

Diploma Thesis

# Metamodels for computationally intensive applications in structural mechanics

submitted in satisfaction of the requirements for the degree of  
Diplom-Ingenieur  
of the TU Wien, Faculty of Civil Engineering

---

Diplomarbeit

## Metamodelle für rechenintensive Anwendungen in der Strukturmechanik

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines  
Diplom-Ingenieurs  
eingereicht an der Technischen Universität Wien, Fakultät für Bauingenieurwesen

von

**Norbert Zehetmayer, BSc**

Matr.Nr.: 01225475

unter der Anleitung von

Prof. Dipl.-Ing. Dr.techn. **Christian Bucher**

Institut für Hochbau, Baudynamik und Gebäudetechnik  
Forschungsbereich Strukturodynamik und Risikobewertung von Tragwerken  
Technische Universität Wien  
Karlsplatz 13/208, 1040 Wien, Österreich

Wien, im Oktober 2021

---



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Kurzfassung

In der vorliegenden Arbeit werden Metamodelle und deren Anwendungen erläutert.

Im ersten Teil wird auf die Theorie der Metamodelle eingegangen. Es wird im Speziellen die Herleitung von linearen Regressionsmodellen erläutert. Im Anschluss werden Arten vorgestellt, um Antwortflächen zu beschreiben. Unter den vorgestellten Methoden sind polynomiale Regression, Weighted-Interpolation, Moving-Least-Square-Regression, Radiale-Basis-Funktion-Interpolation, Kriging, Gauß-Prozess-Regression und künstliche neuronale Netzwerke. Im Folgenden wird auf die Validierung von Metamodellen eingegangen. Hierbei werden Fehlermaße, der Determinationskoeffizient, Split-Sampling, Kreuzvalidierung und der Prognosekoeffizient näher behandelt. Als Nächstes werden Methoden zur statistischen Versuchsplanung beschrieben. Es werden Verfahren wie die vollfaktoriellen Versuchspläne, Central-Composite-Design und teilfaktorielle Versuchspläne erklärt. Zuletzt werden das Monte-Carlo- und Latin-Hypercube-Sampling vorgestellt.

Im nächsten Kapitel werden Anwendungsmöglichkeiten von Metamodellen in der Strukturmechanik erläutert. Hierbei wird auf die Sensitivitätsanalyse, Berechnung der Versagenswahrscheinlichkeit und Optimierung eingegangen. Zusätzlich wird der Aufbau von Metamodellen für Feldgrößen erläutert.

Im nächsten Kapitel wird mithilfe der vorgestellten Methoden die Heißgaswand eines Raketentriebwerks untersucht. Hierbei wird zuerst durch eine Sensitivitätsanalyse die wichtigsten Parameter ermittelt. Danach wird ein Feld-Metamodell für die plastischen Verzerrungen für acht Zyklen aufgebaut, um von diesen die Anzahl von Lastzyklen, bei dem ein Versagen der Heißgaswand auftritt, zu berechnen. Mit diesem Modell wird im Anschluss ein Histogramm der Zyklen bis zum Versagen erstellt.

Im letzten Kapitel wird generell über die Anwendbarkeit von Metamodell diskutiert.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Abstract

In this master thesis, metamodels and their applications are explained.

The first part deals with the theory of metamodels. In particular, the derivation of linear regression models is explained. Subsequently, types of describing response surfaces are presented. These include polynomial regression, weighted interpolation, moving least square regression, radial basis function interpolation, kriging, Gaussian process regression and artificial neural networks. In the following, the validation of metamodels is discussed. Here, error measures, the coefficient of determination, split sampling, cross-validation and the prediction coefficient are discussed in more detail. Next, methods for statistical experimental design are described. Methods such as full factorial experimental designs, central composite design and partial factorial experimental designs are explained. Finally, Monte Carlo and Latin hypercube sampling are introduced.

In the next chapter, possible applications of metamodels in structural mechanics are explained. Sensitivity analysis, calculation of failure probability and optimisation are discussed. In addition, the structure of metamodels for field variables is explained.

In the next chapter, the hot gas wall of a rocket engine is examined using the methods presented. First, the most important parameters are determined by a sensitivity analysis. Afterwards, a field metamodel for the plastic strain is built for 8 cycles in order to calculate the number of load cycles at which a failure of the hot gas wall occurs. This model is then used to create a histogram of the cycles to failure.

In the last chapter, the applicability of metamodels is discussed in general.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
<b>2</b>	<b>Metamodelle</b>	<b>9</b>
2.1	Metamodellierung . . . . .	9
2.1.1	Grundlagen . . . . .	9
2.1.2	Regressionsmodelle . . . . .	10
2.1.3	Polynomiale Regression . . . . .	11
2.1.4	Weighted-Interpolation . . . . .	13
2.1.5	Moving-Least-Square Regression . . . . .	14
2.1.6	Radiale-Basis-Funktionen-Interpolation . . . . .	16
2.1.7	Kriging-Verfahren . . . . .	18
2.1.8	Gauß-Prozess-Regression . . . . .	23
2.1.9	Künstliche neuronale Netzwerke (Artificial Neural Networks ANN) . . . . .	24
2.2	Validierung . . . . .	28
2.2.1	Fehlermaße . . . . .	28
2.2.2	Split-Sampling . . . . .	29
2.2.3	Kreuzvalidierung . . . . .	30
2.3	Statistische Versuchsplanung (Design of Experiments) . . . . .	30
2.3.1	Vollfaktorieller Versuchsplan . . . . .	31
2.3.2	Central Composite Designs . . . . .	32
2.3.3	Teilfaktorieller Versuchsplan . . . . .	32
2.3.4	Monte-Carlo-Methode . . . . .	34
2.3.5	Latin-Hypercube-Sampling . . . . .	34
<b>3</b>	<b>Anwendungen</b>	<b>36</b>
3.1	Sensitivitätsanalyse . . . . .	36
3.1.1	Bedeutungskoeffizient . . . . .	36
3.1.2	Varianzbasierte Sensitivitätsanalyse . . . . .	37
3.2	Versagenswahrscheinlichkeit . . . . .	38
3.2.1	Response-Surface-Method (RSM) . . . . .	38
3.2.2	Directional-Sampling . . . . .	40
3.3	Optimierungsprobleme . . . . .	42
3.4	Approximation von Feldgrößen mittels Singulärwertzerlegung . . . . .	44
<b>4</b>	<b>Lebensdauer Vorhersage der Heißwand einer Brennkammer</b>	<b>47</b>
4.1	Geometrie und Randbedingungen . . . . .	47
4.2	Materialmodell . . . . .	48
4.3	Sensitivitätsanalyse . . . . .	50
4.4	Statistische Analyse . . . . .	52
<b>5</b>	<b>Schlussfolgerungen</b>	<b>59</b>

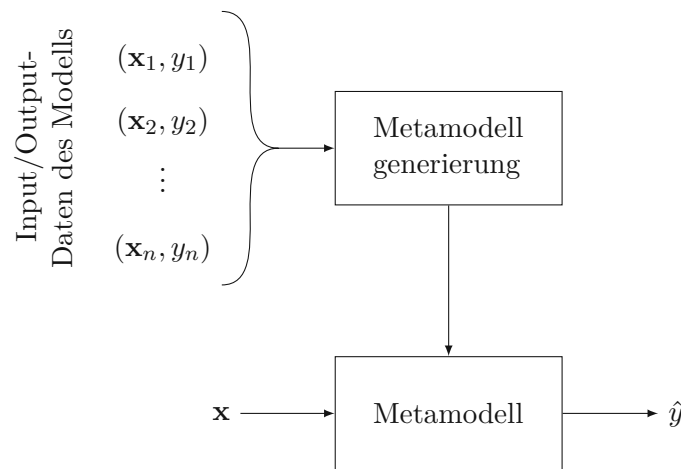
# Kapitel 1

## Einleitung

Durch die Steigerung der Leistungsfähigkeit von Computern in den letzten Jahrzehnten und die damit einhergehende Entwicklung von numerischen Methoden in den Ingenieurwissenschaften, können Strukturen im Ingenieurbereich genauer untersucht werden. Einerseits können Modelle feiner modelliert und in derselben Zeit berechnet werden wie gröbere Modelle vor einigen Jahren. Andererseits können nichtlineare und/oder zeitabhängige Probleme berücksichtigt werden. Es haben sich daher auch flexible Methoden wie die Finite-Elemente-Methode (FEM) durchgesetzt. Mit ihnen können reale Strukturen genauer approximiert und nichtlineares u. dynamisches Verhalten untersucht werden.

Jedoch wird durch nichtlineare und dynamische Probleme der Rechenaufwand erheblich erhöht, da das Rechenmodell oft ausgewertet werden muss, wie es z. B. bei Optimierungsproblemen oder Zuverlässigkeitsanalysen der Fall ist. Um den Zeitaufwand zu verringern, können Metamodelle verwendet werden. Hierbei wird das Modell an ausgewählten Stützstellen im Raum (Eingangsvariablen des Modells) berechnet und mithilfe von Regressions- oder Interpolationsverfahren ein Metamodell erstellt. Metamodelle sind mathematisch einfach aufgebaut und können daher schneller ausgewertet werden. Durch die einfache Gestalt können auch mathematische Operationen wie z. B. Ableitungen leichter durchgeführt werden.

In Abbildung 1.1 ist der Ablauf bei der Metamodellierung vereinfacht dargestellt. Zuerst müssen für ein Set an Inputparametern  $\mathbf{x}_i$   $i = 1, \dots, n$  die Ergebnisse des Modells  $z_i = \mathcal{M}(\mathbf{x}_i)$  berechnet werden. Mit den Eingabe- und Ausgabewerten  $(\mathbf{x}_i, z_i)$  wird im Anschluss ein Metamodell erstellt und verwendet um mit anderen Werten der Inputvariablen  $\mathbf{x}$  die dazugehörigen Ausgabewerte  $\hat{y}$  zu ermitteln. Man muss sich jedoch im Klaren sein, dass es sich bei  $\hat{y}$  nicht um das wahre Ergebnis des realen Modells  $y = \mathcal{M}(x)$  handelt, sondern lediglich um eine Approximation.



**Abb. 1.1:** Metamodellierung

In Kapitel 2 dieser Arbeit wird auf die Grundlagen der Metamodelle eingegangen. Es werden Ansätze für Regression- und Interpolationsfunktionen erläutert und beschrieben wie mit den vorhandenen Daten diese aufgebaut werden können. Da es sich bei Metamodellen um eine Approximation der realen Funktion handelt, muss getestet werden, ob das Metamodell die Werte der realen Funktion korrekt wiedergeben kann. Verfahren zur Validierung der Genauigkeit der Approximation werden im anschließenden erläutert. Zuletzt wird auf das Erzeugen von Stützstellen eingegangen. An diesen Stellen muss das reale Modell ausgewertet werden, wobei mit diesen Werten das Metamodell erzeugt wird. Auf diese Verfahren ist ein besonderes Augenmerk zu legen, da durch die Wahl der Stützstellen auch die Güte des Metamodells beeinflusst wird. In Kapitel 3 wird auf Anwendungen von Metamodellen eingegangen, im Speziellen für Ingenieurprobleme. In Kapitel 4 werden Metamodelle verwendet, um die Lebensdauervorhersage der Heißgaswand einer Brennkammer zu bestimmen.



# Kapitel 2

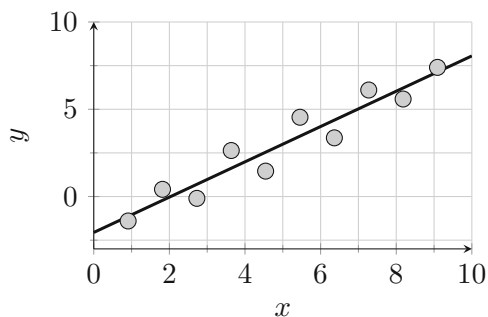
## Metamodelle

### 2.1 Metamodellierung

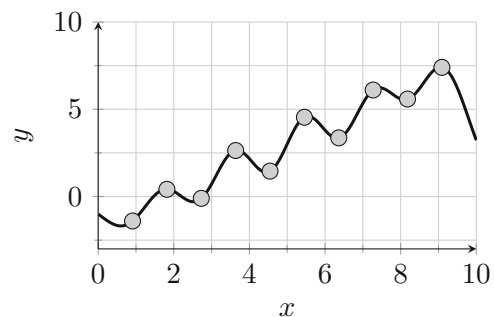
#### 2.1.1 Grundlagen

Metamodelle oder auch Antwortflächenmodelle genannt (*engl.* Response-Surface-Models) sind einfache mathematische Modelle, um Ergebnisse von (numerischen) Experimenten eines komplexeren Modells zu beschreiben. Die Antwortflächenmodelle sind Funktionen von veränderlichen Eigenschaften des komplexen Modells, z. B. können diese für mechanische Probleme, geometrische Eigenschaften, materielle Eigenschaften oder Art und Größe von Belastungen sein. Es soll jenes Modell gewählt werden, das die Daten der Experimente am besten beschreiben kann. Es werden im Grunde zwei Arten von Response-Surface-Models unterschieden, siehe Abbildung 2.1:

- Regressionsmodelle
- Interpolation Modelle



(a) Regressions Modell



(b) Interpolation für lineare Modell

**Abb. 2.1:** Arten von Response-Surface-Modellen

Nimmt man an, dass ein Modell  $n$  Variablen besitzt  $x_i; i = 1, \dots, n$ , die zum Vektor  $\mathbf{x} = [x_1 \dots x_n]^T$  zusammengefasst werden können und die Antwort mit  $z(\mathbf{x})$  gegeben ist. Die exakte Funktion soll durch eine mathematisch einfachere  $\eta(\mathbf{x})$  ersetzt werden, die jedoch den Zusammenhang zwischen den Ergebnissen  $z$  und Variablen  $\mathbf{x}$  abbilden kann. Um Ungenauigkeiten zwischen  $z(\mathbf{x})$  und  $\eta(\mathbf{x})$  auszugleichen, wird ein zufälliger Fehlerterm  $\varepsilon$  in Rechnung gestellt, der einen Erwartungswert von null besitzt. Der Zusammenhang kann wie folgt angeschrieben werden:

$$z(\mathbf{x}) = \eta(p_1, \dots, p_v; x_1, \dots, x_n) + \varepsilon = \eta(\mathbf{p}; \mathbf{x}) + \varepsilon \quad (2.1)$$

mit den Parametern  $p_j; j = 1, \dots, v$  der Approximationsfunktionen  $\eta(\cdot)$ . Als Nächstes wird der Erwartungswertoperator  $\mathbf{E}[\cdot]$  auf  $z(\mathbf{x})$  angewandt:

$$\eta(\mathbf{p}; \mathbf{x}) = \mathbf{E}[z(\mathbf{x})] \quad (2.2)$$

daraus folgt, dass der Erwartungswert von  $z$  die Funktion  $\eta$  ist. Diese wird Antwortfläche (Response-Surface) genannt [3]. Die Parameter  $\mathbf{p}$  von  $\eta$  müssen so gewählt werden, damit die Gleichung (2.2) erfüllt ist. In Abbildung 2.2 ist eine Antwortfläche für zwei Variablen ( $x_1, x_2$ ) dargestellt.

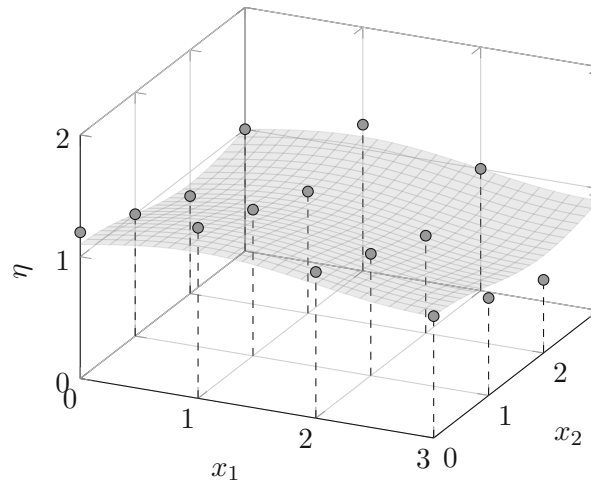


Abb. 2.2: Response-Surface und Datenpunkte

### 2.1.2 Regressionsmodelle

Wie im Abschnitt 2.1.1 angeführt werden zwei Arten von Modellierung beschrieben: Regression und Interpolation. Bei Regressionsmodellen werden die Parameter  $\mathbf{p}$  durch Minimierung der Kleinsten-Fehlerquadratsumme bestimmt:

$$S = \sum_{k=1}^N [z_k - \eta(\mathbf{x}_k; \mathbf{p})]^2 \rightarrow \min \quad (2.3)$$

wobei  $N$  die Anzahl der Zufallsexperimente ist und der aufsummierte Term den Fehler zwischen der realen Funktion  $z$  und der Antwortfläche  $\eta$  bei Experiment  $k$  angibt. Es werden meist lineare Regressionsmodelle verwendet, bei denen die Funktion  $\eta$  linear von  $\mathbf{p}$  abhängig ist. Die Eingangsvariablen  $\mathbf{x}$  können auch nichtlinear vorhanden sein. Daher kann die Antwortflächenfunktion wie folgt geschrieben werden:

$$\eta(\mathbf{x}; \mathbf{p}) = \sum_{j=1}^v f_j(\mathbf{x}) p_j = \mathbf{f}^T(\mathbf{x}) \mathbf{p} \quad (2.4)$$

wobei  $f_j(\mathbf{x})$  Funktionen sind, die im Allgemeinen von allen Eingangsvariablen  $x_i$ ;  $i = 1, \dots, n$  abhängen. Im zweiten Teil von Gleichung (2.4) ist die Vektorschreibweise angeführt. Durch Einfügen von Gleichung (2.4) in (2.3) erhält man die linear von  $\mathbf{p}$  abhängige Fehlerquadratsumme:

$$S = \sum_{k=1}^N \left[ z_k - \sum_{j=1}^v f_j(\mathbf{x}_k) p_j \right]^2 = [\mathbf{z} - \mathbf{F}\mathbf{p}]^T [\mathbf{z} - \mathbf{F}\mathbf{p}] \quad (2.5)$$

wobei  $\mathbf{z} = [z_1 \dots z_N]^T$  der Vektor mit den Ergebnissen der Zufallsexperimente und  $\mathbf{F}$  eine Matrix mit den Funktionsauswertungen  $F_{kj} = f_j(\mathbf{x}_k)$  ist. Durch Ableiten von Gleichung (2.5) nach  $\mathbf{p}$  und null setzen erhält man die Bestimmungsgleichung für  $\mathbf{p}$ :

$$\begin{aligned} \frac{\partial S}{\partial \mathbf{p}} &= (\mathbf{z} - \mathbf{F}\mathbf{p})^T (\mathbf{z} - \mathbf{F}\mathbf{p}) = 0 \\ &= -2\mathbf{F}\mathbf{z} + 2\mathbf{F}\mathbf{F}^T \mathbf{p} = 0 \end{aligned} \quad (2.6)$$

Den Ausdruck für den Parametervektor erhält man durch Auflösen nach  $\mathbf{p}$ :

$$\mathbf{p} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{z} \quad (2.7)$$

Bei Gleichung (2.7) handelt es sich um ein spezielles  $\mathbf{p}^*$ , weil durch die Wahl anderer Experimente (Eingangsvariablen  $\mathbf{x}$  und Ergebnisse  $z$ ) sich die Matrix  $\mathbf{F}$  und der Ergebnisvektor  $\mathbf{z}$  ändern [3]. Durch Anwendung des Erwartungswertoperators auf  $\mathbf{p}^*$  erhält man den Erwartungswert  $\mathbf{p}$ :

$$\mathbf{E}[\mathbf{p}^*] = \mathbf{p} \quad (2.8)$$

Durch Annahme, dass sich die Kovarianz der Ergebnisse wie folgt berechnet:

$$\mathbf{C}_{zz} = \mathbf{E}[(\mathbf{z} - \mathbf{E}[\mathbf{z}])(\mathbf{z} - \mathbf{E}[\mathbf{z}])^T] = \sigma^2 \mathbf{I} \quad (2.9)$$

wobei  $\sigma^2$  die Varianz des Fehlers  $\varepsilon$  ist, kann auch die Kovarianz des Parameters  $\mathbf{p}$  angegeben werden [5]:

$$\mathbf{C}_{pp} = \mathbf{E}[(\mathbf{p}^* - \mathbf{p})(\mathbf{p}^* - \mathbf{p})^T] = \sigma^2 (\mathbf{F}^T \mathbf{F})^{-1} \quad (2.10)$$

Mit der Kovarianzmatrix  $\mathbf{C}_{pp}$  kann die Varianzfunktion der Antwortfläche  $\eta$  wie folgt beschrieben werden:

$$\sigma_\eta^2(\mathbf{x}) = \mathbf{E}[\eta(\mathbf{x}; \mathbf{p}) - \eta(\mathbf{x}; \mathbf{p}^*)]^2 = \mathbf{f}^T(\mathbf{x}) \mathbf{C}_{pp} \mathbf{f}(\mathbf{x}) \quad (2.11)$$

Die Varianz des zu schätzenden Wertes  $z$  kann durch die Berücksichtigung des Fehlerwertes  $\varepsilon$  angegeben werden:

$$\sigma_z^2(\mathbf{x}) = \mathbf{E}[(z - \mathbf{E}[z])^2] = \sigma^2 \left[ 1 + \mathbf{f}^T(\mathbf{x}) (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{f}(\mathbf{x}) \right] \quad (2.12)$$

Mit der Varianzfunktion können Konfidenzintervalle für die zu schätzenden Werte  $z$  angegeben werden.

### 2.1.3 Polynomiale Regression

Bei der polynomialen Regression werden die Funktionen  $f_i(\mathbf{x})$  in Gleichung 2.4 durch Polynome der Eingangsvariable  $\mathbf{x}$  und deren Grad definiert. Eine Polynomfunktion vom Grad zwei und mit Eingangsvariablen  $x_i$ ;  $i = 1, \dots, n$  lässt sich wie folgt darstellen [3]:

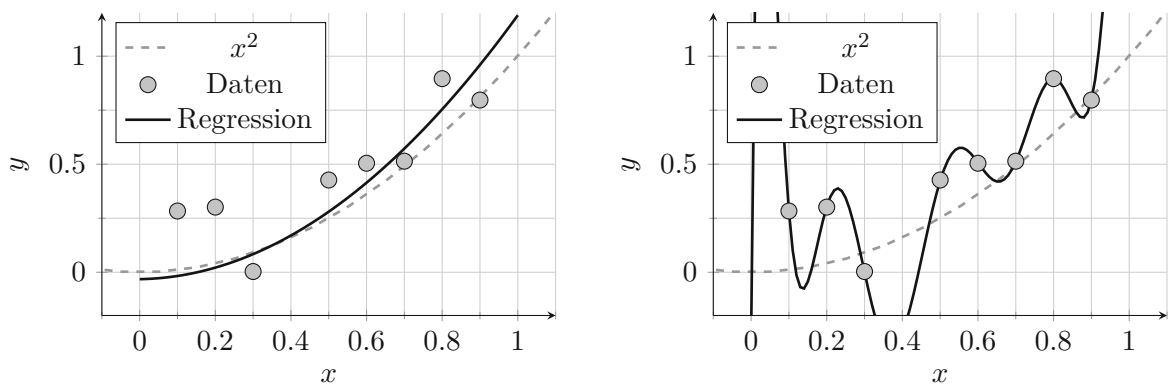
$$f(\mathbf{x}) = \beta_0 + \sum_i^n \beta_i x_i + \sum_i^n \sum_{j \leq i}^n \beta_{ij} x_i x_j \quad (2.13)$$

Gleichung (2.13) kann so umgeformt werden, dass sie der Gleichung (2.4) entspricht. Für eine Polynomfunktion vom Grad zwei und mit  $\mathbf{x} = [x_1 x_2]^T$  würde sich der Vektor  $\mathbf{f}(\mathbf{x})$  aus Gleichung (2.4) wie folgt darstellen lassen:

$$\mathbf{f}^T(\mathbf{x}) = \begin{bmatrix} 1 & x_1 & x_2 & x_1 x_2 & x_1^2 & x_2^2 \end{bmatrix} \quad (2.14)$$

und der Parametervektor hätte sechs Einträge  $\mathbf{p} = [p_1 \dots p_6]^T$ .

Bei der Polynomialregression muss erwähnt werden, dass mit zunehmendem Grad die Funktion genauer an die Ergebnisse angepasst werden kann. Jedoch kann es hierbei zu einer Überanpassung der Daten kommen. Dabei werden die Ergebnisse  $z_k$ , die zum Erstellen der Regressionsfunktion verwendet wurden, genau von der Antwortfläche wiedergegeben. Werte an anderen Stellen der Funktion sagen nichts über die zu approximierende Funktion aus, siehe Abbildung 2.3.



(a) Regressionspolynom vom Grad zwei

(b) Regressionspolynom vom Grad sechs (Überanpassung)

**Abb. 2.3:** Überanpassung von  $x^2 + \varepsilon$  mit  $\varepsilon \sim \mathcal{N}(0, 0.04)$

Für die Antwortflächenmethode werden oft Polynome vom Grad eins (Gleichung (??)) oder zwei (Gleichung (2.13)) verwendet.

$$f(\mathbf{x}) = \beta_0 + \sum_i^n \beta_i x_i \quad (2.15)$$

Es werden auch Funktionen vom Grad eins verwendet, bei denen zusätzlich die gemischten quadratischen Terme vorkommen:

$$f(\mathbf{x}) = \beta_0 + \sum_i^n \beta_i x_i + \sum_i^n \sum_{j<i}^n \beta_{ij} x_i x_j \quad (2.16)$$

Polynomfunktionen werden auch in Kombination mit anderen Funktionstypen verwendet. Hierbei bildet die Polynomfunktion den Trend/Mittelwert der zu approximierenden Funktion ab und eine darübergelegte Funktion beschreibt lokale Phänomene [3].

