these studies, the dimensionality of the model response needed to be reduced. A second difficulty stems from the fact that the data points (constituting the experimental design) can not be chosen freely in the context of this project. This marks a difference compared to traditional surrogate modelling where the computational model can be evaluated at any point and the input space can be sampled with sophisticated sampling techniques. Polynomial chaos expansion has also been used for purely data-driven (machine-learning) applications in [20].

In additive manufacturing simulations surrogate modelling has been ap-

plied in different ways. A large literature review of machine learning methods in additive manufacturing is presented in [1]. For example, machine learning methods are used to optimize process parameters such that the overall build quality is improved. In [18], Gaussian processes are used to model the laser melt pool depths in dependence of the laser power and speed based on experimental data. Contrarily, in the current project the surrogate model is used to predict the whole temperature evolution around the melt pool during the printing process and not only certain parameters.

The remainder of this thesis is structured as follows. First, in Chapter 2, a broad overview of selective laser melting and probabilistic modelling with the polynomial chaos expansion is provided. A short introduction to the multi-scale model, that is the basis of the proposed surrogate model follows in Chapter 3. There, also the surrogate model itself is presented, together with convergence studies of the most relevant model parameters. The surrogate model is then applied to different printing geometries and the results are compared to finite element simulation results of the multi-scale model in Chapter 4. Finally, in Chapter 5, the thesis is wrapped up and an outlook on further research is given.

Theory

2.1 Additive manufacturing

Additive manufacturing, in contrast to traditional subtractive manufacturing techniques, is a process that gradually builds up parts, often layer upon layer. Common additive manufacturing techniques include sheet lamination, powder bed fusion, extrusion or beam deposition for different materials like plastics or metals. Additive manufacturing processes began to arise with laser sintering and stereolithography in the year 1980 [16]. Today, the additive manufacturing process usually begins with a 3D geometry constructed in a computer-aided-design tool. Then, a so-called STL file is generated by slicing the designed part into thin layers. Depending on the manufacturing platform additional files are needed with detailed information about the layers.

The most common types of processes for the additive building of metal parts are powder bed fusion and directed energy deposition [7]. In this project we will only work with powder bed fusion additive manufacturing, more precisely with selective laser melting.

A significant advantage of additive manufacturing, compared to traditional manufacturing, is the possibility of building more complex parts without additional costs. This is particularly beneficial for parts with complex internal structures or with overhang structures. However, for parts with low complexity additive manufacturing is less capable than conventional manufacturing. Drawbacks of additive manufacturing techniques typically include higher costs and difficulties to assure good mechanical properties like residual stresses and distortions. The goal of simulations of the additive manufacturing process is to better understand the process and to optimize the process parameters to improve the aforementioned mechanical properties.

2.1.1 Selective laser melting

Selective laser melting is a manufacturing process in the group of powder bed fusion processes. The process starts by adding a thin layer of powder to the build plate. The thickness of the powder layer depends on the application but is normally in the range of $20\mu\text{m}$ and $100\mu\text{m}$. The whole build process is situated inside a build chamber that is filled with inert gas to protect the metal from oxidation. After the first layer of powder is ready, a high-energy laser is used to melt and fuse the powder at the locations where the part should be built (according to the print data). The laser beam is usually deflected using a laser beam deflection unit (also called the scan head). After the first layer is finished, a second thin layer of metal powder is put on top of the first. This continues until the part is completed. Afterwards, the loose powder is removed and the part is separated from the build plate. Depending on the application, post processing steps like heat or surface treatment are conducted.

Components in selective laser melting can be highly complex. The small laser diameter and layer thickness allows the building of tiny structures. One restriction, however, is the ability of conducting the heat out of the build part. To improve the cooling, sometimes additional support structures are added in the construction process. Also overhangs of minimum angle smaller than about 45 degrees need special care with supportive structures [4].

The selective laser melting manufacturing process has a range of drawbacks that prevent its usage in industry. The printed parts can have high residual stresses that make heat treatment in a post processing step necessary [21]. Also cracking and porous or rough surfaces can reduce the quality of the components mechanical properties. Possible distortions are another problem. These drawbacks mostly stem from the high temperature gradients induced by the high-intensity laser and from the layer by layer heating that belongs to the process intrinsically.

To reduce the impact of these drawbacks, a lot of studies have been conducted. Some important process parameters, which can be optimized to improve the manufacturing process and the quality of the printed parts, have been identified. These parameters include the laser power, scanning speed, hatch spacing and layer thickness. Scanning speed is the speed with which the laser head moves over the powder and hatch spacing is the vertical distance between two lines of laser movement. The parameters are studied both experimentally and in simulations [10].

One more drawback is the long build time. This prevents the usage in applications where a high number of parts are needed. Selective laser melting is rather used for small series of parts, for prototypes and very customized

parts.

2.2 Polynomial chaos expansion

Polynomial chaos expansion (PCE) is a spectral decomposition similar as the Fourier series expansion. It is also commonly used as a technique to construct cheap approximations of a complicated model. It aims at emulating the input-output map of the original, expensive model in a computationally cheap way. It is based on polynomial functions forming a basis representation on to which the model is projected. PCE can be used non-intrusively, this means only evaluations of the model are needed to construct the approximation and no knowledge about the implementation are necessary. Polynomial chaos expansion tries to minimize the global error of the approximation but it does not interpolate at the points of the experimental design.

This section is based on the material in the UQLab manual on polynomial chaos expansion by Stefano Marelli and Bruno Sudret [12]. For a general model with result $Y = \mathcal{M}(\mathbf{X})$ the polynomial chaos expansion can be written as

$$Y = \mathcal{M}(\mathbf{X}) = \sum_{\alpha \in \mathbb{N}^M} y_\alpha \Psi_\alpha(\mathbf{X}). \quad (2.1)$$

The functions Ψ_α form an infinite dimensional polynomial basis and $y_\alpha \in \mathbb{R}$ are their coefficients for any multi-index $\alpha \in \mathbb{N}^M$. The argument $\mathbf{X} \in \mathbb{R}^M$ of the computational model follows a multi variate probability distribution characterized by the density $f_{\mathbf{X}}$. Then the basis functions Ψ_α have to be orthonormal with respect to $f_{\mathbf{X}}$ and the model must have a finite variance

$$\int_{\mathbb{R}^M} \mathcal{M}(x) f_{\mathbf{X}}(x) dx < \infty. \quad (2.2)$$

For any practical usage the infinite sum must be truncated to get a tractable expression. The finite set of multi-indices considered in the truncated expansion is written as $\mathcal{A} \subset \mathbb{N}^M$. Then the expansion becomes

$$Y \approx \mathcal{M}^{PC}(\mathbf{X}) = \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{X}). \quad (2.3)$$

2.2.1 Orthonormal polynomials

On a domain D we can define an inner product in the following way:

$$\langle p, q \rangle_f = \int_D p(x)q(x)f(x)dx, \quad (2.4)$$

where p and q are polynomials and f is a weighting function.

Using this inner product, the notion of orthogonality of polynomials is equivalent to orthogonality in other vector spaces: two polynomials p and q are orthogonal if $\langle p, q \rangle_f = 0$.

In the same way, a family of polynomials $\{p_l, l \in \mathbb{N}\}$ is orthonormal if and only if

$$\langle p_k, q_l \rangle_f = \delta_{k,l} \quad (2.5)$$

with the Kronecker-Delta

$$\delta_{k,l} = \begin{cases} 1, & \text{for } k = l, \\ 0, & \text{otherwise.} \end{cases}$$

Families of orthonormal polynomials can be constructed for any positive weighting function using for example the Gram-Schmidt orthogonalization or the Stieltjes procedure. In the present case, the weighting function is the probability density function (PDF) of model input space. For some probability distributions the families of orthogonal polynomials have special names. Some examples are:

- for the uniform distribution $X \sim \mathcal{U}(-1, 1)$: Legendre polynomials,
- for Gaussian distribution $X \sim \mathcal{N}(0, 1)$: Hermite polynomials,
- for the gamma distribution $X \sim \Gamma(1, k)$: Laguerre polynomials,
- for the beta distribution $X \sim \mathcal{B}(r, s, -1, 1)$: Jacobi polynomials.

Until now, we considered only univariate polynomials. For a random input vector $\mathbf{X} \in \mathbb{R}^M$, whose components are independent of each other, a basis of multi-variate polynomials can be constructed in a straightforward way by using tensor product polynomials of the uni-variate family of polynomials which are orthonormal with respect to the distribution of the i th component of the random variable \mathbf{X} :

$$\Psi_\alpha(\mathbf{x}) = \prod_{i=1}^M \psi_{\alpha_i}(x_i). \quad (2.6)$$

The multi-index $\alpha = \{\alpha_1, \dots, \alpha_M\}$ contains the degree of each polynomial. The orthonormality naturally carries over to the multi-variate case

$$\langle \Psi_\alpha(\mathbf{x}), \Psi_\beta(\mathbf{x}) \rangle = \delta_{\alpha,\beta}, \quad (2.7)$$

with the multi-dimensional Kronecker-Delta, which is equal to one if $\alpha_i = \beta_i$ for all $i \in \{1, \dots, M\}$.

If the components of the random input vector \mathbf{X} are not independent, the tensor product polynomials can not be applied directly. Instead one could

use a so called isoprobabilistic transformation τ to transform the input vector \mathbf{X} to a random vector $\mathbf{Z} = \tau(\mathbf{X})$ which has independent components with marginal distributions with known orthogonal polynomials.

The polynomials are then written in terms of \mathbf{Z}

$$\Psi_\alpha(\tau(\mathbf{X})) = \prod_{i=1}^M \psi_{\alpha_i}(\tau_i(\mathbf{X})). \quad (2.8)$$

This transformation, however, can be very non-linear, leading to a decrease in the overall accuracy of the polynomial chaos expansion [20]. In the following sections it is assumed that the input \mathbf{X} has independently distributed components.

2.2.2 Truncation of infinite-dimensional basis

For the infinite dimensional polynomial chaos expansion to be tractable it is necessary to truncate it. In Eq. (2.3) the truncation is indicated by summing only over multi-indices contained in a finite set \mathcal{A} instead of \mathbb{N}^M . The most straight-forward way is to use only polynomials up to a certain total degree p . The total degree of a multi-variate polynomial is the sum of the degrees of the underlying one-dimensional polynomials. Then we can write

$$\mathcal{A}^{M,p} = \{\alpha \in \mathbb{N}^M : |\alpha| \leq p\}, \quad (2.9)$$

with $|\alpha| = \alpha_1 + \dots + \alpha_M$. The number of elements of this set (or equivalently the cardinality of the corresponding basis) is

$$\text{card } \mathcal{A}^{M,p} = \binom{M+p}{p}. \quad (2.10)$$

Following, two further schemes to truncate the polynomial series are described.

Maximum interaction

The maximum interaction truncation scheme aims at reducing the interaction between different components of the random input by limiting the non-zero entries of the elements of the set $\mathcal{A}^{p,M}$ defined in (2.9). For a maximum interaction coefficient of r this truncation set is written as

$$\mathcal{A}^{M,p,r} = \{\alpha \in \mathcal{A}^{M,p} : \|\alpha\|_0 \leq r\}, \quad (2.11)$$

with the zero-norm giving the number of non-zero elements $\|\alpha\|_0 = \sum_{i=1}^M \mathbf{1}_{\alpha_i > 0}$. With this basis it is possible to reduce the number of elements while retaining polynomials of relatively high degrees. This is especially effective for high dimensional input vectors.

Hyperbolic truncation

Another approach to reduce the interaction terms and keep high degrees in single variables is the hyperbolic truncation scheme that makes use of the so called q -norm

$$\|\alpha\|_q = \left(\sum_{i=1}^M \alpha_i^q \right)^{1/q}. \quad (2.12)$$

The q -norm is similar to ℓ^p -norms for series. The hyperbolic truncation set is then

$$\mathcal{A}^{M,p,q} = \{\alpha \in \mathcal{A}^{M,p} : \|\alpha\|_q \leq p\}. \quad (2.13)$$

2.2.3 Computation of the coefficients

When the truncated basis $\{\psi_\alpha\}_{\alpha \in \mathcal{A}}$ is fixed, the next thing is calculating the coefficients $\{y_\alpha\}_{\alpha \in \mathcal{A}}$ of the expansion (Eq. (2.3)). Two different methods to do this are presented here, a projection method and a least squares regression method. With both these methods the computational model can be treated as a black box and only a set of sample points and the corresponding model response is needed. This set of input/response pairs is also called the experimental design. The experimental design can be given *a-priori* or it can be generated using different sampling techniques. In this project the experimental design is always fixed beforehand because of reasons outlined in Section 3.2.

2.2.4 Projection method

When we take the expectation of the polynomial chaos expansion (Eq. (2.1)) multiplied by the polynomial Ψ_β (equivalent to integrating over the domain of \mathbf{X}) we get

$$\mathbb{E}[\Psi_\beta(\mathbf{X})\mathcal{M}(\mathbf{X})] = \int_{\mathcal{D}_X} \Psi_\beta(x)\mathcal{M}(x)f_X(x)dx, \quad (2.14)$$

$$= \int_{\mathcal{D}_X} \Psi_\beta(x) \left(\sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(x) \right) f_X(x)dx, \quad (2.15)$$

$$= \sum_{\alpha \in \mathcal{A}} y_\alpha \int_{\mathcal{D}_X} \Psi_\alpha(x)\Psi_\beta(x)f_X(x)dx, \quad (2.16)$$

$$= y_\beta, \quad (2.17)$$

using the orthonormality of the polynomial basis. For calculating the expectation techniques from numerical integration like Gaussian quadrature can be used. A N -point Gaussian quadrature rule is defined by a set of weights $\{w^{(l)}\}_{l=1}^N$ and a set of nodes $\{x^{(l)}\}_{l=1}^N$. Applying it to the previous integral

(Eq. (2.14)) yields

$$y_\beta = \int_{\mathcal{D}_x} \Psi_\beta(x) \mathcal{M}(x) f_{\mathbf{X}}(x) dx \approx \sum_{l=1}^N w^{(l)} \Psi_\beta(x^{(l)}) \mathcal{M}(x^{(l)}). \quad (2.18)$$

For multi-variate integrals usually so called tensor product quadrature rules are applied. This means that a one dimensional quadrature rule is applied in every dimension of the input variable. Using such a scheme, the number of model evaluations needed grows very quickly such that computing the coefficients for high dimensional models using quadrature is computationally not feasible. This is one phenomenon of the *curse of dimensionality*. As a remedy to this problem sparse quadrature rules can be applied. Sparse quadrature rules reduce the number of quadrature points by applying a sparse grid, ideally without losing accuracy compared to the tensor product quadrature.

2.2.5 Least squares regression method

A different method for finding the expansion coefficients can be derived by writing it as a least squares problem. First, we rewrite the computational model as a sum of the truncated expansion and a residual r :

$$\mathcal{M}(\mathbf{X}) = \mathcal{M}^{PC}(\mathbf{X}) + r. \quad (2.19)$$

Then the goal is to find coefficients y_α that minimize the mean value of the squared residual (the so called *mean square error*). This can be written as

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}} \mathbb{E}[r^2] = \arg \min_{\mathbf{y}} \mathbb{E} \left[\left(\mathcal{M}(\mathbf{X}) - \sum_{\alpha \in \mathcal{A}} y_\alpha \Psi_\alpha(\mathbf{X}) \right)^2 \right]. \quad (2.20)$$

By defining the coefficient vector $\mathbf{y} = \{y_0, \dots, y_{P-1}\}^T$ and the vector with values of all the polynomials in the truncation set $\Psi(\mathbf{x}) = \{\Psi_0(\mathbf{x}), \dots, \Psi_{P-1}(\mathbf{x})\}^T$ for P being the number of elements in the truncation set, we can write the least squares minimization problem as

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}} \mathbb{E} \left[\left(\Psi(\mathbf{X})^T \mathbf{y} - \mathcal{M}(\mathbf{X}) \right)^2 \right]. \quad (2.21)$$

Given an experimental design of size N with nodes $\mathcal{X} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ and model responses $\mathcal{Y} = \{y^{(i)}\}_{i=1}^N$ the minimization problem (2.21) can be solved by

Ordinary Least Squares (OLS). The mean squared residual can be approximated by a sum of the squared residuals of every sample point in the experimental design:

$$\mathbb{E} \left[\left(\Psi(\mathbf{X})^T \mathbf{y} - \mathcal{M}(\mathbf{X}) \right)^2 \right] \approx 1/N \sum_{i=1}^N \left(\Psi(\mathbf{x}^{(i)})^T \mathbf{y} - \mathcal{M}(\mathbf{x}^{(i)}) \right)^2, \quad (2.22)$$

$$= 1/N (\mathbf{A}\mathbf{y} - \mathcal{Y}), \quad (2.23)$$

where

$$\mathbf{A} = \{ \Psi(\mathbf{x}^{(1)}), \dots, \Psi(\mathbf{x}^{(N)}) \} \quad (2.24)$$

is the matrix containing the polynomial values of all the sample points (called the experimental matrix). To find the coefficient vector that minimizes (2.22), we can now apply the least squares method to get the least squares solution

$$\hat{\mathbf{y}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathcal{Y}. \quad (2.25)$$

Compared to computing the coefficients with Gaussian quadrature, with the least squares method any set of points can be used for the experimental design. For ordinary least squares, the only condition is that the number of sample points N must be larger than the number of unknown coefficients P . Otherwise the matrix $\mathbf{A}^T \mathbf{A}$ (also called the *Gramian* matrix of \mathbf{A}) is not invertible.

2.2.6 Adaptive polynomial chaos expansion

Using methods for assessing the error of a PC expansion it is possible to construct adaptive methods for building the final PCE. One possibility is basis-adaptive PCE. This means that the basis of the expansion is gradually built up from a very small initial basis always assessing the generalization error of the expansion. In the end, the basis with the smallest error is selected.

A different family of adaptive algorithms are sparse PCE [2]. In sparse PCE the starting point is a large basis. Subsequently, certain basis functions are removed in order to get a sparse representation without increasing the prediction error. Having a sparse representation helps to reduce issues with overfitting that may occur when a too large basis is chosen. Especially because in many problems only low order interaction terms are important. Besides that sparse bases are also beneficial in terms of computational efficiency.

The basis truncation methods presented in Section 2.2.2 can be used here. An alternative way to produce a sparse PCE are algorithms that penalize models which are not sparse. Such models are commonly used in statistical regression and classification (e.g. LASSO or Ridge regression).

In the setting presented here such a penalty can be written as

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}} \mathbb{E} \left[\left(\Psi(\mathbf{X})^T \mathbf{y} - \mathcal{M}(\mathbf{X}) \right)^2 \right] + \lambda \|\mathbf{y}\|_1. \quad (2.26)$$

The 1-norm is defined as $\|\mathbf{y}\|_1 = \sum_{\alpha \in \mathcal{A}} |y_\alpha|$. This term causes the algorithm to choose a solution with small coefficients. The parameter λ can be chosen such that the resulting coefficients give the smallest generalization error. From the point of view of sparsity, it would be preferable to have a term that penalizes the number of non-zero coefficients, i.e. use the 0-norm. But this problem is computationally infeasible, thus the penalty is relaxed to the 1-norm.

2.3 Principal component analysis

This section uses the well written tutorial on principal component analysis (PCA) in [17]. PCA is a method for analyzing a set of vectors in \mathbb{R}^d and finding a suitable lower dimensional representation of them. The goal is to construct a set of uncorrelated variables which represent the original data, the observations, in a way that the noise in the data is reduced and that a representation with low redundancy is found. The PCA tries to find the best *linear* transformation of the data with these properties.

Writing the i^{th} observation, $i \in \{1, \dots, n\}$, as a column vector $x_i \in \mathbb{R}^m$ we can define the data matrix as

$$\mathbf{X} = [x_1, \dots, x_n] \in \mathbb{R}^{m,n}. \quad (2.27)$$

The columns of \mathbf{X} are the observations and the rows are the measurements of a certain component. The basis transformation of the PCA can then be seen as multiplying \mathbf{X} with a suitable matrix $\mathbf{P} \in \mathbb{R}^{m,m}$:

$$\hat{\mathbf{X}} = \mathbf{P}^T \mathbf{X}, \quad (2.28)$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{m,n}$ is the matrix containing the transformed observations as columns, also called the *component scores*. The columns of \mathbf{P} form the basis of the transformed space. In PCA this basis should be orthonormal and the basis vectors are called the principal components. Now the question remains how to choose this new basis $\{\mathbf{p}_1, \dots, \mathbf{p}_m\}$.

