# 1 Practical Applications

## 1.1 Parameter Estimation

We will now apply the concept of shape-constraint P-splines onto real-life data to incorporated a priori domain knowledge into the modeling process. The data is generated from a heat treatment process of aluminum. The aluminum is heated up to a specific temperature, hold at this temperature for some predefined time and then cooled using water jets. The controlled heating and cooling enhances structural properties of the aluminum.

The area of interest is the cooling phase, in which some highly non-linear processes take place due to phase changes of the cooling medium. Therefore, modeling using first-principle methods is quite difficult. Nevertheless, we can use these first-principle ideas in the form of a priori domain knowledge. To model the cooling phase, we try to estimate the heat transfer coefficient, i.e.

$$\alpha := \alpha(T, \dot{m}), \tag{1.1}$$

as a function of the temperature $T$ of the aluminum and the mass flow $\dot{m}$ of the cooling medium. We know beforehand, that the heat transfer coefficient $\alpha$ may only increase with increasing mass flow $\dot{m}$ and that it shows unimodal peak behavior for increasing temperature $T$, motivated by the so-called Leidenfrost effect, see [28] and [29].

The data situation is visualized in Figure 1.1. Here we show how many data points are given in a square area of approximately 50 K and 0.9 kg/s, which represent approximately 1% of the whole input space. We have some regions, where no data is available, while the majority of data points is located in small areas. This problem, i.e. of irregular data distribution, is often encountered in real-world situations.

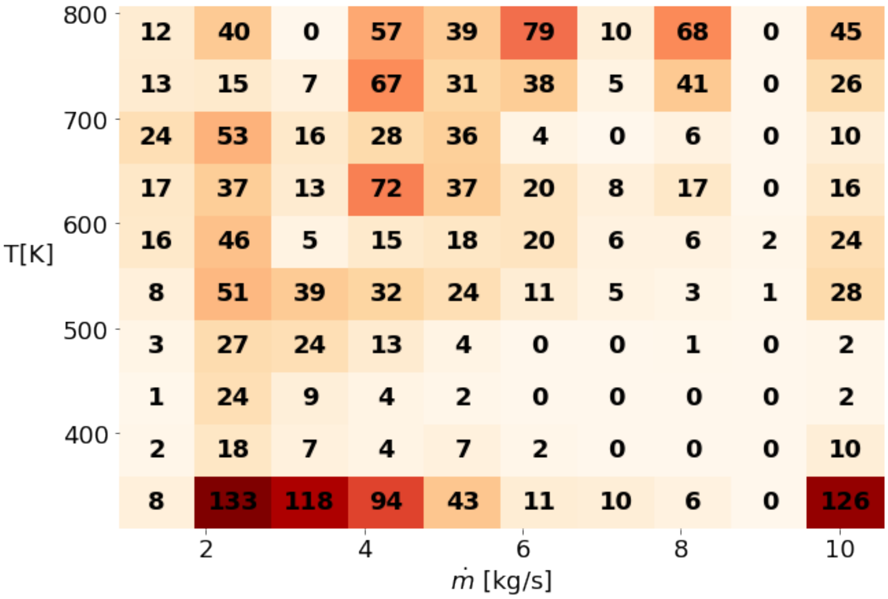

Figure 1.1: Data situation.

The data distribution may lead to some difficulties when using B-splines and tensor-product B-splines, as they do not handle sparse data situations well, cf. Chapter **??**. Nevertheless, using regularization via smoothness and constraint penalties should help to cope the data situation.

We will now use various models based on B-spline bases, tensor-product B-spline bases and their shape-constraint alternatives to recover a model for the heat transfer coefficient given the data. The following list, in which $s(i)$ denotes the use of a B-spline basis for input $i$ and $t(i,i)$ denotes the use of a tensor-product B-spline basis for inputs $i$ and $j$, describes the models. We use the mass flow $\dot{m}$ as input dimension 1 and the temperature $T$ as input dimension 2.

(i) $\mathrm{M1} = s(1) + s(2)$

(ii) $\mathrm{M2} = s(1) + s(2)$, using monotonicity for $s(1)$ and unimodal peak behavior for $s(2)$

(iii) $\mathrm{M3} = t(1, 2)$

(iv) $\mathrm{M4} = t(1, 2)$, using monotonicity for input 1

(v) $\mathrm{M5} = s(1) + s(2) + t(1, 2)$, as additive model using M1 and M3

(vi) $\mathrm{M6} = s(1) + s(2) + t(1, 2)$, as additive model using M2 and M4

To fit the models listed above, we perform a randomized train-validation split on the data, i.e. we split the data into a training set $\mathcal{D}_{\text{train}}$ and a validation set $\mathcal{D}_{\text{val}}$, fit the models to the resulting training set and evaluate its performance by calculating the mean squared error on the validation set $\mathcal{D}_{\text{val}}$ as well as by visual inspection. We choose to split the data into sets of the same size to generate a more stable estimation of the prediction error for previously unseen data. A visual inspection of the of the data distribution given by the train-validation split is given in Figure 1.2.