### 2.1.4 Weighted-Interpolation

Die zweite Art um Antwortflächen aufzubauen ist mithilfe von Interpolationsmodellen. Eine gängige Methode ist die Weighted-Interpolation. Hierbei wird die Antwortfläche über die Multiplikation von Gewichten  $w_i$  und den Ergebnissen an Stützstellen  $z_i$  gebildet:

$$\eta(\mathbf{x}) = \frac{\sum_{i=1}^N w_i z_i}{\sum_{i=1}^N w_i} \quad (2.17)$$

wobei die Summe der Gewichtungen im Nenner zur Normierung verwendet wird. Die Gewichtungen sind selbst Funktionen, die von  $\mathbf{x}$  abhängen [3] [22].

Eine gängige Funktion für  $w_i$  wäre die inverse Distanzfunktion. Dabei wird die Distanz zwischen zwei Punkten  $\mathbf{x}_j$  und  $\mathbf{x}_k$  über den euklidischen Abstand ermittelt [22]:

$$d(\mathbf{x}_j, \mathbf{x}_k) = \|\mathbf{x}_j - \mathbf{x}_k\| \quad (2.18)$$

Die Gewichtungsfunktion ergibt sich aus der Inversen des Abstands zu:

$$w(d) = 1/(d + \epsilon)^a \quad (2.19)$$

wobei  $\epsilon$  eine kleine positive Nummer zur Regularisierung ist und  $a$  ein positiver Exponent, der oft mit 2 gewählt wird [3]. Daraus ergibt sich die Interpolationsfunktion zu:

$$\eta(\mathbf{x}) = \frac{\sum_{k=1}^N w(\|\mathbf{x}_i - \mathbf{x}\|) z_i}{\sum_{k=1}^N w(\|\mathbf{x}_i - \mathbf{x}\|)} \quad (2.20)$$

Weighted-Interpolationsfunktionen können ohne großen Rechenaufwand (keine Matrixmultiplikationen oder -invertierungen) mit zusätzlichen Punkten erweitert werden. Somit lässt sich die Antwortfläche leicht verfeinern [5]. In Abbildung~2.4 wird mittels Weighted-Interpolation eine Versagenskurve approximiert.

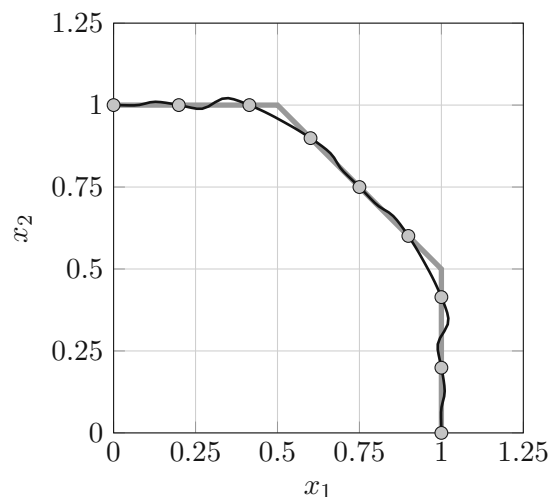


Abb. 2.4: Approximation einer Versagenskurve mittels Weighted interpolation

### 2.1.5 Moving-Least-Square Regression

Das Moving-Least-Square-Verfahren ist ähnlich zu der polynomialen Regression. Die Funktion ist wie in Gleichung (2.4) aufgebaut. Der Unterschied besteht darin, dass nicht die Fehlerquadratsumme (Gleichung (2.5)) zu minimieren ist, sondern eine gewichtete Fehlerquadratsumme [22]:

$$S(\mathbf{x}) = \sum_{k=1}^N w_k(\mathbf{x}, \mathbf{x}_k) \left[ z_k - \sum_{j=1}^v f_j(\mathbf{x}_k) p_j \right]^2 = [\mathbf{z} - \mathbf{F}\mathbf{p}]^T \mathbf{W}(\mathbf{x}) [\mathbf{z} - \mathbf{F}\mathbf{p}] \quad (2.21)$$

wobei die Matrix  $\mathbf{W}(\mathbf{x})$  eine Diagonalmatrix ist. Aus der Gleichung (2.21) ist ersichtlich, dass die Minimierung für jeden zu untersuchenden Punkt  $\mathbf{x}$  durchzuführen ist. Durch Ableiten von Gleichung (2.21) nach  $\mathbf{p}$ , Null setzen und Auflösen nach  $\mathbf{p}$  erhält man eine Bestimmungsgleichung für den Parametervektor [22]:

$$\begin{aligned} \frac{\partial S}{\partial \mathbf{p}} &= \frac{\partial [\mathbf{z} - \mathbf{F}\mathbf{p}]^T \mathbf{W}(\mathbf{x}) [\mathbf{z} - \mathbf{F}\mathbf{p}]}{\partial \mathbf{p}} = 0 \\ -2\mathbf{F}^T \mathbf{W}(\mathbf{x}) \mathbf{z} + 2\mathbf{F}^T \mathbf{W}(\mathbf{x}) \mathbf{F} \mathbf{p} &= 0 \end{aligned} \quad (2.22)$$

$$\mathbf{p} = [\mathbf{F}^T \mathbf{W}(\mathbf{x}) \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{W}(\mathbf{x}) \mathbf{z} \quad (2.23)$$

Wie man in Gleichung (2.23) sieht, ist der Parametervektor  $\mathbf{p}$  von dem Eingangsvektor  $\mathbf{x}$  abhängig, damit muss  $\mathbf{p}$  für jedes zu untersuchende  $\mathbf{x}$  neu berechnet werden. Die Antwortfläche ergibt sich somit zu:

$$\eta(\mathbf{x}) = \mathbf{f}^T(\mathbf{x}) [\mathbf{F}^T \mathbf{W}(\mathbf{x}) \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{W}(\mathbf{x}) \mathbf{z} \quad (2.24)$$

Da in Gleichung (2.24) für jeden Punkt die Matrix  $\mathbf{F}^T \mathbf{W}(\mathbf{x}) \mathbf{F}$  invertiert werden muss ist der Grad der Polynomfunktion von entscheidender Bedeutung, da er und die Anzahl der Eingangsvariablen die Größe der Matrix bestimmen [22].

Für die Wichtungsfunktion gibt es verschiedene Formen, es wird lediglich gefordert, dass die Wichtungsfunktionen monoton fallend sind [22]. Wie auch bei der Weighted-Interpolation wird der Abstand zwischen dem untersuchten Punkt  $\mathbf{x}$  und einem Referenzpunkt  $\mathbf{x}_k$  als Argument für die Wichtungsfunktion gewählt. Es können zusätzliche Parameter in den Funktionen enthalten sein. Die Gaußsche Glockenfunktion wäre eine oft verwendete Wahl für die Wichtungsfunktion [22]:

$$w(d) = e^{-\frac{1}{2} \left( \frac{d}{D} \right)^2} \quad (2.25)$$

Eine weitere Formulierung wäre:

$$w(d) = \frac{1}{1 + ad^b} \quad (2.26)$$

In Abbildung 2.5 sind Abhängigkeiten der Wichtungsfunktionen nach Gleichung (2.25) und (2.26) von ihren Parametern dargestellt. In Abbildung 2.6 und 2.7 wird die Funktion  $z(x) = x \sin(x)$  mittels Moving-Least-Square-Regression approximiert und der Einfluss des Polynomgrads sowie des Einflussradius dargestellt.

Um numerische Probleme bei der Inversion zu vermeiden, können wie bei der Weighted-Interpolation die Wichtungen regularisiert werden [22]:

$$w(d_i) = \frac{\hat{w}(d_i)}{\sum_{k=1}^N \hat{w}(d_k)} \quad (2.27)$$

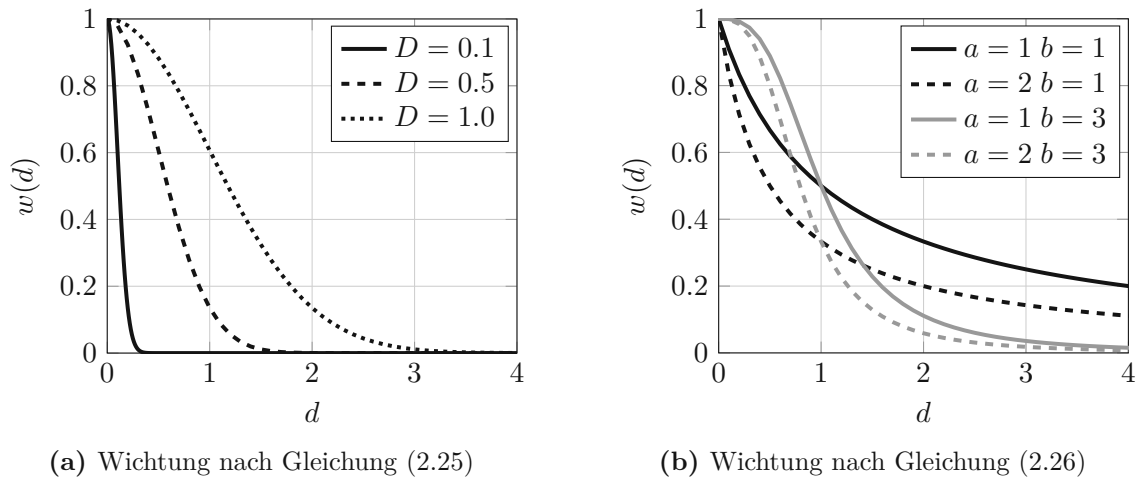


Abb. 2.5: Grafische Darstellung von Wichtungsfunktionen

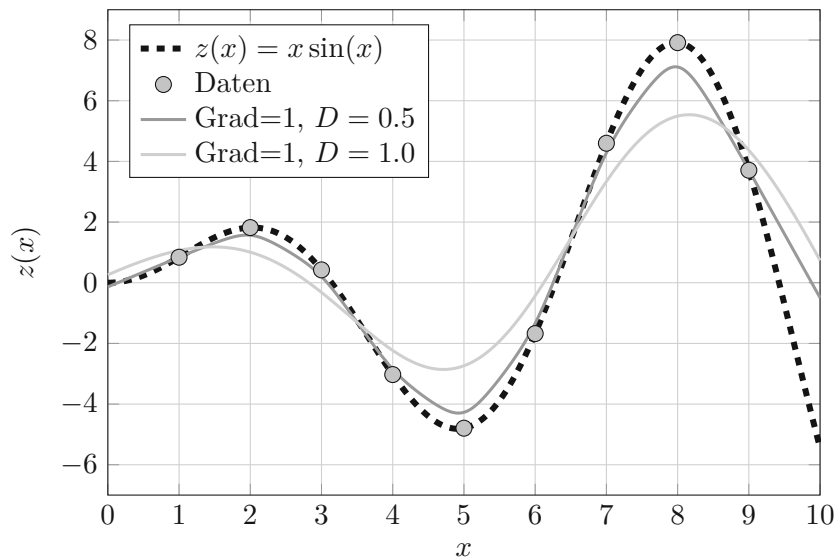
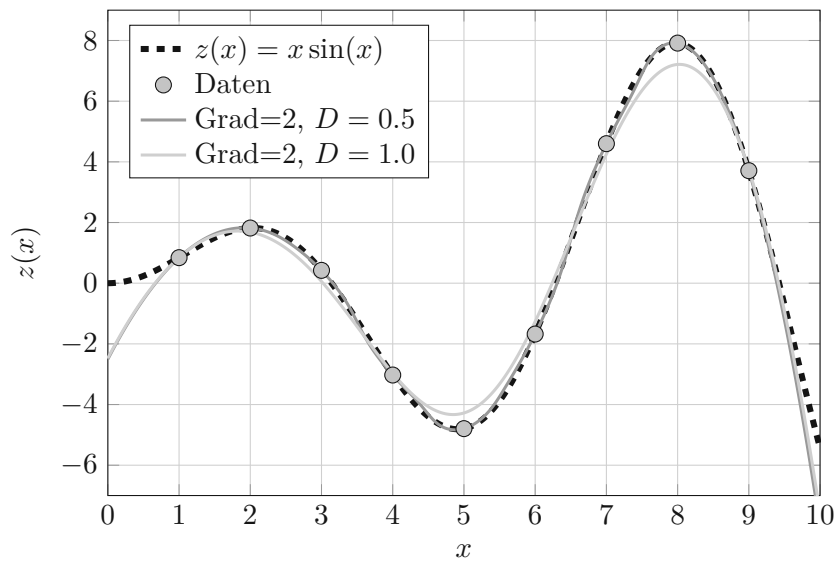


Abb. 2.6: Approximation von  $z(x) = x \sin(x)$  mit Moving-Least-Square-Regression, mit Grad 1 u. Wichtungsfunktion aus Gleichung (2.25)



**Abb. 2.7:** Approximation von  $z(x) = x \sin(x)$  mit Moving-Least-Square-Regression, mit Grad 2 u. Wichtungsfunktion aus Gleichung (2.25)

Wie in [14] beschrieben kann Gleichung (2.24) auch wie folgt geschrieben werden:

$$\eta(\mathbf{x}) = \Phi^{MLS}(\mathbf{x})\mathbf{z} \quad (2.28)$$

wobei man  $\Phi(\mathbf{x})$  als Interpolationsvektor interpretieren kann. Um ein Interpolationsverfahren zu erhalten, muss der Interpolationsvektor  $\Phi^{MLS}$  folgende Eigenschaft besitzen:

$$\Phi_i^{MLS}(\mathbf{x}_j) \approx \delta_{ij} \quad (2.29)$$

Daraus ergibt sich, dass man mit geeigneter Wahl der Wichtungsfunktion eine starke Anpassung an die Ergebnisse  $\mathbf{z}$  erzielen kann. Glättungen, wie sie bei der Regression üblich sind, treten keine mehr auf [22]. Für die Wichtungsfunktion wird in [14] folgende Funktion angegeben:

$$\hat{w}(d) = \frac{\left(\frac{d^2}{D^2} + \epsilon\right) - (1 + \epsilon)^{-2}}{\epsilon^{-2} - (1 + \epsilon)^{-2}} \quad (2.30)$$

wobei  $D$  der Einflussradius und  $\epsilon$  ein Regularisierungsparameter ist, der in [14] mit dem Wert  $10^{-5}$  angegeben ist. Es ist anzumerken, dass die Antwortfläche durch diese Wichtungsfunktion nahezu unabhängig vom Einflussradius ist.

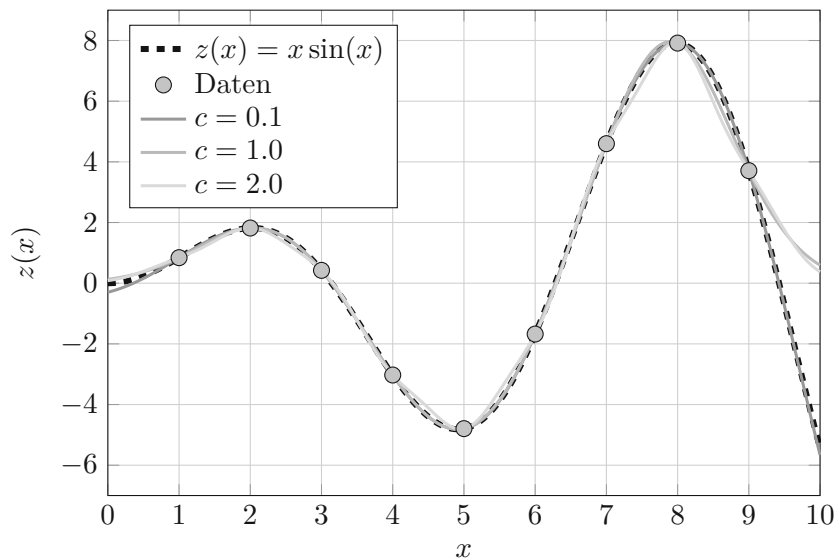
### 2.1.6 Radiale-Basis-Funktionen-Interpolation

Radiale-Basis-Funktionen-Interpolation ist eine Interpolationsmethode, bei der die Antwortfläche wie folgt aufgebaut ist [5]:

$$\eta(\mathbf{x}) = \sum_{j=1}^N \varphi_j(\|\mathbf{x} - \mathbf{x}_j\|) p_j = \boldsymbol{\varphi}^T(\mathbf{x}) \mathbf{p} \quad (2.31)$$

wobei  $\varphi_j(\cdot)$  die namensgebenden Radialen-Basis-Funktionen sind, die als Argument den euklidischen Abstand von den Punkten  $\mathbf{x}$  und  $\mathbf{x}_j$  übernehmen. Die Punkte  $\mathbf{x}_j$  sind Stützpunkte an





**Abb. 2.8:** Approximation von  $z(x) = x \sin(x)$  mit Radial-Basis-Funktion-Interpolation, mit der Radialen-Basis-Funktion aus Gleichung (2.34)

denen die Ergebnisse  $z_j$  bekannt sind. Die Radialen-Basis-Funktionen sind einfache Funktionen, die lediglich vom Abstand zweier Punkte abhängen. Im Folgenden sind einige angeführt [6].

- Lineare:

$$\varphi(r) = cr \quad (2.32)$$

- "thin plate spline":

$$\varphi(r) = cr \log(r) \quad (2.33)$$

- Gaußische Funktionen:

$$\varphi(r) = e^{-cr^2} \quad (2.34)$$

Da es sich um eine Interpolationsmethode handelt, muss es genau soviel Parameter wie Stützstellen geben. Um die Koeffizienten bestimmen zu können, wird Gleichung (2.31) mit den Ergebnissen gleichgesetzt [3]:

$$z_k = z(\mathbf{x}_k) = \sum_{j=1}^N \varphi_j(\|\mathbf{x}_k - \mathbf{x}_j\|) p_j ; \quad k = 1, \dots, N \quad (2.35)$$

und in Matrix-Vektor-Schreibweise:

$$\mathbf{z} = \mathbf{\Phi} \mathbf{p} \quad (2.36)$$

wobei die Matrix  $\mathbf{\Phi}$  die Auswertungen der Radialen-Basis-Funktionen  $\Phi_{kj} = \varphi_j(\mathbf{x}_k - \mathbf{x}_j)$  enthält. Auflösen von Gleichung (2.36) nach  $\mathbf{p}$  liefert eine Gleichung für die Koeffizienten [26] [3]:

$$\mathbf{p} = \mathbf{\Phi}^{-1} \mathbf{z} \quad (2.37)$$

In Abbildung~2.8 wird die Funktion  $z(x) = x \sin(x)$  durch die Radiale-Basis-Funktionen-Interpolation approximiert. Als Funktion wurde die gaußische Funktion gewählt und für unterschiedliche Hyperparameter  $c$  ausgewertet.

Ein wichtiger Nachteil bei Radialen-Basis-Funktionen ist, dass sie nicht in der Lage sind einfache Funktionen wie konstante oder lineare darzustellen. Eine Möglichkeit dem entgegenzuwirken, ist es, sie mit der linearen Regression zu kombinieren [3]:

$$z^{(k)} = p'_0 + \sum_{l=1}^n x_l^{(k)} p'_l + \sum_{j=1}^m \varphi(\|\mathbf{x}^{(k)} - \mathbf{x}^{(j)}\|) p_j \quad (2.38)$$

und in Matrix-Vektor-Schreibweise:

$$\mathbf{z} = \mathbf{F}\mathbf{p}' + \mathbf{\Phi}\mathbf{p} \quad (2.39)$$

Es kann zuerst das Regressionsmodell und im Anschluss das Radiale-Basis-Funktionen-Modell aus der Differenz der Daten und des Regressionsmodells ermittelt werden [3]. Eine weitere Variante ist in [1] beschrieben. Es wird gefordert, dass  $\mathbf{F}^T \mathbf{p} = \mathbf{0}$  erfüllt ist, woraus sich folgendes Gleichungssystem für die Parameter  $\mathbf{p}'$  und  $\mathbf{p}$  ergibt [1]:

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{F} \\ \mathbf{F}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{p}' \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \mathbf{0} \end{bmatrix} \quad (2.40)$$

### 2.1.7 Kriging-Verfahren

Kriging ist eine Gruppe von Verfahren, die den Wert an einem Punkt, an dem kein Ergebnis vorhanden ist, durch Interpolation von räumlich nahen Punkten, an denen Ergebnisse vorhanden sind, vorhersagen. Kriging zählt zu den linearen Schätzverfahren, da der zu schätzende Wert durch eine Linearkombination von bekannten Werten ermittelt wird. Ein Krigingmodell sieht im Allgemeinen wie folgt aus [7] [22]:

$$\eta(\mathbf{x}) = \mu(\mathbf{x}) + Y(\mathbf{x}) + \epsilon \quad (2.41)$$

wobei  $\mu(\mathbf{x})$  eine Mittelwertfunktion und  $Y(\mathbf{x})$  ein stochastischer Gaußscher Prozess ist. Je nach Verfahren werden Annahmen für die Mittelwertfunktion getroffen. Der zu schätzende Wert wird durch Linearkombination der bekannten Werte ermittelt [20] [22]:

$$\eta(\mathbf{x}) = \sum_{i=1}^N \lambda_i z(\mathbf{x}_i) = \boldsymbol{\lambda}^T \mathbf{z} \quad (2.42)$$

wobei  $\mathbf{z} = [z(\mathbf{x})_1, \dots, z(\mathbf{x})_N]^T$  bekannte Werte an den Stellen  $\mathbf{x}_i$   $i = 1, \dots, n$  sind und  $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_N]^T$  die zu bestimmenden Gewichtungen. Die Gewichtungsfaktoren sind vom Punkt  $\mathbf{x}$  abhängig an dem der Wert zu schätzen ist. Eine Annahme des Kriging-Schätzers ist, dass er erwartungstreu ist [20] [22]:

$$\mathbf{E}[\eta(\mathbf{x})] = \mathbf{E}[z(\mathbf{x})] \quad (2.43)$$

wobei  $z(\mathbf{x})$  der wahre Wert an der Stelle  $\mathbf{x}$  ist. Die Wichtungen werden durch die Minimierung der Fehlervarianz zwischen dem wahren und geschätzten Wert ermittelt [20]:

$$\mathbf{E} \left[ \left( z(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{z} \right)^2 \right] \rightarrow \min \quad (2.44)$$

Für das Kriging-Verfahren wird auch eine Annahme über die Kovarianz benötigt. Diese wird meist über das Variogramm  $\gamma$  beschrieben [22] [7]:

$$2\gamma(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{Var}((z(\mathbf{x}_i) - z(\mathbf{x}_j))) \quad (2.45)$$

wobei  $\mathbf{Var}(\cdot)$  die Varianz ist. Durch Annahme, dass es sich bei der gesuchten Funktion um einen stationären Prozess ( $\mathbf{E}[z(\mathbf{x})] = m$  und  $\mathbf{Cov}(z(\mathbf{x}_i), z(\mathbf{x}_j)) = \mathbf{Cov}(h)$ ) handelt, kann auch das Variogramm lediglich durch den Abstand  $\mathbf{h}$  der Punkte  $\mathbf{x}_i$  und  $\mathbf{x}_j$  angeschrieben werden:

$$2\gamma(\mathbf{h}) = \mathbf{Var}(z(\mathbf{x}_i + \mathbf{h}) - z(\mathbf{x}_j)) \quad (2.46)$$

Das empirische Variogramm wird über folgende Gleichung berechnet [22] [7]:

$$\gamma(\mathbf{h}) = \frac{1}{2N(\mathbf{h})} \sum_{N(\mathbf{h})} (z(\mathbf{x}_i) - z(\mathbf{x}_j)) \quad (2.47)$$

Dazu werden die Abstände der Datenpunkt  $\mathbf{x}_i$  zueinander und die Differenz ihrer Ergebnisse  $z(\mathbf{x}_i)$  bestimmt. Im Anschluss werden die Abstände in Gruppen unterteilt und für jede Gruppe der Variogrammwert (Gleichung 2.47) berechnet. Die so berechneten Punkte werden an einem zu wählenden Variogrammansatz angepasst. Die meisten Ansätze werden über drei Parameter definiert, Nugget, Still und Range. Das Nugget  $c_0$  beschreibt die Varianz am Punkt der Beobachtung. In Anwendungen ohne Rauschen muss das Nugget den Wert null haben, damit die Gleichung (2.46) erfüllt ist. Still  $c$  ist der Schwellwert den das Variogramm maximal erreichen kann. Range  $r$  gibt den Einflussbereich des Variogramms an [22] [24]. Die gebräuchlichsten Ansätze sind in der folgenden Aufzählung angegeben [22]:

- Lineares Modell

$$\gamma(h) = \begin{cases} c_0 + \frac{c-c_0}{r}h & \text{für } 0 \leq h \leq r \\ c & \text{für } r \leq h \end{cases} \quad (2.48)$$

- Sphärisches Modell

$$\gamma(h) = \begin{cases} c_0 + (c - c_0) \left( \frac{3h}{2r} - \frac{h^3}{2r^3} \right) & \text{für } 0 \leq h \leq r \\ c & \text{für } r \leq h \end{cases} \quad (2.49)$$

- Exponentielles Modell

$$\gamma(h) = c_0 + (c - c_0) \left( 1 - e^{-\frac{h}{r}} \right) \quad (2.50)$$

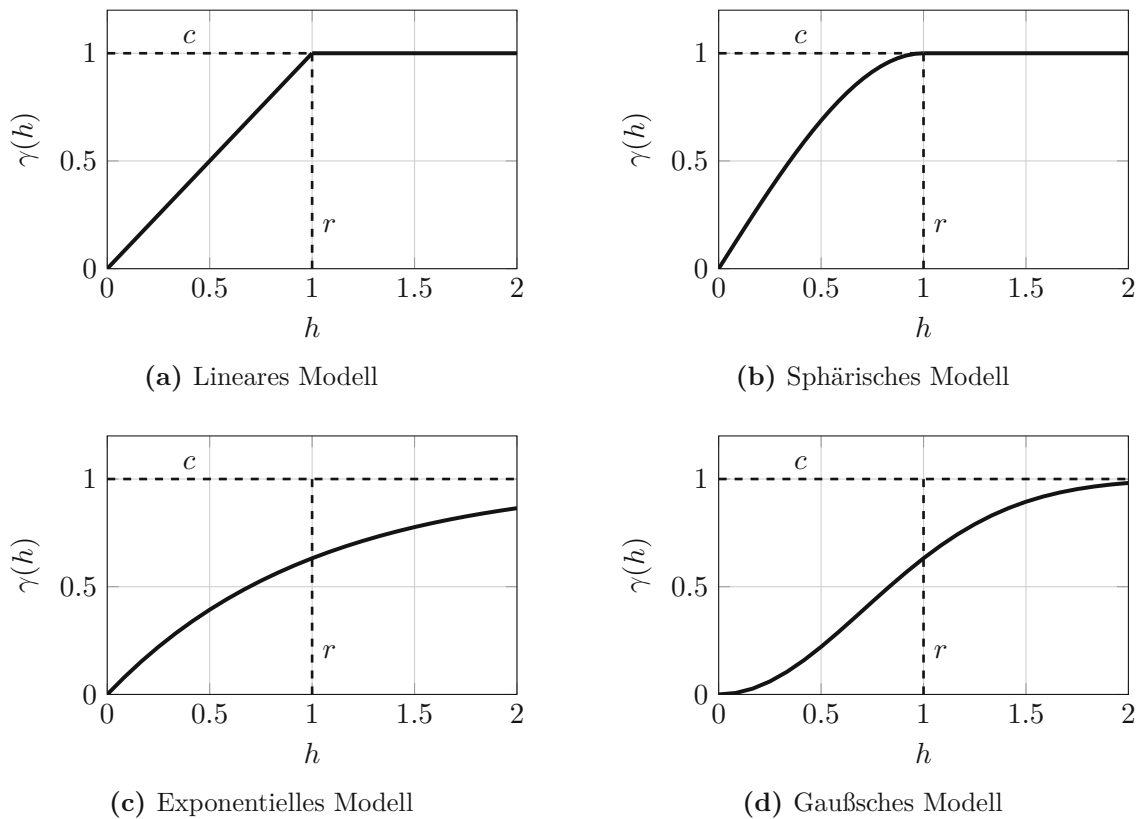
- Gaußsches Modell

$$\gamma(h) = c_0 + (c - c_0) \left( 1 - e^{-\left(\frac{h}{r}\right)^2} \right) \quad (2.51)$$

In Abbildung 2.9 sind die aufgezählten Variogramansätze grafisch dargestellt.