To answer this we look at the covariance matrix of \mathbf{X} and $\hat{\mathbf{X}}$ that is defined as

$$\mathbf{C}_\mathbf{X} = \frac{1}{n-1} \mathbf{X} \mathbf{X}^T \in \mathbb{R}^{m,m}. \quad (2.29)$$

The sample variance of each measurement is represented through the diagonal terms of the covariance matrix and the covariances between different

measurements are on the off-diagonals. Consequently, we call the covariance matrix of the transformed observations $\mathbf{C}_{\hat{\mathbf{X}}}$. In order to have uncorrelated transformed variables \hat{x}_i , the off-diagonal entries of $\mathbf{C}_{\hat{\mathbf{X}}}$ must be zero. Additionally the diagonal entries of $\mathbf{C}_{\hat{\mathbf{X}}}$ should be ordered according to their size to get an ordering of the importance of each transformed variable in explaining the variance of the original observations. This means the sought transformation should diagonalize the covariance matrix of \mathbf{X} with an orthonormal basis.

A conceptually easy way to think of what the principal component analysis does is the following: In a first step, choose the first basis vector $\mathbf{p}_1 \in \mathbb{R}^m$ as the direction in which \mathbf{X} has the largest variance. The second direction should again maximize the variance of \mathbf{X} . However, the second basis vector \mathbf{p}_2 must be orthogonal to \mathbf{p}_1 . These steps can be repeated until m basis vectors are found which all are orthogonal to all the others. Practically however, the principal components are found by either diagonalizing the covariance matrix $\mathbf{C}_{\mathbf{X}}$ or by computing the singular value decomposition of \mathbf{X} . In the first case, the principal components correspond to the eigenvectors of $\mathbf{C}_{\mathbf{X}}$. The transformed variables are then computed according to Eq. (2.28). To account for the sensitivity of the PCA to the scaling of the observation usually the mean of the measurements is subtracted first from the observations before computing the covariance matrix (normalization).

Model Development

In this chapter the process of model development is described. Firstly, the original multi-scale model developed by Pooriya Ghanbari at Empa is presented, secondly the surrogate model developed in this project.

Modelling the process of additive manufacturing is needed to better understand what is going on and how to improve the manufacturing process. Due to the localized laser source very large temperature gradients evolve in vicinity of the melt pool. These temperature gradients and the very fast heating and cooling of the material may lead to residual stresses [21]. Residual stresses can severely impact the material quality of the final part and can cause distortions, delamination or cracking which is not desirable. Simulations are used to predict the residual stresses in printed parts, to understand their formation and to optimize process parameters in order to minimize the formation of residual stresses.

There exist various models and scientific simulations to study the process of selective laser melting. That is very complex due to the large range of length and time scales included. The powder grain sizes are usually in the range of micrometers whereas the size of the part to be built can be in the order of centimeters. Currently, it is practically infeasible to accurately simulate the manufacturing process in full complexity on all length scales even with a super computer. There are simulation models focusing on the full thermo-fluid-dynamic description of the melt pool with laser absorption and heat transfer mechanisms like conduction and convection. These simulations are so costly that they usually only simulate a few layers of the whole component. On the other hand there are simulations that focus on the whole part and apply approximations to the heat transfer and the melt pool dynamics to lower the computational costs. These component-scale models aim at predicting residual stresses or distortions in the final component, which is not possible with models acting on a smaller scale. A review of different process models and simulations for additive manufacturing can be found in [8].

3.1 Multi-scale model

To improve the accuracy of simulations on the component-scale level, a multi-scale simulation is being developed at *Empa*. The goal of this multi-scale is to have a more accurate model in the melt pool region without substantially increasing the computational costs. In this section the already existing multi-scale simulation is briefly described.

The multi-scale model is based on a finite element model simulated in Abaqus. It is composed of one global simulation and multiple local simulations. The laser is applied to the finite element model as a heat source using Abaqus subroutines. The laser absorption of the powder is included to compute the increase of total energy of elements in lower layers.

3.1.1 Global simulation

The global model includes the complete printing part and it simulates the duration of the whole printing process. Each printing layer is simulated using a step with laser heating and a cooling step in the end. After the cooling step, a new powder layer is added by changing the properties of the elements on top of the last printing layer. Because the global model contains the whole component, the elements of its finite element mesh need to be coarse in order to be able to simulate it within a reasonable time. The time steps are rather large as well. This leads to a simulation with limited accuracy.

3.1.2 Local simulation

Compared to the global simulation, the local simulations only model the region directly around the laser and the melt pool. In the course of a layer multiple local simulations are used. The first local simulation starts at the beginning of the layer and is positioned such that the laser is in its center. The laser movement is then simulated with the local model for a short time period. The area of the next local model is shifted in the direction of the laser such that the laser is again in its center. In this manner, the local models in the end cover the whole layer. The local models use the temperature data from the global model as boundary condition. The initial temperature values come from the previous local model except for the very first model in each layer, which takes the initial values from the global model.

The finite element mesh of the local model is much finer than the one of the global model and also the time steps are smaller. This gives a more accurate simulation in the region of the melt pool where high temperature gradients exist and high accuracy is needed.

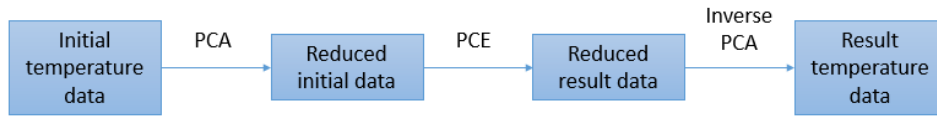


Figure 3.1: Proposed workflow of the surrogate model with dimensional reduction and reconstruction of the features in the original space.

Finally, the simulation results of the local and the global simulations can be combined to get the temperature evolution in the whole printing component. This procedure helps in getting accurate results of the printing process within a reasonable simulation time. The most time consuming part of the simulation is solving all the local models. The solution of one local model is not really expensive but the number of local models to solve is typically large. These local models always have a similar geometry, slightly different initial and boundary conditions and, of course, model the same physical system of the laser beam. The goal of this project is to substitute these local finite element models with a cheap surrogate model to reduce the computation time for the whole multi-scale simulation.

3.2 Surrogate model

In this section we describe the surrogate model that is developed in this project. The surrogate model is a data driven model to replace the local simulations of the finite element model. This means that the surrogate model is not developed based on physical principles but it rather tries to find and approximate a map between some input and output data. In the local model, the input data are the initial temperatures of all the nodes in the local finite element model and the temperature values of the nodes on the boundary of the local model for each of the ten time steps that one local model covers. The output (or result) data of the local simulation are the temperature values of the nodes for all ten time steps.

The surrogate model of this project consists of two fundamentally different parts: dimension reduction with the principal component analysis and the statistical model itself (polynomial chaos expansion). The method of principal component analysis (PCA) is described in more detail in Section 2.3. In this project we use PCA to reduce the number of dimensions of the input and result data and to extract the important features because the dimension of the input data (about 2500) and result data (about 22000) is too high to be handled directly.

The goal of the actual surrogate model is then to find the mapping between the space of reduced input and result temperatures. For this task a polynomial chaos expansion (PCE) model is used. The corresponding theory is

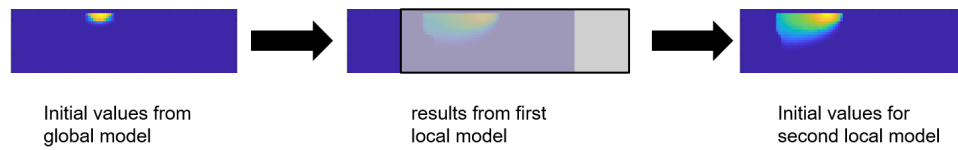


Figure 3.2: Workflow for the prediction of a layer. The prediction results of a local model are interpolated to the coordinates of the next local model and used as input values to the next surrogate model.

described in Section 2.2. The parameters of the polynomial chaos expansion model are determined using training temperature data from the finite element model. With the high-fidelity FE model, the temperature evolution can only be computed starting from the beginning of a printing layer because the local models (and thus the time steps) depend on their predecessors. As the first local model only depends on the global model the thermal data can be computed on each layer independently of the previous layers.

It is important to distinguish between the dimension reduction by principal component analysis and the polynomial chaos expansion. The principal component basis can be computed from the same training data as for the PCE. Then it is specifically suited for the training data set and provides very good reconstruction rates. Yet, for slightly different test data the performance of the reconstruction might drop significantly, especially if the test set contains features the training set does not. An alternative is to build the basis from specially gathered (even artificial) data that provide a good overview of possible features arising in the local simulations. This can be done beforehand without actually solving the finite element model. It is only necessary to have a large set of initial and boundary temperatures for the input data and a set of typical result temperatures for the output data. The principal component spaces for the input and the output can be trained independently. In the current project, the polynomial chaos expansion model is trained and built anew every time it is used for a new printing setting. It is trained with training data that must be computed with the expensive finite element model. Thus, it is very important to investigate how much training data is really needed. Using too few training data will lead to poor surrogate model results while using very much training data is computationally expensive. In a further step it should be studied if it is possible to train the polynomial chaos model for a more general setting and not just for one printing condition.

With the model described above it is possible to predict the temperatures of one local model (including ten time steps). For practical purposes in the simulation it is necessary to be able to predict the temperatures of all local models in one layer given only the initial temperatures from the first

local model of the layer and the boundary temperatures for every time step. As described before, the covered area of successive local models is shifted in the laser direction but has a substantial overlap. In the finite element code the initial temperature of a local model is found by interpolation of the temperature values of the previous local model. In the surrogate model this is achieved in a very similar way. The temperature results of the last time step of one local model are interpolated to the nodes of the next local model. The boundary temperatures for the next ten time steps from the global model are added. These temperature values then form the input data for the surrogate model to compute the next ten time steps.

All computations regarding the polynomial chaos expansion, its training and evaluation are done using the UQLab¹ Matlab software from the Chair of Risk, Safety and Uncertainty at ETH Zurich [13].

3.2.1 Cross validation

When a statistical model is built it is not always simple to determine its quality. The difficulty is due to models often being used for predicting responses to new observations where the true model response is not known. Usually, gathering additional data for model validation is expensive or not possible at all.

The error of the model response to a new data point is called the *generalization error*. One strategy to determine this error is to scale the prediction error of the model on the training data set in some sophisticated and mostly theoretically justified way. A different widely used approach is *cross validation* or the *validation set* approach. In the validation set approach one splits the original data randomly into two sets: a training data set and a validation set. All the steps required to build the model are conducted only on the training set. When the model is built, it is evaluated on the second data set retained as the validation set. Based on these observations the test error of the model is estimated. This approach has the advantage that the model is validated on a independent set of observations. When finally the productive model is built, it is of course trained on the whole data set. This method usually works well but the error obtained is dependent on the (random) data set chosen for validation.

In k -fold cross validation, the observation data is randomly split into k similarly sized data sets. Then, k different models are fit. For each of these different models one data set is not used for training. To get the generalization error, the models are evaluated on that data set. The generalization error is averaged over the k different models. A typical choice for k is 5 or 10. The higher k the better the estimation of the prediction error is. However,

¹<http://www.uqlab.com>

the cost of fitting k models also increases substantially (if no computational short-cut exists). In the case of k being equal to n , the size of the total data set, it is called *leave-one-out cross validation*.

3.2.2 Error measures

One frequently used error measure is the normalized root mean square (RMS) error. This measure computes the L^2 deviation of the model responses from the true data and normalizes that deviation by the standard deviation of the data. The following formula shows how the normalized RMS error is calculated on a validation set of size n_{val} with true observation results y_i , model responses \hat{y}_i and $\bar{y}_i = 1/n_{val} \sum_{i=1}^{n_{val}} y_i$ being the mean value on the validation set:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n_{val}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_{val}} (y_i - \bar{y}_i)^2}}. \quad (3.1)$$

3.3 Theoretical error estimates

In the combination of PCE and PCA it is difficult to obtain theoretical estimates for the prediction error. The total L_2 prediction error we are interested in is written as

$$\Sigma = \|\mathcal{M}(\mathbf{x}) - \hat{\mathcal{M}}(\mathbf{x})\|^2, \quad (3.2)$$

with the true finite element model \mathcal{M} and the surrogate model $\hat{\mathcal{M}}$. As shown in [3], this total error can be decomposed into different parts. There is an emulation error from the polynomial chaos expansion

$$\Sigma_{PCE} = \sum_{i=1}^{K_{output}} \mathbb{E} \left[(B_i - \hat{B}_i)^2 \right], \quad (3.3)$$

where K_{output} is the number of principal components used to decompose the output space and B_i and \hat{B}_i are the true and emulated (from the PCE) principal components, respectively. The error of the polynomial chaos expansion can be practically estimated with the *leave-one-out* error of every PC model. The *leave-one-out* error is efficiently computed during the training process of the PC model in UQLab and is readily available afterwards. This error depends on the truncation parameters of the polynomial basis used in the polynomial chaos expansion and on the experimental design. In the adaptive PCE algorithm, which is applied in UQLab, ranges for the truncation

parameters can be prescribed. The algorithm then finds the truncated basis with the lowest *leave-one-out* error within these bounds.

A different error contribution stems from the application of principal component analysis to reduce the dimension of the input and the output space. The covariance matrix of the data is computed with a finite sample. This results in wrong empirical eigenvalues and eigenvectors (compared to utilizing the actual (unknown) covariance matrix). This error can be decreased by computing the principal components on a data set that provides a reasonable selection of the possible data.

The principal component basis is truncated after the first few components. The error made in this step can be estimated by looking at the non-retained eigenvalues of the decomposition. This error can be decreased by using more principal components. However, more principal components can substantially increase the training and evaluation time of the surrogate model. Additionally, many principal components contain less and less predictive information. Training the PCE model with these components might lead to fitting noise.

The explanations above are specific for the prediction of a single frame but can also be applied to the prediction of a whole layer. Then, additional error terms need to be considered because of the interpolation and the usage of previous result temperatures as new input conditions.

3.4 Building the principal component space

The theoretical details of the principal component analysis are explained in Section 2.3. In this section we describe how the principal component basis is computed in this project. As mentioned before, dimensionality reduction is needed because the input data as well as the output data for the surrogate model is very high-dimensional and could not be used for surrogate modelling directly. With help of the principal component analysis the high dimensional data is transformed in to a low dimensional representation of principal components, which still contain the largest part of the variance of the original data. In Figure 3.3, the first four principal component eigenmodes of a set of initial temperature data are shown.

Using principal component analysis dimensional reduction methods in this project gives two important advantages. First, the computational costs of the surrogate model are drastically reduced (and surrogate modelling how it is used in this project is only made possible). Second, dimension reduction helps to find important features and structure in the data. This is specially important for vector valued output data. The different polynomial chaos models for each output dimension are constructed independently from each other. Yet, the output components themselves (the 2D temperature field in

3. MODEL DEVELOPMENT

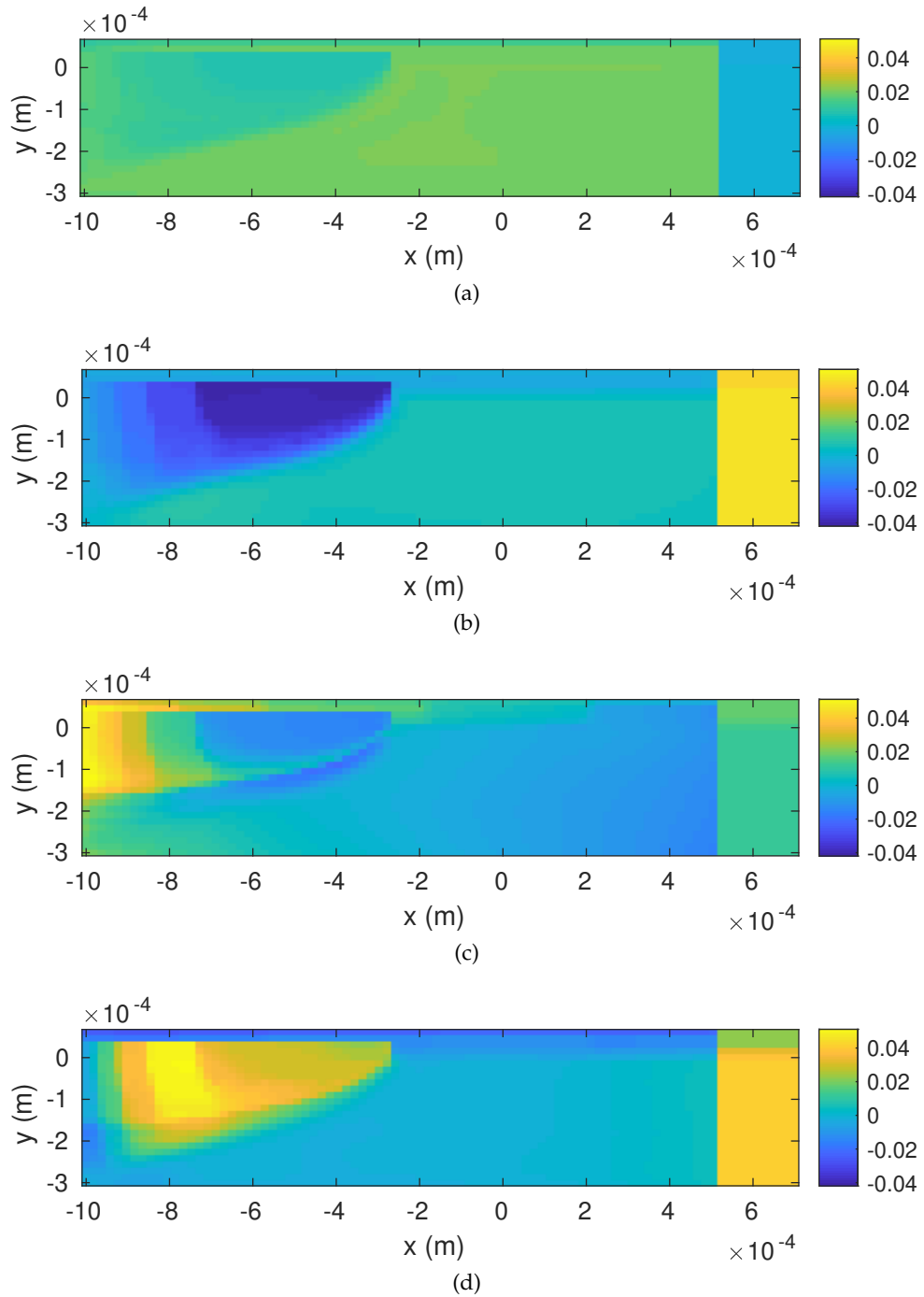


Figure 3.3: First four elements of the principal component basis computed from normalized initial temperature data.

3.4. Building the principal component space

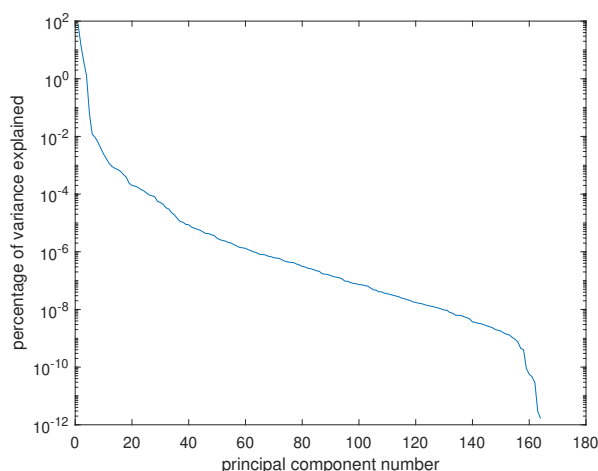


Figure 3.4: Percentage of the total variance in the initial temperature data that is explained by each principal component.

our case) might depend on each other. The principal components include this dependence. Additionally, performing a principal component analysis can help to reduce the noise in the data.

To see how well the principal component analysis works for a data set, one often looks at the percentage of the total variance of the data set that is explained by each principal component. In Figure 3.4 that is visualized for a set of typical initial temperature values. We see that the percentage of variance that each principal component explains drops very fast. This is a good sign as it means that already very few principal components capture and explain the largest part of the total variance in the data.