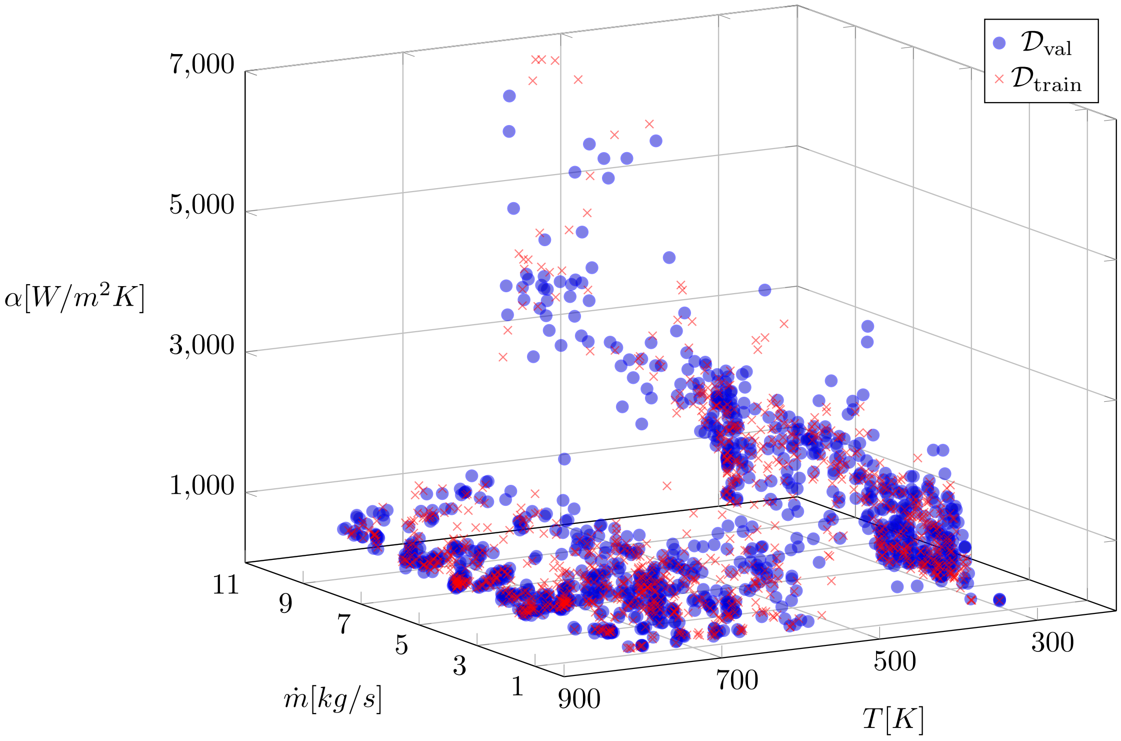

Figure 1.2: Training set $\mathcal{D}_{\text{train}}$ and validation set $\mathcal{D}_{\text{val}}$.

Note that we use a smoothness penalty optimized via generalized cross-validation for each individual basis. The unconstraint models M1, M3 and M5 are therefore P-spline models rather than B-spline models. The shape-constraint models M2, M4 and M6 are SCP-spline models.

The mean squared errors evaluated on the validations set $\mathcal{D}_{\text{val}}$ are given in Table 1.1. According to these, the best model is M4, i.e. the shape-constraint tensor-product B-spline with a monotonicity constraint in the mass flow dimension. The models M3, i.e. the tensor-product B-spline, and M6, i.e. the additive model using shape-constraints, perform nearly as well as M4 according to the mean squared error on the validation set.

| Model | $\text{MSE}_{\text{val}}$ |
|---|---|
| M1 | $1.38 \cdot 10^6$ |
| M2 | $3.46 \cdot 10^5$ |
| M3 | $2.14 \cdot 10^5$ |
| M4 | $2.05 \cdot 10^5$ |
| M5 | $4.36 \cdot 10^6$ |
| M6 | $2.15 \cdot 10^5$ |

Table 1.1: Mean squared errors on the validation set $\mathcal{D}_{\text{val}}$.

The predictions for model M4 are shown in Figure 1.3. The peak behavior in the temperature dimension is clearly visible, as well as an increasing trend within the massflow dimension. Therefore, we conclude that model M4 is the superior model with regards to the domain knowledge and data fidelity.

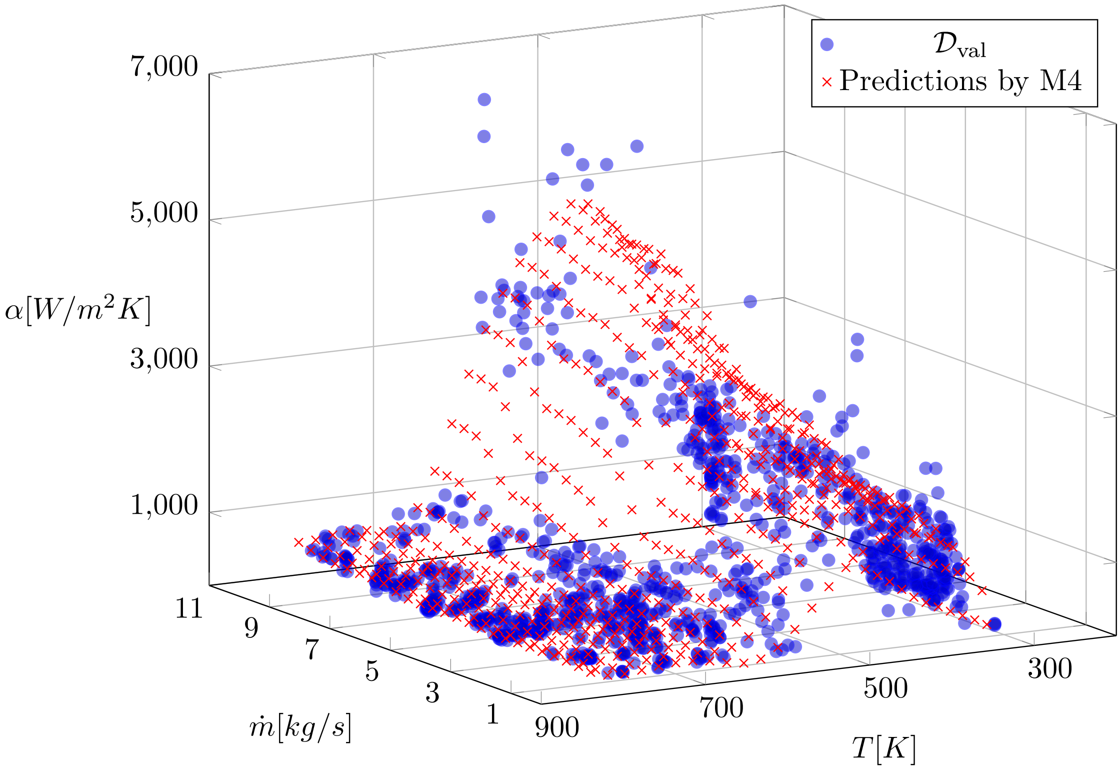

Figure 1.3: Validation set $\mathcal{D}_{\text{val}}$ and predictions by model M4.

