
Neue Konzepte für die Planung und Analyse von Computereperimenten

Dissertation

zur Erlangung des Grades
eines Doktors der Naturwissenschaften
der Technischen Universität Dortmund



Der Fakultät Statistik
der Technischen Universität Dortmund

vorgelegt von
Thomas Mühlenstädt

Essen 2010

1. Gutachter: PD Dr. Sonja Kuhnt

2. Gutachter: Prof. Dr. Joachim Kunert

Tag der mündlichen Prüfung: 17. Dezember 2010

INHALTSVERZEICHNIS

1	Einführung	1
2	Delaunaytriangulation	5
3	Raumfüllende Versuchspläne	9
3.1	Bekannte Versuchspläne	10
3.1.1	Latin Hypercubes	10
3.1.2	Maximin Versuchspläne	11
3.1.3	Uniform Designs	13
3.1.4	Weitere Kriterien	16
3.2	Neue simplexbasierte Versuchspläne	18
3.2.1	Der „NRSM“ Algorithmus	19
3.2.2	Abwandlungen des NRSM Algorithmus	27
3.2.3	Rangversion der NRSM basierten Versuchspläne	31
3.3	Vergleich und Diskussion	32
4	Modellierung von Computerexperimenten	39
4.1	Kriging	40
4.1.1	Schätzung der Parameter	43

4.1.2	Vorhersage beim Kriging	44
4.1.3	Unsicherheitsvorhersage der Schätzung	45
4.1.4	Flexibilität des Krigingansatzes	46
4.2	Radiale Basis Funktion Interpolation	47
4.2.1	Allgemeine radiale Basisfunktion	47
4.2.2	Thin Plate Spline	50
4.2.3	Flexibilität der Interpolation mit radialen Basisfunktionen	51
4.3	Inverse Distance Weighting	52
4.4	Weitere Methoden	55
5	Kerninterpolation	57
5.1	Allgemeine Idee: Kombiniere lokal angepasste lineare Funktionen	57
5.2	Auswahl der Gewichtsfunktion	58
5.3	Eigenschaften der Kerninterpolation	60
5.4	Unsicherheitsvorhersage bei der Kerninterpolation	67
5.5	Simulationen und Anwendungen	68
5.5.1	Analytische Funktionen	70
5.5.2	Anwendungsdatensätze	96
5.6	Schlussfolgerungen	101
6	Validierung von Computerexperimenten	107
6.1	Anwendungsbeispiel	108
6.2	Test auf Korrektheit des Computerexperiments	111
6.2.1	Permutationstests	112

6.2.2	Permutationstests für Computerexperimente	114
6.2.3	Simulationsstudie und Anwendung	118
6.3	Lokale Auswahl von Computerexperimenten	124
6.3.1	Anwendung	127
6.3.2	Aufteilung des Versuchsraums	128
6.3.3	Anwendung und Interpretation	130
7	Fazit	133
A	Symbolverzeichnis	145
B	Berechnung der Umkugel	149
C	Datensätze aus Kapitel 5 zur Kerninterpolation	151
D	Datensatz aus Kapitel 6	169

KAPITEL 1

EINFÜHRUNG

In der vorliegenden Arbeit werden Methoden für die Planung, Analyse und Validierung von Computereperimenten erarbeitet. Computereperimente sind ein relativ neues Feld der Statistik, bei dem für ein real existierendes Phänomen computergestützte Simulationen vorhanden sind, die im Idealfall echte Experimente ersetzen sollen. Die Simulationen haben dabei den Vorteil dass sie oftmals kostengünstiger und schneller sind als echte Experimente. Dennoch haben die Simulation meistens einen so hohen Rechenbedarf, dass nicht beliebig viele Läufe der Simulation zur Verfügung stehen. Ziel der Statistik in diesem Zusammenhang ist es, die Simulationen so zu nutzen, dass möglichst viele Informationen aus einer begrenzten Anzahl an Simulationsläufen erhalten werden. Dies beinhaltet eine adäquate Planung der Simulationen und, da üblicherweise nicht alle möglichen Kombinationen von Einflussgrößen durchprobiert werden können, das Bereitstellen von Hilfsmodellen, die Vorhersagen für Kombinationen von Einflussgrößen machen, die nicht beobachtet wurden. Da die Simulationen als ein Modell für das reale Experiment interpretiert werden können, werden die Hilfsmodelle auch oft als Metamodell bezeichnet. Die Simulationen haben fast immer Eigenschaften, die sehr unterschiedlich zu konventionellen, echten Experimenten sind: Die Simulationen sind deterministisch und haben meistens eine relativ hohe Komplexität. Dies macht es erforderlich, Methoden zu verwenden, die auf diese Situation angepasst sind. Ein wichtiger Punkt bei der Behandlung von Computereperimenten ist der Vergleich mit echten Experimenten, falls möglich. Liefert die Simulation keine gute Annäherung an die echten

Experimente ist die Nützlichkeit der Simulation zumindest fragwürdig.

Das Ziel dieser Arbeit ist es, in den drei Bereichen Planung, Modellierung und Validierung von Computereperimenten neue Methoden vorzustellen, die teils erheblich von vorhandenen Methoden abweichen und die Güte dieser Methoden zu untersuchen.

Im zweiten Kapitel werden dafür einige Konzepte aus dem mathematischen Gebiet der Geometrie vorgestellt, die im dritten Kapitel für die Entwicklung von Versuchsplänen und im fünften Kapitel für die Herleitung einer Interpolationsmethode von Bedeutung sind. Hier ist insbesondere die Delaunaytriangulation für das weitere Vorgehen wichtig.

Im dritten Kapitel wird eine neue Art raumfüllender Versuchsplan vorgestellt, der auf dem Konzept der Delaunaytriangulation aufbaut und durch ein sehr einfaches Metamodell (eine stückweise lineare stetige Funktion) motiviert ist. Diese Art Versuchspläne ist konkurrenzfähig zu herkömmlichen raumfüllenden Versuchsplänen, hat aber den Vorteil, dass größtenteils keine aufwendigen Optimierungsalgorithmen verwandt werden. Darüberhinaus ist die Konstruktion von Versuchsplänen mittels der neu vorgestellten Systematik sehr schnell und es sind Versuchspläne verfügbar, die nicht ein verallgemeinertes Rechteck als Versuchsraum voraussetzen.

Da das Metamodell der stückweise linearen stetigen Funktionen sehr einfach und in seiner eigentlichen Form nicht zur Modellierung von Computereperimenten geeignet ist, wird in Kapitel 5 ein Metamodell vorgestellt, das diese Methode verfeinert und die Nachteile der stückweise linearen stetigen Funktion beseitigt. Das neu hergeleitete Metamodell hat Ähnlichkeit zum Inverse Distance Weighting, übertrifft diese Interpolationsmethode bzgl. der Vorhersagegüte jedoch und ist insbesondere für kleine Stichprobenumfänge konkurrenzfähig zu etablierten Methoden wie dem Kriging oder der Interpolation mit radialen Basisfunktionen, die in Kapitel 4 vorgestellt werden.

In den meisten Fällen sollen Computereperimente als vollwertiger Ersatz für echte Experimente fungieren. Deshalb ist es wesentlich, die Güte der Vorhersage der Computereperimente zu überprüfen. Dafür wird in Kapitel 6 ein Permutationstest vorgestellt, der die Hypothese, dass das Computereperiment echte Experimente im Mittel richtig

vorhersagt, überprüft. Für die Situation, dass mehrere Simulation für dieselbe Situation vorliegen, aber keine der vorliegenden Simulationen ein perfekter Ersatz für die echten Experimente ist, wird im Anschluss eine Methodik vorgestellt, um lokal die beste Simulationsvorhersage auszuwählen.

Ein Fazit fasst die gefundenen Ergebnisse zusammen.

DELAUNAYTRIANGULATION

Da die Delaunaytriangulation an mehreren Stellen dieser Arbeit verwendet wird, wird sie hier vorab eingeführt. Für eine gute Erklärung der Delaunaytriangulation sei auf Fortune (2004) und Okabe et al. (2000) verwiesen. Gegeben sei die Punktmenge $P_n = \{\vec{x}_1, \dots, \vec{x}_n\}$ im \mathbb{R}^d mit $n \geq d + 1$. Das Ziel einer Triangulation ist es, die konvexe Hülle $\text{conv}(P_n)$ der Punktmenge P_n in N Simplexe S_1, \dots, S_N aufzuteilen, so dass, abgesehen von Nullmengen, jeder Punkt der konvexen Hülle genau einmal überdeckt wird und die Menge der Ecken der Simplexe gleich der Menge der Punkte in P_n ist. Die Ecken des Simplex S_j werden mit $\vec{x}_0^j, \dots, \vec{x}_d^j$ bezeichnet und sind Teilmenge von P_n : $\forall j = 1, \dots, N, \forall i = 0 \dots, d : \vec{x}_i^j \in P_n$. In einer Dimension ist eine solche Aufteilung eindeutig, für $d \geq 2$ ist die Aufteilung bis auf den Spezialfall $n = d + 1$ (d.h. es gibt nur einen möglichen Simplex) jedoch nicht eindeutig. Eine oft gewählte Möglichkeit ist die Delaunaytriangulation. Hier wird die Delaunaytriangulation in Bezug auf das sogenannte Umkreis Kriterium definiert. Für jeden beliebigen d -dimensionalen Simplex mit nicht leerem Inhalt kann eine Umkugel berechnet werden:

Definition 2.1 (Umkugel). *Betrachte die Punkte $\vec{x}_0, \dots, \vec{x}_d \in \mathbb{R}^d$ mit der Eigenschaft, dass der Simplex mit Ecken $\vec{x}_0, \dots, \vec{x}_d$ nichtleeren Inhalt hat. Die Umkugel C mit Radius r und Mittelpunkt \vec{c} ist die eindeutig definierte Kugel, für die gilt, dass $\|\vec{c} - \vec{x}_i\| = r, i = 0, \dots, d$.*

Die Berechnung der Umkugel ist im Anhang 1 erklärt. Damit die im folgenden defi-

nierte Delaunaytriangulation eindeutig ist, wird eine Bedingung an die Punktmenge P_n gestellt:

Definition 2.2 (Non-cocircularity). *Die Punktmenge $P_n \subset \mathbb{R}^d$ ist non-cocircular, wenn keine $d + 2$ Punkte aus P_n auf einer d -dimensionalen Kugel liegen.*

Die Delaunaytriangulation hat folgende interessante Eigenschaft, anhand der sie auch definiert werden kann.

Definition 2.3 (Delaunaytriangulation). *Betrachte die non-cocirculäre Punktmenge $P_n = \{\vec{x}_1, \dots, \vec{x}_n\}$, $\vec{x}_i \in \mathbb{R}^d$, $n \geq d + 1$ und die Triangulation $T(P_n)$ mit Simplex S_1, \dots, S_N und zu den Simplex S_j gehörenden Umkugeln C_1, \dots, C_N . $T(P_n)$ ist genau dann eine Delaunaytriangulation, wenn für jeden Simplex S_j , $j = 1, \dots, N$ gilt:*

(D1) *Für alle Simplexe S_j mit Umkugel C_j , Umkugelmittelpunkt \vec{c}_j und Radius r_j gilt:*

$$\|\vec{p} - \vec{c}_j\| \geq r_j \quad \forall \vec{p} \in P_n, \text{ d.h. kein einziger Punkt in } P_n \text{ liegt innerhalb der Umkugeln } C_j.$$

(D2) *Die einzigen Punkte, die auf dem Rand von C_j liegen, sind die, die die Ecken des Simplex S_j darstellen: $\|\vec{p} - \vec{c}_j\| = r_j \Leftrightarrow \vec{p} \in \{\vec{x}_0^j, \dots, \vec{x}_d^j\} \quad \forall \vec{p} \in P_n$.*

Um eine Eindeutigkeit der Delaunaytriangulation zu erreichen, muss die Punktmenge P_n „non-cocircular“ sein (Okabe et al. (2000)), andernfalls gibt es mehrere Triangulation, die die Eigenschaft (D1) erfüllen. Dann liegen nicht mehr nur die Punkte auf dem Rand des Umkreises C_j , die die Ecken des Simplex S_j bilden. Weitere interessante Eigenschaften der Delaunaytriangulation können in Rajan (1994) gefunden werden. Über eine beliebige Triangulation kann eine stückweise lineare Funktion definiert werden:

Bemerkung 2.1 (Stückweise lineare Funktion). *Sei die Punktmenge $P_n = \{\vec{x}_1, \dots, \vec{x}_n\}$, $x_i \in \mathbb{R}^d$ mit dazugehörigen Werten $y_1, \dots, y_n \in \mathbb{R}$ gegeben. $T(P_n)$ sei eine beliebige Triangulation der Punktmenge P_n mit Simplex S_1, \dots, S_N . Dann kann für jeden Simplex S_j mit Ecken $\vec{x}_0^j, \dots, \vec{x}_d^j \in P_n$ und zugehörigen y -Werten*

$y_0^j, \dots, y_d^j \in \{y_1, \dots, y_n\}$ eine lineare Funktion $y^j(x)$ berechnet werden, die die Werte $(\vec{x}_0^j, y_0^j), \dots, (\vec{x}_d^j, y_d^j)$ interpoliert. Werden die Funktionen $y^j(x)$ jeweils lokal innerhalb des Simplex S_j betrachtet, ergibt sich eine innerhalb der konvexen Hülle der Punktmenge P_n eindeutige Funktion $y_T(x)$, die stetig und stückweise linear ist.

Die in Bemerkung 4 beschriebene Funktion wird im folgenden als die zu der Triangulation $T(P_n)$ zugehörigen stückweise lineare stetige Funktion bezeichnet. Rippa (1990) definiert das folgende Glattheitskriterium für stückweise stetige lineare Funktionen:

Definition 2.4 (Glattheitsmaß). *Es sei die Punktmenge $P_n := \{\vec{x}_1, \dots, \vec{x}_n\} \subset \mathbb{R}^d$ mit dazugehörigen Werten $y_1, \dots, y_n \in \mathbb{R}$ gegeben. Für eine beliebige Triangulation $T(P_n)$ mit zugehöriger stückweiser linearer stetiger Funktion $y_T(x)$ definiere das Glattheitsmaß $R(y_T)$ als:*

$$R(y_T) := |y_T|, \text{ wobei } |y_T|^2 := \sum_{j=1}^N |\hat{y}^j|_{S_j}^2$$

und

$$|\hat{y}^j|_{S_j}^2 := \int_{S_j} \left[\left(\frac{\partial y^j}{\partial x_1} \right)^2 + \dots + \left(\frac{\partial y^j}{\partial x_d} \right)^2 \right] dx_1 \dots dx_d.$$

Auch wenn das Glattheitsmaß für beliebige Dimensionen definiert ist, ist die Optimalität der Delaunaytriangulation nur für 2 Dimensionen bekannt:

Satz 2.1 (Rippa (1990)). *Seien die Punktmenge $P_n := \{\vec{x}_1, \dots, \vec{x}_n\} \subset \mathbb{R}^2$ mit dazugehörigen Werten $y_1, \dots, y_n \in \mathbb{R}$ gegeben. y_T bezeichne eine stückweise lineare stetige Funktion zu einer Triangulation T . Ist T die Delaunaytriangulation und \tilde{T} eine beliebige andere Triangulation, dann gilt:*

$$R(y_T) \leq R(y_{\tilde{T}}).$$

Eine geometrische Struktur, die dual zu der Delaunaytriangulation ist, ist das Voronoidiagramm (Okabe et al. (2000)). Das Voronoidiagramm ist eine Aufteilung des \mathbb{R}^d in bis auf Nullmengen disjunkte Teilmengen, die Voronoizellen genannt werden.

Definition 2.5 (Voronoidiagramm). Sei $P_n = \{\vec{x}_1, \dots, \vec{x}_n\} \subset \mathbb{R}^d$ mit $2 \leq n < \infty$ und $d \in \mathbb{N}$ gegeben. Die Voronoizelle V_i zu Punkt \vec{x}_i ist gegeben durch

$$V_i := \{\vec{x} \in \mathbb{R}^d : \|\vec{x} - \vec{x}_i\| \leq \|\vec{x} - \vec{x}_j\| \text{ für } j = 1, \dots, n, j \neq i\}. \quad (2.1)$$

Das Voronoidiagramm $V(P_n)$ ist die Menge der durch P_n erzeugten Voronoizellen:
 $V(P_n) := \{V_1, \dots, V_n\}$.

Die Voronoizellen sind abgeschlossene oder halboffene Polygone im \mathbb{R}^d . Die Ecken der Voronoizelle V_i werden mit \vec{c}_{ij} bezeichnet. Ist ein Punkt \vec{c} gleichzeitig auf dem Rand von zwei Voronoizellen, so sagt man, dass diese benachbart sind. Die Dualität ergibt sich aus folgender Eigenschaft:

Bemerkung 2.2. Es sei $P_n = \{\vec{x}_1, \dots, \vec{x}_n\} \subset \mathbb{R}^d$ mit $d + 1 \leq n < \infty$ und $d \in \mathbb{N}$ mit zugehöriger Delaunaytriangulation $T(P_n)$ gegeben. Bezeichne mit \vec{c}_j den Umkreismittelpunkt für Simplex $S_j \in T(P_n)$ und $C_D := \{\vec{c}_1, \dots, \vec{c}_N\}$ die Menge aller Umkreismittelpunkte. Weiterhin sei

$C_V := \{\vec{c}_{1,1}, \dots, \vec{c}_{1,n_1}, \vec{c}_{2,1}, \dots, \vec{c}_{n,1}, \dots, \vec{c}_{n,n_n}\}$ die Menge aller Ecken der Voronoizellen von P_n . Dann gilt:

$$C_D = C_V. \quad (2.2)$$

Nähere Erklärungen zu diesem Zusammenhang können in Fortune (2004) gefunden werden.

KAPITEL 3

RAUMFÜLLENDE VERSUCHSPÄNE

Für die Planung von Computerexperimenten wurden in den letzten Jahren viele verschiedene Versuchspläne vorgeschlagen, die alle unter dem Stichwort „raumfüllend“ zusammengefasst werden. Da bei einem Computerexperiment üblicherweise kein Zufallsfehler vorliegt, ist die Anwendung der sonst in der Statistik üblichen Versuchspläne zu überdenken. Versuchsplänen wie z.B. D-optimalen Versuchsplänen liegt immer ein Modell mit einem Zufallsfehler zugrunde (Atkinson und Donev (1992)), für das versucht wird eine möglichst gute Schätzung (z.B. der Modellparameter) unter Berücksichtigung des Zufallsfehlers zu erreichen. In einer Situation ohne Zufallsfehler verschiebt sich das Augenmerk: Die Versuche sollen so angeordnet werden, dass der Versuchsraum möglichst gleichmäßig mit Beobachtungen abgedeckt ist. Unterschiede bestehen in der Art, wie der Begriff „gleichmäßig“ definiert wird. Im folgenden werden verschiedene Beispiele für raumfüllende Versuchspläne vorgestellt. Als Versuchsraum wird immer der d -dimensionale Einheitswürfel $\mathfrak{D} = [0, 1]^d$ verwandt. Dabei wird ein Versuchsplan mit n Punkten mit D oder, falls auf Stichprobengröße und Dimension hingewiesen werden soll, mit $D_{n,d} \in [0, 1]^{n \times d}$ bezeichnet und die einzelnen Punkte mit $\vec{x}_1 = [x_{1,1}, \dots, x_{1,d}], \dots, \vec{x}_n = [x_{n,1}, \dots, x_{n,d}]$. Der Raum aller $D_{n,d}$ -Versuchspläne wird mit $\mathfrak{D}_{n,d}$ bezeichnet.

3.1 Bekannte Versuchspläne

3.1.1 Latin Hypercubes

Eine sehr eingängige Klasse von Versuchsplänen sind Latin Hypercubes, die von McKay et al. (1979) vorgeschlagen wurden.

Definition 3.1 (Latin Hypercube). *Ein Latin Hypercube $D = [\vec{x}_1, \dots, \vec{x}_n]'$ in d Dimensionen und $n \in \mathbb{N}$ Lufen erfullt die Bedingung, dass in jeder Spalte die Zahlen $(0, \dots, n-1)/(n-1)$ genau einmal vorkommen, d.h. jede Spalte ist eine Permutation der Zahlen $1, \dots, n$, normiert auf $[0, 1]$.*

Abweichend existiert auch die Normierung auf $[1/(n+1), \dots, n/(n+1)]$. Der ursprungliche Vorschlag von McKay et al. (1979) beinhaltet, dass dann in jedem, durch einen Punkt im Latin Hypercube definierten Teilquadrat ein Punkt aus einer Gleichverteilung uber das jeweilige Quadrat erzeugt wird. Dafur zeigen McKay et al. (1979) folgende Optimalitatseigenschaft. Betrachtet sei die Zufallsvariable $f(\vec{X})$ mit \vec{X} gleichverteilt uber den Einheitswurfel und f eine mebare Funktion. Gesucht ist ein Schatzer fur den Erwartungswert von $f(\vec{X})$. Fur Schatzer der Form $\frac{1}{n} \sum_{i=1}^n f(\vec{x}_i)$ gilt: Falls f monoton in jeder einzelnen Komponente ist, so hat der Schatzer, der die Werte $\vec{x}_1, \dots, \vec{x}_n$ entsprechend einem Latin Hypercube wahlt, kleinere Varianz als ein normaler Monte Carlo Schatzer, der $\vec{x}_1, \dots, \vec{x}_n$ entsprechend einer Gleichverteilung uber dem Versuchsraum wahlt. Die Klasse der Latin Hypercubes ist attraktiv, da fur jeden Versuchsplan in dieser Klasse die eindimensionalen Projektionen aquidistant verteilt sind. Auch wenn die Klasse der Latin Hypercubes insgesamt attraktiv ist, ist durch einen Latin Hypercube alleine noch nicht notwendigerweise abgesichert, dass ein guter raumfullender Versuchsplan vorliegt. Ein Standardbeispiel fur einen ungunstigen Latin Hypercube ist der Versuchsplan, der in jeder Spalte die gleiche Permutation der Zahlen $1, \dots, n$ enthalt. Dies bedeutet anschaulich, dass nur auf der Diagonalen des Versuchsraums Beobachtungen liegen, was offensichtlich nicht raumfullend ist. Nichtsdestotrotz sind in

der Klasse der Latin Hypercubes sehr gute raumfüllende Versuchspläne, weshalb oftmals innerhalb dieser Klasse mittels Gütekriterien und numerischen Suchalgorithmen nach guten raumfüllenden Versuchsplänen gesucht wird.

3.1.2 Maximin Versuchspläne

Johnson et al. (1990) haben Maximin Versuchspläne für Computereperimente vorgeschlagen. Der Grundgedanke dabei ist, dass keine zwei Punkte des Versuchsplans zu nahe im Versuchsraum zusammenliegen sollten. Wird angenommen, dass die zu modellierende Computersimulation stetig ist, würden zwei Versuchsläufe, die zu nahe beieinander liegen keinen weiteren Informationsgewinn ergeben.

Definition 3.2 (Maximin Versuchsplan). *Für einen beliebigen d -dimensionalen Versuchsplan D mit n Läufen ist der minimale Abstand von D gegeben durch*

$$Mm(D) := \min_{\vec{x}_i, \vec{x}_j \in D} \|\vec{x}_i - \vec{x}_j\|. \quad (3.1)$$

Ein Maximinversuchsplan maximiert dieses Gütekriterium unter allen möglichen Versuchsplänen:

$$D_{n,d}^{Mm} := \arg \max_{D \in \mathcal{D}_{n,d}} Mm(D). \quad (3.2)$$

In den meisten Fällen wird wie hier als Norm die euklidische Norm verwendet, jedoch kann dies Konzept auf andere Normen wie die L_∞ Norm oder die L_1 Norm erweitert werden (siehe z.B. van Dam et al. (2005)). Johnson et al. (1990) argumentieren, dass Maximinversuchspläne in einem asymptotischen Sinn optimale Versuchspläne sind für das Schätzen von Korrelationsparametern in einem Krigingmodell mit geringen Korrelationen.

Problematisch an diesem Ansatz ist, dass zwei unterschiedliche Versuchspläne den gleichen minimalen Abstand haben können. Dies kann insbe-

sondere dann ein Problem sein, wenn bei einem Versuchsplan der minimale Abstand nur für ein Punktepaar angenommen wird, für einen anderen jedoch für mehrere Punktepaare. Das Maximin Kriterium $Mm(D)$ beurteilt zwei solche Versuchspläne gleich, auch wenn intuitiv verständlich ist, dass ein Versuchsplan, der den minimalen Abstand nur für ein Punktepaar annimmt besser ist, als einer, der den minimalen Abstand für mehrere Punktepaare annimmt. Als Ausweg aus dieser Situation haben Morris und Mitchell (1995) ein abgeändertes Gütekriterium vorgestellt:

Definition 3.3 (Maximin Versuchsplan nach Morris und Mitchell (1995)). *Für einen Versuchsplan D mit $Mm(D) > 0$ sei m die Anzahl an unterschiedlichen Punkteabständen in dem Versuchsplan und d_1, \dots, d_m seien die angenommenen euklidischen Abstände. Weiter sei J_k die Anzahl, mit der der Abstand d_k angenommen wird. Dann ist das Maximinkriterium nach Morris und Mitchell (1995) gegeben durch*

$$\Phi_p(D) := \left(\sum_{j=1}^m J_j d_j^{-p} \right)^{(1/p)}, p \in \mathbb{N}, \quad (3.3)$$

Ein Φ_p -Maximinversuchsplan minimiert dieses Gütekriterium unter allen möglichen Versuchsplänen:

$$D_{n,d}^{\Phi_p} := \arg \min_{D \in \mathcal{D}_{n,d}} \Phi_p(D). \quad (3.4)$$

Das Gütekriterium $\Phi_p(D)$ hat nicht den eben beschriebenen Nachteil. Jedoch muss der Parameter p gewählt werden. Für kleine Werte (z.B. $p = 1$) liefert das Kriterium nicht notwendigerweise ähnliche Ergebnisse wie das Maximinkriterium $Mm(D)$, ist jedoch leicht numerisch zu optimieren. Für große Werte (z.B. $p = 100$) liegt eine bessere Beurteilung vor, jedoch wird die numerische Optimierung schwieriger. Morris und Mitchell (1995) schlagen vor, das Kriterium $\Phi_p(D)$ innerhalb der Klasse der Latin Hypercubes mittels eines Simulated Annealing Algorithmus zu optimieren.

3.1.3 Uniform Designs

Der Grundgedanke hinter Uniform Designs ist, eine Stichprobe über dem Versuchsraum zu konstruieren, deren empirische Verteilungsfunktion möglichst nahe an der theoretischen Verteilungsfunktion einer Gleichverteilung über den Versuchsraum ist. Formal werden unterschiedliche Versuchspläne mittels einer Teststatistik („Diskrepanz“) auf Gleichverteilung beurteilt und der Versuchsplan gewählt, der die Diskrepanz minimiert. Intuitiv bietet sich die Teststatistik des Kolmogorov Smirnov Tests an, die in diesem Zusammenhang oftmals Star Diskrepanz genannt wird:

Definition 3.4 (Star Diskrepanz). Sei $F_n(\vec{x})$ die empirische Verteilungsfunktion des Versuchsplans D an der Stelle \vec{x} , d.h. $F_n(\vec{x}) := \frac{1}{n} \sum_{i=1}^n I_{(-\infty, \vec{x}]}(\vec{x}_i)$, wobei $I_{(-\infty, \vec{x}]}(\vec{x}_i)$ die Indikatorfunktion für die Menge der Punkte $\{\vec{x} \in \mathbb{R}^d | \vec{x} \leq \vec{x}_i\}$ ist. Dann ist die Star Diskrepanz eines Versuchsplans definiert als

$$SD(D) := \sup_{\vec{x} \in \mathfrak{D}} \|F_n(\vec{x}) - Vol([0, \vec{x}])\|, \quad (3.5)$$

wobei $Vol([0, \vec{x}]) = \prod_{k=1}^d x_k$ das Lebesguemaß der Menge $[0, \vec{x}]$ ist.

Interessant ist, dass in dem Kontext von Monte Carlo Integration der Integrationsfehler durch die Koksma Hlawka Ungleichung (Niederreiter (1992)) abgeschätzt werden kann:

$$|I(f) - Q(f)| \leq SD(D)V(f), \quad (3.6)$$

wobei $I(f) := \int_{\mathfrak{D}} f(\vec{x}) d\vec{x}$, $Q(f) := \frac{1}{n} \sum_{i=1}^n f(\vec{x}_i)$ und $V(f)$ die Variation von f im Sinne von Hardy und Krause (siehe Niederreiter (1992), S. 19). Definiere dafür zunächst die Variation im Sinne von Vitali als

$$V^{(d)}(f) = \int_0^1 \cdots \int_0^1 \left| \frac{\partial^d f}{\partial x_1 \cdots \partial x_d} \right| dx_1, \dots, dx_d. \quad (3.7)$$

Die benötigten Ableitungen in der Variation nach Vitali sollen stetig über den Einheitswürfel sein. Es gibt eine allgemeinere Formel für den Fall, dass die notwendige Differenzierbarkeit nicht vorliegt (Niederreiter (1992), S. 19), da diese jedoch formal

kompliziert ist, für das Verständnis aber nicht unbedingt weiterhilft wird sie hier nicht angegeben. Für $1 \leq k \leq d$ betrachte $V^{(k)}(f; j_1, \dots, j_k)$, die Einschränkung von $V^{(d)}(f)$ auf die Dimensionen x_{j_1}, \dots, x_{j_k} . Dann ist die Variation im Sinne von Hardy und Krause definiert als

$$V(f) = \sum_{k=1}^d \sum_{1 \leq j_1 < \dots < j_k \leq d} V^{(k)}(f; j_1, \dots, j_k). \quad (3.8)$$

Die inhaltliche Aussage dieser Ungleichung ist, dass die Güte der Schätzung des Integrals $I(f)$ allein durch den Versuchsplan kontrolliert werden kann, unabhängig von der unbekanntem Funktion f . Auch wenn sich die Star Diskrepanz gut für theoretische Herleitungen eignet und in der asymptotischen Theorie der Uniform Sequenzen (Kuipers und Niederreiter (2006)) eine große Rolle spielt, hat sie den großen Nachteil, dass sie numerisch schwer zu berechnen ist (Fang et al. (2000)). Die Star Diskrepanz beschränkt sich weiterhin bei der Beurteilung der Güte des Versuchsplans auf Intervalle der Form $[0, \vec{x}]$, ist also am Nullpunkt „verankert“ und somit nicht invariant unter Rotationen. Aus diesem Grund wurden verschiedene Alternativen vorgeschlagen, von denen die Wrap Around Diskrepanz (Hickernell (1998a), Hickernell (1998b), Fang et al. (2006)) hier näher erklärt werden soll. Dafür wird die folgende Indexnotation verwandt: Der Vektor $\vec{x}_u, u \subset \{1, \dots, d\}$ ist der (Teil-) Vektor des Vektors \vec{X} , der nur die Koordinaten in u enthält

Definition 3.5 (Wrap Around Diskrepanz). *Die Wrap Around Diskrepanz eines Versuchsplans D ist gegeben durch*

$$WD(D)^2 := \sum_{u \neq \emptyset} \int_{C^u} \int_{C^u} \left| \frac{N(D_u, J(\vec{x}_u, \vec{y}_u))}{n} - Vol(J(\vec{x}_u, \vec{y}_u)) \right|^2 d\vec{x}_u d\vec{y}_u. \quad (3.9)$$

$Vol(J)$ bezeichnet wieder das Lebesguemaß der Menge J . Die Summation $\sum_{u \neq \emptyset}$ bezieht sich auf alle möglichen nichtleeren Teilmengen $u \subset \{1, \dots, d\}$. $N(D_u, J(\vec{x}_u, \vec{y}_u))$ ist die Anzahl an Punkten des Versuchsplans, die innerhalb der Menge $J(\vec{x}_u, \vec{y}_u)$ liegen. Die Menge $J(\vec{x}_u, \vec{y}_u)$ ist dabei als kartesisches Produkt von eindimensionalen Mengen

$j(x_k, y_k)$ definiert:

$$J(\vec{x}_u, \vec{y}_u) := \bigotimes_{i \in u} j(x_i, y_i),$$

mit

$$j(x_i, y_i) = \begin{cases} [x_i, y_i], & \text{falls } x_i < y_i \\ [0, y_i) \cup [x_i, 1) & \text{falls } x_i > y_i. \end{cases}$$

Die Wrap Around Diskrepanz kann analytisch wie folgt ausgewertet werden:

$$WD(D)^2 = - \left(\frac{4}{3}\right)^d + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \prod_{k=1}^d \left[\frac{3}{2} - |x_{id} - x_{jd}|(1 - |x_{id} - x_{jd}|)\right]. \quad (3.10)$$

Ein Versuchsplan mit minimaler Wrap Around Diskrepanz unter allen Versuchsplänen der Menge $\mathfrak{D}_{n,d}$ wird Uniform Versuchsplan bzgl. der Wrap Around Diskrepanz genannt:

$$D_{n,d}^{WD} = \arg \min_{D \in \mathfrak{D}_{n,d}} WD(D)^2. \quad (3.11)$$

Die Warp Around Diskrepanz hat viele gute Eigenschaften: Sie ist numerisch leicht zu berechnen, ist invariant unter Rotationen und Umordnen der Indizes. Fang und Ma (2001) zeigen, dass der Erwartungswert der Wrap Around Diskrepanz von zufälligen Latin Hypercubes niedriger ist als der Erwartungswert der Wrap Around Diskrepanz bei einer gleichgroßen über den Versuchsraum gleichverteilten Zufallsstichprobe. Auch gibt es eine zu 3.6 analoge Abschätzung:

$$|I(f) - Q(f)| \leq WD(D)V^*(f), \quad (3.12)$$

vgl. Hickernell (1998a) und Hickernell (1998b), wobei hier die Variation im Sinne von Hardy und Krause durch eine auf die Warp Around Diskrepanz angepasste Variation ersetzt wurde. Für einige spezielle Situationen sind analytische Lösungen für die Minimierung der Wrap Around Diskrepanz bekannt (Fang et al. (2005)), im Allgemeinen muss die Wrap Around Diskrepanz jedoch numerisch minimiert werden. Dies kann z.B. wieder mittels Simulated Annealing in der Klasse der Latin Hypercubes geschehen.

3.1.4 Weitere Kriterien

Es gibt eine Vielzahl an weiteren Kriterien für die Bestimmung der Güte eines raumfüllenden Versuchsplans. Minimax Versuchspläne (Johnson et al. (1990)) haben eine große Ähnlichkeit zu Maximin Versuchsplänen, da sie ebenfalls auf Punktabständen basieren. Ein Minimax Versuchsplan minimiert das Kriterium

$$mM(D) := \max_{\vec{x} \in \mathfrak{D}} \min_{\vec{x}_i \in D} \|\vec{x}_i - \vec{x}\|. \quad (3.13)$$

In der Praxis hat sich das Minimax Kriterium nicht durchgesetzt, da es schwer zu berechnen ist (die Optimierung über alle $\vec{x} \in \mathfrak{D}$ ist aufwendig und hat für die euklidische Norm $\|\cdot\|$ einen starken Bezug zum Voronoidiagramm. Allerdings existieren tabellierte Minimax Versuchspläne für die Klasse der Latin Hypercubes in zwei Dimensionen (van Dam (2005)).

Kriterien, die eher einem Bayes Ansatz folgen sind das Maximum Entropie Kriterium (Shewry und Wynn (1987)) und das Integrated Mean Square Error Kriterium (Sacks et al. (1989)). Beide Kriterien nehmen an, dass das Computereperiment über Kriging modelliert wird, wobei auch die Korrelationsfunktion sowie eine apriori-Verteilung für die vorhandenen Parameter angegeben werden müssen. Dies und die anspruchsvolle numerische Optimierung haben dazu geführt, dass diese (theoretisch interessanten) Kriterien in der Praxis nicht oft Anwendung finden. Für genauere Details sei auf Santner et al. (2003) verwiesen.

Ebenfalls Anwendung finden gelegentlich sogenannte Uniform Sequences (Lemieux (2009), Kuipers und Niederreiter (2006)). Dabei ist das Ziel, eine Folge von Punkten zu erstellen, so dass im Grenzwert gegen Unendlich die empirische Verteilungsfunktion der Punktfolge mit der theoretischen Verteilungsfunktion einer Gleichverteilung übereinstimmt. Die Asymptotik für diese Folgen wird meistens über die Star Diskrepanz betrachtet. Oft verwendete Uniform Sequences sind die Sobol' Sequenz, die Halton Sequenz oder die Niederreiter Sequenz. Ein Nachteil dieser Sequenzen ist jedoch oftmals, dass sie schlechte Projektionseigenschaften haben und bei fester Stichprobengröße nicht

so effizient sind wie ein für die Stichprobengröße n optimierter Versuchsplan.

3.2 Neue simplexbasierte Versuchspläne

Im Gegensatz zu den eben beschriebenen Ansätzen zur Erstellung eines raumfüllenden Versuchsplans soll hier eine Methode vorgestellt werden, die nicht auf der Optimierung eines Gütekriteriums beruht. Die Methode ist durch ein einfaches Metamodell für Computereperimente motiviert: Wird an die Beobachtungen eine stückweise lineare, stetige Funktion (ein Polygon) angepasst, die die Daten interpoliert (siehe Kapitel 2.1, Definition 2.1), kann diese Funktion als eine erste Näherung für die unbekannte Funktion betrachtet werden, die den Output des Computereperiments erzeugt. Mit der Konvention, dass die „Ecken“ des Polygons in den Beobachtungspunkten liegen ist ein solches Polygon für den Fall einer eindimensionalen Einflussgröße eindeutig definiert. In höheren Dimensionen muss jedoch zuerst eine Triangulation bestimmt werden, die angibt, welche Punkte zu einem Simplex verbunden werden. Hier wird die in Kapitel 2 erläuterte Delaunaytriangulation verwendet. Die Vorhersagegüte eines solchen (einfachen) Modells hängt entscheidend von den Simplexen in der Triangulation ab. Liegt keine Vorinformation über die unbekannte, datengenerierende Funktion vor, ist es sinnvoll, die Simplexe so zu wählen, dass sie möglichst gleich sind in Bezug auf Größe und Form. Weiterhin ist es sinnvoll, die Simplexe möglichst so zu konstruieren, dass sie nahezu regulär sind, d.h. die Ecken der Simplexe gleiche Punkt zu Punkt Abstände haben.

Das Problem der Auswahl der Versuchspunkte im Sinne der obigen Argumentation ist also äquivalent dazu, die Punkte so auszuwählen, dass die Delaunaytriangulation der Punkte möglichst gleichmässig und regulär ist.

In zwei Dimensionen ist es leicht ein Punktegitter zu konstruieren, so dass die Delaunaytriangulation der Punkte (abgesehen von den Randbereichen) nur aus regulären Dreiecken besteht. Dieses Resultat kann jedoch nicht auf höhere Dimensionen verallgemeinert werden. In drei Dimensionen besteht die einzige reguläre Partition des \mathbb{R}^3 aus Würfeln (Coxeter (1963), S. 68). In höheren Dimensionen wurde kein entsprechendes

Ergebnis gefunden, jedoch legen die numerisch erzielten Ergebnisse dieser Arbeit nahe, dass in höheren Dimensionen keine reguläre Triangulation existiert. Aufgrund dieser qualitativen Unterscheidung werden im Folgenden nur Versuchspläne mit Dimension $d \geq 3$ betrachtet.

Ein logischer Ausweg ist es, eine Triangulation zu konstruieren, die aus möglichst regulären und möglichst ähnlichen Simplexen besteht. Dazu wird im folgenden ein Algorithmus (NRSM, Nearly Regular Simplex Mesh) vorgeschlagen, der sukzessive zu bereits existierenden Punkten neue Punkte hinzufügt, ausgehend von einem regulärem Simplex als Startsimplex. Die Eckpunkte der Simplexe werden dann in einem späteren Schritt als potentielle Versuchspunkte im \mathbb{R}^d interpretiert.

3.2.1 Der „NRSM“ Algorithmus

Der NRSM Algorithmus benötigt zwei Dateien: In P werden die Ecken der Simplexe als Punkte im \mathbb{R}^d zeilenweise gespeichert. In S wird die Delaunaytriangulation gespeichert. Üblicherweise werden dabei die (Zeilen-) Indizes der Punkte in P , die zu einem Simplex gehören, zeilenweise in S gespeichert. In d Dimensionen mit n Punkten und N Simplexen ist also $P \in \mathbb{R}^{n \times d}$ und $S \in \mathbb{N}^{N \times (d+1)}$. Am Anfang besteht P aus den $d + 1$ Punkten eines regulären Simplex. Die Konstruktion eines regulären Simplex in beliebiger Dimension wird angegeben. Die Triangulation $T(P)$ enthält zu Beginn nur eine Zeile mit den Indizes $1, \dots, d + 1$. Beginnend mit dem Startsimplex schlägt der NRSM Algorithmus einen neuen Punkt $\vec{x}_{neu}(i, j)$ vor, aufbauend auf dem Simplex S_j und der Ecke \vec{x}_i^j von S_j . Existiert bisher kein Punkt in einer Umgebung von $\vec{x}_{neu}(i, j)$, fügt der Algorithmus den Punkt $\vec{x}_{neu}(i, j)$ als Punkt Nr. $n + 1$ zu P hinzu. Andernfalls wird $\vec{x}_{neu}(i, j)$ gelöscht. Die Umgebung um $\vec{x}_{neu}(i, j)$, in der keine anderen Punkte akzeptiert werden, wird dabei wie folgt definiert: $\{\vec{x} \in \mathbb{R}^d : \|\vec{x} - \vec{x}_{neu}(i, j)\| \leq L(i, j)\}$ mit $L(i, j) > 0$ gegeben. Die genaue Wahl der Funktionen $\vec{x}_{neu}(i, j)$ und $L(i, j)$ ist dabei von großer Bedeutung. Der Punkt $\vec{x}_{neu}(i, j)$ wird vorgeschlagen und nur dann zu der

Punktmenge P hinzugefügt, wenn in einer Kugel um $\vec{x}_{neu}(i, j)$ mit Radius $L(i, j)$ kein weiterer Punkt existiert. Dabei kann der Radius entweder konstant für alle Punkte gewählt werden oder abhängig von dem aktuellen Simplex. Wird ein Punkt $\vec{x}_{neu}(i, j)$ zu P hinzugefügt, wird die Delaunaytriangulation aktualisiert und in zwei Schritten sortiert. Zuerst werden die Indizes in jeder Zeile der Größe nach in absteigender Reihenfolge geordnet. Dann werden die Zeilen entsprechend dem ersten (größten) Index in den Zeilen angeordnet, wobei Bindungen entsprechend den verbleibenden Indizes aufgebrochen werden. Dieses Sortieren ist notwendig, um zu gewährleisten, dass die Punktwolke gleichmässig von innen nach aussen konstruiert wird. Ein Beispiel für das Sortieren ist im folgenden gegeben:

$$\begin{pmatrix} 3 & 1 & 4 \\ 3 & 4 & 6 \\ 5 & 1 & 2 \\ 1 & 3 & 2 \\ 2 & 6 & 5 \end{pmatrix} \rightarrow \begin{pmatrix} 4 & 3 & 1 \\ 6 & 4 & 3 \\ 5 & 2 & 1 \\ 3 & 2 & 1 \\ 6 & 5 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 2 & 1 \\ 4 & 3 & 1 \\ 5 & 2 & 1 \\ 6 & 4 & 3 \\ 6 & 5 & 2 \end{pmatrix}$$

NRSM Algorithmus für die Konstruktion von Punktwolken

1. $i := 0, j := 1, N \in \mathbb{N}$.
2. Ausgehend von Simplex S_j mit Ecken $\vec{x}_0^j, \dots, \vec{x}_d^j$ und $\vec{x}_i^j \in \{\vec{x}_0^j, \dots, \vec{x}_d^j\}$, berechne $\vec{x}_{neu}(i, j)$ und $L(i, j)$.
3. Wenn $\forall \vec{x} \in P : \|\vec{x}_{neu}(i, j) - \vec{x}\| > L(i, j)$: Füge $\vec{x}_{neu}(i, j)$ zu P hinzu und aktualisieren die Delaunaytriangulation T . Sortiere T entsprechend dem oben beschriebenen Verfahren.
4. Falls $i = d$ dann: $j := j + 1, i := 0$ sonst: $i := i + 1$.
5. Falls $\#P < N$ dann: Gehe zu 2. sonst: Stop.

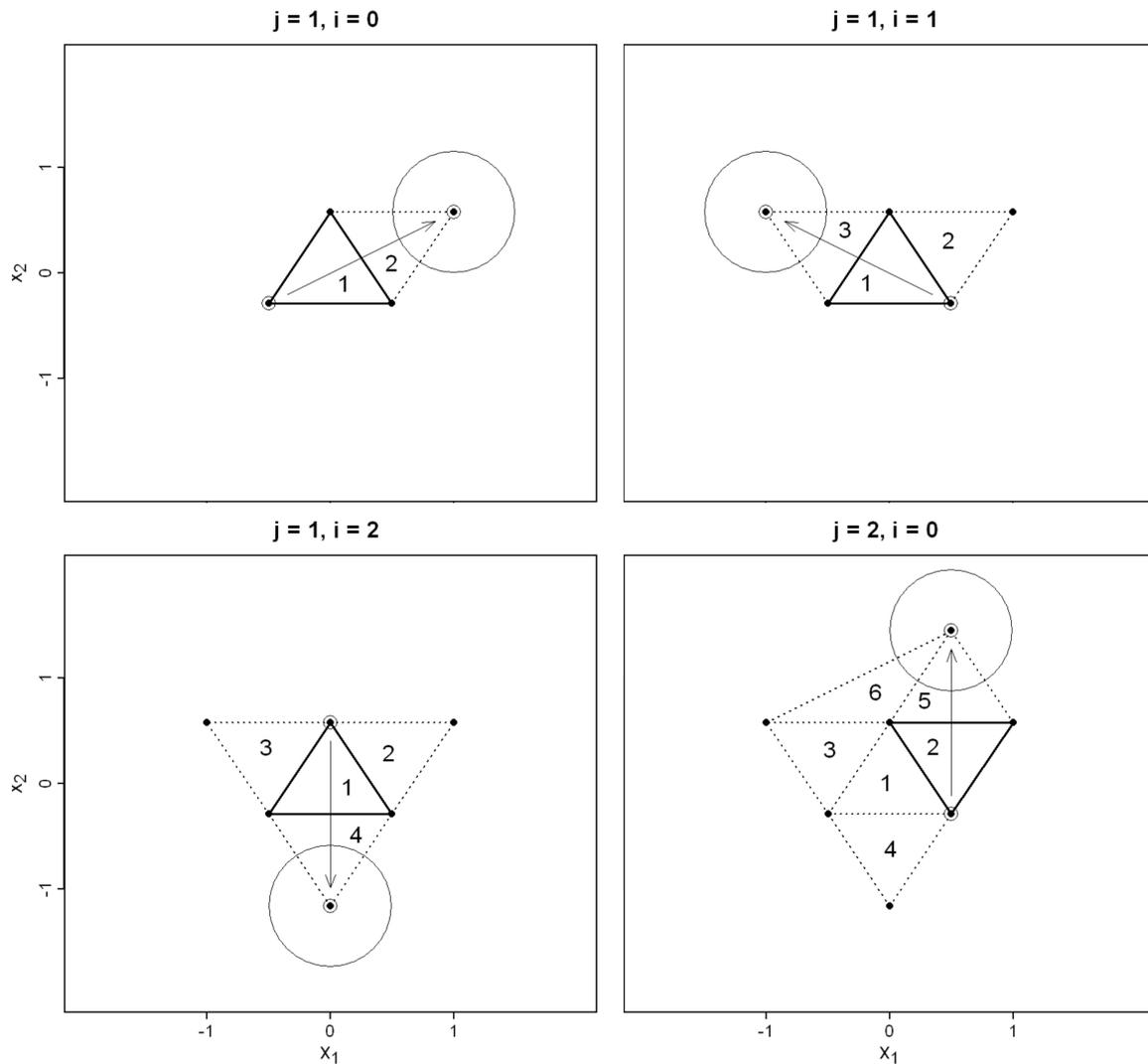


Abbildung 3.1: Schematische Darstellung des NRSM Algorithmuses in 2 Dimensionen. Hier sind die ersten 4 Schritte abgebildet. Die Kreise repräsentieren den Bereich um einen Punkt $\vec{x}_{neu}(i, j)$, in dem keine anderen Punkte erlaubt sind. Das durchgezogene Dreieck ist der Simplex, der gerade bearbeitet wird und der Pfeil zeigt die Ecke an, die gerade zur Erzeugung eines neuen Punktes genutzt wird. Die Zahl in jedem Dreieck ist der Zeilenindex des Dreiecks in der Matrix S .