In this project, besides explaining a large part of the total variance, another property of the principal component analysis is very important: reconstruction. As the result data is compressed, the output of the polynomial chaos expansion model is given in terms of principal components. To be of any use, the output of the model must be transformed back into the original result space. Since the principal component transformation is a linear transformation with a matrix with orthogonal columns, the inverse transform is straight forward to define. The inverse transformation is not exact here because the covariance matrix of the data is only estimated from a finite data set and because the number of principal components is truncated (as mentioned in the previous section). Thus, some information is lost through the reconstruction. We need to make sure that the reconstruction error is small enough for our purposes. This is especially important as the reconstructed temperature profiles of one local model are used as input data to the next surrogate model (cf. Section 3.2 and the description of the prediction of a layer).

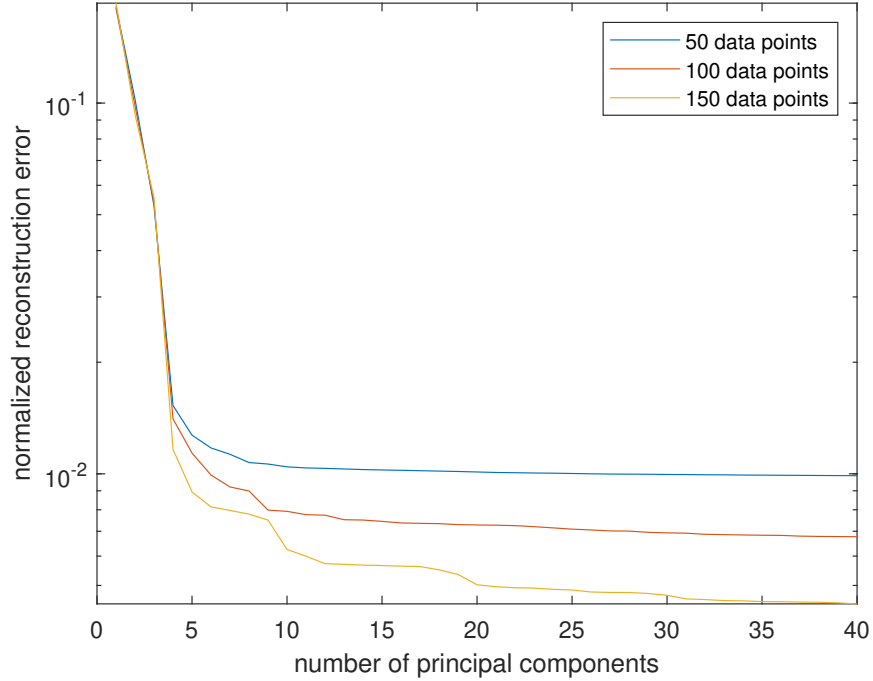


Figure 3.5: Normalized reconstruction error of the principal component dimension reduction for the result temperature data set of the geometry 3.7. The principal component basis is computed with 50, 100 and 150 random data points of 165 in total.

In Figure 3.5 the results of a convergence test for the reconstruction is shown. For that test, we use the result temperature data gathered in the simulation of the geometry shown in 3.7. This data set has 165 data points in total. For different numbers of randomly selected data points, we build the principal component basis, project the remaining points of the data set on to this basis and make the back-transformation of the transformed data points. In the end, we determine the normalized difference between the reconstructed and the original temperature data (the reconstruction error). This error is plotted versus the number of principal components utilized to perform the reconstruction.

The more principal components are used for the reduction, the smaller the error of the reconstructed data (and the better the reconstruction properties of the dimension reduction method). Of course, using more principal components comes with the price of higher computational costs. The lowest reconstruction error that can be achieved is determined by the quality of the data set for training the PCA.

The principal component basis can be calculated using the training data from the finite element simulation. If the training data is chosen well and

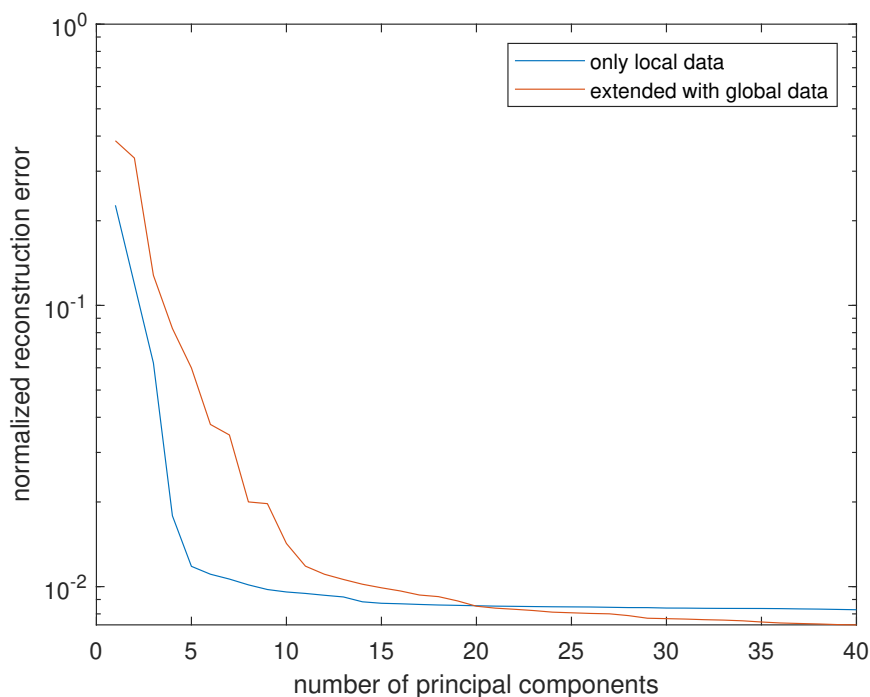


Figure 3.6: Normalized reconstruction error of the principal component dimension reduction for the result temperature data set of the geometry 3.7. The principal component basis is computed once with only local data and once with additional global data.

contains all the representative features of the temperature distributions that can appear in the setting studied, this approach works very well. However, with this strategy, there must be enough training data available. For example in Section 4.1, the surrogate model is trained with a total of only 10 data points from local simulations of two layers. In that case, only 9 principal components can be computed and used for the polynomial chaos expansion.

From the global simulation, rough estimates for the temperature distribution are known. Combining these estimates with the given training data to build the principal component basis also leads to relatively good results. In Figure 3.6 the normalized reconstruction error for different sizes of principal component bases are plotted for temperature data from the square geometry. The principal component basis is computed with only 20 data points of the local temperature data set, extended with data from the global temperature simulation (red curve). For comparison, the blue curve shows the reconstruction error of a principal component basis computed on half of the available data from the local simulations (82 data points). For small principal component bases, utilizing only the local data gives much lower errors. For larger bases both methods perform similarly well. If the training set of

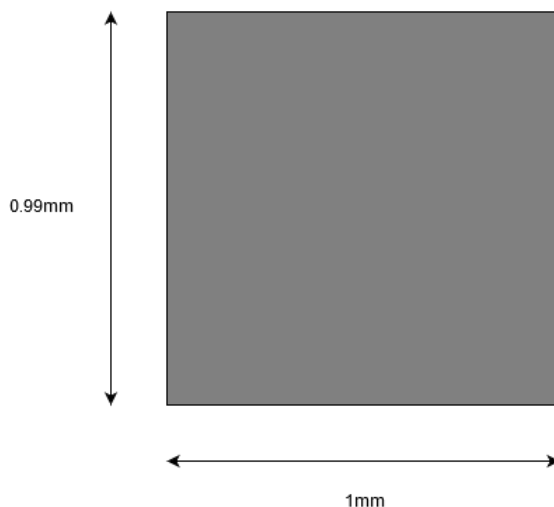


Figure 3.7: Sketch of the square block printing geometry.

local temperature data is large enough to compute the required number of principal components, it seems best to only use that data. However, if not enough local data is available (e.g. with 20 data points like in Figure 3.6, only 19 principal components could be computed), the usage of additional global data helps to include more features in the training data.

3.5 Single model prediction

For the first convergence experiments, the fact that training data is only available in whole layers is ignored and we start by studying the behavior of the model when predicting only the results of a single local model. To achieve this, the surrogate model is trained with temperature data randomly chosen from the local finite element simulations.

3.5.1 Input dimension convergence test

In a first convergence test we want to study the influence of the number of principal components used as input parameter for the polynomial chaos expansion (PCE) model. In this test the focus lies on the PCA of the initial and boundary conditions and neither on the PCE model nor on the PCA of the result data. The parameters of the polynomial chaos expansion model are set to be (cf. Section 2.2):

- adaptive polynomial degree ranging from 1 to 20,
- adaptive q -norm truncation with $q \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$,

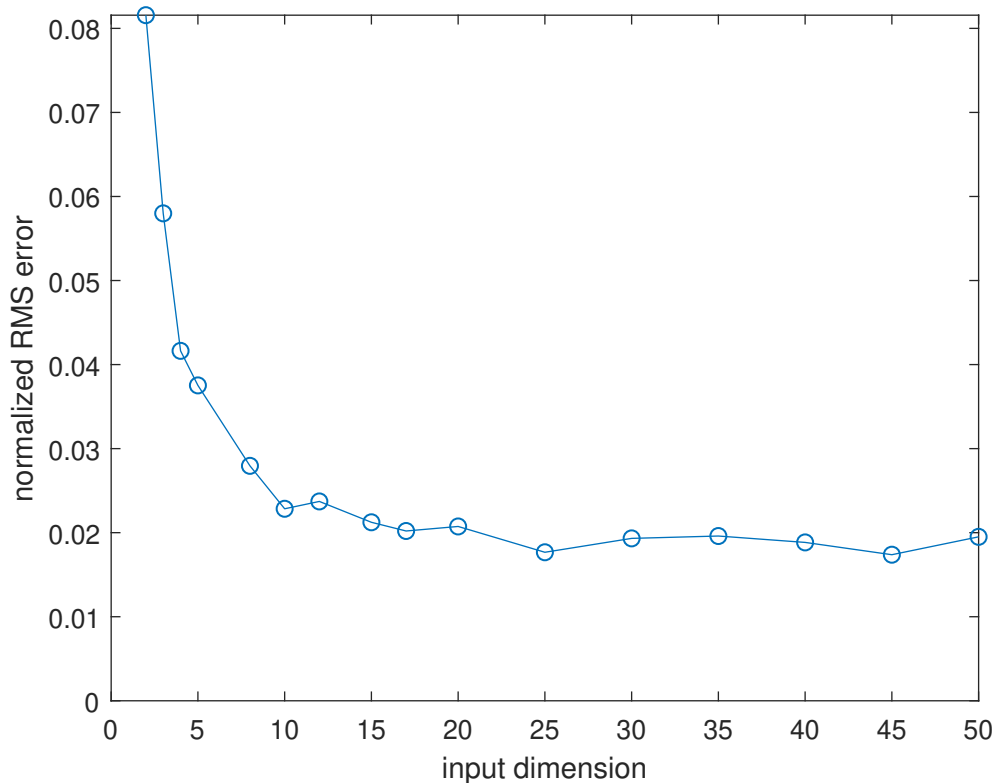


Figure 3.8: Normalized root mean square error plotted against the number of principal components used as input parameters for printing a solid block.

- maximum interaction order of 2 (3 was also tested and the results were very similar).

The principal component space of the results was truncated to the first 30 modes.

The geometrical setup is printing a simple rectangular geometry without any holes or further difficulties. It is shown in Figure 3.7. In this geometry there are 33 printing layers with 5 local models each. This means we have set of 165 pairs of initial and result data. In order to postpone the task of building a good PCA basis we use all the available data for building the PCA. Of course, in a practical use case with new data this would not be possible (cf. Section 3.4).

We now compute the normalized RMS prediction error (see Eq. (3.1)) for different numbers of principal components of said basis as input to the polynomial chaos model. The RMS error is computed with 2-fold cross validation. The resulting errors are shown in Figure 3.8. Initially, the error decreases rapidly for an increasing input dimension. Utilizing more than about 20 modes of the input data, does not increase the prediction accuracy anymore.

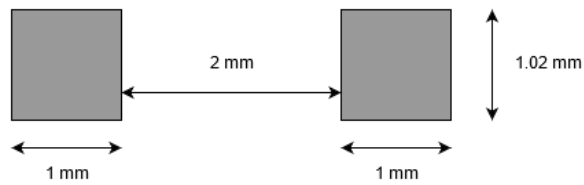


Figure 3.9: Sketch of the geometry for printing two equally sized blocks with a void area in between.

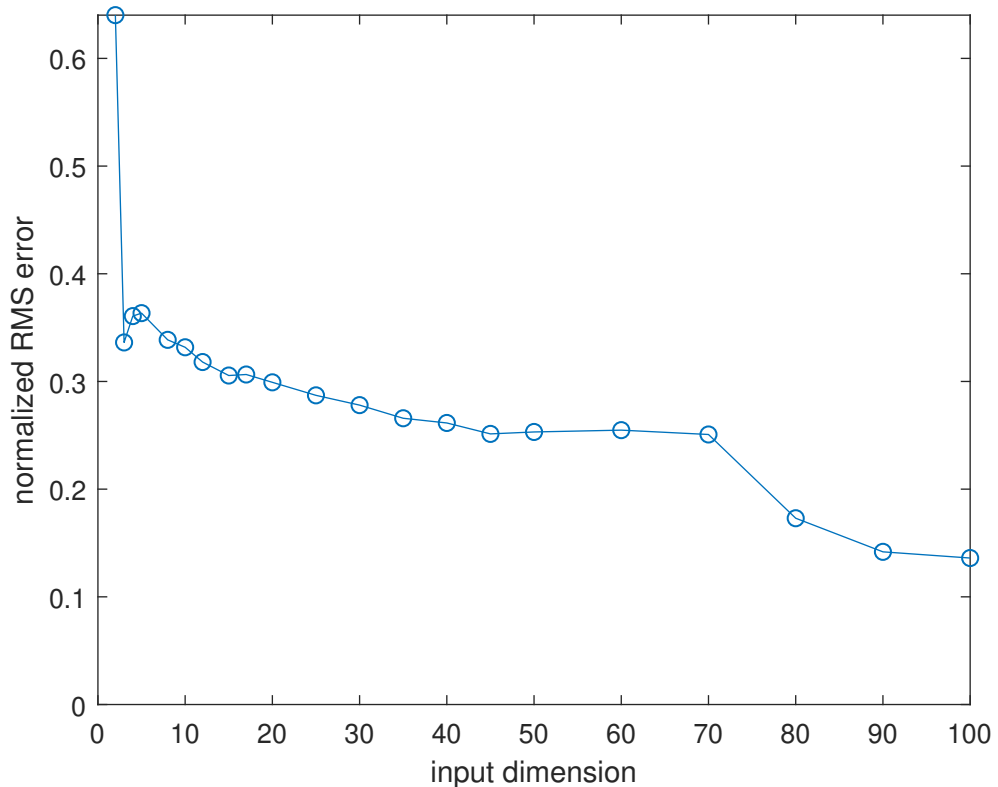


Figure 3.10: Normalized root mean square error plotted against the number of principal components used as input parameters in the setting of printing two columns (Figure 3.9).

The normalized prediction error then is in the range of 2% which is quite small already. Therefore, for further studies with this geometry input dimension of about 16 will be chosen.

Next, the same test is conducted for a more complicated geometry, see Figure 3.9. This geometry contains a region where nothing should be printed. This region will be filled with powder and there is no laser activity. In the local simulations there, the temperature is only conducted and no additional heat is added to the system. For the surrogate model the different behavior in the two regions proves to be an additional difficulty as we can see

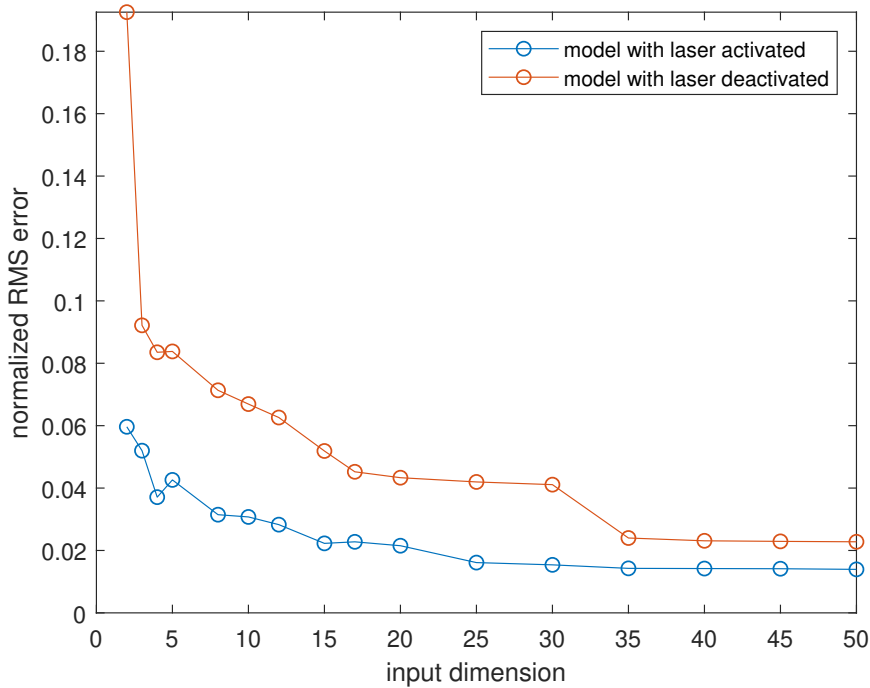


Figure 3.11: Normalized root mean square error plotted against the number of principal component input parameters in the setting of printing two blocks (Figure 3.9) with two different surrogate models depending on the laser activation.

in Figure 3.10. This is a plot of the same convergence test as done before for the simpler geometry. The lowest normalized root mean square error in this setting is about 13% for 100 principal component modes. The error is much higher than in the previous example and also the minimal error is reached for a much higher number of principal components. Fitting the polynomial chaos expansion with such a high dimensional input vector is computationally very demanding and it may lead to overfitting.

When looking at predictions of the temperature at certain locations we see that the model can not differentiate between regions where the laser is activated and regions where it is not. This is not surprising as the model is not provided any information about laser activity. The logical next step is then to include such information into the model. The behavior of the physical system to be emulated is very different in regions with laser activation and in regions without. Therefore, laser activation is not included as an additional input parameter to the polynomial chaos expansion but two separate models for the two states of laser activation are trained. To this end, the training data is split into two parts according to laser activation and for each part a separate principal component basis is computed. For the principal com-

ponent modes of these bases, the two polynomial chaos expansion models are trained. In this setting we again conduct the same convergence test as before. The results for the two different models are shown in Figure 3.11. The model with laser activation performs similarly well as the model for the simple geometry (cf. Figure 3.8). The local models in the solid blocks look very very similar in both cases. The surrogate model for the region without laser activation does not reach such a small error and the error decays much slower in terms of the number of principal components. The physical processes involved in the models without laser activation are less complex than in models with laser activation. However, the temperature profiles appearing in the models without laser activation are much more diverse than in the models with, as the latter always have the very dominant peak which in the former is smoothed out over time. Therefore, the dimensionality reduction might be less accurate for the models without laser, which leads to higher errors. Another reason could be that the processes in the models with laser activation are dominated by the laser source in the simulation domain. In the models without laser, the temperature evolution depends much more on the far-field boundary conditions, which lie outside of the surrogate model domain. At the same time, FE models are much cheaper to solve in regions without laser activation than in regions with (because larger time steps can be applied). Finally, the relative error might be larger because the normalization factor is smaller in the models with deactivated laser due to the temperatures being lower on average.