The predictions for model M6 are shown in Figure 1.4. Here, the peak behavior can also be identified, but in a weaker fashion as for model M4 in Figure 1.3. We also obtain an increasing trend in the massflow dimension.

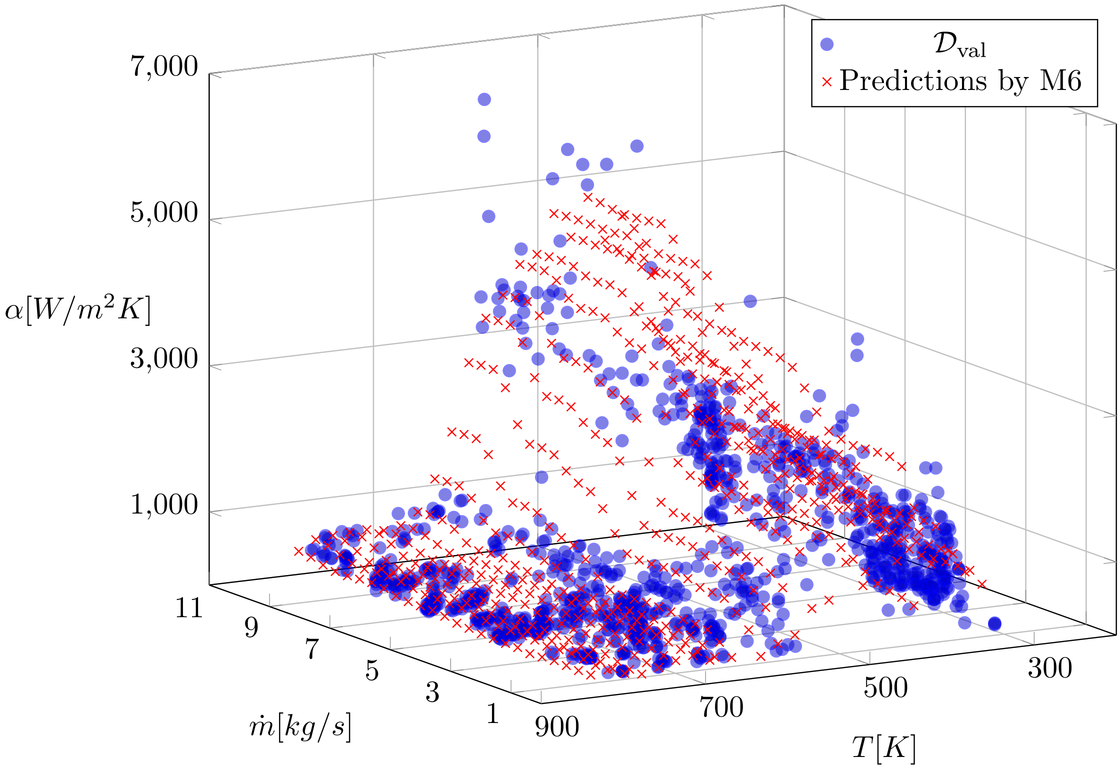

Figure 1.4: Validation set $\mathcal{D}_{\text{val}}$ and predictions by model M6.

The visual inspection of the predictions of model M3 in Figure 1.5 indicates that there is massive overfitting present. Neither smooth, nor the a priori known behavior (increasing in the massflow dimension and a peak behavior in the temperature dimension) is identifiable.