Das Variogramm kann auch in Verbindung mit der Kovarianz gebracht werden, dazu wird Gleichung 2.45 aufgelöst [7]:

$$\begin{aligned} 2\gamma(\mathbf{x}_i, \mathbf{x}_j) &= \mathbf{Var}(z(\mathbf{x}_i) - z(\mathbf{x}_j)) = \\ &= \mathbf{E} \left[ ((z_i - z_j) - (\mathbf{E}[z_i] - \mathbf{E}[z_j]))^2 \right] = \\ &= \mathbf{E} \left[ ((z_i - \mathbf{E}[z_i]) - (z_j - \mathbf{E}[z_j]))^2 \right] = \\ &= \mathbf{E} \left[ (z_i - \mathbf{E}[z_i])^2 - 2(z_i - \mathbf{E}[z_i])(z_j - \mathbf{E}[z_j]) + (z_j - \mathbf{E}[z_j])^2 \right] = \\ &= \mathbf{Var}(z_i, z_i) - 2\mathbf{Cov}(z_i, z_j) + \mathbf{Var}(z_j, z_j) \end{aligned} \quad (2.52)$$



**Abb. 2.9:** Ansätze für das Variogram ohne Nugget-Effekt

Wenn noch berücksichtigt wird, dass es sich bei  $z(\mathbf{x})$  um einen stationären Prozess handelt, bei dem die Kovarianz nur durch den Abstand  $\mathbf{h}$  der betrachteten Punkte bestimmt ist, ergibt sich der Zusammenhang zu:

$$2\gamma(\mathbf{h}) = 2[\mathbf{Cov}(0) - \mathbf{Cov}(\mathbf{h})] \quad (2.53)$$

Im Folgenden wird auf das Simple- und das Ordinary-Kriging näher eingegangen. In der Literatur finden sich noch weitere Kriging-Verfahren, die vor allem in der Geostatistik Anwendungen finden [7].

**Simple-Kriging:** Beim Simple-Kriging wird angenommen, dass eine Mittelwertfunktion  $\mu(\mathbf{x})$  vorhanden und bekannt ist [7]. Sie kann z.B. durch ein Regressionsmodell (siehe Abschnitt 2.1.2) ermittelt werden. Mithilfe der Mittelwertfunktion kann die Differenz  $y(\mathbf{x}) = z(\mathbf{x}) - \mu(\mathbf{x})$  an jedem Punkt bestimmt werden. Beim Simple-Kriging wird nun die Differenz  $y(\mathbf{x})$  approximiert:

$$y(\mathbf{x}) = [z(\mathbf{x}) - \mu(\mathbf{x})] = \sum_{i=1}^N \lambda_i [z(\mathbf{x}_i) - \mu(\mathbf{x}_i)] = \boldsymbol{\lambda}^T \mathbf{y} \quad (2.54)$$

Das Simple-Kriging kann auch so interpretiert werden, dass der Mittelwert gleich null ist, das kann durch Anwenden von Gleichung (2.43) auf (2.54) erkannt werden. Durch Einsetzen von Gleichung (2.54) in (2.44) erhält man die Bestimmungsgleichung für die Gewichtungen  $\lambda_i$ :

$$\begin{aligned} \mathbf{E} \left[ \left( y(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{y} \right)^2 \right] &= \mathbf{E} \left[ y(\mathbf{x})^2 - 2\boldsymbol{\lambda}^T \mathbf{y} y(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{y} \mathbf{y}^T \boldsymbol{\lambda} \right] = \\ &= \mathbf{Var} (y(\mathbf{x})) - 2\boldsymbol{\lambda}^T \mathbf{Cov} (\mathbf{y}, y(\mathbf{x})) + \boldsymbol{\lambda}^T \mathbf{Cov} (\mathbf{y}, \mathbf{y}) \boldsymbol{\lambda} = \\ &= \sigma^2 - 2\boldsymbol{\lambda}^T \mathbf{c}_{\mathbf{y},y} + \boldsymbol{\lambda}^T \mathbf{C}_{\mathbf{y},\mathbf{y}} \boldsymbol{\lambda} \end{aligned} \quad (2.55)$$

wobei  $\mathbf{Var} (y(\mathbf{x}))$  die Varianz am Punkt  $\mathbf{x}$ , der Vektor  $\mathbf{Cov} (\mathbf{y}, y(\mathbf{x}))$  (mit  $c_i = \mathbf{Cov} (y(\mathbf{x}_i), y(\mathbf{x}))$ ) die Kovarianz zwischen den bekannten Ergebnissen  $\mathbf{y}$  und den unbekanntem Wert  $y(\mathbf{x})$  ist. Die Matrix  $\mathbf{Cov} (\mathbf{y}, \mathbf{y})$  (mit  $C_{ij} = \mathbf{Cov} (y(\mathbf{x}_i), y(\mathbf{x}_j))$ ) ist die Kovarianz zwischen den bekannten Ergebnissen. Durch Ableiten von Gleichung (2.55) nach  $\boldsymbol{\lambda}^T$  und Umformen erhält man die Gleichung für den Vektor der Gewichtungen  $\boldsymbol{\lambda}$ :

$$\boldsymbol{\lambda} = \mathbf{C}_{\mathbf{y},\mathbf{y}}^{-1} \mathbf{c}_{\mathbf{y},y} \quad (2.56)$$

Durch Einsetzen in Gleichung (2.54) erhält man die Antwortfläche:

$$y(\mathbf{x}) = \mathbf{c}_{\mathbf{y},y}^T \mathbf{C}_{\mathbf{y},\mathbf{y}}^{-1} [\mathbf{z} - \boldsymbol{\mu}] \quad (2.57)$$

und durch Einsetzen in Gleichung (2.55) erhält man die Varianz:

$$\sigma_{SK}^2(\mathbf{x}) = \sigma^2 - \mathbf{c}_{\mathbf{y},y}^T \mathbf{C}_{\mathbf{y},\mathbf{y}}^{-1} \mathbf{c}_{\mathbf{y},y} \quad (2.58)$$

**Ordinary-Kriging:** Beim Ordinary-Kriging wird die Mittelwertfunktion als konstant  $\mu(\mathbf{x}) = \mu$  und unbekannt angenommen [7]. Die angesetzte Funktion hat die gleiche Gestalt wie in Gleichung (2.42). Der Erwartungswert der realen Funktion  $z(\mathbf{x})$  und der Antwortfläche  $\eta(\mathbf{x})$  muss gleich dem Mittelwert  $\mu$  sein, daraus ergibt sich:

$$\mu = \mathbf{E} [z(\mathbf{x})] = \mathbf{E} \left[ \sum_{i=1}^N \lambda_i z(\mathbf{x}_i) \right] = \sum_{i=1}^N \lambda_i \mathbf{E} [z(\mathbf{x}_i)] = \sum_{i=1}^N \lambda_i \mu = \mu \quad (2.59)$$

Aus Gleichung (2.59) wird ersichtlich, dass die Summe der Gewichtungen gleich 1 ergeben muss [7]:

$$\sum_{i=1}^N \lambda_i = 1 \quad (2.60)$$

Durch Berechnung der Varianz des Fehlers zwischen  $z(\mathbf{x})$  und  $\eta(\mathbf{x})$  kann die Bestimmungsgleichung für  $\boldsymbol{\lambda}^T$  gefunden werden:

$$\begin{aligned} \mathbf{Var} (z(\mathbf{x}) - \eta(\mathbf{x})) &= \mathbf{E} \left[ \left( z(\mathbf{x}) - \eta(\mathbf{x}) - \mathbf{E} [z(\mathbf{x})] + \mathbf{E} [\eta(\mathbf{x})] \right)^2 \right] = \\ &= \mathbf{E} \left[ \left( z(\mathbf{x}) - \eta(\mathbf{x}) - \mu + \mu \right)^2 \right] = \\ &= \mathbf{E} \left[ z^2(\mathbf{x}) - 2 \sum_{i=1}^N \lambda_i z(\mathbf{x}_i) z(\mathbf{x}) + \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j z(\mathbf{x}_i) z(\mathbf{x}_j) \right] \end{aligned} \quad (2.61)$$

Durch die Addition mit null:

$$0 = \sum_{i=1}^N \lambda_i z^2(\mathbf{x}_i) - \sum_{i=1}^N \lambda_i \frac{z^2(\mathbf{x}_i)}{2} \underbrace{\sum_{j=1}^N \lambda_j}_{=1} - \sum_{i=1}^N \lambda_i \underbrace{\sum_{j=1}^N \lambda_j}_{=1} \frac{z^2(\mathbf{x}_j)}{2} \quad (2.62)$$

und der Definition des Variogramms (Gleichung (2.45)) wird Gleichung (2.61) zu [7]:

$$\begin{aligned} & \mathbf{E} \left[ 2 \sum_{i=1}^N \lambda_i \frac{(z(\mathbf{x}_i) - z(\mathbf{x}))^2}{2} - \sum_{i=i}^N \sum_{i=i}^N \lambda_i \lambda_j \frac{(z(\mathbf{x}_i) - z(\mathbf{x}_j))^2}{2} \right] = \\ & = 2 \sum_{i=1}^N \lambda_i \frac{\mathbf{E} [(z(\mathbf{x}_i) - z(\mathbf{x}))^2]}{2} - \sum_{i=i}^N \sum_{i=i}^N \lambda_i \lambda_j \frac{\mathbf{E} [(z(\mathbf{x}_i) - z(\mathbf{x}_j))^2]}{2} = \\ & = 2 \sum_{i=1}^N \lambda_i \gamma(z(\mathbf{x}_i), z(\mathbf{x})) - \sum_{i=i}^N \sum_{i=i}^N \lambda_i \lambda_j \gamma(z(\mathbf{x}_i), z(\mathbf{x}_j)) = \\ & = 2\boldsymbol{\lambda}\boldsymbol{\gamma} - \boldsymbol{\lambda}^T \boldsymbol{\Gamma} \boldsymbol{\lambda} \end{aligned} \quad (2.63)$$

Um  $\boldsymbol{\lambda}$  ermitteln zu können, muss folgendes Minimierungsproblem gelöst werden:

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & 2\boldsymbol{\lambda}\boldsymbol{\gamma} - \boldsymbol{\lambda}^T \boldsymbol{\Gamma} \boldsymbol{\lambda} \\ \text{s.t.} \quad & \boldsymbol{\lambda}^T \mathbf{1} - 1 = 0 \end{aligned} \quad (2.64)$$

wobei  $\mathbf{1}$  ein Vektor mit nur Einsen ist. Das Minimierungsproblem wird mithilfe der Lagrange-Multiplikator-Methode gelöst. Hierzu wird die Lagrange-Funktion  $\varphi(\boldsymbol{\lambda}, \omega)$  aufgestellt:

$$\varphi(\boldsymbol{\lambda}, \omega) = 2\boldsymbol{\lambda}\boldsymbol{\gamma} - \boldsymbol{\lambda}^T \boldsymbol{\Gamma} \boldsymbol{\lambda} + 2\omega (\boldsymbol{\lambda}^T \mathbf{1} - 1) \quad (2.65)$$

wobei  $\omega$  der Lagrange-Multiplikator ist. Anschließend wird nach  $\boldsymbol{\lambda}$  und  $\omega$  abgeleitet und man erhält ein Gleichungssystem für die Unbekannten:

$$\begin{aligned} \frac{\partial \varphi(\boldsymbol{\lambda}, \omega)}{\partial \boldsymbol{\lambda}} &= 2\boldsymbol{\gamma} - 2\boldsymbol{\Gamma} \boldsymbol{\lambda} + 2\omega \mathbf{1} = 0 \\ \frac{\partial \varphi(\boldsymbol{\lambda}, \omega)}{\partial \omega} &= \mathbf{1}^T \boldsymbol{\lambda} - 1 = 0 \end{aligned} \quad (2.66)$$

Durch Auflösen des Gleichungssystems (2.66) erhält man Ausdrücke für  $\boldsymbol{\lambda}$  und  $\omega$  [7]:

$$\boldsymbol{\lambda} = \boldsymbol{\Gamma}^{-1} \left( \boldsymbol{\gamma} + \mathbf{1} \frac{1 - \mathbf{1}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}}{\mathbf{1}^T \boldsymbol{\Gamma}^{-1} \mathbf{1}} \right) \quad (2.67)$$

$$\omega = \frac{1 - \mathbf{1}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}}{\mathbf{1}^T \boldsymbol{\Gamma}^{-1} \mathbf{1}} \quad (2.68)$$

Durch Einsetzen von Gleichung (2.67) in (2.41) ergibt sich die Antwortfläche zu:

$$\eta(\mathbf{x}) = \left( \boldsymbol{\gamma} + \mathbf{1} \frac{1 - \mathbf{1}^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}}{\mathbf{1}^T \boldsymbol{\Gamma}^{-1} \mathbf{1}} \right)^T \boldsymbol{\Gamma}^{-1} \mathbf{z} \quad (2.69)$$

und durch Einsetzen in (2.63) erhält man die Kriging-Varianz:

$$\sigma_{OK}^2 = \boldsymbol{\gamma}^T \boldsymbol{\Gamma} \boldsymbol{\gamma} - \frac{(1 - \mathbf{1}^T \boldsymbol{\Gamma} \boldsymbol{\gamma})^2}{\mathbf{1}^T \boldsymbol{\Gamma} \mathbf{1}} \quad (2.70)$$

### 2.1.8 Gauß-Prozess-Regression

Die Gauß-Prozess-Regression ist im Grunde dieselbe Methode wie Kriging, jedoch mit einer anderen Herangehensweise. Ein Gauß-Prozess ist eine Verallgemeinerung einer gaußverteilten Zufallsvariable auf Funktionen. Ein Gauß-Prozess ist durch eine Erwartungswertfunktion  $\mu(\mathbf{x})$  und eine Kovarianzfunktion  $C(\mathbf{x}_j, \mathbf{x}_j)$  definiert [9].

Da bei realen Aufgaben endliche Stützstellen einer Funktion betrachtet werden, kann ein Gauß-Prozess in diesem Fall auch als mehrdimensionale Normalverteilung mit Erwartungswertvektor  $\boldsymbol{\mu} = [\mu(\mathbf{x}_1), \dots, \mu(\mathbf{x}_n)]^T$  und Kovarianzmatrix  $\mathbf{C}_{zz}$  angegeben werden:

$$p(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{C})}} \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\mathbf{z} - \boldsymbol{\mu})\right) \quad (2.71)$$

wobei  $d$  die Dimension der Normalverteilung angibt.

Nimmt man an die gesammelten Informationen aus Experimenten  $(\mathbf{x}_i, z_i)$  folgt der Verteilung aus Gleichung (2.71) und man will für einen Punkt  $\mathbf{x}$  seinen Funktionswert  $z(\mathbf{x})$  vorhersagen, dann kann man die Verteilung für  $z_{N+1} = [z(\mathbf{x}), \mathbf{z}_N^T]^T$  mit Gleichung (2.71) beschreiben, wobei  $\mathbf{z}_N = [z_1, \dots, z_n]^T$  ist. Die Kovarianzmatrix wird zu:

$$\mathbf{C} = \begin{pmatrix} \sigma^2 & \mathbf{c}_{zz} \\ \mathbf{c}_{zz}^T & \mathbf{C}_{zz} \end{pmatrix} \quad (2.72)$$

wobei  $\sigma^2$  die Varianz am Punkt  $\mathbf{x}$  und  $\mathbf{c}_{zz}$  der Kovarianzvektor ( $c_i = \mathbf{Cov}(z(\mathbf{x}), z(\mathbf{x}_i))$ ) zwischen dem neuen Punkt  $\mathbf{x}$  und den Stützstellen  $\mathbf{x}_i$   $i = 1, \dots, n$  ist. Die Inverse der Kovarianzmatrix kann laut [9] wie folgt angeschrieben werden:

$$\mathbf{C}^{-1} = \begin{pmatrix} \kappa & \mathbf{m}_{zz} \\ \mathbf{m}_{zz}^T & \mathbf{M}_{zz} \end{pmatrix} \quad (2.73)$$

Durch die Identität  $\mathbf{C}\mathbf{C}^{-1} = \mathbf{I}$  können  $\kappa$ ,  $\mathbf{m}_{zz}$  und  $\mathbf{M}_{zz}$  wie folgt angegeben werden:

$$\kappa = \left(\sigma^2 - \mathbf{c}_{zz}^T \mathbf{C}_{zz}^{-1} \mathbf{c}_{zz}\right)^{-1} \quad (2.74)$$

$$\mathbf{m}_{zz} = \frac{-\mathbf{C}_{zz}^{-1} \mathbf{c}_{zz}}{\sigma^2 - \mathbf{c}_{zz}^T \mathbf{C}_{zz}^{-1} \mathbf{c}_{zz}} \quad (2.75)$$

$$\mathbf{M}_{zz} = \mathbf{C}_{zz}^{-1} + \frac{\mathbf{C}_{zz}^{-1} \mathbf{c}_{zz} \mathbf{c}_{zz}^T \mathbf{C}_{zz}^{-1}}{\sigma^2 - \mathbf{c}_{zz}^T \mathbf{C}_{zz}^{-1} \mathbf{c}_{zz}} \quad (2.76)$$

Um die Wahrscheinlichkeit für den Wert  $z(\mathbf{x})$  ermitteln zu können, benötigt man die bedingte Wahrscheinlichkeitsverteilung  $p(z(\mathbf{x})|\mathbf{z})$  zufolge der Informationen aus Experimenten  $(\mathbf{x}_i, z_i)$ . Die bedingte Wahrscheinlichkeitsverteilung kann wie folgt ermittelt werden:

$$p(z(\mathbf{x})|\mathbf{z}) = \frac{p(z(\mathbf{x}), \mathbf{z})}{p(\mathbf{z})} \quad (2.77)$$

Daraus ergibt sich die Wahrscheinlichkeitsverteilung von  $p(z(\mathbf{x})|\mathbf{z})$  zu [9] [24]:

$$p(z(\mathbf{x})|\mathbf{z}) = \frac{Z_N}{Z_{N+1}} \exp \left[ -\frac{1}{2} \mathbf{y}_{N+1}^T \mathbf{C}_{N+1}^{-1} \mathbf{y}_{N+1} - \mathbf{y}_N^T \mathbf{C}_N^{-1} \mathbf{y}_N \right] \quad (2.78)$$

wobei  $\mathbf{y}_{N+1}$  eine Abkürzung für  $(\mathbf{z}_{N+1} - \boldsymbol{\mu}_{N+1})$  und  $\mathbf{y}_N$  eine für  $(\mathbf{z}_N - \boldsymbol{\mu}_N)$  ist.  $Z_N$  und  $Z_{N+1}$  sind ebenfalls Abkürzungen für die Terme vor der Exponentialfunktion  $\exp(\cdot)$  in Gleichung (2.71). Gleichung (2.78) kann in folgende Gestalt gebracht werden:

$$p(z(\mathbf{x})|\mathbf{z}) = \frac{1}{Z} \exp \left( -\frac{(z(\mathbf{x}) - \bar{z}(\mathbf{x}))^2}{2\sigma^2(\mathbf{x})} \right) \quad (2.79)$$

wobei  $\bar{z}(\mathbf{x})$  der Erwartungswert an der Stelle  $\mathbf{x}$  und  $\sigma^2(\mathbf{x})$  die Varianz ist. Diese Werte ergeben sich zu:

$$\bar{z}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) + \mathbf{c}_{zz}^T \mathbf{C}_{zz}^{-1} (\mathbf{z} - \boldsymbol{\mu}) \quad (2.80)$$

$$\sigma^2(\mathbf{x}) = \sigma^2 - \mathbf{c}_{zz}^T \mathbf{C}_{zz}^{-1} \mathbf{c}_{\eta z} \quad (2.81)$$

$\bar{z}(\mathbf{x})$  ist im Falle der Gauß-Prozess-Regression die Antwortfläche  $\eta(\mathbf{x})$ . Durch Vergleichen mit Gleichung (2.57) und (2.58) erkennt man, dass Simple-Kriging und Gauß-Prozess-Regression formal zu denselben Ergebnissen kommen [9] [24].

Wie bei den Kriging-Methoden müssen auch bei der Gauß-Prozess-Regression die Hyperparameter der Kovarianzfunktion an die Daten angepasst werden. Hierzu wird die Log-Likelihood-Methode angewandt. Die Likelihood-Funktion  $L_z(\boldsymbol{\theta})$  der Daten  $(\mathbf{x}_i, z_i)$  ist formal identisch zur Mehrdimensionalen-Normalverteilung:

$$L_z(\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^p \det(\mathbf{C}_{zz})}} \exp \left( -\frac{1}{2} \mathbf{y}^T \mathbf{C}_{zz}^{-1} \mathbf{y} \right) \quad (2.82)$$

wobei  $\boldsymbol{\theta}$  die Parameter der Kovarianzfunktion sind und  $\mathbf{y}$  als Abkürzung für  $\mathbf{z} - \boldsymbol{\mu}$  verwendet wurde. Die Log-Likelihood-Funktion ergibt sich zu:

$$\mathcal{L} = \frac{1}{2} \log(\det(\mathbf{C}_{z,z})) - \frac{1}{2} \mathbf{y}^T \mathbf{C}_{zz}^{-1} \mathbf{y} - \frac{N}{2} \log(2\pi) \quad (2.83)$$

wobei  $N$  die Anzahl der Daten  $(\mathbf{x}_i, z_i)$  ist. Ableiten der Log-Likelihood-Funktion nach dem Parameter  $\boldsymbol{\theta}$  ergibt:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{1}{2} \text{tr} \left( \mathbf{C}_{zz}^{-1} \frac{\partial \mathbf{C}_{zz}}{\partial \boldsymbol{\theta}} \right) + \frac{1}{2} \mathbf{y}^T \mathbf{C}_{zz}^{-1} \frac{\partial \mathbf{C}_{zz}}{\partial \boldsymbol{\theta}} \mathbf{C}_{zz}^{-1} \mathbf{y} \quad (2.84)$$

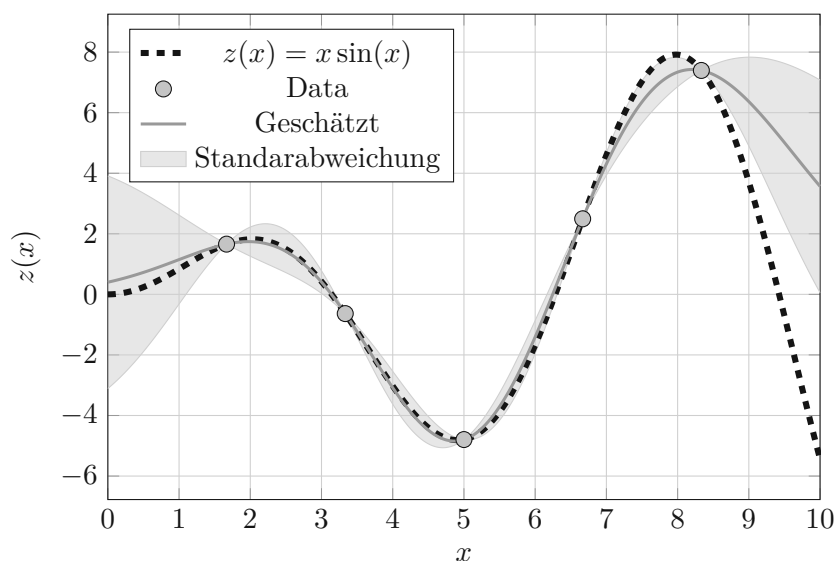
Die optimalen Kovarianzparameter müssen im Anschluss durch ein Optimierungsverfahren ermittelt werden. In diesem Zusammenhang muss noch erwähnt werden, dass die Log-Likelihood-Funktion im Allgemeinen mehrere lokale Maxima besitzt und manche dieser zu unbrauchbaren Regressionskurven führen [9].

In Abbildung 2.10 wird die Funktion  $z(x) = x \sin(x)$  mit Gauß-Prozess-Regression approximiert. Es ist der geschätzte Funktionsverlauf sowie der Bereich der Standardabweichung dargestellt.