3.5.2 Output dimension convergence test

To gather more insight in the importance of the principal component basis for the model output (the result temperatures of ten frames), we conduct a similar study for the number of principal components as in the previous section. In this convergence study we keep the number of principal components used for the model input constant at 20 and vary the number of components of result space basis. The other parameters are equal to the ones in the previous section. Again, we first look at the simple filled block. The results are shown in Figure 3.12. There, we see that the minimal root mean square prediction error on the test set is reached already for 20 principal components as result parameters. For the more complicated geometry with the two blocks, we again use two different models, one that is trained with data from the region where the laser is active and one trained with data from the region where the laser is not active. In the results in Figure 3.13 we see a similar effect as for the convergence test regarding the input dimension. For the model where the laser is active, the root mean square error quickly decays to a value similar as in the experiment with the solid block. For the model without laser activation, however, the error decays more slowly and reaches its minimum again for a higher number of principal components in

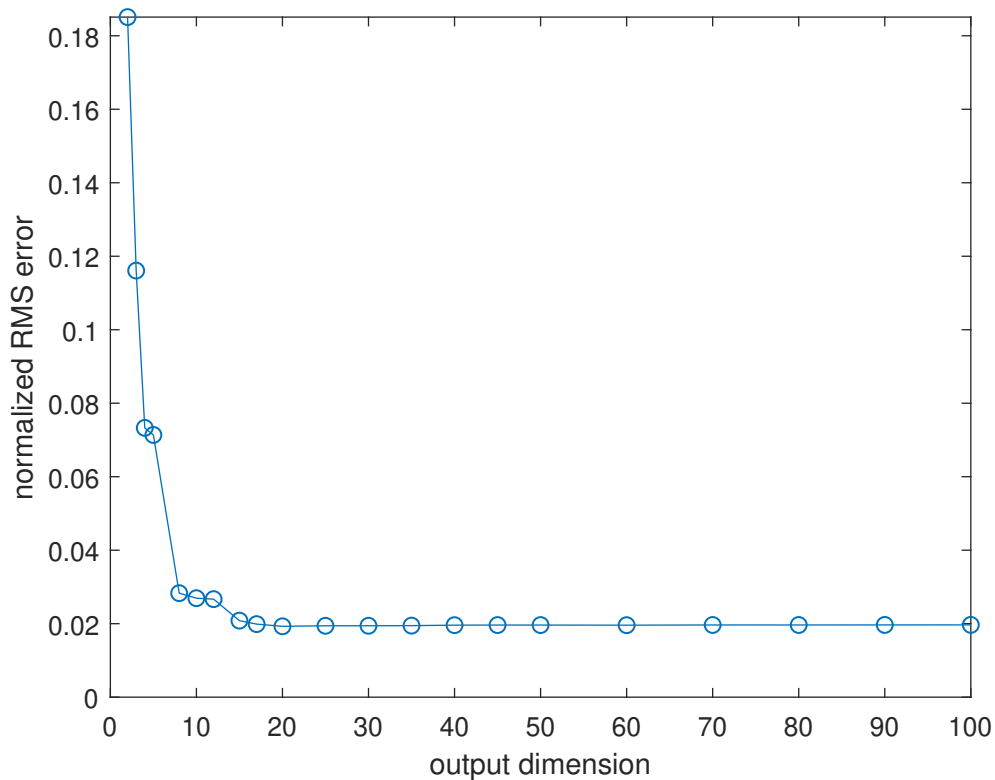


Figure 3.12: Normalized root mean square error plotted against the number of principal components used as output parameters in the setting of printing a solid block.

the result space.

The cost of training and evaluating the PCE model increases only linearly with the number of output parameters (for every parameter an independent model is built) in contrast to the number of input parameters. Therefore, it is affordable to use more modes for the result space than strictly needed according to the convergence tests. In further experiments the result space will always be reduced to 30 dimensions.

3.5.3 Influence of experimental design size

In the previous experiments the principal components were always computed using all the available data. This was done to rule out the influence of not having an optimal reduced basis. Now, in this experiment we want to see just this influence. To that end, we choose only a subset of all observations for building the principal component basis. With this principal component basis we then repeat the experiment from Section 3.5.1 for the solid block geometry, i.e. computing the normalized root mean square test error using different numbers of principal components for fitting the PCE

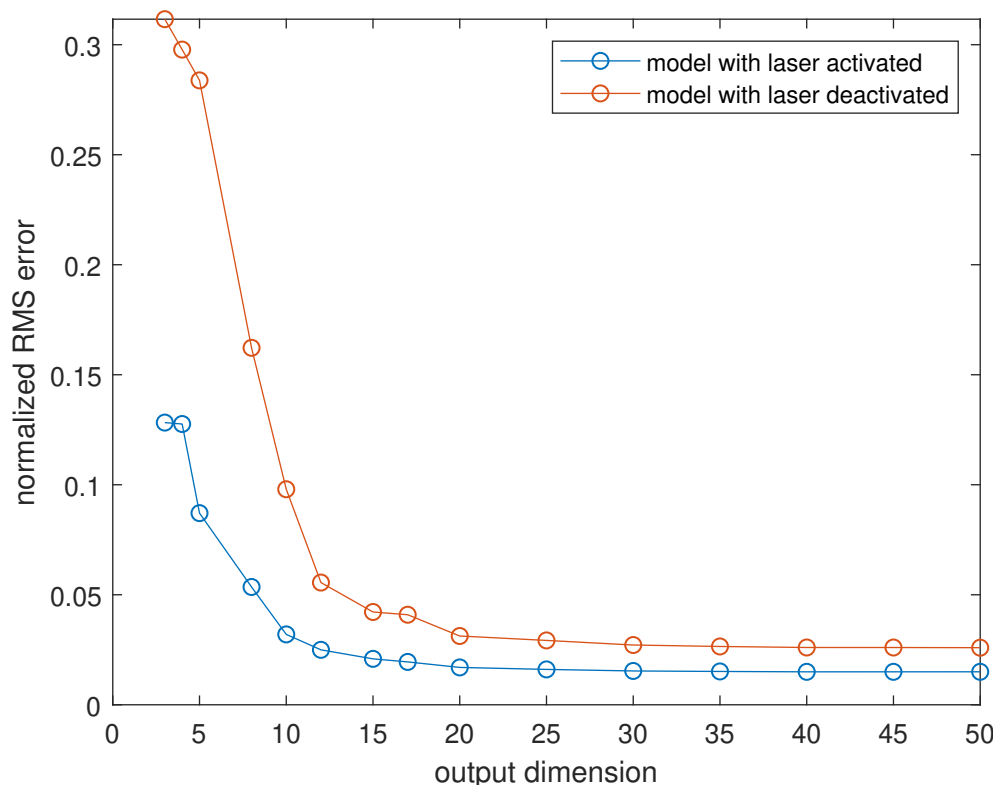


Figure 3.13: Normalized root mean square error plotted against the number of principal components used as output parameters in the setting of printing two columns (Figure 3.9) with two different models depending on the laser activation.

model. We repeat this test for principal component bases built with the full, half and a quarter of the data set. The three different error curves are shown in Figure 3.14. Note that when using only a quarter of the data set to build the PC basis it is not possible to find more than 40 principal components because the data set itself consists only of 41 observations. Therefore the yellow curve stops at an input dimension of 40.

In the plot of the results we see that the error curves for the differently sized experimental designs are very similar. For all three experimental designs the plateau of the root mean square error is reached for about 20 principal components. Also, the smallest root mean square error reached does not differ significantly between the three curves. It seems that using 40 observations to build the principal component basis (as in the yellow curve) is enough to find the features needed to train a good surrogate model.

Now we have seen that in the simple geometric setting the size of the experimental design does not play such an important role for computing the principal component basis. The polynomial chaos model was still computed

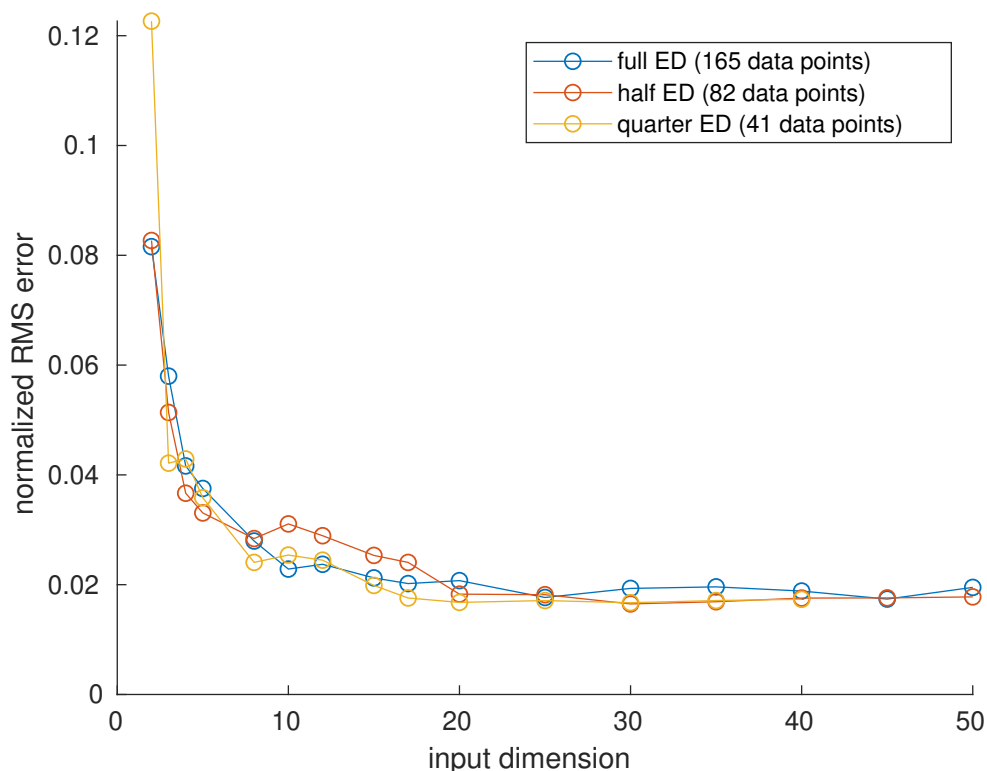


Figure 3.14: Normalized root mean square error plotted against the number of principal components used as input parameters for printing a solid block. For the different curves, the principal components were computed using training data sets of different sizes. In the blue curve the full, in the red curve half and in the yellow curve a quarter of the experimental design (ED) was used for building the principal component space.

using the complete experimental design, to not mix the different errors of the two steps. In the next experiment we are interested in how the polynomial chaos model performs for different sizes of the experimental design.

To test this, we choose a validation set of 25 frames out of the 165 observations. We then train the polynomial chaos model with 16 principal components as input parameters and 30 as output parameters. The principal components are again computed using the full experimental design. For training the PCE model we use training sets of increasing size (from 10 to 140 observations). Finally, the normalized root mean square test error of the surrogate model is computed using the held-out validation set. To reduce the variance induced by the randomness in the splits of the observations into a training and validation set, the test errors are averaged over ten different such splits.

The results are shown in Figure 3.15. For increasing size of the experimental design, i.e. of the set of training data for the polynomial chaos model,

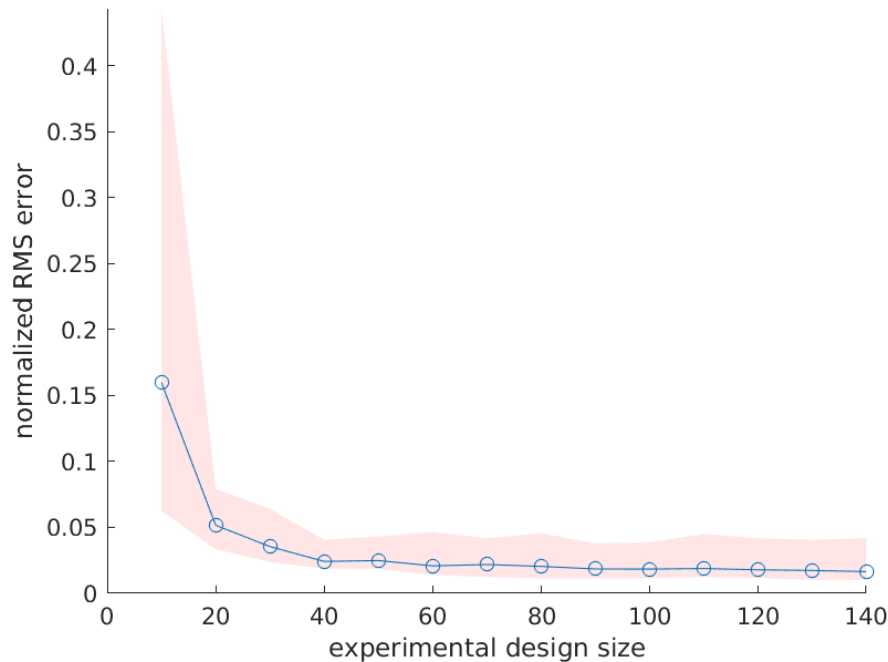


Figure 3.15: Normalized root mean square error plotted against the size of the experimental design for training the polynomial chaos model. The solid line shows the average test error over the different random splits whereas the shaded region depicts the minimum and maximum values of the error.

the error decreases. This is to be expected because such statistical models typically become better when trained on more data. More training data generally makes it easier for the surrogate model to learn the map of the underlying physical model. The reduction of the test error is steep in the beginning but flatter for larger training sets. At some point, there is no large gain in error reduction anymore when using more training data.

3.6 Full layer prediction

In the previous section only the prediction of the temperature evolution for one local model was studied. In this section, the expansion to predicting the temperature of a whole printing layer is shown, given only the initial temperature at the beginning of a layer and the boundary temperatures for every time step. The exact procedure of predicting the temperatures in a whole layer is described in Section 3.2. Here, we show the results of a few numerical experiments.

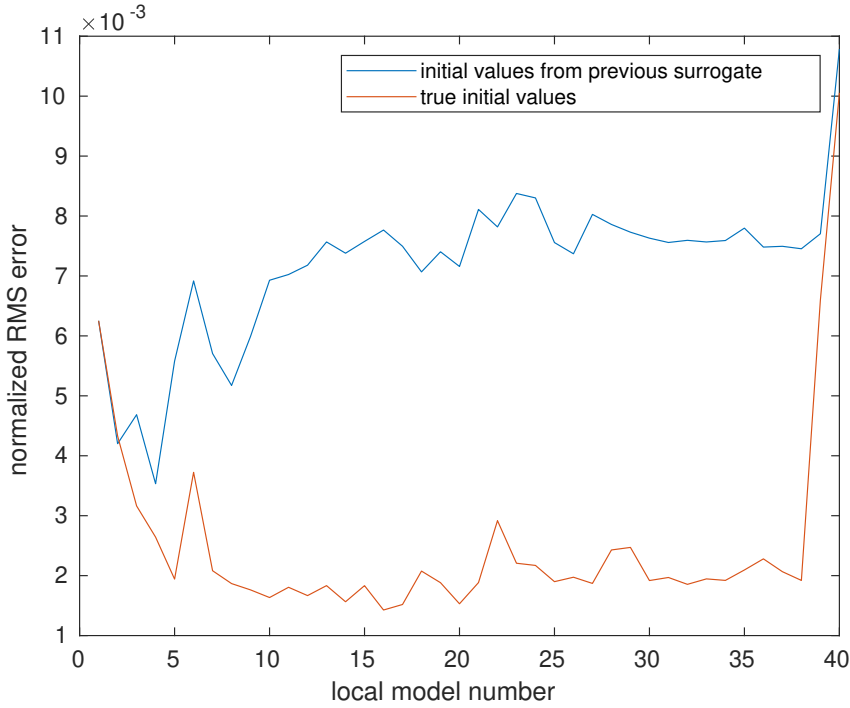


Figure 3.16: Evolution of the normalized prediction error over one layer when using the prediction scheme for a complete layer (blue curve) and when always using the true initial temperatures from the finite element model (red curve).

3.6.1 Accumulation of error within a layer

The most important difference between surrogate model prediction of single frames, as it has been done in the previous section, and predictions of a full printing layer are the input values for the surrogate model. In the previous section the initial temperature values came from the finite element simulation and represented the ground truth. When predicting a full layer, the initial temperatures of the local models can not assumed to be known. Therefore, the temperature results from one local model are used as input values for the next. Only the boundary temperatures, coming from the global model, and the initial temperatures at the beginning of each layer do not contain any error (taking the finite element model as reference).

Using the (erroneous) results of the previous surrogate model as input to the next might lead to accumulation of the prediction error. To see how much this influences the overall prediction error, temperature prediction of a complete layer is studied in two different ways. First, the true initial values from the finite element simulation are utilized as initial data for every local surrogate model. Second, the prediction scheme for layers described above is applied. For every local model the normalized root mean square error

is computed and plotted in Figure 3.16. The data for this experiment come from printing a large solid block with 40 local models in every layer. In this setting the surrogate model generally gives very good results with small errors. When using the true initial values an overall root mean square error of less than half a percent is achieved. For the layer prediction with the results of the previous surrogate model as input to the next, the error is a bit larger but still well below one percent. In the figure we see that for the predictions with true input temperatures (red curve), the root mean square error takes very small values until the rise in the last local model. With the resulting temperatures from the previous model as input (blue curve) instead, the error starts to increase after a few local models and is always a few times larger than the other error values. Most geometries used in this project are much shorter than the one tested here. Thus, the accumulation of the error within one layer does not play a too crucial role and is not investigated much further here.

3.6.2 Convergence for number of training layers

In this section we conduct a numerical experiment to test how much data is needed for training the surrogate model when predicting full layers (compared to predictions of single frames in the previous section). Training data from the FEM model can only be acquired for a whole layer (or a part of it starting from the beginning of the layer) as outlined before. Thus, in this experiment we take training data from whole layers, similarly as it would be given in a real application. We want to find out, how many layers of training data are required to get accurate temperature predictions. All the layers are split in a set of training layers and a set of test layers. Then, the polynomial chaos expansion model is trained on the set of training layers and the prediction error (normalized root mean square error) is computed on the set of test layers. This prediction error is computed for different numbers of training layers. To remove the dependence on the random split into test and training layers, the prediction errors are averaged over twenty different such splits.

The split into training and test sets used here is only relevant for training the polynomial chaos expansion. The principal components are computed with all the available data. Building a good basis of principal components is treated in Section 3.4 instead.

The described experiment has been run for different geometries. First, we look at the simple geometry of a fully printed block (Figure 3.7). In this setting we use 16 principal components for the input to the model and 30 principal components for the model result. These parameters are set according to the convergence tests in Section 3.5.1, refer to Figures 3.8 and 3.12. The chosen geometry has a height of 33 layers and we take between 2 and 30 layers to train the model. The resulting normalized prediction errors are shown

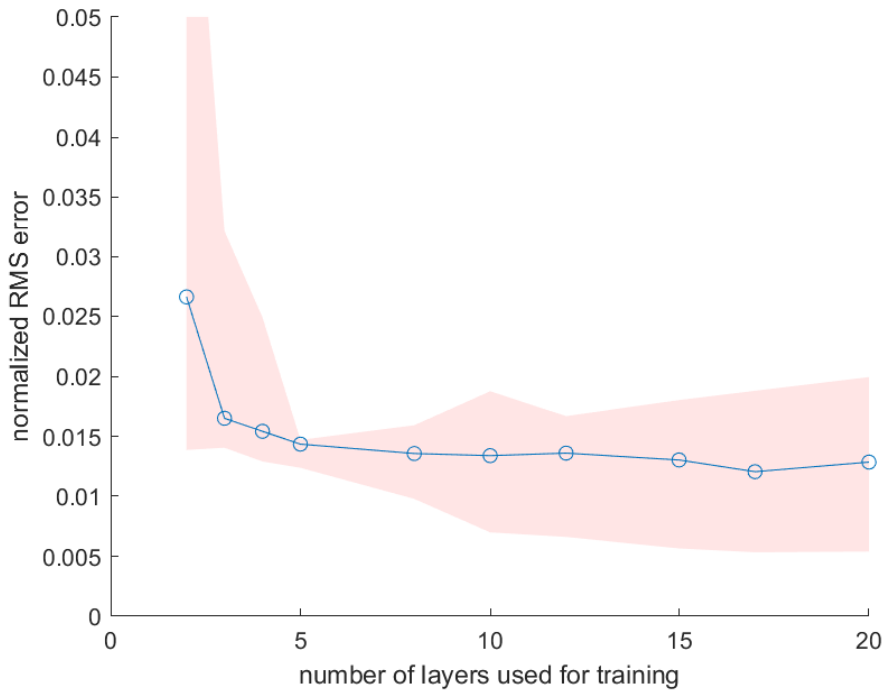


Figure 3.17: Normalized root mean square error at the test layers plotted against the number of training layers for the polynomial chaos expansion model in the setting of a filled block (Figure 3.7). The solid line shows the average test error over the different random splits in training and test data. The shaded region depicts the minimum and maximum values of the error.

in Figure 3.17. In that figure, the blue curve shows the average value of the prediction error over the twenty different random splits in test and training sets. To see the variability of the prediction error also the minimum and maximum prediction errors are indicated with the red shaded area. Overall, the error decreases with an increasing number of training layers as expected. However, there seems to be a lot variability due to the random splits. Therefore, it is important to choose the training layers carefully. In Section 3.6.3 this matter will be investigated more. In this experiment, random splits are favored in order to not be biased by the choice of a certain strategy of selecting the training layers. Concluding, in this setting training data of two layers is enough to get a relatively small prediction error around (or even slightly below) two percent.