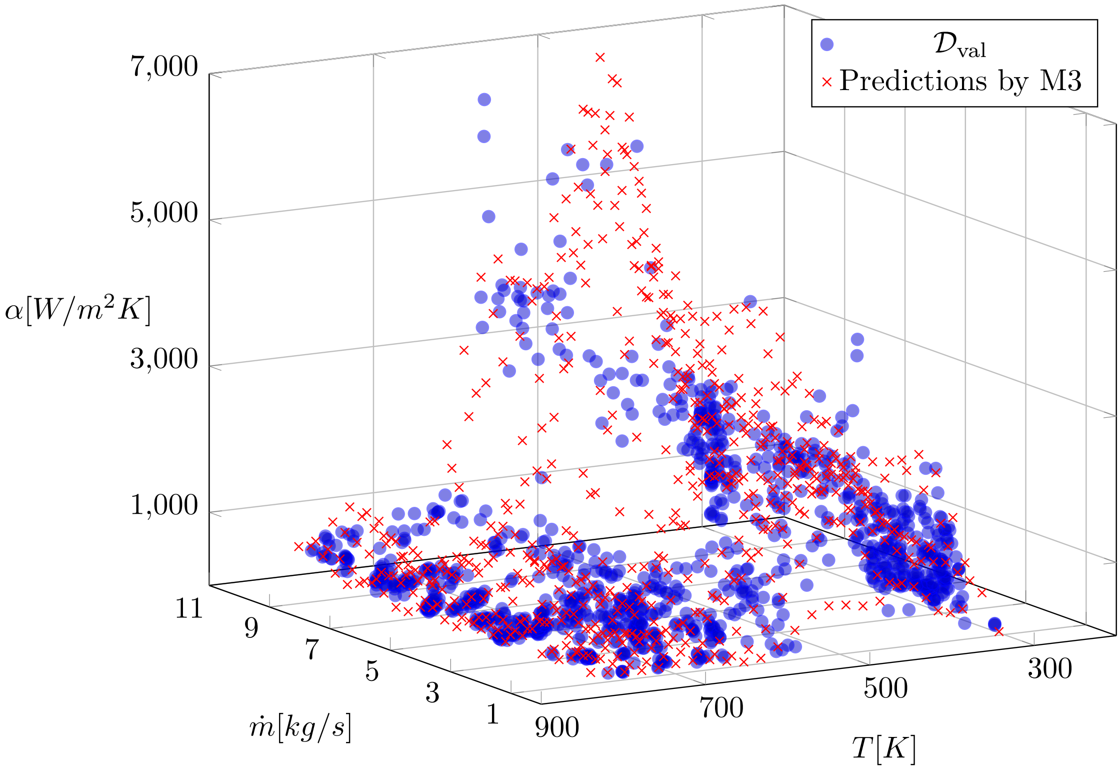

Figure 1.5: Validation set $\mathcal{D}_{\text{val}}$ and predictions by model M3.

We omit the visual presentation of the predictions by the models M1, M2 and M5, since for all of these the mean squared errors on the validation set $\mathcal{D}_{\text{val}}$ is at least a magnitude higher, indicating even worse models.

To summarized, we see that the incorporation of a priori domain knowledge improves the quality of the fit in all of the above models, e.g. M2 is the better model according to the mean squared error on the validation set $\mathcal{D}_{\text{val}}$ than M1, M4 is better than M3 and M6 is better than M5. Therefore, we conclude that the use of a priori domain knowledge through shape-constraints improves the model quality for our real-world data example.

## 1.2 Surrogate Servo-Compensation

As second example, we will now investigate the behavior of a servo-compensation, see Figure 1.6 for a block diagram of the servo-compensation. The region of interest is marked by the red square. We try to model the differential current $i_m^d$ by two inputs. The first input is the measured position of the main valve $s_h$. The second input is the differential flow $q_p^d$.

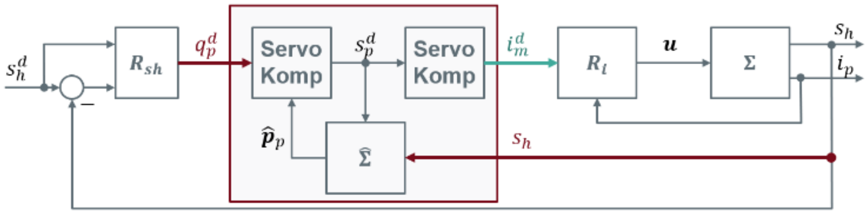

Figure 1.6: Block diagram of the servo-compensation.

The data $\mathcal{D}$ consisting of 1000 points is generated using a validated model based on first-principles and visualized in Figure 1.7. The figure shows two distinct model regions with a discontinuity at $s_h = 0$. The main challenge of here is to model the discontinuity in a smooth way without introducing modeling artifacts, i.e. over- or undershooting behavior. We know beforehand that the differential current $i_m^d$ is monotonic increasing with the differential flow $q_p^d$. We further constrain the models to be monotonic increasing with the differential flow $q_p^d$ to enforce a smooth transition at the discontinuity at $s_h = 0$.

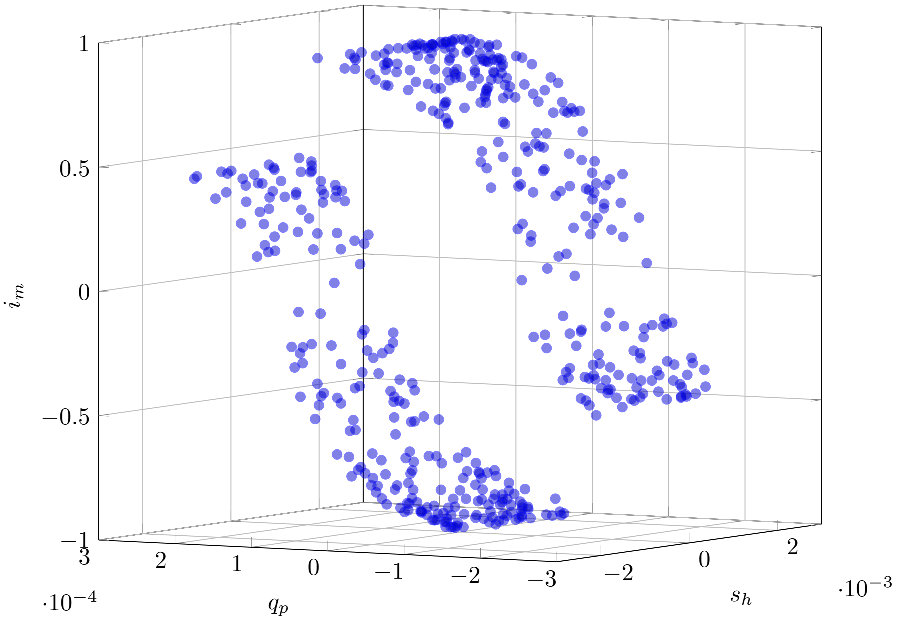

Figure 1.7: Data Situation

We will now use various models based on B-spline bases, tensor-product B-spline bases and their shape-constraint alternatives to recover a model for the differential current $i_m^d$. The following list, in which $s(i)$ denotes the use of a B-spline basis for input $i$ and $t(i, j)$ denotes the use of a tensor-product B-spline basis for inputs $i$ and $j$, describes the models.