## 2.1.9 Künstliche neuronale Netzwerke (Artificial Neural Networks ANN)

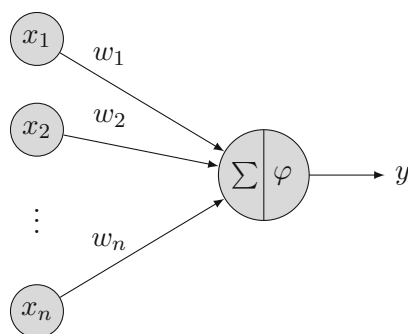
Künstliche neuronale Netzwerke sind durch Knoten und Verbindungen zwischen diesen aufgebaut. Die Verbindungen werden mit Wichtungen assoziiert, sie sind jene Parameter, die trainiert werden müssen, damit das Netzwerk eine Funktion approximieren kann [5]. Eine der einfachsten Foren





**Abb. 2.10:** Approximation von  $z(x) = x \sin(x)$  mit Gauß-Prozess-Regression

eines künstlichen neuronalen Netzwerkes ist in Abbildung 2.11 dargestellt. Hierbei handelt es sich



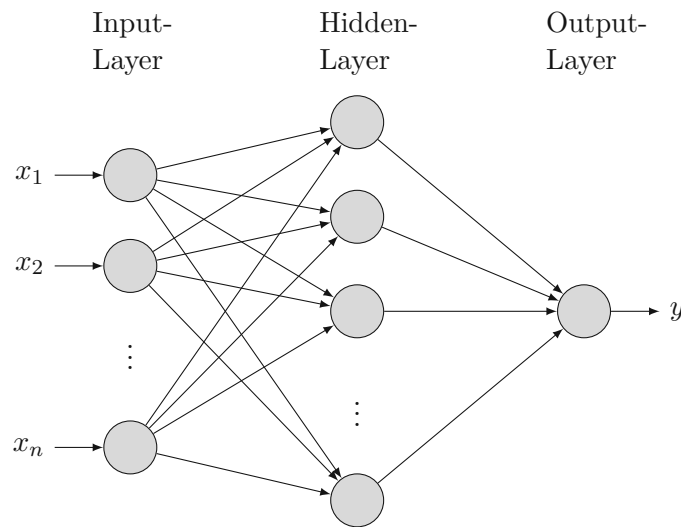
**Abb. 2.11:** Künstliches Neuron mit den Inputs  $x_i$ , den Wichtungen  $w_i$ , der Summenfunktion  $\Sigma$ , der Aktivierungsfunktion  $\varphi$  und dem Output  $y$

um ein einziges Neuron mit  $n$  Inputs. Die Inputs werden durch die Verbindungen zum Neuron geleitet und mit den Wichtungen versehen. Im Neuron selbst werden die einzelnen gewichteten Inputs über die Summenfunktion summiert, wobei hier noch ein Bias addiert werden kann, und im Anschluss durch die Aktivitätsfunktion geleitet. Der Output des Neurons wird durch die Aktivitätsfunktion bestimmt. Diese können linear und nichtlinear sein, wobei durch nichtlineare Aktivitätsfunktionen die künstlichen neuronalen Netzwerke ihre Mächtigkeit erhalten [2].

Es können auch mehrere Neuronen verwendet werden, die ihrerseits wieder den Input für die nächsten Neuronen liefern und somit ein Netzwerk erzeugen, wie in Abbildung 2.12 dargestellt [10] [2].

Mathematisch wird die Ausgabe eines künstlichen Neurons wie folgt beschrieben:

$$y_j = \varphi \left( \sum_{i=1}^{N_i} w_{ji} x_i + b_j \right) \quad (2.85)$$



**Abb. 2.12:** Künstliches neuronales Netzwerk mit drei Layer, Input-, Hidden- und Output-Layer

wobei  $y_j$  der Output des Knotens  $j$  in einem Layer,  $w_{ji}$  die Wichtung zwischen Input  $i$  und Output  $j$ ,  $b_j$  ein Bias im Knoten und  $N_i$  die Anzahl der Inputvariablen ist. Bei mehreren Layern ist  $y_j$  wiederum die Inputvariable für den nächsten Layer:

$$z_k = \varphi \left( \sum_{j=1}^{N_j} w_{kj} y_j + b_k \right) \quad (2.86)$$

Dieses Schema kann sich über mehrere Layer erstrecken. In der Literatur wird bei Netzwerken mit vielen Zwischenlayern auch von tiefen Netzwerken (Deep Networks) oder tiefen Lernen (Deep Learning) gesprochen [10] [2].

In der Literatur können häufig benutzte Aktivierungsfunktionen gefunden werden [10] [2]. Im Folgenden sind einige Aktivierungsfunktionen angeführt und in Abbildung 2.13 dargestellt:

- Linear

$$\varphi(x) = x \quad (2.87)$$

- Logistische

$$\varphi(x) = \frac{1}{1 + \exp(-x)} \quad (2.88)$$

- Rectifier

$$\varphi(x) = \max(0, x) \quad (2.89)$$

Damit ein künstliches neuronales Netzwerk Funktionen approximieren kann, muss es trainiert werden. Dies wird in vielen Fällen mit dem Backpropagation-Verfahren durchgeführt [10] [2]. Das Backpropagation-Verfahren optimiert die Wichtungen  $w$  damit eine Fehlerfunktion minimal wird. Die Fehlerfunktion lautet wie folgt:

$$S = \frac{1}{2} \left( z_i - z_i^{ANN} \right)^2 \quad (2.90)$$

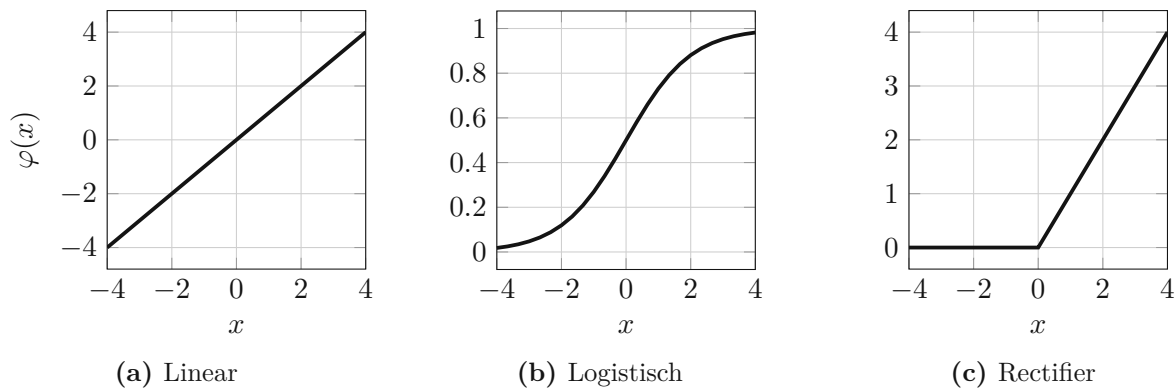


Abb. 2.13: Aktivitätsfunktionen

wobei  $z_i^{ANN}$  das Ergebnis des Netzwerks mit den Inputs  $\mathbf{x}_i$  ist. Ein neuronales Netzwerk kann im Allgemeinen auch mehrere Outputs haben, die zu einem Vektor zusammengefasst werden können. In diesem Fall nimmt die Fehlerfunktion die Form der Fehlerquadratsumme an:

$$S = \frac{1}{2}(\mathbf{z}_i - \mathbf{z}_i^{ANN})^T(\mathbf{z}_i - \mathbf{z}_i^{ANN}) \quad (2.91)$$

Die Wichtungen  $w$  sollen nun so gewählt werden, damit die Fehlerfunktion  $S$  ein Minimum einnimmt. Die Bestimmung der Wichtungen wird mithilfe des stochastischen-Gradientenverfahren (*engl.* Stochastic-Gradient-Descent-Methode) vorgenommen [2]. Hierbei wird zufällig ein Datenpunkt  $(\mathbf{x}_i, z_i)$  gewählt und mit diesem das Ergebnis des neuronalen Netzwerks  $z_i^{ANN}$  und die Fehlerfunktion  $S$  ermittelt. Im Anschluss werden die Wichtungen nach folgendem Schema aktualisiert:

$$w_{l,ij}^{k+1} = w_{l,ij}^k - \alpha \frac{\partial S}{\partial w_{l,ij}^k} \quad (2.92)$$

wobei der Index  $l$  die Layernummer, der Index  $ij$  die Verbindung von Input  $j$  zu Neuron  $i$ , der Superskript  $k$  die Iterationsnummer und  $\alpha$  die Lernrate angibt. Der Gradient der Fehlerfunktion  $S$  kann sehr leicht über die Kettenregel ermittelt werden [10] [2]. Daraus wurde ein rekursives Verfahren entwickelt, um den Gradienten zu berechnen [24]:

$$\frac{\partial S}{\partial w_{l,ij}} = \delta_j y_i \quad (2.93)$$

mit

$$\delta_j = \frac{\partial S}{\partial y_j} \frac{\partial y_j}{\partial n_j} = \begin{cases} (z_j - z_j^{ANN}) \frac{d\varphi(n_j)}{dn_j} & \text{falls } j \text{ ein Output Neuron ist} \\ (\sum_l w_{jl} \delta_l) \frac{d\varphi(n_j)}{dn_j} & \text{falls } j \text{ ein verdecktes Neuron ist} \end{cases} \quad (2.94)$$

wobei  $n_j$  der Input für die Aktivitätsfunktion  $\sum_j w_{ij} y_i$  ist.

Die Herausforderung bei künstlichen neuronalen Netzwerken liegt darin, die richtige Anzahl an Zwischenlayern (Hidden-Layer) und Neuronen zu finden, um weder eine Unteranpassung (Underfitting) noch Überanpassung (Overfitting) zu erhalten. Wie bei der linearen Regression kann dies durch Validierung verschiedener Architekturen von neuronalen Netzwerken geschehen [24].

## 2.2 Validierung

Die Einsetzbarkeit von Metamodellen hängt davon ab, wie gut das Modell die wahren Werte approximieren kann. Um die Einsetzbarkeit gewährleisten zu können, muss das Metamodell validiert werden. Dazu gibt es etliche Verfahren. Im folgenden Abschnitt werden einige Verfahren vorgestellt.

### 2.2.1 Fehlermaße

Fehlermaße dienen dazu, den Fehler zwischen Werten des wahren Modells und dem Metamodell zu bestimmen. Der Fehler an einem Punkt  $\mathbf{x}_i$  wird durch die Differenz zwischen dem realen und approximierten Modellwert definiert:

$$\varepsilon(\mathbf{x}) = z(\mathbf{x}) - z^*(\mathbf{x}) \quad (2.95)$$

Der Fehler wird oft quadriert, dadurch fallen größere Fehler stärker ins Gewicht als kleinere.

$$e(\mathbf{x}) = \varepsilon^2(\mathbf{x}) = (z(\mathbf{x}_i) - z^*(\mathbf{x}_i))^2 \quad (2.96)$$

Zum Aufbau von Metamodellen wird oft der quadrierte Fehler herangezogen, wie z. B. im Abschnitt 2.1.2. Es werden noch weitere Verlustfunktionen für die Validierung verwendet [22]~[18]:

- Laplace:

$$e(\mathbf{x}) = |\varepsilon(\mathbf{x})| \quad (2.97)$$

- Huber:

$$e(\mathbf{x}) = \begin{cases} \varepsilon^2(\mathbf{x}) & \text{für } |\varepsilon(\mathbf{x})| < \epsilon \\ 2|\varepsilon(\mathbf{x})|\epsilon - \epsilon^2 & \text{für } |\varepsilon(\mathbf{x})| \geq \epsilon \end{cases} \quad (2.98)$$

- $\epsilon$ -sensitiv::

$$e(\mathbf{x}) = \begin{cases} 0 & \text{für } |\varepsilon(\mathbf{x})| < \epsilon \\ |\varepsilon(\mathbf{x})| - \epsilon & \text{für } |\varepsilon(\mathbf{x})| \geq \epsilon \end{cases} \quad (2.99)$$

Bei der Laplace-Verlustfunktion werden alle Fehler linear gewichtet. Durch die Huber-Verlustfunktion wird in einem Bereich von  $\epsilon$  der Fehler quadratisch und außerhalb linear gewichtet. Die  $\epsilon$ -sensitiv-Funktion lässt Fehler in einem Bereich von  $\epsilon$  zu und jene, die darüberliegen, werden linear gewichtet.

Die oben genannten Fehlermaße beschreiben den Fehler lediglich lokal an einem Punkt  $\mathbf{x}$ , man ist jedoch auch an globalen Maßen interessiert. Eine einfache Möglichkeit ist es den maximalen Fehler des Modells heranzuziehen [22]:

$$\max(e(\mathbf{x})) \quad (2.100)$$

Eine weitere Möglichkeit wäre es, den Fehler gewichtet zu integrieren [22]:

$$\int e(\mathbf{x})w(\mathbf{x})d\mathbf{x} \quad (2.101)$$

wobei  $w(\mathbf{x})$  eine Wichtungsfunktion ist. Da diese Integration in vielen Fällen einfach nicht möglich ist, wird stattdessen die Summe über  $N$  Punkte, an denen Ergebnisse des wahren Modells vorhanden ist, gebildet [22]:

$$\sum_i^N e(\mathbf{x}_i)w(\mathbf{x}_i) \quad (2.102)$$

### 2.2.1.1 Determinationskoeffizient $R^2$

Der Determinationskoeffizient oder auch Bestimmtheitsmaß ist eine weitere Möglichkeit, die Genauigkeit eines Metamodells zu bestimmen. Er ist ein statistisches Maß, welches die Korrelation zwischen dem realen Modellwert und dem approximierten Werten angibt. Der Determinationskoeffizient ist wie folgt definiert [3] [4]:

$$R^2 = \left( \frac{\mathbf{Cov}(Z, Z^*)}{\sigma_Z \sigma_{Z^*}} \right)^2 = \left( \frac{\mathbf{E} \left[ (Z - \bar{Z}) (Z^* - \bar{Z}^*) \right]}{\sigma_Z \sigma_{Z^*}} \right)^2 \quad (2.103)$$

wobei  $\bar{Z}$ ,  $\bar{Z}^*$ ,  $\sigma_Z$  und  $\sigma_{Z^*}$  die Mittelwerte und die Standardabweichungen der beobachteten und approximierten Modellwerte sind. Bei Metamodellen mit einer hohen Genauigkeit ist der Determinationskoeffizient nahe an eins.

Im Falle von Regressionsmodellen und im Speziellen der polynomialen Regression, geht das Bestimmtheitsmaß gegen eins, wenn die Anzahl der Parameter nahe der Anzahl der beobachteten Werte ist, weil es durch die höhere Anzahl an Parametern zu einer Überanpassung kommt. Aus diesem Grund wurde das adjustierte Bestimmtheitsmaß eingeführt:

$$R_{adj}^2 = R^2 - \frac{k-1}{N-k} (1 - R^2) \quad (2.104)$$

wobei  $k$  die Anzahl der Parameter und  $N$  die Anzahl beobachteter Werte ist. Wenn nun der Parameter  $k$  nahe an die Anzahl der Beobachtung kommt, bewertet das adjustierte Bestimmtheitsmaß das Metamodell schlechter [3].

Für Interpolationsmodelle kann der Determinationskoeffizient in der oben dargestellten Form nicht angewandt werden, da das Modell die beobachteten Daten perfekt approximiert. Für diese Modelle müssen andere Verfahren zur Validierung angewandt werden. In den folgenden Abschnitten werden Verfahren vorgestellt, die auch für Interpolationsmodelle angewendet werden können [18].

### 2.2.2 Split-Sampling

Beim Split-Sampling werden die beobachteten Werte in eine Trainings- und Testmenge unterteilt. Mit der Trainingsmenge wird das Metamodell aufgebaut und mit der Testmenge wird die Validierung durchgeführt. Interpolationsmodelle können somit auch mit den oben erwähnten Fehlermassen analysiert werden. Der Fehler eines Trainingspunkts  $i$  wird über die Differenz des wahren Werts  $z_i$  und des vorhergesagten Werts an dieser Stelle  $z_i^*$  definiert [18]:

$$\epsilon_i = z_i - z_i^* \quad (2.105)$$

Dem Prozess des Teilens der Menge kommt eine große Bedeutung zu, da es einen großen Einfluss auf das Ergebnis hat [22].

### 2.2.3 Kreuzvalidierung

Bei der Kreuzvalidierung wird die Menge der beobachteten Werte in  $k$  Untermengen geteilt, die alle ungefähr gleich viele Datenpunkte enthalten sollen. Im nächsten Schritt wird eine Untermenge entfernt und aus den übrigen  $k - 1$  Mengen das Metamodell gebildet. An den Punkten der entfernten Menge werden die Fehler berechnet. Dieses Prozedere wird nun für jede Teilmenge durchgeführt, damit jede Untermenge  $k - 1$ -mal für den Modellaufbau und einmal für die Validierung verwendet wird [24].

Ein spezieller Fall ist die Leave-One-Out-Kreuzvalidierung bei der  $k$  gleich der Anzahl der Punkte in der Menge ist. Es wird hier nur ein Element aus der Menge entfernt, mit den restlichen wird das Metamodell aufgebaut und mit den entfernten Elementen der Fehler bestimmt [24].

#### 2.2.3.1 Prognosekoeffizient

In [15] wird der Prognosekoeffizient (engl. Coefficient of Prognosis  $CoP$ ) als Maß, welches unabhängig vom gewählten Modell ist, vorgeschlagen. Er ist wie folgt definiert:

$$CoP = \left( \frac{\mathbf{E} \left[ (Z_{test} - \bar{Z}_{test})(Z_{test}^* - \bar{Z}_{test}^*) \right]}{\sigma_{Z_{test}} \sigma_{Z_{test}^*}} \right)^2 \quad (2.106)$$

wobei  $Z_{test}$  die beobachteten Werte aus einer Testmenge und  $Z_{test}^*$  deren Schätzwerte zufolge des Metamodells sind. Der Prognosekoeffizient ist eine Verallgemeinerung des Determinationskoeffizienten aus Abschnitt 2.2.1.1. Er nimmt Werte von 0 bis 1 an, wobei 0 ein schlechtes Modell und 1 ein gutes beschreibt. Der Prognosekoeffizient wird oft mithilfe der Leave-one-out-Kreuzvalidierung berechnet. Ebenso kann er als Optimierungsziel für die Ermittlung von Hyperparametern, wie sie in Abschnitt 2.1.6 oder 2.1.8 vorkommen, verwendet werden.

In [15] und [16] wird der Prognosekoeffizient dazu benutzt, um ein Metamodell mit der besten Prognosefähigkeit (engl. Metamodel of Optimal Prognosis MOP) zu erstellen. Hierbei wird der  $CoP$  für verschiedene Metamodelle mit unterschiedlicher Anzahl an Eingangsvariablen verglichen und jenes gewählt, welches den besten  $CoP$  besitzt.

## 2.3 Statistische Versuchsplanung (Design of Experiments)

Um ein Metamodell aufbauen zu können, werden Informationen über die wahre Funktion benötigt. Um diese zu erhalten, wird zuerst ein Plan aufgestellt, an welchen Punkten eine Systemantwort zu ermitteln ist. Es wird angestrebt, aus so wenig Punkten wie möglich die höchstmögliche Information zu ziehen. Es ist daher wichtig, dass die gewählten Stützpunkte die wichtigsten Zusammenhänge darstellen können. In vielen Fällen wird die statistische Versuchsplanung (engl. design of experiments (DOE)) angewendet. Die Anfänge nahm das DOE in der Landwirtschaft und hat sich durch seine Vorzüge in andere Disziplinen ausgebreitet [8].

Es wird empfohlen die Versuche in einem dimensionslosen Raum zu planen. Der Versuch wird um einen zentralen Punkt, der von Interesse sein könnte, z. B. dem Mittelwert eines zu untersuchenden Bereichs oder dem Mittelwert einer Zufallsvariablen, bestimmt. In welcher

Distanz die Versuchspunkte liegen, ausgehend vom zentralen Punkt, wird über eine Variable gesteuert [3]. Ein Versuchspunkt ergibt sich damit zu:

$$\mathbf{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} = \begin{pmatrix} c_1 + \xi_1^{(k)} s_1 \\ c_2 + \xi_2^{(k)} s_2 \\ \vdots \\ c_n + \xi_n^{(k)} s_n \end{pmatrix} \quad (2.107)$$

In Abhängigkeit von der Anzahl an erzeugten Punkten  $N$  und der Anzahl an Koeffizienten des Metamodells  $v$ , spricht man von übersättigten ( $N > v$ ), gesättigten ( $N = v$ ) oder untersättigten ( $N < v$ ) Versuchsplänen. Für den Aufbau von Metamodellen wird ein gesättigter oder übersättigter Versuchsplan benötigt. Gesättigte kommen zur Anwendung, wenn mittels linearer Regression lineare oder quadratische Modelle erzeugt werden sollen oder bei Interpolationsverfahren. Übersättigte kommen bei verrauschten Daten zum Einsatz, dadurch werden die Daten geglättet [22].

### 2.3.1 Vollfaktorieller Versuchsplan

Eine der einfachsten Methoden ist es ein Gitternetz aus Punkten aufzuspannen. Hierfür wird jede Raumrichtung in eine feste Anzahl an Punkten unterteilt und jede Kombination aus diesen ist ein Versuch. Dieses Vorgehen wird als vollfaktorieller Versuchsplan bezeichnet. Die Inputvariablen  $x_1, \dots, x_n$  werden als Faktoren bezeichnet und die Anzahl der zu untersuchenden Punkte sind die Stufen. In Abbildung 2.14 sind die Versuchspunkte bei einer 2-stufigen 2-Faktoren bzw. 3-Faktoren Versuchsplan dargestellt. Bei 2-Stufen sitzen die Stützstellen in den Eckpunkten des zu untersuchenden Bereichs. In Abbildung 2.15 sind hingegen die Versuchspunkte bei einem 3-Stufen 2-Faktoren bzw. 3-Faktoren Versuchsplan dargestellt. Hier sind zusätzlich noch Zwischenpunkte vorhanden. Im Allgemeinen ergibt sich die Anzahl der Versuchspunkte zu:

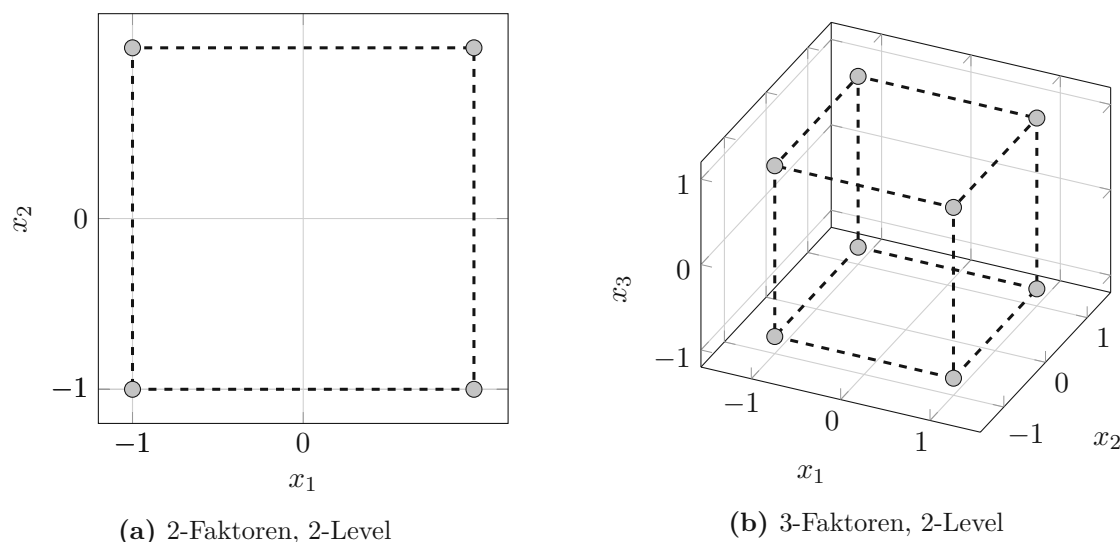
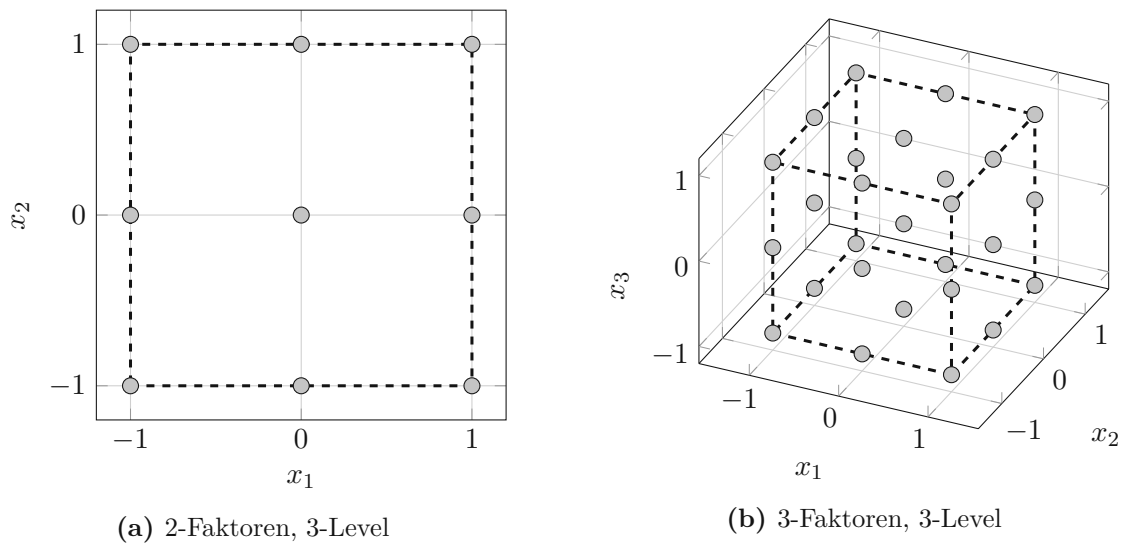


Abb. 2.14: Vollfaktorieller Versuchsplan mit 2-Levels

$$N = l^n \quad (2.108)$$



**Abb. 2.15:** Vollfaktorieller Versuchsplan mit 3-Levels

wobei  $l$  die Stufenanzahl und  $n$  die Faktorenanzahl ist. Für 2- bzw. 3-Stufen ergibt sich:

$$N = 2^n \quad N = 3^n \quad (2.109)$$

Die 2- und 3-Stufen Designs sind auch die am meist genutzten, da mit ihnen lineare bzw. quadratische Zusammenhänge abgebildet werden können. Wie man erkennen kann, beeinflusst die Anzahl der Inputvariablen die Anzahl an benötigten Versuchspunkten sehr stark. Daher sollte diese Methode eher für Systeme mit einer kleinen Anzahl an Inputparametern gewählt werden [22].