The numerical experiment is repeated for the geometry depicted in Figure 3.9 with two filled blocks and an empty region in between. As in the previous section we apply two different models, one in the region where the laser is activated and one in the region where the laser is not active. Here we use 20 principal components for the input space and 30 for the result space (in

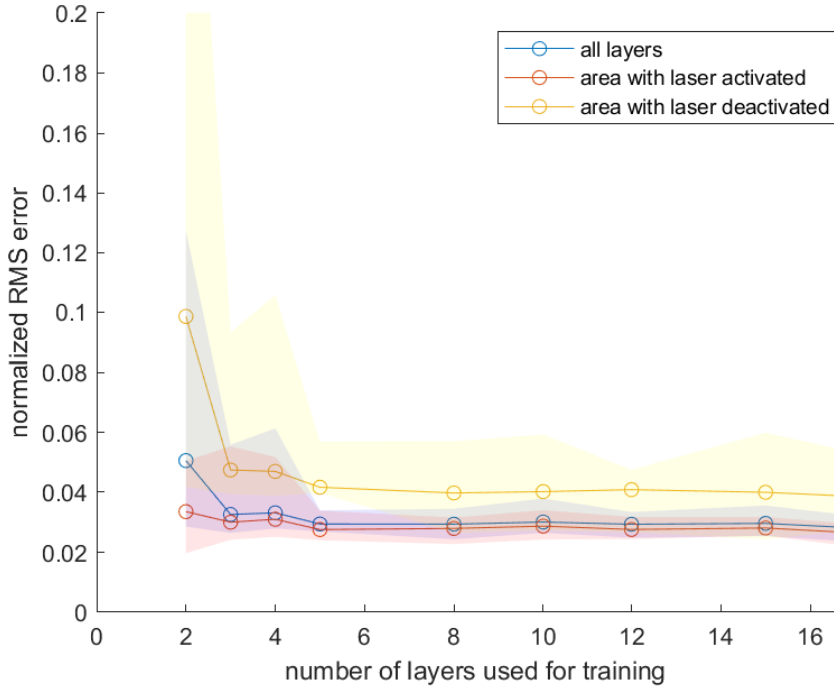


Figure 3.18: Normalized root mean square error at the test layers plotted against the number of training layers for the polynomial chaos expansion. For this test the geometrical setup with two separated blocks (Figure 3.9) is used and two different models are trained, for the region with activated and deactivated laser, respectively. The solid lines show the average test error over the different random splits in training and test data. The shaded regions depict the minimum and maximum values of the error.

accordance with the convergence test in the previous section). This geometry consists of 33 layers like the solid block before. In Figure 3.18 the average normalized prediction errors for the different number of training layers are shown with a solid line as before. The errors are separated for the region with active and inactive laser. Also the minimum and maximum error for every number of training layers are indicated with the shaded regions in the three different colors. Again, we see a large variability in the prediction error with the different splits of the layers, especially when using only few training layers. As expected, the test error is high for very few training layers but already for more than five training layers it drops to about three percent and a plateau is reached. Similar to when predicting only one frame, the errors in the region without laser are always a few percent higher than in the region with laser.

In tests with more geometries we found that the surrogate model has problems predicting the temperatures specially at the interfaces between solid and void regions, i.e. between regions being printed and regions remaining

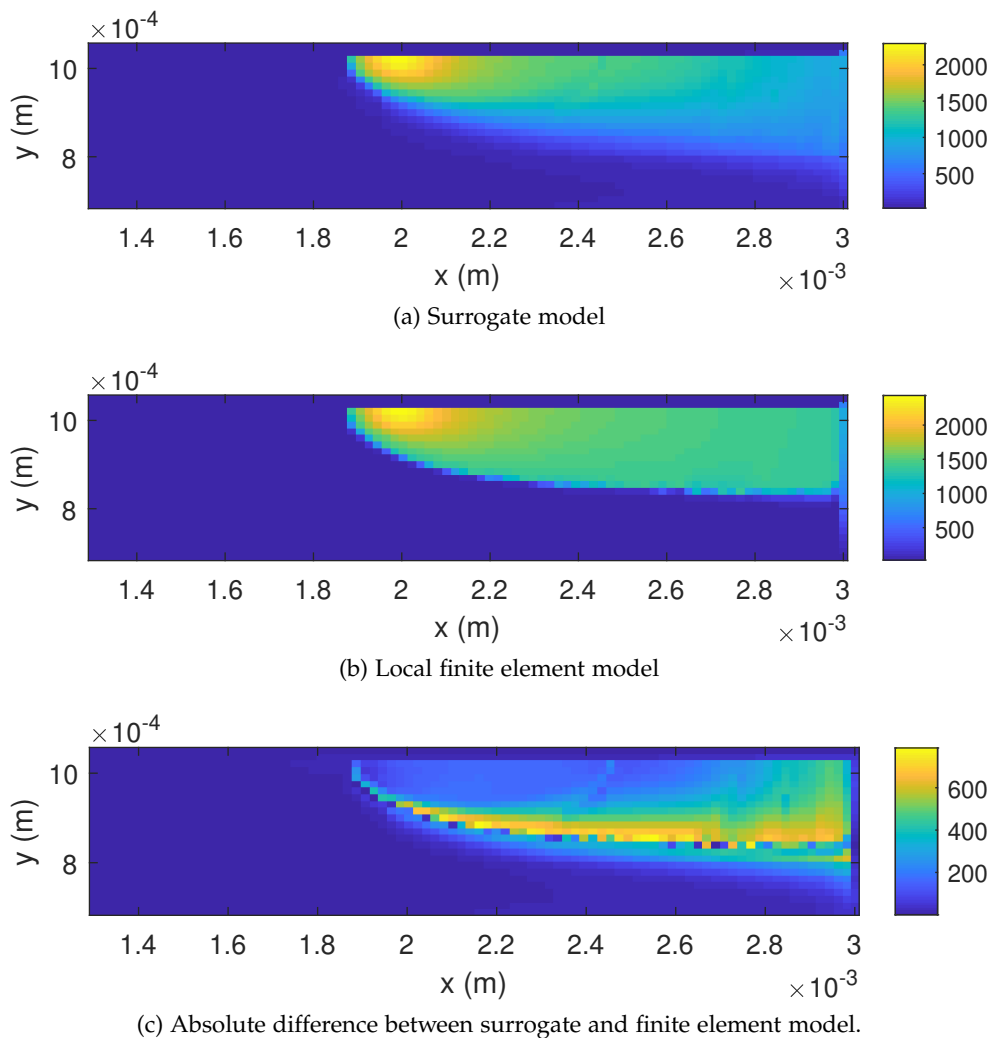


Figure 3.19: Comparison of the temperature profile of a local and a surrogate model two layers above an overhang structure.

empty. Already in the experiments before this has become visible. In the geometry of the solid block, the prediction error was smaller than in the geometry with two such blocks separated by an empty space. The largest part of the error is made in the interface region, even if the overall error is still in an acceptable range.

The problem becomes much more pronounced when looking at horizontal instead of vertical interfaces, as they exist in geometries including an overhang. Such overhangs are not unusual in additive manufacturing, so it is important to study such situations more carefully. An example of such a geometry is shown in Figure 4.5. It is like the geometry with the two columns used before with a horizontal bar added on top. In Figure 3.19 the temper-

ature profiles of the surrogate and the finite element model of a local simulation in the second layer above the overhang are compared. The surrogate model has been trained with two different models for laser activated/deactivated like before. The temperature conduction near the overhang (and generally near interfaces) is different from other regions because the powder (that contains a lot of air) has a very low heat conductivity and almost acts as an insulator. Thus, only a small part of the powder region below the overhang will be heated up by the laser and the temperature drops very fast with increasing distance from the laser.

In the temperature profile of the finite element model, the sharp boundary at the interface between high and low temperature can be seen well. The surrogate model temperature profile, in contrast, shows a rather smooth decay of the temperature similar to regions where there is printed material beneath. Thus, the difference between surrogate and finite element model here is quite large. The temperature prediction of the surrogate model is so wrong because the model does not "know" about the printing geometry. In the view of the surrogate model, the local models only differ by laser activation. All cases with activated laser are modeled in the same way (of course depending on the input parameters as well).

In tests with the surrogate model predicting only a single time step (like in the previous section), the issues with the geometry were not so severe. The initial temperatures from the finite element model already contain information about the geometry of the local model and about the expected form of the temperature distribution. When predicting a full layer this information is only available in the first step of each layer. If the geometry seen in the local models changes within one layer (as it is the case in the setup in Figure 4.5 in the layers above the overhang) the surrogate model does not notice the change.

To test the behavior of the surrogate model in overhang areas more thoroughly, it is applied to a printing geometry with a large overhang. To incorporate information about the geometry of the local model into the surrogate different strategies are tested:

- the input data of the surrogate model is extended with a number telling the location of the interface between printed material and powder within the local model.
- a separate surrogate model (principal component basis and polynomial chaos model) is built for the first few printing layers just above the overhang.
- a surrogate model is built to "correct" the predictions in the overhang region. This model should predict the difference between the normal surrogate model and the true temperature values in the overhang re-

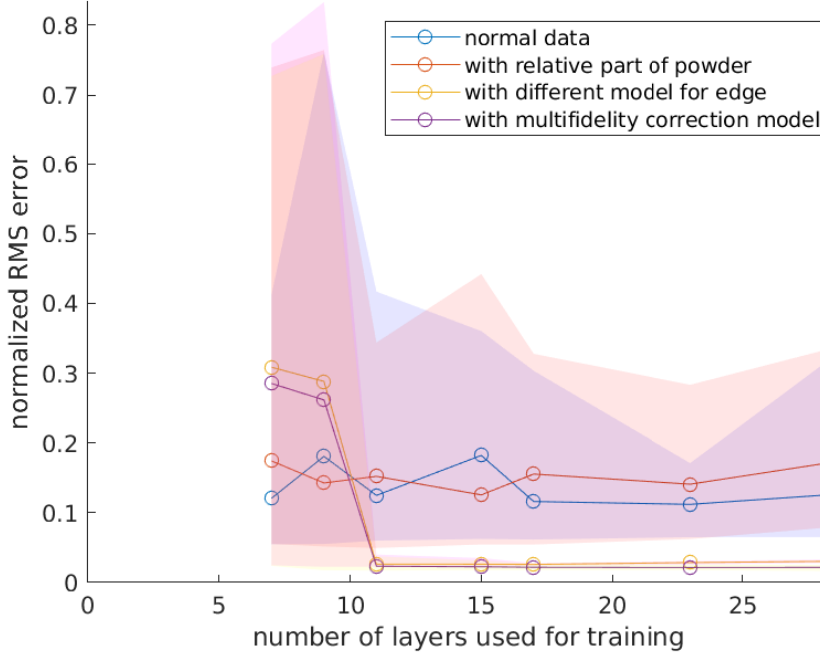


Figure 3.20: Normalized root mean square error at the test layers plotted against the number of training layers for the polynomial chaos expansion. For this test we take a geometrical setup with a horizontal bar. The solid lines show the average test error over the different random splits in training and test data. The shaded regions depict the minimum and maximum values of the error.

gion. These predictions are then added to the normal surrogate model (this can be seen as a so called *multi-fidelity* approach).

The first strategy aims at giving the surrogate model additional input information in the hope that it “learns” to act differently when there is an overhang (or generally an interface). The second and third strategy are chosen because the behavior of the model in the overhang region is substantially different from the behavior in other regions (similar to laser active/inactive) and a totally different model trained explicitly in this region might lead to better results than enhancing the already existing model.

If there is a systematic error in the predictions of the normal surrogate model predicting the error instead of the complete temperature data might be less complex (especially for the PCA as the error tends to be smooth). This is the idea behind the third strategy.

To see the performance of the three different approaches, the convergence tests for the number of training layers conducted with the other geometries before are repeated for the overhang data. The results of this test is shown in Figure 3.20. The blue solid line shows the prediction error for the normal sur-

rogate model used before as a reference. The red solid line shows the results of the first approach, the yellow line the second with a separate model for the interface and the violet line the third with the correction model. Again, the minimum and maximum test errors for different random splits into test and training data are indicated with the shaded areas.

The prediction errors for the first two strategies without additional model are similar. They only decrease to about 10% which is quite a large error. With a separate model for the interface or a correction model, much smaller prediction errors can be reached (around 3% for both).

Applying an additional model brings two advantages. First, it improves the predictions in the (difficult) overhang region because it was explicitly and only trained on data of that area. Second, the prediction error in the other regions are also reduced because the predictions there are not influenced by the data of the overhang region. In that regard, the principal component basis plays an important role. Because of the sharp, almost discontinuous temperature profile in the overhang region, the variance explained by the principal components decays much slower than for the rest of the data. This means that many more principal components are needed to capture the most important characteristics of the data and to be able to reconstruct it.

3.6.3 Selection of training layers

As already written before, the choice of the layers to train the surrogate model can have a large impact on the prediction accuracy. In this section, different strategies for the selection of training layers are compared. For the comparison, data of the square geometry (3.7) are used. In this geometry the temperature evolution is very similar over all the layers and there are no complicated geometrical features like in the example with a corner and an overhang. Still, the variation of the prediction error with respect to the choice of the training layers is large as seen in Figure 3.17 where the training layers were randomly chosen.

Instead of randomly selected training layers, in this experiment the training layers are chosen from all the layers to be approximately equidistantly distributed. From preliminary tests, we have seen that the bottom layer might have slightly different dynamics compared to the other layers. So it is not clear whether to include this layer for training or not. For the convergence test, four slightly different strategies are applied:

- equidistantly distributed training layers without the top and the bottom layer;
- equidistantly distributed training layers without the top but including the bottom layer;

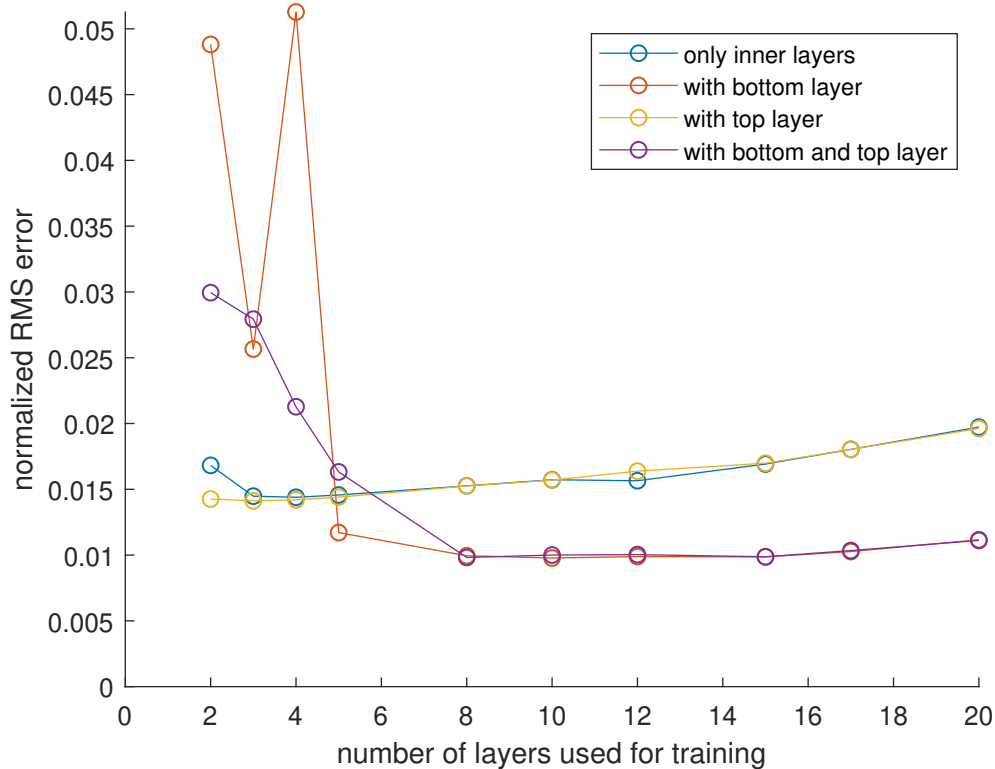


Figure 3.21: Normalized root mean square error of the surrogate model plotted against the number of training layers. The training layers are selected with different (non-random) strategies. For this experiment, data of the square geometry in Figure 3.7 is used.

- equidistantly distributed training layers including the top but without the bottom layer;
- equidistantly distributed training layers including the top and the bottom layer.

The normalized prediction error of the surrogate model trained on layers chosen with these different strategies are shown in Figure 3.21. For very few training layers it seems better to use only layers from the middle including the highest layer. The models trained on layers including the one on the bottom have a significantly higher error. For larger number of training layers the opposite happens. The prediction errors of the models that do not use the lowest layer are almost twice as high as when including the bottom layer in the training set.

This slightly paradox result might be explained as follows. Since the temperature in the lowest layer shows a different behavior than in the other layers, training a surrogate model with data largely coming from the lowest layer yields less accurate results. The lowest layer does not have very much pre-

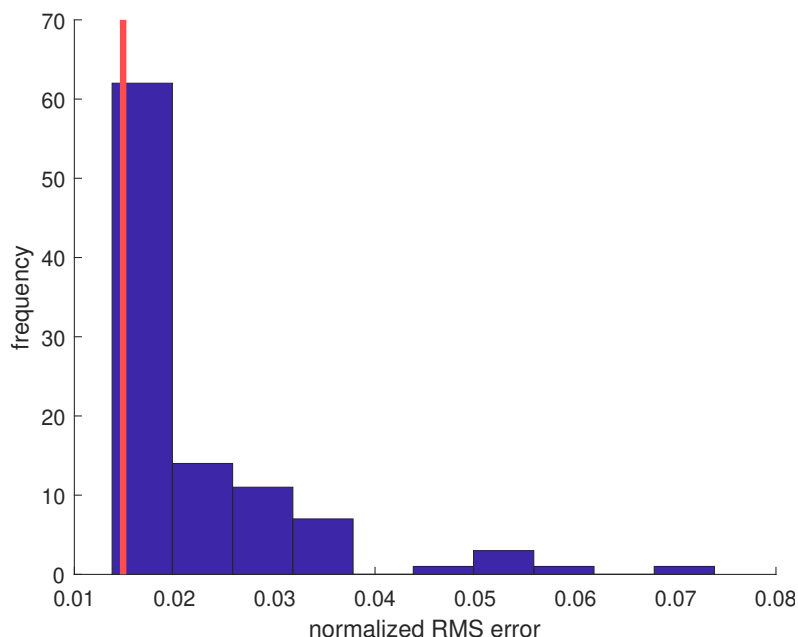


Figure 3.22: Histogram of the normalized prediction error of surrogate models trained on different random selections of two training layers. The vertical red line indicates the prediction error of a model trained on two layers with equal distance to each other and to the bottom and the top of the geometry.

dictive power on the other layers. Contrarily, when the lowest layer is not used for training it is included in the test set. The predictions on the lowest layer generally have a larger error than on the other layers. Therefore, the test error is strongly influenced by the prediction error on the lowest layer, especially for a large number of training layers and small number of test layers. That is why the test error of the strategies that do not include the lowest layer increases for a growing training set.

To show that choosing the training layers equidistantly is a reasonable choice, we compare the prediction error of a surrogate model trained on two layers with equal distance to each other and to the first and the last, respectively, with models trained on two randomly selected layers. The random selection is repeated 100 times and a histogram of the resulting normalized prediction errors is shown in Figure 3.22. The vertical red line indicates the prediction error of the model trained on equidistant layers. That selection of training layers achieved almost the smallest test error of all the random choices.