(i) $\text{M1} = s(1) + s(2)$

(ii) $\text{M2} = s(1) + s(2)$, using monotonicity for $s(2)$

(iii) M3 $= t(1,2)$

(iv) M4 $= t(1,2)$, using monotonicity for input $s(1)$ and $s(2)$

(v) M5 $= s(2) + t(1,2)$, as additive model using M1 and M3

(vi) M6 $= s(2) + t(1,2)$, as additive model using the constraints in M2 and M4

To fit the models listed above, we perform a randomized train-validation split on the data, i.e. we split the data into a training set $\mathcal{D}_{\text{train}}$ with 750 data points and a validation set $\mathcal{D}_{\text{val}}$ with 250 data points, fit the models to the resulting training set and evaluate its performance by calculating the mean squared error on the validation set $\mathcal{D}_{\text{val}}$ and the effective degree of freedom of the models EDoF, see Section **??**, as well as by visual inspection.

Note that we use a smoothness penalty optimized via generalized cross-validation for each individual basis. The unconstraint models M1, M3 and M5 are therefore P-spline models rather than B-spline models. The shape-constraint models M2, M4 and M6 are SCP-spline models. The mean squared errors evaluated on the validations set $\mathcal{D}_{\text{val}}$ are given in Table 1.2 as well as the effective degree of freedom of the models as measure of the model complexity.

| Model | $\text{MSE}_{\text{val}}$ | EDoF |
|---|---|---|
| M1 | $3.45 \cdot 10^{-2}$ | 20.49 |
| M2 | $3.46 \cdot 10^{-2}$ | 15.48 |
| M3 | $5.83 \cdot 10^{-3}$ | 262.33 |
| M4 | $5.08 \cdot 10^{-3}$ | 57.22 |
| M5 | $5.7 \cdot 10^{-3}$ | 265.21 |
| M6 | $4.72 \cdot 10^{-3}$ | 53.6 |

Table 1.2: Mean squared errors on the validation set $\mathcal{D}_{\text{val}}$ and the effective degree of freedom EDoF of the models.

The purely B-spline based models M1 and M2 show low degrees of freedom but higher values of the mean squared error on the validation set $\mathcal{D}_{\text{val}}$. Therefore, they lack some flexibility to model the data accurately. The predictions for these models are visualized in Figure 1.8 and Figure 1.9. Note the left plot is the projection of the three dimensional data onto the $s_h$ axis for better visualization of the model predictions. The purely tensor-product B-spline based models M3 and M4 have a significantly lower mean squared error on the validation set $\mathcal{D}_{\text{val}}$. Both models are visualized in Figure 1.10 and Figure 1.11. With regards to the effective degree of freedom, we see that the regularization through the shape-constraints clearly decreases the model flexibility which leads to a better generalization behavior seen by the lower $\text{MSE}_{\text{val}}$ of M4 compared to M3. This can also be seen in the left plot of Figure 1.10. Note the prediction point (red) at $s_h \approx 0.7 * 10^{-4}$. Such prediction is only possible for a very wiggly and non-smooth function. The additive models using B-splines and tensor-product B-splines, i.e. M5 and M6, both show low mean squared errors on the validation set $\text{MSE}_{\text{val}}$. We again see the effect of the shape-constraint in the effective degree of freedom as well as in the plots in Figure 1.12 and Figure 1.13.

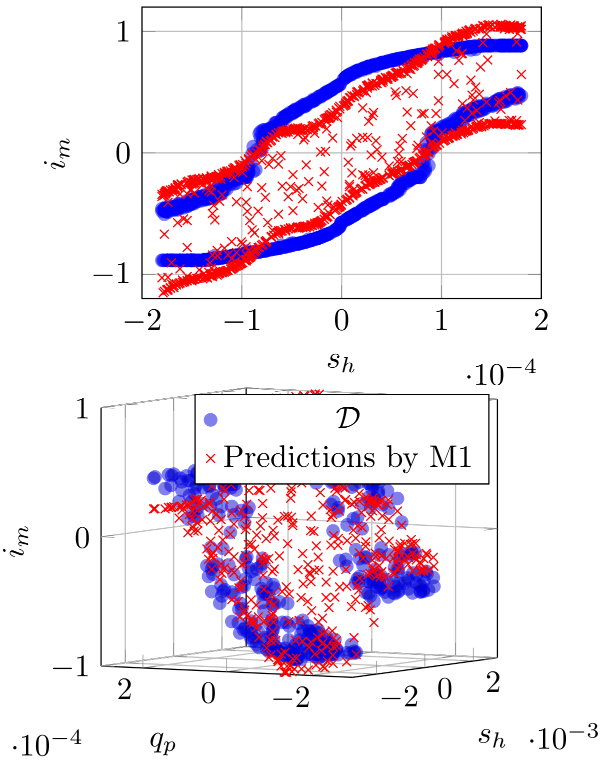

Figure 1.8: Data and predictions by model M1.

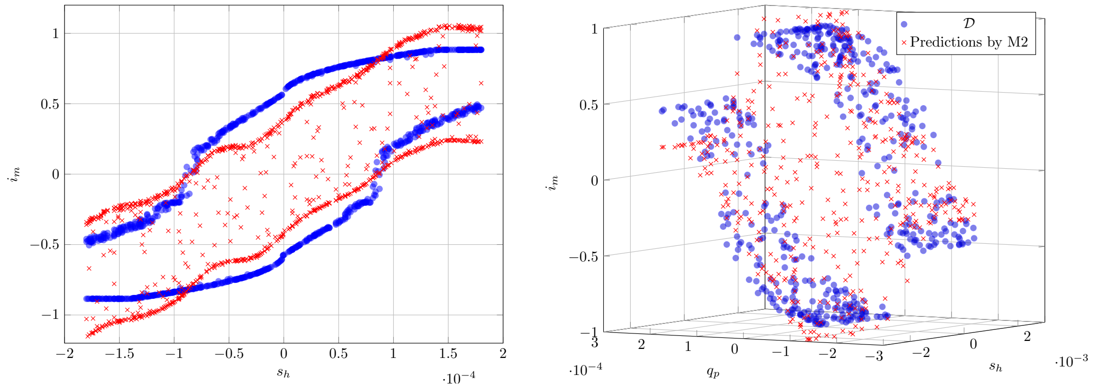

Figure 1.9: Data and predictions by model M2.