Der NRSM Algorithmus ist graphisch in 2 Dimensionen in Abbildung 3.1 beschrieben. Die Eigenschaften der resultierenden Punktwolke werden hauptsächlich durch die Wahl von $\vec{x}_{neu}(i, j)$ und $L(i, j)$ beeinflusst. Der Punkt $\vec{x}_{neu}(i, j)$ bestimmt die Form

und Größe eines neuen Simplex $\{\vec{x}_m^j | m = 0, \dots, d, m \neq i\} \cup \vec{x}_{neu}(i, j)$. Die Funktion $L(i, j)$ bestimmt den Bereich um $\vec{x}_{neu}(i, j)$, in dem keine anderen Punkte akzeptiert werden.

Startsimplex

Auch wenn im Allgemeinen keine reguläre Triangulation möglich ist, soll zumindest der Startsimplex des NRSM Algorithmus regulär sein. Die Kantenlänge des Startsimplex soll 1 betragen und alle Punkte sollen so liegen, dass ein Punkt im Nullpunkt liegt und alle anderen nichtnegative Koordinaten haben. Für $d = 1, 2, 3$ sind reguläre Simplexe gegeben durch:

$$\begin{aligned} S_1 &= \{s_0 = 0, s_1 = 1\} \\ S_2 &= \{\vec{s}_0 = 0, \vec{s}_1 = [1, 0]', \vec{s}_2 = [0.5, \sqrt{3/4}]'\} \\ S_3 &= \{\vec{s}_0 = 0, \vec{s}_1 = [1, 0, 0]', \vec{s}_2 = [0.5, \sqrt{3/4}, 0]', \\ &\quad \vec{s}_3 = [0.5, \sqrt{1/12}, \sqrt{2/3}]'\} \end{aligned} \tag{3.14}$$

Für $d > 3$ kann folgende Bemerkung zur Erzeugung regulärer Startsimplexe genutzt werden, welche durch vollständige Induktion bewiesen werden kann.

Bemerkung 3.1. Die $(d + 1)$ Punkte $\vec{s}_0, \dots, \vec{s}_d$ mit $\vec{s}_0 = 0$ und für $k > 0, \vec{s}_k = (s_{k,1}, \dots, s_{k,d})'$ mit

$$s_{k,i} = \begin{cases} \frac{1}{\sqrt{2i(i+1)}}, & i < k; \\ \sqrt{\frac{k+1}{2k}}, & i = k; \\ 0, & i > k. \end{cases} \tag{3.15}$$

bilden einen d -dimensionalen regulären Simplex mit Seitenlänge 1.

Auswahl für $\vec{x}_{neu}(i, j)$ und $L(i, j)$

Die Vorschlagsfunktion $\vec{x}_{neu}(i, j)$ sollte eine selbstregulierende Eigenschaft haben: Ausgehend von einem Simplex S_j wird ein neuer Simplex konstruiert, indem eine Ecke von S_j weggelassen wird und eine neue Ecke hinzugefügt wird. Die Vorschlagsfunktion $\vec{x}_{neu}(i, j)$ sollte die neue Ecke so hinzufügen, dass der neue Simplex regulärer ist bzgl. Größe und Form als der alte Simplex. Allgemein sollte eine Vorschlagsfunktion in dem folgenden Sinne „zulässig“ sein:

Definition 3.6 (Zulässige Vorschlagsfunktion). *Gegeben sei ein Simplex S_j mit Ecken $\{\vec{x}_k | k = 0, \dots, d\}$. Eine Vorschlagsfunktion konstruiert einen neuen Simplex S_{neu} indem eine Ecke \vec{x}_i von S_j entfernt wird und eine neue Ecke $\vec{x}_{neu}(i, j) \neq \vec{x}_i$ hinzugefügt wird: $S_{neu} = \{\vec{x}_{neu}(i, j), \vec{x}_k, k = 0, \dots, d, k \neq i\}$. Die Vorschlagsfunktion ist zulässig falls gilt: Für einen regulären Simplex S_j mit Kantenlänge 1 ist der Simplex S_{neu} ebenfalls regulär mit Kantenlänge 1.*

Es gilt jedoch anzumerken, dass durch die Verwendung einer zulässigen Vorschlagsfunktion nicht nur reguläre Simplexe erzeugt werden. In Abbildung 3.1 ist in Schritt 4 gut zu sehen, dass der explizit neu erzeugte Simplex bestehend aus der neuen Ecke $\vec{x}_{neu}(2, 0)$ und den beiden Ecken von Simplex 2 zwar regulär ist, jedoch muss noch ein Simplex zur Triangulation hinzugefügt werden, der nicht regulär ist, um die konvexe Hülle komplett abzudecken.

Vorschlagsfunktion 1

Eine einfache zulässige Vorschlagsfunktion ist definiert als

$$\vec{x}_{neu}(i, j) := \left(\frac{2}{d} \sum_{k=0, k \neq i}^d \vec{x}_k^j \right) - \vec{x}_i^j, \quad (3.16)$$

wobei $\{\vec{x}_k^j | k = 0, \dots, d\}$ die Ecken von S_j sind. Geometrisch bedeutet dies eine Punktspiegelung von \vec{x}_i^j an dem arithmetischen Mittel der Ecken ohne \vec{x}_i^j . Dies ist jedoch keine gute Wahl. Ein so konstruierter neuer Simplex hat exakt den gleichen Inhalt und die gleichen Punkt-zu-Punkt Abstände wie der alte Simplex, was heisst, dass der neue Simplex nicht korrigiert wird, wenn der alte Simplex eine schlechte Form hat oder nicht ungefähr den gleichen Inhalt hat wie der Startsimplex.

Vorschlagsfunktion 2

Eine bessere Methode, die im folgenden immer benutzt wird, arbeitet unabhängig von der Ecke \vec{x}_i^j , die weggelassen wird. Die neue Ecke wird als ein Vektor aufgefasst mit Endpunkt \vec{v} . Der Anfangspunkt ist der Umkreismittelpunkt \vec{u} der Ecken $\{\vec{x}_k^j | k = 0, \dots, d, k \neq i\}$. Die Berechnung des Umkreismittelpunkts ist in Anhang B erklärt. Der Vektor wird so konstruiert, dass er orthogonal zu der affinen Hyperebene ist, die durch die Punkte $\{\vec{x}_k^j | k = 0, \dots, d, j \neq i\}$ definiert wird. Die Länge des Vektors wird so konstruiert, dass der Abstand zwischen \vec{v} und den Ecken $\{\vec{x}_k^j | k = 0, \dots, d, k \neq i\}$ immer genau 1 ergibt. Dies kann aufgrund der speziellen Eigenschaften des Umkreismittelpunktes \vec{u} erreicht werden. Um einen orthogonalen Vektor zu konstruieren, wird das Kreuzprodukt verwendet.

Definition 3.7 (Kreuzprodukt). *Seien $d - 1$ Punkte $\vec{x}_1, \dots, \vec{x}_{d-1}$ in \mathbb{R}^d gegeben. Das Kreuzprodukt $VP(\vec{x}_1, \dots, \vec{x}_{d-1})$ ist dann definiert als*

$$VP(\vec{x}_1, \dots, \vec{x}_{d-1}) := \sum_{k=1}^d \vec{e}_k \det(\vec{x}_1, \dots, \vec{x}_{d-1}, \vec{e}_k). \quad (3.17)$$

Dabei ist \vec{e}_k der k -te d -dimensionale Einheitsvektor. Der Vektor $\vec{v} = VP(\vec{x}_1, \dots, \vec{x}_{d-1})$ ist orthonormal zu der Hyperebene, die durch den Nullpunkt und die Punkte $\vec{x}_1, \dots, \vec{x}_{d-1}$ definiert ist. Das Kreuzprodukt ist eindeutig definiert bis auf das Vorzeichen.

Um das Kreuzprodukt anwenden zu können, muss zuerst der gerade betrachtete Simplex S_j so verschoben werden, dass ein $\vec{x}_k^j \neq \vec{x}_i^j$ in dem Koordinatenursprung liegt. Dies wird im folgenden immer \vec{x}_0^j oder, falls $i = 0$, \vec{x}_1^j sein: $\vec{z}_k^j := \vec{x}_k^j - \vec{x}_0^j, k = 1, \dots, d$. Das Kreuzprodukt wird dann auf $\vec{z}_k^j, k = 1, \dots, d, k \neq i$ angewandt:

$$\vec{w} := VP(\vec{z}_1^j, \dots, \vec{z}_{i-1}^j, \vec{z}_{i+1}^j, \dots, \vec{z}_d^j).$$

Um zu erreichen, dass der Anfangspunkt des orthogonalen Vektors in dem Umkreismittelpunkt \vec{u} liegt, müssen die Ecken des Simplex nochmals verschoben werden: $\vec{s}_k := \vec{z}_k - \vec{u}, k = 1, \dots, d$. Jetzt kann die Eigenschaft des Umkreismittelpunktes ausgenutzt und die Länge von \vec{w} so mit einer Konstante b skaliert werden, dass der euklidische Abstand zwischen \vec{v} und $\vec{s}_k^j, k = 1, \dots, d, k \neq i$ genau 1 ergibt. Schreibe dafür $\vec{v} := b\vec{w}, b \in \mathbb{R}$:

$$\|\vec{s}_k - b\vec{w}\|_2^2 = 1 \Leftrightarrow b = \pm \sqrt{\frac{1 - \|\vec{s}_k\|_2^2}{\|\vec{w}\|_2^2}}.$$

Unter Berücksichtigung der vorgenommenen linearen Transformationen ergibt sich $\vec{x}_{neu}(i, j)$ zu:

$$\vec{x}_{neu}(i, j) = \pm b\vec{w} + \vec{u} + \vec{x}_0^j.$$

Um die „korrekte“ Orientierung von $\vec{x}_{neu}(i, j)$ zu erhalten, muss überprüft werden, welche der beiden Möglichkeiten den größeren Abstand von \vec{x}_i^j hat. Abbildung 3.2 zeigt anschaulich, wie die zweite Vorschlagsfunktion funktioniert.

Wahl von $L(i, j)$

Im Zusammenhang mit der Vorschlagsfunktion 2 wird $L(i, j) \equiv 1$ gewählt, da durch die spezielle Konstruktion der Vorschlagsfunktion erreicht wird, dass der euklidische Abstand zwischen einem Punkt $\vec{x}_{neu}(i, j)$ und den Ecken $\vec{x}_k^j, k = 0, \dots, d, k \neq i$ immer

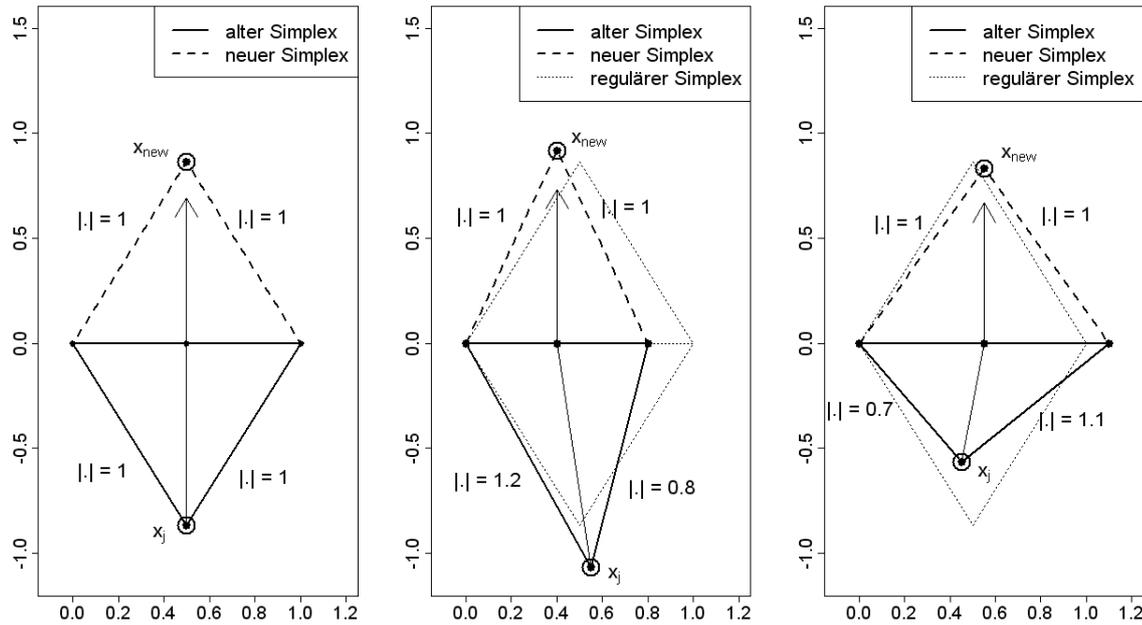


Abbildung 3.2: Darstellung der zweiten Vorschlagsfunktion für unterschiedliche Simplexe. Die neue Ecke wird so konstruiert, dass der Punkt zu Punkt Abstand von der neuen Ecke zu einer beliebigen alten Ecke immer 1 beträgt. Dadurch entstehen aus regulären Simplexen wieder reguläre Simplexe (linke Seite). Für die mittlere und rechte Graphik ist der alte Simplex nicht regulär, jedoch kann durch die Konstruktion der neuen Ecke erreicht werden, dass alle neuen Kanten die gleiche Länge haben.

1 ergibt. Wird $L(i, j) \equiv 1$ gewählt, so wird eine Punktwolke konstruiert, deren minimaler euklidischer Abstand zwischen 2 Punkten 1 ist. Generell können jedoch auch andere Optionen sinnvoll sein, die $L(i, j)$ abhängig von Simplex S_j und der Ecke \vec{x}_i^j wählen, siehe Mühlenstädt (2008). Jedoch hat die dort exklarte Vorschlagsfunktion schlechtere Ergebnisse geliefert als die zweite Vorschlagsfunktion, sodass sie hier nicht weiter erläutert wird.

Auswahl eines Versuchsplans

Die entsprechend dem NRSM Algorithmus konstruierten Punktwolken stellen noch keine brauchbaren Versuchspläne dar. Um einen Versuchsplan aus einer Punktwolke zu extrahieren, wird die Punktwolke zuerst um den Mittelwert der Punkte zentriert: $\bar{P} := \{\vec{x} - \bar{\vec{x}} | \vec{x} \in P, \bar{\vec{x}} := \frac{1}{\#P} \sum_{\vec{x} \in P} \vec{x}\}$. Um einen Versuchsplan auszuwählen, suche alle Punkte $x \in \bar{P}$ mit $\|x\|_\infty < a \in \mathbb{R}_+$. Dann wird a so variiert, dass die gewünschte Anzahl Versuchspunkte resultiert und skaliere die Punkte auf den Einheitswürfel. Eine weitere Verbesserung des Designs bzgl. des Maximinkriteriums kann sein, jede Spalte des Versuchsplan so zu skalieren, dass das spaltenweise Maximum und Minimum 1 bzw. 0 annehmen.

Es ist wichtig, dass $\#P \gg n$. Der NRSM Algorithmus erstellt eine Punktwolke, deren konvexe Hülle einer Sphäre gleicht. Um zu verhindern, dass die „Ecken“ des Versuchsraum leer sind, muss in P eine ausreichende Anzahl an Punkten vorhanden sein. Der Radius der Sphäre kann durch $r := \max_{\vec{x} \in \bar{P}}(\|\vec{x}\|)$ approximiert werden. Der Versuchsraum ist ein Kubus mit Seitenlänge $2a$, zentriert um den Nullpunkt. Der Punkt im Versuchsraum mit der größten L_2 Norm ist gegeben durch $a\sqrt{d}$. Daher sollte ein Versuchsplan vom Umfang n , der aus \bar{P} extrahiert wird, (vor dem skalieren auf den Einheitswürfel) höchstens eine Seitenlänge von r/\sqrt{d} haben.

3.2.2 Abwandlungen des NRSM Algorithmus

Die Versuchspläne, die mittels des NRSM Algorithmus erstellt wurden, weisen gute raumfüllende Eigenschaften im \mathbb{R}^d auf. Dies impliziert jedoch nicht automatisch, dass auch Projektionen der Versuchspläne in niedrigere Dimensionen gute raumfüllende Eigenschaften haben. In Abbildung 3.3 ist eine Scatterplotmatrix für zwei Versuchspläne mit $n = 50$ Läufen in $d = 5$ Dimensionen gegeben. Die Plots in der oberen rechten Hälfte wurden mit Hilfe des NRSM Algorithmuses vorgeschlagen, die Plots in der unteren

linken Halfte stellen einen zufalligen Latin Hypercube dar. Offensichtlich hat der Latin Hypercube bessere raumfullende Eigenschaften in den zweidimensionalen Projektionen als der simplexbasierte Versuchsplan, obwohl der simplexbasierte Versuchsplan einen viel besseren Wert bzgl. des Φ_{40} -Kriteriums hat als der zufallige Latin Hypercube.

Aus diesem Grund werden zwei Methoden vorgeschlagen, wie die Projektionseigenschaften der aus dem NRSM Algorithmus resultierenden Punktwolken (und damit der Versuchsplane) verbessert werden konnen. Beide Methoden konnen entweder gemeinsam oder getrennt verwendet werden. Die erste Methode besteht in einer Rotation der kompletten Punktwolke. Die euklidischen Abstande der Punkte im \mathbb{R}^d sind invariant unter Rotationen, jedoch konnen durch Rotationen die Projektionseigenschaften beeinflusst werden. Als zweite Verbesserung konnen die Versuchsplane, die aus einer NRSM Punkterwolke extrahiert wurden, in die Klasse der Latin Hypercubes transformiert werden, fur die bekannt ist, dass Versuchsplane in dieser Klasse gute Projektionseigenschaften haben.

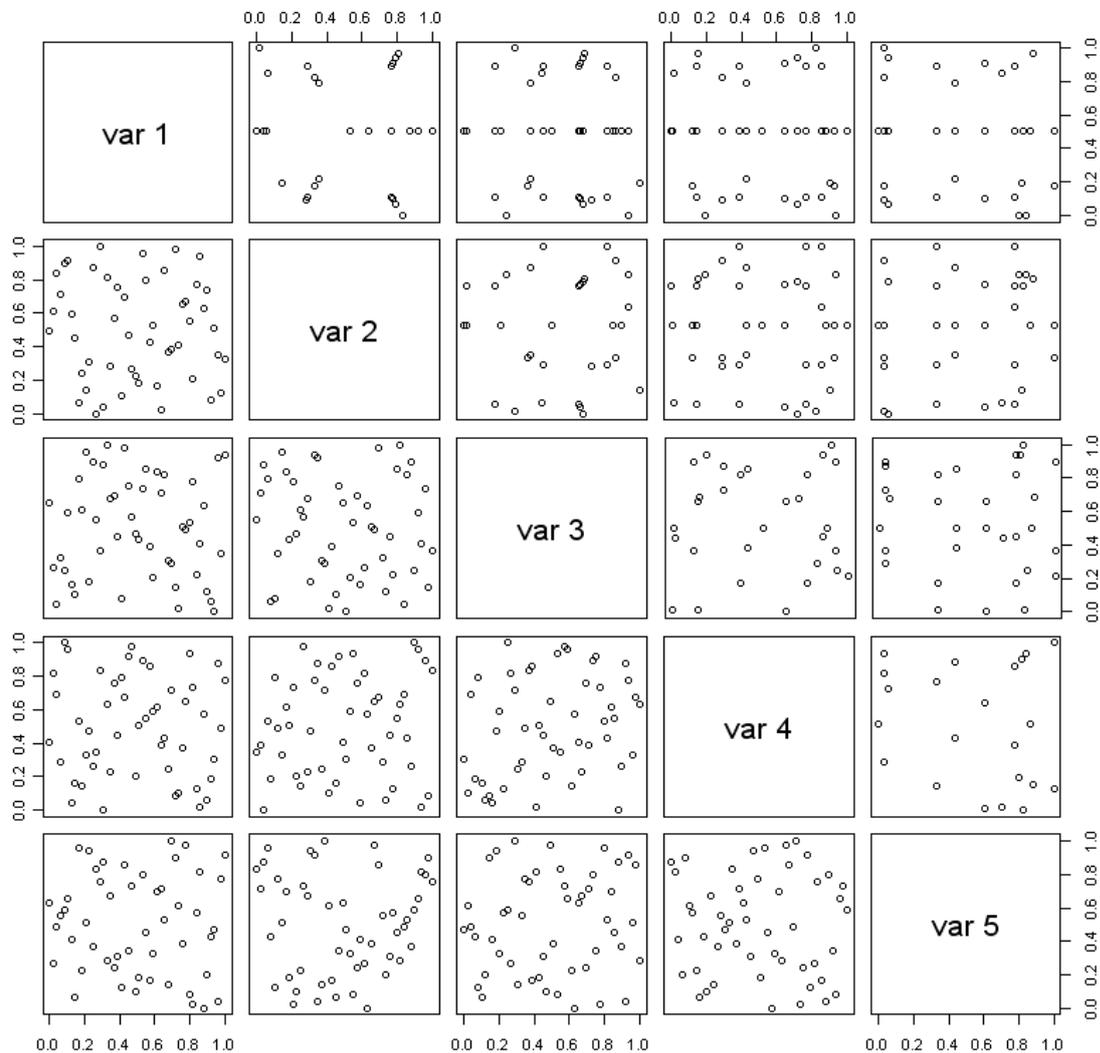


Abbildung 3.3: Scatterplotmatrix für zwei $D_{50,5}$ Versuchspläne. Die obere rechte Hälfte zeigt Scatterplots, die zu einem Versuchsplan gehören, der aus einem Simplexgitter extrahiert wurde. Die unteren linken Scatterplots stellen einen zufälligen Latin Hypercube dar.

Rotierte Punktwolken

Eine reelle quadratische Matrix R , die

$$RR' = I \text{ (Orthogonalität) und } \det(R) = 1$$

wobei $(PR(\vec{\theta}))_k$ die k -te Spalte von $PR(\vec{\theta})$ ist. Die Minimierung dieses Zielkriteriums bedeutet, eine Rotation zu suchen, die möglichst gute eindimensionale Projektionseigenschaften hat. Unter allen möglichen Parametervektoren $\{0 \leq \vec{\theta} \leq 2\pi\} := \{0 \leq \theta_1 \leq 2\pi, \dots, 0 \leq \theta_{d(d-1)/2} \leq 2\pi\}$ wird der Vektor $\vec{\theta}^*$, der $\Delta(\vec{\theta})$ minimiert, numerisch mit einer Simulated Annealing Variante gesucht.

3.2.3 Rangversion der NRSM basierten Versuchspläne

Die Klasse der Latin Hypercubes wird für Computerexperimente allgemein als attraktiv angesehen, da Versuchspläne in dieser Klasse gute Projektionseigenschaften haben. Um die guten Eigenschaften eines Latin Hypercubes zu erhalten, können die aus den Punktwolken extrahierten Versuchspläne in diese Klasse transformiert werden. Dafür wird auf jede Spalte des Versuchsplans $D_{n,d}$ die Rangtransformation angewandt und die resultierenden Spalten wieder auf das Einheitsintervall skaliert. Anhand von numerischen Vergleichen kann gezeigt werden, dass die so erhaltenen Versuchspläne gute Eigenschaften bzgl. des Maximinkriteriums haben, jedoch deutlich bessere WD Werte liefern als Versuchspläne, die nicht in die Klasse der Latin Hypercubes transformiert wurden. Sei $D_{n,d} \in \mathfrak{D}_{n,d}$ dafür wieder ein Versuchsplan, der aus einem (rotierten) Simplexgitter extrahiert wurde. Für jede Spalte $D_{n,d,k}$ von $D_{n,d}$ werden die Ränge der n Einträge gebildet. Ist also für den Eintrag (i, k) $rank(D_{n,d,i,k}) = m \in \{1 \dots, n\}$ bedeutet dies, dass es $m - 1$ Einträge j in der Spalte k gibt mit $D_{n,d,j,k} < D_{n,d,i,k}$ und $n - m$ Einträge j mit $D_{n,d,j,k} > D_{n,d,i,k}$. Durch die Rangtransformation wird die Reihenfolge der Einträge in einer Spalte nicht verändert, jedoch werden die eindimensionalen Punkt zu Punkt Abstände normalisiert. Die Scatterplotmatrix in Abbildung 3.4 stellt einen Vergleich der zweidimensionalen Projektionen von zwei Versuchsplänen dar. In der oberen rechten Hälfte ist ein simplexbasierter Versuchsplan gezeigt, der aus einem unrotierten Simplexgitter extrahiert wurde und auf den keine Rangtransformation angewandt wurde. In der unteren linken Hälfte ist ein Versuchsplan dargestellt, der aus einem rotierten

Gitter extrahiert wurde und auf den die Rangtransformation angewandt wurde. Obwohl bei beiden Versuchsplänen das Φ_{40} Kriterium vergleichbare Werte liefert, sind die zweidimensionalen Projektionen des rangtransformierten Versuchsplan, extrahiert aus einem rotierten Gitter, deutlich besser verteilt.

d	n	$\Phi_{40}(D_S)$	$\Phi_{40}(D_R)$	$\Phi_{40}(D_{random})$	$\Phi_{40}(D_\Phi)$
3	30	3.214	3.394	10.570	3.787
	60	3.976	4.582	18.140	5.269
4	40	2.429	2.723	7.368	2.768
	80	3.072	3.293	11.630	3.563
5	50	2.058	2.237	5.556	2.327
	100	2.467	2.557	7.916	2.993
6	60	1.829	1.966	4.444	2.088
	120	2.144	2.193	5.919	2.430
7	70	1.642	1.723	3.634	1.794
	140	1.883	1.919	4.668	2.185
8	80	1.484	1.598	3.113	1.619
	160	1.736	1.727	3.797	1.904

Tabelle 3.1: Vergleichsergebniss für das Φ_{40} -Kriterium. Die numerisch optimierten Versuchspläne (D_Φ) wurden entsprechend Morris und Mitchell (1995) mittels Simulated Annealing innerhalb der Klasse der Latin Hypercubes gesucht.

3.3 Vergleich und Diskussion

Die raumfüllenden Versuchspläne, die entsprechend der oben erklärten Methode konstruiert wurden, werden jetzt mit Versuchsplänen verglichen, die numerisch entsprechend den Kriterien $Mm(D)$, $\Phi_{40}(D)$ und $WD(D)$ optimiert wurden. Die Kri-

terien Φ_p und WD müssen minimiert werden um optimale Versuchspläne zu erhalten. Die Optimierung wurde mittels eines Simulated Annealing Algorithmus innerhalb der Klasse der Latin Hypercubes durchgeführt. Die Rechenzeiten für die numerische Optimierung lagen dabei zwischen 14.9 und 279.7 Sekunden (Φ -Kriterium) und 17.8 und 766.8 Sekunden für das WD -Kriterium auf einem 2 GHz Prozessor unter Windows XP in R.2.8.1. Die Extraktion eines Versuchsplans aus einem Simplexgitter hat in allen Fällen < 1 Sekunden benötigt, unabhängig von der Dimension und der Anzahl an Versuchspunkten. Das Mm Kriterium muss maximiert werden. Hier wurden, wo verfügbar, Versuchspläne verwandt, deren Konstruktion in Grosso et al. (2009) beschrieben ist und die auf der Website www.spacefillingdesigns.nl (Datum 30.05.2009) aufgelistet sind. Für diese Versuchspläne sind keine Rechenzeiten vorhanden. Seien D_R, D_S Versuchspläne, die aus rotierten Simplexgittern extrahiert wurden mit bzw. ohne Rangtransformation. Soll ein raumfüllender Versuchsplan konstruiert werden, ist eine Minimalanforderung, dass die Güte des Versuchsplans besser ist als zufällige Latin Hypercubes. Deswegen wurde in die Vergleiche der Mittelwert der Gütekriterien für 1000 zufällige Latin Hypercubes mit aufgenommen, bezeichnet mit D_{random} . Die Ergebnisse sind in Tabelle 3.1 (Φ_p), Tabelle 3.2 (Mm) und Tabelle 3.3 (WD) aufgelistet. Für alle drei Tabellen enthalten die ersten zwei Spalten die Werte der Kriterien für die Versuchspläne D_S und D_R . In der dritten Spalten ist der Mittelwert der Kriterien über 1000 zufällige Latin Hypercubes aufgeführt und in der letzten Spalte die Ergebnisse der numerisch optimierten Versuchspläne.

Die Ergebnisse aus Tabelle 3.2 und Tabelle 3.1 zeigen, dass die simplexbasierten Versuchspläne D_S und D_R , extrahiert von rotierten Simplexgittern, äußerst konkurrenzfähig sind bzgl. der beiden Maximinkriterien im Vergleich zu numerisch optimierten Versuchsplänen. Für das Φ_{40} -Kriterium in Tabelle 3.1 sind alle Versuchspläne, die aus rotierten Simplexgittern extrahiert wurden, besser als Versuchspläne, die mittels Simulated Annealing bzgl. Φ_{40} optimiert wurden. Dabei sind die Versuchspläne aus einem

d	n	$Mm(D_S)$	$Mm(D_R)$	$Mm(D_{random})$	$Mm(D_{Mm})$
3	30	0.3436	0.2966	0.1028	0.3600
	60	0.2830	0.2249	0.06046	0.2800
4	40	0.4534	0.3795	0.1451	0.4763
	80	0.3715	0.3157	0.0937	0.3899
5	50	0.5435	0.4600	0.1908	0.5894
	100	0.4659	0.4194	0.1339	0.4949
6	60	0.6147	0.5127	0.2344	0.6879
	120	0.5402	0.4855	0.1776	-
7	70	0.7004	0.6021	0.2855	0.7798
	140	0.6244	0.5504	0.2235	-
8	80	0.7736	0.6474	0.3305	0.8673
	160	0.6781	0.6201	0.2712	-

Tabelle 3.2: Vergleichsergebnisse für das Mm -Kriterium. Wo verfügbar, wurden numerisch optimierte Versuchspläne von der Webseite www.spacefillingdesigns.nl genommen.

Simplexgitter ohne Rangtransformation besser als Versuchspläne, die mittels der Rangtransformation in die Klasse der Latin Hypercubes transformiert wurden.

Die numerisch optimierten Versuchspläne in Tabelle 3.2 wurden von Grosso et al. (2009) mittels einer „iterated local search heuristic“ innerhalb der Klasse der Latin Hypercubes konstruiert. Dabei wird die lokale Suche auf den Latin Hypercube aus dem letzten Optimierungsschritt angewandt, der leicht verfälscht wurde. Die aus einem Simplexgitter extrahierten Versuchspläne ohne Rangtransformation werden vergleichbar bewertet wie die numerisch bzgl. dem Mm Kriterium optimierten Versuchspläne. Die simplexbasierten Versuchspläne, die mittels der Rangtransformation in die Klasse der Latin Hypercubes transformiert wurden, weisen einen schlechteren Wert auf als die numerisch optimierten Versuchspläne, jedoch immer noch einen deutlich besseren Mm -Wert als

das Mittel über 1000 zufällige Latin Hypercube Versuchspläne.

k	n	$WD(D_S)$	$WD(D_R)$	$WD(D_{random})$	$WD(D_{WD})$
3	30	0.01706	0.00458	0.00606	0.00418
	60	0.00709	0.00156	0.00244	0.00144
4	40	0.02055	0.00830	0.01024	0.00725
	80	0.00949	0.00293	0.00457	0.00262
5	50	0.03752	0.01309	0.01732	0.01274
	100	0.01483	0.00553	0.00808	0.00485
6	60	0.05431	0.02191	0.02865	0.02063
	120	0.02527	0.01010	0.01371	0.00851
7	70	0.12332	0.04109	0.04656	0.03178
	140	0.05574	0.01889	0.02254	0.01447
8	80	0.17347	0.06892	0.07446	0.05354
	160	0.09463	0.03039	0.03628	0.02618

Tabelle 3.3: Vergleichsergebnisse für das WD -Kriterium. Die numerisch optimierten Versuchspläne wurden mit Hilfe des gleichen Simulated Annealing Algorithmuses erzielt wie für die Φ_{40} Versuchspläne.

Die Ergebnisse für die Uniform Versuchspläne bzgl. des WD Kriterium in Tabelle 3.3 zeigen ein anderes Ergebnis. Hier sind die Versuchspläne aus einem Simplexgitter ohne Rangtransformation schlechter als vergleichbare numerisch optimierte Versuchspläne und auch schlechter als das Mittel über 1000 zufällige Latin Hypercubes. Die simplexbasierten Versuchspläne ohne Rangtransformation stellen also keine attraktiven Uniformversuchspläne dar und sind somit nicht für die numerische Integration geeignet. Eine mögliche Erklärung dafür ist, dass beim Start des NRSM Algorithmuses einige reguläre Simplexe konstruiert werden, nachher jedoch nicht mehr reguläre Simplexe hinzugefügt werden, wodurch sich strukturelle Unterschiede für die resultierende Punktwolke ergeben. Wird jedoch die Rangtransformation auf den aus einem rotier-

ten Simplexgitter extrahierten Versuchsplan angewandt, verbessert sich der WD -Wert des Versuchsplans erheblich, sodass sich Zielwerte ergeben, die deutlich besser sind als das Mittel der 1000 zufälligen Latin Hypercubes jedoch immer noch etwas schlechter als die numerisch optimierten Versuchspläne.

Numerische Aspekte

Der rechentechnisch intensivste Teil des NRSM Algorithmuses ist die fortlaufende Berechnung der Delaunaytriangulation der Punktwolken. Für eine feste Anzahl n von Ecken gilt, dass die Anzahl Simplexe in der Delaunaytriangulation sehr schnell wächst (Klee (1980)). Für Dimensionen größer als 7 macht dies die Berechnung einer ausreichenden Anzahl an Punkten auf einem Desktop PC praktisch unmöglich, da nicht genügend Arbeitsspeicher vorhanden ist. Auf einem Server mit 8 GB Arbeitsspeicher unter Linux war es möglich, in 8 Dimensionen 601 Punkte zu berechnen. Zur Berechnung der Delaunaytriangulation wurde der quickhull Algorithmus verwendet (Barber et al. (1996)). Der NRSM Algorithmus wurde in 'R 2.8.1' implementiert unter Ausnutzung des Befehls `delaunayn` des packages `geometry`.

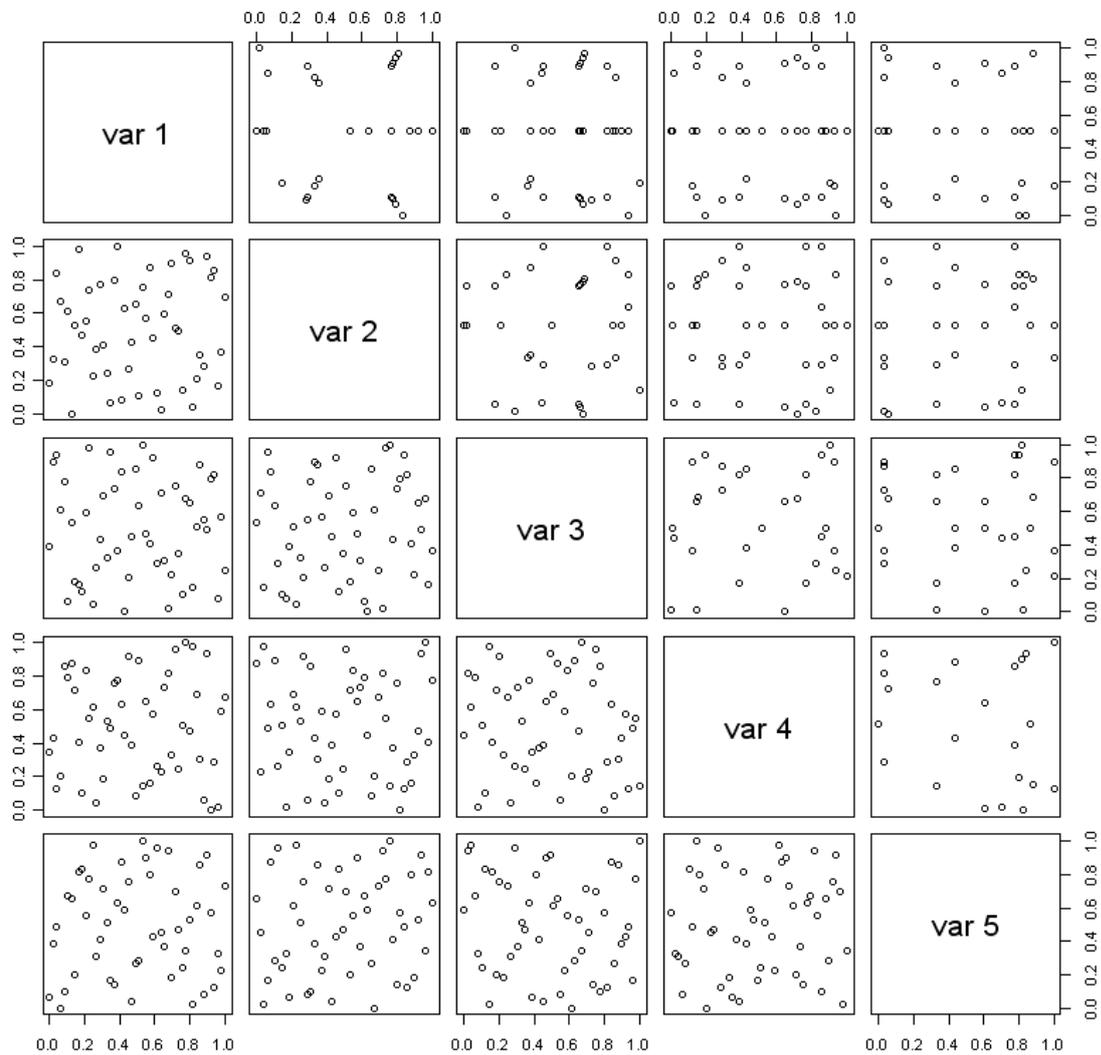


Abbildung 3.4: Scatterplotmatrix für zwei $D_{50,5}$ Versuchspläne. Die obere rechte Hälfte zeigt den gleichen, aus einem nichtrotierten Simplexgitter extrahierten Versuchsplan wie in Abbildung 3.3. In der linken unteren Hälfte ist diesmal ein aus einem rotierten Simplexgitter extrahierter Versuchsplan dargestellt, auf den die Rangtransformation angewandt wurde.

MODELLIERUNG VON COMPUTEREXPERIMENTEN

Für die Metamodellierung von Computerexperimenten gibt es viele verschiedene Methoden. Da die zu modellierenden Computerexperimente meistens deterministisch sind, sollte das Metamodell die Beobachtungen interpolieren. Formal lässt sich die Aufgabenstellung der Interpolation wie folgt darstellen: Es liegen die Beobachtungen $(y_1, \vec{x}_1), \dots, (y_n, \vec{x}_n)$ vor mit $\vec{x}_i \in \mathfrak{D} \subset \mathbb{R}^d$ und $y_i = f(\vec{x}_i)$. Dabei ist $f \in C^0(\mathfrak{D})$ eine unbekannte Funktion, für die ein passender Ersatz $\hat{f}(\vec{x})$ gefunden werden soll. Im Zusammenhang der Computerexperimente wird die Funktion $\hat{f}(\vec{x})$ oft als Metamodell bezeichnet, da die zugrundeliegenden Beobachtungen meistens ebenfalls aus einem (nichtzufälligen) Modell berechnet wurden. Dies sind oftmals numerische Lösungen von partiellen Differentialgleichungen, die sehr rechenintensiv sind, so dass nur eine begrenzte Anzahl an Versuchsläufen vorhanden ist. Ein gutes Metamodell sollte also ein möglichst guter Ersatz für das Computerexperiment sein (z.B. im Sinne der Vorhersagegüte) und leicht zu berechnen sein, da ein rechenintensiver Ersatz keinen zusätzlichen Nutzen zu den Computerexperimenten liefert. Im folgenden werden Metamodelle vorgestellt, die in drei Klassen eingeteilt werden können: Krigingmodelle, Interpolation mit Radialen Basisfunktionen und Inverse Distance Weighting. Im Anschluss daran wird in Kapitel 5 ein neues Metamodell vorgestellt, das auf den in Kapitel 2 beschriebenen stückweise linearen stetigen Funktionen aufbaut und diese entscheidend verbessert. Die vorgestellten Metamodelle werden anhand numerischer Vergleiche beurteilt.

4.1 Kriging

Kriging ist die Standardmethode zur Modellierung von Computerexperimenten (Santner et al. (2003), Sacks et al. (1989)). Die Methode wurde ursprünglich in der Geostatistik entwickelt (Cressie (1993)) und kann als eine Adaption der gewöhnlichen Kleinsten Quadrate Regression angesehen werden für den Fall, dass die Fehlerterme in einem linearen Regressionsmodell nicht paarweise unabhängig sind. Je nachdem wie die Abhängigkeitsstruktur der Fehler in dem Regressionsmodell angenommen wird, kann mit der Krigingmethodik interpoliert werden. Auch wenn die Beobachtungen eines Computerexperiments deterministisch sind, wird der Output Y des Computerexperiments als Zufallsgröße modelliert:

$$Y(\vec{x}_i) = \vec{g}(\vec{x}_i)^T \vec{\beta} + Z(\vec{x}_i). \quad (4.1)$$

Dabei ist $\vec{g}(\vec{x})$ eine bekannte (vektorwertige) Regressionsfunktion und $Z(\vec{x}_i)$ eine normalverteilte Zufallsgröße. Übliche Annahmen an die Regressionsfunktion sind $g(\vec{x}) = 1$ oder $\vec{g}(\vec{x}) = (1, \vec{x})'$, also ein konstantes Mittel oder eine lineare Regressionsfunktion. Die Vektoren $\vec{g}(\vec{x}_1), \dots, \vec{g}(\vec{x}_n)$ werden in der Matrixschreibweise zur Matrix G zusammengefasst. Würde für $Z(\vec{x}_1), \dots, Z(\vec{x}_n)$ angenommen, dass sie unabhängig identisch normalverteilt sind, ergäbe sich die gewöhnliche Kleinsten Quadrate Regression. Um jedoch der Anforderung der Interpolation zu genügen, wird beim Kriging angenommen, dass Y einem Gaußprozess folgt:

Definition 4.1 (Gaußprozess, Fang et al. (2006), S. 249). *Ein stochastischer Prozess $\{Y(x), x \in \mathbb{R}^d\}$ heisst Gaußprozess, wenn jede endlich dimensionale Randverteilung einer Normalverteilung folgt, d.h. für alle endlichen Mengen $\{\vec{x}_1, \dots, \vec{x}_n\}$ mit $\vec{x}_i \in \mathbb{R}^d$ und beliebigem $n \in \mathbb{N}$ folgt die gemeinsame Verteilung des Zufallsvektors $\vec{Y} = [Y(\vec{x}_1), \dots, Y(\vec{x}_n)]'$ einer Normalverteilung:*

$$\vec{Y} \sim \mathbb{N}(\vec{\mu}(\vec{x}_1, \dots, \vec{x}_n), \Sigma(\vec{x}_1, \dots, \vec{x}_n)). \quad (4.2)$$

Offensichtlich gilt, dass die Zufallsvariable $Y(\vec{x})$ aus Gleichung (4.1) auch ein Gaußprozess ist, wenn $Z(\vec{x})$ ein Gaußprozess ist. Eine weitere Annahme, die fast immer getroffen wird, ist die Stationarität.

Definition 4.2 (Stationarität, Cressie (1993), S. 53). *Der stochastische Prozess $Z(\vec{x})$ heißt stationär, falls $E(Z(\vec{x})) = \mu$, $\text{var}(Z(\vec{x})) = \sigma^2$ und $\text{cov}(Z(\vec{x}_1), Z(\vec{x}_2)) = \gamma(\vec{x}_1 - \vec{x}_2)$, d.h. der Erwartungswert und die Varianz des Prozesses hängen nicht von \vec{x} ab und die Kovarianz zwischen $Z(\vec{x}_1)$ und $Z(\vec{x}_2)$ hängt nur von der Differenz $\vec{h} := \vec{x}_1 - \vec{x}_2$ ab.*

Die hier getroffene Definition ist genau genommen schwache Stationarität (siehe z.B. Cressie (1993), S. 53) oder second order Stationarität. Ein anderer oft verwandter Stationaritätsbegriff ist die starke Stationarität (Cressie (1993), S. 53). Bei der starken Stationarität wird gefordert, dass für die gemeinsame Verteilungsfunktion $F_{Y(\vec{x}_1), \dots, Y(\vec{x}_n)}(y_1, \dots, y_n)$ gilt: $F_{Y(\vec{x}_1), \dots, Y(\vec{x}_n)}(y_1, \dots, y_n) \equiv F_{Y(\vec{x}_1 + \vec{t}), \dots, Y(\vec{x}_n + \vec{t})}(y_1, \dots, y_n)$ für beliebige $\vec{t} \in \mathbb{R}^d$. Da jedoch ein Gaußprozess (und nur diese werden hier betrachtet) vollständig über die ersten beiden Momente definiert ist, stimmen die beiden Arten der Stationarität hier überein.

Für den Gaußprozess $Z(\vec{x})$ aus Gleichung (4.1) wird im Krigingmodell Stationarität mit Mittelwert 0 angenommen. Mögliche Korrelationsfunktionen $R_\theta(h)$ für einen Gaußprozess $Z(x)$ mit eindimensionalem Inputwert x sind in Tabelle 4.1 angegeben (aufgrund der Stationarität in Abhängigkeit von der Differenz $h = x_1 - x_2$).