Concluding, it is not easy to determine a good choice of a training set beforehand. In the example considered here, few equidistantly distributed training layers from the inner of the geometry yield good results. This might be generally true for simple geometries. For geometries with more complicated features the layers can not be chosen just equidistantly but the location of

the features must be taken into account as well and selecting the training layers becomes more difficult.

3.7 Marginal distributions

One important choice in the modelling process has not been discussed yet. The polynomial chaos expansion relies on a basis of polynomials that are orthogonal with respect to a weighting function. This weighting function should be equal to the probability density function of the input data when seen as a random variable (remember that the PCE was designed to model uncertainties in systems with uncertain input parameters). When the distribution of the different model input parameters is known, it is straightforward to choose the weighting function accordingly. The polynomial chaos expansion profits from a good choice of the weighting function, especially if it is not only applied for predicting new observations but also for uncertainty quantification, like computing statistical moments or performing a sensitivity analysis. In such cases the conclusions drawn from a polynomial chaos model with wrong marginal distributions can be very misleading.

In our case it is not straightforward to determine the marginal distribution of the model parameters. Even for the raw temperature values we could only tell a minimum value (25 degrees) and maybe a rough estimate of the maximum value based on our modelling experience. For the principal components of the temperature values, however, we have no good information about the distribution beforehand. The easiest thing is to assume a uniform distribution based on the bounds we see from the data at hand. This involves finding the minimum and the maximum value of each principal component based on the given data and enlarging this range by a threshold of, say, 10 percent to be on the safe side. This strategy for determining the marginal distributions has been applied throughout this project. As the results have shown, it works fairly well as long as no principal component values outside of this range are encountered. That is a further reason, why it is important to have a good principal component basis that includes all relevant cases.

Figure 3.23 shows the values of the first principal component of all the observations of the initial temperatures in the square-like geometry. With the strategy of choosing the marginal distributions outlined above, in this case a uniform distribution between about -2 and 1 is found. However, this is clearly not true here. The values of the first principal components are aligned in bands representing certain local models in every layer.

A second thing to consider is if the model parameters are independent of each other or if they are coupled. The principal components are not independent from each other because they all depend on the same data set. Because this dependence is difficult to describe rigorously, in this project the princi-

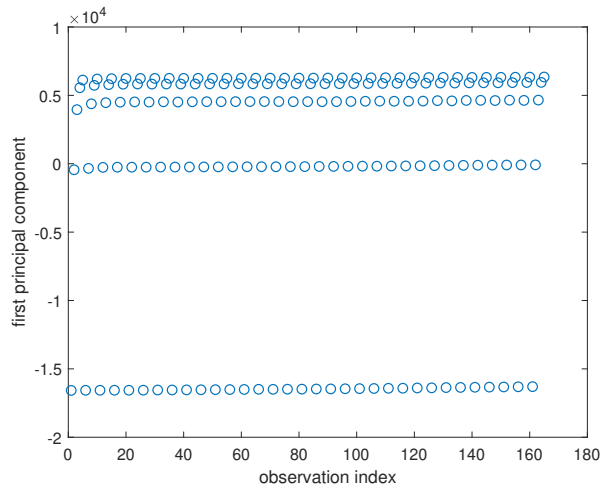


Figure 3.23: First principal component values for all the observations of the initial temperature data in the square geometry.

pal components are modeled as independent. Dependent marginals could be achieved through the choice of so called copula functions. Copulas are multivariate distributions over the unit cube with uniform marginals but certain dependencies between the different dimensions. With the right choice of copulas, univariate marginal distribution can be coupled to form certain joint distributions. The copulas are responsible for the *dependence structure*. More (mathematical) details on this topic can be found in the UQLab manual about specifying the uncertain inputs [9].

To see how the choice of assuming independent and uniformly distributed model parameters influences the prediction accuracy, we conduct a small test. The polynomial chaos expansion model is trained with different choices of marginal distributions and copula types on two data sets. The first data set comes from the simple square like geometry, the second from the geometry with two columns. We tested uniform marginals, like used in this project otherwise, marginals inferred by the data through statistical inference and non-parametric estimates of the marginals through kernel density estimation. More details on how these procedures are implemented in UQLab can be found in [9] and [19]. As copula types independent copulas and copulas inferred from the data through statistical inference are applied. The test errors achieved with the different marginal and copula combinations are presented in Table 3.1.

We see that the test errors for uniform marginals and independent copula are among the smallest test errors for both data sets. The test errors for the more realistic, dependent copulas (inferred from the data) are almost twice as high in all cases. Also more realistic marginals do not give any advantage, in most cases the test error is rather increased. Similar behavior

marginal	copula type	test error	
		square geometry	geometry Fig. 4.5
uniform	independent	0.0068	0.0356
uniform	inferred	0.0117	0.0373
inferred	independent	-	0.0600
inferred	inferred	-	0.0670
KS	independent	0.0095	0.0347
KS	inferred	0.0107	0.0564

Table 3.1: Test errors for different combinations of marginal distribution and copula type for the square geometry (Figure 3.7) and the geometry with two columns (Figure 4.5). Marginal types are uniform, inferred from the data by statistical methods and kernel smoothing (KS). The dash indicates that it is not possible to fit a model.

has also been observed in [20], where the authors also apply polynomial chaos expansion models to data sets for which the distributions are not known. The authors in that paper suspect that the higher errors stem from the very nonlinear transformation that must be applied to the orthogonal polynomials when using certain type of copulas.

3.8 Prediction of peak temperatures

To determine the chance of significant metal evaporation and the build quality of printed parts, knowledge of the peak temperatures within each layer is important. In Chapter 4, peak temperatures are calculated based on temperature predictions of the surrogate model and are compared with peak temperatures from the finite element simulations. In this section, we try to build a surrogate model that predicts only the peak temperatures instead of the temperatures at every node of the local model. In that surrogate model no dimensional reduction for the output data is needed as the output consists of only one value.

As input data for this surrogate model the local temperature data can not be used like before because it is only available in the beginning of each layer. In the other surrogate model, the input data is interpolated from the results of the previous model. This is not possible here as only the peak temperature is predicted. Thus, temperature data from the global model are used as input to the surrogate model here. Besides the input data, the new surrogate model works very similarly to the one before. The polynomial chaos expansion is applied and the dimension of the input data is reduced with principal component analysis. To see if this surrogate model provides any advantage compared to the one studied before, we compare the root mean square errors of predictions of the peak temperatures of the two different surrogate models. The results of this comparison for different numbers of

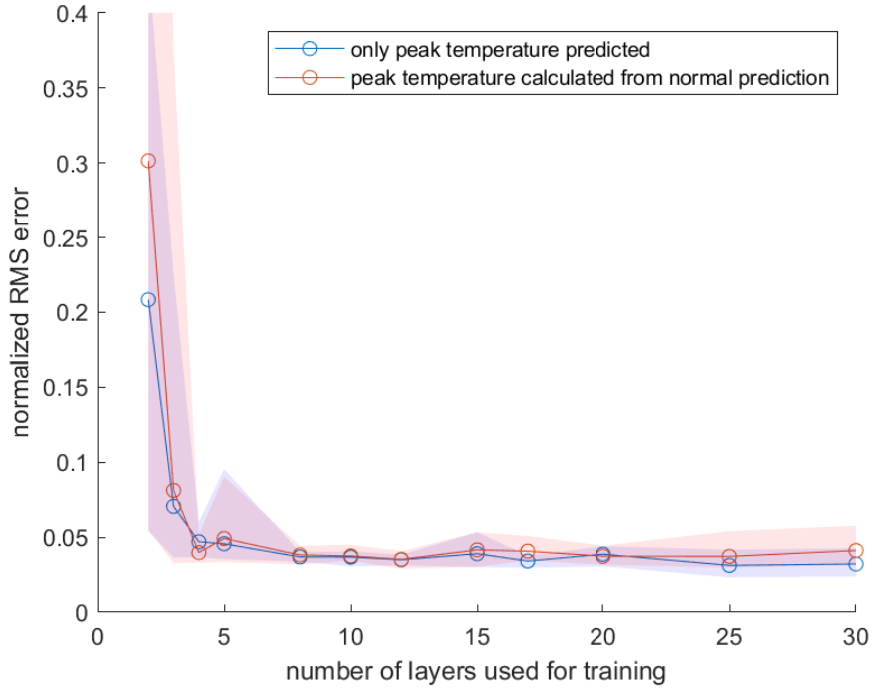


Figure 3.24: Comparison of the prediction accuracy between predicting only peak temperatures and computing the peak temperatures from the predictions of the complete temperature fields.

training layers are shown in Figure 3.24. The temperature data for this experiment are taken from the geometry with two columns (Figure 4.5).

We see that both surrogate models perform very similarly in terms of normalized root mean square errors. Predicting only the peak temperature does not give any higher accuracy while giving much less information at the same time. This is probably because the principal component analysis finds the peak temperature region as an important feature in the data and thus the peak temperature is represented well within the principal component basis and also in the reduced data. Therefore, the peak temperature is found with high enough accuracy also when predicting the whole temperature field.

Chapter 4

Results

In this chapter the results of a few simulation studies in different printing scenarios are presented.

4.1 Printing a solid block

First, the performance of the surrogate model in the case of printing a solid block of size $1 \times 0.99\text{mm}^2$ is shown. For the geometrical details, see Figure 3.7. The part is printed in 33 layers. Each layer contains 5 local models in the multi-scale simulation.

To train the surrogate model, two layers in the inside of the domain are selected. The surrogate model is then built with the principal component analysis and the polynomial chaos expansion model like it is described in Section 3.2. We take the first 16 principal components for reduction of the input space of the model and the first 30 principal components for the output space. Having trained the surrogate model, it is applied to find the temperature evolution in the layers that were not used to train the model.

global simulation	45 min
local simulations	155 min
total time multiscale simulation	200 min

Table 4.1: Time needed for running the multiscale simulation for the solid block geometry.

global simulation	45 min
two layers of local simulations	9 min
training and evaluation of surrogate model	1 min
total time surrogate model	55 min

Table 4.2: Time needed for training and running the surrogate model for the solid block geometry.

4. RESULTS

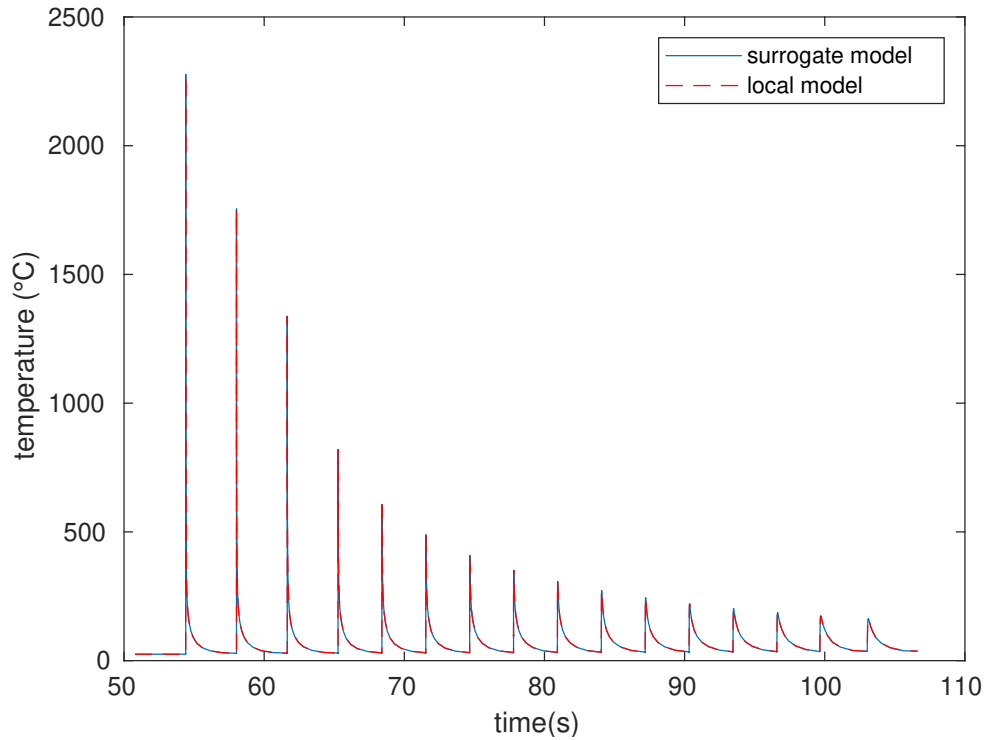


Figure 4.1: Comparison of the temperature evolution provided by the surrogate model (solid blue line) and the (exact) local model (dashed red line) at the point $(x, y) = (0.5\text{mm}, 0.5\text{mm})$.

An important goal of the construction of the surrogate model was to reduce the total simulation time compared to the multi-scale finite element simulation. For training and evaluation of the surrogate model on the remaining layers about 30 seconds are needed. To get a fair comparison, the time needed to compute the training data must be added. That is the time the multi scale simulation needs to simulate the two layers on which the surrogate model is trained and the time to run the global simulation. The global simulation is required for the initial temperatures in the beginning of every layer and for the boundary conditions of all the local models. These timings are presented in the Tables 4.1 and 4.2. The surrogate model needs about one fourth of the time of the multi scale model. This is a significant reduction but not as much as wanted. The largest part of the simulation time of the surrogate model falls on the global simulation. Depending on where exactly the two layers are positioned in the component, it may not be needed to run the whole global simulation and the time needed by the surrogate model could be reduced further.

In Figure 4.1 you can see the temperature evolution over the whole simulation time at the point $(x, y) = (0.5\text{mm}, 0.5\text{mm})$. The results of the multi-scale simulation (dashed red line) and the values of the surrogate model (solid

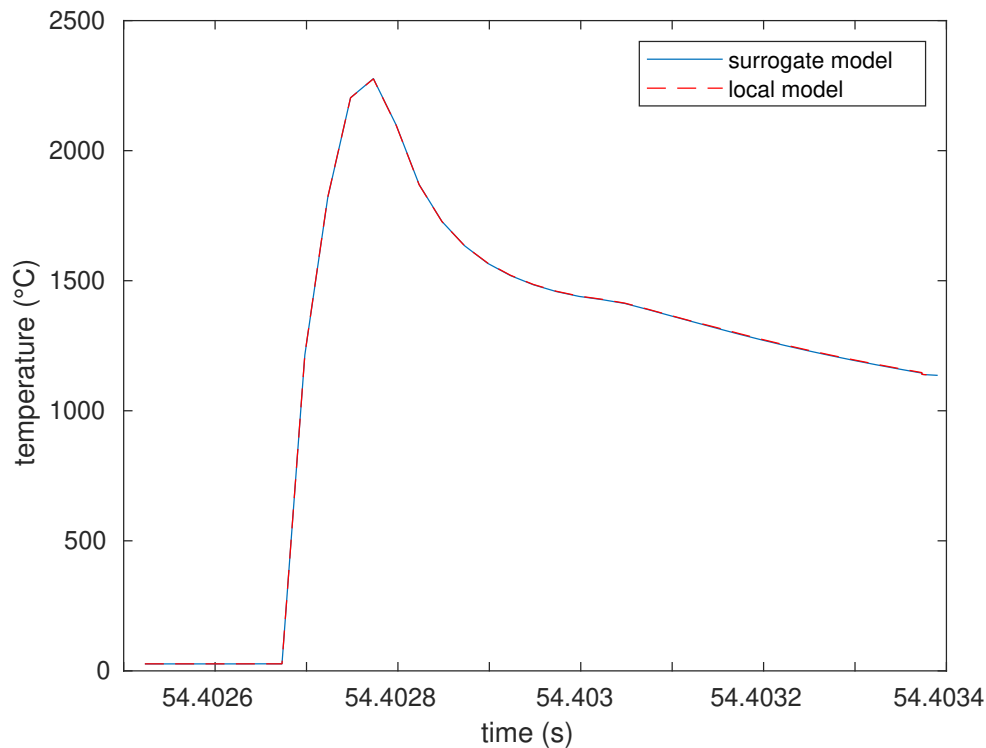


Figure 4.2: Comparison of the temperature evolution provided by the surrogate model (solid blue line) and the (exact) local model (dashed red line) at the point $(x,y) = (0.5\text{mm},0.5\text{mm})$ for the first pass of the laser over this point.

blue line) are compared there. The temperature data from the surrogate model is merged together with the data from the global model (including the cooling phase at the end of each layer). This shows that the surrogate model fits nicely into the multi-scale simulation framework. Every rise in temperature characterises a new pass of the laser and, thus, a new layer. The peaks get weaker with time because the observation point is covered by more and more printed material through which the heat must be conducted.

Little to no difference can be seen between the true temperature curve of the local model and the temperature curve of the surrogate model. Also when zooming in to the first peak (when the laser first melts the powder around the observation point) no difference between the two curves is visible, see Figure 4.2.

Next, in Figure 4.3 the nodal temperatures of one local model are compared with the corresponding surrogate model temperatures. The location of this local model is in the uppermost printing layer but in other models the results look very similar. We see that the absolute difference, which is equivalent to the error of the surrogate model, is at most about 15 degrees Celsius. The largest deviations are found right in front and below the melt pool. In the