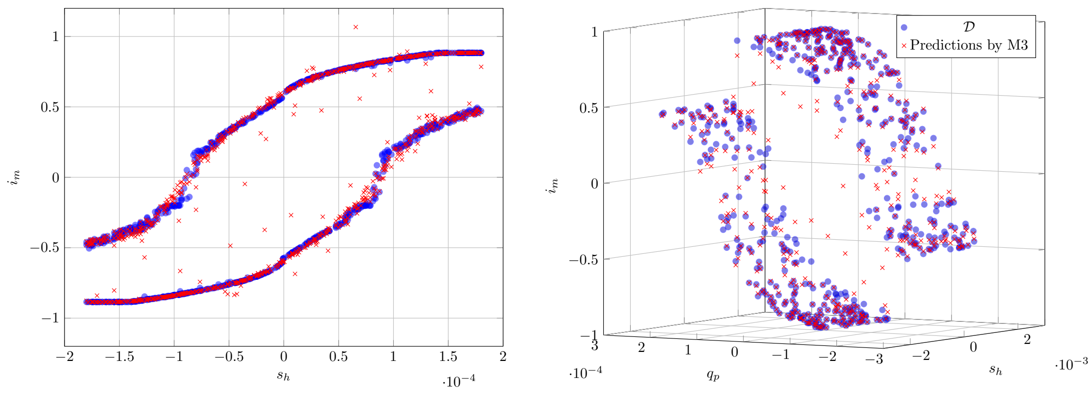

Figure 1.10: Data and predictions by model M3.

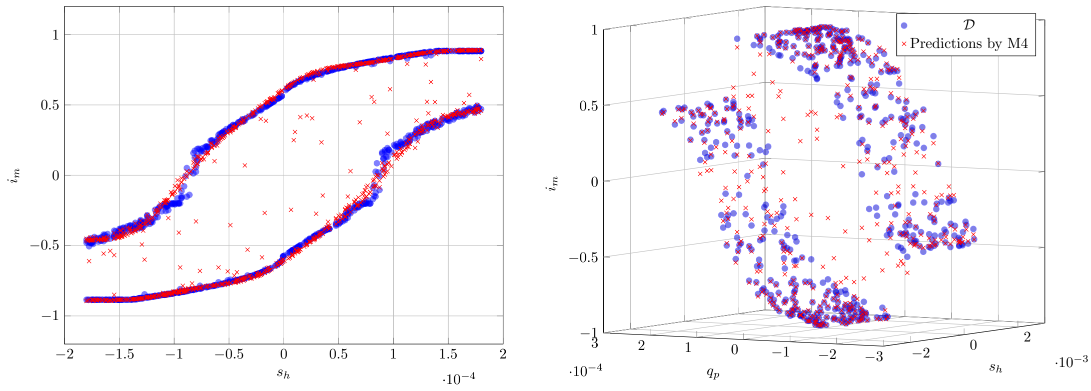

Figure 1.11: Data and predictions by model M4.

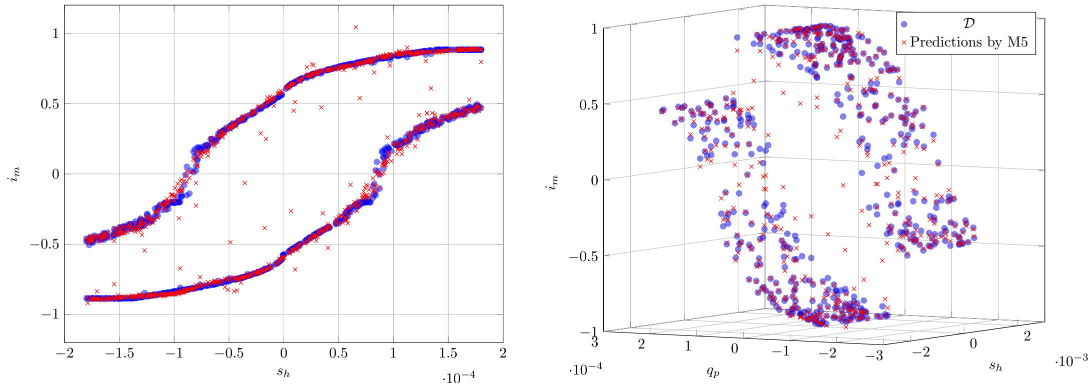

Figure 1.12: Data and predictions by model M5.

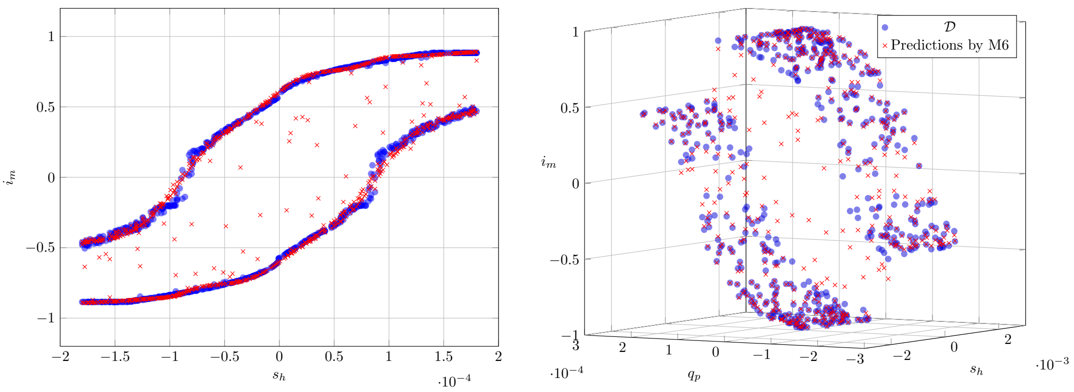

Figure 1.13: Data and predictions by model M6.