### 2.3.2 Central Composite Designs

Das Central-Composite-Design setzt sich aus einem Zwei-Stufen Versuchsplan, einem Axialen-Design und dem Zentralpunkt zusammen. Bei einem Axialen-Design werden lediglich 2 Versuchspunkte in jede Achsrichtung aufgestellt. In Abbildung 2.16 sind Central Composite Designs für zwei und drei Variablen dargestellt. Die Anzahl der generierten Versuche in Abhängigkeit der Variablenanzahl ist [22]:

$$N = 2^n + 2n + 1 \quad (2.110)$$

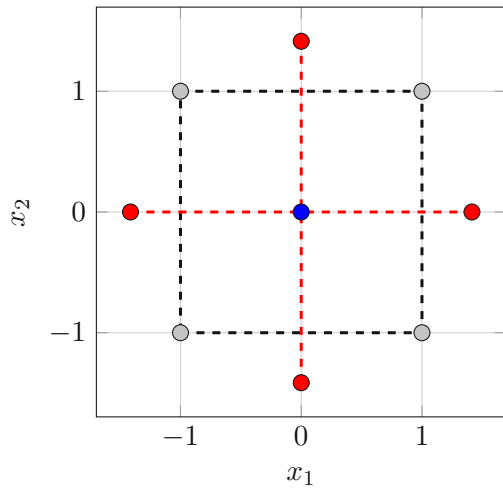
### 2.3.3 Teilfaktorieller Versuchsplan

Da bei vollfaktoriellen Versuchsplänen die Anzahl der Versuche bei Zunahme der Faktoren stark ansteigt, werden stattdessen teilfaktorielle Versuchspläne benutzt. Hierbei wird die Versuchsanzahl durch eine Verringerung der Faktoren  $p$  bestimmt:

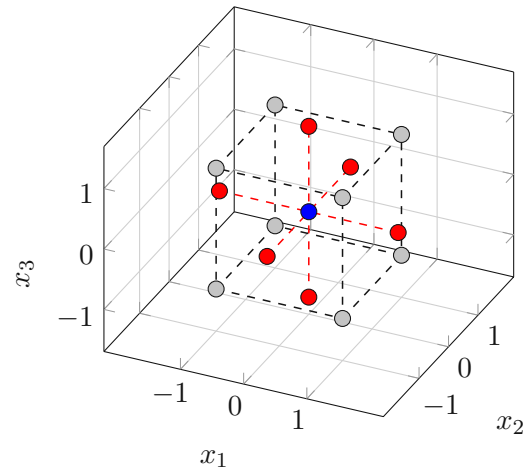
$$N = l^{n-p} \quad (2.111)$$

Für einen 2-Stufen-Versuch mit 3 Faktoren würde man eine Anzahl von  $2^{3-1} = 4$  erhalten. Mögliche Versuchspunkte sind in Abbildung 2.17 dargestellt. In der Literatur sind verschiedene Schemata angeführt, um die Versuchspunkte zu konstruieren [24].



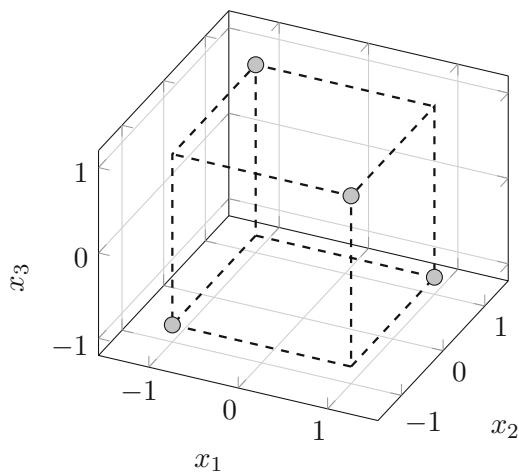


(a) Design für zwei Variablen

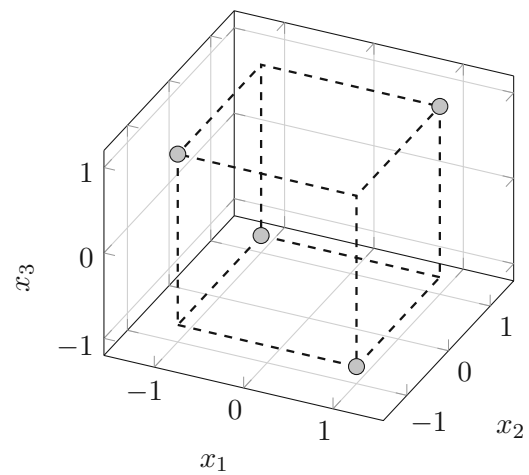


(b) Design für zwei Variablen

**Abb. 2.16:** Central Composite Designs: Zentraler Punkt (blau), Axiales Design (rot), Zwei Level Faktorielle Design (grau)



(a) 3-Faktoren, 2-Level



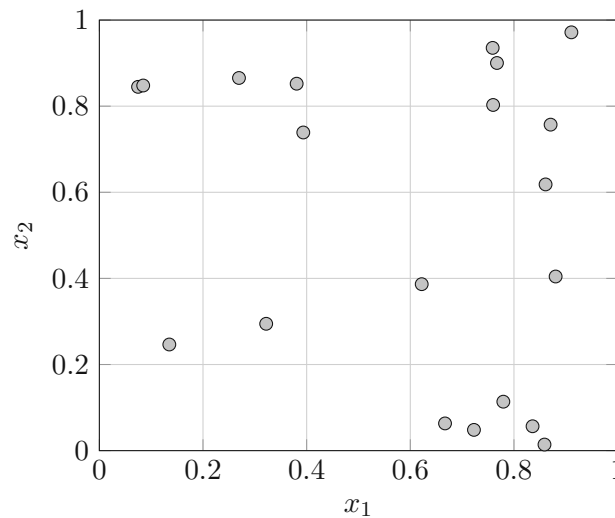
(b) 3-Faktoren, 2-Level

**Abb. 2.17:** Teilfaktorieller Versuchsplan

Der Grundgedanke einer teilfaktoriellen Versuchsplanung liegt darin, Terme höherer Ordnung zu vernachlässigen, z. B. kann man bei einem  $2^3$  vollfaktoriellen Versuchsplan ein Regressionsmodell mit acht Koeffizienten erzeugen, wobei ein Koeffizient die Konstante, drei Koeffizienten die Haupteffekte, drei Koeffizienten die zweifache Wechselwirkung und ein Koeffizient die dreifache Wechselwirkung beschreibt. Mit einem teilfaktoriellen Versuchsplan von  $2^{3-1}$  kann ein Modell mit 4 Koeffizienten beschrieben werden, wobei hier lediglich der konstante Effekt und die drei Haupteffekte beschrieben werden können [24].

### 2.3.4 Monte-Carlo-Methode

Die Monte-Carlo-Methode ist eine oft genutzte Methode. Bei ihr werden durch einen Zufalls-generator  $N$  Stichproben für die Inputvariablen generiert. Für deterministische Variablen wird eine Gleichverteilung mit einem unteren und oberen Grenzwert angenommen. Ebenso kann für stochastische Variablen eine Verteilungsfunktion für die Generierung mit den entsprechenden stochastischen Parametern herangezogen werden. Der Vorteil der Methode ist ihre Einfachheit, das leichte Hinzufügen von neuen Punkten und die leichte Programmierbarkeit. Der größte Nachteil liegt darin, dass für aussagekräftige Ergebnisse oft sehr viele Stützstellen benötigt werden. Ein weiterer Punkt ist, dass der Variablenraum unzureichend abgedeckt ist oder es zu Korrelationen zwischen den erzeugten Inputvariablen kommt [22] [24]. In Abbildung 2.18 ist ein Beispiel für die Generierung von Datenpunkten im 2-dimensionalen Raum mithilfe der Monte-Carlo-Methode dargestellt.



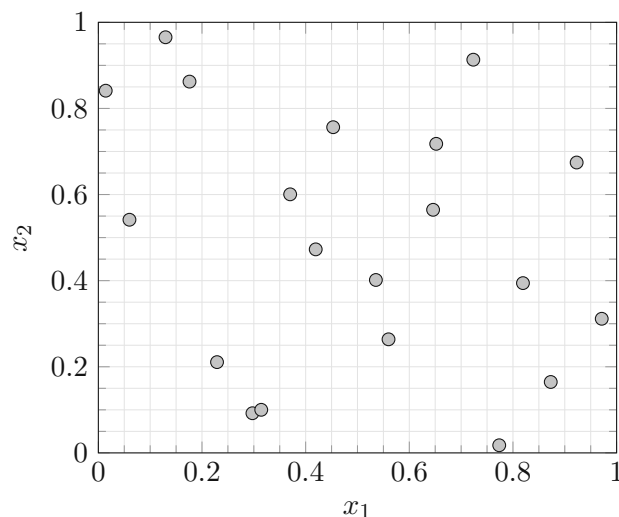
**Abb. 2.18:** 2D Monte-Carlo-Sampling mit 20 Punkten im Bereich  $[0, 1] \times [0, 1]$

Um die Probleme der Monte-Carlo-Methode zu kompensieren, sind Quasi-Monte-Carlo-Methoden entwickelt worden. Hierbei wird versucht eine bessere Abdeckung im Variablenraum und eine geringere Anzahl an Stützstellen zu erhalten. Bei diesen Methoden werden die Stützstellenpunkte jedoch nicht mehr durch einen Zufallsgenerator, sondern durch deterministische Methoden ermittelt. Einige der zur Generierung angewandten Algorithmen werden in [24] beschrieben.

### 2.3.5 Latin-Hypercube-Sampling

Ein weiteres Verfahren, welches versucht, die Nachteile der Monte-Carlo-Methode auszugleichen, ist das Latin-Hypercube-Sampling. Bei dieser Methode werden  $N$  Stichproben für  $n$  Variablen

ermittelt. Dazu wird zuerst der Bereich der Variablen in  $N$  Abschnitte mit gleicher Größe unterteilt. Danach wird in jedem Intervall ein Wert gewählt. Im Anschluss müssen die Werte der Variablen kombiniert werden, um die Stützstellen zu erhalten. Es können dadurch  $N^n$  Kombinationen erzeugt werden, wobei einige schlechte und andere gute Eigenschaften besitzen. Ein Kriterium zur Optimierung des Stützstellensets ist, dass sie annähernd unkorreliert sein sollen. Andere Optimierungsziele können die Maximierung der minimalen Distanz oder das Minimieren der maximalen Distanz sein [22]. In Abbildung 2.19 ist ein Beispiel für die Generierung von Datenpunkten im 2-dimensionalen Raum mittels Latin-Hypercube-Sampling dargestellt.



**Abb. 2.19:** 2D Latin-Hypercube-Sampling mit 20 Punkten im Bereich  $[0, 1] \times [0, 1]$

# Kapitel 3

## Anwendungen

### 3.1 Sensitivitätsanalyse

Bei der Sensitivitätsanalyse wird der Einfluss der Inputvariablen auf die Outputvariablen bestimmt. Dieser kann benutzt werden, um wichtige Eingangsvariablen zu erkennen, die im Anschluss näher untersucht werden. Auf weniger wichtige Parameter wird im Anschluss ein geringerer Wert gelegt. Im Bezug auf die Metamodellierung können so Parameter, die die Systemantwort nur wenig beeinflussen, als konstant betrachtet werden und können als Inputvariable aus dem Metamodell entfernt werden. Dies kann auch zu einer Erhöhung der Approximationsgenauigkeit führen, da unwichtige Variablen einen Überanpassungseffekt erzeugen [4]. In den folgenden Abschnitten werden Verfahren zur Sensitivitätsanalyse vorgestellt, bei denen Metamodelle genutzt werden.

#### 3.1.1 Bedeutungskoeffizient

Eine Möglichkeit, die Sensitivität von Inputvariablen zu bestimmen, ist über den Determinations- oder Prognosekoeffizienten, wie sie in Abschnitt 2.2.1.1 und 2.2.3.1 beschrieben wurden. In [3] wird mithilfe des Determinationskoeffizienten ein Maß für die Bedeutung der Variablen ermittelt. Hierbei wird zuerst für das volle Modell der Determinationskoeffizient berechnet. Im Anschluss wird immer eine Variable aus dem Modell entfernt und erneut der Determinationskoeffizient, des verringerten Modells,  $R_e^2$  berechnet. Der Unterschied zwischen diesen beiden Koeffizienten führt zur folgenden Differenz:

$$\Delta_e = R^2 - R_e^2 \quad (3.1)$$

Diese Differenz wird für jede Eingangsvariable berechnet. Als Nächstes werden die Differenzen mit dem Determinationskoeffizienten des vollen Modells normalisiert:

$$I_e = \frac{\Delta_e}{R^2} \quad (3.2)$$

wobei  $I_e$  ein Maß für die Bedeutung (*engl.* Importance) der entfernten Variable ist. In [4] wird ein ähnliches Verfahren für das Bedeutungsmaß erläutert. Hierbei wird der Qualitätskoeffizient (CoQ) herangezogen, der wie der Prognosekoeffizient aus 2.2.3.1 berechnet wird. Die Differenz  $\Delta_e$  wird auf die gleiche Art und Weise berechnet wie beim Bestimmtheitsmaß:

$$\Delta_e = CoQ - CoQ_e \quad (3.3)$$

wobei der Index  $e$  wieder für ein reduziertes Modell steht. Das Maß für die Bedeutung wird wie folgt berechnet:

$$I_e = \frac{\Delta_e}{\sum_e \Delta_e} CoQ \quad (3.4)$$

Hierbei wird durch die Summe der Differenzen normalisiert und zusätzlich mit dem Qualitätskoeffizienten multipliziert. Wie man aus Gleichung 3.4 ableiten kann, muss die Summe der

Bedeutungsmaße gleich dem Qualitätskoeffizienten sein  $\sum_e I_e = CoQ$ . Dies gilt jedoch nicht für jene aus Gleichung 3.2. Ein sehr kleines oder auch negatives Bedeutungsmaß ist ein Zeichen dafür, dass die Variable aus der Modellbildung entfernt werden soll [4].

### 3.1.2 Varianzbasierte Sensitivitätsanalyse

Bei dieser Methode wird die Sensitivität von Variablen durch die Varianz der realen Funktion ermittelt. Es wird eine Funktion in der Form  $Y = f(X_1, X_2, X_2, \dots, X_k)$  angenommen. Des Weiteren wird vorausgesetzt, dass die Variablen  $X_i$  in einem Bereich von  $[0, 1]$  gleichverteilt sind. Die Funktion  $f$  kann wie folgt zerlegt werden [27] [21]:

$$f = f_0 + \sum_i f_i + \sum_i \sum_{j>i} f_{ij} + \dots + f_{12\dots k} \quad (3.5)$$

wobei  $f_i = f_i(X_i)$ ,  $f_{ij} = f_{ij}(X_i, X_j)$  und so weiter, für  $2^k$ -Terme, ist. Wie in [21] ausgeführt, kann die Varianz der Funktionswerte  $Y$  in ähnlicher Weise geschrieben werden:

$$V(Y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \dots + V_{12\dots k} \quad (3.6)$$

Durch Division durch  $V(Y)$  erhält man:

$$\sum_i S_i + \sum_i \sum_{j>i} S_{ij} + \dots + S_{12\dots k} = 1 \quad (3.7)$$

wobei  $S_i$  der Sensitivitätskoeffizient erster Ordnung (*engl.* first order sensitivity coefficient) ist und als Maß für die Sensitivität genutzt wird. Dieser ist wie folgt definiert [21]:

$$S_i = \frac{V_{X_i}(E_{\mathbf{X}_{\sim i}}(Y|X_i))}{V(Y)} \quad (3.8)$$

wobei  $V_{X_i}(\cdot)$  die Varianz über  $X_i$  und  $E_{\mathbf{X}_{\sim i}}(\cdot)$  der Erwartungswert über alle Variablen außer  $X_i$  ist. Er beschreibt Effekte erster Ordnung der Variable  $X_i$  auf die Funktionswerte  $Y$ . Interaktionen höherer Ordnung werden nicht berücksichtigt.

Ein weites beliebtes varianzbasiertes Maß ist der Totaler-Effekt-Index [21]:

$$S_{Ti} = \frac{E_{\mathbf{X}_{\sim i}}(V_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)} = 1 - \frac{V_{\mathbf{X}_{\sim i}}(E_{X_i}(Y|\mathbf{X}_{\sim i}))}{V(Y)} \quad (3.9)$$

Er beschreibt alle Effekte der Variable  $X_i$  auf die Funktionswerte  $Y$ , erster und höherer Ordnung,

In [21] sind einige Formeln für die numerische Berechnung von  $S_i$  und  $S_{Ti}$  angegeben. Im Folgenden sind jene angeschrieben die in [21] vorgestellt wurden:

$$V_{X_i}(E_{\mathbf{X}_{\sim i}}(Y|X_i)) = \frac{1}{N} \sum_{j=1}^N f(\mathbf{B})_j \left( f(\mathbf{A}_{\mathbf{B}}^{(i)})_j - f(\mathbf{A})_j \right) \quad (3.10)$$

$$E_{\mathbf{X}_{\sim i}}(V_{X_i}(Y|\mathbf{X}_{\sim i})) = \frac{1}{2N} \sum_{j=1}^N \left( f(\mathbf{A})_j - f(\mathbf{A}_{\mathbf{B}}^{(i)})_j \right)^2 \quad (3.11)$$

wobei  $\mathbf{A}$  und  $\mathbf{B}$  Stichprobenmatrizen sind. Die Zeilen repräsentieren die Stichproben und die Spalten die Variablen. Die Matrix  $\mathbf{A}_{\mathbf{B}}^{(i)}$  ist ident zu  $\mathbf{A}$  mit dem Unterschied, dass die Spalte  $i$  durch jene von  $\mathbf{B}$  ersetzt wurde. Der Index  $j$  gibt an, welche Stichprobe (Zeile) ausgewertet

wurde. Die Auswertung von Gleichung (3.10) und (3.11) geschieht über die Monte-Carlo-Methode. Für rechenintensive Funktionen  $f$  ist dies mit hohem Zeitaufwand verbunden daher kann in diesen Fällen die Funktion  $f$  durch ein Metamodell approximiert werden, wodurch sich der Rechenaufwand verringert.

In [16] wird der Totale-Effekt-Index  $S_{T_i}$  zusammen mit dem Prognosekoeffizienten  $CoP$  (Abschnitt 2.2.3.1) verwendet, um auf der Basis eines Metamodells mit der besten Prognosefähigkeit (MOP) die Wichtigkeit der Variablen zu quantifizieren. Der Anteil der Varianz einer Eingangsvariable wird aus dem Produkt des Prognosekoeffizienten und Totalen-Effekt-Index errechnet

$$CoP(X_i) = CoP \cdot S_T^{MOP}(X_i) \quad (3.12)$$

wobei  $S_T^{MOP}(X_i)$  der Totale-Effekt-Index der Variable  $X_i$  ist, der mithilfe des Metamodells mit der besten Prognosefähigkeit (MOP) ermittelt wurde.

## 3.2 Versagenswahrscheinlichkeit

Strukturen sollen so bemessen werden, dass sie sicher und nicht eine Gefahr für Personen darstellen. Daher müssen sie auf die Wahrscheinlichkeit ihres Versagens überprüft werden. Die Versagenswahrscheinlichkeit sollte sehr klein sein. Zur Berechnung der Versagenswahrscheinlichkeit müssen numerische Integrationen durchgeführt werden, mit klassischen Integrationsverfahren kann dies jedoch nicht effizient durchgeführt werden. Daher kommen hierfür andere Methoden zum Einsatz [3]. Auch hier müssen komplexe Modelle mechanischer Strukturen ausgewertet werden, die durch Metamodelle ersetzt werden können.

Im Allgemeinen wird das Versagen über eine Grenzzustandsfunktion  $g(\cdot)$  (*engl.* limit state function) definiert, wobei Versagen eintritt, wenn diese für einen gegebenen Input  $\mathbf{X}$  kleiner gleich null ist  $\mathcal{F} = \mathbf{X} : g(\mathbf{X}) \leq 0$ . Der Wert der Grenzzustandsfunktion  $Z = g(\mathbf{X})$  wird als der Sicherheitsabstand bezeichnet. Bei mechanischen Problemen ist die Grenzzustandsfunktion z. B. durch maximale Verschiebungen oder Spannungen definiert. Wenn die Verschiebungen oder Spannungen größer als die vorgegebenen Werte sind wird sie negativ. Die Versagenswahrscheinlichkeit wird über die Auftretenswahrscheinlichkeit des Fehlers definiert [3]:

$$p_f = \mathbf{Prob} [\{\mathbf{X} : g(\mathbf{X}) \leq 0\}] \quad (3.13)$$

Eine Möglichkeit, die Versagenswahrscheinlichkeit zu berechnen, ist über ein multi-dimensionales-Integral:

$$p_f = \mathbf{Prob} [\{\mathbf{X} : g(\mathbf{X}) \leq 0\}] = \int_{g(\mathbf{X}) \leq 0} \cdots \int f_{\mathbf{x}} d\mathbf{x} \quad (3.14)$$

wobei  $f_{\mathbf{x}}$  eine multi-dimensionale-Wahrscheinlichkeitsverteilung der Variablen  $\mathbf{X}$  ist. Aus Gleichung (3.14) ist ersichtlich, dass das Hauptproblem die Auswertung der Grenzzustandsfunktion ist. Es ist wichtig zu erwähnen, dass die Grenzzustandsfunktion lediglich dazu dient, die Grenzen des Integrals zu bestimmen. Es kann grafisch gezeigt werden, dass nur ein kleiner Bereich um den Bemessungspunkt (*engl.* Design-Point) einen Beitrag zum Wert des Integrals leistet. Diese Sachverhalte machten es schwierig, Gleichung (3.14) mit herkömmlichen numerischen Integrationsverfahren zu ermitteln [3].

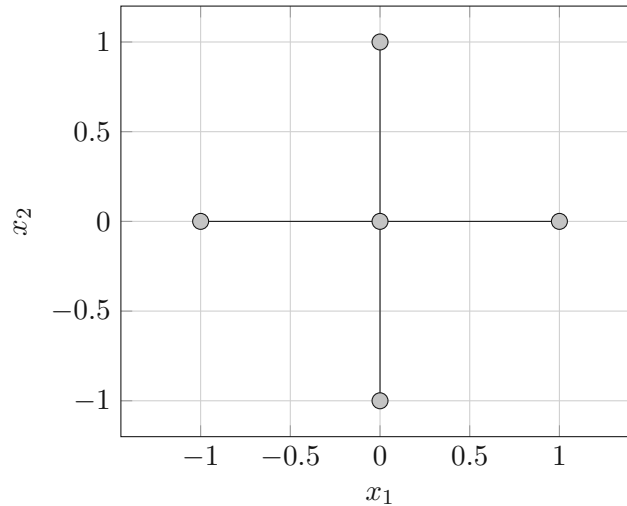
### 3.2.1 Response-Surface-Method (RSM)

Die Grenzzustandsfunktion ist in vielen Fällen nur implizit bekannt oder sehr intensiv an Rechenzeit, daher wird sie bei der Response-Surface-Method durch eine Antwortfläche approximiert.

Es gibt einige Vorschläge, wie die Antwortfläche aufgebaut werden soll, in den meisten Fällen wird sie als eine polynomiale Funktion von Grad zwei ohne Kreuzterme angesetzt [28] [13]:

$$\hat{g}(\mathbf{x}) = a + \sum_{i=1}^n b_i x_i + \sum_{i=1}^n c_i x_i^2 \quad (3.15)$$

Zur genauen Bestimmung der Koeffizienten  $a$ ,  $b_i$  u.  $c_i$  werden  $2n + 1$  Auswertungen der realen Grenzzustandsfunktion benötigt. Als Versuchsplan wird häufig ein axiales Design benutzt (Abbildung 3.1) [13].



**Abb. 3.1:** Axiales Design

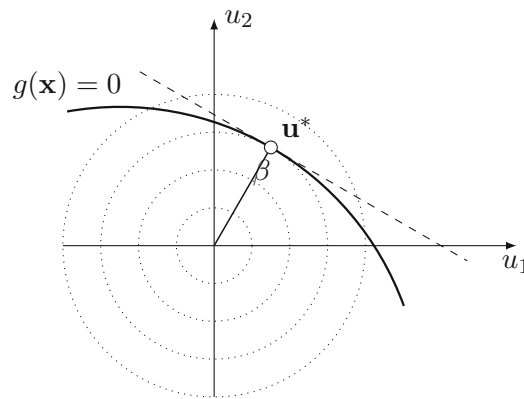
Die Antwortfläche wird genutzt, um den Bemessungspunkt zu ermitteln. An diesem wird nun der reale Funktionswert ermittelt. Im Anschluss wird über eine lineare Interpolation ein neuer Zentralpunkt für eine erneute Iteration bestimmt. Die Interpolation für den nächsten Zentralpunkt  $\mathbf{x}_m^{(k+1)}$  in der  $k$ -ten Iteration wird wie folgt angeschrieben:

$$\mathbf{x}_m^{(k+1)} = \mathbf{x}_m^{(k)} + (\mathbf{x}_D^{(i)} - \mathbf{x}_m^{(k)}) \frac{g(\mathbf{x}_m^{(k)})}{g(\mathbf{x}_D^{(k)}) - g(\mathbf{x}_m^{(k)})} \quad (3.16)$$

wobei  $\mathbf{x}_m^{(k)}$  der Zentralpunkt und  $\mathbf{x}_D^{(k)}$  der ermittelte Bemessungspunkt in der aktuellen Iteration sind und  $g(\mathbf{x}_m^{(k)})$  u.  $g(\mathbf{x}_D^{(k)})$  deren Funktionswert [28] [13].