Power Exponential	$R_\theta(h) = \exp(- h/\varphi ^p), \quad \varphi > 0, 0 < p \leq 2, \vec{\theta} = (\varphi, p)$
Gauß	$R_\theta(h) = \exp(- h/\theta ^2), \quad \theta > 0$
Exponential	$R_\theta(h) = \exp(- h/\theta), \quad \theta > 0$
Matern	$R_\theta(h) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{2\sqrt{\nu} h }{\varphi}\right)^\nu K_\nu\left(\frac{2\sqrt{\nu} h }{\varphi}\right), \nu > 0, \varphi > 0, \vec{\theta} = (\varphi, \nu)$

Tabelle 4.1: Verschiedene Korrelationsfunktion. Für die Matern Korrelationsfunktion bezeichnet $K_\nu(t)$ eine modifizierte Besselfunktion vom Grad ν (Abramowitz und Stegun (1972), Kapitel 9 und 10) und $\Gamma(t)$ die Gammafunktion.

Soll aus diesen eindimensionalen Korrelationsfunktionen eine multivariate Korrelationsfunktion konstruiert werden, so werden in der Praxis meistens zwei Verallgemeinerungen genutzt: Die erste ist, dass als Maß für den Abstand im mehrdimensionalen die euklidische Norm verwandt wird: $h := \|\vec{x}_1 - \vec{x}_2\|$. Dies impliziert, dass der resultierende Gaußprozess $Z(\vec{x})$ isotrop ist (Cressie (1993), S. 53). Das Konzept der Isotropie bedeutet, dass die Verteilung des stationären Gaußprozesses invariant ist unter Rotationen der \vec{x} -Variablen. Häufiger angewandt werden jedoch anisotropische Korrelationsfunktionen, bei denen die mehrdimensionale Korrelationsfunktion durch Produktbildung aus eindimensionalen Korrelationsfunktionen gebildet wird:

$$R(\vec{h}) := \prod_{k=1}^d R_{\theta_k}(h_k) \quad (4.3)$$

mit $\vec{h} = \vec{x}_1 - \vec{x}_2 = (h_1, \dots, h_d)^T$.

Die in Tabelle 4.1 aufgeführten Korrelationsfunktionen stellen oft verwandte eindimensionale Korrelationsfunktionen dar, die auf eine der beiden gerade erläuterten Weisen zu mehrdimensionalen Korrelationsfunktionen erweitert werden können. Dabei sind die Gauß- und Exponentialkorrelationsfunktion Spezialfälle der Powerexponentialkorrelationsfunktion. Für die Maternkorrelationsfunktion wird der Parameter ν oftmals auf einen Wert festgesetzt. Beliebte Werte sind $\nu = 0.5$ oder $\nu = 1.5$. Für $\nu = 0.5$ ist die Maternkorrelationsfunktion formal identisch zu der Exponentialkorrelationsfunktion (Diggle (2009)). Desweiteren gilt für die Maternkorrelationsfunktion, dass sie für $\nu \rightarrow \infty$ gegen die Gaußkorrelationsfunktion konvergiert. Für die Powerexponentialkorrelationsfunktion als auch für die Maternkorrelationsfunktion gibt der Parameter $\vec{\theta}$ an, wie weit die Korrelation in dem Inputparameter reichen soll und \vec{p} bzw. $\vec{\nu}$ bestimmen Glattheitseigenschaften des resultierenden Gaußprozesses. Um die Glattheitseigenschaften eines Gaußprozesses genauer zu beschreiben, wird das Konzept der Differenzierbarkeit im quadratischen Mittel (bzw. Stetigkeit im quadratischen Mittel) verwandt (siehe z.B. Santner et al. (2003), S. 32 ff., Abrahamsen (1997)). Die Glattheit eines Prozesses hängt hauptsächlich von dem Verhalten der Korrelationsfunktion $R_{\theta}(\vec{h})$ für $\vec{h} \rightarrow 0$ ab. Je glatter die Korrelationsfunktion im Nullpunkt ist, desto glatter wird auch der resultierende

Gaußprozess sein.

4.1.1 Schätzung der Parameter

Wird ein Gaußprozess entsprechend Formel (4.1) mit einer Regressionsfunktion $\vec{g}(\vec{x})^T \vec{\beta}$ und der Korrelationsfunktion $R_{\vec{\theta}}(\vec{h})$ angenommen, müssen die unbekannt Parameter $(\vec{\beta}, \vec{\theta}, \sigma^2)$ aus einer Stichprobe geschätzt werden. Im gewöhnlichen Regressionsproblem, bei dem die Zufallsvariablen $Z(\vec{x}_1), \dots, Z(\vec{x}_n)$ als unabhängig identisch verteilt angenommen werden, ist eine Lösung durch den Kleinsten Quadrate Schätzer gegeben ($\hat{\vec{\beta}} = (G^T G)^{-1} G^T \vec{y}$). Sind die Parameter der Korrelationsfunktion, $\vec{\theta}$, bekannt, ergeben sich als Schätzer für die unbekannt Parameter $\vec{\beta}$ und σ^2 die verallgemeinerten Kleinsten Quadrate Schätzer:

$$\hat{\vec{\beta}} = (G^T V(\vec{\theta}) G)^{-1} G^T V(\vec{\theta}) \vec{y}. \quad (4.4)$$

und

$$\hat{\sigma}^2 = \frac{1}{n - L - 1} (\vec{y} - G \hat{\vec{\beta}})' V^{-1}(\vec{\theta}) (\vec{y} - G \hat{\vec{\beta}}) \quad (4.5)$$

Dabei ist $V(\theta)$ die Korrelationsmatrix mit Einträgen $v_{i,j}(\vec{\theta}) = R_{\vec{\theta}}(\vec{x}_i - \vec{x}_j)$. L ist die Länge des Vektors $\vec{\beta}$. Wird in Gleichung (4.5) der Vorfaktor $\frac{1}{n-L-1}$ durch $1/n$ ersetzt, ergibt sich der (verzerrte) ML Schätzer für σ^2 .

Im Allgemeinen ist es jedoch unrealistisch, dass die Parameter der Korrelationsfunktion bekannt sind, folglich müssen sie ebenfalls aus der Stichprobe geschätzt werden. Üblicherweise werden dafür Maximum Likelihood Ansätze verwandt. Die Loglikelihood aufgrund der Normalverteilungsannahme (ohne Konstanten) ist gegeben durch:

$$l_{ML}(\vec{\beta}, \sigma^2, \vec{\theta}) = -\frac{n}{2} \log(\sigma^2) - \frac{1}{2} \log(|V(\vec{\theta})|) - \frac{1}{2\sigma^2} (\vec{y} - G \vec{\beta})^T V^{-1}(\vec{\theta}) (\vec{y} - G \vec{\beta}). \quad (4.6)$$

Problematisch bei dem ML Ansatz ist, dass die Likelihood üblicherweise nicht mehr direkt in allen Parametern gleichzeitig analytisch maximiert werden kann. Als Aus-

weg können z.B. iterative Schätzverfahren verwandt werden. Das Vorgehen für die Schätzung bei unbekanntem Korrelationsparameter sieht dann wie folgt aus (Fang et al. (2006), S. 149): Es wird eine erste Schätzung (meistens die gewöhnliche kleinste Quadrate Schätzung) für $\hat{\beta}$ und $\hat{\sigma}^2$ berechnet. Diese Schätzwerte werden dann in die Loglikelihoodfunktion eingesetzt und diese bezüglich der Korrelationsparameter $\vec{\theta}$ optimiert. Da dies nicht mehr analytisch möglich ist, kommen hier Optimierungsalgorithmen zum Einsatz wie z.B. Nelder-Mead Optimierung oder Gradientenverfahren. Aus den resultierenden Korrelationsparametern $\hat{\vec{\theta}}$ wird die Korrelationsmatrix $V(\hat{\vec{\theta}})$ neu berechnet und die verallgemeinerte kleinste Quadrate Methode wird verwandt, um eine Aktualisierung der Parameterschätzung für $\hat{\beta}$ und $\hat{\sigma}^2$ vorzunehmen. Dieser Vorgang wird iterativ so lange wiederholt, bis keine weitere (nennenswerte) Verbesserung der Loglikelihood erzielt wird.

4.1.2 Vorhersage beim Kriging

Sind die Parameter des Gaußprozesses geschätzt, kann eine Vorhersage für einen beliebigen Punkt $\vec{x} \in \mathfrak{D}$ gemacht werden. Um die Formel der Vorhersagefunktion herzuleiten, wird für den Vorhersagepunkt \vec{x} und die Beobachtungspunkte $X = (\vec{x}_1^T, \dots, \vec{x}_n^T)$ des Versuchsplans eine gemeinsame Normalverteilung angenommen:

$$\begin{pmatrix} Y(\vec{x}) \\ \vec{Y}(X) \end{pmatrix} \sim \mathbb{N} \left(\begin{pmatrix} g(\vec{x})^T \vec{\beta} \\ G\vec{\beta} \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & r_{\vec{\theta}}(\vec{x})^T \\ r_{\vec{\theta}}(\vec{x}) & V_{\vec{\theta}}(X) \end{pmatrix} \right), \quad (4.7)$$

wobei $r_{\vec{\theta}}(\vec{x}) = (R_{\vec{\theta}}(\vec{x} - \vec{x}_1), \dots, R_{\vec{\theta}}(\vec{x} - \vec{x}_n))^T$. Als Vorhersage wird dann der bedingte Erwartungswert von $Y(\vec{x})$ gegeben $Y(\vec{x}_1), \dots, Y(\vec{x}_n)$ genutzt. Dabei kann ausgenutzt werden, dass für die multivariate Normalverteilung die Verteilung von $Y(\vec{x})$ gegeben $Y(\vec{x}_1), \dots, Y(\vec{x}_n)$ wieder eine Normalverteilung ist mit folgenden Parametern (Fang et al. (2006), S. 248):

$$E(Y(\vec{x})|Y(\vec{x}_1), \dots, Y(\vec{x}_n)) = g(\vec{x})^T \vec{\beta} + r_{\vec{\theta}}(\vec{x})^T V_{\vec{\theta}}^{-1}(X)(\vec{y} - G\vec{\beta}), \quad (4.8)$$

$$\text{Var}(Y(\vec{x})|Y(\vec{x}_1), \dots, Y(\vec{x}_n)) = \sigma^2(1 - r_{\vec{\theta}}(\vec{x})^T V_{\vec{\theta}}^{-1}(X)r_{\vec{\theta}}(\vec{x})). \quad (4.9)$$

Als Vorhersagefunktion wird dann Formel (4.8) mit eingesetzten Schätzwerten verwendet:

$$\hat{y}(\vec{x}) = g(\vec{x})^T \hat{\vec{\beta}} + r_{\hat{\vec{\theta}}}(\vec{x})^T V_{\hat{\vec{\theta}}}^{-1}(X)(\vec{y} - G\hat{\vec{\beta}}). \quad (4.10)$$

4.1.3 Unsicherheitsvorhersage der Schätzung

Ein in vielen Situationen nützlicher Aspekt des Krigingmodells ist, dass die Genauigkeit der Vorhersage am Punkt \vec{x} mittels einer Unsicherheitsvorhersage beurteilt werden kann. Dafür kann entweder die Varianz der bedingten Verteilung aus Gleichung (4.9) (unter Einsetzen der geschätzten Parameterwerte) genutzt werden, oder es wird der Mean Square Error der Vorhersagefunktion ($MSE[\hat{y}(\vec{x})]$) herangezogen (vgl. Sacks et al. (1989), Gleichung (8)):

$$MSE[\hat{y}(\vec{x})] = \sigma^2 \left\{ 1 - (g^T(\vec{x}), r_{\vec{\theta}}^T(\vec{x})) \begin{pmatrix} 0 & G^T \\ G & V_{\vec{\theta}}(X) \end{pmatrix}^{-1} \begin{pmatrix} g(\vec{x}) \\ r_{\vec{\theta}}(\vec{x}) \end{pmatrix} \right\} \quad (4.11)$$

Ist der Vorhersagepunkt \vec{x} sehr weit von den Beobachtungspunkten entfernt, beträgt die durch das Modell vorhergesagte Unsicherheit σ^2 , also die Varianz des angenommenen Gaußprozesses. Liegt der Vorhersagepunkt sehr nahe an einem Beobachtungspunkt, so ist die Unsicherheit sehr klein und wird, falls $\vec{x} \rightarrow \vec{x}_i$, gegen 0 konvergieren. Um eine Unsicherheitsvorhersage zu erhalten, werden in Formel (4.11) die ML-Schätzer $\hat{\vec{\theta}}$ und $\hat{\sigma}^2$ eingesetzt.

4.1.4 *Flexibilität des Krigingansatzes*

Ein Grund für die Beliebtheit von Kriging ist die hohe Flexibilität des Ansatzes. So können viele Datensituation mittels Kriging bearbeitet werden, wie z.B. gleichzeitiges Modellieren von mehrdimensionalem Output (Santner et al. (2003), Kapitel 4.2), Vorhersage auch bei kategorialen Einflussvariablen (Han et al. (2009)) und die Einbindung von verschiedenen Computerexperimenten mit unterschiedlicher Genauigkeit (Qian et al. (2006), Qian und Wu (2008)). Es stehen unterschiedliche Schätzmethoden zur Verfügung wie z.B. ML-Ansätze, REML Ansätze, Kreuzvalidierung (Santner et al. (2003)) oder z.B. Ansätze über das Variogramm (Cressie (1993)). Auch können bei einer Krigingmodellierung oftmals Aussagen über die Größe des Einflusses verschiedener Inputparameter gemacht werden (über den Korrelationsparametervektor $\vec{\theta}$) und es ist auch relativ unproblematisch, wenn Einflussgrößen in dem Modell sind, die keinen wirklichen Einfluss haben. Dies ist bei den anderen Alternativen, die im folgenden vorgestellt werden, nicht der Fall. Ein weiterer Grund ist die oftmals erstaunlich hohe Vorhersagefähigkeit von Kriging, die auch hier in einer Simulationsstudie zum Vorschein kommt.

Jedoch ist die Handhabung von Krigingmodellen oftmals anspruchsvoller als z.B. von Interpolatoren mit radialen Basisfunktionen. So kann die Optimierung der Likelihood schwierig sein und zu unerwünschten Ergebnissen führen. Dies kann insbesondere dann der Fall sein, wenn die Korrelationsmatrix schlecht konditioniert ist. Weiterhin muss unter vielen verschiedenen Möglichkeiten (welche Korrelationsfunktion, welche Regressionsfunktion, Schätzverfahren) eine Krigingvariante ausgewählt werden. Die Annahme der Stationarität sollte, wenn möglich, überprüft werden und es sollte, falls Stationarität nicht als gegeben angesehen werden kann, eine passende Transformation der Zielgröße Y vorgenommen werden. Nichtsdestotrotz ist Kriging (zu Recht) die Standardmethode zur Modellierung von Computerexperimenten.

4.2 Radiale Basis Funktion Interpolation

Eine weitere oft verwandte Interpolationsmethode ist die Interpolation mit radialen Basisfunktionen. Dabei ist die allgemeine Form der Interpolationsfunktion gegeben durch

$$\hat{f}(\vec{x}) := \sum_{i=1}^n \lambda_i \phi(\|\vec{x} - \vec{x}_i\|) + p_m(\vec{x}). \quad (4.12)$$

Hier ist $\phi(d)$ eine radiale Basisfunktion und p_m ein Polynom vom Grad $m \in \mathbb{N}$. Die Parameter $\lambda_1, \dots, \lambda_n$ und die Koeffizienten für das Polynom p_m sind dabei so zu bestimmen, dass die resultierende Funktion die Beobachtungen interpoliert. Im folgenden wird eine Unterscheidung zwischen der Gauß'schen, invers Multiquadratischen, linearen, multiquadratischen und kubischen radialen Basisfunktionen auf der einen Seite und dem Thin Plate Spline auf der anderen Seite gemacht.

4.2.1 Allgemeine radiale Basisfunktion

Ist eine radiale Basisfunktion $\phi(x)$ und das Polynom p_m ausgewählt worden, müssen die Parameter $\vec{\lambda} = [\lambda_1, \dots, \lambda_n]$ und die Koeffizienten des Polynoms p_m , $\vec{\beta}$ so bestimmt werden, dass die Funktion $\hat{f}(\vec{x})$ aus (4.12) die Beobachtungen interpoliert. Dafür spielt die sogenannte Interpolationsmatrix $A \sim (n \times n)$ eine entscheidende Rolle. Die einzelnen Elemente (i, j) sind durch $A_{i,j} := \phi(\|\vec{x}_i - \vec{x}_j\|)$, $i = 1, \dots, n$, $j = 1, \dots, n$ gegeben. Strukturell hat die Interpolationsmatrix eine Ähnlichkeit zu der Korrelationsmatrix beim Kriging. Damit die Funktion $\hat{f}(\vec{x})$ eindeutig definiert ist, muss im Allgemeinen die radiale Basisfunktion die Eigenschaft haben, dass die Interpolationsmatrix A für jede beliebige Punktemenge $\vec{x}_1, \dots, \vec{x}_n \in \mathbb{R}^d$ nichtsingulär ist. Wird in Gleichung (4.12) kein Polynom $p_m(\vec{x})$ mit in die Funktion aufgenommen, ergeben sich die Koeffizienten $\vec{\lambda} := [\lambda_1, \dots, \lambda_n]'$ durch das folgende Gleichungssystem:

$$A\vec{\lambda} = \vec{y}. \quad (4.13)$$

Wird ein Polynom mit in die Vorhersagefunktion eingearbeitet, ergeben sich zusätzliche Freiheitsgrade, die durch Nebenbedingungen fixiert werden. Dabei wird eine Orthogonalität zwischen dem Vektor $\vec{\lambda}$ und dem Raum aller Polynome vom Grad m in d Variablen, \mathbb{P}_m^d gefordert:

$$\sum_{i=1}^n \lambda_i q(\vec{x}_i) = 0 \quad \forall q \in \mathbb{P}_m^d, \quad (4.14)$$

vergleiche Buhmann (2000), Gleichung (1.4). Durch diese Forderung werden zwar die zusätzlichen Freiheitsgrade fixiert, jedoch muss jetzt an die Beobachtungspunkte $\vec{x}_1, \dots, \vec{x}_n$ eine Bedingung (\mathbb{P}_m^d -Unisolvenz) gestellt werden (Buhmann (2008), S. 107):

$$\forall q \in \mathbb{P}_m^d : \quad q(\vec{x}_i) = 0, i = 1, \dots, n \Rightarrow q \equiv 0. \quad (4.15)$$

Anschaulich bedeutet dies bei $m = 1$, dass die Punkte $\vec{x}_1, \dots, \vec{x}_n$ nicht auf einer Hyperebene des \mathbb{R}^d liegen, also den gesamten \mathbb{R}^d aufspannen. Die zusätzlichen Koeffizienten des Polynoms, hier mit $\lambda_{n+1}, \dots, \lambda_{n+\hat{m}}$ bezeichnet, müssen durch ein zu (4.13) ähnliches Gleichungssystem bestimmt werden. Sei dafür $\{b_j(x) : j = 1, \dots, \hat{m}\}$ eine Basis für \mathbb{P}_m^d . Üblicherweise ist dies eine Monombasis, d.h. $b_j(\vec{x}) = \vec{x}^\alpha = \prod_{k=1}^d x_k^{\alpha_k}$, wobei für den Multiindex $\sum_{k=1}^d \alpha_k \leq m$ gilt. Aus der Interpolationsbedingung und der Orthogonalitätsforderung folgt das Gleichungssystem

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \vec{\lambda} \\ \vec{\beta} \end{pmatrix} = \begin{pmatrix} \vec{y} \\ \vec{0} \end{pmatrix}, \quad (4.16)$$

mit $B_{ij} = b_j(\vec{x}_i)$, vergleiche z.B. Powell (2005), Gleichung (2.2).

Oft verwandte radiale Basisfunktionen sind in Tabelle 4.2 gegeben. Dabei gilt für die in Tabelle 4.2 angegebenen radialen Basisfunktionen, dass das Gleichungssystem immer nichtsingulär ist, falls die Zusatzbedingung (4.15) gegeben ist. Theoretische Herleitungen, wann eine radiale Basisfunktion in einer nichtsingulären Interpolationsmatrix resultiert, können in Buhmann (2008), S. 11 ff., S. 100 ff. gefunden werden. Die Konstanten α und c für die inverse Multiquadratische, die Multiquadratische bzw. die Gaußsche radiale Basisfunktion können beliebig größer 0 gewählt werden: Die resultierenden

Gauß	$\phi(t) = \exp(-\alpha t^2), \quad \alpha > 0$
Inverse Multiquadratische	$\phi(t) = (t^2 + c^2)^{-0.5}, \quad c > 0$
Linear	$\phi(t) = t$
Multiquadratisch	$\phi(t) = (t^2 + c^2)^{0.5}, \quad c > 0$
Kubisch	$\phi(t) = t^3$

Tabelle 4.2: Verschiedene radiale Basisfunktionen.

Funktionen entsprechend Gleichung (4.12) interpolieren immer die Beobachtungen. Die Vorhersagefähigkeit der Interpolationsfunktion kann stark durch unterschiedliche Werte beeinflusst werden. Für die multiquadratische Basisfunktion liefert in Powell (2001) der Wert $c = 2(1/n)^{1/d}$ gute Vorhersagewerte. Für die gaußsche radiale Basisfunktion scheint bisher keine Systematik vorzuliegen, wie die Konstante in dem hier gemachten Zusammenhang gewählt werden sollte. Fang et al. (2006), S. 177, schlagen eine Abwandlung des ML Ansatzes des Krigings zur Bestimmung vor, jedoch ergibt sich aus diesem Ansatz, dass nicht mehr das Gleichungssystem (4.16) zur Berechnung der Koeffizienten benutzt werden kann. Franke (1982) merkt an, dass für die gaußsche radiale Basisfunktion die Vorhersagefähigkeit stark von dem Parameter α abhängt, eine gute Wahl des Parameters jedoch wiederum von der unbekanntem Funktion abzuhängen scheint. Da keine konkrete Empfehlung gefunden wurde, auf welchen Wert der Parameter α gesetzt werden soll, wird der Parameter heuristisch auf $\alpha = 1$ festgelegt. Wird der Versuchsraum auf den Kubus $[-1, 1]^d$ normiert, ist dieser Wert ein Kompromis zwischen einer zu starken Lokalität der Basisfunktion um die Beobachtungen und einem zu großen Wert. Ein interessanter Aspekt der radialen Basisfunktionen in Tabelle 4.2 ist, dass es radiale Basisfunktionen gibt (linear, multiquadratisch, kubisch), die nicht für $t \rightarrow \infty$ gegen 0 konvergieren.

4.2.2 Thin Plate Spline

Eine besondere Stellung innerhalb der Familie der Radialen Basisfunktionen nehmen die sogenannten Thin Plate Splines ein (Micula (2002), Buhmann (2008), S. 61, S.115 ff.). Diese werden über einen Ansatz aus der Variationsrechnung hergeleitet: Es wird eine Funktion gesucht, die möglichst glatt ist unter der Nebenbedingung, dass die Beobachtungswerte interpoliert werden. Das Glattheitskriterium ist hier formal eine Seminorm auf einem Funktionenraum:

$$\|f\| := \int_{\mathbb{R}^d} \sum_{|\alpha|=\delta, \alpha \in \mathbb{Z}_+^d} \frac{\delta!}{\alpha!} (D^\alpha f(\vec{x}))^2 d\vec{x}, \delta \in \mathbb{N} \quad (4.17)$$

vgl. Buhmann (2008), S. 107. Dies ist keine Norm, da nicht nur das Nullelement auf die Null abgebildet wird, ansonsten erfüllen Seminormen die gleichen Bedingungen wie Normen. Bildlich gesprochen werden also alle Ableitungen der Ordnung δ quadriert über den \mathbb{R}^d integriert. Dies ist eine Verallgemeinerung der Herleitung von kubischen Splines im eindimensionalen auf den höherdimensionalen Fall. Als Menge aller Funktionen, in denen eine optimale Lösung gesucht wird, werden alle Funktionen verwandt, die eine endliche Seminorm entsprechend (4.17) haben und die Interpolationsbedingung erfüllen. Intuitiv ist klar, dass die optimale Lösung ein Polynom vom Grad $m = \delta - 1$ beinhaltet, da Polynome bis zum Grad $m = \delta - 1$ unter der Seminorm (4.17) invariant sind. Die radiale Basisfunktion sieht dann wie folgt aus:

$$\phi(r) = \begin{cases} r^{2\delta-d} \log(r), & 2\delta - d \text{ ist eine gerade ganze Zahl;} \\ r^{2\delta-d}, & 2\delta - d \text{ ist eine ungerade ganze Zahl,} \end{cases} \quad (4.18)$$

und es wird ein Polynom vom Grad $\delta - 1$ mit eingebunden. Ein entscheidender Nachteil des Thin Plate Splines in beliebiger Dimension ist jedoch, dass der Wert δ eine Bedingung erfüllen muss: $2\delta > d$, d.h. die kleinste mögliche Polynombasis wächst mit der Dimension des Problems. Nur wenn diese Bedingung gegeben ist, sind alle Funktionen in der betrachteten Klasse stetig (Buhmann (2008), S. 61, S. 106), was für die Vorhersage einer stetigen Funktion als eine Minimalanforderung angesehen werden kann. Ein

weiterer interessanter Gesichtspunkt ist, dass die radialen Basisfunktionen $r^2 \log(r)$ des Thin Plate Spline ohne ein Polynom unter Umständen zu singulären Interpolationsmatrizen führen, z.B. für den Spezialfall, dass $n = d + 1$ Punkte vorliegen, die einen regulären Simplex bilden, also alle den gleichen Punkt zu Punkt Abstand haben. Werden die Punkte so skaliert, dass der Abstand 1 ist, ergibt sich die Interpolationsmatrix zur Nullmatrix, die offensichtlich singulär ist (Powell (2001), S.4). Dies wird durch das Hinzufügen eines Polynoms behoben.

Desweiteren ist anzumerken, dass wenn der Wert δ der Seminorm immer minimal gewählt wird ($\delta = \lceil 0.5d \rceil$), die Differenzierbarkeit der resultierenden Vorhersagefunktion in den Beobachtungen von der Dimension abhängt ($r = \|\cdot\|$ ist nicht differenzierbar, $r^2 \log(r) = \|\cdot\|^{2.5} \log(\|\cdot\|^2)$ ist differenzierbar).

4.2.3 *Flexibilität der Interpolation mit radialen Basisfunktionen*

RBF Interpolation ist insgesamt eine sehr attraktive Interpolationsmethode, da sie flexibel über die Wahl der radialen Basisfunktion ist und einige Basisfunktionen, wie z.B. die multiquadratische Basisfunktion einen Tuningparameter zulassen. Auch ist RBF Interpolation in hohen Dimensionen und mit relativ vielen Beobachtungspunkten möglich, solange das zugrunde liegende Gleichungssystem gut konditioniert ist, was ein Vorteil zum Kriging ist. Dabei hat die Vorhersagefunktion (4.12) eine sehr große Ähnlichkeit zu der Vorhersagefunktion des Kriging (4.10).

Franke (1982) stellt fest, dass insbesondere die multiquadratische Basisfunktion in der Praxis sehr gute Vorhersagewerte liefert. Ein Vorteil von RBF Interpolation über das Krigingverfahren ist, dass die Koeffizienten nur durch ein lineares Gleichungssystem bestimmt werden und nicht wie beim Kriging über eine Maximierung der Loglikelihood. Jedoch liefert Interpolation mit radialen Basisfunktionen keine Unsicherheitsaussage wie das Krigingverfahren. Auch findet bei der Interpolation mit radialen Basisfunktionen keine Gewichtung der einzelnen Inputvariablen statt, wie dies beim Kriging mit

einer anisotropischen Kovarianzfunktion geschieht.

4.3 Inverse Distance Weighting

Eine Interpolationsmethode, die von Shepard (1968) vorgeschlagen wurde und seit dem in vielen Artikeln weiterentwickelt wurde (McLain (1974), Franke und Nielson (1980), Renka (1988)), ist das sogenannte Inverse Distance Weighting (IDW). Grundidee dabei ist, das man die Vorhersage für einen Punkte \vec{x} nur auf Beobachtungspunkten basieren lassen sollte, die in der Nähe von \vec{x} liegen. Diese Methode wird hier mit aufgenommen, da die im späteren Verlauf neu vorgestellte Methode der Kerninterpolation einen engen Bezug zu dem Inverse Distance Weighting hat. Die ursprüngliche Idee von Shepard sieht folgende Interpolationsfunktion vor:

$$\hat{f}(\vec{x}) = \frac{\sum_{i=1}^n d_i(\vec{x})^{-u} y_i}{\sum_{i=1}^n d_i(\vec{x})^{-u}}. \quad (4.19)$$

Dabei ist $d_i(\vec{x})$ eine Abstandsfunktion zwischen \vec{x} und \vec{x}_i , z.B. $d_i(\vec{x}) = \|\vec{x} - \vec{x}_i\|$. Shepard (1968) empfiehlt $u = 2$, und merkt an, dass die Vorhersage durch die Hinzunahme eines Richtungsfaktors in zwei Dimensionen weiter verbessert werden kann. Da dieses Konzept jedoch nicht für beliebige Dimensionen verallgemeinert werden kann, wird es hier nicht erläutert. Die Vorhersagefunktion in Gleichung (4.19) hat Gradienten 0 in den Beobachtungen, was intuitiv nicht attraktiv ist. Shepard selbst schlägt eine Methode vor, um den Wert der Ableitung in den Beobachtungen auf einen gewünschten Wert zu setzen. Da jedoch unklar ist, wo diese Informationen herkommen sollen, wird auch dies nicht weiter beschrieben.

McLain (1974) beschreibt eine Verbesserung des Inverse Distance Weighting, die Distance Weighted Least Squares Approximation genannt wird. Dabei wird eine gewichtete kleinste Quadrate Regression an die Daten angepasst mit einer polynomialen Regressionsfunktion. Ist dafür $P_{\theta}(\vec{x})$ das Polynom (meistens linear oder quadratisch), das lokal an die Daten angepasst werden soll, so ergibt sich der geschätzte Parametervektor

$\vec{\theta}$ im Punkt \vec{x} durch das folgende Minimierungsproblem:

$$\min_{\vec{\theta}} \sum_{i=1}^n (P_{\vec{\theta}}(\vec{x}_i) - y_i)^2 w(d_i(\vec{x})). \quad (4.20)$$

McLain (1974) schlägt dabei als Gewichtsfunktion $w(t^2) = \exp(-\alpha t^2)/t^2$ mit einem Tuningparameter α vor. Es wird also für einen Vorhersagepunkt \vec{x} genau das Polynom gesucht, welches im kleinste Quadrate Sinne optimal ist. Problematisch daran ist, dass für jeden Vorhersagepunkt \vec{x} eine neue kleinste Quadrate Regression berechnet werden muss, man also genaugenommen den Parametervektor $\vec{\theta}$ in Abhängigkeit von \vec{x} schreiben müsste ($\vec{\theta}(\vec{x})$).

Deswegen schlagen Franke und Nielson (1980) und Renka (1988) eine Abänderung vor, bei der nicht mehr für jeden Vorhersagepunkt eine Kleinste Quadrate Regression durchgeführt werden muss. Dabei wird in jedem Beobachtungspunkt eine lokale Approximation $Q_i(\vec{x})$ bestimmt, die genau diesen Beobachtungspunkt interpoliert. Als lokale Approximation wird ein Polynom (in den Artikeln von Franke und Nielson (1980) und Renka (1988) ein Polynom zweiten Grades) über eine gewichtete (homogene) kleinste Quadrate Anpassung für $Q_i(\vec{x})$ bestimmt. Werden die Ausgangsdaten passend transformiert ($\vec{x}_{l,i} := \vec{x}_l - \vec{x}_i, y_{l,i} := y_l - y_i$) kann durch die homogene Regression erreicht werden, dass die Funktion $Q_i(\vec{x})$ den Punkt \vec{x}_i interpoliert. Als Gewichte für die Regressionsanpassung in jedem Beobachtungspunkt wird

$$q_i(\vec{x}) := \left(\frac{(R_{q,i} - \|\vec{x} - \vec{x}_i\|)_+}{R_{q,i} \|\vec{x} - \vec{x}_i\|} \right)^2 \quad (4.21)$$

mit

$$(R_{q,i} - t)_+ = \begin{cases} R_{q,i} - t, & \text{falls } t < R_{q,i} \\ 0, & \text{sonst} \end{cases} \quad (4.22)$$

verwandt. Die Parameter $R_{q,i}$ werden bei Franke und Nielson (1980) konstant gewählt, bei Renka (1988) aber so, dass immer eine feste Anzahl N_q an Beobachtungspunkten innerhalb des Radius um den Punkt \vec{x}_i liegt. Zu beachten ist, dass N_q immer groß genug gewählt werden muss, damit sich die homogene Regression auch durchführen lässt.

Dieser erste Gewichtungsprozess bezieht sich nur auf die Funktionen $Q_i(\vec{x})$, die eine lokale Anpassung an die Daten um den Punkt \vec{x}_i ergeben soll. Prinzipiell könnten hier auch andere Gewichtungen wie z.B. bei Kernregressionsmethoden verwandt werden, das einzig entscheidende ist, dass die Funktion $Q_i(\vec{x})$ den Beobachtungspunkt (\vec{x}_i, y_i) exakt reproduziert.

Als interpolierende Funktion ergibt sich dann

$$\hat{f}(\vec{x}) = \frac{\sum_{i=1}^n Q_i(\vec{x}) w_i(\vec{x})}{\sum_{j=1}^n w_j(\vec{x})} \quad (4.23)$$

mit einem zweiten Gewichtungsprozess $w_i(\vec{x})$, für den Franke und Nielson (1980) und Renka (1988) folgende Gewichtsfunktion $w_i(\vec{x})$ vorschlagen:

$$w_i(\vec{x}) := \left(\frac{(R_{w,i} - \|\vec{x} - \vec{x}_i\|)_+}{R_{w,i} \|\vec{x} - \vec{x}_i\|} \right)^2. \quad (4.24)$$

Formal ist das die gleiche Funktion wie in Gleichung (4.21), nur mit anderen Parametern. Der Parameter $R_{w,i}$ stellt den Radius um den Punkt \vec{x}_i dar, in dem der Punkt Einfluss hat. Während Franke und Nielson (1980) diesen Parameter wieder fest wählen, schlägt Renka (1988) vor, auch diesen Parameter variabel so zu wählen, dass immer genau N_w Punkte in dem Radius um \vec{x}_i sind. Die resultierende Funktion $\hat{f}(\vec{x})$ interpoliert die Beobachtungen, da für $\vec{x} \rightarrow \vec{x}_i$ nur die Funktion $Q_i(\vec{x})$ positives Gewicht erhält, alle anderen Funktionen $Q_{i'}(\vec{x}), i' \neq i$ erhalten im Grenzwert das Gewicht 0. Die Funktion $Q_i(\vec{x})$ ist aber gerade so konstruiert, dass in dem Punkt \vec{x}_i der Wert y_i ausgegeben wird. Da bei Renka (1988) nur zweidimensionale Interpolationsprobleme betrachtet werden und von Datensituationen mit relativ vielen Beobachtungspunkten ausgegangen wird, sind die dortigen Parameterempfehlungen hier nicht gut anwendbar, weswegen der Parameter N_q so gewählt wird, dass er nahezu minimal ist und $N_w \approx n/3$ gewählt wird.

Ein prinzipielles Problem bei den Modifikationen von McLain (1974), Franke und Nielson (1980) und Renka (1988) ist, dass ein Polynom zweiten Grades angepasst wird. Dies impliziert, dass in d Dimensionen mindestens $0.5d^2 + 1.5d + 1$ Beobachtungen vorliegen müssen. Dies ist im Kontext von Computerexperimenten eine Einschränkung, der

nicht immer entsprochen werden kann. Insbesondere die Modifikationen von Franke und Nielson (1980) und Renka (1988) haben eine große Ähnlichkeit zu der später vorgestellten Kerninterpolation. Inverse Distance Weighting Interpolation wird in Ripley (2004) als nichtparametrische Alternative zum Kriging in der Geostatistik aufgeführt.

4.4 Weitere Methoden

Es existiert eine Vielzahl an weiteren Methoden zur mehrdimensionalen Interpolation, auf die hier aus verschiedenen Gründen nicht eingegangen wird. Polynomiale Interpolation im eindimensionalen kann problemlos auf den höherdimensionalen Fall übernommen werden, jedoch ergibt sich auch hier das Problem von großen Oszillationen bei höherem Polynomgrad. Ausserdem ist im Allgemeinen unklar, welches Polynom für die Interpolation verwandt werden soll. Bates et al. (2008) erarbeiten eine Methode, die genau diesen Nachteil der polynomialen Interpolation beseitigt: Es wird das Polynom gesucht, das eine möglichst kleine Variation im Sinne der zweiten Ableitung/ Hessematrix liefert. Dies hat eine enge Verwandtschaft zum Thin Plate Spline, da dieser das gleiche Maß für Variation in einer geeigneten (nichtpolynomialen) Funktionenklasse minimiert. Da die Methode von Bates et al. (2008) dem Verhalten von dem Thin Plate Spline sehr nahe kommt, wird sie hier ebenfalls nicht betrachtet.

Ein einfacher Ansatz zu der Interpolationsaufgabe sind stückweise konstante Funktionen in Verbindung mit nächste Nachbarn Regeln, da diese jedoch noch nicht einmal die Minimalanforderung der Stetigkeit erfüllen, werden sie hier außer Acht gelassen. Ebenso nicht betrachtet werden Methoden, die aufbauend auf einer Triangulation über den einzelnen Simplexen stückweise Polynome verwenden (Lawson (1976)), da diese nicht ohne weiteres in eine beliebige Dimension verallgemeinert werden können.

Eine weitere Interpolationsmethode ist die Natürliche Nachbarn Interpolation, die das Konzept des Voronoidiagramm für eine Vorhersage ausnutzt. Diese Methode hat sich jedoch in einer Simulationsstudie (Mühlenstädt und Kuhnt (2009)) als nicht attraktiv

erwiesen und wird deswegen nicht betrachtet.

Es gibt eine Vielzahl an weiteren Methoden, die für die Modellierung von Computer Experimenten herangezogen werden können wie z.B. Multivariate adaptive regression splines und Artificial Neural networks (Chen et al. (2006)). Diese wurden nicht mit aufgenommen, da in Chen et al. (2006) diese Modelle im Allgemeinen keine gute Vorhersagewerte lieferten.

KAPITEL 5

KERNINTERPOLATION

In Kapitel 3 wurden Versuchspläne vorgestellt, die durch eine stückweise lineare Funktion, die über eine Triangulation definiert ist, motiviert wurden. Jedoch gibt es einige prinzipielle Nachteile einer solchen Modellierung. Die Vorhersagefunktion des Modells ist nicht differenzierbar und nur innerhalb der konvexen Hülle definiert, d.h. es können keine Werte außerhalb der konvexen Hülle der Beobachtungswerte vorhergesagt werden. Aus diesem Grund soll, aufbauend auf dem stückweise linearen Modell, ein Modell herausgearbeitet werden, dass diese Nachteile beseitigt. Ein ähnliches Ziel wird zum Beispiel mit Methoden wie der Lawsoninterpolation in zwei Dimensionen verfolgt (Lawson (1976)), jedoch ist hier nicht ohne weiteres eine Verallgemeinerung auf beliebige Dimensionen möglich.

5.1 Allgemeine Idee: Kombiniere lokal angepasste lineare Funktionen

Die generelle Idee zur Verbesserung der stückweise linearen Funktion ist, die lokal angepassten linearen Funktionen global zu betrachten und dann ein geeignetes Mittel über die unterschiedlichen Vorhersagewerte zu berechnen. Dafür sei wieder $T(P_n)$ die Delaunaytriangulation der Punktmenge $P_n = \{\vec{x}_1, \dots, \vec{x}_n\}$, bestehend aus den Simplexen $S_j, j = 1, \dots, N$ mit der gleichen Notation für Ecken wie in Kapitel 2 und 3. An die Simplexe wird die in Kapitel 2 vorgestellte stückweise lineare Funktion (Bemerkung

2.1) angepasst. Die auf dem Simplex S_j definierte Funktion wird mit $y^j(\vec{x})$ bezeichnet. Diese linearen Funktionen sind nicht nur innerhalb des Simplexes eine gute erste Näherung für die Funktion, sondern auch in einer begrenzten Umgebung um den Simplex. Die Funktionen $y^j(\vec{x})$ werden mit Hilfe einer Gewichtsfunktion $g_j(\vec{x})$ in der folgenden Weise zu einer globalen Vorhersage kombiniert:

$$\hat{y}(\vec{x}) := \begin{cases} y_i, & \vec{x} = \vec{x}_i, i = 1, \dots, n; \\ \frac{\sum_{j=1}^N g_j(\vec{x}) y^j(\vec{x})}{\sum_{j=1}^N g_j(\vec{x})}, & \text{sonst,} \end{cases} \quad (5.1)$$

was einem gewichteten Mittel der N unterschiedlichen Vorhersagewerte entspricht. Von den Eigenschaften der Gewichtsfunktion $g_j(\vec{x})$ hängt ab, welche Eigenschaften $\hat{y}(\vec{x})$ aufweist.

5.2 Auswahl der Gewichtsfunktion

Um eine stetige Interpolationsfunktion zu erzielen, müssen die Gewichte gut gewählt werden. Dafür sei die Gewichtsfunktion in zwei Teile aufgeteilt:

$$g_j(\vec{x}) := w_{h,j}(D_j(\vec{x})), \quad (5.2)$$

wobei $D_j(\vec{x})$ eine Abstandsfunktion zwischen \vec{x} und Simplex S_j ist und $w_{h,j}(t)$ eine Kernfunktion ist, die von einem Parameter h abhängt und/oder von Eigenschaften des Simplex S_j abhängt. Die in der folgenden Definition aufgeführten Bedingungen sind hinreichend dafür, dass sich eine stetige Interpolationsfunktion ergibt.

Definition 5.1. Eine Funktion $D_j \in C^0(\mathbb{R}^d, \mathbb{R}_{\geq 0})$ mit

(D1) $\bigcap_{\epsilon > 0} \eta_\epsilon^j = \{\vec{x}_0^j, \dots, \vec{x}_d^j\}$, wobei $\eta_\epsilon^j := \{\vec{x} \mid D_j(\vec{x}) < \epsilon\}$ und

(D2) $D_j(\vec{x}) \xrightarrow{\|\vec{x}\| \rightarrow \infty} \infty$

heißt reguläre Abstandsfunktion. Eine Funktion $w_h \in C^0(\mathbb{R}_{>0}, \mathbb{R}_{\geq 0})$ mit

(K1) $w_{h,j}(t) \xrightarrow{t \rightarrow 0} \infty$ und

(K2) $w_{h,j}(t) \xrightarrow{t \rightarrow \infty} 0$

heißt reguläre Kernfunktion für $\hat{y}(\vec{x})$. Eine reguläre Gewichtsfunktion $g_j(\vec{x}) = w_{h,j}(D_j(\vec{x}))$ ist gegeben, wenn die folgenden beiden Bedingungen zutreffen:

(W1) $w_{h,j}$ und D_j sind regulär.

(W2) Die Divergenzordnung von $g_j(\vec{x})$ für $\vec{x} \rightarrow \vec{x}_i^j, i = 0, \dots, d$, hängt nicht von S_j ab, d.h. $g_j(\vec{x}) = O(g)$ für $\vec{x} \rightarrow \vec{x}_i^j$ wobei g nicht von j abhängt.

Für eine reguläre Abstandsfunktion $D_j(\vec{x})$ wird also nicht gefordert, dass sie minimal wird in Punkten innerhalb des Simplex S_j , sondern nur, dass sie minimal wird, wenn \vec{x} gegen eine Ecke von S_j konvergiert. Die Kernfunktion ist so definiert, dass für $\vec{x} \rightarrow \vec{x}_i$, die einzigen Simplexe, die im Grenzwert ein strikt positives Gewicht bekommen, jene sind, die \vec{x}_i als Ecke haben.

In dieser Arbeit wird die reguläre Abstandsfunktion

$$D_j(\vec{x}) = \prod_{i=0}^d \|\vec{x} - \vec{x}_i^j\| \quad (5.3)$$

und die Kernfunktion

$$w_{h,j}(t) := \frac{v_j}{t^h} \quad (5.4)$$

verwandt. Dabei ist v_j das Volumen von Simplex S_j . Die Gewichtung mit dem Volumen ändert nichts an dem Konvergenzverhalten für $\vec{x} \rightarrow \vec{x}_i$, allerdings wird dadurch der sehr unterschiedlichen Größe der Simplexe Rechnung getragen. Daraus resultiert die Gewichtsfunktion

$$g_j(\vec{x}) = \frac{v_j}{\left(\prod_{i=0}^d \|\vec{x} - \vec{x}_i^j\|\right)^h}. \quad (5.5)$$

In allen Anwendungsbeispielen dieser Arbeit wurde die Gewichtsfunktion (5.5) mit $h = 2$ für die Kernfunktion verwandt, da diese erstens zu guten Vorhersageergebnissen geführt hat und zweitens eine glatte Vorhersagefunktion entsteht.

Viele andere Wahlen von Abstands- und Kernfunktion sind möglich. Abhängig von $D_j(\vec{x})$ und $w_{h,j}(t)$ können unterschiedliche Eigenschaften der Kerninterpolationsfunktion erreicht werden, wie z.B. Interpolationsfunktionen mit unterschiedlicher Differenzierbarkeit oder mit nur einem lokalen Träger für die Vorhersage.

5.3 Eigenschaften der Kerninterpolation

Die Kerninterpolation mit der Vorhersagefunktion $\hat{y}(x)$ hat verschiedene interessante Eigenschaften, die nachfolgend erläutert werden. Die wichtigste ist, dass die Vorhersagefunktion stetig ist, wenn reguläre Kern- und Abstandsfunken im Sinne von Definition 5.1 verwandt werden.

Satz 5.1.

- (i) Jede Kerninterpolationsfunktion $\hat{y}(\vec{x})$ ist linear in $(y_1, \dots, y_n)'$ für $\vec{x} \in \mathfrak{D} \setminus \{\vec{x}_1, \dots, \vec{x}_n\}$.
- (ii) Jede Kerninterpolationsfunktion $\hat{y}(\vec{x})$ mit regulärer Gewichtsfunktion ist stetig und interpoliert die Beobachtungen $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$.