To summarized, we see that the incorporation of a priori domain knowledge improves the quality of the fit for this example too. The constrained models show lower mean squared errors on the validation data as well as lower effective degrees of freedom. The best model is clearly M6, since it has a significantly lower $\text{MSE}_{\text{val}}$ as well as EDoF.

# 2 Summary and Outlook

The main goal of this thesis was to find a way of including a priori domain knowledege into the fitting process of a data driven modeling approach. We showed that this is possible using the concept of shape-constraints, see Chapter **??**, and B-splines as well as additive regression for multi-dimensional data as presented in Chapter **??** and Chapter **??**. Further, we presented an overview of the related literature regarding the topics of linear models, model selection and B-splines in Chapter **??**.

With regards to future work, it would be interesting to try to create an algorithm that automatically finds the best possible combination of B-splines and tensor-product B-splines for a multi-dimensional problem with some a priori domain knowledge. For example, using 3 inputs, there are various combinations possible to create an additive model, e.g. a B-spline for dimension 1 and a tensor-product B-spline for dimension 2 and 3. An algorithm that automatically chooses the optimal combination with respect to some predefined criterion would help to enhance the usage of this approach.

# A Appendix A

## A.1 Kronecker product

The Kronecker product $\mathbf{A} \otimes \mathbf{B}$ of the $n \times p$-matrix $\mathbf{A}$ and the $r \times q$-matrix $\mathbf{B}$ is defined as $nr \times pq$-matrix

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \dots & a_{1p}\mathbf{B} \\ \vdots & & \vdots \\ a_{n1}\mathbf{B} & \dots & a_{np}\mathbf{B} \end{bmatrix}. \tag{A.1}$$

As example, we define $\mathbf{A} \in \mathbb{R}^{3\times 2}$ and $\mathbf{B} \in \mathbb{R}^{2\times 2}$ as

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}, \tag{A.2}$$

resulting in the matrix $\mathbf{C} \in \mathbb{R}^{6\times 4}$ given by

$$\mathbf{C} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 & 4 \\ 1 & 2 & 2 & 4 \\ 3 & 6 & 4 & 8 \\ 3 & 6 & 4 & 8 \\ 5 & 10 & 6 & 12 \\ 5 & 10 & 6 & 12 \end{bmatrix}. \tag{A.3}$$

## A.2 Row-wise Kronecker product

Given the $n \times p$-matrix$\mathbf{A}$ and the $n \times q$-matrix $\mathbf{B}$, the row-wise Kronecker product is defined as $n \times pq$-matrix $\mathbf{C}$ given by

$$\mathbf{C} = \mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B}_1 & \dots & a_{1p}\mathbf{B}_1 \\ \vdots & & \vdots \\ a_{n1}\mathbf{B}_n & \dots & a_{np}\mathbf{B}_n \end{bmatrix}, \tag{A.4}$$

where $\mathbf{B}_i$ denotes the $i$-th row of the matrix $\mathbf{B}$. As example, we define $\mathbf{A} \in \mathbb{R}^{3\times 3}$ and $\mathbf{B} \in \mathbb{R}^{3\times 2}$ as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \mathbf{B}_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix}, \tag{A.5}$$

resulting in the matrix $\mathbf{C} \in \mathbb{R}^{3\times 6}$ given by

$$\mathbf{C} = \begin{bmatrix} \mathbf{A}_1 \otimes \mathbf{B}_1 \\ \mathbf{A}_2 \otimes \mathbf{B}_2 \\ \mathbf{A}_3 \otimes \mathbf{B}_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 & 4 & 3 & 6 \\ 4 & 8 & 5 & 10 & 6 & 12 \\ 7 & 14 & 8 & 16 & 9 & 19 \end{bmatrix}. \tag{A.6}$$

The row-wise Kronecker product is also known as *face-splitting product* or as *transposed Khatri-Rao product* [30].

# B Appendix B

The following derivation is take from [5]. To find an optimal solution for the penalized least squares objective function for shape-constraint P-splines given by

$$\mathrm{L}(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\boldsymbol{\beta}^{\mathrm{T}}\mathbf{K}\boldsymbol{\beta} + \lambda_c\boldsymbol{\beta}^{\mathrm{T}}\mathbf{K}_c\boldsymbol{\beta}, \tag{B.1}$$

we use a Newton-Raphson scheme. At each iteration $i$, the new estimate $\hat{\boldsymbol{\beta}}_{i+1}$ is computed such that

$$\mathbf{g}(\boldsymbol{\beta}_i) + \mathbf{H}(\boldsymbol{\beta}_i)(\boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}_i) = 0, \tag{B.2}$$

with $\mathbf{g}$ being the gradient of and $\mathbf{H}$ being the Hessian of $\mathrm{L}(\boldsymbol{\beta})$. The gradient of $\mathrm{L}(\boldsymbol{\beta})$ equals to

$$\mathbf{g}(\boldsymbol{\beta}) = -2\mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\Big[\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + \lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c(\boldsymbol{\beta})\mathbf{D}_c\Big]\boldsymbol{\beta} + \lambda_c\boldsymbol{\beta}^{\mathrm{T}}\mathbf{D}_c^{\mathrm{T}}\frac{\partial\mathbf{V}_c(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}\mathbf{D}_c\boldsymbol{\beta}, \tag{B.3}$$

which can be simplified, see [5], to

$$\mathbf{g}(\boldsymbol{\beta}) = -2\mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\Big[\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + \lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c(\boldsymbol{\beta})\mathbf{D}_c\Big]\boldsymbol{\beta}. \tag{B.4}$$