In der ersten Iteration wird für den Zentralpunkt oft der Mittelwert der Inputvariable verwendet. Die Abstände vom zentralen Punkt zu den Axialpunkten wird über die Standardabweichung der Inputvariablen und einem Skalierfaktor definiert. In den folgenden Iterationen kann der Skalierfaktor sukzessive verkleinert werden. In der Literatur sind auch erweiterte Verfahren zu finden, die eine genauere Anpassung an die Grenzzustandsfunktion ermöglichen [28] [13].

Für die Definition des Bemessungspunktes werden alle Inputvariablen in den unkorrelierten Standardgaußraum transformiert. In diesen Raum ist der Bemessungspunkt jener, der den kürzesten Abstand vom Ursprung zur Grenzzustandsfläche hat (siehe Abbildung 3.2) [3]. Einerseits kann die Versagenswahrscheinlichkeit wie in der First-Order-Reliability-Methode [3] über die Länge des Vektors  $\|\mathbf{u}^*\| = \beta$  bestimmt werden [3]. Die minimale Distanz  $\beta$  wird auch als Zuverläss-



**Abb. 3.2:** Bemessungspunkt  $\mathbf{u}^*$  im unkorrelierten Standardgaußraum

sigkeitsindex (*engl.* reliability index) bezeichnet. Mit diesen wird die Versagenswahrscheinlichkeit wie folgt berechnet:

$$p_f = \Phi(-\beta) \quad (3.17)$$

wobei  $\Phi(\cdot)$  die Verteilungsfunktion der Standardnormalverteilung ist.

Es kann jedoch auch eine Monte-Carlo-Simulation mit der approximierten Grenzzustandsfunktion durchgeführt werden. Hierbei werden mithilfe der Monte-Carlo-Methode  $N$  Stichproben der  $n$  Inputvariablen im Standardgaußraum erzeugt, in den originalen Raum zurücktransformiert und für jede Stichprobe die approximierte Grenzzustandsfunktion ausgewertet. Im Anschluss wird überprüft, ob die Bedingung für Versagen  $g(\mathbf{x}) \leq 0$  erfüllt ist [3]. Die Versagenswahrscheinlichkeit wird über folgende Formel bestimmt:

$$p_f = \frac{1}{N} \sum_{k=1}^N I_g(\mathbf{x}^{(k)}) \quad (3.18)$$

wobei die Funktion  $I_g(\mathbf{x})$  wie folgt definiert ist:

$$I_g(\mathbf{x}) = \begin{cases} 1 & \text{für } g(\mathbf{x}) \leq 0 \\ 0 & \text{sonst} \end{cases} \quad (3.19)$$

### 3.2.2 Directional-Sampling

Eine weitere Möglichkeit, die Versagenswahrscheinlichkeit zu berechnen, ist über das Directional-Sampling. Das Directional-Sampling setzt voraus, dass alle Inputvariablen unabhängig voneinander und standardnormalverteilt sind oder z.B. durch die Nataf-Transformation transformiert werden können [3]. Jeder Punkt  $\mathbf{u}$  im Standardgaußraum kann durch Polarkoordinaten ausgedrückt werden:

$$\mathbf{u} = r\mathbf{a} \quad (3.20)$$

wobei  $r$  der Radius und  $\mathbf{a}$  der Einheitsrichtungsvektor ist, siehe Abbildung 3.3. Die Dichtefunktion von der standardnormalverteilten Zufallsvariable  $\mathbf{U}$  ist:

$$f_{\mathbf{U}}(\mathbf{u}) = \frac{1}{\sqrt{(2\pi)^p}} \exp\left(-\frac{1}{2}\mathbf{u}^T\mathbf{u}\right) \quad (3.21)$$



und durch Einsetzen von Gleichung (3.20) für  $\mathbf{u}$  erhält man für den Exponenten:

$$\mathbf{u}^T \mathbf{u} = (r\mathbf{a})^T (r\mathbf{a}) = r^2 \mathbf{a}^T \mathbf{a} = r^2 \quad (3.22)$$

Daraus ergibt sich, dass die Dichtefunktion  $f_{\mathbf{U}}(\mathbf{u})$  rotationssymmetrisch und unabhängig von  $\mathbf{A}$  ist. Zusätzlich sind  $R$  und  $\mathbf{A}$  voneinander unabhängig, daraus folgt:

$$f_{\mathbf{U}}(\mathbf{u}) = f_{R,\mathbf{A}}(r, \mathbf{a}) = f_{R|\mathbf{A}}(r|\mathbf{a}) f_{\mathbf{A}}(\mathbf{a}) = f_R(r) f_{\mathbf{A}}(\mathbf{a}) \quad (3.23)$$

wobei  $R$  und  $\mathbf{A}$  Zufallsvariablen des Radius und Einheitsrichtungsvektors sind. Die Dichtefunktion für  $f_R(r)$  ergibt sich durch Integration der gemeinsamen Dichtefunktion (3.21) über die Oberfläche  $S_n$  einer  $n$ -dimensionalen Kugel mit Radius  $r$  zu:

$$f_r(r) = S_n r^{n-1} \frac{1}{\sqrt{(2\pi)^p}} \exp\left(-\frac{r^2}{2}\right) \quad (3.24)$$

Es wird vorausgesetzt, dass es in jede Richtung  $\mathbf{a}$  nur einen Punkt  $r^*(\mathbf{a})$  gibt, der Folgendes ergibt:  $g(r^*(\mathbf{a})) = 0$ . Somit kann die Versagenswahrscheinlichkeit wie folgt angeschrieben werden:

$$\begin{aligned} p_f &= \int \cdots \int_{g(\mathbf{X}) \leq 0} f_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} = \\ &= \int_{S_n} \cdots \int_{r^*}^{\infty} f_{R|\mathbf{A}}(r|\mathbf{a}) f_{\mathbf{A}}(\mathbf{a}) dr d\mathbf{a} = \\ &= \mathbf{E}[p_f|\mathbf{a}] \end{aligned} \quad (3.25)$$

Die bedingte Versagenswahrscheinlichkeit wird wie folgt berechnet:

$$\begin{aligned} p_f|\mathbf{a} &= \mathbf{Prob}[R \geq r^*(\mathbf{a})] = \\ &= \int_{r^*}^{\infty} f_{R|\mathbf{A}}(r|\mathbf{a}) dr = \\ &= \int_{r^*}^{\infty} S_n r^{n-1} \frac{1}{\sqrt{(2\pi)^p}} \exp\left(-\frac{r^2}{2}\right) dr = \\ &= 1 - \chi_n^2(r^{*2}(\mathbf{a})) \end{aligned} \quad (3.26)$$

wobei  $\chi_n^2(\cdot)$  die Chi-Quadrat-Verteilung mit Freiheitsgrad  $n$  ist. Ein Schätzwert für die Versagenswahrscheinlichkeit  $p_f$  aus Gleichung (3.25) ergibt sich zu [19]:

$$p_f \approx \frac{1}{N} \sum_{i=1}^N 1 - \chi_n^2(r^{*2}(\mathbf{a})) \quad (3.27)$$

Der aufwendigste Teil an Gleichung (3.27) liegt in der Berechnung des Radius vom Ursprung zur Grenzzustandsfunktion  $r^*(\mathbf{a})$ . Hierbei muss die Grenzzustandsfunktion  $g(\cdot)$  mehrere Male berechnet werden, was bei komplexen Modellen rechenzeitintensiv sein kann. Hier können Metamodelle verwendet werden um die Funktion  $r^*(\mathbf{a})$  zu approximieren. Die Inputvariablen sind in diesem Fall die Richtungsvektoren  $\mathbf{a}$  und die Funktionswerte sind die Radien. In Fällen in denen die Grenzzustandsfunktion nicht geschlossen ist kann es sinnvoll sein als Ausgabewert

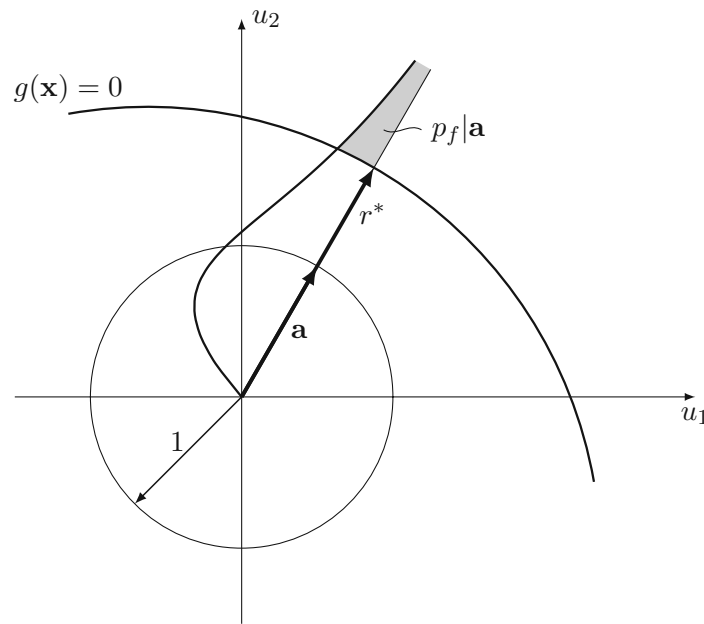


Abb. 3.3: Directional-Sampling

statt den Radius diesen Kehrwert  $\frac{1}{r^*}$  zu benutzen [3]. Um die Versagenswahrscheinlichkeit  $p_f$  (Gleichung (3.27)) zu berechnen kann die Monte-Carlo-Methode herangezogen werden.

### 3.3 Optimierungsprobleme

Ein wichtiger Teil im Bemessungsprozess von mechanischen Strukturen ist deren Optimierung. Es müssen oft kostentreibende Faktoren wie das Volumen des eingesetzten Materials verringert werden und zugleich muss die Struktur in der Lage sein, den Belastungen standzuhalten. Im Allgemeinen ist ein Optimierungsproblem wie folgt definiert:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0; \quad i = 1, \dots, m \end{aligned} \quad (3.28)$$

wobei  $f(\mathbf{x})$  die Zielfunktion und  $g_i(\mathbf{x})$  die Nebenbedingungen sind. Die Nebenbedingungen sind bei mechanischen Strukturen oft Anforderungen an die Verschiebungen oder inneren Kräfte. Für rechenzeitintensive Modelle können hier für die Zielfunktion und die Nebenbedingungen Metamodelle zum Einsatz kommen.

Es ist oft nicht klar, wo sich das Optimum im Raum befindet und daher muss ein großer Teil des Raumes mithilfe eines Versuchsplans abgedeckt werden. Jedoch ist es dem Metamodell hierbei oft nicht möglich, die reale Funktion genau zu beschreiben und das korrekte Optimum zu finden. Daher kann ein iterativer Prozess angewandt werden, indem man zuerst einen Versuchsplan in einem großen Bereich aufstellt und ein Metamodell damit aufbaut. Mit diesem wird ein Optimum bestimmt. Dann wird um das gefundene Optimum ein neuer Versuchsplan erstellt und der zu untersuchende Bereich verkleinert. Dieses Prozedere kann einige Male wiederholt werden, um sich an das wahre Optimum heranzutasten [4]. In Abbildung 3.4 ist das Schema grafisch dargestellt.

Eine weitere Möglichkeit, Optimierung mithilfe von Metamodellen durchzuführen, wird in [11] beschrieben. Hierbei wird davon ausgegangen, dass jeder Punkt des Metamodells einen Mittelwert und eine Standardabweichung besitzt, diese können bei Kriging-Methoden oder Gauß-Prozess-

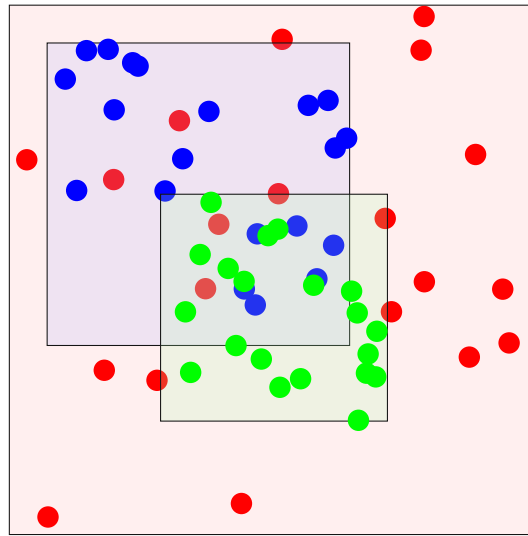


Abb. 3.4: Adaptive Antwortflächen Methode

Regression einfach angeben werden. Es wird ein Funktionswert  $z_{min}$  benötigt, der als aktuelles Optimum bekannt ist. Eine Verbesserung an einem Punkt  $\mathbf{x}$  wird durch folgende Funktion angegeben:

$$I(\mathbf{x}) = \max(z_{min} - z^*(\mathbf{x}), 0) \quad (3.29)$$

wobei  $z^*(\mathbf{x})$  der Metamodellwert an der Stelle  $\mathbf{x}$  ist. Der Funktionswert  $z^*(\mathbf{x})$  ist jedoch eine Zufallsvariable  $Z^*$  und daher muss der Erwartungswertoperator bei Gleichung (3.29) angewandt werden:

$$\mathbf{E}[I(\mathbf{x})] = \mathbf{E}[\max(z_{min} - Z^*, 0)] = (z_{min} - z^*) \Phi\left(\frac{z_{min} - z^*}{s}\right) + s\phi\left(\frac{z_{min} - z^*}{s}\right) \quad (3.30)$$

wobei  $\Phi(\cdot)$  und  $\phi(\cdot)$  die Standardnormalverteilungsfunktion und Standardnormaldichtefunktion sind. Mithilfe von Gleichung (3.30) können Stellen im Inputvariablenraum ermittelt werden, die womöglich das globale Optimum sind. Bei dieser Methode muss zu Beginn ein Versuchsplan ausgewertet werden, um das Metamodell und dessen Standardabweichung zu erhalten. Im Anschluss können mögliche Optimumstellen durch Gleichung (3.30) gefunden und ausgewertet werden. Diese Punkte müssen dem Metamodell hinzugefügt werden, um eine stetige Verbesserung zu erreichen.

Eine Kategorie von Optimierungsproblemen, in denen die Anwendung von Metamodellen beliebt ist, sind zuverlässigkeitsbasierte Optimierungen (*engl.* Reliability-Based Design Optimization (RBDO)) in der Strukturmechanik. Hierbei sind stochastische Variablen (z.B. Steifigkeiten und Belastung) in der Optimierung beteiligt. Ein typisches zuverlässigkeitsbasiertes Optimierungsschema ist in Abbildung 3.5 dargestellt. Die Parameter werden hierbei in deterministische und stochastische aufgeteilt [22]. Die deterministischen Parameter sind häufig die Designparameter, die es zu optimieren gilt, und die stochastischen beschreiben die Unsicherheit im Modell. Es kann auch sein, dass Designparameter stochastische Eigenschaften besitzen, hierbei wird für die Optimierungsschleife (laut Abbildung 3.5) meist der Mittelwert als Wert gebenutzt [22]. An Abbildung 3.5 kann schon erahnt werden, dass diese Problemstellung für rechenzeitintensive Modelle viele Funktionsauswertungen benötigt und die Nutzung von Metamodellen hier einen Vorteil bringt.

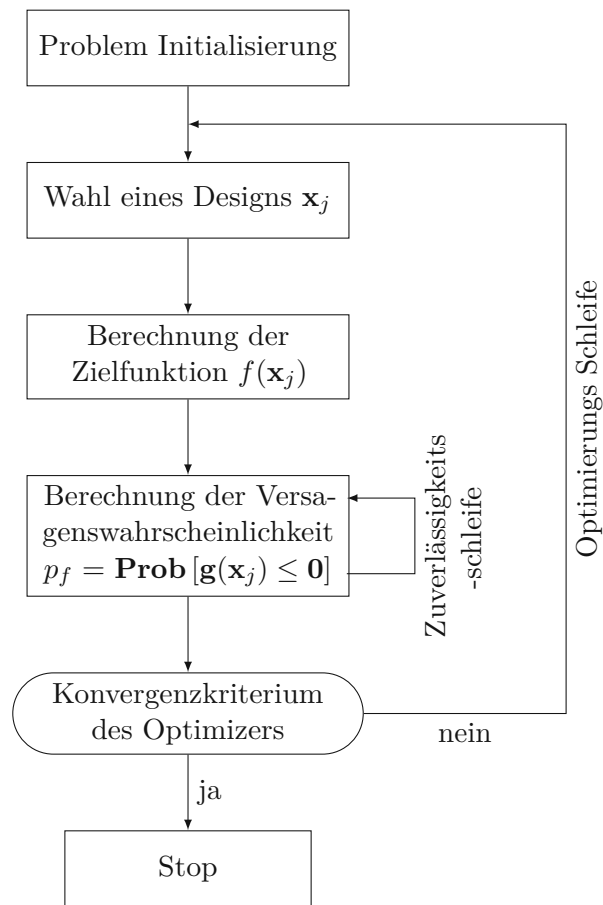


Abb. 3.5: Typisches zuverlässigkeitsbasiertes Optimierungsschema [4]

### 3.4 Approximation von Feldgrößen mittels Singulärwertzerlegung

Metamodelle, die in Kapitel 2 vorgestellt wurden, können nur einen Wert approximieren (bis auf künstliche neuronale Netzwerke). Jedoch kann es hilfreich sein Werte über einen gesamten Bereich zu analysieren, z.B. Verschiebungsfelder aus einer Strukturanalyse oder Zeitverläufe aus dynamischen Berechnungen. Hierbei können einerseits künstliche neuronale Netzwerke eingesetzt werden oder mithilfe der Singulärwertzerlegung (*engl.* Singular-Value-Decomposition (SVD)) Feld-Metamodelle genutzt werden.

Für die Singulärwertzerlegung wird eine Datenmatrix in folgender Form betrachtet:

$$\mathbf{X} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_m \\ | & | & & | \end{bmatrix} \quad (3.31)$$

wobei  $\mathbf{x}_k$ ;  $k = 1, \dots, m$  die Spalten der Matrix  $\mathbf{X}$  sind. Die Datenmatrix hat  $n$  Zeilen und  $m$  Spalten. Die Spalten repräsentieren den Ergebnisverlauf eines Experiments, daraus folgt, dass der Ergebnisverlauf  $n$  Werte besitzt. Die  $m$  Spalten repräsentieren die einzelnen Experimente. Für diese können auch Ergebnisse in höheren Dimensionen verwendet werden, diese müssen jedoch in die Form eines Spaltenvektors gebracht werden.

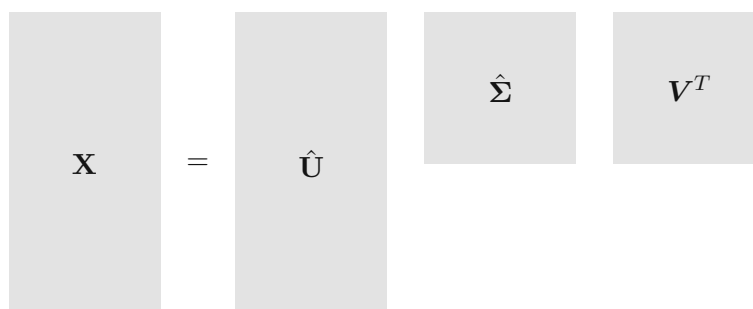
Mit der Singulärwertzerlegung wird die Datenmatrix  $\mathbf{X}$  in folgender Weise zerlegt:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (3.32)$$

wobei  $\mathbf{U}$  ( $n \times n$ ) und  $\mathbf{V}$  ( $m \times m$ ) orthogonale Matrizen sind und  $\mathbf{\Sigma}$  ( $n \times m$ ) eine Matrix mit nicht negativen Einträgen ist [2]. Für den Fall, dass  $n \geq m$  ist, sind die meisten nicht null Einträge in  $\mathbf{\Sigma}$  auf der Diagonalen und  $\mathbf{X}$  kann über die Ökonomische-SVD repräsentiert werden:

$$\mathbf{X} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^T. \quad (3.33)$$

In Abbildung 3.6 ist die Form der Matrizen bei der Ökonomische-SVD dargestellt. Es kann sinnvoll sein, den Mittelwert über die Zeilen  $\bar{\mathbf{X}}$  von der Datenmatrix  $\mathbf{X}$  abzuziehen.



**Abb. 3.6:** Schema der Matrizen bei der ökonomischen SVD

Eine Möglichkeit, die Matrizen  $\mathbf{U}$  und  $\mathbf{\Sigma}$  zu interpretieren, ist im Zusammenhang mit der Kovarianzmatrix. Die Kovarianz des Ergebnisverlaufs  $\mathbf{x}$  ist wie folgt definiert:

$$C_{ij} = \mathbf{Cov}(x_i, x_j), \quad (3.34)$$

wobei  $\mathbf{x}_i$  u.  $\mathbf{x}_j$  diskrete Punkte im Ergebnisverlauf sind. Numerisch lässt sich die Kovarianzmatrix folgendermaßen berechnen:

$$\mathbf{C}_{\mathbf{X}\mathbf{X}} = \frac{1}{N-1} \mathbf{X}\mathbf{X}^T. \quad (3.35)$$

Wenn man nun statt der Datenmatrix  $\mathbf{X}$  ihre Singulärwertzerlegung verwendet, ergibt sich folgender Ausdruck:

$$\mathbf{C}_{\mathbf{X}\mathbf{X}} = \frac{1}{N-1} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T = \frac{1}{N-1} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \frac{1}{N-1} \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T \quad (3.36)$$

Gleichung (3.36) beschreibt ein Eigenwertproblem der Kovarianzmatrix  $\mathbf{C}_{\mathbf{X}\mathbf{X}}$ , wobei  $\mathbf{U}$  die Matrix der Eigenvektoren und  $\mathbf{\Sigma}^2$  die der Eigenwerte ist [2].

Die Zeile  $i$  der Matrix  $\mathbf{V}$  gibt an, wie die Werte der Zeilen in Matrix  $\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}$  kombiniert werden müssen, um die Ergebnisse des  $i$ -ten Experiments zu erhalten. Für z. B. einen Zeitverlauf eines dynamischen Systems würde sich das Experiment  $k$  über folgende Summe wiederherstellen lassen:

$$x_k(t) = \sum_{i=1}^N v_{ki} \hat{\sigma}_i \hat{u}_i(t) \quad (3.37)$$

wobei  $\hat{u}_i(t)$  die  $i$ -te Eigenfunktion,  $\hat{\sigma}_i$  die  $i$ -te Eigenkreisfrequenz und  $v_{ki}$  die Kombinationskoeffizienten für das  $k$ -te Experiment repräsentieren. Wenn man nun annimmt, dass ein Experiment  $k$

mithilfe von Inputparameter  $\mathbf{p}_k$  ermittelt wurde, kann Gleichung (3.37) wie folgt angeschrieben werden:

$$x(t; \mathbf{p}_k) = \sum_{i=1}^N v_i(\mathbf{p}_k) \hat{\sigma}_i \hat{u}_i(t) \quad (3.38)$$

Um unbekannte Parameterkombinationen  $\mathbf{p}$  auswerten zu können, ohne das reale Modell zu evaluieren, können die Kombinationskoeffizienten  $v_i(\mathbf{p})$  mithilfe von Metamodellen approximiert werden. Die Anzahl der aufzustellenden Metamodelle  $N$  ist gleich zu jener der Experimente. Um diese Anzahl zu verringern, können lediglich die wichtigsten Eigenformen und -werte berücksichtigt werden. Ein Kriterium hierfür kann sein, dass so viele Eigenformen  $N_{red}$  verwendet werden sollen, damit 90% der summierten Eigenwerte abgedeckt sind:

$$\frac{\sum_{i=1}^{N_{red}} \hat{\sigma}_i}{\sum_{i=1}^N \hat{\sigma}_i} \approx 0.9 \quad (3.39)$$

Wie in [25] beschrieben, kann ein Metamodell der besten Prognosefähigkeit (Abschnitt 2.2.3.1) für die Kombinationskoeffizienten bestimmt werden. Dies führt zu einem Feld-Metamodell der besten Prognosefähigkeit (F-MOP). Ebenso wird in [25] in Analogie zu Abschnitt 2.2.3.1 erläutert, dass der Prognosekoeffizient (CoP) gegen z. B. die Zeitachse aufgetragen werden kann um den Feld-Prognosekoeffizienten (F-CoP) zu erhalten. Dieser kann genutzt werden um Sensitivitätsmaße der Eingangsvariablen, wie in Abschnitt 3.1.2, über z. B. die Zeitachse zu erhalten. Die Modellqualität und der Einfluss der Eingangsvariablen können so lokal in Bezug zur Systemantwort gesetzt werden [25].

# Kapitel 4

## Lebensdauer Vorhersage der Heißwand einer Brennkammer

In diesem Kapitel wird die Vorhersage der Lebensdauer einer Heißwand einer Brennkammer untersucht (siehe Abbildung 4.1). Bei derartigen Strukturen will man sicherstellen, dass es

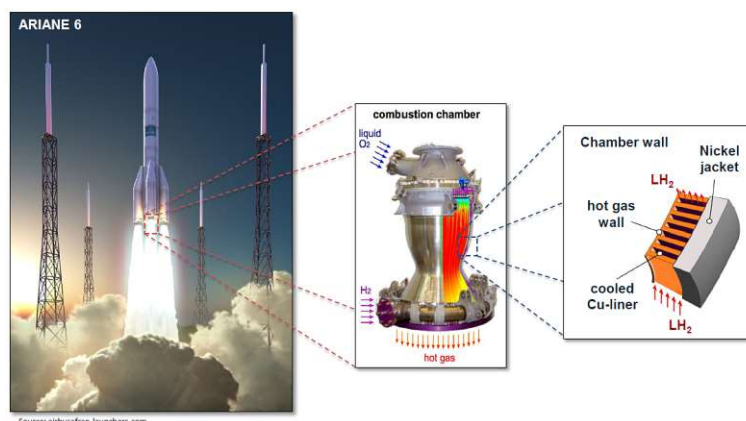


Abb. 4.1: Heißwand des Triebwerks einer Ariane 6 Rakete [12]

zu keinem Versagen kommt, damit die Fracht der Rakete nicht zerstört wird. Daher ist die Lebensdauer hier von entscheidender Bedeutung.