Beweis von Satz 5.1. $\hat{y}(\vec{x})$ interpoliert die Beobachtungen nach Definition. Um zu sehen, dass $\hat{y}(\vec{x})$ stetig ist wird eine Fallunterscheidung gemacht. Für $\vec{x} \notin \{\vec{x}_1, \dots, \vec{x}_n\}$ ist $\hat{y}(\vec{x})$ stetig, da alle verwandten Teilfunktionen stetig sind. Für $\vec{x}_i \in \{\vec{x}_1, \dots, \vec{x}_n\}$ können die Simplexe in der Triangulation $T(P)$ in zwei komplementäre Mengen aufgeteilt werden: T_i enthält die Simplexe, die \vec{x}_i als Ecke haben, T_{-i} enthält die restlichen Simplexe, die \vec{x}_i nicht als Ecke haben. Die Indexmenge der Simplexe in T_i wird mit N_i

bezeichnet, die von T_{-i} mit N_{-i} . Dann gilt:

$$\frac{\sum_{j=1}^N g_j(\vec{x})y^j(\vec{x})}{\sum_{j'=1}^N g_{j'}(\vec{x})} = \frac{\sum_{j \in N_i} g_j(\vec{x})y^j(\vec{x})}{\sum_{j'=1}^N g_{j'}(\vec{x})} + \frac{\sum_{j \in N_{-i}} g_j(\vec{x})y^j(\vec{x})}{\sum_{j'=1}^N g_{j'}(\vec{x})}. \quad (5.6)$$

Definition 5.1 ist so konstruiert, dass für $\vec{x} \rightarrow \vec{x}_i, \vec{x} \in \mathfrak{D}$ alle lokalen Vorhersagen $y^j(\vec{x})$, die zu einem Simplex in T_{-i} gehören, beschränktes Gewicht $g_j(\vec{x}_i)$ haben. Alle lokalen Vorhersagen $y^j(\vec{x})$ die zu einem Simplex in T_i gehören, erhalten Gewicht $g_j(\vec{x}) \rightarrow \infty$ für $\vec{x} \rightarrow \vec{x}_i$. Deswegen hat der zweite Teil der Summe in Gleichung (5.6) den Grenzwert 0 und kann vernachlässigt werden. Da im Grenzwert Konstanten im Nenner keinen Einfluss auf die Divergenzordnung haben können alle Indizes in N_{-i} weggelassen werden. Daraus folgt

$$\frac{\sum_{j \in N_i} g_j(\vec{x})y^j(\vec{x})}{\sum_{j' \in N_i} g_{j'}(\vec{x})} \xrightarrow{\vec{x} \rightarrow \vec{x}_i} \sum_{j \in N_i} c_{i,j}y^j(\vec{x}_i) \quad (5.7)$$

mit $\sum_{j \in N_i} c_{i,j} = 1, c_{i,j} \geq 0 \forall j \in N_i$. Dies folgt aufgrund von Eigenschaft (W2) in Definition 5.1, die fordert, dass alle übrigen Gewichtsfunktionen $g_j(\vec{x})$ mit $j \in N_i$ die gleiche Divergenzordnung haben. Für alle lokalen Vorhersagen $y^j(\vec{x})$ mit Simplex $S_j \in T_i$ gilt $y^j(\vec{x}_i) = y_i$ laut Konstruktion. Folglich ist $\hat{y}(x)$ stetig. Weiterhin ist leicht zu sehen, dass $\hat{y}(x)$ linear in y_1, \dots, y_n ist. \square

Die Kerninterpolationsfunktion $\hat{y}(x)$ ist differenzierbar ausserhalb der Beobachtungen, wenn die Gewichtsfunktion ebenfalls differenzierbar ist.

Satz 5.2. *Wenn die Gewichtsfunktion $g_j(\vec{x}) = w_{j,h}(D_j(\vec{x}))$ stetig differenzierbar ist, so ist auch $\hat{y}(x)$ stetig differenzierbar für alle $x \in \mathfrak{D} \setminus \{\vec{x}_1, \dots, \vec{x}_n\}$.*

Beweis von Satz 5.2. Die Differenzierbarkeit ausserhalb der Beobachtungen ergibt sich, da $g_j(\vec{x})$ nach Voraussetzung stetig differenzierbar ist, die linearen Funktionen $y^j(\vec{x})$ unendlich oft stetig differenzierbar sind und die Vorhersagefunktion aus Gleichung (5.1) die Gewichtsfunktion und die linearen Funktionen stetig differenzierbar verknüpfen. \square

Die Differenzierbarkeit in den Beobachtungspunkten hängt davon ab, wie schnell die Gewichte für Simplexe, die \vec{x}_i als Ecke haben, gegen ∞ divergieren:

Satz 5.3. Sei $\vec{x}_i \in \{\vec{x}_1, \dots, \vec{x}_n\}$ und $g_j(\vec{x}) \in C^1(\mathbb{R}^d \setminus \{\vec{x}_1, \dots, \vec{x}_n\}, \mathbb{R})$. Ferner gelte

$$\frac{1}{\|\vec{t}\| \sum_{j'=1}^N g_{j'}(\vec{x}_i + \vec{t})} \xrightarrow{\vec{t} \rightarrow 0} 0, \quad (5.8)$$

wobei $\vec{t} \in \mathbb{R}^d$. Dann ist die Kerninterpolationsfunktion $\hat{y}(\vec{x}_i)$ (total) differenzierbar in \vec{x}_i . Die partielle Ableitung in Richtung des k -ten Einheitsvektors in dem Punkt \vec{x}_i ist gegeben durch

$$\frac{\partial}{\partial x_k} \hat{y}(\vec{x}_i) = \sum_{j \in N_i} c_{i,j} \beta_k^j, \quad (5.9)$$

wobei $c_{i,j}$ die Konstanten aus dem Beweis von Satz 5.1 sind.

Beweis von Satz 5.3. Für den Beweis wird die gleiche Notation (N_i, N_{-i}) wie im Beweis von Satz 5.1 verwandt. Für die totale Differenzierbarkeit muss gezeigt werden, dass

$$\lim_{\vec{t} \rightarrow 0} \frac{\hat{y}(\vec{x}_i + \vec{t}) - y_i - L\vec{t}}{\|\vec{t}\|} = 0, \quad (5.10)$$

wobei L ein Kandidat für die Ableitung von \hat{y} im Punkt \vec{x}_i ist. Dieser Kandidat ist hier

$$L = \sum_{j \in N_i} c_{i,j} \beta^{j^T}, \quad (5.11)$$

der durch Berechnung der partiellen Ableitungen von $\hat{y}(\vec{x}_i)$ erhalten wird. Weiter kann der Wert y_i auf folgende Art umgeschrieben werden:

$$\forall j \in N_i : y_i = \beta_0^j + \beta^{j^T} \vec{x}_i = \frac{\sum_{j \in N_i} g_j(\vec{x}_i + \vec{t})(\beta_0^j + \beta^{j^T} \vec{x}_i)}{\sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})} \quad (5.12)$$

da für alle $j \in N_i$ gilt: $y^j(\vec{x}_i) = y_i$. Die Summe von $\hat{y}(\vec{x})$ kann wieder entsprechend N_i und N_{-i} aufgeteilt werden:

$$\begin{aligned} & \lim_{\vec{t} \rightarrow 0} \frac{\hat{y}(\vec{x}_i + \vec{t}) - y_i - L\vec{t}}{\|\vec{t}\|} \\ &= \lim_{\vec{t} \rightarrow 0} \left[\frac{\sum_{j \in N_i} (\beta_0^j + \beta^{j^T} (\vec{x}_i + \vec{t})) g_j(\vec{x}_i + \vec{t})}{\|\vec{t}\| \sum_{j' \in N} g_{j'}(\vec{x}_i + \vec{t})} \right] \end{aligned} \quad (5.13)$$

$$\begin{aligned}
& + \frac{\sum_{j \in N_{-i}} (\beta_0^j + \beta^{j^T}(\vec{x}_i + \vec{t})) g_j(\vec{x}_i + \vec{t})}{\|\vec{t}\| \sum_{j' \in N} g_{j'}(\vec{x}_i + \vec{t})} \\
& - \frac{\sum_{j \in N_i} (\beta_0^j + \beta^{j^T} \vec{x}_i) g_j(\vec{x}_i + \vec{t})}{\|\vec{t}\| \sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})} \\
& - \frac{\sum_{j \in N_i} c_{i,j} \beta^{j^T} \vec{t}}{\|\vec{t}\|} \Big].
\end{aligned}$$

Der zweite Teil der Summe ergibt im Grenzwert 0 aufgrund der Bedingung (5.8) und der Definition von N_{-i} . Für den ersten Teil kann der Index der Summe im Nenner zu N_i umgeschrieben werden da Konstanten im Grenzwert keinen Einfluss haben. Damit folgt

$$\begin{aligned}
& = \lim_{\vec{t} \rightarrow 0} \left[\frac{\sum_{j \in N_i} (\beta_0^j + \beta^{j^T}(\vec{x}_i + \vec{t})) g_j(\vec{x}_i + \vec{t})}{\|\vec{t}\| \sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})} \right. \\
& - \frac{\sum_{j \in N_i} (\beta_0^j + \beta^{j^T} \vec{x}_i) g_j(\vec{x}_i + \vec{t})}{\|\vec{t}\| \sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})} \\
& \left. - \frac{\sum_{j \in N_i} c_{i,j} \beta^{j^T} \vec{t}}{\|\vec{t}\|} \right] \tag{5.14}
\end{aligned}$$

$$= \lim_{\vec{t} \rightarrow 0} \left[\sum_{j \in N_i} \frac{(\beta_0^j + \beta^{j^T}(\vec{x}_i + \vec{t}) - (\beta_0^j + \beta^{j^T} \vec{x}_i)) g_j(\vec{x}_i + \vec{t})}{\|\vec{t}\| \sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})} - \frac{c_{i,j} \beta^{j^T} \vec{t}}{\|\vec{t}\|} \right] \tag{5.15}$$

$$= \lim_{\vec{t} \rightarrow 0} \left[\sum_{j \in N_i} \frac{\beta^{j^T} \vec{t} g_j(\vec{x}_i + \vec{t})}{\|\vec{t}\| \sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})} - \frac{c_{i,j} \beta^{j^T} \vec{t}}{\|\vec{t}\|} \right] \tag{5.16}$$

$$= \lim_{\vec{t} \rightarrow 0} \left[\sum_{j \in N_i} \left(\frac{g_j(\vec{x}_i + \vec{t})}{\sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})} - c_{i,j} \right) \frac{\beta^{j^T} \vec{t}}{\|\vec{t}\|} \right] \tag{5.17}$$

Da $\frac{\vec{t}}{\|\vec{t}\|} = O(1)$ kann dieser Term vernachlässigt werden. Die Konstanten $c_{i,j}$ sind gerade definiert als der Grenzwert von $\frac{g_j(\vec{x}_i + \vec{t})}{\sum_{j' \in N_i} g_{j'}(\vec{x}_i + \vec{t})}$ für $\vec{t} \rightarrow 0$, was den Beweis vervollständigt. \square

Die Bedingung aus Gleichung (5.8) wird im Folgenden für die Gewichtsfunktion aus Gleichung (5.5) überprüft. Um Differenzierbarkeit zu erhalten muss gelten:

$$\frac{1}{\|\vec{t}\| \sum_{j=1}^N \frac{v_j}{(\prod_{k=0}^d \|\vec{x}_{i'} + \vec{t} - \vec{x}_k^j\|)^h}} \rightarrow 0 \text{ für } \vec{t} \rightarrow 0. \quad (5.18)$$

Zur Vereinfachung der Notation sei o.B.d.A. $\vec{x}_{i'} = 0$. Weiterhin sollen die Ecken aller Simplexe S_j mit $j \in N_i$ so geordnet sein, dass $\vec{x}_0^j = \vec{x}_{i'} (= 0)$. Die Summe im Nenner kann wieder aufgeteilt werden in zwei Teilsummen entsprechend $N_{i'}$ und $N_{-i'}$. Alle Simplexe mit Index in $N_{-i'}$ haben keinen Einfluss da ihr Gewicht im Grenzwert endlich ist. Folglich vereinfacht sich Gleichung (5.18) zu

$$\lim_{\vec{t} \rightarrow 0} \frac{1}{\|\vec{t}\| \sum_{j \in N_i} \frac{v_j}{\|\vec{t}\|^h \prod_{k=1}^d \|\vec{t} - \vec{x}_k^j\|^h}} = \lim_{\vec{t} \rightarrow 0} \frac{\|\vec{t}\|^{h-1}}{\sum_{j \in N_i} \frac{v_j}{\prod_{k=1}^d \|\vec{t} - \vec{x}_k^j\|^h}}, \quad (5.19)$$

was für $h > 1$ gegen 0 konvergiert.

Bei Kernregressionsmethoden in der Statistik spielt meistens der Glättungsparameter eine zentrale Rolle in Bezug auf Güteeigenschaften. Hier hat der Glättungsparameter h der Kernfunktion auch einen starken Einfluss. Für die Kernfunktion $w_{j,h}(t) = \frac{v_j}{t^h}$, welche im Folgenden immer verwandt wird, gelten die Eigenschaften des folgenden Satzes.

Satz 5.4. *Betrachte die Kerninterpolationsfunktion $\hat{y}(\vec{x})$ mit beliebiger regulärer Abstandsfunktion $D_j(\vec{x})$, Kernfunktion $w_{j,h}(t) = \frac{v_j}{t^h}$ und Beobachtungen $(y_1, \vec{x}_1), \dots, (y_n, \vec{x}_n)$. Dann gilt:*

(i) Für $h \rightarrow \infty$: $\hat{y}(\vec{x}) \rightarrow \sum_{j \in N^*(\vec{x})} \frac{y^j(\vec{x})v_j}{\sum_{j' \in N^*(\vec{x})} v_{j'}}$ wobei

$$N^*(\vec{x}) := \{j \in N \mid D_j(\vec{x}) = \min\{D_1(\vec{x}), \dots, D_N(\vec{x})\}\} \quad (5.20)$$

die Menge aller Indizes j ist, für die $D_j(\vec{x})$ minimal ist.

(ii) Für $h \rightarrow 0$ gilt: $\hat{y}(\vec{x}) \rightarrow \sum_{j=1}^N \frac{y^j(\vec{x})v_j}{\sum_{j'=1}^N v_{j'}}$.

Beweis von Satz 5. Unter Ausnutzung der Definition von $\hat{y}(\vec{x})$ für $\vec{x} \neq \vec{x}_i$ gilt

$$\hat{y}(\vec{x}) = \sum_{j=1}^N \frac{y^j(\vec{x})}{\sum_{j'=1}^N \left(\frac{D_j(\vec{x})}{D_{j'}(\vec{x})} \right)^h \frac{v_{j'}}{v_j}}.$$

Für die meisten \vec{x} Werte wird $N^*(\vec{x})$ nur aus einem Index bestehen, aber es existieren \vec{x} Punkte, für die mehr als ein $D_j(\vec{x})$ das Minimum annimmt. Betrachte $\left(\frac{D_j(\vec{x})}{D_{j'}(\vec{x})} \right)^h$ für festes j : Wenn $j \in N^*(\vec{x})$ gilt

$$\forall j' : \frac{D_j(\vec{x})}{D_{j'}(\vec{x})} \leq 1,$$

wobei Gleichheit genau dann gilt wenn $j' \in N^*(\vec{x})$. Daraus folgt

$$\sum_{j'=1}^N \left(\frac{D_j(\vec{x})}{D_{j'}(\vec{x})} \right)^h \frac{v_{j'}}{v_j} \xrightarrow{h \rightarrow \infty} \sum_{j \in N^*(\vec{x})} \frac{v_{j'}}{v_j}$$

Für $j \notin N^*(\vec{x})$ gilt dass es mindestens ein j' gibt mit $D_j > D_{j'}$. Folglich gilt:

$$\exists j' : \frac{D_j(\vec{x})}{D_{j'}(\vec{x})} > 1$$

und

$$\sum_{j'=1}^N \left(\frac{D_j(\vec{x})}{D_{j'}(\vec{x})} \right)^h \frac{v_{j'}}{v_j} \xrightarrow{h \rightarrow \infty} \infty.$$

Zusammen folgt daraus

$$\sum_{j=1}^N \frac{y^j(\vec{x})}{\sum_{j'=1}^N \left(\frac{D_j(\vec{x})}{D_{j'}(\vec{x})} \right)^h \frac{v_{j'}}{v_j}} \xrightarrow{h \rightarrow \infty} \sum_{j \in N^*(\vec{x})} \frac{y^j(\vec{x}) v_j}{\sum_{j' \in N^*(\vec{x})} v_{j'}}.$$

Der zweite Teil des Satzes folgt aus dem Grenzwert

$$\forall c \in \mathbb{R}_{>0} : c^h \xrightarrow{h \rightarrow 0} 1.$$

□

Der Parameter h ist dementsprechend wichtig für eine gute Anpassung. Heuristisch wurden mit $h = 2$ gut Ergebnisse gewählt. Ein weiteres Resultat ist, dass je größer der

Abstand zwischen dem Vorhersagepunkt \vec{x} und den Beobachtungen ist, desto unzuverlässiger ist die Vorhersage:

$$\forall j : \frac{g_j(\vec{x})}{\sum_{j'=1}^N g_{j'}(\vec{x})} \rightarrow \frac{v_j}{\sum_{j'=1}^N v_{j'}} \quad \text{für} \quad \min_{\vec{z} \in \text{conv}(\vec{x}_1, \dots, \vec{x}_n)} \|\vec{x} - \vec{z}\| \rightarrow \infty, \quad (5.21)$$

d.h. die Vorhersage ist in diesem Fall ein gewichtetes arithmetisches Mittel mit den (normierten) Gewichten v_j als Gewichtsvektor. Dies ist rein intuitiv kein sinnvoller Wert für eine Extrapolation.

Um die Kerninterpolation zu implementieren, wird ein effizienter Algorithmus zur Bestimmung der Triangulation benötigt. Wie in Kapitel 3 wird der quickhull Algorithmus von Barber et al. (1996) verwandt, der schon in R (im Package `geometry`) und Matlab implementiert ist. Außerdem sollte angemerkt werden, dass für Simplexe mit keinem oder sehr kleinem Volumen die lineare Funktion, die über dem Simplex definiert ist, nicht berechnet werden kann. Diese Simplexe werden von der Vorhersage ausgeschlossen. Solche Simplexe treten oft am Rand der konvexen Hülle auf. Die notwendige Rechenarbeit sollte gut organisiert werden, um eine akzeptable Rechenzeit zu erreichen. Dafür können die Koeffizienten für die linearen Funktionen separat berechnet und gespeichert werden, so dass sie wiederverwendet werden können. Auch kann durch geschicktes Abspeichern der euklidischen Abstände zwischen einem Vorhersagepunkt \vec{x} und den Beobachtungspunkten $\vec{x}_1, \dots, \vec{x}_n$ viel Rechenzeit gespart werden.

Die hier vorgestellte Methode hat eine enge Beziehung zum Inverse Distance Weighting, insbesondere zu der Version des Inverse Distance Weighting von Franke und Nielson (1980) und Renka (1988). Dort wird in jedem Beobachtungspunkt ein Polynom (in den Originalartikeln ein quadratisches) über gewichtete lineare Regression angepasst, dass diesen Beobachtungspunkt interpoliert und der Gewichtungsprozess anschließend über die einzelnen Vorhersagen der n Polynome vorgenommen. In dem hier vorgestellten Ansatz wird anstattdessen an jeden Simplex aus der Delaunaytriangulation über den Versuchsplan eine lineare Funktion angepasst und der Gewichtungsprozess auf die N verschiedenen Vorhersagen für jeden Simplex angewandt. Ein prinzipieller Vorteil der hier vorgestellten Methode, ist, dass die ursprünglich von Franke und Nielson (1980)

und Renka (1988) vorgestellten Versionen des Inverse Distance Weighting in höheren Dimensionen sehr viele Beobachtungspunkte benötigen (um die Koeffizienten der quadratischen Polynome schätzen zu können). Wohingegen die hier vorgestellte Kerninterpolation ab $d + 1$ Beobachtungspunkten funktioniert.

5.4 Unsicherheitsvorhersage bei der Kerninterpolation

Ein entscheidender Vorteil des Kriging ist, dass eine lokale Unsicherheitsaussage aufbauend auf dem stochastischen Modell möglich ist (4.11). Bei vielen sequentiellen Verfahren wie z.B. Jones et al. (1998) oder Lehman et al. (2004) wird diese Unsicherheitsaussage benötigt, um z.B. festzulegen, wo weitere Beobachtungspunkte erhoben werden. Eine ähnliche Unsicherheitsaussage kann für die Kerninterpolation bestimmt werden. Hier wird bewusst nicht von Konfidenzintervallen gesprochen, da dies vermitteln würde, dass die Unsicherheitsaussage bei der Kerninterpolation auf einem stochastischen Modell beruht. Definiere die Unsicherheitsvorhersage als

$$U(\hat{y})(\vec{x}) := \begin{cases} 0, & \vec{x} = \vec{x}_i, i = 1, \dots, n; \\ \sqrt{\frac{\sum_{j=1}^N g_j(\vec{x}) \sigma_j^2(\vec{x})}{\sum_{j=1}^N g_j(\vec{x})}}, & \text{sonst} \end{cases} \quad (5.22)$$

mit $\sigma_j^2(\vec{x}) := (\hat{y}(\vec{x}) - y^j(\vec{x}))^2$. Die Motivation ist also, die Einigkeit der lokalen Anpassungen als Referenz für die Unsicherheit zu nutzen. Streuen die Vorhersagen $y^j(\vec{x})$ in dem Punkt \vec{x} sehr stark wird sich eine höhere Unsicherheitsvorhersage ergeben, als wenn sich die Vorhersagen nur gering von einander unterscheiden. Ein prinzipieller Unterschied dieser Unsicherheitsvorhersage zu der Unsicherheitsvorhersage beim Kriging ist, dass Funktionen, die lokal ein sehr unterschiedliches Verhalten aufweisen, auch lokal unterschiedliche Unsicherheitsvorhersagen haben. Dies ist beim Kriging (aufgrund der Annahme der Stationarität) nicht gegeben. Desweiteren konvergieren die Unsicherheitsvorhersagen beim Kriging gegen einen einzelnen Wert, einem Vielfachen der geschätzten Standardabweichung des angenommenen Prozesses, wenn sich der Vorhersagepunkt \vec{x} immer weiter von den Beobachtungen entfernt. Für die Kerninterpolation

divergiert die Unsicherheitsvorhersage, falls sich der Vorhersagepunkt \vec{x} immer weiter von der Beobachtungspunkten entfernt. Wird nicht ein stationärer stochastischer Prozess angenommen, ist dies sicher ein intuitiv sinnvollerer Verhalten.

Zur Verdeutlichung der Unterschiede wird folgendes (eindimensionales) Beispiel auf dem Intervall $[-1, 1]$ betrachtet: $f(x) = 0.05 \sin(3\pi x) + F(x)$. Dabei ist $F(x)$ die Verteilungsfunktion einer normalverteilten Zufallsvariable mit den Parametern $(\mu, \sigma) = (0, 0.05)$. Es werden Beobachtungen an den Stellen $x_1 = -0.5, x_2 = -0.3, x_3 = -0.1, x_4 = 0.1, x_5 = 0.3, x_6 = 0.5$ genommen mit zugehörigen y -Werten y_1, \dots, y_6 . An diese wird ein Krigingmodell mit konstantem Regressionsteil und der Gaußkorrelationsfunktion angepasst (mittels des R-packages `mlegp`, Dancik (2010)). Als Alternative wird ein Kerninterpolationsmodell angepasst, mit der Gewichtsfunktion (5.5). Für beide wird die Unsicherheitsvorhersage bestimmt. Die Ergebnisse sind in Abbildung 5.1 dargestellt.

Für das Kriging ist in der Graphik der Vorhersagewert entsprechend Gleichung (4.10) \pm eine geschätzte Standardabweichung abgetragen. Für die Kerninterpolation ist der Vorhersagewert entsprechend Gleichung (5.1) $\pm U(x)$ abgetragen. Es ist deutlich zu erkennen, dass die Standardabweichung der Krigingvorhersagefunktion außerhalb der Beobachtungen gegen einen konstant Wert konvergiert. Im Gegensatz dazu divergiert die Unsicherheitsvorhersage der Kerninterpolation außerhalb der Beobachtungen.

5.5 Simulationen und Anwendungen

Im Folgenden sollen die Eigenschaften der Kerninterpolation im Vergleich zu verschiedenen anderen Interpolationsverfahren erkundet werden. Dafür werden verschiedene analytische Funktionen betrachtet, für die anhand der ausgewählten Methoden eine Vorhersage gemacht werden soll. Weiterhin stehen einige Anwendungsdatensätze zur Verfügung, anhand derer die Methoden verglichen werden. Zur Beurteilung der Anpassungsgüte wird der mittlere quadratische Fehler (root mean square error, RMSE)

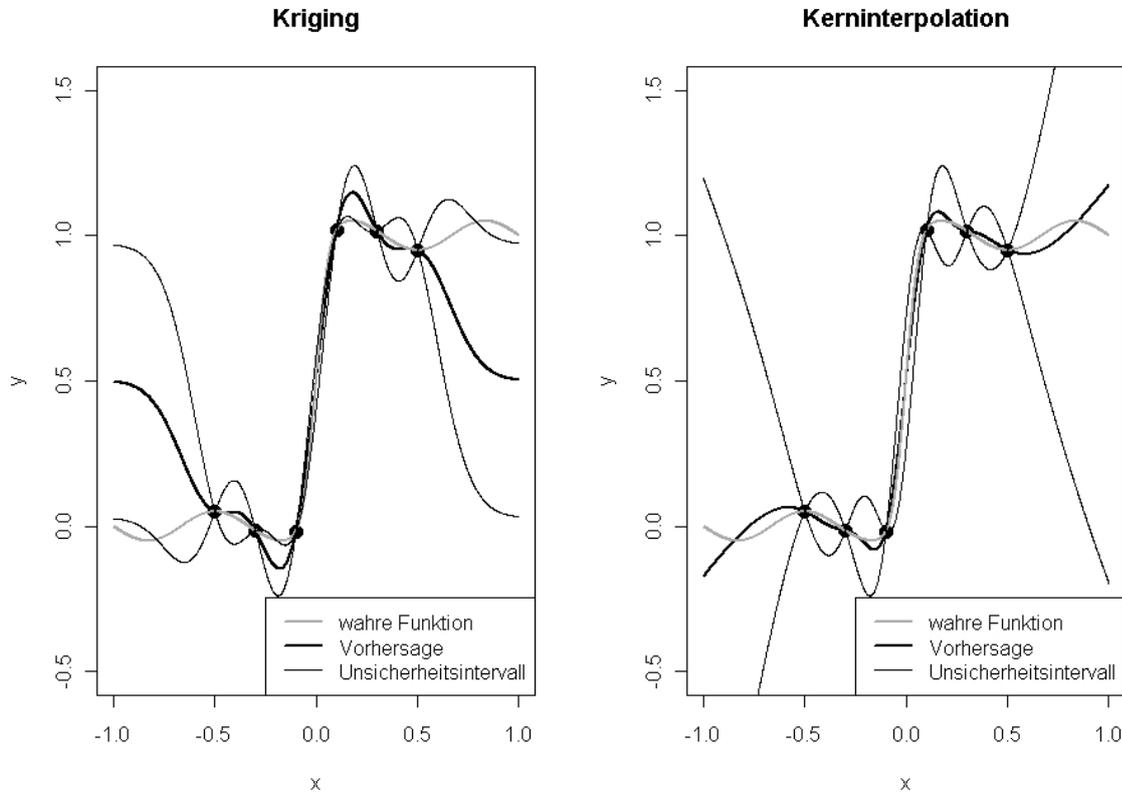


Abbildung 5.1: Vergleich der Unsicherheitsvorhersagen für das Kriging und die Kerninterpolation.

betrachtet:

Definition 5.2 (RMSE). *Der mittlere quadratische Fehler einer Vorhersagefunktion $\hat{y}(\vec{x})$ für die wahre Funktion $y(\vec{x})$ an Punkten $\vec{x}_1, \dots, \vec{x}_M$ ist gegeben durch*

$$RMSE(\hat{y}(x), y(x)) := \sqrt{\frac{1}{M} \sum_{i=1}^M (y(x_i) - \hat{y}(x_i))^2}. \quad (5.23)$$

Sämtliche Simulationen wurden in R-2.10.1 (R Development Core Team (2009)) durchgeführt.

Für das Kriging wurde jeweils ein Modell mit einer konstanten und linearen Trendfunktion angepasst. Als Korrelationsfunktion wurde in beiden Fällen die Gaußkorrelations-

funktion entsprechend Tabelle 4.1 und als Schätzmethode das Maximum Likelihood Verfahren verwandt. Diese Korrelationsfunktion wurde ausgewählt, da sie in vielen Arbeiten als Standardkorrelationsfunktion gewählt wird. Die Berechnungen wurden mit der R-Funktion `mlegp` des Packages `mlegp` (Dancik (2010)) durchgeführt.

Für die Interpolation mit radialen Basisfunktionen wurde die Gaußbasis und die multiquadratische Basisfunktion entsprechend Tabelle 4.2, jeweils mit einem konstanten Trend und einem linearen Trend verwandt. Als Parameter wurden für die Gaußbasis $\alpha = 1$ und für die multiquadratische Basisfunktion $c^2 = 2(\frac{1}{n})^d$ verwandt, wobei n die Stichprobengröße ist und d die Dimension des Versuchsraums. Desweiteren wurde, falls die Stichprobengröße es zulässt, der Thin Plate Spline mit Basisfunktion entsprechend Formel (4.18) verwandt. Um in möglichst vielen Fällen den Thin Plate Spline anwenden zu können, wird der Wert m immer minimal gewählt (also $m = \lceil 0.5d \rceil$, wobei $\lceil d \rceil$ die kleinste ganze Zahl ist, die größer oder gleich d ist).

Für das Inverse Distance Weighting wurde die ursprünglich von Shepard (1968) vorgeschlagene Variante entsprechend Formel (4.19) mit $u = 2$ verwandt. Desweiteren wurde die Abänderung von Renka (1988) mit einem quadratischen Trend und einem linearen Trend verwandt. Die Konstante N_w wurde dabei minimal gewählt, so dass die quadratische oder lineare Funktion gerade noch angepasst werden können. Für den Parameter N_q wurde $N_q = \lceil n/3 \rceil$, wobei $\lceil \cdot \rceil$ der auf eine ganze Zahl gerundete Wert ist, gewählt.

Für die Kerninterpolation wurde die Gewichtsfunktion entsprechend Gleichung (5.5) verwandt. Die verwandten Methoden sind in Tabelle 5.1 zusammengefasst.

5.5.1 Analytische Funktionen

Um die Eigenschaften der verschiedenen Methoden anhand analytischer Funktionen zu untersuchen werden verschiedene analytische Funktionen verwandt. Der Versuchsraum ist immer auf $[-1, 1]^d$ normiert. Es wurden drei unterschiedliche Arten Versuchspläne

Bezeichnung	Abkürzung
Kriging mit konstantem Trend	<i>Kr0</i>
Kriging mit linearem Trend	<i>Kr1</i>
RBF mit Gaußbasis, konstanter Trend	<i>G0</i>
RBF mit Gaußbasis, linearer Trend	<i>G1</i>
RBF mit multiquadratischer Basis, konstanter Trend	<i>M0</i>
RBF mit multiquadratischer Basis, linearer Trend	<i>M1</i>
Thin Plate Spline	<i>TPS</i>
IDW entsprechend Shepard	<i>S</i>
IDW entsprechend Renka und Nielson, linearer Trend	<i>R1</i>
IDW entsprechend Renka und Nielson, quadratischer Trend	<i>R2</i>
Kerninterpolation mit Gewichtsfunktion (5.5)	<i>KI</i>

Tabelle 5.1: Ausgewählte Interpolationsmethoden.

zur Erzeugung der Beobachtungen ausgewählt: Latin Hypercubes, Latin Hypercubes mit Clustern und über den Versuchsraum gleichverteilte Zufallszahlen. Latin Hypercubes wurden ausgewählt, da sie sich mittlerweile als eine Art Standard für die Planung von Computereperimenten herauskristallisiert haben. Die durch Cluster erweiterten Latin Hypercubes sind durch viele sequentielle Methoden für Computereperimente motiviert. Bei diesen sequentiellen Methoden wird meistens ein erster Versuchsplan erhoben und im Folgenden werden an „vielversprechenden“ Stellen im Versuchsraum nochmals Läufe plziert. Dies resultiert üblicherweise in Clustern im Versuchsraum. Gleichverteilte Zufallszahlen über den Versuchsraum wurden als naive Alternative für raumfüllende Versuchspläne betrachtet, bei denen erwartet wird, dass der mittlere RMSE höher ist als bei den zufälligen Latin Hypercubes.

Als Stichprobengrößen n wurden für die Latin Hypercubes und die gleichverteilten Zufallszahlen $n = d + 2i, i = 1, \dots, 6$ betrachtet. Für die Latin Hypercubes mit Clustern wurden zusätzlich um zwei zufällige Versuchspunkte jeweils weitere

$\lfloor n/4 \rfloor$ Punkte erzeugt. Diese zusätzlichen Punkte stammen aus einer multivariaten Normalverteilung mit Mittelwert gleich dem Versuchspunkt und Kovarianzmatrix $\sigma^2 I_d, \sigma^2 = 0.05$, so dass an zwei Stellen im Versuchsraum Cluster entstehen. Die gewählten Stichprobengrößen sollen Situationen darstellen, in denen extrem wenige Beobachtungen vorhanden sind. Für jede Stichprobengröße wurden dann von jeder Art Versuchsplan 50 zufällige ausgewählt, für die daraus resultierenden Versuchsläufe die Zielvariable berechnet und an diese Daten dann die Modelle aus Tabelle 5.1 angepasst. Die Güte der Anpassung wurde anhand des RMSE beurteilt, der aufbauend auf einem zufälligen Latin Hypercube der Größe 2000 berechnet wurde. Um die Vorhersagegüte beurteilen zu können, wurde zusätzlich für jede Funktion $f(\vec{x})$ die Variation $V := \sqrt{\int (f(\vec{x}) - \int f(\vec{z})d\vec{z})^2 d\vec{x}}$ numerisch durch eine Monte Carlo Schätzung bestimmt. Ist der RMSE einer Vorhersagemethode nicht wesentlich kleiner als die Variation V einer Funktion f , so ist die Vorhersagemethode nicht sinnvoll, was insbesondere bei sehr kleinen Stichprobenumfängen passieren kann. Da dieses Vorgehen in einer recht hohen Rechenzeit resultiert, wurden die Berechnungen auf dem LIDong Cluster der TU Dortmund durchgeführt.

Die hier genutzten Funktionen wurden so ausgewählt, dass sie nach Möglichkeit durch die Anwendung motiviert sind und schon in anderen Artikeln verwandt wurden. In 2 Dimensionen gibt es dafür eine Fülle an Funktionen, in höheren Dimensionen dagegen deutlich weniger Beispiele. Die Ergebnisse der Simulationen für eine Funktion bei einer bestimmten Art Versuchsplan sind in Boxplots zusammengefasst, wobei zu beachten ist, dass für die y -Skala eine logarithmische Einteilung gewählt wurde, so dass auch bei kleinen RMSE Werten eine gute Unterscheidbarkeit möglich ist. Ist für eine Kombinationen von Stichprobengröße und Dimension eine Interpolationsart nicht durchführbar (aufgrund einer zu niedrigen Stichprobengröße), wird der entsprechende Boxplot weggelassen. Dies tritt insbesondere bei dem Inverse Distance Weighting entsprechend Renka und Nielson (linearer und quadratischer Trend) und dem Thin Plate Spline auf. In wenigen Fällen wurden extreme Ausreißer der RMSE-Werte abgeschnitten, um eine bessere Vergleichbarkeit zu gewährleisten.

Diese Ausreißer sind entweder beim Kriging oder beim *IDW* nach Renka mit quadratischem Trend aufgetreten. Die Variation V wurde durch eine horizontale Linie mit in die Boxplots eingezeichnet.

Braninfunktion

Das erste Beispiel ist die Braninfunktion, die z.B. in Lehman et al. (2004) verwandt wird, allerdings auch in vielen anderen Arbeiten Anwendung findet. Die Braninfunktion hat den zweidimensionalen Versuchsraum $[-5, 10] \times [0, 15]$ und die Funktionsvorschrift

$$f(\vec{x}) = \left(x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos(x_1) + 10. \quad (5.24)$$

Die Ergebnisse sind in den Abbildungen 5.2, 5.3 und 5.4 zusammengefasst. Für kleine Stichproben sind dabei die Mindestanforderungen an die Stichprobengröße für die Inverse Distance Weighting Varianten *R1* und *R2* nicht erfüllt, weswegen die entsprechenden Boxplots weggelassen werden. Für die Latin Hypercubes sind insgesamt keine starken Unterschiede zwischen den gewählten Interpolationsmethoden zu erkennen. Bei den kleinen Stichprobengrößen schneidet Kriging etwas schlechter ab als die restlichen Methoden. Für die größeren Stichprobengrößen sind die verschiedenen Varianten der Radial Basis Funktionen Interpolation am besten bzgl. des RMSE. Die Kerninterpolation liefert insgesamt gute Ergebnisse. Für die Latin Hypercubes mit Clustern fällt auf, dass Kriging in beiden Varianten für kleine Stichprobengrößen schlechte RMSE-Werte liefert, für größere Stichproben allerdings mit Abstand die besten Ergebnisse liefert. Hier schneidet die Kerninterpolation für kleine n gut ab, für größere ist nur Kriging eindeutig besser. Die verschiedenen Varianten des *IDW* schneiden eher schlechter ab. Für die gleichverteilten Versuchspläne erscheinen insgesamt größere Unterschiede zwischen den Methoden. Interessant in Bezug auf Kriging mit konstantem Mittel ist, dass sich hier das Verhalten im Vergleich zu den Latin Hypercubes mit Clustern umgekehrt hat: Für kleine n sind hier gute RMSE-Werte zu beobachten, für größere n hebt sich dieser Effekt allerdings auf. In Bezug auf die Variation V lässt sich sagen, dass insbesondere für die kleinste Stichprobengröße die RMSE-Werte teilweise deutlich über dem Wert von V liegen, dies jedoch für die Latin Hypercubes und die Latin Hypercubes mit Clustern schon bei der nächst größeren betrachteten Stichprobengröße nicht mehr so stark auftritt. Bei den gleichverteilten Versuchsplänen hingegen ist es häufiger zu

beobachten, dass der RMSE der Vorhersagen deutlich über V liegt, unabhängig von der gewählten Interpolationsmethode. Dies bestätigt die eingangs gemachte Vermutung, dass die Klasse der Latin Hypercubes deutlich attraktiver für die Planung von Computereperimenten ist als einfaches Ziehen aus einer Gleichverteilung über den Versuchsraum.

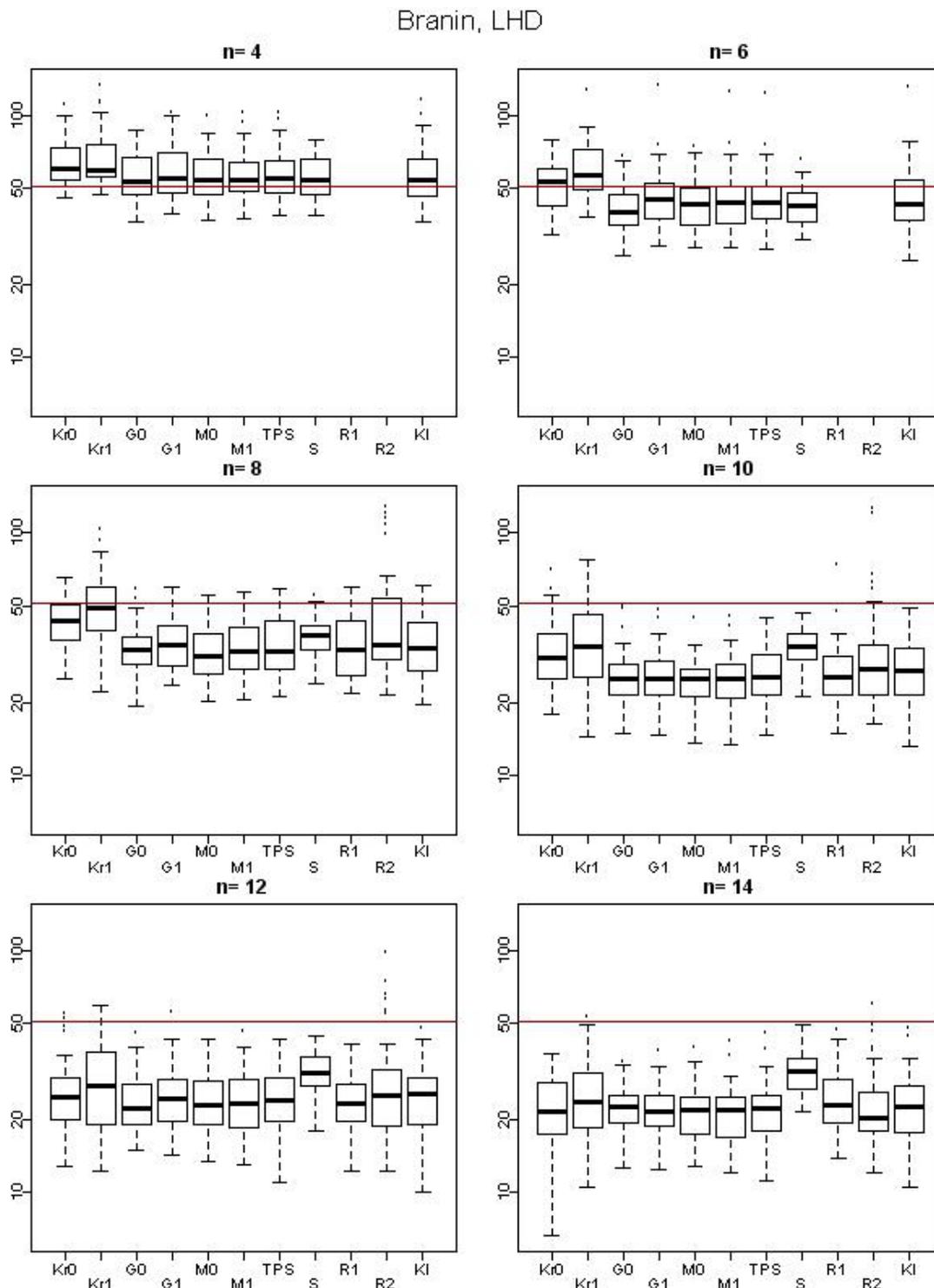


Abbildung 5.2: Boxplots der RMSE-Werte für die Braninfunktion basierend auf Latin Hypercubes.

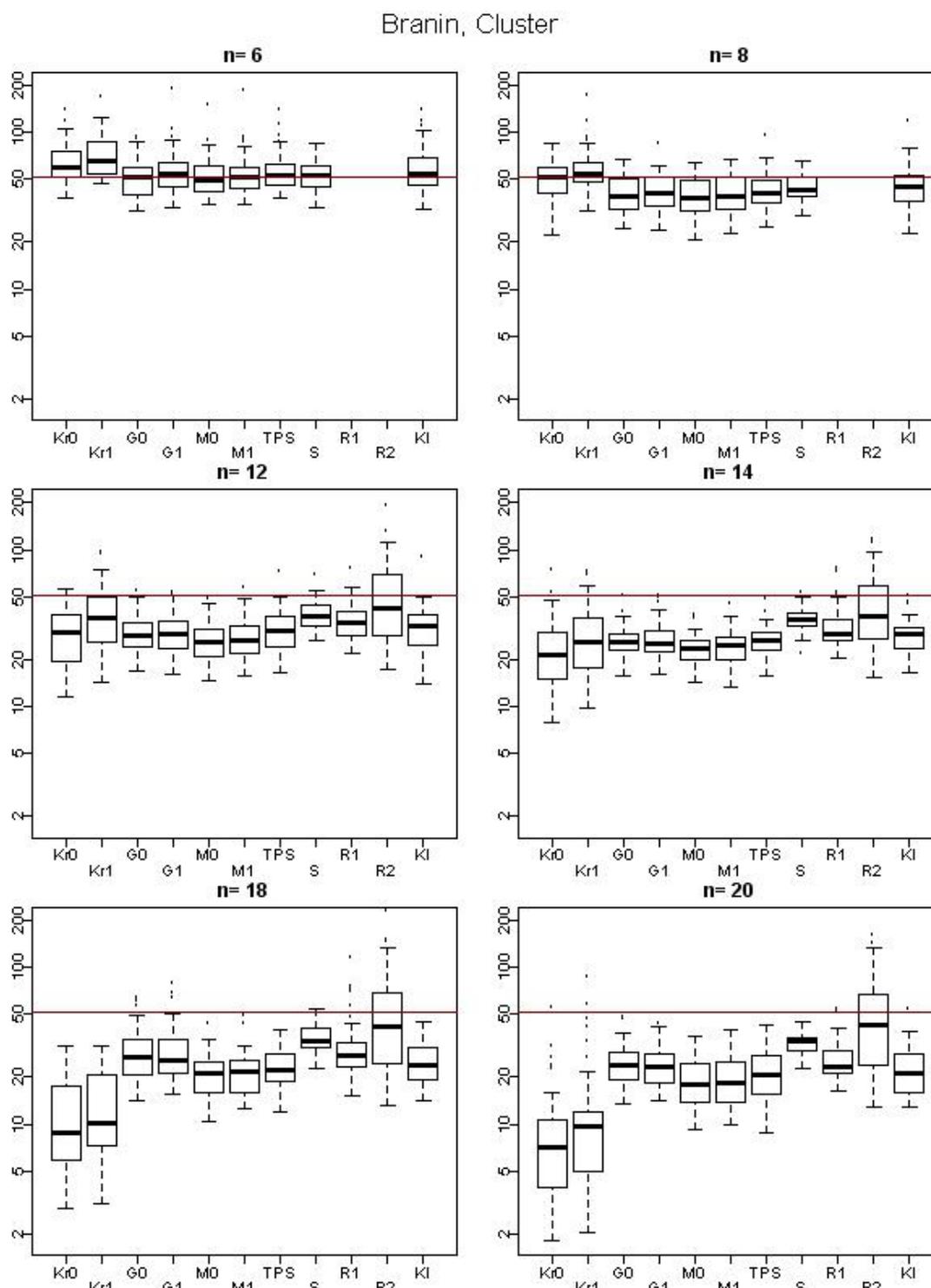


Abbildung 5.3: Boxplots der RMSE-Werte für die Braninfunktion basierend auf Latin Hypercubes mit Clustern.