The gradient is therefore a piecewise linear function of $\boldsymbol{\beta}$ since $\mathbf{V}(\boldsymbol{\beta})$ is a diagonal matrix with 0s and 1s as diagonal elements, see (B.4). This implies that the Hessian is a step function of $\boldsymbol{\beta}$ and equal to

$$\mathbf{H}(\boldsymbol{\beta}) = 2\mathbf{X}^{\mathrm{T}}\mathbf{X} + 2\lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + 2\lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c(\boldsymbol{\beta})\mathbf{D}_c + 2\lambda_c\mathbf{D}_c^{\mathrm{T}}\frac{\partial\mathbf{V}_c(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}\mathbf{D}_c. \tag{B.5}$$

The last part can be neglected again, see [5]. Substituting (B.4) and (B.5) into (B.2) leads to

$$\begin{aligned}0 = \; & -2\mathbf{X}^{\mathrm{T}}\mathbf{y} \\ & + 2\Big[\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + \lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c(\boldsymbol{\beta}_i)\mathbf{D}_c\Big]\boldsymbol{\beta}_i \\ & + 2\Big[\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + \lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c(\boldsymbol{\beta}_i)\mathbf{D}_c\Big](\boldsymbol{\beta}_{i+1} - \boldsymbol{\beta}_i),\end{aligned} \tag{B.6}$$

which can be reformulated as

$$-\mathbf{X}^{\mathrm{T}}\mathbf{y} + \Big[\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + \lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c(\boldsymbol{\beta}_i)\mathbf{D}_c\Big]\boldsymbol{\beta}_{i+1} = 0, \tag{B.7}$$

and, hence, we obtain

$$\boldsymbol{\beta}_{i+1} = \Big[\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + \lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c(\boldsymbol{\beta}_i)\mathbf{D}_c\Big]^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}. \tag{B.8}$$

# Bibliography

[1] W. R. Ashby, *An Introduction to Cybernetics*. Chapman & Hall Ltd, 1961.

[2] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2018, pp. 0210–0215.

[3] L. Fahrmeir, T. Kneib, S. Lang, and B. Marx, *Regression*. Springer, 2007.

[4] B. Hofner, J. Müller, and T. Hothorn, "Monotonicity-constrained species distribution models," *Ecology*, vol. 92, no. 10, pp. 1895–1901, 2011.

[5] K. Bollaerts, P. H. C. Eilers, and I. Van Mechelen, "Simple and multiple p-splines regression with shape constraints," *British Journal of Mathematical and Statistical Psychology*, vol. 59, no. 2, pp. 451–469, 2006.

[6] O. Nelles, *Nonlinear System Identification*. Springer, 2001.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[8] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, pp. 1–67, 1991.

[9] C. De Boor, *A practical Guide to Splines*. Springer, 1978, vol. 27.

[10] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[12] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[13] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[14] Y. S. Abu-Mostafa, "Learning from hints in neural networks," *Journal of Complexity*, vol. 6, no. 2, pp. 192–198, 1990.

[15] J. Sill and Y. S. Abu-Mostafa, "Monotonicity hints," in *Advances in Neural Information Processing Systems*, 1997, pp. 634–640.

[16] E. Garcia and M. Gupta, "Lattice regression," in *Advances in Neural Information Processing Systems*, 2009, pp. 594–602.

[17] M. M. Fard, K. Canini, A. Cotter, J. Pfeifer, and M. Gupta, "Fast and flexible monotonic functions with ensembles of lattices," in *Advances in Neural Information Processing Systems*, 2016, pp. 2919–2927.

[18] M. Gupta, A. Cotter, J. Pfeifer, K. Voevodski, K. Canini, A. Mangylov, W. Moczydlowski, and A. Van Esbroeck, “Monotonic calibrated interpolated look-up tables,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3790–3836, 2016.

[19] S. You, D. Ding, K. Canini, J. Pfeifer, and M. Gupta, “Deep lattice networks and partial monotonic functions,” in *Advances in Neural Information Processing Systems*, 2017, pp. 2981–2989.

[20] S. N. Wood, *Generalized Additive Models*. CRC Press, 2017.

[21] D. G. Luenberger, Y. Ye, *et al.*, *Linear and Nonlinear Programming*. Springer, 2016, vol. 2.

[22] V. Blobel and E. Lohrmann, *Statistische und numerische Methoden der Datenanalyse*. Springer, 1998.

[23] G. H. Golub, M. Heath, and G. Wahba, “Generalized cross-validation as a method for choosing a good ridge parameter,” *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[24] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[25] R. L. Eubank and C. H. Spiegelman, “Testing the goodness of fit of a linear model via nonparametric regression techniques,” *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 387–392, 1990.

[26] P. H. Eilers and B. D. Marx, “Flexible smoothing with *b*-splines and penalties,” *Statistical Science*, pp. 89–102, 1996.

[27] F. O’Sullivan, “A statistical perspective on ill-posed inverse problems,” *Statistical Science*, pp. 502–518, 1986.

[28] F. Mayinger, *Strömung und Wärmeübergang in Gas-Flüssigkeits-Gemischen*. Springer-Verlag, 2013.

[29] B. H. and S. K., *Heat and Mass Transfer*. Springer, 2006, vol. 2.Auflage.

[30] V. I. Slyusar, “Analytical model of the digital antenna array on a basis of face-splitting matrixs product,” in *Proceedings of International Conference on Antenna Theory and Techniques*, 1997, pp. 108–109.

# Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct – Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In– noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, XX. Monat, Jahr

Vorname Nachname