Im Zuge dieser Arbeit wurde die Heißgaswand einer Raketenbrennkammer, die aus einer Kupferlegierung besteht, auf ihr Versagen untersucht. Die Analyse der mechanischen Struktur wurde mit *Ansys Mechanical 2019 R1* durchgeführt. Die Sensitivitätsanalyse und statische Analyse wurden mit *Ansys OptiSlang* und *Ansys Dynardo SoS* durchgeführt. Ebenso wurde *Python 3.7* für die Auswertung von Daten verwendet.

### 4.1 Geometrie und Randbedingungen

Da es sich um eine rotationssymmetrische Struktur handelt, wird aus Gründen der Vereinfachung ein System wie in Abbildung 4.2 betrachtet. Der Mantel besteht aus einer Zinklegierung mit der Dicke  $h_{ni}$  auf welcher sich nach innen hin die Kupferlegierung befindet. In dieser befinden sich Schlitze zur Kühlung des Triebwerks mit der Breite  $w_{cc}$  und der Tiefe  $h_{cc}$ . Die Dicke der Heißgaswand ist mit  $h_{gw}$  bezeichnet. Auf die Heißgaswand (Abbildung 4.2 blaue) wirken der Druck des heißen Gases  $p_{gas}$ , die Wandtemperatur  $T_{wg}$  und der Wärmestrom  $f$  ein. Im Inneren der Kühlkanäle wirkt auf die Struktur der Kühldruck  $p_{H_2}$  und die Flüssigkeitstemperatur  $T_{bu}$  ein. Im ersten Schritt des Belastungszyklus wird das System innerhalb von 2s über den Kühlkanal auf 50 K gekühlt. Im Anschluss wird das Triebwerk gestartet und nach 3s stellt sich in der

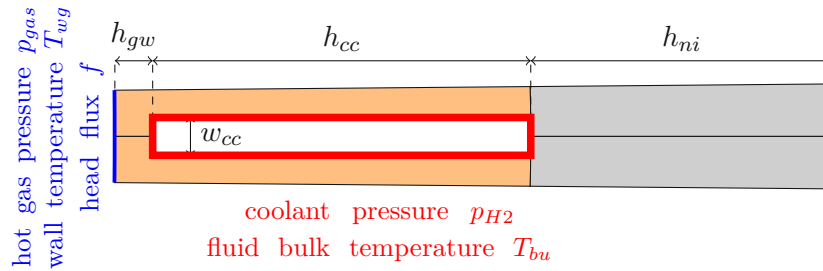


Abb. 4.2: Geometrie und Randbedingungen

Brennkammer eine Temperatur von 1200 K ein. Das Triebwerk läuft 600 s und wird danach abgeschaltet, nach weiteren 3 s wird auch die Kühlung beendet [12].

## 4.2 Materialmodell

Im Folgenden wird auf das Materialmodell der Kupferlegierung eingegangen. Es wird jedoch nicht in allen Einzelheiten erläutert da es nicht Teil der Untersuchung war und ein Programmcode, der in *Ansys Mechanical* implementiert werden konnte, zur Verfügung stand. Weitere Informationen zum Materialmodell können aus [23] entnommen werden.

Das verwendete Modell beschreibt ein Material mit viskoplastischen Materialverhalten, Alterung und Schädigung. Es wird kinematische und isotrope Verfestigung berücksichtigt. Im Folgenden wird die Funktion der Fließfläche für die Beschreibung des plastischen Verhaltens angegeben:

$$f(\boldsymbol{\sigma}, \mathbf{X}, R) = \|\boldsymbol{\sigma} - \mathbf{X}\| - (\sigma_y + R) \quad (4.1)$$

wobei  $\sigma_y$  die Fließspannung und  $\mathbf{X}$  u.  $R$  Terme zur Beschreibung von kinematischer u. isotroper Verfestigung sind. Die kinematische Verfestigung wird durch folgende Gleichungen beschrieben:

$$\dot{X}_{ij}^{(m)} = \frac{2}{3} C^{(m)} \dot{\varepsilon}_{ij}^p - \dot{p} \gamma^{(m)} X_{ij}^{(m)} \quad (4.2)$$

$$X_{\infty}^{(m)} = C / \gamma^{(m)}, \quad m \in 1; 2 \quad (4.3)$$

Die kinematische Verfestigung wird in zwei Teile geteilt  $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$ . Die isotropische Verfestigung wird durch folgende Gleichungen beschrieben:

$$\dot{R} = \dot{p} (Q - \beta R), \quad R_{\infty} = Q / \beta \quad (4.4)$$

wobei  $X_{\infty}$  u.  $R_{\infty}$  die Sättigung der kinematischen und isotropen Verfestigung und  $\gamma$  u.  $\beta$  die sogenannten Dynamic recovery Exponenten der kinematischen u. isotropen Verfestigung sind.

Die viskose Spannungserhöhung wird durch folgende Gleichung definiert:

$$\sigma_v = K \dot{p}^{1/n}$$



wobei  $n$  der Viskositätsexponent und  $K$  der sogenannte Drag stress ist. Das Ermüdungsschädigungsmodell ist wie folgt definiert:

$$N_R = \left( \frac{\Delta \varepsilon}{S} \right)^{-s} \quad (4.5)$$

wobei  $S$  der Duktilitätserschöpfung (Ductility exhaustion) und  $s$  der Ermüdungsexponent sind.

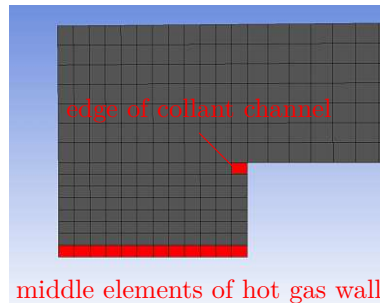
In Tabelle 4.1 sind alle Parameter mit dem Variationskoeffizienten (COV) angegeben. Sie sind alle normalverteilt.

**Tab. 4.1:** Parameter des Heißgaswandmodells

Beschreibung	Parameter	Einheit	COV
<i>Geometrie Parameter:</i>			
Wanddicke	$h_{gw}$	mm	0.014
Kanalweite	$w_{cc}$	mm	0.010
<i>Operative Parameter:</i>			
Wärmestrom	$q_{gw}$	mW/mm	0.001
Wandtemperatur	$T_{wg}$	K	0.008
Kühltemperatur	$T_{bu}$	K	0.030
Heißgasdruck	$p_{gas}$	MPa	0.010
Kühldruck	$p_{H2}$	MPa	0.005
<i>Material Parameter:</i>			
Wärmeausdehnung	$\alpha$	1/K	0.006
Wärmeleitfähigkeit	$\lambda$	mW K/mm	0.005
Elastizitätsmodul	$E$	MPa	0.0002
Poissonzahl	$\nu$	-	0
Fließspannung	$f_y$	MPa	0.040
Drag stress	$K$	MPa	0.007
Viskositätsexponenten	$n$	-	0.017
1st kinematic saturation	$X_\infty^{(1)}$	MPa	0.010
1st kinematic exponent	$\gamma^{(1)}$	-	0.005
1st kin. static recovery	$A_x^{(1)}$	-	0.010
2st kinematic saturation	$X_\infty^{(2)}$	MPa	0.010
2st kinematic exponent	$\gamma^{(2)}$	-	0.040
2st kin. static recovery	$A_x^{(2)}$	-	0.007
1st isotropic saturation	$R_\infty^{(1)}$	MPa	0.010
1st isotropic exponent	$\beta^{(1)}$	-	0.050
1st iso. static recovery	$A_r^{(1)}$	-	0.010
Duktilitätserschöpfung	$S$	-	0.017
Ermüdungsexponent	$s$	-	0.005

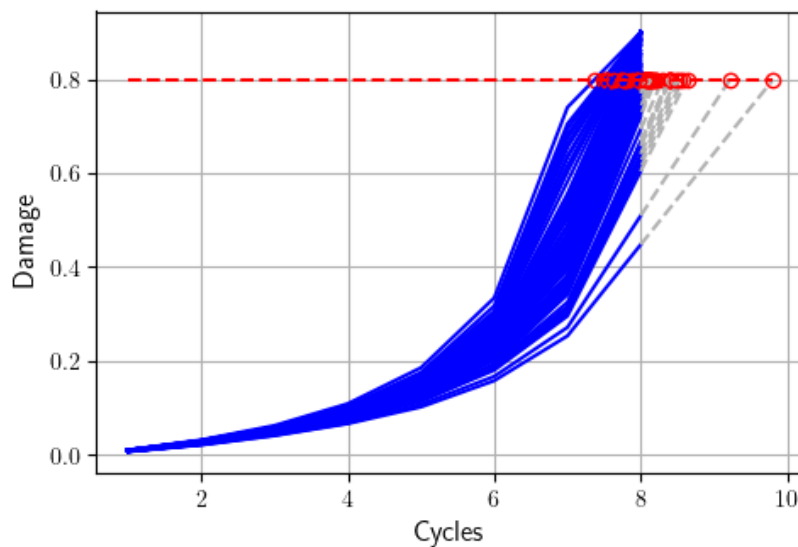
### 4.3 Sensitivitätsanalyse

Für die Sensitivitätsanalyse werden die Elemente in der Mitte der Heißgaswand und in der Ecke des Kühlkanals betrachtet (siehe Abbildung 4.3). Es wird die Schädigung der einzelnen



**Abb. 4.3:** Elemente der Heißgaswand die für die Sensitivitätsanalyse betrachtet werden

Elemente ermittelt und falls diese eine Schädigung von über 0,9 besitzen wird die Simulation abgebrochen. Um die Berechnung von zu vielen Zyklen zu vermeiden, wird die Simulation nach acht Zyklen beendet, siehe Abbildung 4.4. Der Lastzyklus des Versagens wird durch Interpolation



**Abb. 4.4:** Maximale Schädigung im Modell in Abhängigkeit der Belastungszyklen

bzw. Extrapolation der maximalen Schädigung der beiden letzten Zyklen berechnet, siehe Abbildung 4.4.

Für die Sensitivitätsanalyse wurden 100 Designs ausgewertet, die mittels Latin-Hypercube-Sampling erzeugt wurden. Mithilfe dieser wurde in *Ansys OptiSlang* ein Metamodell erzeugt, welches genutzt wurde, um die Sensitivität der Parameter zu berechnen. In *Ansys OptiSlang* wird der *CoP* Coefficient of Prognosis ausgegeben, siehe Abschnitt 2.2.3.1. Die Wichtigkeit eines Parameters  $X_i$  wird über den  $CoP(X_i)$  angegeben, siehe Abschnitt 3.1.2.

Die Sensitivität für den Lastzyklus des Versagens ist in Abbildung 4.5 dargestellt. Hier ist zu erkennen, dass die Temperatur der Heißgaswand  $T_{wg}$  und die Duktilitätserschöpfung (Ductility exhaustion)  $S$  den größten Einfluss auf die Versagenszyklen haben.

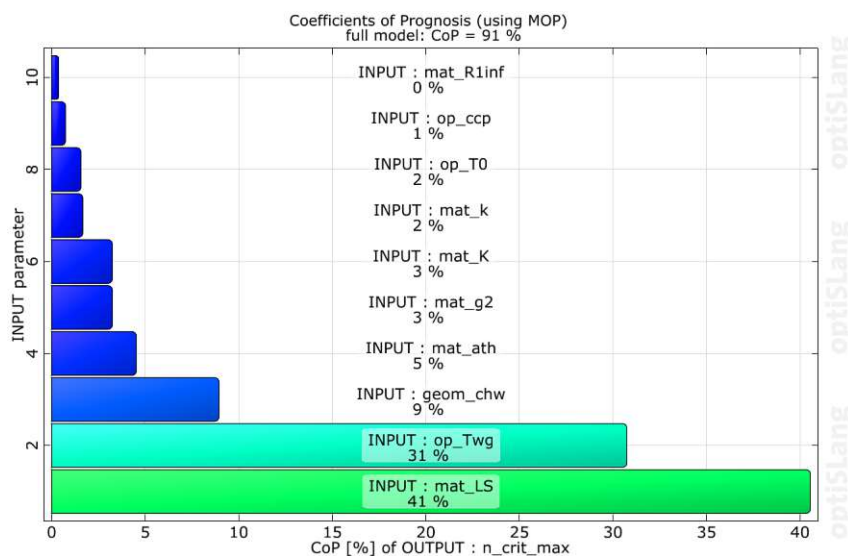


Abb. 4.5: Sensitivität des Lastzyklus des Versagens

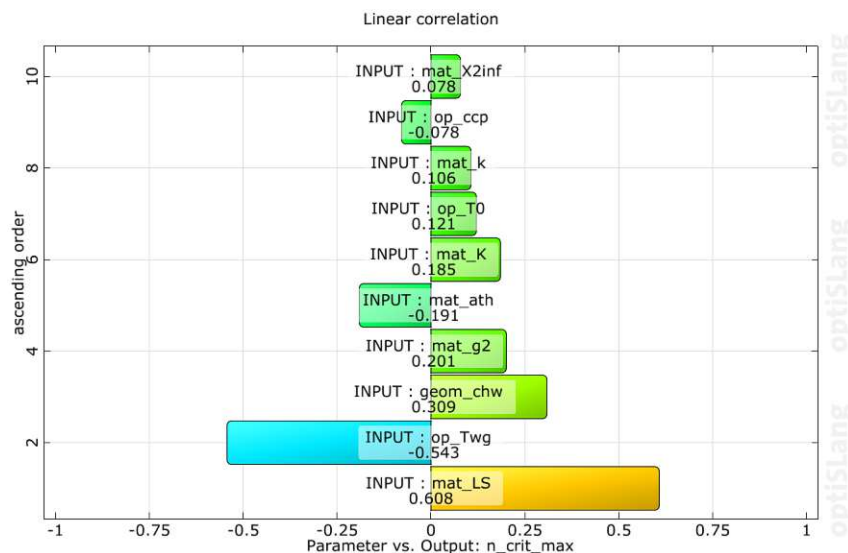


Abb. 4.6: Zyklen bis Versagen Histogramm

In Abbildung 4.6 ist die lineare Korrelation der Parameter  $X_i$  dargestellt. Hier kann abgelesen werden, dass wie auch in Abbildung 4.5 die Temperatur der Heißgaswand  $T_{wg}$  und die Duktilitätsererschöpfung (Ductility exhaustion)  $S$  den größten Einfluss besitzen. Durch eine Erhöhung von  $S$  werden die Zyklen bis zum Versagen erhöht und durch Erhöhen von  $T_{wg}$  werden die Zyklen verringert.

In Abbildung 4.7 ist das Histogramm der Zyklen bis zum Versagen dargestellt, welches aus 100 Latin-Hypecube-Designs resultiert. Der Mittelwert ergibt sich zu 7.945 und Standardabweichung zu 0.345.

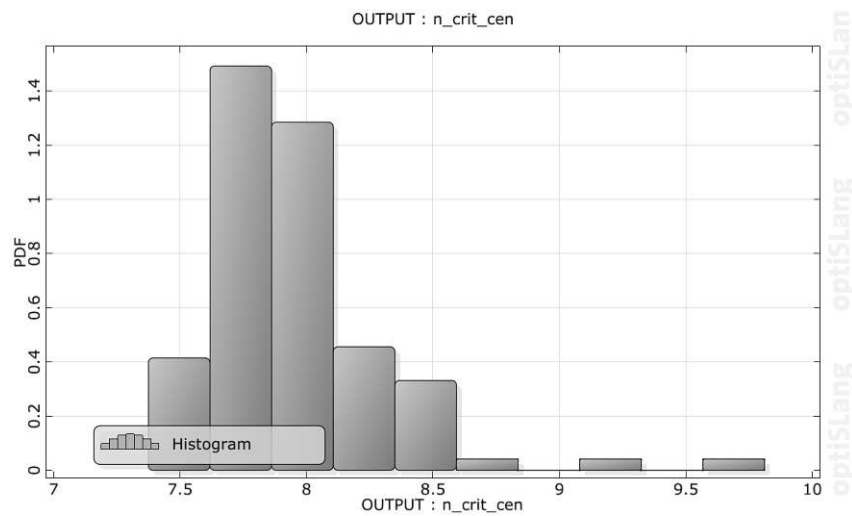


Abb. 4.7: Zyklen bis Versagen Histogramm

#### 4.4 Statistische Analyse

Nach der Sensitivitätsanalyse wurden die Ergebnisse des FE-Modells mithilfe der Software *Ansys Dynardo SoS* analysiert. *SoS* ist die Abkürzung für Statistics of Structure und mithilfe dieser Software können statistische Eigenschaften über jedes Element in einem FE-Modell errechnet werden. Hierzu werden Ergebnisse wie z. B. Schädigung und Verzerrungen der 100 Latin-Hypercube-Designs aus der Sensitivitätsanalyse genommen und für jedes Element statistische Werte über die Designs ermittelt, siehe Abbildung 4.8. Es können Feld-Metamodelle erzeugt werden, wie in Abschnitt 3.4 beschrieben.

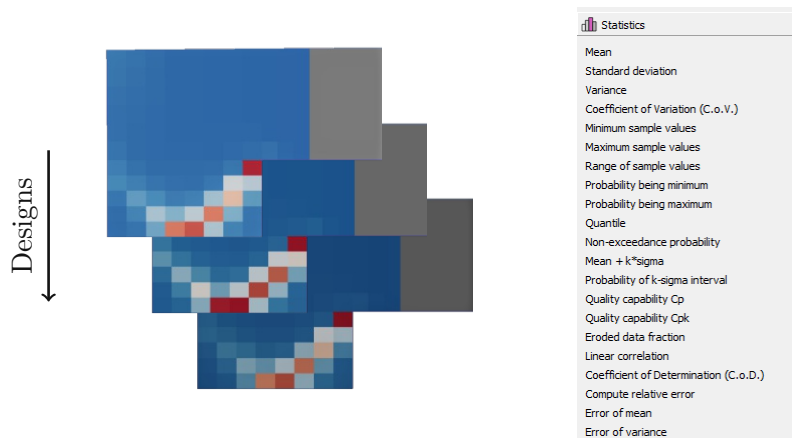


Abb. 4.8: Statistische Eigenschaften über alle Designs für jedes Element

Zuerst wurde die Schädigung der einzelnen Elemente betrachtet. Hierbei wurde der Mittelwert und die Standardabweichung ermittelt, siehe Abbildung 4.9 und 4.10. Da die Schädigung jedoch mit 90% begrenzt ist und sie in den Elementen von Interesse immer 0.9 ist, muss zum Aufbau eines Feld-Metamodells eine andere Größe herangezogen werden. Da die Schädigung laut Gleichung (4.5) von den Verzerrungen abhängt, wurden die plastischen Verzerrungen herangezogen. Aus Abbildung 4.11 und 4.12 kann entnommen werden, dass die maximalen plastischen Verzerrungen an denselben Elementen auftreten wie die maximale Schädigung.

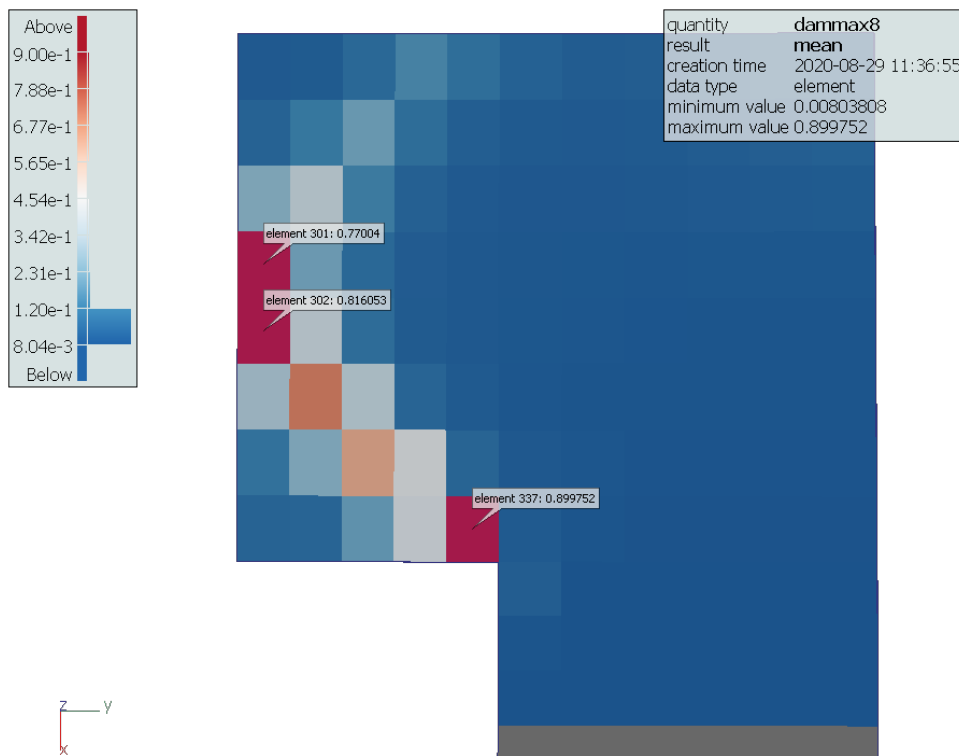


Abb. 4.9: Mittelwert der Element Schädigung

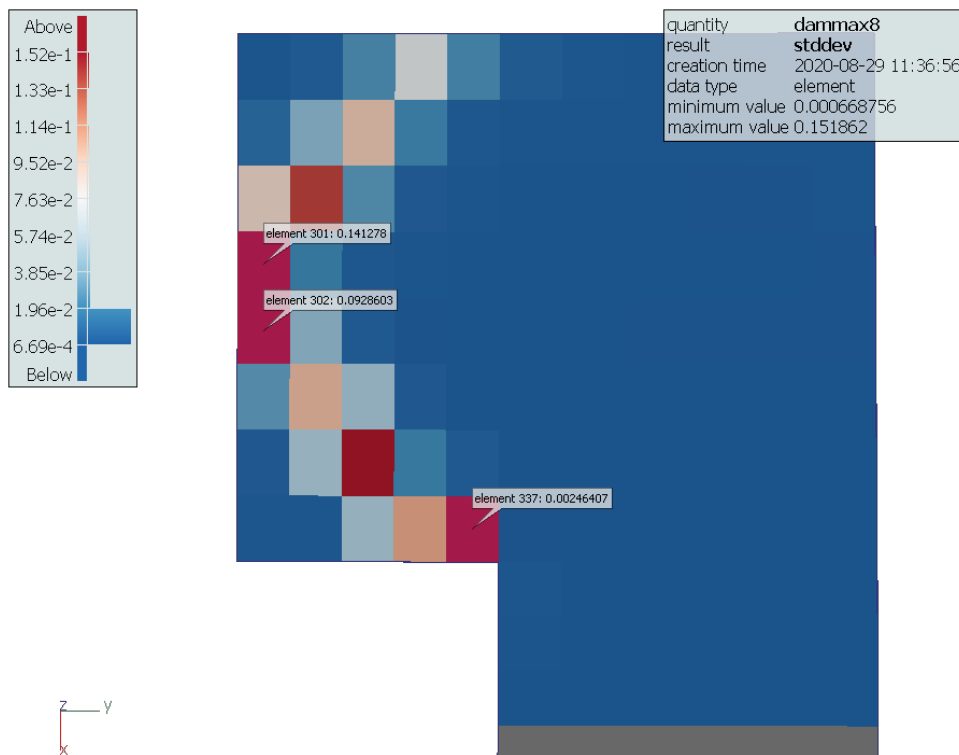


Abb. 4.10: Standardabweichung der Element Schädigung

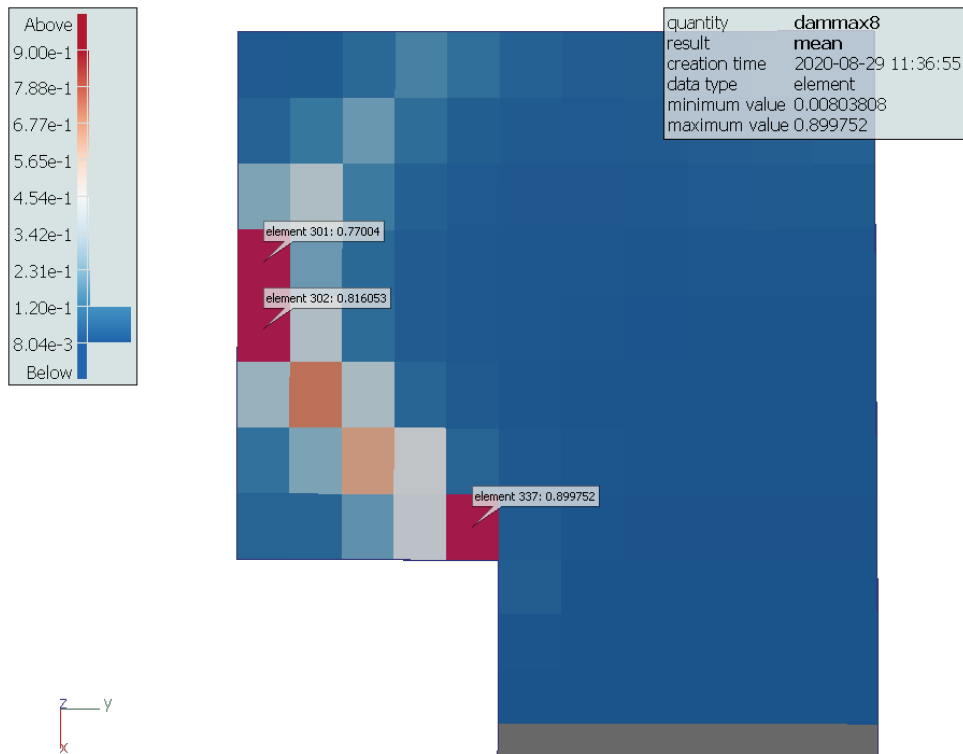


Abb. 4.11: Mittelwert der Schädigung

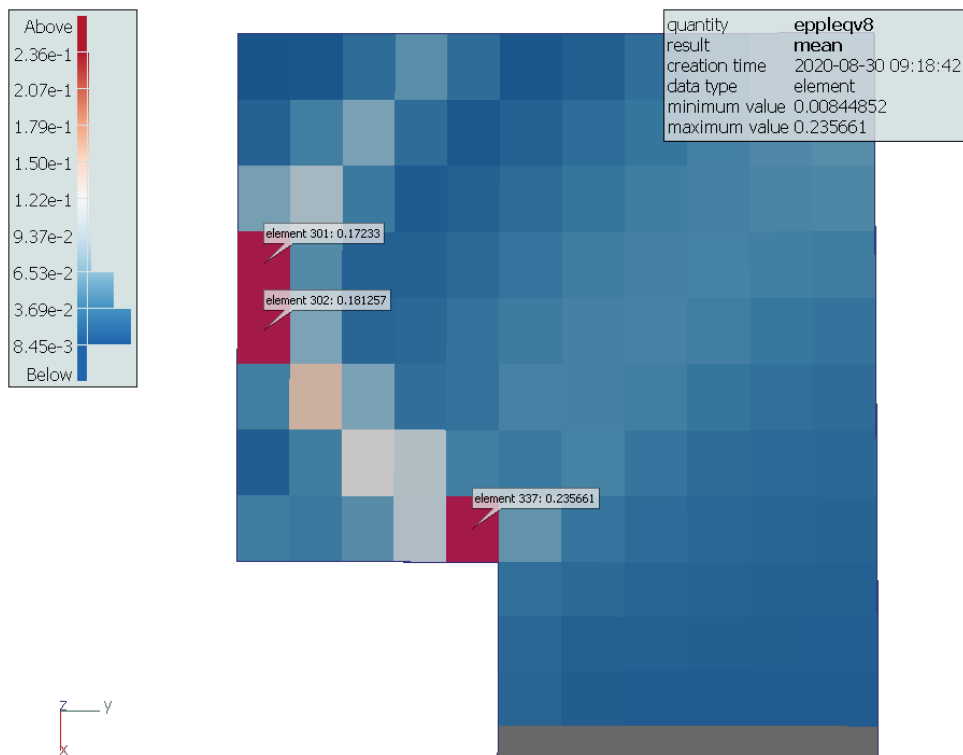


Abb. 4.12: Mittelwert der plastischen Verzerrungen

Für die plastischen von-Mises-Verzerrungen  $\varepsilon_{vM}^{pl}$  wird in *SoS* für jeden der acht Lastzyklen ein Feld-Metamodell erstellt. In Abbildung 4.13 sind die Güte der Metamodelle und die Einflüsse der Parameter angegeben. Wie aus Abbildung 4.13 entnommen werden kann, werden die plastischen von-Mises-Verzerrungen ebenso durch die Heißgastemperatur  $T_{wg}$  und die Duktilitätserschöpfung (Ductility exhaustion)  $S$  am meisten beeinflusst. Der F-CoP[Total] in Abbildung 4.14 gibt die beschreibbare Variation im gesamten Feld an. Bereiche mit kleiner Standardabweichung  $\sigma[\text{FMOP}]$ , siehe Abbildung 4.15, haben typischerweise kleine F-CoP-Werte und jene mit großer Standardabweichung haben große F-CoP-Werte. Dies resultiert daraus, dass jene Bereich mit einer hohen Standardabweichung genau beschrieben werden müssen, um die Unsicherheiten abzudecken.