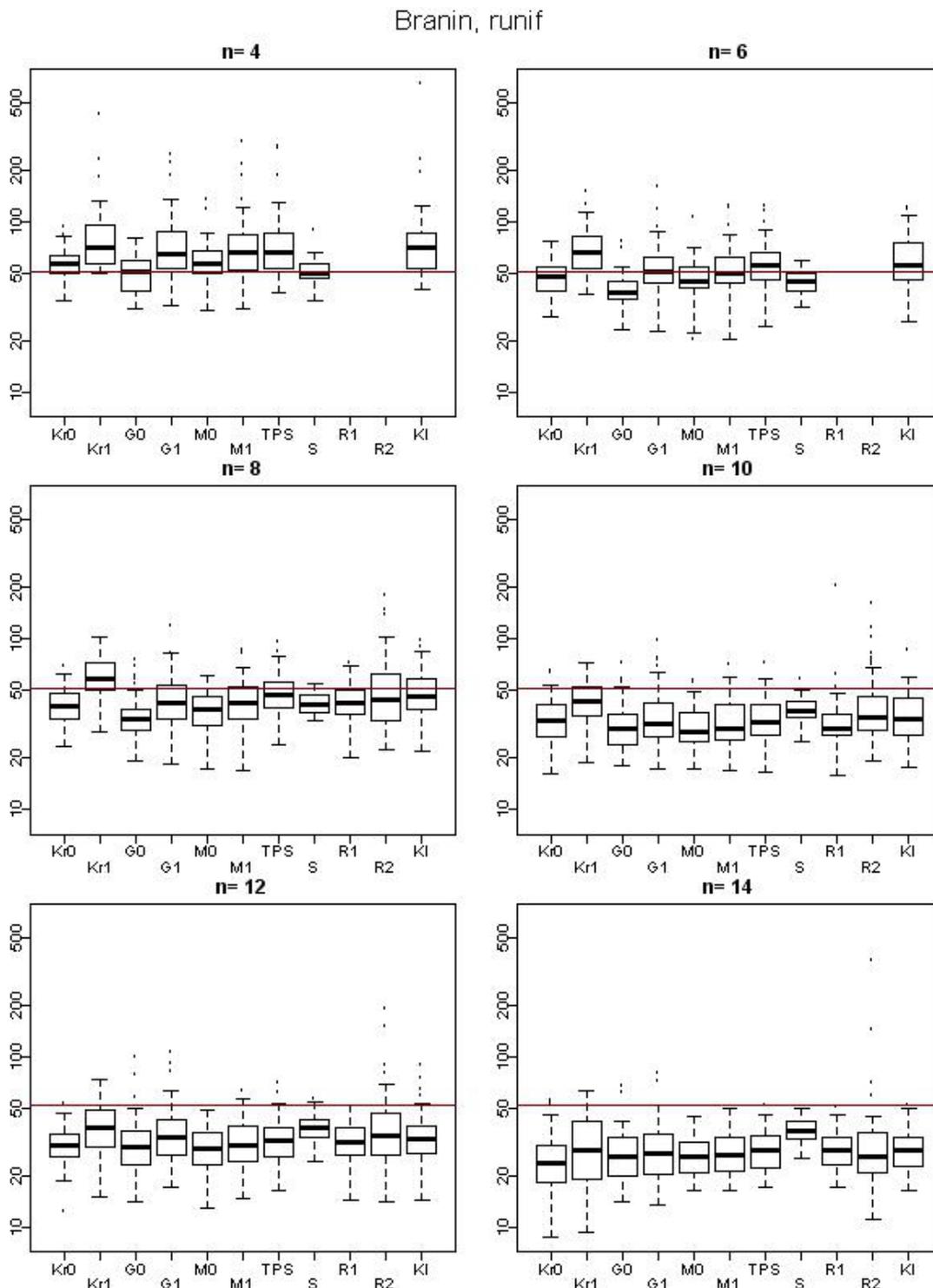


Abbildung 5.4: Boxplots der RMSE-Werte für die Braninfunktion basierend auf gleichverteilten Versuchsplänen.

Bates et al. (2003)

Das zweite Beispiel wurde in Bates et al. (2003) zur Demonstration von polynomialen Interpolationsmodellen verwandt:

$$f(\vec{x}) = 1 + 2 \exp(x_1) + \sin(\pi x_3)^2, \quad (5.25)$$

mit dreidimensionalem Versuchsraum $[0, 1]^3$. Aktiv sind hier nur 2 der 3 Einflussgrößen. Die einzige Methode, die für eine solche Situation eine Gewichtung der Einflussgrößen vornimmt ist das Kriging, weswegen zu erwarten ist, dass bei diesem Beispiel Kriging die besten Ergebnisse liefert. Dies wird bestätigt durch die Boxplots in den Abbildungen 5.5, 5.6 und 5.7. Insgesamt ist für dieses Beispiel Kriging (in beiden Varianten) eindeutig am besten, was damit erklärt werden kann, dass beim Krigingansatz implizit eine Art Screening mit durchgeführt wird. Dieser Vorteil wird insbesondere bei den größeren Stichprobenumfängen stark deutlich. Die restlichen Methoden verhalten sich allesamt ähnlich zu einander. Die Kerninterpolation bewegt sich auf einem leicht schlechteren Level als die radialen Basisfunktionen, von denen hier die multiquadratische radiale Basisfunktion für alle drei Arten von Versuchsplänen die besten Ergebnisse liefert. Die RMSE-Werte des *IDW* sind in allen drei Varianten nicht konkurrenzfähig. Bei allen Stichprobenarten liegen die RMSE-Werte sämtlicher Vorhersagemethoden für die kleinste Stichprobe um oder deutlich über der Variation V . Interessant zu beobachten ist, dass für das *IDW* nach Renka mit quadratischer Anpassung die Vorhersage so schlecht ist, dass sogar für größere Stichprobenumfänge der RMSE oftmals höher ist als die Variation V .

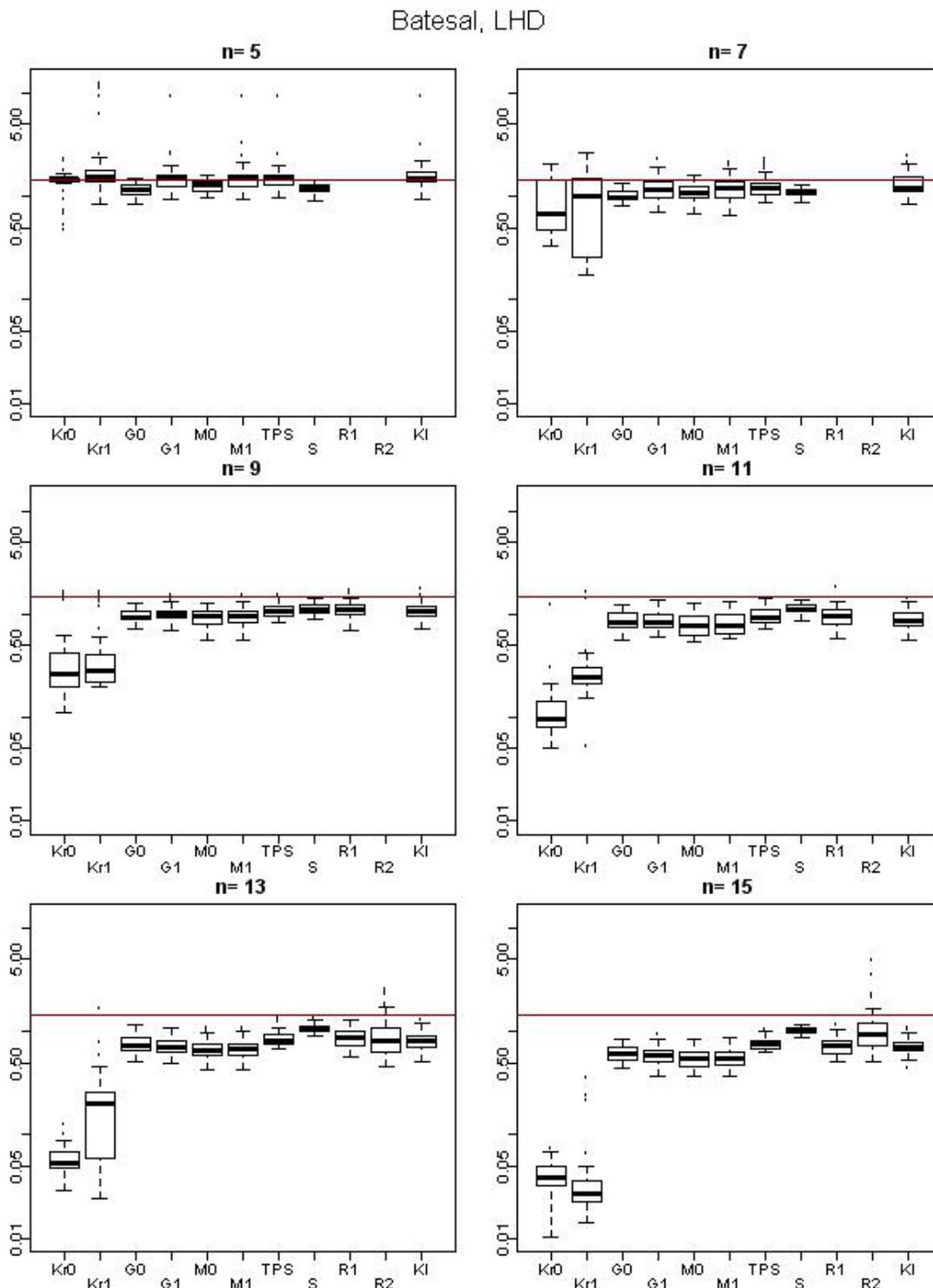


Abbildung 5.5: Boxplots der RMSE-Werte für die Testfunktion von Bates et al. (2003) basierend auf Latin Hypercubes.

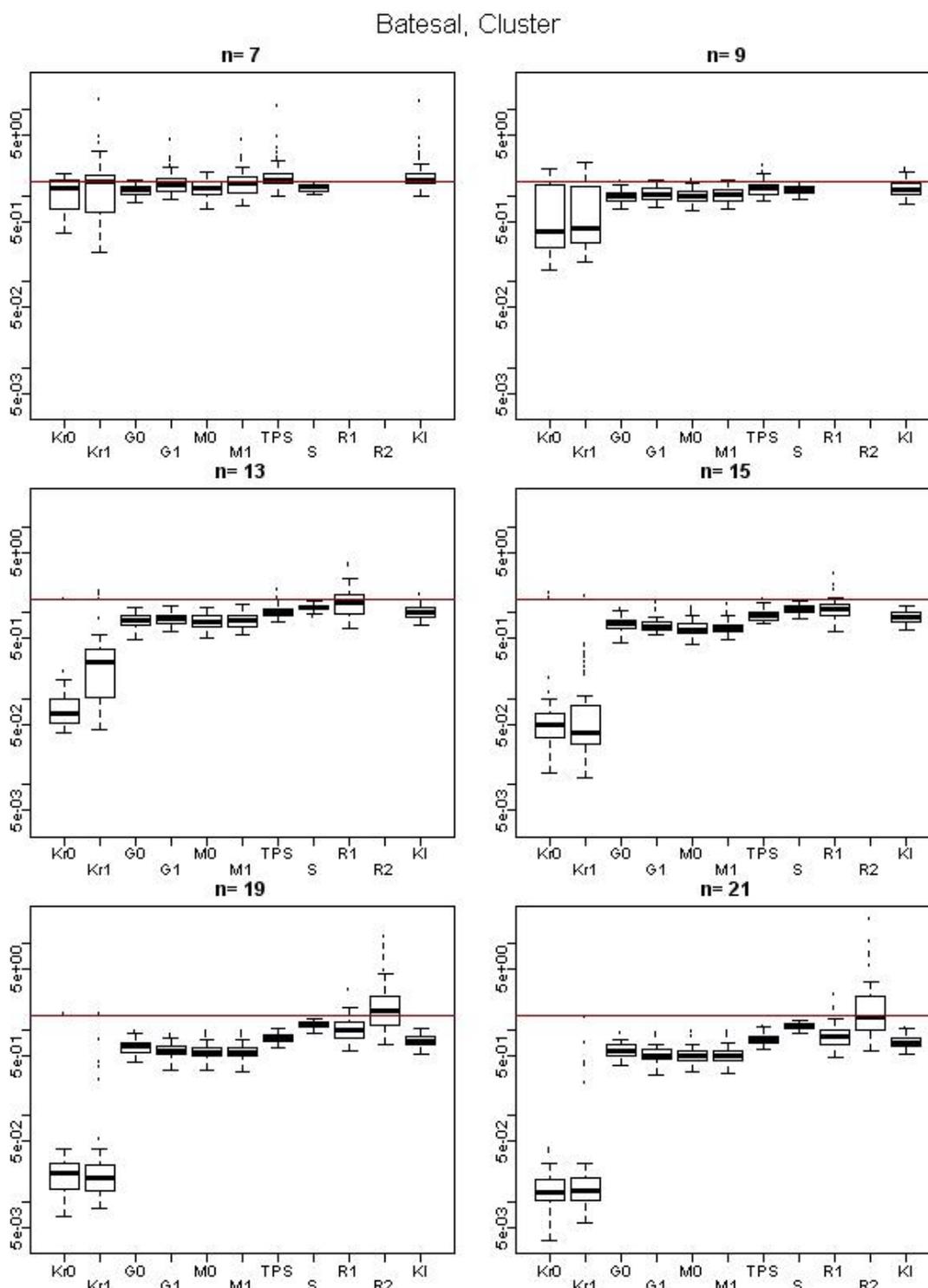


Abbildung 5.6: Boxplots der RMSE-Werte für die Testfunktion von Bates et al. (2003) basierend auf Latin Hypercubes mit Clustern.

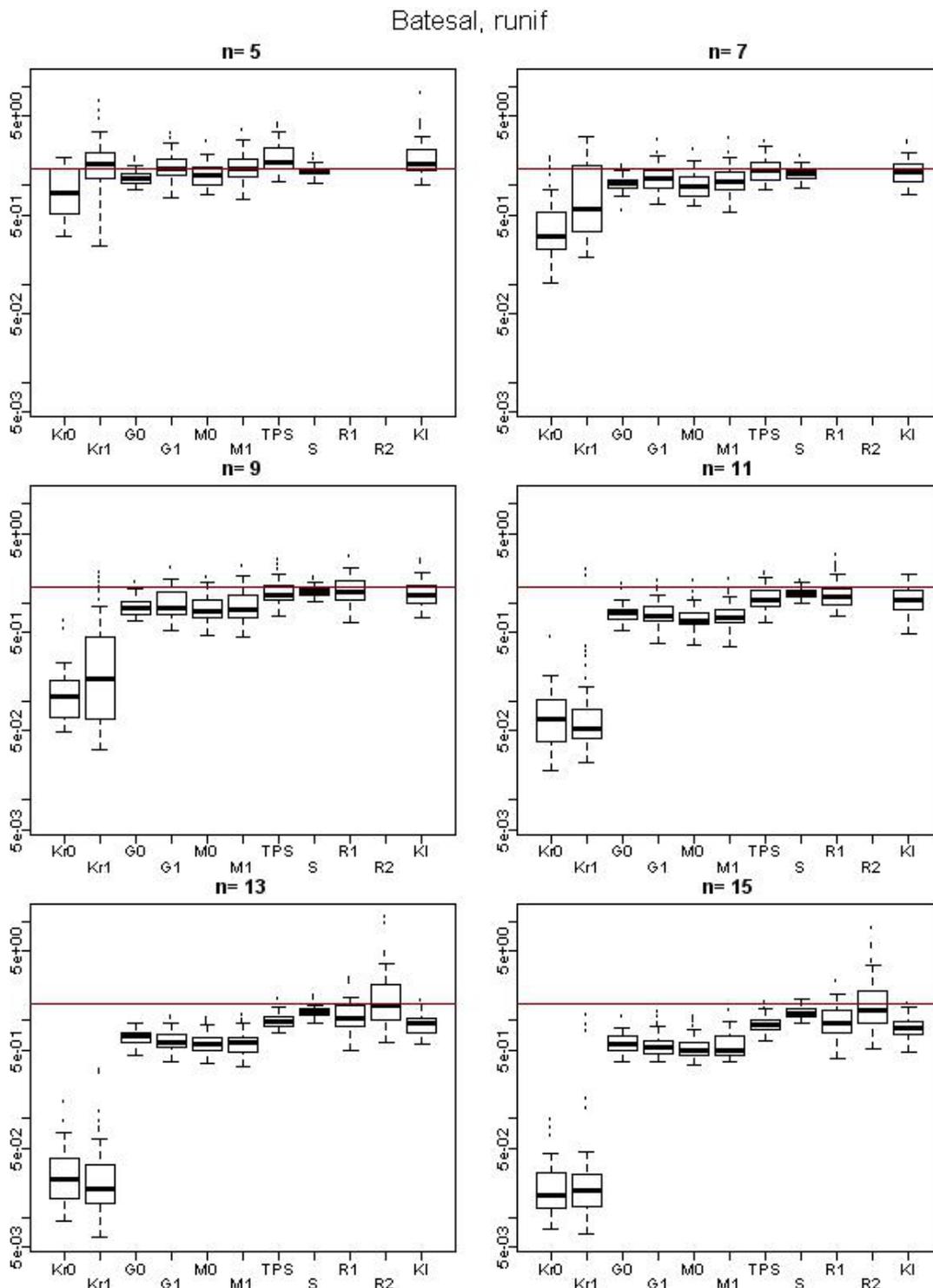


Abbildung 5.7: Boxplots der RMSE-Werte für die Testfunktion von Bates et al. (2003) basierend auf gleichverteilten Versuchsplänen.

Plastic Seal Beispiel

Das nächste Beispiel stammt aus Allen et al. (2003) und ist eine dreidimensionale Funktion, die sich aus einem echten Design Problem über Finite Elemente Simulationen herleitet. Die Zielvariable ist die Kraft, die benötigt wird, um eine Art Stempel in eine Fuge zu drücken in Abhängigkeit von Parametern, die geometrische Eigenschaften der Fuge beschreiben. Die Funktion ist gegeben durch

$$f(\vec{x}) = (105(0.58(x_2 + x_3 - 0.85) + 3)^3 x_3) \left(\frac{\sin(1.5x_3/(x_1 - 2))}{(x_1 - 2)^3} \right), \quad (5.26)$$

mit dem Versuchsraum $[4, 7] \times [0.7, 1.7] \times [0.055, 0.5]$. In Allen et al. (2003) werden RSM Methoden und Kriging als Metamodelle verwandt. Für das Kriging wurde dabei die Gaußsche Korrelationsfunktion mit konstantem Mittel verwandt. Als Alternative wurde ein quadratisches RSM Modell verwandt. Dabei ergab sich, dass bei kleinen Stichproben Kriging und RSM Methoden vergleichbar sind, bei größeren jedoch Kriging Vorteile hat.

Die Ergebnisse der hier vorgenommenen Simulationen sind in den Abbildungen 5.8 (Latin Hypercube), 5.9 (Latin Hypercube mit Clustern) und 5.10 (Gleichverteilt) dargestellt. Wieder fällt auf, dass für die jeweils kleinste Stichprobe die RMSE-Werte sämtlicher Vorhersagemethoden um oder über der Variation V liegen, für größere Stichproben jedoch RMSE-Werte erreicht werden, die deutlich unter V liegen. Dies legt den Schluss nahe, dass der Stichprobenumfang $n = d + 2$ deutlich zu klein ist für eine sinnvolle Modellierung von Computerexperimenten. Für die Latin Hypercubes liefern Methoden mit konstantem Mittel beim kleinsten Stichprobenumfang die besten Ergebnisse. Bei mittlerem n sind insgesamt alle Methoden ähnlich bzgl. des RMSE, für größere n erzielt Kriging die besten Ergebnisse, auch wenn die Streuung der Ergebnisse recht groß ist. Für die Latin Hypercubes mit Clustern und den gleichverteilten Versuchsplänen stellt sich die Situation ähnlich dar, bei größerem n ist Kriging am Besten, hat jedoch auch eine relativ große Streuung in den unterschiedlichen Vorhersagen, die Kerninterpolation liefert ähnlich gute Ergebnisse wie die multiquadratischen radialen Basisfunktionen.

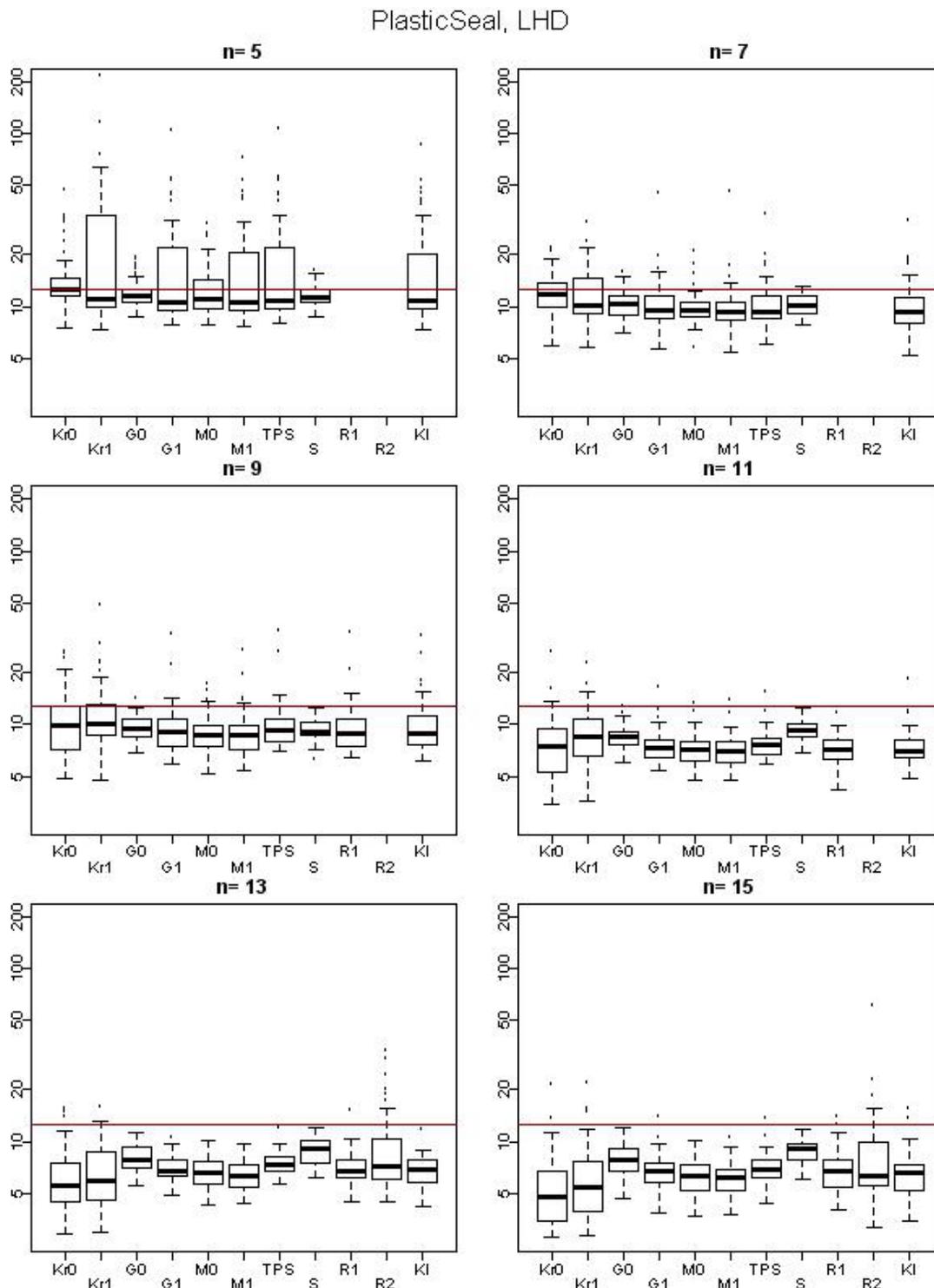


Abbildung 5.8: Boxplots der RMSE-Werte für die Testfunktion von Allen et al. (2003) basierend auf Latin Hypercubes.

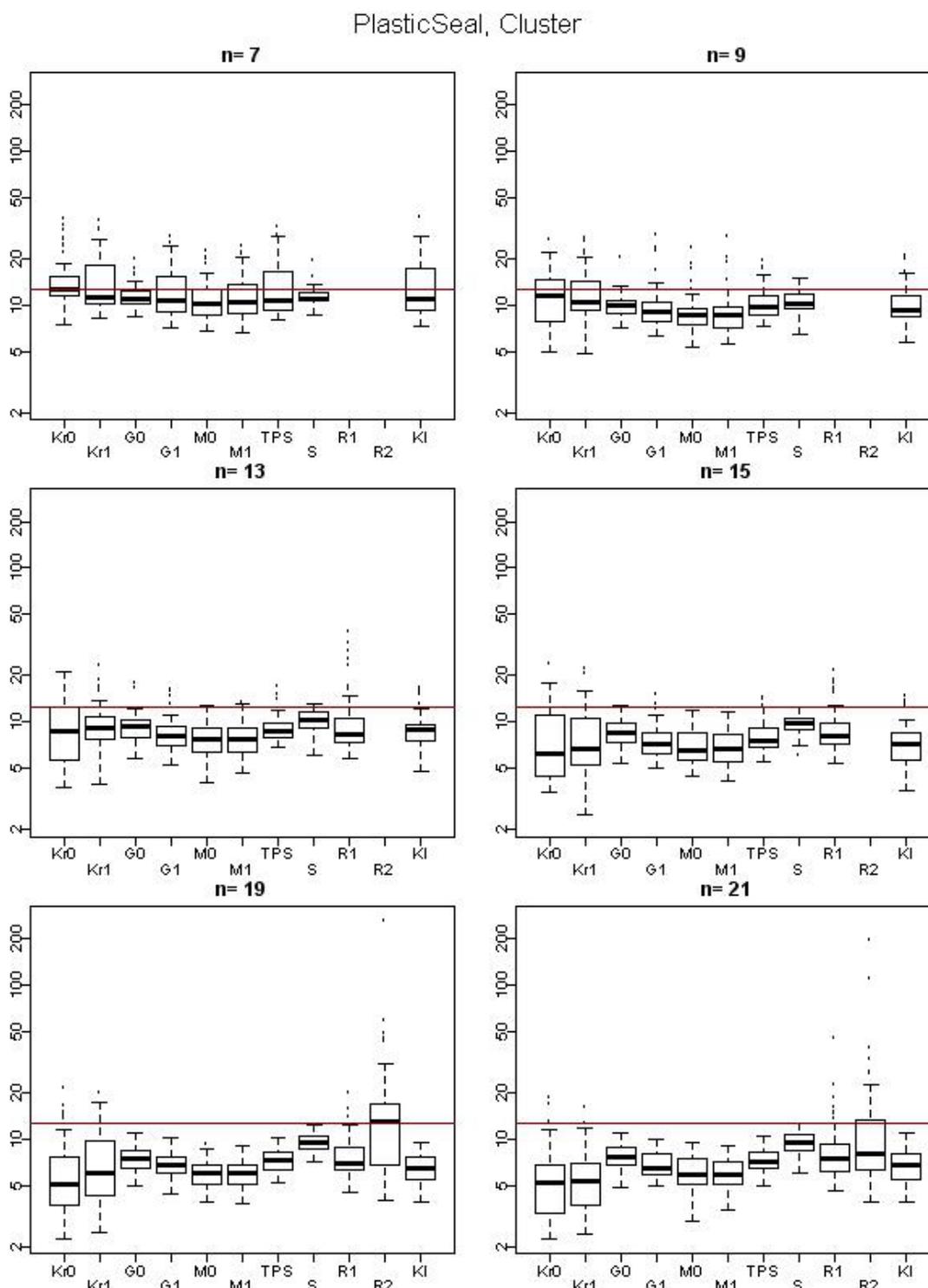


Abbildung 5.9: Boxplots der RMSE-Werte für die Testfunktion von Allen et al. (2003) basierend auf Latin Hypercubes mit Clustern.

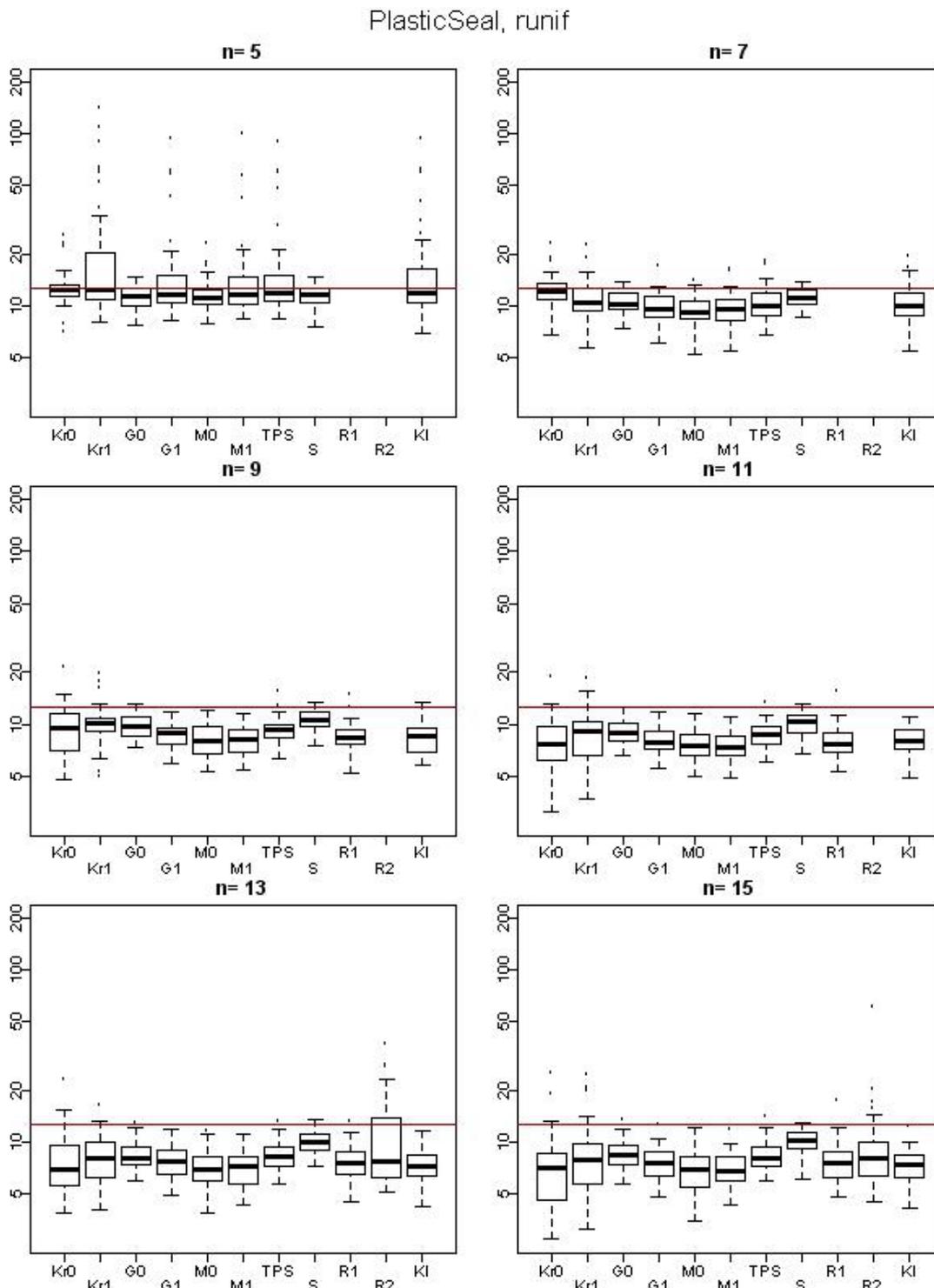


Abbildung 5.10: Boxplots der RMSE-Werte für die Testfunktion von Allen et al. (2003) basierend auf gleichverteilten Versuchsplänen.

Lastarm (Cantilever)

Die nächste Funktion wurde Goel et al. (2009) entnommen. Die Funktion hat einen 5-dimensionalen Versuchsraum und die Funktionvorschrift

$$f(\vec{x}) = 2.25 - \frac{4 * 100^3}{x_3 x_4 x_5} \sqrt{\left(\frac{x_2}{x_5}\right)^2 + \left(\frac{x_1}{x_4}\right)^2}. \quad (5.27)$$

Der Versuchsraum sieht wie folgt aus: $[700, 1300] \times [900, 1300] \times [20 * 10^6, 35 * 10^6] \times [2, 3] \times [3, 5]$. Die Funktion stellt die Abweichung eines Lastarms unter Belastung dar. Die Ergebnisse anhand von Boxplots sind in den Abbildungen 5.11, 5.12 und 5.13 zu finden. Für die Latin Hypercubes mit größerem n ist Kriging mit konstantem Mittel ($Kr0$) am besten. Kerninterpolation, gaußsche radiale Basisfunktion mit linearem Trend, die beiden Varianten der multiquadratischen radialen Basisfunktion und IDW nach Renka mit linearem Trend verhalten sich ähnlich, die gaußsche radial Basisfunktion mit konstantem Mittel und IDW nach Shepard liefern insgesamt schlechtere Ergebnisse. Ein ähnliches Bild ergibt sich für die Latin Hypercubes mit Clustern. Kriging mit konstantem Mittel schneidet insgesamt am besten ab, auch wenn es teilweise eine größere Streuung der RMSE-Werte hat. IDW liefert in aller drei Varianten schlechtere Ergebnisse und Kerninterpolation und gaußsche radiale Basisfunktion mit linearem Trend und multiquadratische radiale Basisfunktionen in beiden Varianten ähnlich gute Ergebnisse. TPS und die gaußsche radiale Basisfunktion mit konstantem Mittel liefern eher schlechte Ergebnisse. Für die gleichverteilten Versuchspläne lassen sich die Ergebnisse ähnlich zusammen fassen.

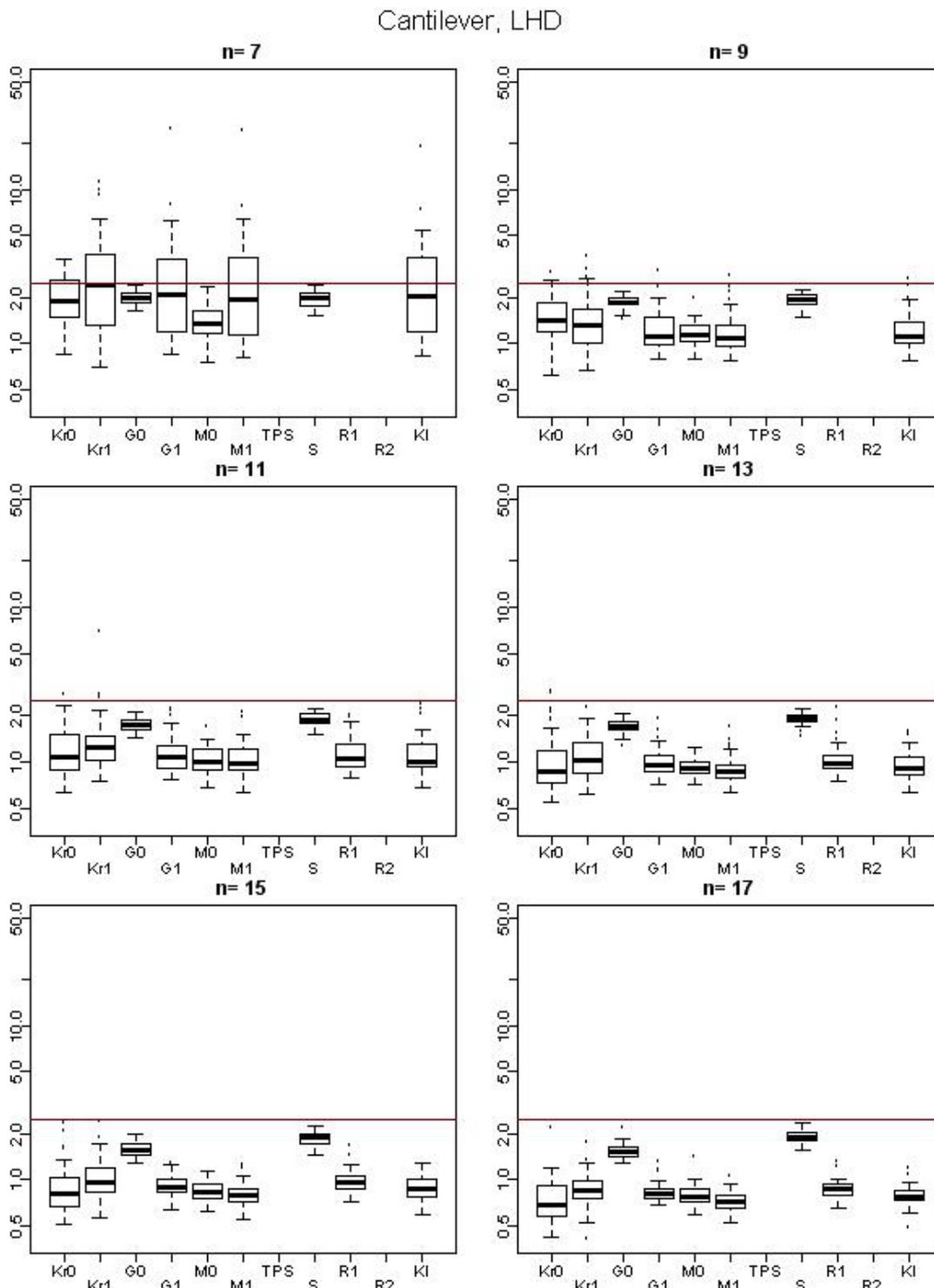


Abbildung 5.11: Boxplots der RMSE-Werte für die Lastarmfunktion von Goel et al. (2009) basierend auf Latin Hypercubes.

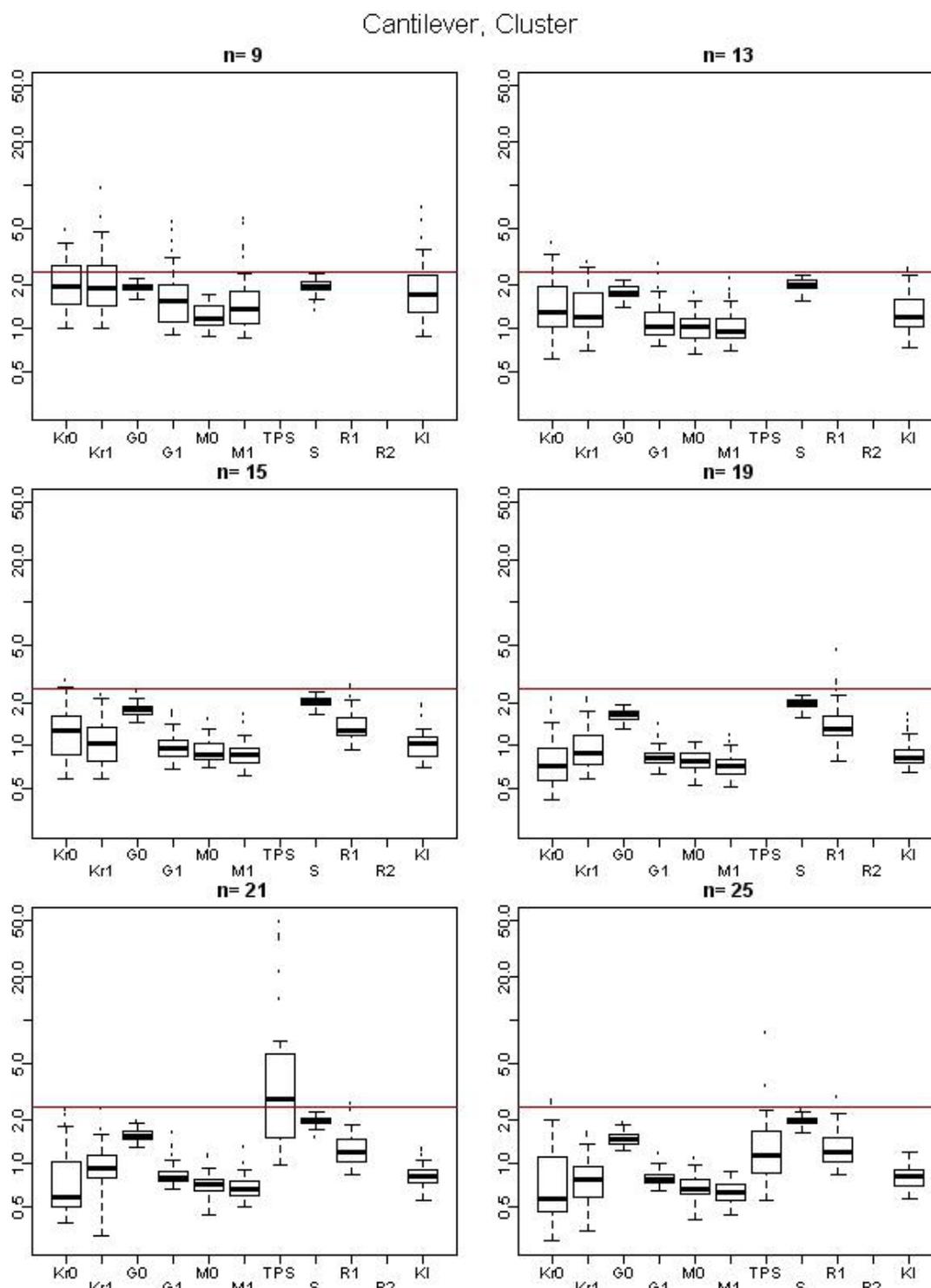


Abbildung 5.12: Boxplots der RMSE-Werte für die Lastarmfunktion von Goel et al. (2009) basierend auf Latin Hypercubes mit Clustern.

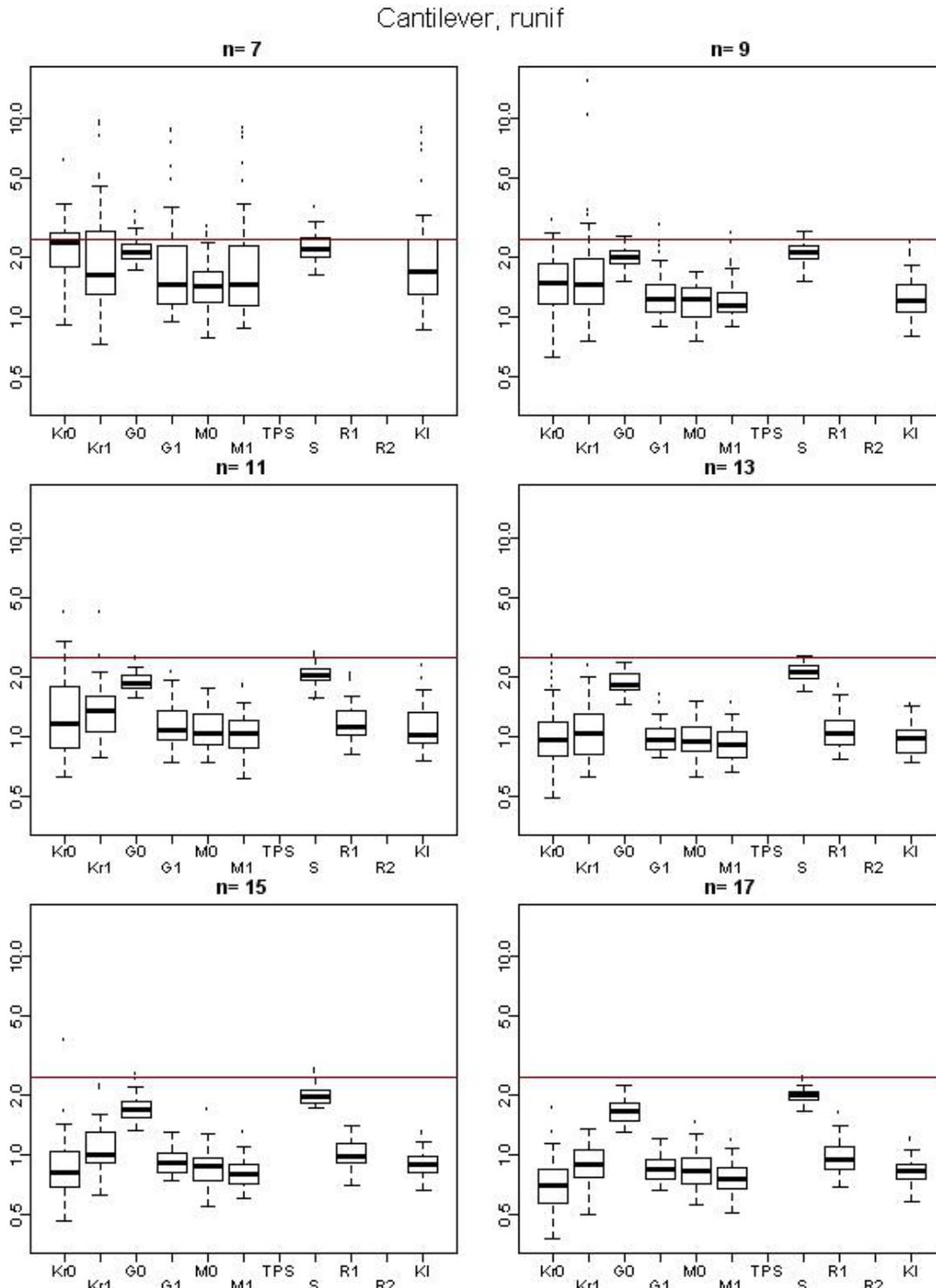


Abbildung 5.13: Boxplots der RMSE-Werte für die Lastarmfunktion von Goel et al. (2009) basierend auf gleichverteilten Versuchsplänen.

Bohrlochfunktion

Das letzte Beispiel wurde in Morris et al. (1993) und Morris und Mitchell (1995) betrachtet und ist ein physikalisches Modell für den stationären Durchfluss von Wasser eines Bohrlochs zwischen zwei Wasserschichten in Abhängigkeit von 8 Einflussgrößen. Die Funktionsvorschrift lautet

$$f(\vec{x}) = \frac{2\pi x_3(x_4 - x_6)}{\log(x_2/x_1) \left[1 + \frac{2x_7x_3}{\log(x_2/x_1)x_1^2x_8} + \frac{x_3}{x_5} \right]}. \quad (5.28)$$

Der Versuchsraum umfasst den Bereich $[0.05, 0, 15] \times [100, 50000] \times [63070, 115600] \times [990, 1110] \times [63.1, 116] \times [700, 820] \times [1120, 1680] \times [9855, 12045]$. Die Ergebnisse sind graphisch wieder durch Boxplots in den Abbildungen 5.14 (Latin Hypercubes), 5.15 (Latin Hypercube mit Clustern) und 5.16 (gleichverteilt) dargestellt. Für die Latin Hypercubes ist Kriging mit konstantem Mittel bei allen Stichprobenumfängen am besten. Auffällig ist hier, dass Kriging mit konstantem Mittel insgesamt eindeutig die besten RMSE-Werte liefert, aber auch gleichzeitig bei den größeren Stichproben sehr unterschiedliche Ergebnisse für verschiedene Versuchspläne aufweist. Die schlechtesten Ergebnisse liefern hier gaußsche radiale Basisfunktionen mit konstantem Mittel und *IDW* nach Shepard. Die RMSE-Werte von RBF Interpolation mit konstantem Trend und gaußscher radialer Basis und von *IDW* nach Shepard liegen selbst für größere Stichproben noch sehr nahe bei der Variation V . Für die gaußsche radiale Basisfunktion lässt dies vermuten, dass der Parameter α nicht optimal gewählt wurde. Kriging mit linearem Trend liefert für kleinere n nicht unbedingt zufriedenstellende Ergebnisse. Kerninterpolation, gaußsche radiale Basisfunktionen mit linearem Trend und multi-quadratische radiale Basisfunktionen und *IDW* nach Renka mit linearem Trend liefern insgesamt ähnliche Ergebnisse für Latin Hypercubes. Für Latin Hypercubes mit Clustern ergibt sich ein leicht unterschiedliches Bild. Hier ist Kriging mit konstantem Mittel nicht mehr eindeutig als beste Methode zu identifizieren. Zwar liefert es insbesondere für größere Stichproben teilweise sehr gute Ergebnisse, jedoch streuen die RMSE-Werte insgesamt sehr stark. Dies kann eventuell durch eine starke Abhängigkeit von der Güte

des Versuchsplan erklärt werden. Gaußsche radiale Basisfunktion mit konstantem Mittel und *IDW* nach Shepard liefern wieder durchgängig das schlechteste Ergebnis, ein weiterer Unterschied zu den Latin Hypercubes ist, dass *IDW* nach Renka mit linearem Trend hier ebenfalls keine guten Ergebnisse liefert. Kerninterpolation, gaußsche radiale Basisfunktion mit linearem Trend und multiquadratische radiale Basisfunktionen liefern wieder ähnlich gute Ergebnisse. Die Resultate für die gleichverteilten Versuchspläne sind sehr ähnlich zu denen für Latin Hypercubes, nur dass die RMSE-Werte insgesamt etwas schlechter liegen als bei den Latin Hypercubes.