4. RESULTS

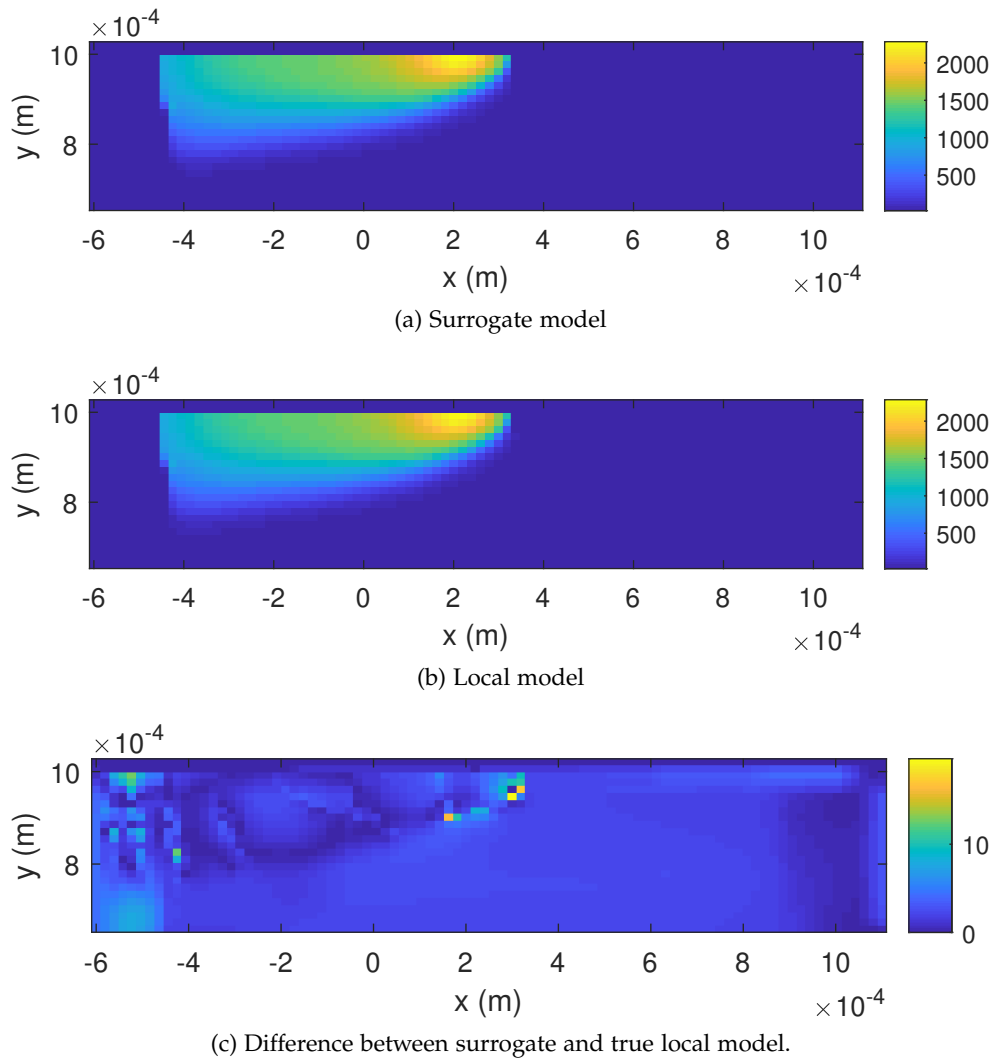
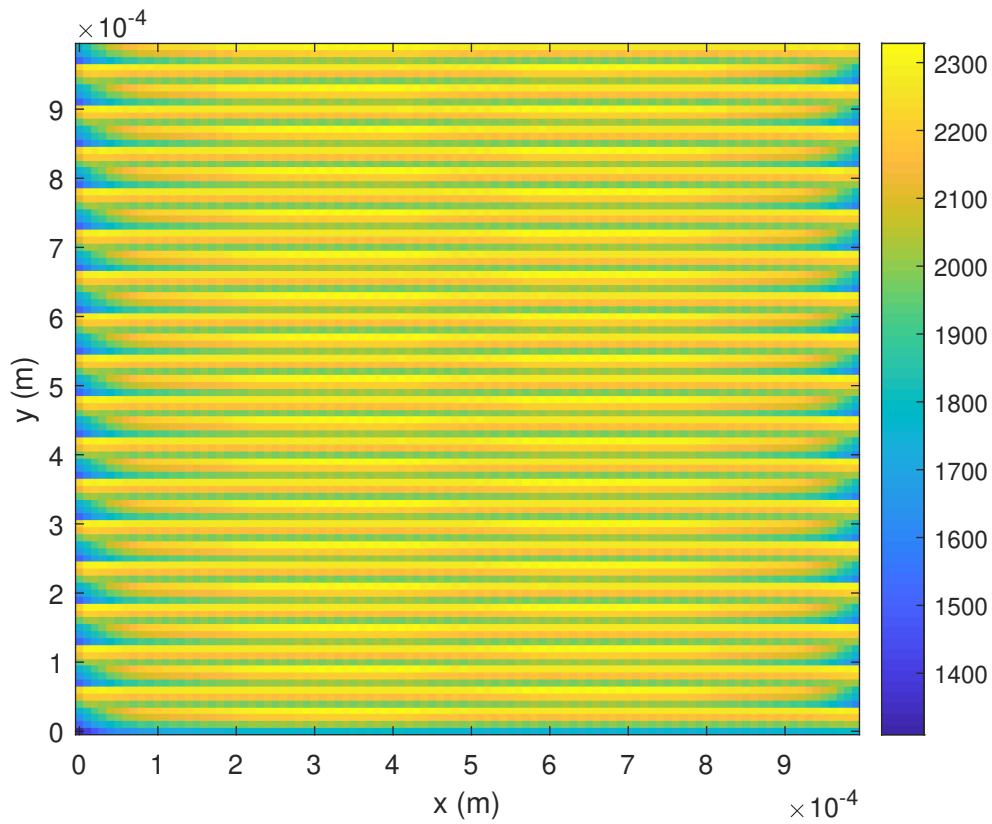
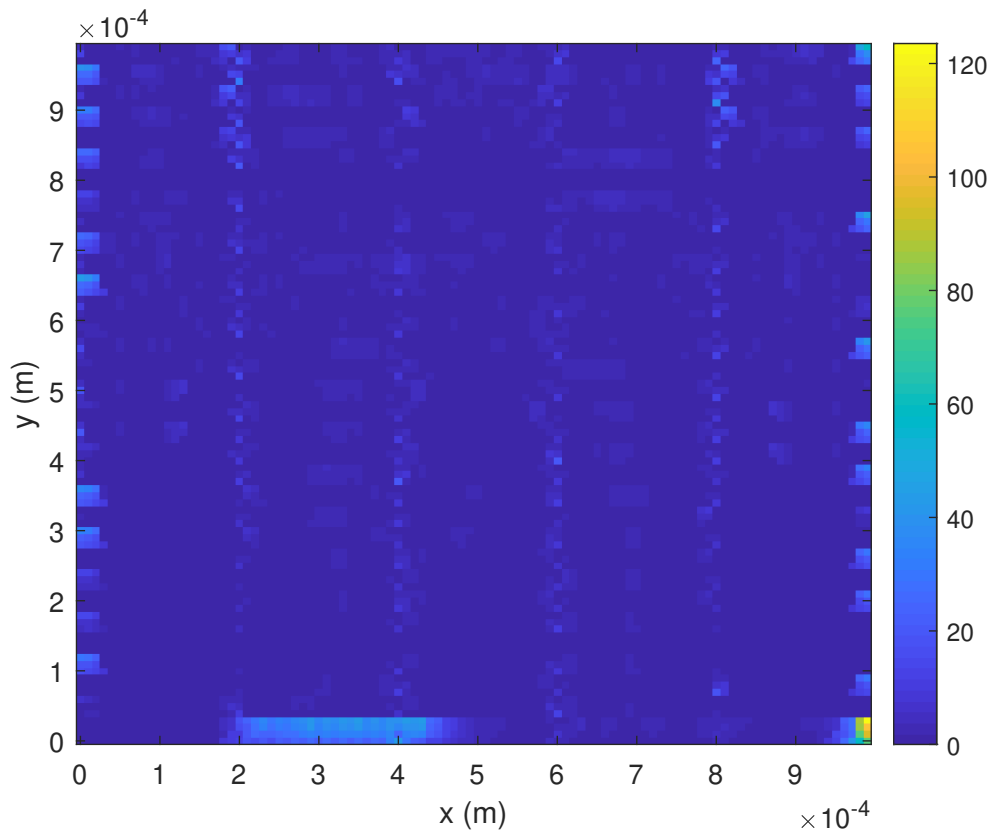


Figure 4.3: Comparison of the temperature profile of a local and a surrogate model in the uppermost printing layer.



(a) Peak temperature computed from the surrogate model



(b) Absolute difference in peak temperature with respect to the true values from the finite element model.

Figure 4.4: Comparison of the peak temperatures in the whole printing part between surrogate and finite element model for the solid block geometry.

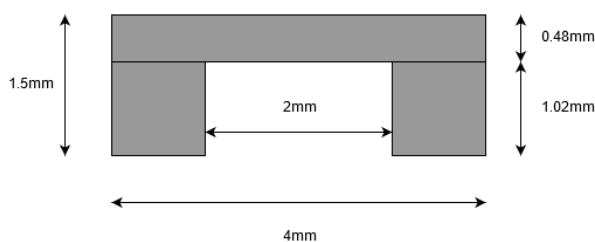


Figure 4.5: Sketch of the two-column structure.

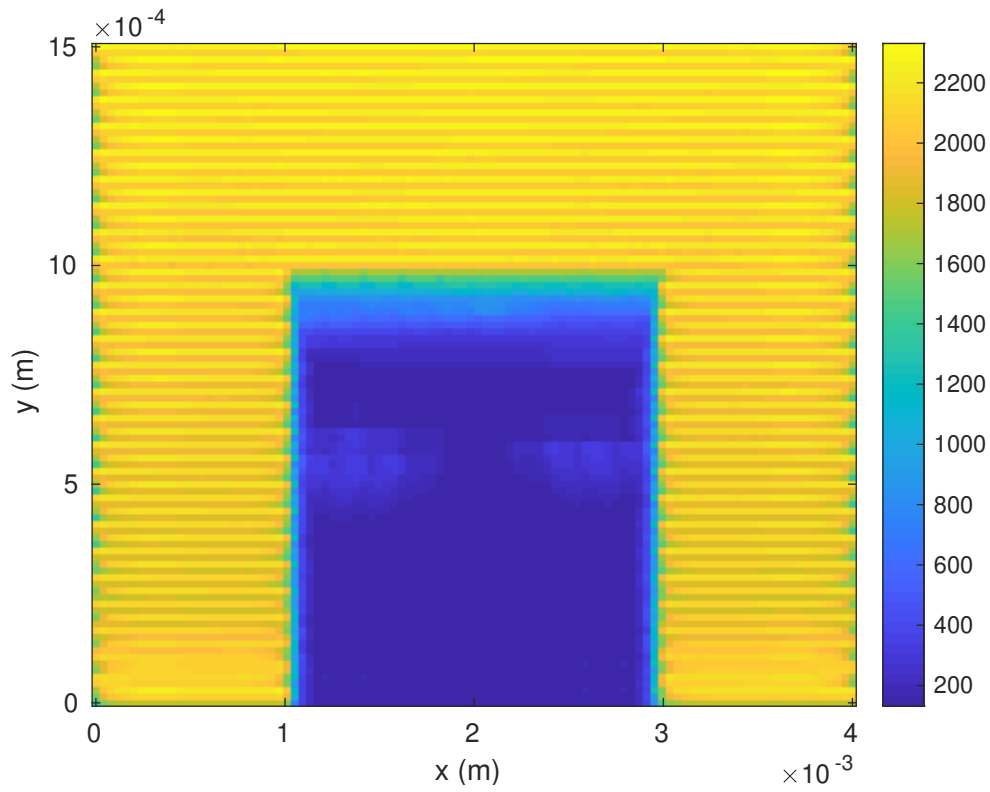
other locations the errors are much smaller.

An important outcome of thermal simulations is the peak temperature in different regions of the printed component. Therefore, we determine the peak temperatures during the complete simulation time for the surrogate model and the finite element model. In Figure 4.4, first the peak temperatures of the surrogate model are depicted and second the absolute differences in the peak temperature between the surrogate model and the finite element model. In the peak temperatures one sees well where the different layers are and where the laser moves. The errors in the peak temperatures are mostly small except for one spot in the lower right corner where the error is as large as 120 degrees Celsius. In the other regions the error is mostly about 20 degrees Celsius which is not large compared to the peak temperatures of more than 2000 degrees Celsius. In other experiments we have seen that the first layer is often the cause of increased errors, which is what we see here as well. What can be seen here also is that the prediction error always increases towards the end of the layer. This is probably due to error accumulation (cf. Section 3.6.1).

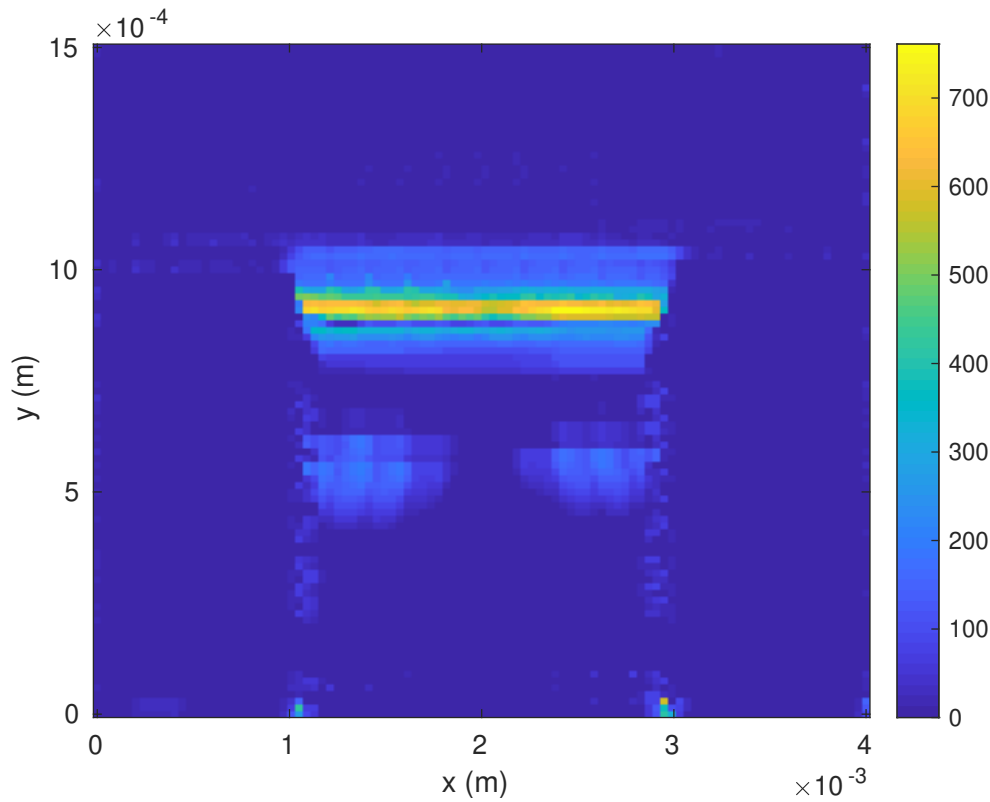
Summarising, for this geometry very good results are obtained from a surrogate model trained only on simulation data from two layers of the multi-scale model. The accuracy of the temperature data computed with the surrogate model is good and the computational resources needed are much smaller than for the complete multi-scale model.

4.2 Printing a two-column-structure

Next, we test the surrogate model on a more complicated geometry. The part that was printed in the last section is simple in the sense, that it had clear straight edges and it is printed everywhere, meaning there are no wholes in the component. On that simple part the surrogate model performed fairly well. In this section we try to simulate the printing of the geometry in Figure 4.5. The part is 4 mm long and 1.5 mm high. This gives 50 layers to be printed and 20 local models in every layer. This geometry is very similar to the structure used in Chapter 3, but it has an additional horizontal bar on



(a) Peak temperature computed from the surrogate model.



(b) Absolute difference in peak temperature with respect to the true values from the finite element model.

Figure 4.6: Comparison of the peak temperatures in the whole printing part between surrogate and finite element model for the geometry in Figure 4.5.

4. RESULTS

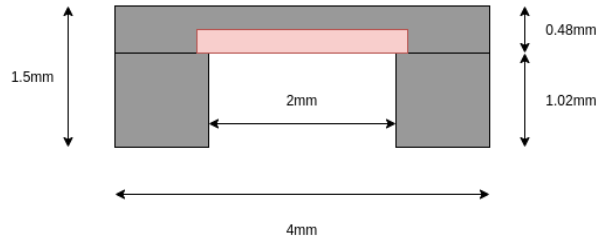


Figure 4.7: Sketch of the two-column structure with the separate overhang model indicated in red.

top that connects the two columns at the left and right. As there is a region that is not printed in the middle, two different surrogate models are trained. One model where the laser is active and where physical printing is happening and one model, where the laser is not active. These two models are trained on two different data sets (the procedure is described and justified in Chapter 3). In this experiment, we train the models on 10 of the totally 50 layers. Then, the computational effort for training the surrogate model is, including the global simulation, a bit more than one fourth of the multi-scale simulation model. Twenty input and thirty output dimensions are used for the polynomial chaos expansion model. The other model parameters are the same as before.

To see the quality of the surrogate model in predicting the peak temperatures, like before, the peak temperatures over the whole printing geometry are plotted and compared to the respective values of the multi-scale simulation. These results are shown in Figure 4.6. In the first plot we see the peak temperatures predicted by the surrogate model. One can see well where the laser was active and where it was not. We also see that below the bar, which connects the two vertical pillars, the heat dissipates into the powder to some extent. This is probably the most difficult part for the surrogate model to predict because it has no information about the geometry and about the physical processes that play a role there. We also see that the errors in the peak temperature are very large just below the horizontal bar.

This aspect has already been addressed in Chapter 3. There, the proposed solution was to use an additional surrogate model for the overhang region. In convergence studies this showed good results in decreasing the prediction errors, especially in the overhang region. Thus, also here we try to use this concept. For the eight layers directly on top of the overhang, the region in the center is predicted with a separate model, which is trained on data only from that region (see Figure 4.7). The other two surrogate models for the rest of the printed region and for the region where the laser is not activated remain the same as in the previous test case. In total nine layers are used to train the surrogate models (compared to ten before). To see

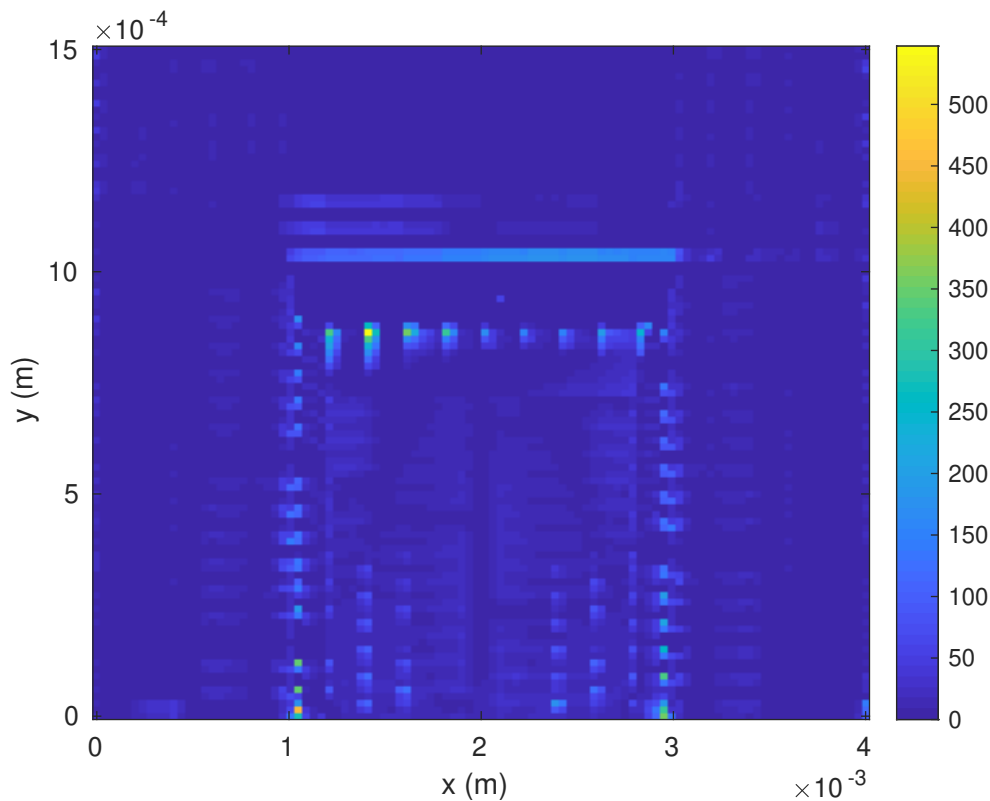


Figure 4.8: Absolute difference in the peak temperatures between the surrogate model with an additional model for the overhang region and the true values from the finite element model printing the same structure as in Figure 4.6.

how well this approach works, the peak temperatures are computed based on the temperature predictions of the three different surrogate models. The absolute differences in peak temperatures between the surrogate and the finite element model are shown in Figure 4.8. Compared to Figure 4.6b, the maximum errors of the predicted peak temperatures are much lower. Especially the region with large errors around the overhang is much smaller in the approach with a separate overhang model.