Um die Anzahl der Zyklen bis zum Versagen mithilfe der plastischen von-Mises-Verzerrungen ermitteln zu können, muss zusätzlich die plastischen von-Mises-Verzerrungen beim Versagen bestimmt werden. Hierzu wird die maximale plastische Verzerrung über die Zyklen aufgetragen und im Anschluss jene Verzerrungen ermittelt, die bei den zugehörigen Versagenszyklen liegen, siehe Abbildung 4.16.

Mithilfe des Feld-Metamodells kann nun die Statistik für den Versagenszyklus für mehrere Designs berechnen werden. Der Programmablauf wird wie folgt durchgeführt:

1. Generieren eines Designs (Inputvariablen Werte)
2. Berechnen des Feld-Metamodells für die plastischen von-Mises-Verzerrungen  $\varepsilon_{vM}^{pl}$  für den 7ten und 8ten Zyklus
3. Berechnen des Versagensverzerrung  $\varepsilon_{vM, fail}^{pl}$
4. Berechnen des Versagenszyklus  $n_{fail}$  durch Interpolation bzw. extrapolation von  $\varepsilon_{vM}^{pl}$

$$n_{fail} = 6 + \frac{\varepsilon_{vM, fail}^{pl} - \varepsilon_{vM, 6}^{pl}}{\varepsilon_{vM, 7}^{pl} - \varepsilon_{vM, 6}^{pl}} \quad (4.6)$$

Dieses Schema kann im Anschluss mithilfe einer Monte-Carlo-Simulation für eine große Anzahl von Parameterkonfigurationen durchgeführt werden. Hiermit können statistische Werte für den Versagenszyklus ermittelt werden. In Abbildung 4.17 ist das Histogramm der Versagenszyklen, die mithilfe des Feld-Metamodells ermittelt wurden, dargestellt. Der Mittelwert und die Standardabweichung ergeben sich zu 7,939 und 0,280. Zum Vergleich, der Mittelwert und die Standardabweichung die mithilfe von Latin-Hypercube-Sampling und dem realen FE-Modell ermittelt wurden liegen bei 7,945 und 0,346.

Feld-Metamodelle haben den Vorteil, dass man nicht vor Beginn der Analyse wissen muss, an welchen Punkten einer Struktur die interessanten Werte auftreten können. In der folgenden Analyse können somit mehrere Punkte genutzt werden, die jedoch je nach Eingangsvariablen räumlich variieren können. Mit den in der Software *Ansys Dymardo SoS* angebotenen Methoden können ebenso Punkte, die von Interesse sind, gefunden werden.

	epplevq1	epplevq2	epplevq3	epplevq4	epplevq5	epplevq6	epplevq7	epplevq8
F-CoPTotall	99.29 %	99.31 %	99.40 %	99.48 %	98.77 %	99.69 %	95.34 %	98.88 %
F-CoPgeom chw1	7.79 %	8.41 %	7.75 %	6.46 %	4.99 %	5.87 %	6.20 %	4.25 %
F-CoPgeom haw1	9.80 %	10.92 %	10.47 %	9.49 %	4.41 %	4.30 %	5.30 %	4.41 %
F-CoPmat Ar11	0.03 %	0.02 %	0.01 %	0.01 %	0.02 %	0.01 %	1.82 %	0.83 %
F-CoPmat Ax11	0.04 %	0.02 %	0.01 %	0.01 %	0.03 %	0.01 %	1.91 %	0.91 %
F-CoPmat Ax21	0.03 %	0.03 %	0.01 %	0.01 %	0.03 %	0.01 %	1.94 %	0.92 %
F-CoPmat EI	0.04 %	0.02 %	0.01 %	0.01 %	0.04 %	0.01 %	1.95 %	0.90 %
F-CoPmat KI	18.83 %	15.43 %	11.75 %	8.75 %	4.82 %	3.12 %	3.03 %	1.14 %
F-CoPmat LSI	0.04 %	0.03 %	0.70 %	3.12 %	18.86 %	18.53 %	22.42 %	28.72 %
F-CoPmat Lsl	0.03 %	0.02 %	0.01 %	0.29 %	1.17 %	1.66 %	3.45 %	3.65 %
F-CoPmat R.limfl	2.98 %	2.00 %	1.54 %	1.48 %	1.89 %	1.79 %	4.91 %	3.24 %
F-CoPmat X.limfl	23.37 %	15.61 %	11.20 %	8.00 %	4.15 %	2.55 %	3.25 %	1.40 %
F-CoPmat X2imfl	0.14 %	0.95 %	1.57 %	1.98 %	1.04 %	1.95 %	3.56 %	2.61 %
F-CoPmat ath1	6.15 %	5.12 %	5.89 %	7.47 %	8.59 %	10.74 %	12.86 %	13.39 %
F-CoPmat b1l	0.05 %	0.03 %	0.02 %	0.01 %	0.05 %	0.02 %	1.86 %	0.88 %
F-CoPmat a1l	0.04 %	0.02 %	0.01 %	0.01 %	0.02 %	0.01 %	1.82 %	0.88 %
F-CoPmat a2l	4.19 %	10.88 %	15.02 %	16.09 %	10.53 %	10.93 %	13.28 %	12.11 %
F-CoPmat kl	7.27 %	7.50 %	6.95 %	5.75 %	3.95 %	2.28 %	2.37 %	1.44 %
F-CoPmat lam1	0.08 %	0.05 %	0.02 %	0.02 %	0.09 %	0.05 %	1.90 %	0.90 %
F-CoPmat nl	0.05 %	0.03 %	0.01 %	0.01 %	0.02 %	0.03 %	1.88 %	0.88 %
F-CoPfoo T0l	0.68 %	0.07 %	0.02 %	0.01 %	0.10 %	0.03 %	2.43 %	1.09 %
F-CoPfoo Twdl	18.95 %	22.67 %	26.69 %	31.93 %	40.30 %	39.94 %	37.60 %	35.19 %
F-CoPfoo ccol	1.95 %	2.06 %	2.00 %	1.92 %	1.56 %	1.25 %	2.63 %	0.99 %
F-CoPfoo hfl	0.03 %	0.02 %	0.01 %	0.01 %	0.04 %	0.02 %	1.95 %	0.87 %
F-CoPfoo haol	0.04 %	0.02 %	0.01 %	0.01 %	0.04 %	0.01 %	1.84 %	0.89 %

Abb. 4.13: Güte und Einfluss der Parameter für die Feld-Metamodell der 8 Lastzyklen



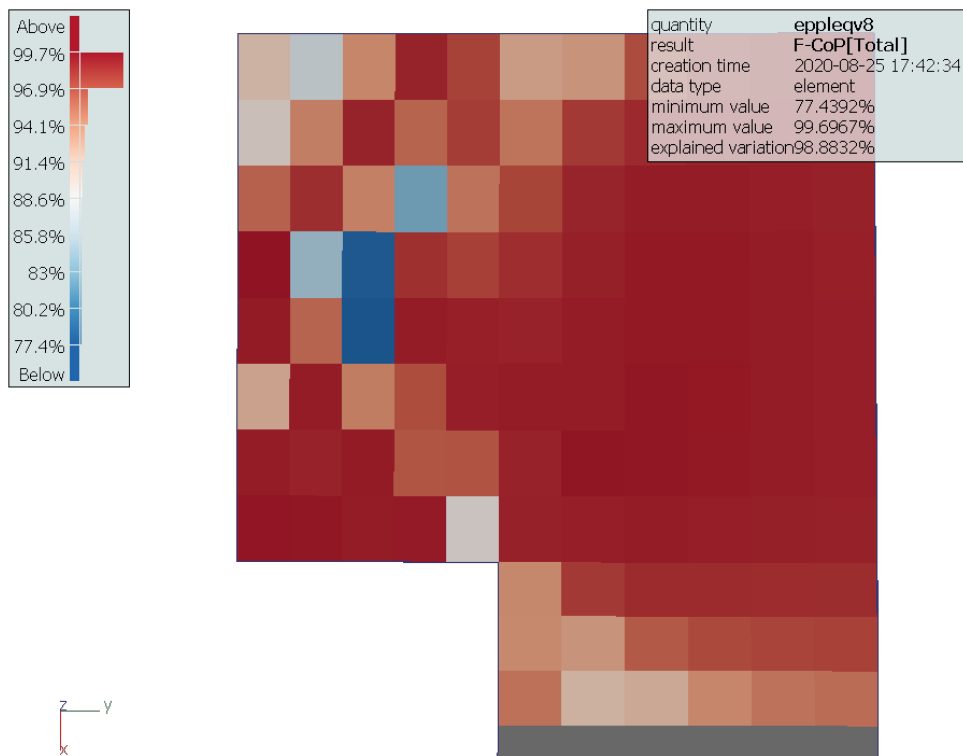


Abb. 4.14: Feldprognosekoeffizient (F-CoP) für das Feldmetamodell der plastischen von-Mises-Verzerrungen

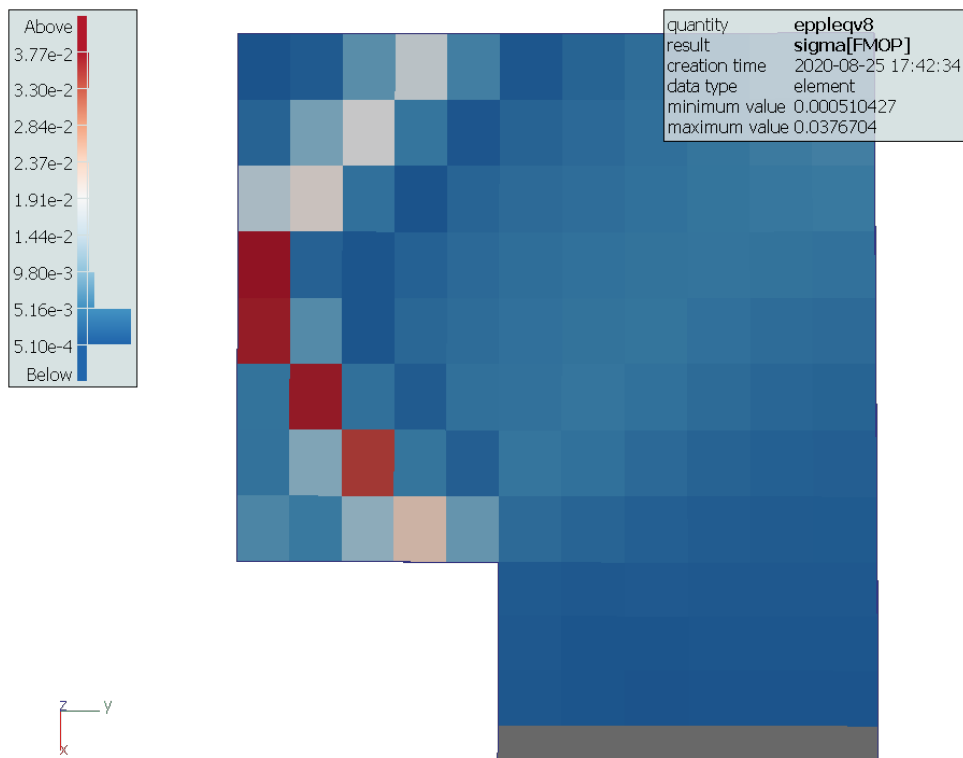


Abb. 4.15: Standardabweichung (sigma[FMOP]) für das Feldmetamodell der plastischen von-Mises-Verzerrungen

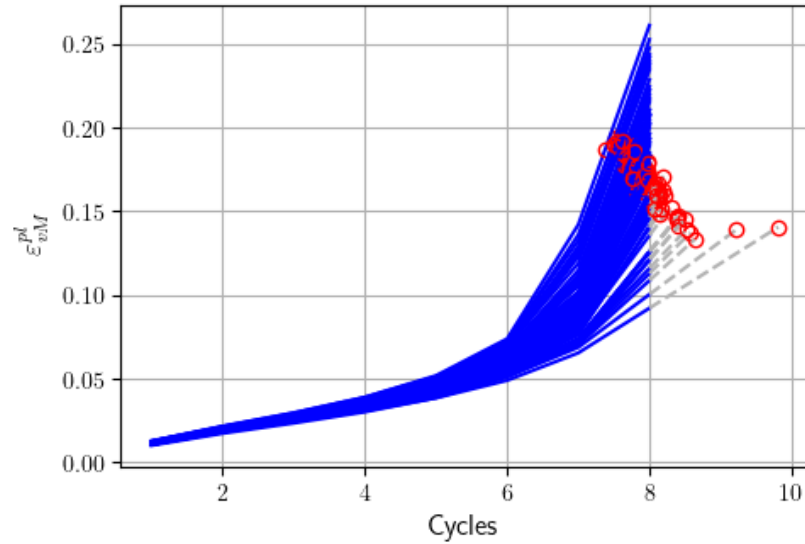


Abb. 4.16: Plastische von-Mises-Verzerrungen bei Versagen  $\epsilon_{vM, fail}^{pl}$

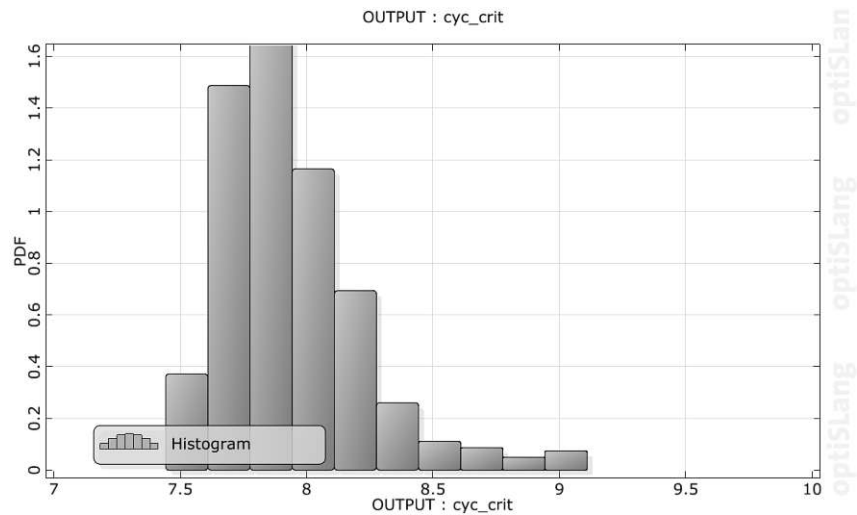


Abb. 4.17: Histogramm der Versagenszyklus die mit dem Feld-Metamodell berechnet wurden

# Kapitel 5

## Schlussfolgerungen

Metamodelle sind Analysewerkzeuge, um aufwendige Berechnungsmodelle (lange Berechnungszeiten) von Strukturen (FE-Modelle) in einer effizienten Art und Weise zu untersuchen. Wie in den vorangegangenen Kapiteln erläutert wurde, ist der mathematische Aufbau einfach und in den schlimmsten Fällen müssen nur Matrizen invertiert werden. Diese Methode kann für eine Vielzahl von Verfahren verwendet werden, um die Rechenzeit zu verringern. Für die mechanische Strukturanalyse relevante Bereiche wären die Sensitivitätsanalyse, die Berechnung von Versagenswahrscheinlichkeiten und die Optimierung. Es sind auch Anwendungen in der maschinellen Herstellung von z.B. Fertigteilen in der Baubranche möglich. In [17] wurde ein Metamodell verwendet um ein FE-Modell, welches die Durchbiegung von Brettschichtholzträger berechnet, zu approximiert. Mit diesen wird im Anschluss mittels Optimierungsverfahren, aus einer gegebenen Anzahl an Holzbrettern, Brettschichtschichtholzträger erzeugt, die ungefähr die gleichen Verformungseigenschaften besitzen.

Es muss jedoch auch erwähnt werden, dass Metamodelle keine Wunderwaffe sind, mit denen im Anschluss alles berechnet werden kann. Sie werden für ein spezifisches Problem aufgestellt und sind meist durch die zur Aufstellung des Modells benutzten Daten limitiert und können z. B. nicht für Stellen außerhalb des untersuchten Bereichs sicher eingesetzt werden. Es ist auch immer die Approximationsgüte genau zu betrachten, da nicht alle Punkte im untersuchten Raum gleich gut approximiert werden können und dadurch womöglich sehr lokale Effekte nicht abgebildet werden können.

Im Rahmen dieser Arbeit wurden die maximalen Lastzyklen bis zum Versagen einer Raketendbrennkammerheißwand mithilfe von Metamodellen untersucht. Durch die Sensitivitätsanalyse wurden jene Parameter erkannt, die den meisten Einfluss auf den maximalen Lastzyklus haben. Im Anschluss wurde mithilfe eines Feld-Metamodells die Ergebnisse eines ganzen Bereichs approximiert, um für die Bestimmung des maximalen Lastzyklus Feldgrößen zur Verfügung zu haben. Um den maximalen Lastzyklus mithilfe des Metamodells berechnen zu können, wurde ein Metamodell für die plastische Verzerrung bei Versagen ermittelt. Durch das Feld-Metamodell können auch jene Bereiche gefunden werden, die durch eine Änderung der Inputvariablen die größten Änderungen erfahren. Mit dem ermittelten Feld-Metamodell kann der Aufwand für eine Monte-Carlo-Simulationen wesentlich verringert werden. Die Auswertung des Metamodells benötigt ca. 30 s und die des FE-Modells ca. 1,7 h.

# Literatur

- [1] M. Björkman und K. Holmström. „Global Optimization of Costly Nonconvex Functions Using Radial Basis Functions“. In: *Optimization and Engineering* 1 (2000), S. 373–397.
- [2] S. L. Brunton und J. N. Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge: Cambridge University Press, 2019. ISBN: 978-1-108-42209-3.
- [3] C. Bucher. *Computational Analysis of Randomness in Structural Mechanics*. 3. Aufl. London: CRC Press, 2009. ISBN: 978-0-415-40354-2.
- [4] C. Bucher. „Metamodels of optimal quality for stochastic structural optimization“. In: *Probabilistic Engineering Mechanics* 54 (2018), S. 131–137.
- [5] C. Bucher und M. Macke. „Application of approximate Response Functions in Structural Reliability Analysis“. In: *Probabilistic Engineering Mechanics* (2018).
- [6] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge: Cambridge University Press, 2003. ISBN: 978-0-511-54324-1.
- [7] N. A. C. Cressie. *Statistics for Spatial Data, Revised Edition*. New-York: John Wiley a. Sons, Inc., 1993. ISBN: 978-0-471-00255-0.
- [8] R. A. Fisher. *The Design of Experiments*. Edinburgh und London: Oliver und Boyd, 1935.
- [9] M. N. Gibbs. „Bayesian Gaussian Processes for Regression and Classification“. Dissertation. University of Cambridge, 1997.
- [10] M. T. Hagan, H. B. Demuth, M. H. Beale und O. De Jesús. *Neural Network Design*. 2. Aufl. Martin Hagan, 2014. ISBN: 978-0-971-73211-7.
- [11] D. R. Jones, M. Schonlau und W. J. Welch. „Efficient Global Optimization of Expensive Black-Box Functions“. In: *Journal of Global Optimization* 13 (1998), S. 455–492.
- [12] M. Lehmann. „TMF Panel optimization“. In: 14th Weimar Optimization and Stochastic Days (1.–2. Juni 2017). Weimar, Germany, Juni 2017.
- [13] X.-J. Meng, S.-K. Jing, L.-X. Zhang, J.-H. Liu und H.-C. Yang. „A New Sampling Approach for Response Surface Method Based Reliability Analysis and Its Application“. In: *Advances in Mechanical Engineering* 7 (Feb. 2014), S. 305473–305473.
- [14] T. Most und c. Bucher. „A Moving Least Squares weighting function for the Element-free Galerkin Method which almost fulfills essential boundary conditions“. In: *Structural Engineering and Mechanics* 21 (2005).
- [15] T. Most und J. Will. „Metamodel of Optimal Prognosis - An automatic approach for variable reduction and optimal meta-model selection“. In: Weimar Optimization and Stochastic Days 5.0 (20.–21. Nov. 2008). Weimar, Germany, Nov. 2008.
- [16] T. Most und J. Will. „Sensitivity analysis using the Metamodel of Optimal Prognosis“. In: Weimar Optimization and Stochastic Days 2011 (24.–25. Nov. 2011). Weimar, Germany, 2011.

- [17] S. Pech, G. Kandler, M. Lukacevic und J. Füssl. „Metamodel assisted optimization of glued laminated timber beams by using metaheuristic algorithms“. In: *Engineering Applications of Artificial Intelligence* 79 (2019), S. 129–141. ISSN: 0952-1976.
- [18] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan und P. Kevin Tucker. „Surrogate-based analysis and optimization“. In: *Progress in Aerospace Sciences* 41.1 (2005), S. 1–28. ISSN: 0376-0421.
- [19] D. Roos. „Approximation und Interpolation von Grenzzustandsfunktionen zur Sicherheitsbewertung nichtlinearer Finite-Elemente-Strukturen“. Dissertation. Bauhaus-Universität at Weimar, 2001.
- [20] J. Sacks, W. J. Welch, T. J. Mitchell und H. P. Wynn. „Design and Analysis of Computer Experiments“. In: *Statistical Science* 4.4 (1989), S. 409–423.
- [21] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto und S. Tarantola. „Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index“. In: *Computer Physics Communications* 181 (Feb. 2010), S. 259–270.
- [22] D. Schneider. „Beitrag zur effizienten Strukturoptimierung unter Berücksichtigung von Unsicherheiten“. Dissertation. Technische Universität Wien, 2010.
- [23] W. Schwarz, S. Schwub, K. Quering, D. Wiedmann, H. W. Höppel und M. Göken. „Life prediction of thermally highly loaded components: modelling the damage process of a rocket combustion chamber hot wall“. In: *CEAS Space Journal* 1 (2011).
- [24] K. Siebertz, D. van Bebber und T. Hochkirchen. *Statistische Versuchsplanung*. 2. Aufl. Germany: Springer: Berlin/Heidelberg, 2017. ISBN: 978-3-662-55743-3.
- [25] „Signal-Based Metamodels for Predictive Reliability Analysis and Virtual Testing“. In: *Advances in Science, Technology and Engineering Systems Journal* 3.1 (2018), S. 342–347.
- [26] A. Sobester, S. J. Leary und A. J. Keane. „A parallel updating scheme for approximating and optimizing high fidelity computer simulations“. In: *Structural and Multidisciplinary Optimization* 27 (2004), S. 371–383.
- [27] A. Sobester, S. J. Leary und A. J. Keane. „A parallel updating scheme for approximating and optimizing high fidelity computer simulations“. In: *Structural and Multidisciplinary Optimization* 27 (2004), S. 371–383.
- [28] W. Zhao und Z. Qiu. „An Efficient Response Surface Method and Its Application to Structural Reliability and Reliability-Based Optimization“. In: *Finite Elem. Anal. Des.* 67 (Mai 2013), S. 34–42.