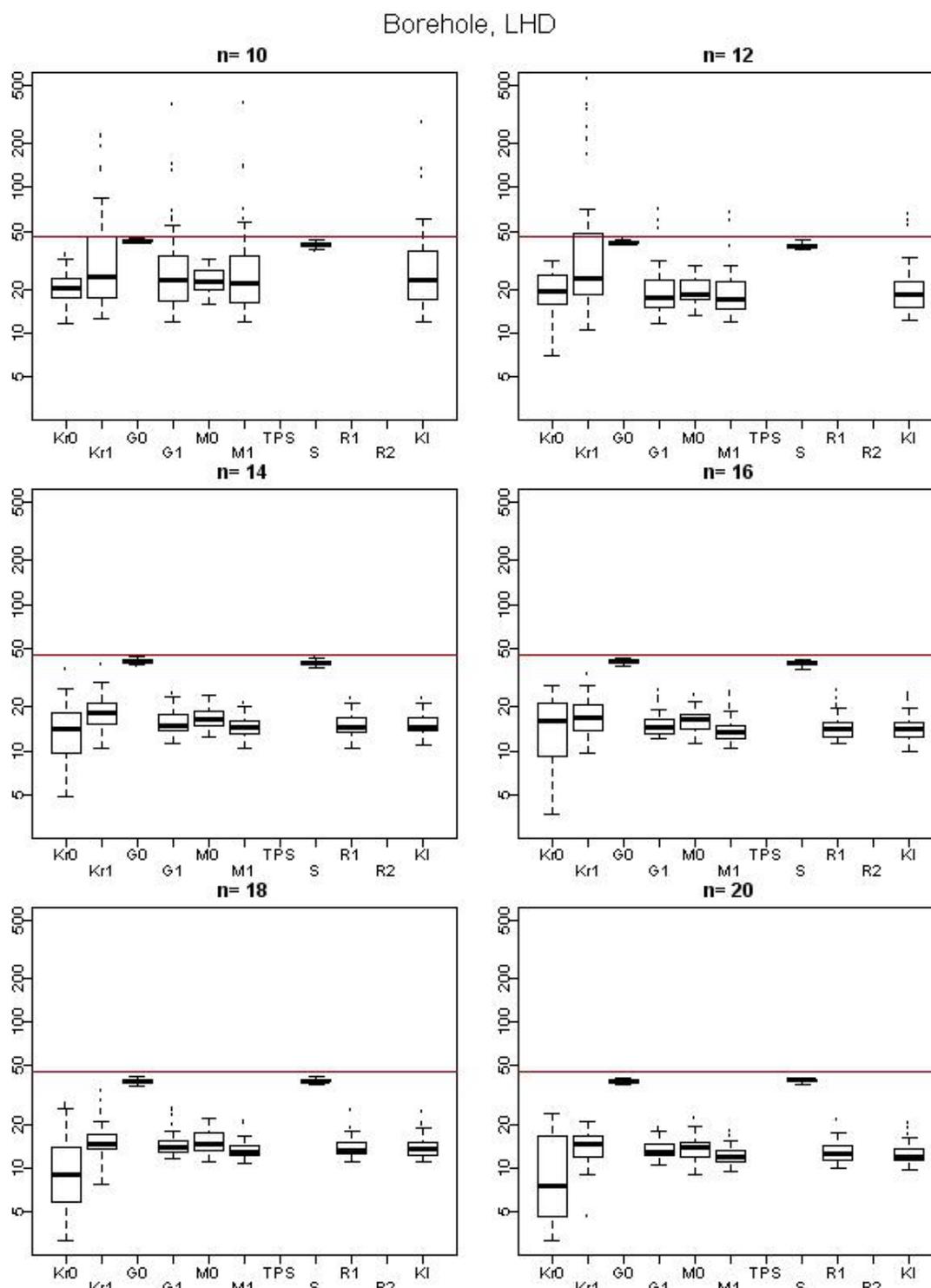


Abbildung 5.14: Boxplots der RMSE-Werte für die Bohrlochfunktion von Morris et al. (1993) basierend auf Latin Hypercubes.

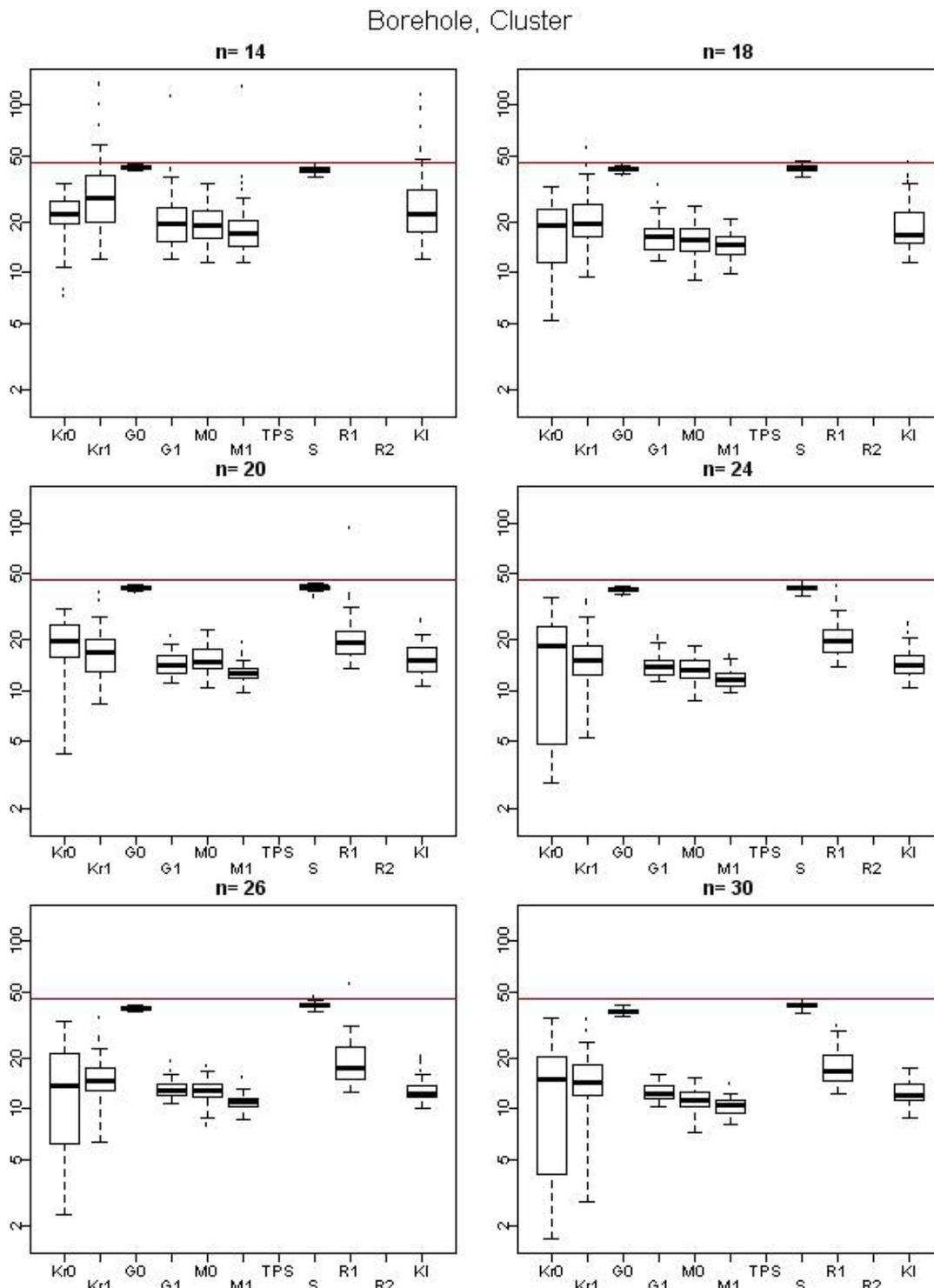


Abbildung 5.15: Boxplots der RMSE-Werte für die Bohrlochfunktion von Morris et al. (1993) basierend auf Latin Hypercubes mit Clustern.

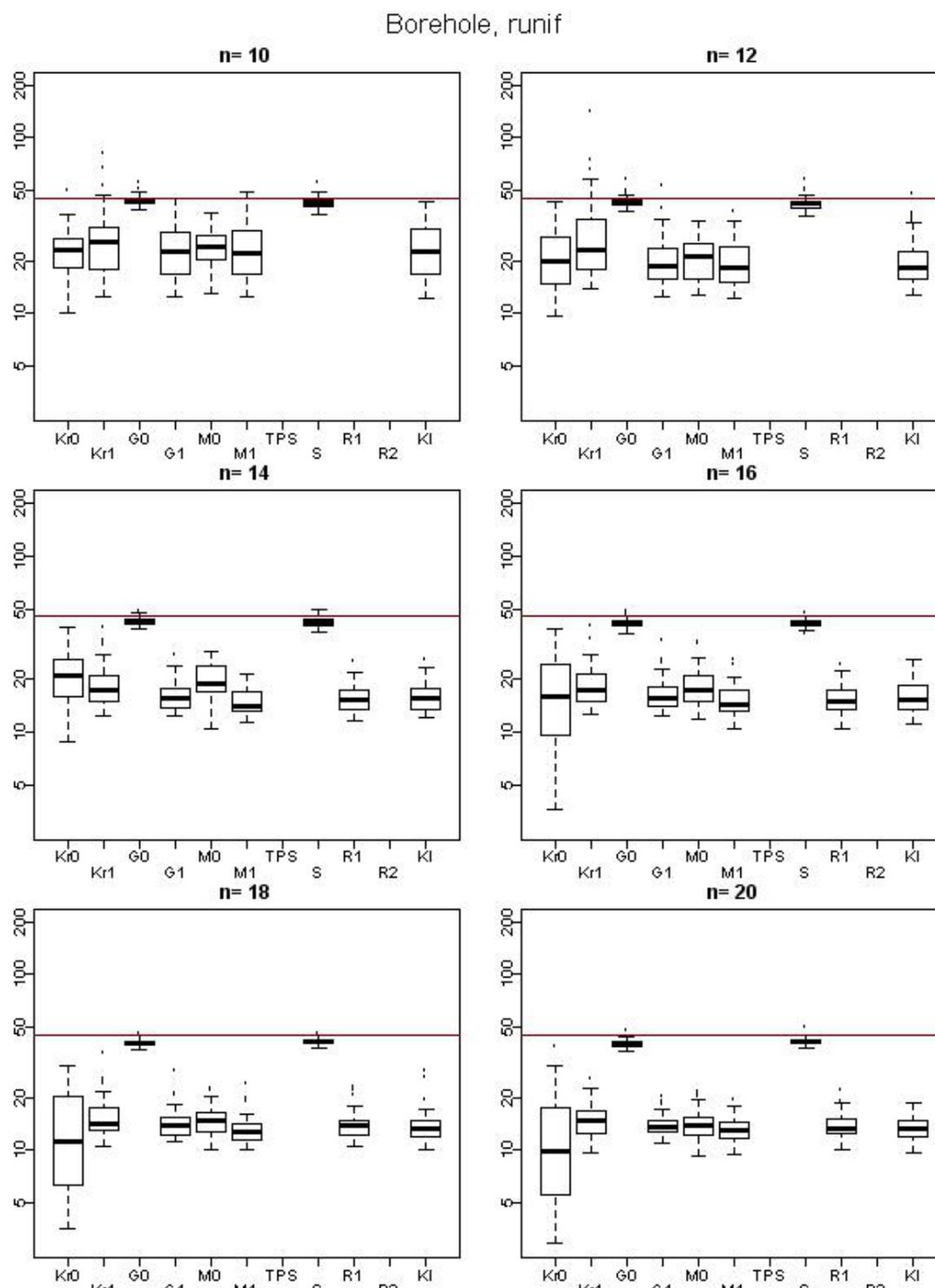


Abbildung 5.16: Boxplots der RMSE-Werte für die Bohrlochfunktion von Morris et al. (1993) basierend auf gleichverteilten Versuchsplänen.

5.5.2 Anwendungsdatensätze

Zur weiteren Analyse der Vorhersagefähigkeit der verschiedenen Modelle werden 4 Datensätze herangezogen, die sich dadurch auszeichnen, dass sowohl ein Trainingsdatensatz vorliegt, an den die Modelle angepasst werden können als auch Testdatensätze, mit denen die Anpassung der verschiedenen Modelle verglichen werden kann.

DS 1: Autoform

Der erste Datensatz kommt aus der Umformtechnik und wurde mit Hilfe der Software Autoform erstellt. Der Hintergrund ist in Gösling et al. (2009) beschrieben. In den Simulationen wird die Umformung eines Metallstreifens zu einem runden Hutprofil betrachtet (Im Gegensatz zu eckigen Hutprofilen im Kapitel 6). Es gibt drei Einflußgrößen, die die Größe der Rückfederung beeinflussen. In der Software Autoform gibt es vorimplementiert einen Monte Carlo Algorithmus, der genutzt werden kann, um Sensitivitätsstudien für die einzelnen Einflussgrößen durchzuführen. Dabei wird allerdings eine hohe Anzahl an Simulationen benötigt. Als Alternative wurde vorgeschlagen, an eine kleine Stichprobe ein Metamodell anzupassen und das Metamodell zur Sensitivitätsanalyse zu nutzen. Aufgrund dieser Simulationen gibt es einen 3^3 vollfaktoriellen Versuchsplan, einen Maximinversuchsplan mit 27 Läufen und weitere 74 Simulationsläufe zur Beurteilung der Anpassung. Angepasst wurden wieder die Modelle aus Tabelle 5.1 jeweils für den Maximinversuchsplan und den vollfaktoriellen Versuchsplan. Die Originaldaten können im Anhang 2 gefunden werden. Die RMSE-Ergebnisse sind in Tabelle 5.2 aufgeführt. Desweiteren sind in Abbildung 5.17 und Abbildung 5.18 die Vorhersagen gegen die echten Werte für die verschiedenen Interpolationsmethoden dargestellt. Insgesamt bietet die Kerninterpolation für diesen Datensatz gute Vorhersagewerte. Es fällt auf, dass die Vorhersagefähigkeit aller Modelle mit Ausnahme des *IDW* nach Shepard für den Maximinversuchsplan besser ist als für den 3^3 Versuchs-

plan bei gleiche Stichprobengröße, was als Hinweis gewertet wird, dass eine adäquate Planung von Computereperimenten sinnvoll ist, bei der nicht einfach nur Konzepte für zufällige Experimente kopiert werden. Die Abbildungen 5.17 und 5.18 zeigen außerdem, dass KI und die meisten anderen Methoden nicht systematisch von den wahren Werten abweichen. Nur für S liegen die Daten eindeutig nicht auf einer Geraden. Für den 3^3 Datensatz liefern außerdem $Kr0$ und $G0$ dem Augenschein nach keine gute Anpassung.

Interpolationsmethode	DS1(3^3)	DS1(Minmax)	DS2	DS3	DS4
$Kr0$	0.1226	0.0860	5.8267	1.4372	1.4845
$Kr1$	0.1129	0.0744	6.0821	2.7843	1.4522
$G0$	0.1771	0.1078	12.904	2.8431	1.5209
$G1$	0.0740	0.0700	7.5666	2.5623	1.3887
$M0$	0.0816	0.0720	6.0204	2.3052	1.2454
$M1$	0.0714	0.0672	5.9607	2.3370	1.2552
TPS	0.0732	0.0668	5.0075	NA	NA
S	0.1923	0.2396	26.127	3.7972	1.5432
$R1$	0.0725	0.0591	6.3260	2.3902	1.3047
$R2$	0.0683	0.0682	7.2140	NA	NA
KI	0.0739	0.0631	6.3070	2.5939	1.3854

Tabelle 5.2: RMSE-Ergebnisse

Datensatz 2: Engine Emission Data set

Der nächste Datensatz wurde in Bates et al. (2003) und Bates et al. (2008) zur Beurteilung der Anpassungsgüte für verschiedene polynomiale Interpolationsfunktionen verwandt. Die Daten sind anonymisiert, weswegen keine genaue Beschreibung des technischen Hintergrundes geliefert werden kann außer dass die Zielgröße Abgaswerte von Motoren beschreibt. Hier steht ein Trainingsdatensatz mit 48 Läufen in 5 Dimensionen und ein Testdatensatz mit 49 Läufen zur Verfügung. Die RMSE Ergebnisse sind wieder in Tabelle 5.2 (DS2) zusammengefasst. Hier liefert *KI* einen akzeptablen Wert, wenn auch nicht den besten Vorhersagewert, der hier durch *TPS* realisiert wird. Bates et al. (2008) berichten für die dort vorgestellte polynomiale Interpolationsmethode einen RMSE von 5.844. Der dort erzielte RMSE Wert von 4.450 ergibt sich mit einer anderen Korrelationsfunktion (Powerexponential entsprechend Tabelle 4.1). Die Anpassungsplots für den Testdatensatz (Abbildung 5.19) zeigt, dass die meisten Methoden wieder eine akzeptable Anpassung liefern, nur *S* und *G0* liefern Ergebnisse, die sichtlich von der Winkelhalbierenden abweichen.

Datensatz 3: Exhaust Manifold Sealing

In Fang et al. (2006), S. 172, ist eine 5-dimensionale Anwendung beschrieben, für die ein Trainingsdatensatz mit 17 Läufen und ein Testdatensatz mit 20 Läufen zur Verfügung steht. Technischer Hintergrund der Experimente ist die Analyse des Abnutzungsprozesses eines Abgaskrümmers eines Motors. Das Design des Abgaskrümmers ist technisch anspruchsvoll, da der Abgaskrümmers innerhalb kurzer Zeit sehr starke Temperaturunterschiede verkraften muss. Die Einflussvariablen für die Lebensdauer stellen geometrische Eigenschaften des Designs des Abgaskrümmers dar. In Fang et al. (2006) wird an die Daten ein Multi-layer perceptron network angepasst, das einen RMSE-Wert von 2.1853 für den Testdatensatz liefert. Allerdings ist dieser Wert nicht direkt mit den hier erzielten Werten vergleichbar, da in dem Anpassungsprozess auch der Testdatensatz verwandt wurde. Die RMSE-Ergebnisse für die hier betrachteten Methoden sind in Tabelle 5.2 (DS3) aufgeführt und der graphische Abgleich zwischen Vorhersage und Testdatensatz in Abbildung 5.20. Hier ist die Größe des Versuchsplans so klein, dass TPS und $R2$ nicht angepasst werden können. Hier liefert $Kr0$ mit Abstand den besten Wert, die restlichen Methoden liefern vergleichbare Ergebnisse, was auch anhand der Graphik erkennbar ist. Nur S liefert wieder eine deutliche systematische Abweichung.

Datensatz 4: Piston Slap

Der Piston Slap (Kolbenschlag) Datensatz (Fang et al. (2006), S. 153 ff.) besteht aus einem Trainingsdatensatz mit 12 Läufen und weiteren 100 Läufen in einem Testdatensatz. Der Anwendungshintergrund bezieht sich auf die Motorenentwicklung im Automobilbau. Kolbenschlag ist hier ein unerwünschtes Geräusch, das die Kundenzufriedenheit stark beeinträchtigt. Es existiert ein Programm, das den Kolbenschlag simuliert in Abhängigkeit verschiedener Einstellmöglichkeiten am Motor. Hier liefert *KI* einen akzeptablen Wert, wenn auch nicht einen der besten Werte, der durch *M1* realisiert wird. Die Anpassungsplots in Abbildung 5.21 zeigen auf, dass bei diesen Experimenten alle Methoden eine relativ hohe Streuung um die Winkelhalbierende aufweisen und desweiteren auch eine systematische Abweichung von der Winkelhalbierenden zu erkennen ist. Dies deutet darauf hin, dass insgesamt der Trainingsdatensatz eventuell zu klein ist, um eine akzeptable Anpassung zu erreichen.

5.6 Schlussfolgerungen

Die hier vorgestellte Methode der Kerninterpolation liefert insgesamt gute Ergebnisse. Die Anpassung benötigt keinen Optimierungsprozess wie das Kriging, ist auch bei sehr kleinen Stichproben anwendbar und erbringt gute Vorhersagewerte. Es kann eine zum Kriging ähnliche, lokale Unsicherheitsaussage gemacht werden und ist linear in den Beobachtungswerten. Die Methode hat einen engen Bezug zum Inverse Distance Weighting, übertrifft diese jedoch meistens in der Vorhersagegüte sowohl bei den analytischen Beispielen als auch bei den Testdatensätzen. Gemessen an dem RMSE ist die Vorhersagegüte der Kerninterpolation ähnlich zu der Vorhersagegüte von multiquadratischen radialen Basisfunktionen, hat allerdings den Vorteil, dass eine Unsicherheits-schätzung vorliegt. Kriging ist insbesondere bei größeren Stichproben oftmals besser im Sinne des RMSE, bei kleinen Stichproben kann sich dieser Effekt allerdings aufheben. Ein Nachteil des Kriging, der sich in den analytischen Funktionen gezeigt hat, ist dass die unterschiedlichen RMSE-Werte oftmals stark streuen, was als eine starke Abhängigkeit der Vorhersagefähigkeit des Kriging von dem Versuchsplan interpretiert werden kann. Andere Methoden wie multiquadratische radiale Basisfunktionen und die Kerninterpolation erweisen sich diesbezüglich als robuster. Ein potentieller Nachteil ist, dass aus praktischen Gründen die Dimension des Interpolationsproblems nicht beliebig hoch angesetzt werden kann, da ansonsten die Anzahl der Simplexe in der benötigten Delaunaytriangulation zu stark ansteigt. Zusammenfassend kann gesagt werden, dass keine der hier betrachteten Interpolationsmethoden immer am Besten war, vielmehr ist es eine Anforderung an den Anwender, unterschiedliche Modelle auf ihre Eignung hin zu vergleichen.

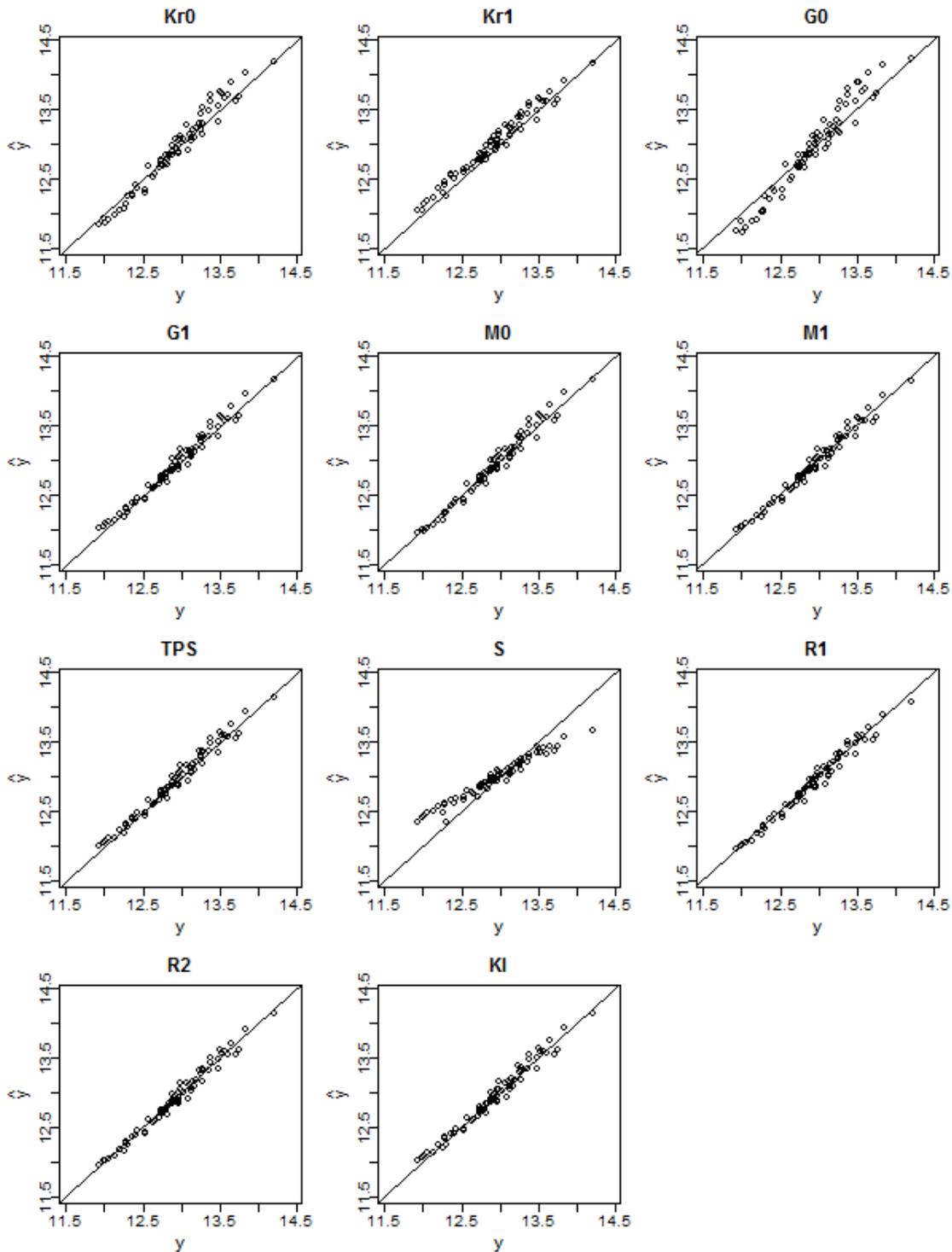


Abbildung 5.17: Graphischer Vergleich von vorhergesagten Werten und Werten des Testdatensatzes für den Autoform Datensatz basierend auf dem vollfaktoriellen 3^3 Versuchsplan.

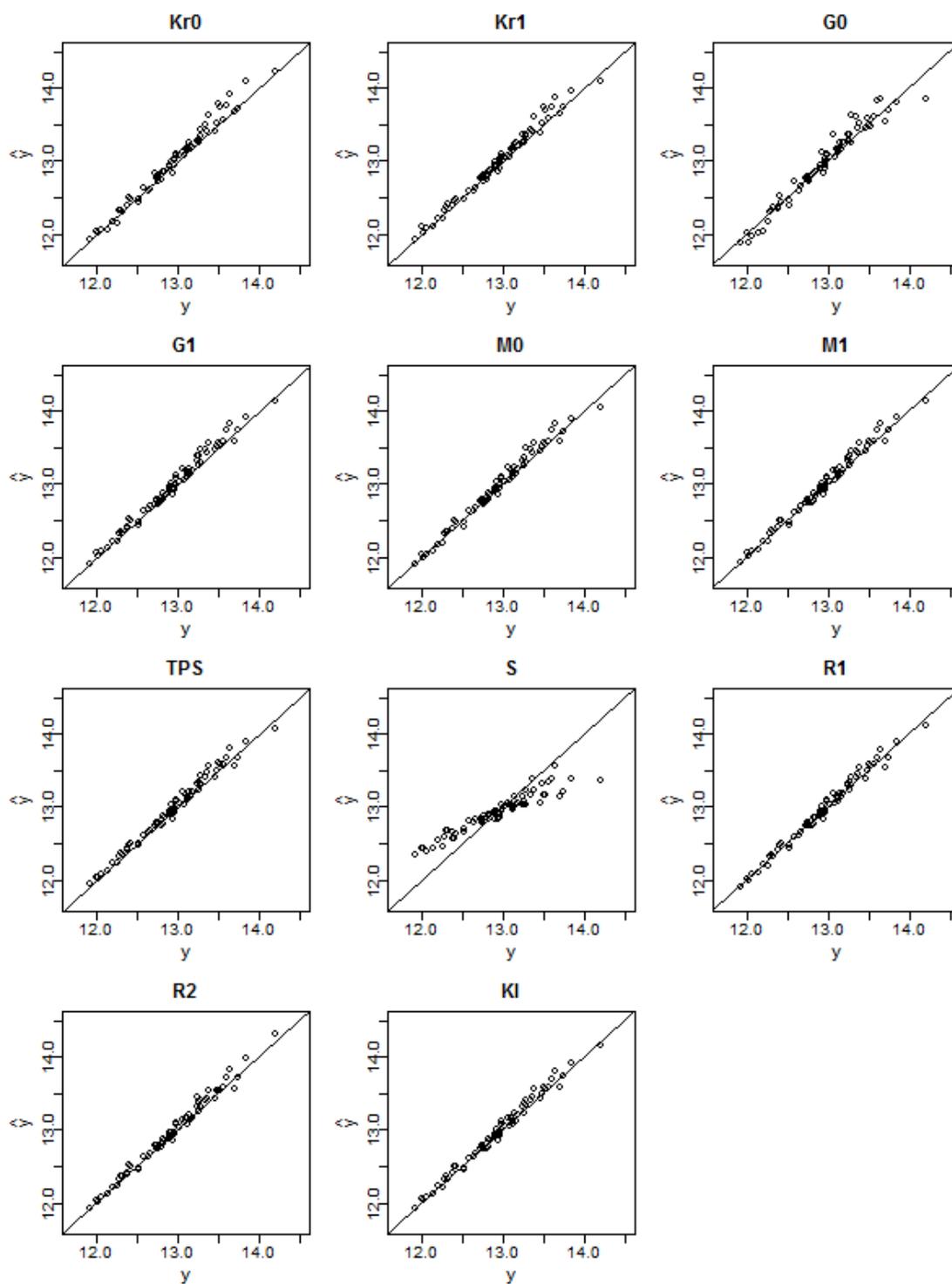


Abbildung 5.18: Graphischer Vergleich von vorhergesagten Werten und Werten des Testdatensatzes für den Autoform Datensatz basierend auf dem Maximinversuchsplan.

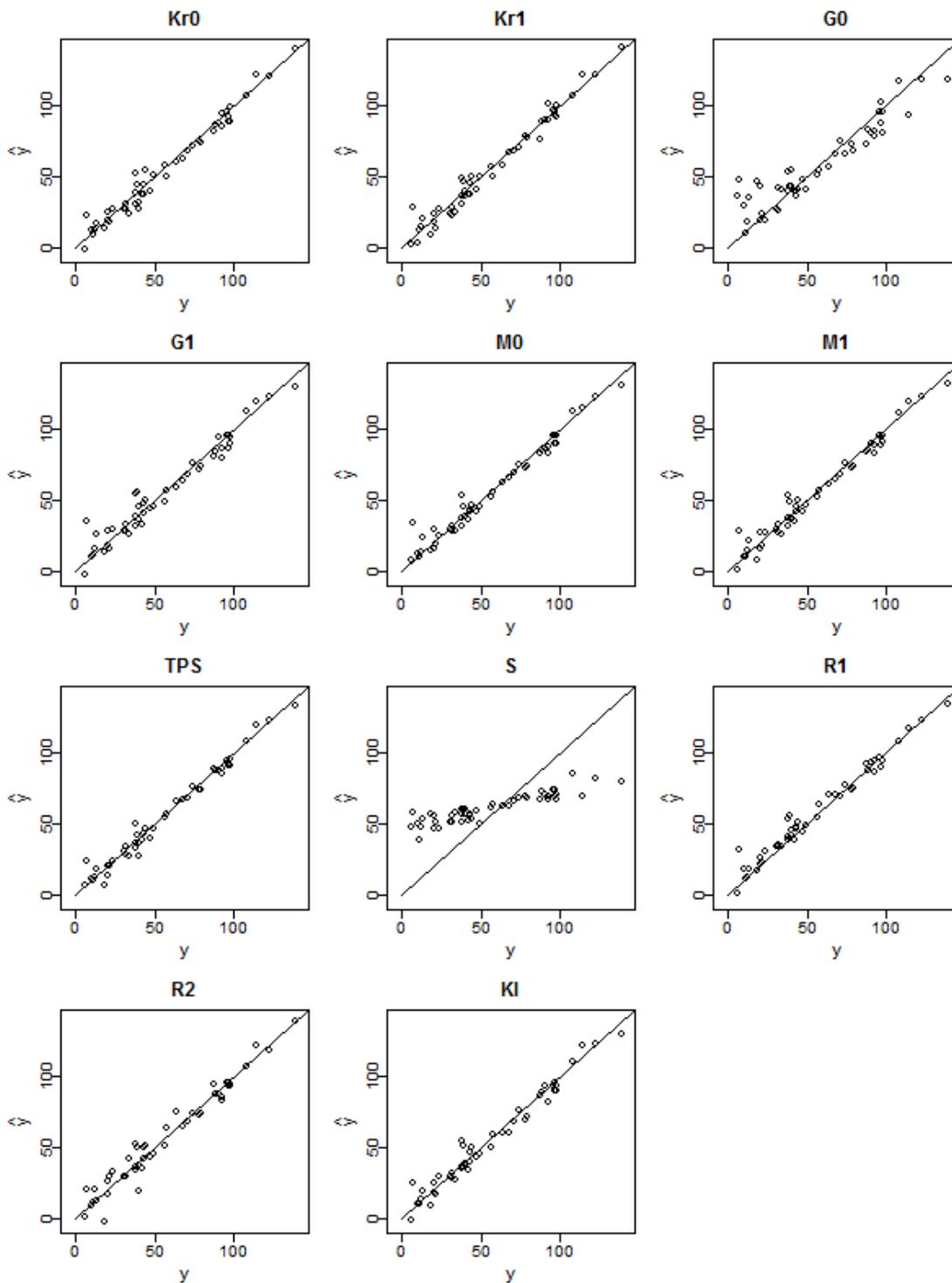


Abbildung 5.19: Graphischer Vergleich von vorhergesagten Werten und Werten des Testdatensatzes für den Engine Emission Datensatz.

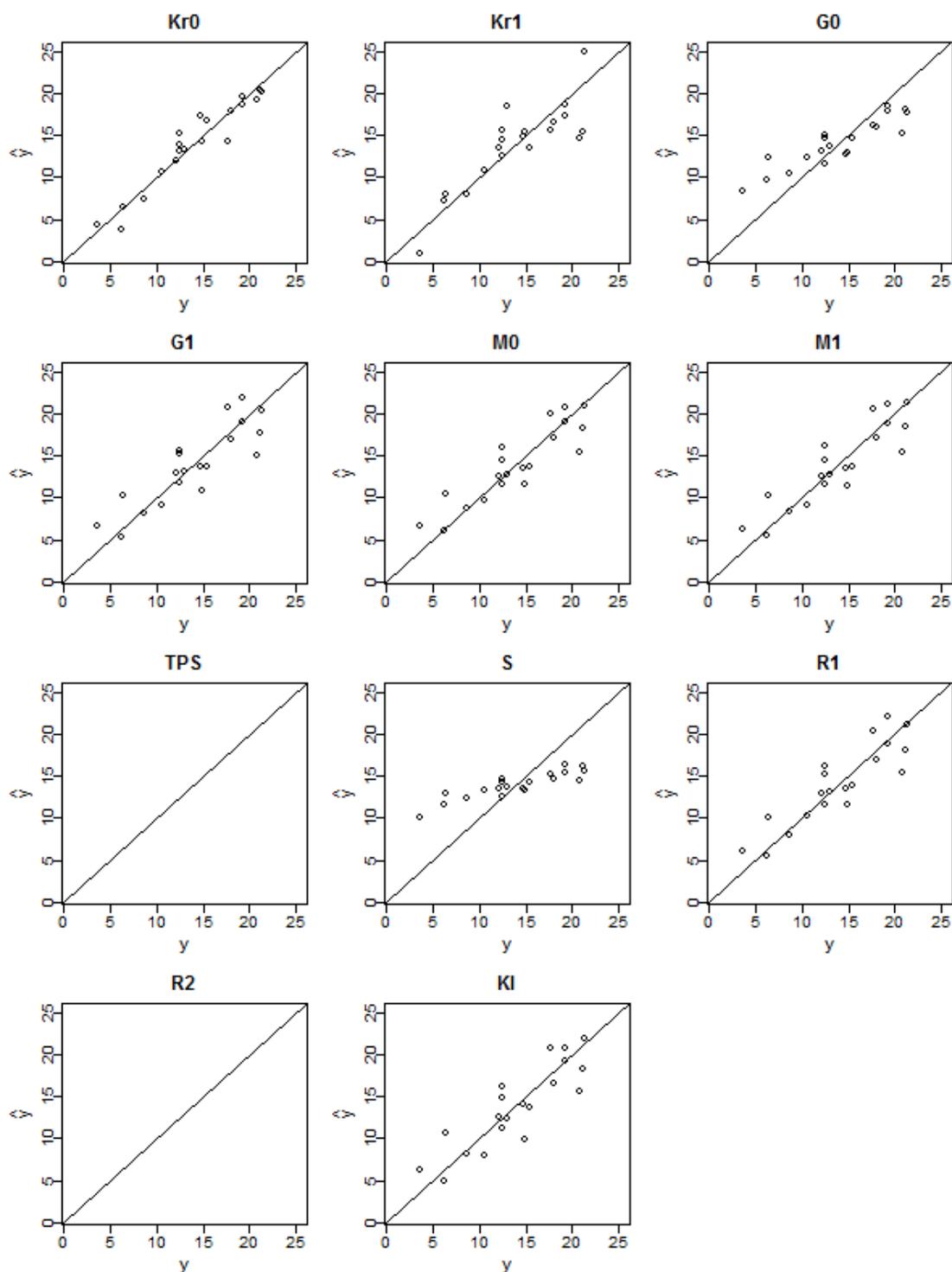


Abbildung 5.20: Graphischer Vergleich von vorhergesagten Werten und Werten des Testdatensatzes für den Exhaust Manifold Datensatz. Ein leeres Feld zeigt an, dass die entsprechende Interpolationsmethode aufgrund eines zu kleinen Datensatzes nicht angewandt werden konnte (*TPS* und *R2*).

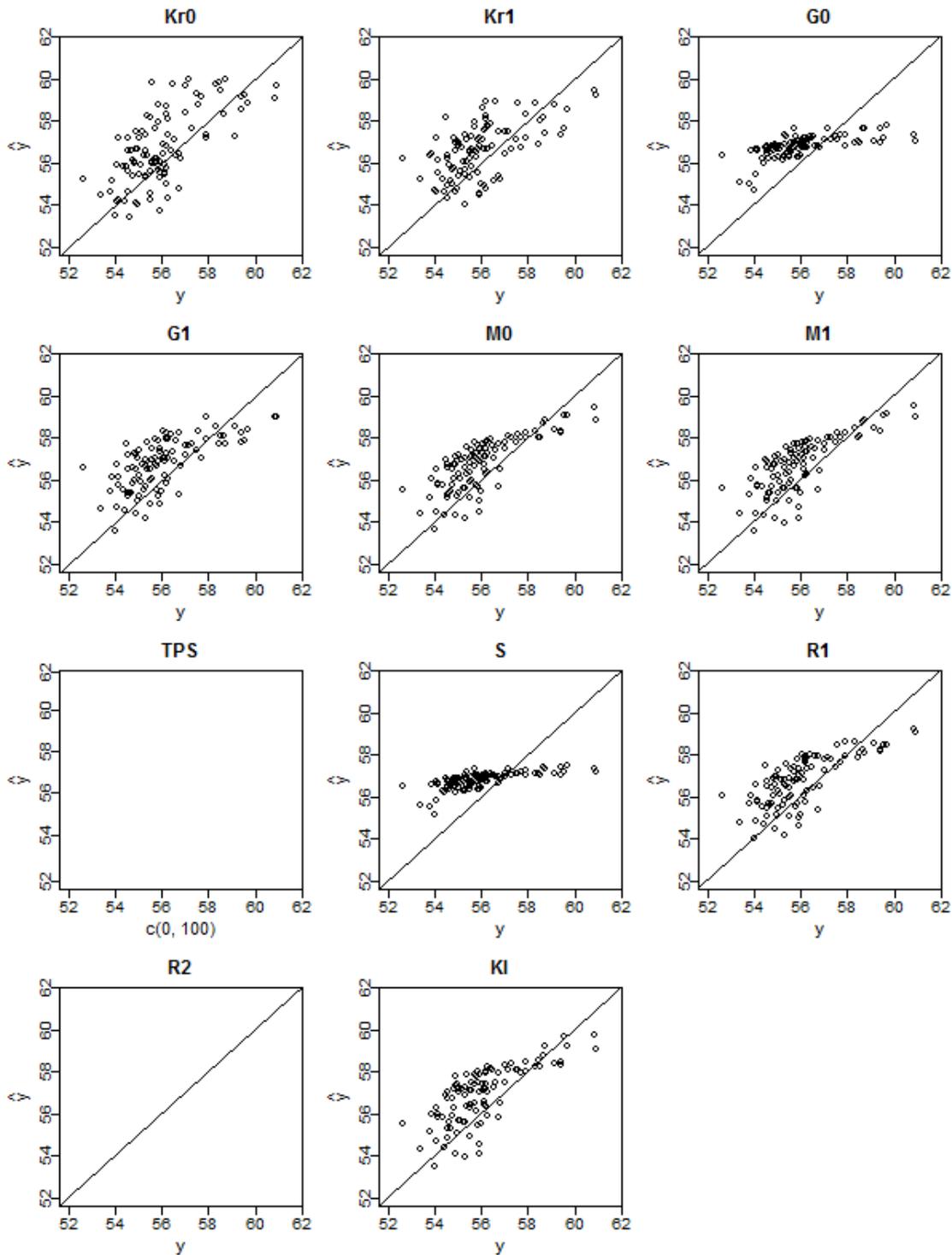


Abbildung 5.21: Graphischer Vergleich von vorhergesagten Werten und Werten des Testdatensatzes für den Piston Slap Datensatz. Ein leeres Feld zeigt an, dass die entsprechende Interpolationsmethode aufgrund eines zu kleinen Datensatzes nicht angewandt werden konnte (*TPS* und *R2*).

VALIDIERUNG VON COMPUTEREXPERIMENTEN

Computerexperimente haben sich schon in sehr vielen Bereichen als nützlich erwiesen, da sie meistens kostengünstiger und schneller als echte Experimente sind. Jedoch ist nicht von vorneherein gesagt, dass das Computerexperiment präzise entsprechende echte Experimente wiedergibt. Zu analysieren, ob systematische Unterschiede zwischen Simulation und echtem Experiment vorliegen, ist die Aufgabe der Validierung von Computerexperimenten. Meistens ist dies ein schwieriger Prozess, da oftmals das Durchführen von echten Experimenten kostspieliger und komplizierter ist als die Simulation. Nichtsdestotrotz kommt der Validierung eine zentrale Bedeutung zu, gerade weil echte Experimente weitestgehend ersetzt werden sollen. In der Literatur existieren verschiedene Ansätze, um eine Validierung durchzuführen und gegebenenfalls eine Korrektur einer vorliegenden Verzerrung vorzunehmen (siehe z.B. Bayarri et al. (2002), Bayarri et al. (2007), Higdon et al. (2004), Kennedy und O'Hagan (2001), Cox et al. (2001), Hidgon et al. (2004), Reese et al. (2004)). Jedoch scheint bisher kein statistischer Test vorzuliegen, der die Korrektheit eines Computerexperiments überprüft. Prinzipiell kann dies durch beliebige Anpassungstests geschehen, nachfolgend wird speziell ein Permutationstest konstruiert.

Desweiteren gibt es bei vielen Computerexperimenten mehr als eine Simulation, die als Ersatz für echte Experimente herangezogen werden kann. Auch für diese Situation gibt es bereits umfangreiche Arbeiten in der Literatur, jedoch keine, die exakt auf die

Datensituation zutreffen, wie sie an einem hier analysierten Datensatz vorliegen. In den Ansätzen von Kennedy und O'Hagan (2000), Qian und Wu (2008) und Qian et al. (2006) wird die Situation vorausgesetzt, dass eine Simulation vorliegt, die für jede Stelle des Versuchsraums das beste Vorhersageergebnis der echten Ergebnisse liefert, die Auswertung dieser Simulation jedoch sehr teuer ist. Als Alternative steht mindestens eine weitere leicht zu evaluierendes Simulation zur Verfügung, für die bekannt ist, dass sie eine schlechtere Vorhersagegüte haben. In den genannten Arbeiten wird aus diesen beiden unterschiedlichen Arten von Simulationen ein gemeinsames Modell erarbeitet. Es gibt jedoch auch Situationen, in denen nicht aus theoretischen Gründen oder Erfahrungswerten bekannt ist, welche Simulation die Bessere ist. Deshalb wird hier eine Methode vorgestellt, die für einen Punkt im Versuchsraum die Entscheidung trifft, welche von mehreren vorliegenden Simulation lokal die beste ist.

6.1 Anwendungsbeispiel

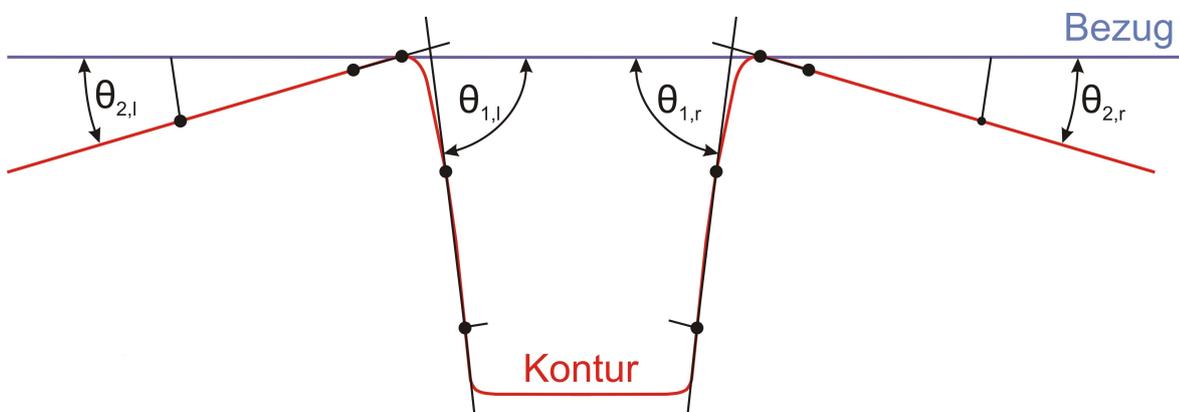


Abbildung 6.1: Das Hutprofil ist eine einfache geometrische Form, die tendenziell starkes Rückfederungsverhalten aufweist und deshalb zum Studium derselben ausgewählt wurde.