4.3 Printing a T-structure

As a last test case, we simulate the printing of a geometry that resembles a "T" (see Figure 4.9). Similar to the previous geometry, we need two different models for the regions with and without activated laser. In addition, a separate model for the lower part of the crossbeam of the T is used to achieve a reduction of the error in this critical area. The geometry is 50 layers high and every layer consists of 15 local models. The surrogate model is trained on ten layers. For the polynomial chaos expansion model the input space is

4. RESULTS

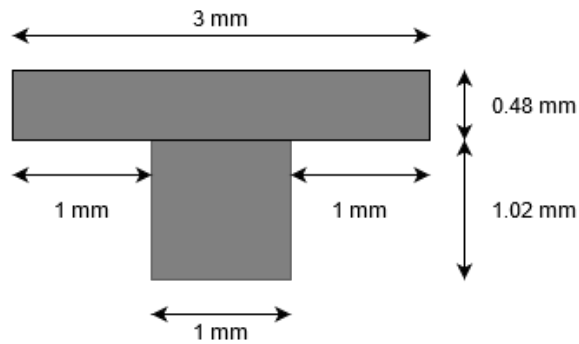
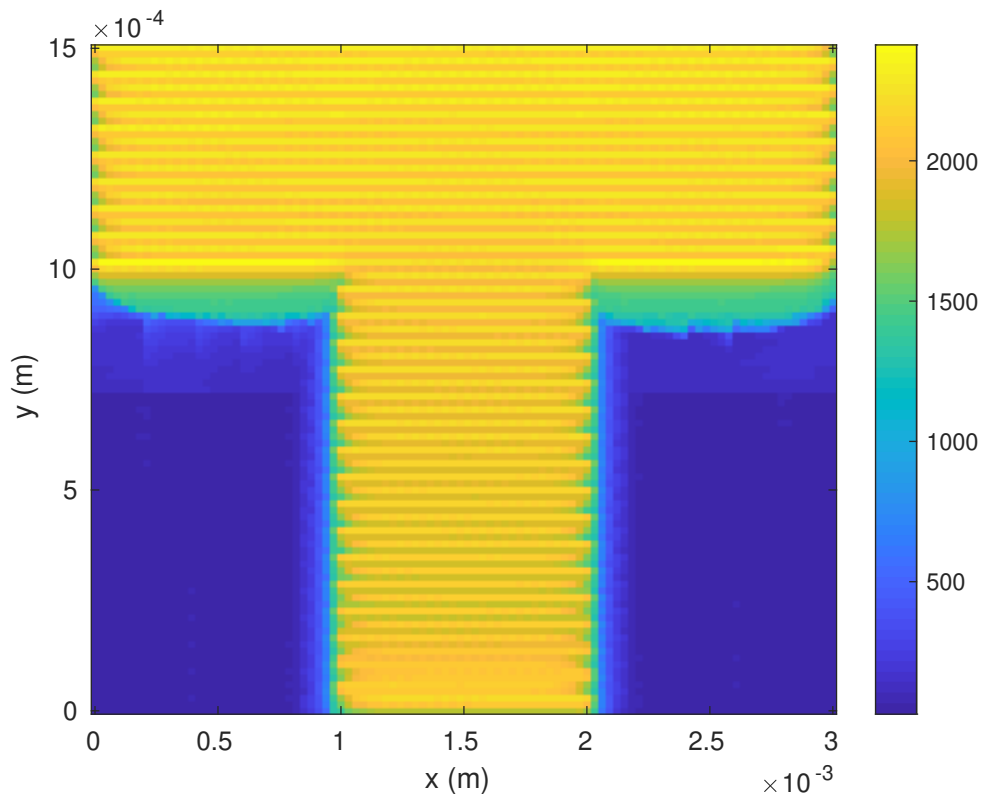


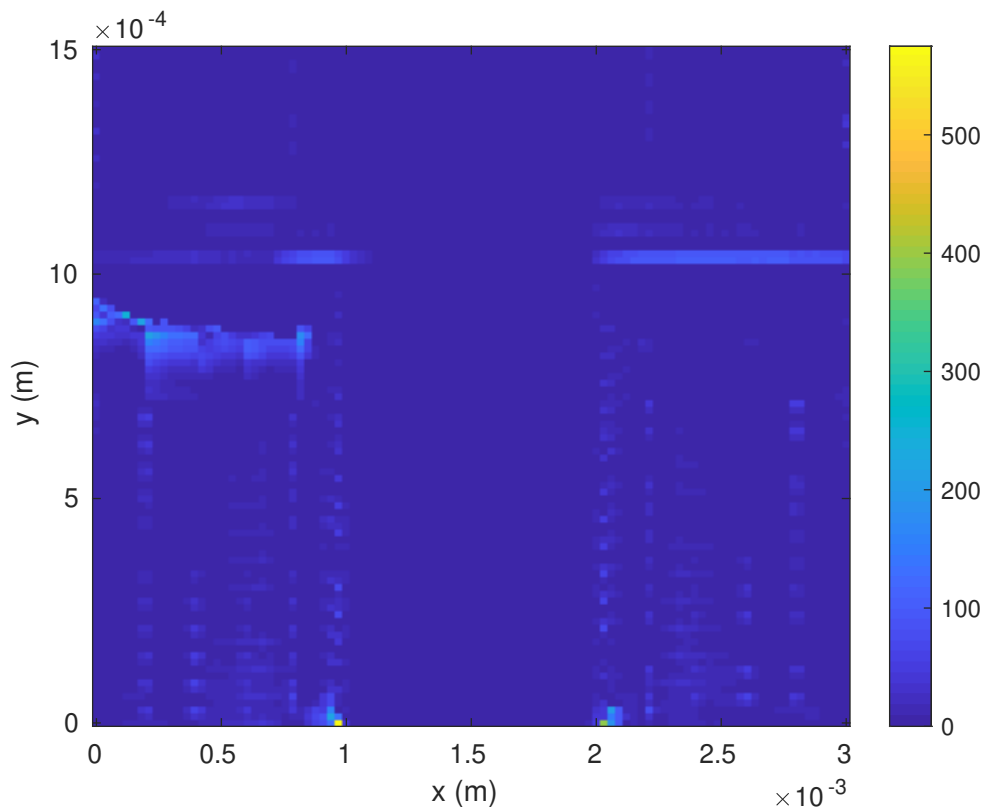
Figure 4.9: Sketch of the T-structure.

reduced to 20 dimensions and the result space to 30.

In Figure 4.10, the peak temperatures of the simulation are shown. As expected, when comparing the peak temperatures of the surrogate model with the ones of the multi-scale finite element simulation we see good agreement. Also in the said region below the crossbeam, the errors are not too large. The largest errors are in two spots at the edge of the structure in the lowest layer.



(a) Peak temperature computed from the surrogate model



(b) Absolute difference in peak temperature with respect to the true values from the finite element model.

Figure 4.10: Comparison of the peak temperatures in the whole printing part between surrogate and finite element model for the geometry in Figure 4.9.

Conclusions and Outlook

The goal of this thesis was to build a surrogate model in a multi-scale simulation framework for selective laser melting in order to reduce the computational resources needed for running the simulation. To this end, we have proposed a combination of dimensional reduction using principal component analysis and surrogate modelling with polynomial chaos expansion. We have also shown a way to apply the proposed surrogate model for predicting the temperature evolution in complete printing layers (where the temperature results of consecutive local model must be combined). Reasonable values for the parameters of the surrogate model have been determined in convergence tests. The surrogate model has then been validated on different printing geometries and it has been found that the proposed approach works well on simple geometries like printing a square. There it showed good results in predicting the temporal evolution of the two dimensional temperature field under the influence of the moving laser beam. The computational costs of training and evaluating the surrogate model are very low. Most of the time needed to construct a surrogate model for a new printing setup is spent in acquiring data to train the surrogate. When comparing the simulation time of the local finite element simulations and the time needed for the surrogate model including training and evaluation, a significant speedup is observed.

For complicated geometries with holes and overhang structures, we have proposed to train the surrogate models independently on different regions of the component geometry to increase the overall prediction accuracy. Still, this needs to be further investigated. In this aspect, temperature data from the global finite element simulation might be of help. The global model data is not as accurate as the local data but it still contains the relevant physical features of the temperature evolution. With global model data the surrogate model might be able to learn the necessary geometrical features. The global temperature data implicitly contains information about the geometry. Of

course, this is the same for the local model data, however, only the global data can be used as input to the surrogate model because the initial temperatures of the local finite element models are not known when using the surrogate model. Also with this approach different surrogate models for regions with different characteristics should be used.

In the course of this thesis the local model training data has always been generated in whole layers since this is the natural way it comes from the finite element model (as described before, the local models can not be run independently). For surrogate modelling it would be beneficial to have training data more evenly spread over the whole geometry than only from a few complete layers. Therefore, we would propose to look into the possibility of running local finite element models inside of layers without the need to solve their predecessors. The missing initial conditions from the previous local FE model could be substituted by interpolation from the global model (the boundary conditions come from the global model anyway). It is important that the surrogate model can learn the evolution of the physical system from the training data. To achieve this goal it does not matter if the local finite element models are solved using initial conditions from the global model or from the previous local FE model.

The most promising approach to improve the surrogate model in our view is the construction of a good principal component basis. As mentioned before, having a good PC basis is key for the surrogate model to provide accurate results. The basis should include all kind of temperature conditions that can appear during the simulations. Improving on the PC basis will most certainly also help with the problems in overhang regions.

As a further step it should be evaluated if the polynomial chaos model needs to be trained afresh for every new usage of the surrogate model. It might be possible to make a more general model that is only partially adapted to new training data. This would be beneficial because for training the general model, a lot of computational resources could be invested.

Assessing the quality of the surrogate model predictions currently causes difficulties. It would be good to have a method of finding the uncertainties in the predicted temperature profiles to know if they should be trusted or if more training data should be gathered.

The proposed surrogate model is restricted to two dimensional geometries. This restriction is due to the finite element simulation that is also only available in two dimensions. Extending the surrogate model to three dimensional geometries should be straightforward. The steps of the 2D modelling process can be applied to three dimensional temperature data as well. Of course, interpolation and other routines specific to the two-dimensional setup must be adapted. The dimensional reduction and the polynomial chaos model

work in the same way, possibly the number of principal components needs to be changed according to convergence tests.

The surrogate model presented in this thesis can be seen as a proof of concept that surrogate models can help to reduce the computational costs of finite element simulations in additive manufacturing while retaining good accuracy. It can certainly be used as a starting point to further projects advancing the use of simulations of additive manufacturing.

Appendix A

Guideline for surrogate model usage

In the appendix a short explanation on the different steps involved in applying the surrogate model is given.

1. Set up the multi-scale finite element simulation for the geometry you want to solve and run the local and global simulation.
2. With the python scripts *postproc_surrogate.py* and *postproc_global.py* in the directory *scripts*, extract the important temperature data from the odb files of the finite element simulation. The local data is stored in csv files separated into boundary, initial and result temperature at the nodes of the local models. From the global model, temperature values at all the nodes for all the global simulation steps (heating and cooling steps) are stored. For the construction of the surrogate model only the data from the local models is needed.
3. Now, the surrogate model can be constructed in Matlab. Generally, the following steps can be followed:
 - Read in the local data;
 - If a complicated geometry with regions with different surrogate models is used, define which local model is situated in which region;
 - Choose the training and test layers for the surrogate model;
 - Compute the principal component basis (for the different regions) and apply the transformations to get the reduced training data;
 - Choose the parameters for the polynomial chaos expansion and train the model.
4. When the surrogate model is built it can be applied to predict the temperature evolution on the local models in the test layers. To simplify

this step a Matlab function is provided that evaluates the surrogate models on all the local models.

5. Finally, the temperature data from the surrogate model can be used like the temperature data from the local finite element models.

It might be helpful to follow the examples in the files *surrogate_model_square.m* and *surrogate_model_two_pillar.m* in the directory *Matlab/examples*. The former uses the (simple) square geometry with only one surrogate model, the latter the geometry with two pillars and three different surrogate models for the regions with activated and deactivated laser and the overhang. All the steps outlined above are present in these examples. Additionally, it is shown how to combine the temperature data with the data from the global model to plot the temperature evolution at certain nodes.

Appendix B

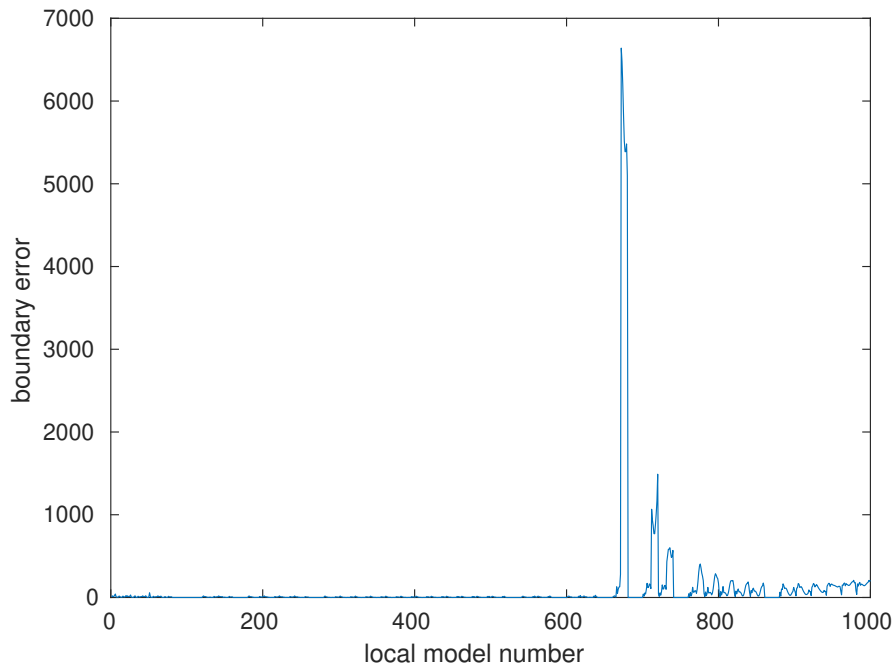
Boundary errors

As mentioned earlier, the uncertainties or the errors in the predictions of the surrogate model can not easily be guessed. There are methods to get a leave-one-out error estimate for the polynomial chaos metamodel. These error estimates are given in terms of the reduced principal component basis and it is difficult to transform them to get estimates in the full temperature space. The error estimates are estimates for the global error. It is useful to know the global error of the surrogate model but this still leaves us without knowledge about the accuracy of the temperature predictions in each local model, or generally, in different parts of the geometry, where accurate temperature values might be very important for post-processing of the simulation results.

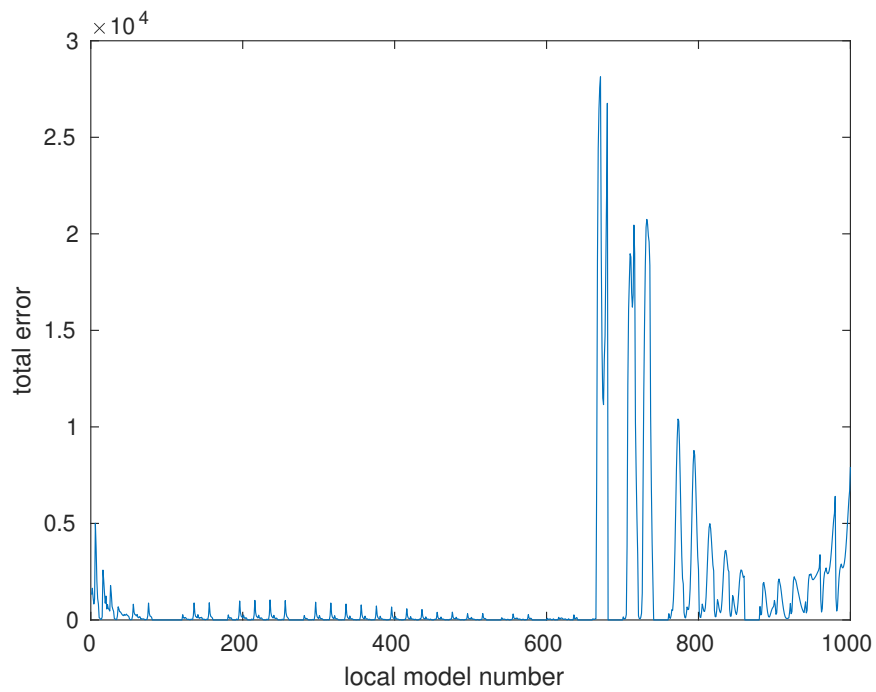
During the presentation of this thesis, an idea of how to estimate the local errors came up. The boundary temperatures of every local model are given as initial values to the local simulations. In the finite element simulation of the local model these boundary values are used to solve the differential equations and of course the prescribed values are met exactly. In the surrogate model, the boundary temperatures are also given as input values and they are used for predicting the result temperatures. Still, the predicted values at the boundary do not equal the initially given boundary values in general. The more accurate the temperature prediction of the surrogate model, the closer the result temperatures at the boundary nodes are to the given boundary values. The idea is that the reverse might also be true to some extent. Namely, if the predicted temperature values of the surrogate model at the boundary nodes match the boundary temperatures well, also the temperature predictions at the interior nodes should be accurate. At least, boundary temperatures being far off from the true values might be a strong indicator that the whole temperature prediction is not really accurate.

In Figure B.1b, the mean square errors of all the surrogate models on the geometry 4.5 are shown. The mean square errors of only the nodes at the boundary of the surrogate model are shown in Figure B.1a. The high

B. BOUNDARY ERRORS



(a) Mean square error on the boundary nodes.



(b) Mean square error of the total surrogate model.

Figure B.1: Mean square error of all the local surrogate models on the geometry 4.5.

peaks around surrogate model number 660 are due to the overhang structure. There, the errors are high also in the boundaries. For higher local model numbers, the errors in the boundary drops faster than the total error but there seems to be some correlation between the two errors in this setting. For small local model numbers, the predictions have a very small error both in the interior as on the boundary. In absence of a better approach on how to quantify the local prediction errors, looking at the errors on the boundary nodes seems to be reasonable (at least to get a qualitative picture).

Bibliography

- [1] Felix Baumann, André Sekulla, Michael Hassler, Benjamin Himpel, and Markus Pfeil. Trends of machine learning in additive manufacturing. *International Journal of Rapid Manufacturing*, 7, 01 2018.
- [2] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6):2345 – 2367, 2011.
- [3] Géraud Blatman and Bruno Sudret. Sparse polynomial chaos expansions of vector-valued response quantities. 01 2013.
- [4] Stijn Clijsters, Tom Craeghs, and Jean-Pierre Kruth. A priori process parameter adjustment for slm process optimization. pages 553–560. Bartolo, PJ, Taylor & Francis Group., 2012.
- [5] William Frazier. Metal additive manufacturing: A review. *Journal of Materials Engineering and Performance*, 23, 06 2014.
- [6] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002.
- [7] J. P. Kruth, X. Wang, Tahar Laoui, and Ludo Froyen. Lasers and materials in selective laser sintering, 2003.
- [8] Melih Kushan, Özgür Poyraz, Yagiz Uzunonut, and Sezan Orak. Review article systematical review on the numerical simulations of laser powder bed additive manufacturing. 12 2018.
- [9] C. Lataniotis, E. Torre, S. Marelli, and B. Sudret. UQLab user manual – The Input module. Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019. Report n# UQLab-V1.3-102.

- [10] Ahmed Maamoun, Jack Xue, Mohamed Elbestawi, and S.C. Veldhuis. Effect of selective laser melting process parameters on the quality of al alloy parts: Powder characterization, density, surface roughness, and dimensional accuracy. *Materials*, 11:2343, 12 2018.
- [11] R. Mao, H. Zhu, L. Zhang, and A. Chen. A new method to assist small data set neural network learning. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 1, pages 17–22, Oct 2006.
- [12] S. Marelli and B. Sudret. UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019. Report # UQLab-V1.3-104.
- [13] Stefano Marelli and Bruno Sudret. *UQLab: A Framework for Uncertainty Quantification in Matlab*, pages 2554–2563.
- [14] Joseph B. Nagel, Jörg Rieckermann, and Bruno Sudret. Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration: Application to urban drainage simulation. *Reliability Engineering & System Safety*, 195:106737, 2020.
- [15] Thomas J. Santner, Brian J. Williams, and William I. Notz. *The Design and Analysis of Computer Experiments*, volume 1. Springer-Verlag, 2003.
- [16] Michael Schmidt, Marion Merklein, David Bourell, Dimitri Dimitrov, Tino Hausotte, Konrad Wegener, Ludger Overmeyer, Frank Vollertsen, and Gideon N. Levy. Laser based additive manufacturing in industry and academia. *CIRP Annals*, 66(2):561 – 583, 2017.
- [17] Jonathon Shlens. A tutorial on principal component analysis, 2014.
- [18] Gustavo Tapia, Saad Khairallah, Manyalibo Matthews, Wayne King, and Alaa Elwany. Gaussian process-based surrogate modeling framework for process planning in laser powder-bed fusion additive manufacturing of 316l stainless steel. *The International Journal of Advanced Manufacturing Technology*, 94, 02 2018.
- [19] E. Torre, S. Marelli, and B. Sudret. UQLab user manual – Statistical inference. Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019. Report n# UQLab-V1.3-114.
- [20] Emiliano Torre, Stefano Marelli, Paul Embrechts, and Bruno Sudret. Data-driven polynomial chaos expansion for machine learning regression. *Journal of Computational Physics*, 388:601 – 623, 2019.

- [21] Bey Vrancken, Ruben Wauthle, Jean-Pierre Kruth, and Jan Humbeeck. Study of the influence of material properties on residual stress in selective laser melting. *24th International SFF Symposium - An Additive Manufacturing Conference, SFF 2013*, pages 393–407, 01 2013.
- [22] Dongbin Xiu and George Em Karniadakis. The wiener-askey polynomial chaos for stochastic differential equations. *SIAM J. Scientific Computing*, 24:619–644, 2002.



Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Surrogate modelling for multiscale thermal simulation of powder-bed additive manufacturing

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

Keller

First name(s):

Fabian

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Richterswil, 04.04.2020

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.