Die Motivation für dieses Kapitel liefert ein Datensatz, der sich ebenso wie der Datensatz aus Kapitel 5 mit der Rückfederungsproblematik beschäftigt. Betrachtet wird

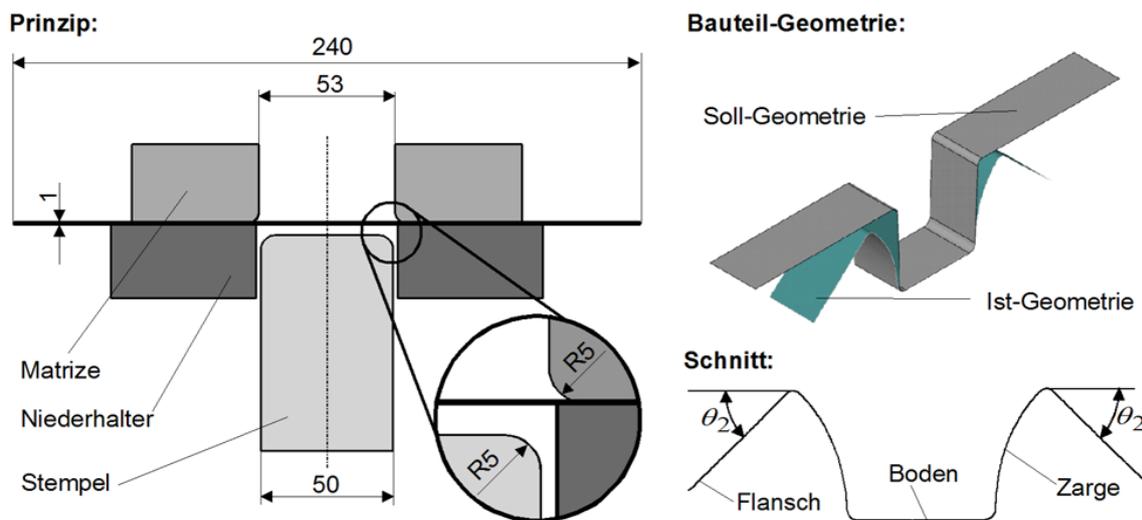


Abbildung 6.2: Schematische Darstellung des Werkzeugs (Erichsen, Modell 142/20) zur Herstellung der Hutprofile.

wieder das Hutprofilziehen, welches in Abbildung 6.1 und Abbildung 6.2 graphisch beschrieben ist. Die Zielgröße zur Beschreibung der Rückfederung ist der Flanschwinkel θ_2 , der in Abbildung 6.2 dargestellt ist. Die Zielgröße θ_2 ist so gewählt, dass kleinere Werte eine geringeres Rückfederungsverhalten widerspiegeln, wobei der Wert 0 angenommen wird, falls keine Rückfederung auftritt. In dem hier betrachteten Versuchsaufbau wurden zwei Einflussgrößen variiert: Der Werkzeugradius (3, 5, 7 mm) und die Ziehtiefe (40, 50, 60 mm). Der Werkzeugradius ist in Abbildung 6.2 dargestellt (Radius $R5$). Die Ziehtiefe bestimmt, wie tief das resultierende Hutprofil ist. In vorherigen Experimenten hat sich erwiesen, dass diese beiden Einflussgrößen einen starken Einfluss auf das Rückfederungsverhalten haben. Für die Vorhersage der echten Experimente stehen drei Finite Elemente Computersimulationen zur Verfügung. Diese unterscheiden sich in der Konstruktion der Finite Elemente Diskretisierung für das Werkstück und in der Wahl der Zeitschritte für das inkrementelle Vorgehen der FE Simulation, (siehe 6.1). Weitere Details zu den zugrundeliegenden Computersimulationen können Gösling (2010) entnommen werden.

Als Daten für die echten Experimente liegen Beobachtungen an allen Stellen eines 3^2

Versuchsplan vor mit Ausnahme der Kombination $(-, -) = (3mm, 40mm)$. Auf jeder der Stufen liegen zwischen 3 und 12 Wiederholungen vor. Für die drei unterschiedlichen Computersimulationen wurde der jeweils ein (vollständiger) 3^2 Versuchsplan zu Erhebung der Daten genutzt, jedoch diesmal ohne Wiederholungen (da alle drei Simulationen deterministisch sind). Die echten Experimente sowie auch die Computerexperimente wurden an dem Institut für Umformtechnik und Leichtbau der TU Dortmund durchgeführt. Die Computerexperimente wurden mit der Software Autoform durchgeführt. Abbildung 6.3 liefert einen ersten graphischen Eindruck der Daten. Es ist erkennbar, dass bei der Zielgröße ein Unterschied zwischen den Ergebnissen der echten Experimente und den Simulationen vorhanden ist. Keine der Simulationen scheint ein perfekter Ersatz für die echten Experimente zu sein. Simulationsmodell 1 weist dabei deutliche Abweichungen von den experimentellen Daten auf, wird jedoch mit in den Vergleich aufgenommen, da es ein Standardmodell für die FE-Modellierung von Umformprozessen ist. Bei den Simulationsmodellen $S2$ und $S3$ ist aufgrund des graphischen Vergleichs keine eindeutige Struktur zu erkennen, welches Simulationsmodell wo bessere Vorhersagen liefert. Der resultierende Datensatz ist im Anhang verzeichnet (Anhang D).

	Diskretisierung	Zeitschritte
$S1$	lokal unterschiedlich, abhängig von dem Werkzeugradius	0.5
$S2$	Gleichmässig ($1mm$)	Voreingestellter Wert von Autoform, adaptiv entsprechend der maximalen Displacement
$S3$	Gleichmässig ($1mm$)	0.5

Tabelle 6.1: Spezifikationen der einzelnen Simulationen.

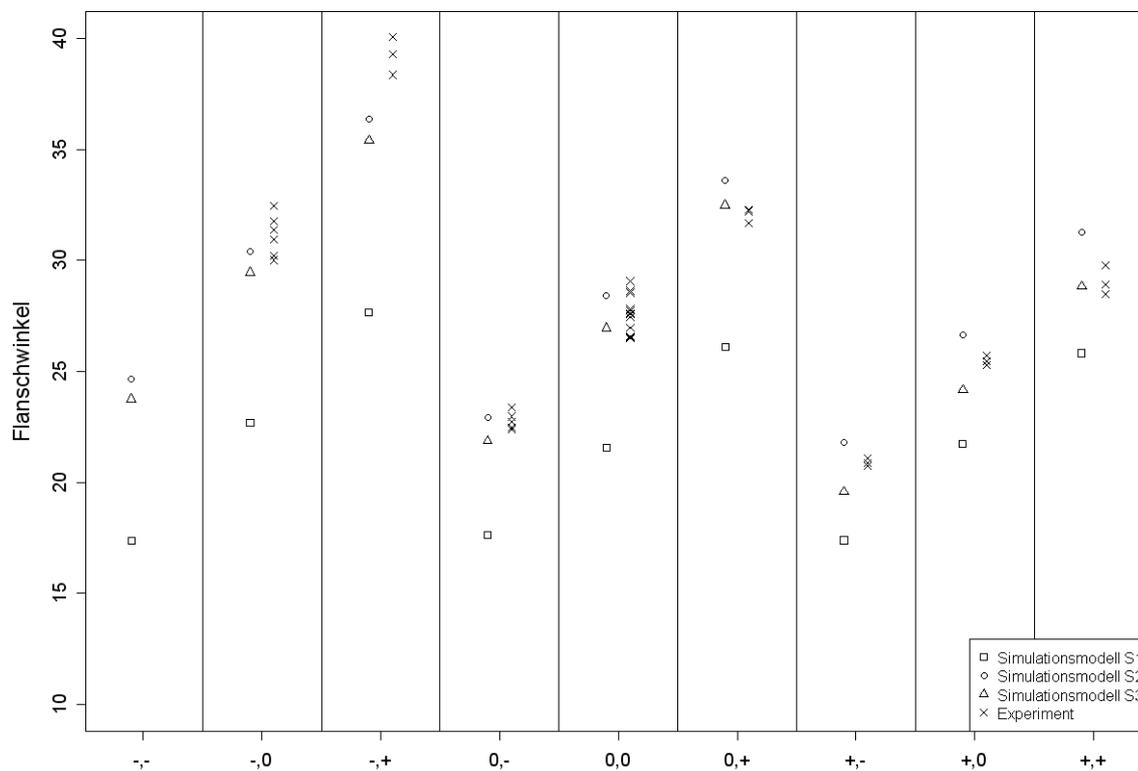


Abbildung 6.3: Graphische Darstellung der Simulationsergebnisse im Vergleich zu den echten Experimenten.

6.2 Test auf Korrektheit des Computerexperiments

Da Computerexperimente in den meisten Fällen nur eine Modellierung der Wirklichkeit durch mathematische Hilfsmittel darstellen, ist es sinnvoll, zu hinterfragen, wie gut diese Modellierung gelingt. Im Idealfall liefert das Computerexperiment eine perfekte Vorhersage des Mittelwertes der echten physikalischen Experimente. Übersetzt in den Kontext statistischer Tests ist die Nullhypothese von Interesse, ob das Computerexperiment „korrekt“ ist, gegen die Alternative, dass es Bereiche in dem Versuchsraum gibt, bei denen das Computerexperiment die echten Experiment nicht korrekt widerspiegelt. Diese Fragestellung soll mit Hilfe eines Permutationstests überprüft werden. Permuta-

tionstests bilden eine attraktive Klasse von statistischen Tests, da sie mit sehr geringen Annahmen auskommen, ihre Gültigkeit nicht auf Asymptotik beruht, sie aber dennoch oftmals asymptotisch die gleiche Güte haben wie gleichmässig beste unverfälschte Tests. Nachfolgend wird die Notation von Lehmann und Romano (2005) verwandt, die zunächst kurz vorgestellt wird. Die in diesem Kapitel vorgestellten Tests sind Weiterentwicklungen von Tests, die in Mühlenstädt und Gather (2008) beschrieben sind.

6.2.1 Permutationstests

Sei $T(\vec{Y})$ eine reelwertige Statistik für eine Hypothese H_0 , die so konstruiert ist, dass große Werte von $T(\vec{Y})$ gegen H_0 sprechen. \vec{Y} sei ein reelwertiger Zufallsvektor mit Beobachtungen $\vec{y} \in \mathbf{Y}$, wobei \mathbf{Y} der Stichprobenraum von \vec{Y} ist. Sei \mathbf{G} eine endliche Gruppe von Transformationen, die \mathbf{Y} nach \mathbf{Y} abbildet mit Kardinalität M , d.h. $\#\mathbf{G} = M$. Für einen Beobachtungsvektor \vec{y} von \vec{Y} seien

$$T^{(1)}(\vec{y}) \leq T^{(2)}(\vec{y}) \leq \dots \leq T^{(M)}(\vec{y})$$

die geordneten Werte der Teststatistiken $T(g\vec{y})$ für alle g in \mathbf{G} . Weiter sei definiert

$$q := M - \lfloor M\alpha \rfloor, \quad \alpha \in (0, 1),$$

wobei $\lfloor \cdot \rfloor$ die Gaussklammer darstellt,

$$M^+(\vec{y}) := \#\{T^{(j)}(\vec{y}) | j = 1 \dots M, T^{(j)}(\vec{y}) > T^{(q)}(\vec{y})\},$$

$$M^0(\vec{y}) := \#\{T^{(j)}(\vec{y}) | j = 1 \dots M, T^{(j)}(\vec{y}) = T^{(q)}(\vec{y})\}$$

und die Randomisationskonstante $a(\vec{y})$ als

$$a(\vec{y}) := \frac{M\alpha - M^+(\vec{y})}{M^0(\vec{y})}. \quad (6.1)$$

Definition 6.1 (Permutationstest). *Mit den eben getroffenen Definitionen der Randomisationskonstante $a(\vec{y})$, den geordneten Teststatistiken $T^{(1)}(\vec{y}) \leq \dots \leq T^{(M)}(\vec{y})$,*

$\alpha \in (0, 1)$ und $q = M - \lfloor M\alpha \rfloor$ ist ein Permutationstest $\Phi(\vec{y})$ definiert durch:

$$\Phi(\vec{y}) = \begin{cases} 1, & \text{falls } T(\vec{y}) > T^{(q)}(\vec{y}) \\ a(\vec{y}), & \text{falls } T(\vec{y}) = T^{(q)}(\vec{y}) \\ 0, & \text{falls } T(\vec{y}) < T^{(q)}(\vec{y}). \end{cases} \quad (6.2)$$

Laut Konstruktion gilt für jedes $\vec{y} \in \mathbf{Y}$:

$$\sum_{g \in \mathbf{G}} \Phi(g\vec{y}) = M^+(\vec{y}) + a(\vec{y})M^0(\vec{y}) = M\alpha.$$

Der Test $\Phi(\vec{y})$ hält das Niveau α ein, falls die Zufallsvariablen $T(\vec{Y})$ und $T(g\vec{Y})$ die gleiche Verteilung haben für alle Transformationen der Gruppe \mathbf{G} . Der folgende Satz ist eine Erweiterung von Theorem 15.2.1 in Lehmann und Romano (2005) und wurde in Mühlenstädt (2007) erarbeitet.

Satz 6.1. *Der Zufallsvektor \vec{Y} habe die Verteilung $P \in \mathcal{P}$. Betrachtet sei die Nullhypothese $\mathcal{P}_0 \subset \mathcal{P}$ und es sei \mathbf{G} eine endliche Gruppe von Transformationen, die \mathbf{Y} nach \mathbf{Y} abbilden. Ist die Teststatistik T für jedes $P \in \mathcal{P}_0$ invariant unter Transformationen $g \in \mathbf{G}$, d.h. $T(\vec{Y})$ und $T(g\vec{Y})$ haben die selbe Verteilung für alle $g \in \mathbf{G}$, dann gilt:*

$$E_P[\Phi(T(\vec{Y}))] = \alpha, \quad \text{für alle } P \in \mathcal{P}_0,$$

wobei $\Phi(T(\vec{Y}))$ ein Permutationstest wie oben beschrieben ist.

Beweis. Laut Konstruktion gilt

$$\sum_{g \in \mathbf{G}} \Phi(T(g\vec{y})) = M^+(\vec{y}) + a(\vec{y})M^0(\vec{y}) = M\alpha \quad \text{für alle } \vec{y} \in \mathbf{Y}.$$

Daraus folgt

$$M\alpha = E_P\left[\sum_g \Phi(T(g\vec{Y}))\right] = \sum_g E_P[\Phi(T(g\vec{Y}))].$$

Unter der Nullhypothese gilt, dass $E_P[\Phi(T(g\vec{Y}))] = E_P[\Phi(T(\vec{Y}))]$, woraus folgt, dass

$$M\alpha = \sum_g E_P[\Phi(T(\vec{Y}))] = ME_P[\Phi(T(\vec{Y}))] \Leftrightarrow \alpha = E_P[\Phi(T(\vec{Y}))].$$

□

Lehmann und Romano (2005) fordern, dass die Verteilung von \vec{Y} invariant ist unter der Nullhypothese. Jedoch ist der Test $\Phi(\vec{Y})$ auch dann ein Test zum Signifikanzniveau α , wenn nur die Teststatistik selbst invariant ist unter allen Transformationen $g \in \mathbf{G}$. Dies ist nur eine kleine Erweiterung des Theorems von Lehmann und Romano (2005), jedoch folgt daraus, dass der im folgenden vorgeschlagene Test auch für heteroskedastische Regressionsmodelle seine Gültigkeit behält. In dem nächsten Abschnitt wird die Randomisationskonstante $a(\vec{y})$ weggelassen, dementsprechend wird die Nullhypothese nur abgelehnt, wenn $T(\vec{y}) > T^{(a)}(\vec{y})$. Es ist leicht nachzuprüfen, dass der Test $\Phi(\vec{Y})$ auch dann ein Test zum Signifikanzniveau α ist.

6.2.2 Permutationstests für Computereperimente

Im folgenden wird ein von Good (2000) vorgeschlagener Test abgeändert, der für manche Alternativen inkonsistent ist. Das Computereperiment wird hier als eine unbekannte stetige Funktion $\tilde{\mu}(\vec{x})$ interpretiert, die von $\vec{x} \in \mathbb{R}^d$ abhängt. Ausserdem seien $Y_1 := Y(\vec{x}_1), \dots, Y_n := Y(\vec{x}_n)$ Zufallsvariablen, für die das folgende Regressionsmodell angenommen wird: $Y(\vec{x}_i) = \mu(\vec{x}_i) + \varepsilon(x_i)$. Dabei ist $\mu(\vec{x})$ eine stetige Funktion. Die Zufallsvariablen $\varepsilon(x_1), \dots, \varepsilon(x_n)$ sind stochastisch unabhängig und um 0 verteilt, ihre genaue Verteilung kann z.B. über die Varianz von \vec{x} abhängen. Es soll die Hypothese $H_0 : \mu(\vec{x}) = \tilde{\mu}(\vec{x})$ gegen $H_1 : \exists \vec{x} \in \mathbb{R}^d : \mu(\vec{x}) \neq \tilde{\mu}(\vec{x})$ überprüft werden, d.h. es soll überprüft werden, ob die Mittelwertfunktion $\mu(\vec{x})$ überall mit den Ergebnisse der Computereperimente, gegeben durch $\tilde{\mu}(\vec{x})$, übereinstimmt. Für die Zufallsvariablen Y_1, \dots, Y_n soll gelten, dass für jedes feste $\vec{x} \in \mathbb{R}^d$, $Y(\vec{x})$ eine stetige, um $\mu(\vec{x})$

symmetrische Verteilung besitzt. Definiere die Zufallsvariable

$$Z(\vec{x}) := \begin{cases} 0, & \text{falls } Y(\vec{x}) - \tilde{\mu}(\vec{x}) > 0; \\ 1, & \text{sonst.} \end{cases}$$

Good (2000), S. 126, schlägt $T(\vec{x}_1, \dots, \vec{x}_n) := \sum_{i=1}^n Z(\vec{x}_i)$ als Teststatistik für das obige Testproblem vor und benutzt alle möglichen Kombinationen von Vorzeichenwechseln von $Y(\vec{x}_i) - \tilde{\mu}(\vec{x}_i)$, $i = 1, \dots, n$, als Transformationsgruppe. Unter der Nullhypothese gilt $P(Y(\vec{x}_i) - \tilde{\mu}(\vec{x}_i) > 0) = P(Y(\vec{x}_i) - \tilde{\mu}(\vec{x}_i) < 0) = 0.5$ für alle $i = 1, \dots, n$. Auch wenn die Differenzen selbst unter der Nullhypothese nicht notwendigerweise identisch verteilt sind, sind die Hilfsvariablen $Z(\vec{x}_i)$ identisch verteilt. Daraus folgt sofort, dass jeder Test, der auf $Z(\vec{x}_1), \dots, Z(\vec{x}_n)$ aufbaut, alle Vorzeichenwechsel als Transformationsgruppe verwendet und entsprechend Kapitel 6.2.1 konstruiert ist, das Niveau einhält. Die von Good vorgeschlagene Teststatistik ist unter der Nullhypothese binomialverteilt $\mathbb{B}(n, 0.5)$. Jedoch ist dieser Test inkonsistent für manche Alternativen. Als eindimensionales Beispiel des obigen Regressionsmodells betrachte $\mu(x) = \beta x$ und $\tilde{\mu}(x) = \gamma x$ mit $\beta \neq \gamma \in \mathbb{R}$, $x \in [-1, 1]$ und $\varepsilon(x_1), \dots, \varepsilon(x_2)$ unabhängig identisch normalverteilt mit Erwartungswert Null und Varianz σ^2 . Für Stichproben mit $-1 = x_1, \dots, x_n = 1$ äquidistant auf $[-1, 1]$ gilt, dass für die Hälfte der Beobachtungen $\mu(x_i) \leq \tilde{\mu}(x_i)$ gilt und für die andere Hälfte $\mu(x_i) \geq \tilde{\mu}(x_i)$. Auch wenn die Alternative wahr ist, wird die von Good vorgeschlagene Teststatistik mit hoher Wahrscheinlichkeit mittlere Werte annehmen und somit nicht ablehnen.

Um diese Inkonsistenz zu vermeiden, wird Good's Test lokal angewandt. Es wird für jeden Datenpunkt eine Teilmenge s_i der n Beobachtungen mit $k_i + 1$ Datenpunkten erstellt ((\vec{x}_i, y_i) und k_i weitere Beobachtungspunkte). Für jede Teilmenge s_i wird dann die Teststatistik $T_i := T(s_i)$ berechnet. Unter H_0 sind die Statistiken T_i binomialverteilt mit Parametern $(k_i + 1, (k_i + 1)/2)$, jedoch nicht unabhängig. Als globale Teststatistik wird

$$T^* := \sum_{i=1}^n D_i,$$

mit $D_i := \left(T_i - \frac{k_i+1}{2}\right)^2$ verwandt, welche klein unter der Nullhypothese ist. Wird als Transformationsgruppe wieder die Vertauschung aller Vorzeichen verwandt, ergibt dies einen Test, der das Niveau einhält, aufgrund von Satz 6.1.

Offen geblieben ist bisher, wie die Teilmengen s_i definiert werden. Im folgenden werden drei Möglichkeiten erwogen: Erstens können die Teilmengen entsprechend der k_i nächsten Nachbarn (k nearest neighbour, kurz knn) in Bezug auf die \vec{x} -Werte gebildet werden. knn Methoden sind weitverbreitete Hilfsmittel in der Klassifikation, siehe z.B. Hastie et al. (2001). Die k_i nächsten Nachbarn eines Punktes \vec{x}_i sind definiert als die k_i Punkte $\vec{x}_i^{(1)}, \dots, \vec{x}_i^{(k_i)}$, die die k_i kleinsten euklidischen Abstände von \vec{x}_i unter allen $\vec{x}_1, \dots, \vec{x}_n$ haben. Wird ein konstanter Wert $k = k_1 = \dots, k_n$ verwandt und die Nullhypothese abgelehnt, liefern die knn Teilmengen mit den höchsten Werten von D_i Informationen darüber, wo die Anpassung des Computerexperiments schlecht ist. Zu beachten ist, dass die k_i nächsten Nachbarn aufgrund der standardisierten \vec{x} -Werte berechnet werden sollten, da ansonsten die k_i nächsten Nachbarn nur von Einflussgrößen mit einer großen Spannbreite bestimmt werden. Diese Version des Tests wird als $x - knn$ Test bezeichnet.

Oftmals liegt bei Computerexperimenten eine sehr große Anzahl an Einflussgrößen vor mit der Einschränkung, dass nur begrenzt viele Versuchsläufe möglich sind. In dieser Situation ist es zwar technisch möglich, die k nächsten Nachbarn bzgl. der Einflussgrößen zu berechnen. Jedoch ist dies aufgrund des Fluchs der hohen Dimension nicht unbedingt sinnvoll (siehe z.B. Hastie et al. (2001) Kapitel 2.5). Deswegen sollen die Beobachtungen nicht bzgl. der Einflussvariablen gruppiert werden sondern entsprechend der $\tilde{\mu}(\vec{x})$ -Werte. Hat die unbekannte Funktion $\tilde{\mu}$ ein monotonen Verhalten, so werden ähnliche $\tilde{\mu}$ -Werte ähnliche \vec{x} -Werte haben. Deswegen besteht die Vermutung, dass das Gruppieren entsprechend den $\tilde{\mu}$ -Werten in einer hohen Güte für den Test resultiert. Diese Version des Tests wird als $y - knn$ Test bezeichnet. Werden auch hier wieder gleichgroße Nachbarschaften für jeden Datenpunkt gewählt, liefern auch hier wieder die D_i mit hohen Werten Informationen darüber, wo das Computerexperiment nicht

die richtige Vorhersage liefert.

Eine dritte Art der Berechnung der Teilmengen ist durch den am Anfang dieses Kapitels vorgestellten Datensatzes motiviert. Der Datensatz beruht auf einem 3^2 Versuchsplan, für den bei den echten Experimenten unterschiedlich viele Wiederholungen pro Versuchspunkt vorhanden sind. In diesem Fall sind knn Regeln nicht sinnvoll, da die k_i nächsten Nachbarn eines Punktes \vec{x}_i nicht unbedingt eindeutig definiert sind. Hier macht es sich positiv bemerkbar, dass die Teilmengen s_i flexibel gewählt werden können. Anstatt knn Regeln sollen alle die Datenpunkte \vec{x}_j in s_i vereint werden, die einen Abstand kleiner als eine Konstante $c > 0$ zu dem Punkt \vec{x}_i haben:

$$s_i := \{j \in \{1, \dots, n\} : \|\vec{x}_j - \vec{x}_i\| < c\}. \quad (6.3)$$

Diese Variante soll *RepPerm*-Test genannt werden, da er auch bei Wiederholungen in den echten Experimenten eindeutig definiert ist. Bei den vorhandenen Daten (3^2 Versuchsplan) ergeben sich als natürliche Kandidaten für c folgende Werte: Wird $c \leq 1$ gewählt, so besteht s_i nur aus den Punkten mit dem gleichen \vec{x} Wert. Für $c \in (1, \sqrt{2}]$ besteht s_i aus den Punkten, die maximal den Abstand 1 zu \vec{x}_i haben.

Im allgemeinen können die Teilmengen s_i auf jede erdenkliche Art und Weise konstruiert werden. Die Güte des Tests gegen bestimmte Alternativen kann damit beeinflusst werden. Gibt es Vorwissen über das Verhalten von $\mu(\vec{x})$ oder $\tilde{\mu}(\vec{x})$, kann dies über die Wahl der Teilmengen s_i mit in den Test eingebaut werden. Ebenso können bestimmte Eigenschaften der Daten wie z.B. der Versuchsplan mit in die Wahl der Teilmengen einfließen.

6.2.3 Simulationsstudie und Anwendung

Latin Hypercube Beispiele

Zur Demonstration der Eigenschaften der $x - knn$ Variante und der $y - knn$ Variante des Permutationstests wird ein 4-dimensionales Beispiel verwandt, das von Lehman et al. (2004) zur Illustration eines sequentiellen Algorithmuses zum Auffinden von robusten Konstellationen von Einflussgrößen bei Computerexperimenten genutzt wird. Die zugrunde liegende Funktion ist

$$\mu(\vec{x}) := \mu(x_1, x_2, x_3, x_4) := \frac{1}{30}\psi(x_1, x_2)\psi(x_3, x_4) + (x_1 - \pi)^2$$

$$\text{mit } \psi(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10.$$

Die Funktion $\psi(x_1, x_2)$, definiert auf dem Quadrat $[-5, 10] \times [0, 15]$, wird oft als Branin-funktion bezeichnet. Das folgende Regressionmodell wird angenommen:

$$Y(x_1, x_2, x_3, x_4) = \mu(x_1, x_2, x_3, x_4) + \varepsilon(x_2),$$

wobei $\varepsilon(x_2) \sim \text{N}(0, (3x_2 + 10)^2)$. Drei unterschiedliche Vorhersagefunktionen werden verwandt, wobei die erste ($\tilde{\mu}_1(\vec{x})$) der Nullhypothese entspricht und die beiden Vorhersagefunktionen $\tilde{\mu}_2(\vec{x})$ und $\tilde{\mu}_3(\vec{x})$ zwei Situationen widerspiegeln, bei denen die Alternative korrekt ist.

$$\begin{aligned} \tilde{\mu}_1(\vec{x}) &= \mu(\vec{x}) && \text{Nullhypothese } H_0 \\ \tilde{\mu}_2(\vec{x}) &= \mu(\vec{x})(1 + 0.007x_3x_4^{1/8}) && \text{Alternative } H_1 \\ \tilde{\mu}_3(\vec{x}) &= \mu(\vec{x})(1 + 0.007(x_2 - 6)^3) && \text{Alternative } H_2 \end{aligned}$$

Ausserdem werden zwei verschiedene Arten von Versuchsplänen verwandt um zu überprüfen, ob die betrachteten Tests von dem Versuchsplan abhängen.

Von besonderem Interesse sind hier simplex basierte Latin Hypercubes, die in Kapitel 3.2 vorgestellt wurden und zufällige Latin Hypercubes (Kapitel 3.1.1). Auf beide Arten

von Versuchsplänen wurden alle drei Funktionen $\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\mu}_3$ angewandt. Die resultierenden Ergebnisse für die 6 Kombinationsmöglichkeiten sind in Tabelle 5.2 zusammengefasst. Für jede Kombination wurden 1000 Datensätze mit $n = 40$ Beobachtungspunkten erzeugt und alle drei Tests (Good's, $x - knn$, $y - knn$) durchgeführt. Dabei wurde bei den zufälligen Latin Hypercubes der Versuchsplan für jeden Durchlauf der Simulation zufällig neu generiert, für den simplex basierten Latin Hypercube wurde jedes mal der gleiche Versuchsplan, erzeugt nach den Methoden in Kapitel 3.1, verwandt. Da die Gesamtzahl an möglichen Permutationen ($\#G = 2^{40}$) zu groß für eine komplette Auswertung ist, wurde für die Tests $x - knn$ und $y - knn$ eine Zufallsauswahl von 1000 Permutationen betrachtet. Für die beiden Tests $x - knn$ und $y - knn$ wurde ausserdem die Größe der Nachbarschaft für alle Punkte \vec{x}_i auf $k_i = 8$ festgelegt. In Tabelle 6.2 sind die Simulationsergebnisse für die Signifikanzniveaus $\alpha = 0.1, 0.05, 0.01$ zusammengefasst. Es sind die prozentualen Anteile der durchgeführten Tests dargestellt, die bei den 1000 generierten Datensätzen die Nullhypothese abgelehnt haben, unter der jeweiligen Situation. Der Anteil der simulierten Datensätze, bei denen die Nullhypothese abgelehnt wurden, ist für alle Tests nahe dem vorgegebenen Signifikanzniveau. Für beide Alternativen, H_1 und H_2 zeigen sich keine Unterschiede zwischen den Ergebnissen für zufällige Latin Hypercubes und simplex basierten Latin Hypercubes. Folglich scheint die Güte der Tests nicht durch die beiden hier betrachteten Versuchspläne beeinflusst zu werden. Für Alternative H_1 liefert der $x - knn$ -Test die höchste Güte in der Simulation, wobei Good's Test ein wenig besser abschneidet als der $y - knn$ -Test. Für die Alternative H_2 zeigt wieder der $x - knn$ -Test die höchste Güte, hier hat jedoch Good's Test eine sehr geringe Güte.

3² vollfaktorielle Versuchspläne

Die Eigenschaften des Permutationstests sollen auch anhand eines Beispiels ähnlich zu dem vorgestellten Hutprofil-Datensatz evaluiert werden. Für die echten Experimente

		α	Good	$x - knn$	$y - knn$
H_0	Simplex	0.1	0.079	0.109	0.078
		0.05	0.030	0.045	0.038
		0.01	0.002	0.008	0.004
	LHD	0.1	0.082	0.093	0.099
		0.05	0.040	0.046	0.044
		0.01	0.002	0.006	0.008
H_1	Simplex	0.1	0.523	0.716	0.379
		0.05	0.374	0.598	0.247
		0.01	0.090	0.297	0.074
	LHD	0.1	0.426	0.633	0.284
		0.05	0.275	0.478	0.186
		0.01	0.065	0.184	0.066
H_2	Simplex	0.1	0.274	0.682	0.682
		0.05	0.270	0.708	0.679
		0.01	0.113	0.397	0.332
	LHD	0.1	0.197	0.760	0.418
		0.05	0.068	0.530	0.257
		0.01	0.004	0.137	0.061

Tabelle 6.2: Simulationsergebnisse für 6 unterschiedliche Datensituationen für die $x - knn$ und $y - knn$ -Tests anhand von unterschiedlichen Latin Hypercubes. Dargestellt ist der Anteil der simulierten Datensätze, für den die Tests die Hypothese abgelehnt haben.

sollen dabei an jedem Punkt eines (vollständigen) 3^2 Versuchsplans unterschiedlich viele Läufe entsprechend Tabelle 6.3 zur Verfügung stehen.

Dafür werden Werte der *echten* Experimente mittels eines quadratisches Modells mit

Lauf	x_1	x_2	Wiederholungen
1	-1	-1	4
2	-1	0	5
3	-1	1	7
4	0	-1	4
5	0	0	4
6	0	1	7
7	1	-1	4
8	1	0	4
9	1	1	3

Tabelle 6.3: Einstellungen und Wiederholungen für den 3^2 Versuchsplan der Rückfederungsexperimente.

Wechselwirkung erzeugt:

$$Y(\vec{x}) = \mu(\vec{x}) + \epsilon(x_1), \quad (6.4)$$

mit

$$\mu(\vec{x}) = \mu(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3x_1x_2 - 0.6x_1 + \pi, \quad (6.5)$$

wobei π die Kreiszahl ist und $\epsilon(x_1) \sim \text{N}(0, 0.1 + (0.01(x_1 + 1.3)))$. Es wurden zwei unterschiedliche Vorhersagefunktionen verwandt:

$$\begin{aligned} \tilde{\mu}_1(\vec{x}) &= x_1^2 + 2x_2^2 - 0.3x_1x_2 - 0.6x_1 + \pi && \text{Nullhypothese } H_0, \\ \tilde{\mu}_2(\vec{x}) &= 1.05x_1^2 + 1.92x_2^2 - 0.4x_1x_2 - 0.4x_1 + \pi && \text{Alternative } H_1. \end{aligned}$$

Für beide Vorhersagefunktionen wurden 2000 Stichproben aus dem betrachteten Regressionsmodell simuliert. Auf die Datensätze wurde Good's Test und der *RepPerm*-Test mit $c = 0.1$ und $c = 1.01$ angewandt, wobei 2000 zufällige Permutationen gezogen wurden, die zur Bestimmung des p-Wertes der Tests benutzt wurden. In der Anwendung auf einen Datensatz im folgenden Kapitel werden 50000 zufällige Anordnungen

verwandt, um einen größeren Teil aller möglicher Vorzeichenwechsel abzudecken, hier werden jedoch nur 2000 gewählt, um die Laufzeit der Simulation nicht zu stark zu erhöhen. In Tabelle 6.4 sind die Ergebnisse der Simulationen dargestellt. Unter der Nullhypothese ist der Anteil an simulierten Datensätzen, für die die Nullhypothese abgelehnt wurde, für alle Tests sehr nahe an dem Signifikanzniveau. Es ist deutlich erkennbar, dass die beiden lokalen Permutationstests der Variante *RepPerm* mit Konstanten $c = 0.1, 1.01$ eine viel höhere Güte haben als der Test von Good ohne Lokalisierung.

	α	Good	<i>RepPerm</i> , $c = 0.1$	<i>RepPerm</i> , $c = 1.01$
H_0	0.1	0.0900	0.0915	0.1055
	0.05	0.0460	0.0470	0.0505
	0.01	0.0085	0.0115	0.0120
H_1	0.1	0.3400	0.880	0.936
	0.05	0.1905	0.746	0.850
	0.01	0.0430	0.464	0.536

Tabelle 6.4: Simulationsergebnisse für die Datensituation eines 3^2 Versuchsplans mit unterschiedlich vielen Wiederholungen. Angewandt wurde der *RepPerm* Test mit den Konstanten $c = 0.1$ und $c = 1.01$ und der Permutationstest nach Good.

Anwendungsdatensatz

Wird der Permutationstest auf den Hutprofil-Datensatz aus Kapitel 6.2 angewandt, ergeben sich die Ergebnisse in Tabelle 6.5. Durch die drei unterschiedlichen Simulationsmodelle mit dem Flanschenwinkel θ_2 als Zielgröße und den beiden interessierenden Einflussgrößen ergeben sich drei Testsituationen, bei denen jeweils die Hypothese „Die Computersimulationen modellieren den Erwartungswert der echten Experimente“ überprüft wird. Für die Permutationstests *RepPerm* wurden jeweils 50000 zufälliger Anordnungen der Vorzeichen zur Berechnung der Testergebnisse verwandt. Für die

Anwendung ist interessant, dass sämtliche Tests für jede der Testsituationen eindeutig zum 5% Niveau ablehnen, was heisst, dass keines der drei Computerexperimente als guter Ersatz für die echten Experimente angesehen werden kann. Dies dient als Motivation für eine Methodik, die lokal die beste Simulation findet. Insgesamt hat sich somit

Zielgröße	Simulation	Good	<i>RepPerm</i> , $c = 0.1$	<i>RepPerm</i> , $c = 1.01$
Flanschwinkel	S1	< 0.0001	0.0089	< 0.0001
	S2	0.0095	0.0209	0.0026
	S3	0.0003	0.0412	0.0013

Tabelle 6.5: p-Werte für die Permutationstests der Rückfederungsdaten.

gezeigt, dass die vorgeschlagenen Permutationstests nützliche Hilfsmittel sind um zu beurteilen, ob ein Computerexperiment eine korrekte Vorhersage für echte Experimente liefert. Sie verbessern einen einfachen Test, der in Good (2000) beschrieben ist durch wiederholtes lokales Anwenden des Tests. Ein weiterer Vorteil der Tests ist, dass sie über unterschiedliche Formulierungen von Nachbarschaftsumgebungen Vorwissen über die Experimente mit einbinden können oder auf bestimmte Datensituationen abgestimmt werden können, wie anhand eines Datensatzes gezeigt werden konnte. Die Anwendung auf einen Datensatz aus der Blechumformung ergibt, dass die dort verwandten Modelle nicht als präziser Ersatz für die echten Experimente angesehen werden können.

6.3 Lokale Auswahl von Computerexperimenten

Die Problematik des oben beschriebenen Datensatzes, dass für ein reales Phänomen mehrere Computersimulationen existieren, ist nicht neu, sondern tritt in verschiedenen wissenschaftlichen Bereichen auf. In der Literatur existieren für solche Situationen bereits Methoden, die verschiedene Vorhersagen miteinander kombinieren können. Dabei wird jedoch meistens angenommen, dass die Simulationen unterschiedliche Präzision haben und vorab bekannt ist, welches die genaueste Simulation ist. Verbunden mit der Annahme, dass die genauere Simulation aufwendiger ist und somit für diese nur wenig Läufe zur Verfügung stehen, jedoch zusätzlich noch viele Läufe von einer weniger präzisen Simulation verfügbar sind, sollen diese beiden Datenquellen miteinander verbunden werden (siehe z.B. Kennedy und O'Hagan (2000), Qian und Wu (2008) und Qian et al. (2006)). Solch eine Datensituation kann beim numerischen Lösen von partiellen Differentialgleichungen entstehen. Im Allgemeinen gilt bei Finite Elemente Simulationen, dass eine feinere Elementformulierung in einer besser approximativen Lösung der partiellen Differentialgleichungen resultieren sollte, was jedoch zugleich einen höheren Rechenbedarf mit sich zieht. Prinzipiell wäre ein solches Vorgehen auch für die hier vorliegenden Computersimulationen interessant.

Jedoch ist bei den zu behandelnden Experimenten keine sichere Vorabaussage möglich, welche Computersimulation global am Besten ist. Vielmehr spiegeln die drei unterschiedlichen Simulationen verschiedene Modellierungsstrategien für die echten Experimente wider, bei denen sich nicht aus technischen Gründen die Überlegenheit eines Simulationsmodells ergibt. Da ausserdem die Permutationstests aus dem letzten Kapitel zu dem Ergebnis kamen, dass keine der vorhandenen Simulationen ein wirklicher globaler Ersatz für die echten Experimente ist, soll hier eine andere Fragestellung bearbeitet werden: Es wird eine Entscheidungsregel gesucht, die lokal entscheidet, welches Computerexperiment (lokal) die beste Vorhersage liefert. Dadurch wird die implizite Annahme der Ansätze von Kennedy und O'Hagan (2000), Qian und Wu (2008) und Qian et al. (2006), dass eine Simulation für den gesamten Versuchsraum am „Besten“

ist, umgangen. Abhängig von den Ergebnissen des im Folgenden beschriebenen Ansatzes kann die Annahme, dass eine Simulation global die Beste, bestätigt oder verworfen werden.

Zur Beschreibung des Ansatzes wird zuerst die Notation festgelegt, dann werden Modellierungen für die echten Experimente und die Computerexperimente gewählt und eine quadratische Verlustfunktion auf die Differenzen zwischen dem Modell der echten Daten und den Modellen für die Computerexperimente angewandt.

Für die echten Experimente wird das folgende Modell angenommen:

$$Y(\vec{x}) = \mu(\vec{x}) + \varepsilon, \quad \vec{x} \in D \subset \mathbb{R}^d, \quad (6.6)$$

wobei $Y(\vec{x})$ eine eindimensionale Zielvariable ist, $d \in \mathbb{N}$ und $\mu(\vec{x})$ eine Funktion ist, die zumindest stetig ist. Die Zufallsvariable ε hat hier den Erwartungswert Null und Varianz σ^2 . Für das Regressionsmodell (6.6) seien die Zufallsvariablen $Y_1 := Y(\vec{x}_1), \dots, Y_n := Y(\vec{x}_n)$, $\vec{x}_i \in D, i = 1, \dots, n$ stochastisch unabhängig mit Beobachtungen y_1, \dots, y_n vorhanden. Ein Schätzer für $\mu(\vec{x})$ mit Konfidenzintervall wird mit $\hat{\mu}(\vec{x})$ und $(lw(\vec{x}), up(\vec{x}))$ bezeichnet. Es wird angenommen, dass es $K \geq 2$ unterschiedliche Computersimulationen gibt, deren zugrunde liegenden Funktionen mit

$$\tilde{\mu}^1(\vec{z}), \dots, \tilde{\mu}^K(\vec{z}) \quad (6.7)$$

bezeichnet werden. Die Computersimulationen sollen ebenfalls eindimensionalen Output haben und den gleichen Versuchsraum D wie die echten Experimente. Für die Computersimulationen seien die Beobachtungen $\tilde{\mu}^1(\vec{z}_1), \dots, \tilde{\mu}^K(\vec{z}_1), \dots, \tilde{\mu}^1(\vec{z}_m), \dots, \tilde{\mu}^K(\vec{z}_m)$ mit $\vec{z}_i \in D, i = 1, \dots, m$, vorliegend. Zur besseren Unterscheidung werden die Einflussparameter der Computerexperimente hier mit \vec{z} bezeichnet. Dies kann insbesondere sinnvoll sein, wenn für die echten Experimente und die Computerexperimente unterschiedliche Versuchspläne vorliegen. Für die echten Experimente können dies z.B. D-optimale Versuchspläne sein (Atkinson und Donev (1992)), für die Computerexperimente hingegen einer der in Kapitel 3 beschriebenen Versuchspläne. Die hier vorgestellte Methodik

funktioniert auch, wenn für alle K Computerexperimente unterschiedlichen Versuchspläne vorliegen würden. Diese Verallgemeinerung wird jedoch nicht betrachtet, da dies die Notation unnötigerweise kompliziert machen würde.

Die hier beschriebene Methodik ist flexibel in dem Sinne, dass unterschiedliche Schätzmethoden für die echten Experimente und die Computerexperimente verwandt werden können.

Für die echten Experimente wurde ein Regressionsmodell an die Beobachtungen angepasst, dass linear in den Koeffizienten β ist, d.h. es wird angenommen, dass $\mu(x)$ die Form

$$\mu(\vec{x}) = f(\vec{x})^T \beta, \quad (6.8)$$

hat, wobei $f(\vec{x}) = [f(\vec{x})_1, \dots, f(\vec{x})_p]$ die Regressorvektoren sind. Die Regressorvektoren $f(\vec{x}_1), \dots, f(\vec{x}_n)$ können zu der Regressormatrix $F = [f(\vec{x}_1), \dots, f(\vec{x}_n)]^T \sim (n \times p)$ zusammengefasst werden. Die Vorhersage $\hat{\mu}(\vec{x})$ ist dann gegeben durch

$$\hat{\mu}(\vec{x}) = f(\vec{x})^T \hat{\beta}, \quad (6.9)$$

wobei $\hat{\beta}$ die kleinste Quadrate Schätzung für β ist:

$$\hat{\beta} = (F^T F)^{-1} F^T y. \quad (6.10)$$

Ein $1 - \alpha$ Konfidenzintervall $(l(\vec{x}), u(\vec{x}))$ für den Erwartungswert $\mu(\vec{x})$ ist dann gegeben durch

$$(l(\vec{x}), u(\vec{x})) := \hat{\mu}(\vec{x}) \pm t_{\alpha/2, n-p} \sqrt{\hat{\sigma}^2 f(\vec{x})^T (F^T F)^{-1} f(\vec{x})}, \quad (6.11)$$

wobei $t_{\alpha, k}$ das α -Quantil der t -Verteilung mit k Freiheitsgraden ist und $\hat{\sigma}^2 = (y - F\hat{\beta})^T (y - F\hat{\beta}) / (n - p)$, d.h. der Schätzer für die Standardabweichung der Fehler im linearen Regressionmodell. Für weitere Details des linearen Regressionsmodells sei auf Myers und Montgomery (1995) verwiesen. Andere Methoden zur Berechnung einer Vorhersagefunktion sind z.B. Kernregression oder Splineregression. Je nach Ziel der

Analyse kann auch anstatt eines Konfidenzintervalls für den unbekanntem Mittelwert ein Konfidenzintervall für die Zufallsvariablen selbst verwandt werden. Letzteres ist überlicherweise viel weiter, da es die Streuung der Einzelbeobachtungen berücksichtigt. Hier wird jedoch ein Konfidenzintervall für den Mittelwert verwandt, da dies den Anspruch an die Computerexperimente widerspiegelt, dass die Computerexperimente die Mittelwertfunktion der echten Experimente ersetzen können.

Die Nutzung eines Metamodells für die Computerexperimente ist dadurch motiviert, dass es meistens rechentechnisch nicht möglich ist, die dem Computerexperiment zugrunde liegende Computersimulation beliebig oft auszuwerten. Ist die Berechnung der Simulationen so günstig, dass kein Metamodell notwendig ist, kann die Simulation direkt genutzt werden. Dies ist jedoch selten der Fall, weswegen an die Computerexperimente ein Metamodell angepasst wird um Vorhersagen für Punkte des Versuchsraums zu machen, an denen keine Beobachtungen vorliegen. Hier soll die in Kapitel 5 vorgestellte Kerninterpolation als Metamodell verwandt werden, deren Vorhersagefunktion in Formel (5.1) beschrieben ist.

Andere mögliche Metamodelle sind z.B. Krigingmodelle (Gaußprozessmodelle, Santner et al. (2003), Kapitel 4.1) oder Interpolation mittels radialen Basisfunktionen (Buhmann (2000), Kapitel 4.2).

6.3.1 Anwendung

Wie schon beschrieben, besteht der vorliegende Hutprofil-Datensatz aus der Rückfederungsanalyse aus einem unvollständigen 3^2 vollfaktoriellen Versuchsplan. Für jeden Punkt im Versuchsraum gibt es zwischen 3 und 12 Wiederholungen. Für den Zielwert θ_2 wurde ein quadratisches Modell mit Wechselwirkung angepasst. Eine Zusammenfassung der Ergebnisse des linearen Modells ist in Tabelle 6.6 gegeben. Der Standardfehler der Residuen beträgt 0.699, das adjustierte Bestimmtheitsmaß 0.98. Eine Analyse der Residuenplots zeigt keine Besonderheiten. Als Vorhersagefunktion $\hat{\mu}(\vec{x})$ wird das voll-

ständige Modell verwandt, d.h. inklusive aller nichtsignifikanter Terme.

Effekt	Schätzung	Std. Fehler	p-Wert
Intercept	27.490	0.187	< .0001
WR	-3.078	0.184	< .0001
ZT	5.273	0.184	< .0001
WR ²	0.602	0.239	0.0169
ZT ²	0.235	0.239	0.3339
WR*ZT	-1.571	0.273	< .0001

Tabelle 6.6: Ergebnisse für die kleinste Quadrate Modelle der echten Experimente. WR steht für den Werkzeugradius, ZT für die Ziehtiefe.

Die Daten der Computerexperimente sind für den vollständigen 3² Versuchsplan gegeben, der für jede der drei Computersimulationen durchgeführt wurde. Diese Art Versuchsplan ist für Computerexperimente eher ungewöhnlich, wurde jedoch verwandt, um Beobachtungen an den gleichen Stellen im Versuchsraum für die echten und die simulierten Experimente zu haben.

6.3.2 Aufteilung des Versuchsraums

Das allgemeine Ziel des nachfolgend beschriebenen Ansatzes ist es, lokal das beste der K unterschiedlichen Simulationsmodelle auszusuchen. Um unterschiedliche Computerexperimente für einen festen Punkt $\vec{x} \in D$ zu vergleichen wird der quadrierte Abstand zwischen der Vorhersage für die echten Experimente $\hat{\mu}(x)$ und den Computerexperimenten (oder den daraus resultierenden Metamodellen) verwendet:

$$\Delta_k(\vec{x}) := (\tilde{\mu}^k(\vec{x}) - \hat{\mu}(\vec{x}))^2, \quad (6.12)$$

$$\Delta(\vec{x}) := (\Delta_1(\vec{x}), \dots, \Delta_K(\vec{x})). \quad (6.13)$$

Ein weiterer Schritt kann sein, $\Delta(\vec{x})$ zu skalieren:

$$\delta(\vec{x}) := \frac{1}{\widehat{\text{var}}(\hat{\mu}(\vec{x}))} \Delta(\vec{x}). \quad (6.14)$$

Der Vorteil des Skalierens ist, dass $\delta(\vec{x})$ einheitenfrei ist und deswegen für unterschiedliche Datensätze/ Versuche verglichen werden kann. Von Nachteil ist, dass unterschiedliche Kombinationen von $\widehat{\text{var}}(\hat{\mu}(\vec{x}))$ und $\Delta(\vec{x})$ zum gleichen $\delta(\vec{x})$ führen können. Die Entscheidungsregel, welche Computersimulation für die Vorhersage genutzt werden soll ist dann naheliegend: Nutze die Computersimulation k , die $\Delta_k(\vec{x})$ minimiert:

$$k^*(\vec{x}) := \operatorname{argmin}_{k=1}^K \Delta_k(\vec{x}). \quad (6.15)$$

Wenn der Schätzer $\hat{\mu}(\vec{x})$ konsistent ist für $\mu(\vec{x})$ für alle $\vec{x} \in D$, dann sind die quadrierten Differenzen $\Delta_k(x)$ ebenso konsistent für $(\tilde{\mu}^k(\vec{x}) - \mu(\vec{x}))^2$ entsprechend dem „continuous mapping“ Theorem (van der Vaart (1998)).

Wenn nur eine Entscheidung bezüglich einem Punkt $\vec{x} \in D$ benötigt wird, ist es ausreichend, $k^*(\vec{x})$ zu berechnen. Um zu einem weiteren Verständnis der Computerexperimente im Vergleich mit den echten Experimenten zu gelangen, ist hier von Interesse, den Versuchsraum so in Bereiche aufzuteilen, das in einem Bereich jeweils ein Simulationsmodell am besten ist. Dies kann Rückschlüssen zulassen, ob für unterschiedliche Kombinationen von Einflussgrößen unterschiedliche Simulationen als adäquat betrachtet werden müssen. Der Versuchsraum wird entsprechend zweier Kriterien aufgeteilt: Zuerst kann der Versuchsraum aufgeteilt werden in die Bereiche, in denen eine Computersimulation existiert, die innerhalb des Vorhersageintervalls für den Mittelwert liegt bzw. in Bereiche, in denen keine Simulation vorliegt, die innerhalb des Vorhersageintervalls für den Erwartungswert $\mu(\vec{x})$ liegt. Liegt keine Computersimulation innerhalb des Vorhersageintervalls für den Mittelwert, so kann auch die beste Simulation nur als Annäherung an die echten Experimente gesehen werden:

$$D_0 := \{\vec{x} \in D \mid \exists k : \tilde{\mu}^k(\vec{x}) \in (l(\vec{x}), u(\vec{x}))\}, \quad (6.16)$$

$$D_1 := \{\vec{x} \in D \mid \forall k : \tilde{\mu}^k(\vec{x}) \notin (l(\vec{x}), u(\vec{x}))\}. \quad (6.17)$$

Zweites kann der Versuchsraum entsprechend der (geschätzten) besten Computersimulation in Bezug auf $k^*(\vec{x})$ aufgeteilt werden:

$$D^k := \{\vec{x} \in D | k = k^*(\vec{x})\}. \quad (6.18)$$

Beide Kriterien können miteinander kombiniert werden:

$$D_0^k := \{\vec{x} \in D | k = k^*(\vec{x}) \text{ und } \tilde{\mu}^k(\vec{x}) \in (l(\vec{x}), u(\vec{x}))\}, \quad (6.19)$$

$$D_1^k := \{\vec{x} \in D | k = k^*(\vec{x}) \text{ und } \tilde{\mu}^k(\vec{x}) \notin (l(\vec{x}), u(\vec{x}))\}. \quad (6.20)$$

In einer perfekten Situation gibt es ein $k \in \{1, \dots, K\}$ mit $D_0^k = D$, d.h. die gleiche Computersimulation ist immer die beste und diese liegt immer innerhalb des Vorhersageintervalls für die Mittelwert von $\mu(\vec{x})$.

Eine weitere Möglichkeit zur Aufteilung des Versuchsraum wäre, Unsicherheitsvorhersagen des Metamodells mitzunutzen. Dies wird hier aus zwei Gründen nicht gemacht: Erstens macht es bei dem hier betrachteten Datensatz keinen Unterschied, da sowohl die Vorhersageintervalle für das Krigingmodell als auch für das Kerninterpolationsmodell sehr eng um die Vorhersage liegen im Vergleich zur Streuung der Daten aus den echten Experimenten. Zweitens würde es in einer komplizierteren Notation und Interpretation resultieren.

6.3.3 Anwendung und Interpretation

Wird die zuvor beschriebene Methodik zur Aufteilung des Versuchsraums auf den Rückfederungsdatensatz angewandt, ergibt sich Abbildung 6.4. Dabei zeigen die Zahlen an, welches Computerexperiment als bestes angesehen wird. Rote Bereiche zeigen an, dass die beste Vorhersage ausserhalb des Vorhersageintervalls für $\mu(\vec{x})$ liegt, wohingegen graue Bereiche anzeigen, dass die Vorhersage der besten Computersimulation innerhalb des Konfidenzbandes für den unbekanntem Mittelwert liegt. Die Stellen, an denen Beobachtungen der echten Experimente vorliegen sind mit \bullet gekennzeichnet. In großen

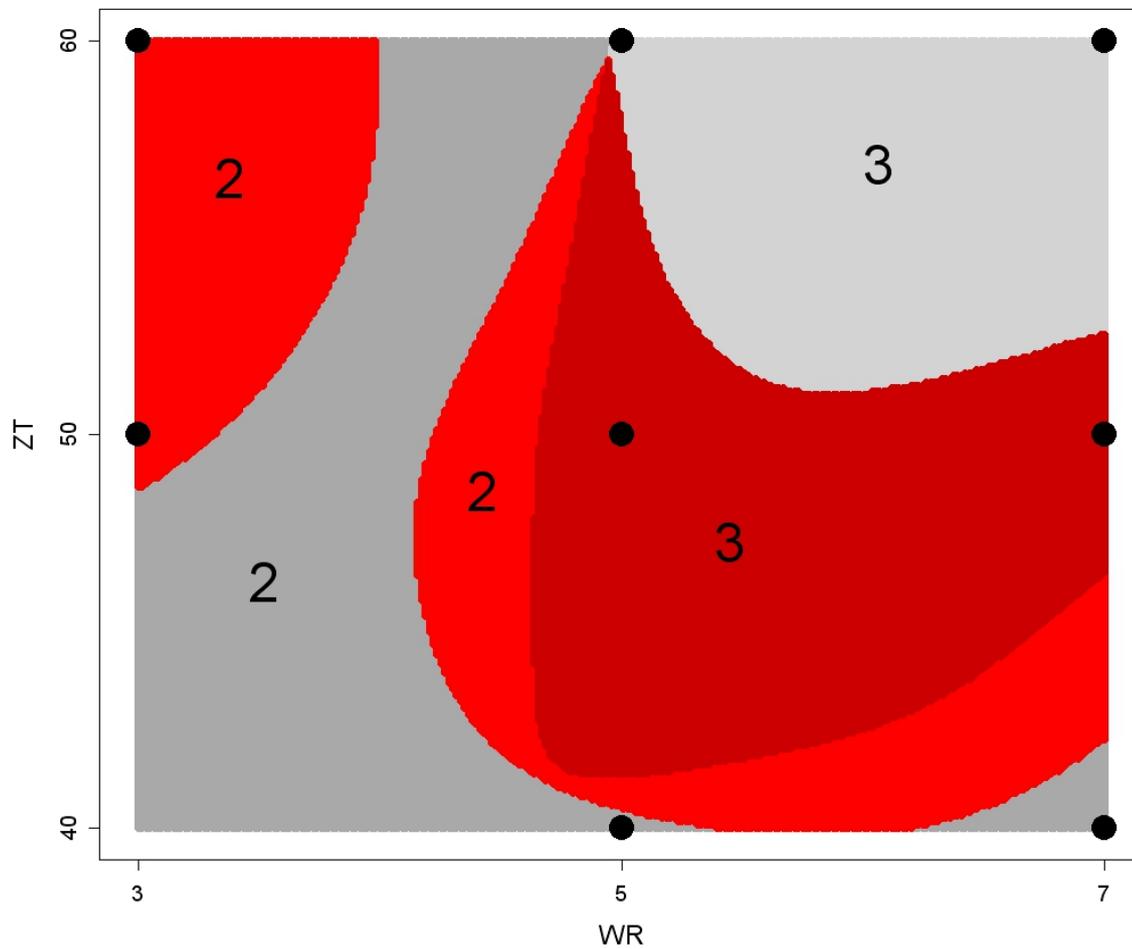


Abbildung 6.4: Ergebnisse der Aufteilung des Versuchsraums für die Zielgröße Flanschwinkel.

Teilen des Versuchsraums gibt es demnach eine korrekte Simulation zur Vorhersage der echten Experimente. Die beste Simulation ist dabei entweder Simulation S_2 oder Simulation S_3 . Simulation S_1 wird nirgendwo als bestes Simulationsmodell erkannt.

KAPITEL 7

FAZIT

In der vorliegenden Arbeit wurden Methoden für die Planung, Modellierung und Validierung von Computerexperimenten erarbeitet. Bei der Methodenauswahl für Computerexperimente muss beachtet werden, dass Computerexperimente deterministisch sind und deswegen viele sonst übliche Methoden der Statistik keine guten Ergebnisse erzielen.

Ein wesentlicher Aspekt für die erfolgreiche Anwendung von Computerexperimenten ist ein guter Versuchsplan, der vorgibt, an welchen Stellen im Versuchsraum Simulationsläufe plaziert werden. Da bei Computerexperimenten in den meisten Fällen kein Zufallsfehler vorliegt und nur wenige Läufe der Simulation zur Verfügung stehen, ist die Anforderung an den Versuchsplan, den Versuchsraum mit den wenigen vorhandenen Läufen möglichst gleichmässig abzudecken. Vorhandene Methoden suchen dafür meistens eine Klasse von Versuchsplänen, z.B. die Klasse der Latin Hypercubes, aus und optimieren numerisch ein Kriterium, das die Güte der Versuchspläne beschreibt in der ausgewählten Klasse von Versuchsplänen. Hier wurde ein anderer Ansatz verfolgt, da die Optimierung eines Versuchsplans in höheren Dimensionen sehr anspruchsvoll sein kann. Dafür wurde eine Überdeckung des Versuchsraum mit Simplexen konstruiert, die möglichst gleichmässige Simplexe haben soll und die Ecken der Simplexe werden als Versuchspunkte verwandt (NRSM Algorithmus). Um gute Projektionseigenschaften zu erhalten, wird anschliessend die simpliziale Überdeckung so rotiert, dass die eindimen-

sionalen Verteilungen der Ecken gut verteilt sind. Ein Vorteil der Methode ist, dass für eine gegebene Dimension d die konstruierten Überdeckungen mehrfach für die Konstruktion von Versuchsplänen unterschiedlicher Größe verwandt werden können. Bis auf die Optimierung der Rotation kommt der Ansatz ohne numerische Optimierung aus. Die resultierenden Versuchspläne sind sehr konkurrenzfähig zu herkömmlich konstruierten Maximinversuchsplänen. Ein Nachteil ist jedoch, dass die Überdeckungen mit Simplexen nicht in beliebig hohen Dimensionen konstruiert werden können, da z.B. für $d = 10$ die Anzahl an Simplexen in einer Triangulation so schnell steigt, dass diese nicht mehr handhabbar sind.

Wurden für ein Computerexperiment Läufe entsprechend einem Versuchsplan erhoben, ist es wünschenswert, aus den vorhandenen Läufen eine Vorhersagefunktion (Metamodell) für das Computerexperiment zu konstruieren, die für das Computerexperiment Vorhersagen machen kann an Stellen, an denen kein Lauf zur Verfügung steht. Da für Computerexperimente keine Zufallsfehler vorliegen, ist eine Anforderung an die Metamodelle, dass die entsprechend dem Versuchsplan gemachten Beobachtungen interpoliert werden. Übliche Ansätze zur Interpolation sind z.B. Kriging oder die Interpolation mit radialen Basisfunktionen. Im Gegensatz dazu wurde eine neue Methode (Kerninterpolation) vorgestellt, die lokal angepasste Modelle gewichtet mittelt und einen engen Bezug zum Inverse Distance Weighting hat. Die Kerninterpolation liefert insbesondere für Situationen mit kleinen Stichproben gute Vorhersagewerte im Vergleich zum Kriging, da sie nicht auf das Schätzen von Parametern angewiesen ist, was besonders bei kleinen Stichproben zu Problemen bei Krigingmodellen führen kann. Ein weiterer Vorteil ist, dass keine Invertierung von großen Matrizen notwendig ist wie es beim Kriging oder der Interpolation mit radialen Basisfunktionen notwendig ist. Außerdem liegt ebenso wie beim Kriging eine Unsicherheitsschätzung für die Vorhersage vor, die zwar nicht durch ein stochastisches Modell motiviert ist, jedoch ähnlich interpretiert werden kann. Solch eine Unsicherheitsvorhersage liegt z.B. für die Interpolation mit radialen Basisfunktionen bisher nicht vor. Genauso wie bei den neu vorgeschlagenen

Versuchsplänen beschränkt jedoch die Größe der Triangulation die Anwendung der Kerninterpolation auf Datensätze mit kleiner bis mittlerer Dimension.

Ein wesentlicher Punkt im Kontext von Computerexperimenten ist, zu validieren, ob das Computerexperiment eine gute Vorhersage der zugrundeliegenden echten Experimente liefert. Dafür wurde ein Permutationstest konstruiert, der die Hypothese, dass die Simulation des Computerexperiments die echten Experimente im Erwartungswert für alle Positionen des Versuchsraums korrekt vorhersagt, überprüft. Der Test hält dabei auch bei kleinen Stichprobengrößen das Niveau ein und ist auch unter Heteroskedasität valide. Als Voraussetzung muss lediglich gegeben sein, dass für alle \vec{x} im Versuchsraum der Fehler der echten Experimente symmetrisch um den Erwartungswert streut. Die Anwendung des Permutationstests auf einen Datensatz, bei dem mehrere verschiedene Computerexperimente für ein echtes Experiment vorliegen, ergab, dass keines der Computerexperimente eine korrekte Vorhersage liefert. Motiviert durch die Beobachtung, dass es unter den verschiedenen Computerexperimenten keins gibt, das für den gesamten Versuchsraum eine korrekte Vorhersage liefert, wurde eine Methodik vorgestellt, die lokal für einen Punkt x des Versuchsraums entscheidet, welches Computerexperiment am besten die echten Experimente vorhersagt.

Die vorgeschlagenen Methoden zeichnen sich insgesamt durch eine hohe Güte aus. Allerdings gibt es noch viele Verbesserungsmöglichkeiten, die verfolgt werden können: Der NRSM Algorithmus sollte nach Möglichkeit so abgeändert werden, dass auch die Konstruktion von Versuchsplänen in hohen Dimensionen ermöglicht wird. Dafür können mögliche Herangehensweisen sein, die Triangulation nur lokal zu betrachten oder den Algorithmus zu parallelisieren. Da auch die Kerninterpolation Triangulationen verwendet und somit eine Anwendung in hohen Dimensionen mit großen Stichprobenumfängen praktisch unmöglich wird, ist auch hier nach Alternativen zu suchen. Ein weiterer interessanter Aspekt für die Kerninterpolation ist alternative Gewichtsfunktionen zu untersuchen, die z.B. nur einen lokalen Träger erlauben.

LITERATURVERZEICHNIS

- Abrahamsen, P. (1997). A review of Gaussian random fields and correlation functions. Technical report 917, Norwegian Computing Center, Box 114, Blindern, N0314 Oslo, Norway.
- Abramowitz, M. und I. Stegun (1972). *Handbook of mathematical functions*. Washington: National Bureau of Standards.
- Allen, T., M. Bernshteyn, und K. Kabiri-Bamoradian (2003). Constructing Meta-Models for Computer Experiments. *Journal of Quality Technology* 35(3), 264–274.
- Atkinson, A. und A. Donev (1992). *Optimum Experimental Designs*. Oxford: Oxford University Press.
- Barber, C., D. Dobkin, und H. Huhdanpaa (1996). The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software* 22(4), 469 – 483.
- Bates, R., B. Giglio, und H. Wynn (2003). A Global Selection Procedure for Polynomial Interpolators. *Technometrics* 45(3), 246–255.
- Bates, R., H. Maruri-Aguilar, und H. Wynn (2008). Smooth supersaturated models. Technical report, Department of Statistics, London School of Economics, London, UK.
- Bayarri, M., J. Berger, und D. Higdon (2002). A Framework for Validation of Computer Models. Technical report, National Institute of Statistical Sciences.

- Bayarri, M., J. Berger, R. Paulo, und J. Sacks (2007). A Framework for Validation of Computer Models. *Technometrics* 49(2), 138–154.
- Buhmann, M. (2008). *Radial Basis Functions*. Cambridge monographs on applied and computational mathematics. Cambridge: Cambridge University Press.
- Buhmann, M. D. (2000). Radial basis functions. In A. Iserles (Ed.), *Acta Numerica*, pp. 1 – 38. Cambridge: Cambridge University Press.
- Chen, V., K.-L. Tsui, und R. Barton (2006). A review on design, modeling and applications of computer experiments. *IIE Transactions* 38, 273 – 291.
- Cox, D., J. Park, und C. Singer (2001). A statistical method for tuning a computer code to a data base. *Computational Statistics & Data Analysis* 37, 77–92.
- Coxeter, H. (1963). *Regular Polytopes*. New York: The Macmillan Company.
- Cressie, N. (1993). *Statistics for spatial data*. Wiley series in probability and mathematical statistics : Applied probability and statistics. New York: Wiley.
- Dancik, G. M. (2010). *mleqp: Maximum Likelihood Estimates of Gaussian Processes*. R package version 3.1.1.
- Diggle, P. (2009). *Model based geostatistics*. Springer series in statistics. New York: Springer.
- Fang, K., D. Lin, P. Winker, und Y. Zhang (2000). Uniform Design: Theory and Application. *Technometrics* 42(.), 237–248.
- Fang, K., Y. Tang, und J. Yin (2005). Lower bounds for wrap-around L_2 -discrepancy and construction of symmetrical uniform designs. *Journal of Complexity* 21, 757–771.
- Fang, K.-T., R. Li, und A. Sudjianto (2006). *Design and Modeling for Computer Experiments*. Computer Science and Data Analysis Series. Boca Raton: Chapman & Hall/CRC.

- Fang, K.-T. und C.-X. Ma (2001). Wrap-Around L_2 -Discrepancy of Random Sampling, Latin Hypercubes and Uniform Designs. *Journal of Complexity* 17, 608–624.
- Fortune, S. (2004). Voronoi Diagrams and Delaunay Triangulations. In J. Goodman (Ed.), *Handbook of discrete and computational geometry*, pp. 513 –528. Boca Raton: Chapman & Hall/CRC.
- Franke, R. (1982). Scattered Data Interpolation: Tests of Some Methods. *Mathematics of Computation* 38(157), 181–200.
- Franke, R. und G. Nielson (1980). Smooth interpolation of large sets of scattered data. *International Journal for numerical methods in Engineering* 15, 1691–1704.
- Goel, T., R. Hafkta, und W. Shyy (2009). Comparing error estimation measures for polynomial and kriging approximation of noise free functions. *Structural and Multidisciplinary Optimization* 38(5), 429 – 442.
- Good, P. (2000). *Permutation Tests - A practical Guide to Resampling Methods for Testing Hypothesis*. New York: Springer.
- Grosso, A., A. Jamali, und M. Locatelli (2009). Finding maximim latin hypercube designs by Iterated Local Search heuristics. *European Journal of Operational Rresearch* 197, 541 – 547.
- Gösling, M. (2010). *Metamodell unterstützte Simulation und Kompensation von Rückfederungen in der Blechumformung*. Dissertation, Technische Universität Dortmund.
- Gösling, M., H. Kracker, A. Brosius, S. Kuhnt, und A. Tekkaya (2009). Simulation und Kompensation rückfederungsbedingter Formabweichungen. In W. Tillmann (Ed.), *SFB 708 - 3. öffentliches Kolloquium*, pp. 155 –170. Dortmund: Verlag Praxiswissen.
- Han, G., T. Santner, W. Notz, und D. Bartel (2009). Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables. *Technometrics* 51(3), 278 – 288.

- Hastie, T., R. Tibshirani, und J. Friedman (2001). *The Elements of Statistical Learning*. New York: Springer.
- Hickernell, F. (1998a). A Generalized Discrepancy and Quadrature Error Bound. *Mathematics of Computation* 67(221), 299 – 322.
- Hickernell, F. (1998b). Lattice rules: how well do they measure up? In P. Hellekalek und G. Larcher (Eds.), *Random and Quasi-Random Point Sets*, pp. 106 – 166. Berlin / New York: Springer Verlag.
- Hidgon, D., M. Kennedy, J. Cavendish, J. Cafeo, und R. Ryne (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal of Scientific Computing* 26(2), 448–466.
- Higdon, D., M. Kennedy, und J. Cavendish (2004). Combining Field Data And Computer Simulations For Calibration And Prediction. *Journal on Scientific Computing* 26(2), 448–466.
- Johnson, M., L. Moore, und D. Ylvisaker (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* 26, 131–148.
- Jones, D. R., M. Schonlau, und W. Welch (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13, 455–492.
- Kennedy, M. und A. O’Hagan (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika* 87(1), 1–13.
- Kennedy, M. und A. O’Hagan (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Association B63*(3), 425–464.
- Klee, V. (1980). On the complexity of d-dimensional Voronoi diagrams. *Archiv der Mathematik* 34, 75 – 80.
- Kuipers, L. und H. Niederreiter (2006). *Uniform Distribution of Sequences*. Mineola, New York: Dover.

- Lawson, C. (1976). C^1 compatible Interpolation over a triangle. Technical report, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.
- Lehman, J., T. Santner, und W. Notz (2004). Designing computer experiments to determine robust control variables. *Statistica Sinica* 14, 571–590.
- Lehmann, E. und J. Romano (2005). *Testing Statistical Hypothesis*. Springer Texts in Statistics. New York: Springer.
- Lemieux, C. (2009). *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer Series in Statistics. New York: Springer.
- McKay, M., R. Beckman, und W. Conover (1979). A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* 21(2), 239–245.
- McLain, D. (1974). Drawing contours from arbitrary data points. *The Computer Journal* 17(4), 318–324.
- Meyer, C. (2000). *Matrix Analysis and Applied Linear Algebra*. Philadelphia: SIAM.
- Mühlenstädt, T. (2007). *Evaluation eines Permutationstests*. Diplomarbeit, Fachbereich Statistik, Universität Dortmund.
- Mühlenstädt, T. (2008). Simplex based space filling designs. Technical report, Faculty of Statistics, Technische Universität Dortmund, Dortmund, Germany.
- Mühlenstädt, T. und U. Gather (2008). Permutation Tests for Validating Computer Experiments. In B. Schipp und W. Krämer (Eds.), *Statistical Inference, Econometric Analysis and Matrix Algebra. Festschrift in Honour of Götz Trenkler*, pp. 77–84. Heidelberg: Physica-Verlag.
- Mühlenstädt, T. und S. Kuhnt (2009). Comparing different interpolation methods on two-dimensional test functions. In *Proceedings of the ENBIS-EMSE, St. Etienne*.

- Micula, G. (2002). A variational approach to spline functions theory. *General Mathematics* 10(1), 21–50.
- Morris, M. und T. Mitchell (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* 43, 381–402.
- Morris, M., T. Mitchell, und D. Ylvisaker (1993). Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction. *Technometrics* 35(3), 243 – 255.
- Myers, R. und D. Montgomery (1995). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York: Wiley.
- Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. CBMS-NSF regional conference series in applied mathematics. Philadelphia: Society for Industrial and Applied Mathematics.
- Okabe, A., B. Boots, K. Sugihara, und S. Chiu (2000). *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics. Chichester: Wiley.
- Powell, M. (2001). Radial basis function methods for interpolation to functions of many variables. Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, England.
- Powell, M. (2005). Five Lectures on Radial Basis Functions. Technical report, Technical University of Denmark.
- Qian, P. und J. Wu (2008). Bayesian Hierarchical modeling for integrating Low-Accuracy and High-Accuracy Experiments. *Technometrics* 50(2), 192–204.
- Qian, Z., C. Seepersad, V. Joseph, J. Allen, und C. Wu (2006). Building Surrogate Models Based on Detailed and Approximate Simulations. *Journal of mechanical design* 128, 668–677.

- R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.
- Rajan, V. (1994). Optimality of the Delaunay Triangulation in \mathbb{R}^d . *Discrete & Computational Geometry* 12, 189–202.
- Reese, C., A. Wilson, M. Hamada, H. Martz, und K. Ryan (2004). Integrated Analysis of Computer and Physical Experiments. *Technometrics* 46(2), 153–164.
- Renka, R. (1988). Multivariate Interpolation of large sets of scattered data. *ACM Transactions on mathematical Software* 14(2), 139 – 148.
- Ripley, B. (2004). *Spatial Statistics*. Wiley series in probability and statistics. New York: Wiley.
- Rippa, S. (1990). Minimal roughness property of the Delaunay triangulation. *Computer Aided Geometric Design* 7(6), 489–497.
- Sacks, J., S. Schiller, und W. Welch (1989). Designs for Copmputer Experiments. *Technometrics* 31(1), 41–47.
- Sacks, J., W. Welch, T. Mitchell, und H. Wynn (1989). Design and Analysis of Computer Experiments. *Statistical Science* 4, 409–435.
- Santner, T., B. Williams, und W. Notz (2003). *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. New York: Springer Verlag.
- Shepard, D. (1968). A two dimensional interpolation function for irregularly-spaced data. In *Proc. 23rd Nat. Conf. ACM*, pp. 517 – 523.
- Shewry, M. und H. Wynn (1987). Maximum entropy sampling. *Journal of Applied Statistics* 14(2), 165 – 170.

- van Dam, E. (2005). Two-dimensional Minimax Latin Hypercube Designs. Technical report, Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands.
- van Dam, E., B. Husslage, D. den Hertog, und H. Melissen (2005). Maximin Latin hypercube designs in two dimensions. Technical report, Department of Econometrics and Operations Research, Tilburg University, Tilburg, The Netherlands.
- van der Vaart, A. (1998). *Asymptotic statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press.

ANHANG A

SYMBOLVERZEICHNIS

Symbol	Beschreibung	Seite
$P, P_n \in \mathbb{R}^{n \times d}$	Punktemenge mit n Elementen im \mathbb{R}^d	5
$T(P)$	Delaunaytriangulation einer Punktemenge P	6
$\ \cdot\ $	L_2 Norm eines Vektors ($= \ \cdot\ _2$)	5
\mathfrak{D}	Versuchsraum	9
$\vec{x} = [x_1, \dots, x_d]^T$, $\vec{x}_i = [x_{i,1}, \dots, x_{i,d}]^T$	Punkte im \mathbb{R}^d	5
\vec{X}	Zufallsvektor	10
$D, D_{n,d}$	Versuchsplan mit n Läufen und d Einflussvariablen	9
$\mathfrak{D}_{n,d}$	Menge aller Versuchspläne mit n Läufen und d Variablen aus dem Versuchsraum \mathfrak{D}	9
$Mm(D_{n,d})$	Maximinkriterium eines Versuchsplan $D_{n,d}$	11
$D_{n,d}^{Mm}$	optimaler Maximinversuchsplan	11
$\Phi_p(D)$	Maximinkriterium nach Morris und Mitchell (1995)	12
$D_{n,d}^{\Phi_p}$	optimaler Versuchsplan entsprechend dem Φ_p -Kriterium	12
$SD(D)$	Star Diskrepanz eines Versuchsplan	13
$V(f)$	Variation der Funktion f im Sinne von Hardy und Krause	14
$WD(D)$	Warp Around Diskrepanz des Versuchsplan D	15
$D_{n,d}^{WD}$	optimaler Versuchsplan bzgl. der Wrap Around Diskrepanz	15
$mM(D)$	Minimaxkriterium eines Versuchsplan D	16
NRSM	Nearly Regular Simplex Algorithmus	19
$VP(\vec{x}_1, \dots, \vec{x}_{d-1})$	Kreuzprodukt der d -dimensionalen Punkte $\vec{x}_1, \dots, \vec{x}_{d-1}$	24

$R(\theta)$	Rotationsmatrix	30
$C^0(\mathfrak{D})$	Funktionsraum aller stetiger Funktionen über den Versuchsraum \mathfrak{D}	39
$Y(\vec{x}), Z(\vec{x})$	Gaußprozesse	40
$R_{\vec{\theta}(\vec{h})}$	Korrelationsfunktion	41
$\phi(d)$	radiale Basisfunktion	47
\mathbb{P}_m^d	Funktionsraum aller Polynome vom Grad m in d Variablen	48
$y^j(\vec{x})$	Lokal angepasste lineare Funktion	58
$g_j(\vec{x})$	Gewichtsfunktion für die Kerninterpolation	58
$D_j(\vec{x})$	Abstandsfunktion für die Kerninterpolation	58
$w_{h,j}(d)$	Kernfunktion für die Kerninterpolation	58
$U(\hat{y})(\vec{x})$	Unsicherheitsschätzung für die Kerninterpolation	67
RMSE	Root mean squared error	69
WR	Werkzeugradius	128
ZT	Ziehtiefe	128
$\Phi(\vec{y})$	Permutationstest	112

ANHANG B

BERECHNUNG DER UMKUGEL

Okabe et al. (2000), S. 80, geben die Berechnung des Umkreises eines Dreiecks in 2 Dimensionen an, wobei der Umkreis den Mittelpunkt \vec{c} und den Radius r hat. Der Mittelpunkt $\vec{c} = (c_1, c_2)^T$ kann aus der folgenden Gleichung berechnet werden:

$$\begin{vmatrix} \sum_{i=1}^2 c_i^2 & c_1 & c_2 & 1 \\ \sum_{i=1}^2 x_{0,i}^2 & x_{1,1} & x_{0,2} & 1 \\ \sum_{i=1}^2 x_{1,i}^2 & x_{2,1} & x_{1,2} & 1 \\ \sum_{i=1}^2 x_{2,i}^2 & x_{3,1} & x_{2,2} & 1 \end{vmatrix} = 0. \quad (\text{B.1})$$

Hier sind $\vec{x}_j = (x_{j,1}, x_{j,2})^T, j = 0, 1, 2$ die Koordinaten der Ecken des Dreiecks. Diese Gleichung kann leicht in höhere Dimensionen $d > 2$ verallgemeinert werden:

$$\begin{vmatrix} \sum_{i=1}^d c_i^2 & c_1 & \dots & c_d & 1 \\ \sum_{i=1}^d x_{0,i}^2 & x_{0,1} & \dots & x_{0,d} & 1 \\ \vdots & & & & \vdots \\ \sum_{i=1}^d x_{d,i}^2 & x_{d,1} & \dots & x_{d,d} & 1 \end{vmatrix} = 0, \quad (\text{B.2})$$

wobei $\vec{x}_j = (x_{j,1}, \dots, x_{j,d})^T, j = 0, \dots, d$ die Koordinaten der Ecken des Simplex sind und $(c_1, \dots, c_d)^T$ die Koordinaten des Umkugelmittelpunkts. Um die Gleichung in \vec{c} zu

lösen wird der Entwicklungssatz von LaPlace nach der ersten Zeile verwandt. Definiere

$$D_j := (-1)^{2+j} \begin{vmatrix} \sum_{i=1}^d x_{0,i}^2 & x_{0,1} & \dots & x_{0,j-1} & x_{0,j+1} & \dots & x_{0,d} & 1 \\ \sum_{i=1}^d x_{1,i}^2 & x_{1,1} & \dots & x_{1,j-1} & x_{1,j+1} & \dots & x_{1,d} & 1 \\ \vdots & & & & & & & \vdots \\ \sum_{i=1}^d x_{d,i}^2 & x_{d,1} & \dots & x_{d,j-1} & x_{d,j+1} & \dots & x_{d,d} & 1 \end{vmatrix} \quad (\text{B.3})$$

und

$$\alpha := \begin{vmatrix} x_{0,1} & \dots & x_{0,d} & 1 \\ x_{1,1} & \dots & x_{1,d} & 1 \\ \vdots & & & \vdots \\ x_{d,1} & \dots & x_{d,d} & 1 \end{vmatrix}, \quad \beta := \begin{vmatrix} \sum_{i=1}^d x_{0,i}^2 & x_{0,1} & \dots & x_{0,d} \\ \sum_{i=1}^d x_{1,i}^2 & x_{1,1} & \dots & x_{1,d} \\ \vdots & & & \vdots \\ \sum_{i=1}^d x_{d,i}^2 & x_{d,1} & \dots & x_{d,d} \end{vmatrix}. \quad (\text{B.4})$$

Mit dem Entwicklungssatz erhalten wir für B.1:

$$\sum_{i=1}^d c_i^2 \alpha - \sum_{j=1}^d D_j c_j + \beta = 0. \quad (\text{B.5})$$

Eine Lösung ist gegeben durch $c_j = \frac{D_j}{2\alpha}$. Der Radius r de Umkugel kann durch

$$r = \sqrt{\frac{\sum_{j=1}^k D_j^2 - 4\alpha\beta}{4\alpha^2}} \quad (\text{B.6})$$

berechnet werden oder alternativ einfach durch die Berechnung des euklidischen Abstands zwischen \vec{c} und einer Ecke des Simplex. Die Wahl der Ecke spielt dabei keine Rolle, da der Umkugelmittelpunkt der eindeutig definierte Punkt ist, der den gleichen Abstand zu allen Ecken hat.

Für die zweite Vorschlagsfunktion in Kapitel 3.1 wird der Umkugelmittelpunkt einer Facette eines Simplex benötigt, d.h. für d Ecken $\vec{x}_0, \dots, \vec{x}_{j-1}, \vec{x}_{j+1}, \dots, \vec{x}_d$ des Simplex S mit Ecken $\vec{x}_0, \dots, \vec{x}_d$. Dies kann aus dem Umkugelmittelpunkt des vollständigen Simplex durch orthogonale Projektion von \vec{c} auf die Hyperebene, die durch die Ecken $\vec{x}_0, \dots, \vec{x}_{j-1}, \vec{x}_{j+1}, \dots, \vec{x}_d$ definiert ist, geschehen. Für die Projektion kann die Moore Penrose Inverse genutzt werden.

ANHANG C

DATENSÄTZE AUS KAPITEL 5 ZUR KERNINTERPOLATION

Beobachtung	x_1	x_2	x_3	y
1	200000	0.11	-20	13.03
2	210000	0.11	-20	12.19
3	220000	0.11	-20	11.79
4	200000	0.13	-20	13.17
5	210000	0.13	-20	12.32
6	220000	0.13	-20	11.89
7	200000	0.15	-20	13.31
8	210000	0.15	-20	12.52
9	220000	0.15	-20	12.02
10	200000	0.11	0	13.34
11	210000	0.11	0	12.66
12	220000	0.11	0	12.16
13	200000	0.13	0	13.52
14	210000	0.13	0	12.83
15	220000	0.13	0	12.26
16	200000	0.15	0	13.65
17	210000	0.15	0	13.01
18	220000	0.15	0	12.39
19	200000	0.11	20	14.14
20	210000	0.11	20	13.36
21	220000	0.11	20	12.76
22	200000	0.13	20	14.30
23	210000	0.13	20	13.54
24	220000	0.13	20	12.93
25	200000	0.15	20	14.46
26	210000	0.15	20	13.72
27	220000	0.15	20	13.14

Tabelle C.1: 3^3 vollfaktorieller Versuchsplan für den Autoformdatensatz (DS1(3^3))

Beobachtung	x_1	x_2	x_3	y
1	203900	0.12	-17.28	12.73
2	213000	0.14	19.83	13.45
3	212000	0.14	3.48	12.85
4	215900	0.15	-0.46	12.65
5	211700	0.11	7.56	12.84
6	219800	0.11	3.15	12.33
7	204500	0.12	-7.63	12.95
8	218200	0.13	-17.01	12.00
9	204900	0.12	17.56	13.91
10	217300	0.14	7.72	12.76
11	213300	0.15	-14.19	12.43
12	217100	0.13	15.20	13.06
13	200400	0.11	9.26	13.55
14	202900	0.13	9.93	13.58
15	209700	0.14	-17.76	12.49
16	210100	0.12	0.90	12.78
17	219400	0.13	-10.00	12.00
18	201900	0.15	6.72	13.67
19	201600	0.14	-2.29	13.44
20	200100	0.11	-15.08	12.98
21	208600	0.14	8.88	13.25
22	207500	0.14	-5.32	12.97
23	216600	0.12	18.83	13.02
24	219600	0.11	-4.64	12.00
25	209900	0.13	-10.79	12.49
26	210300	0.15	8.46	13.26
27	212600	0.11	-16.48	12.11

Tabelle C.2: Maximinversuchsplan für den Autoformdatensatz (DS1(Maximin))

Beobachtung	x_1	x_2	x_3	y
1	210000	0.13	0.00	12.83
2	207200	0.14	13.35	13.53
3	210800	0.13	-12.87	12.52
4	202100	0.13	-18.29	12.97
5	205200	0.12	-4.08	12.97
6	205600	0.15	-7.19	13.13
7	214300	0.12	-10.25	12.20
8	208100	0.13	18.03	13.75
9	204200	0.13	6.94	13.37
10	212800	0.13	1.23	12.75
11	200300	0.11	-10.97	12.99
12	205700	0.14	-16.13	12.73
13	214100	0.13	-5.78	12.52
14	208800	0.15	-8.64	12.82
15	217000	0.12	-11.97	12.05
16	217700	0.13	14.93	13.08
17	203800	0.14	-19.28	12.94
18	207900	0.11	5.10	12.92
19	214700	0.13	2.04	12.64
20	218500	0.13	-6.41	12.25
21	217900	0.11	-12.15	11.92
22	200700	0.12	18.58	14.20
23	205100	0.13	-9.25	12.89
24	218300	0.12	-7.52	12.13
25	206200	0.12	16.25	13.60
26	201300	0.15	-11.29	13.24
27	209500	0.14	2.79	12.99
28	215800	0.13	4.35	12.65

Tabelle C.3: Testdatensatz für den Autoformdatensatz (DS1), Teil 1

Beobachtung	x_1	x_2	x_3	y
29	213100	0.12	19.30	13.26
30	213700	0.15	3.87	12.97
31	210500	0.12	0.51	12.74
32	207700	0.14	13.94	13.50
33	213900	0.14	11.81	13.17
34	218800	0.12	13.11	12.82
35	209300	0.14	17.09	13.70
36	204200	0.14	-4.82	13.15
37	209000	0.12	-2.60	12.74
38	211500	0.14	-15.53	12.42
39	210900	0.14	-3.24	12.76
40	206400	0.11	6.11	13.06
41	201000	0.12	-3.70	13.28
42	207200	0.13	8.10	13.25
43	213600	0.12	15.99	13.12
44	211000	0.13	-12.68	12.37
45	201100	0.12	14.71	13.84
46	201700	0.14	1.89	13.56
47	203500	0.12	14.25	13.65
48	214900	0.14	16.43	13.27
49	208300	0.14	-0.32	13.03
50	215200	0.15	-14.67	12.28
51	211300	0.13	-1.61	12.75
52	211900	0.13	17.83	13.47
53	203200	0.12	-18.77	12.80
54	206700	0.12	-3.03	12.89
55	202400	0.12	-1.48	13.19
56	214600	0.14	11.47	13.13

Tabelle C.4: Testdatensatz für den Autoformdatensatz (DS1), Teil 2

Beobachtung	x_1	x_2	x_3	y
57	202800	0.15	0.17	13.48
58	219700	0.13	-0.94	12.31
59	203200	0.13	-8.13	13.11
60	205900	0.12	10.40	13.28
61	216100	0.14	10.71	12.96
62	216700	0.14	-9.14	12.29
63	215400	0.14	4.50	12.80
64	202300	0.12	-15.65	12.89
65	212300	0.14	-13.53	12.37
66	217400	0.14	11.08	12.96
67	204700	0.11	-18.92	12.57
68	215400	0.13	-19.99	12.00
69	218600	0.15	5.75	12.71
70	207000	0.11	12.58	13.35
71	206300	0.13	12.09	13.38
72	216300	0.12	-13.65	12.01
73	219100	0.12	5.43	12.40
74	208800	0.15	-6.26	12.89

Tabelle C.5: Testdatensatz für den Autoformatdatensatz (DS1), Teil 3

Beobachtung	x_1	x_2	x_3	x_4	x_5	y
1	0.08191	0.60000	0.14286	0.30769	0.53623	78.64200
2	0.42628	1.00000	0.85714	0.53846	0.91304	107.05000
3	0.01638	0.37143	0.85714	0.76923	0.13043	29.47000
4	0.19676	0.97143	0.71429	0.30769	0.43478	64.12500
5	0.59010	0.22857	0.71429	0.00000	0.89855	87.08700
6	0.57372	0.11429	0.14286	0.23077	0.36232	40.98500
7	0.62304	0.01429	0.85714	0.15385	0.17391	12.06700
8	0.24590	0.32857	0.57143	0.76923	0.94203	141.04000
9	0.54095	0.31429	1.00000	0.15385	0.02899	4.03380
10	0.45905	0.57143	0.71429	0.69231	0.20290	29.23200
11	0.91809	0.98571	0.00000	0.53846	0.79710	46.55500
12	0.52457	0.84286	0.85714	0.07692	0.31884	34.79300
13	0.49181	0.91429	1.00000	0.69231	0.60870	63.13800
14	0.75410	0.17143	0.14286	0.61538	0.14493	3.59100
15	0.22952	0.81429	0.57143	0.84615	0.05797	26.52600
16	0.73771	0.54286	0.71429	0.23077	0.50725	36.08500
17	0.26229	0.02857	0.71429	0.38462	0.62319	83.75700
18	0.68857	0.12857	0.28571	0.23077	0.66667	62.80200
19	0.09847	0.65714	0.57143	0.38462	0.55072	80.49600
20	0.36075	0.38571	0.71429	0.92308	0.65217	92.64800
21	0.72133	0.47143	0.14286	0.00000	0.24638	20.17100
22	0.04915	0.68571	0.42857	0.38462	0.82609	118.18000
23	0.81962	0.34286	0.85714	0.84615	0.88406	50.28600
24	0.98362	0.18571	0.00000	0.92308	0.71014	26.19100
25	0.47543	0.62857	1.00000	0.07692	0.69565	69.27700
26	0.65580	0.41429	0.28571	0.15385	0.92754	93.00100
27	0.60666	0.00000	0.28571	0.46154	0.46377	47.91700
28	0.50819	0.61429	0.71429	0.53846	0.75362	85.77700

Tabelle C.6: Versuchsplan für den Engine Emmission Datensatz, Teil 1

Beobachtung	x_1	x_2	x_3	x_4	x_5	y
29	1.00000	0.20000	0.42857	0.15385	0.27536	4.48120
30	0.21314	0.28571	0.14286	0.61538	0.81159	122.38000
31	0.37714	0.42857	0.85714	0.30769	1.00000	113.16000
32	0.00000	0.67143	0.14286	0.69231	0.30435	57.12300
33	0.31143	0.05714	0.85714	0.92308	0.95652	128.28000
34	0.70495	0.08571	0.42857	0.07692	0.15942	14.48800
35	0.44266	0.27143	0.47143	0.38462	0.37681	54.72800
36	0.29505	0.74286	1.00000	0.07692	0.78261	85.34400
37	0.06553	0.55714	0.57143	1.00000	0.84058	135.92000
38	0.13123	0.70000	0.28571	0.61538	0.00000	18.13300
39	0.93447	0.40000	0.42857	0.30769	0.63768	37.63700
40	0.86895	0.44286	0.42857	0.46154	0.33333	14.71800
41	0.27867	0.24286	0.42857	0.23077	0.44928	68.93100
42	0.11485	0.51429	0.28571	0.46154	0.57971	85.93900
43	0.83600	0.75714	1.00000	0.92308	0.68116	31.23100
44	0.16400	0.72857	0.00000	0.69231	0.42029	70.11700
45	0.39352	0.45714	0.28571	0.92308	0.10145	25.35000
46	0.03276	0.52857	0.28571	0.76923	0.73913	115.74000
47	0.77048	0.58571	0.00000	0.53846	0.86957	65.61000
48	0.40990	0.48571	0.14286	0.15385	0.11594	23.52500

Tabelle C.7: Versuchsplan für den Engine Emmission Datensatz, Teil 2

Beobachtung	x_1	x_2	x_3	x_4	x_5	y
1	0.23730	0.96970	0.42857	0.61538	0.57143	92.28800
2	0.52541	0.15152	1.00000	0.07692	0.95714	90.51400
3	0.66845	0.81818	0.57143	0.07692	0.97143	97.68900
4	0.91532	0.01515	0.71429	0.92308	0.50000	17.76800
5	0.28811	0.63636	0.57143	0.38462	0.52857	78.21600
6	0.77964	0.80303	0.00000	0.07692	0.12857	5.85100
7	0.67802	0.24242	0.28571	0.76923	0.88571	87.09200
8	0.38991	0.19697	0.71429	0.00000	0.37143	43.73800
9	0.18649	0.21212	1.00000	0.76923	0.25714	39.82200
10	0.32198	0.53030	0.85714	0.23077	0.32857	43.07800
11	0.22036	0.25758	0.71429	0.84615	0.71429	107.93000
12	0.86432	0.16667	0.14286	1.00000	0.47143	21.13000
13	0.47459	0.69697	0.57143	0.61538	0.60000	78.48200
14	0.01694	1.00000	0.28571	0.53846	0.27143	37.97900
15	0.96613	0.84848	0.57143	0.69231	0.64286	33.61900
16	0.37279	0.00000	0.42857	0.84615	0.62857	96.42600
17	0.72883	0.50000	0.57143	0.84615	0.21429	10.36000
18	0.03387	0.36364	0.85714	0.30769	0.28571	38.50700
19	0.55928	0.60606	0.14286	0.84615	0.35714	42.54400
20	0.35585	0.31818	0.85714	0.23077	0.15714	23.37300
21	0.45766	0.62121	1.00000	0.15385	0.08571	11.63300
22	0.25423	0.07576	0.14286	0.23077	0.87143	114.45000
23	0.74577	0.77273	0.28571	0.61538	0.75714	63.05300
24	0.44072	0.37879	0.57143	0.00000	0.84286	97.79700
25	0.40685	0.27273	0.57143	0.61538	0.70000	96.07800
26	0.81351	0.75758	0.71429	0.53846	0.80000	57.25100
27	0.27117	0.95455	0.28571	0.38462	0.34286	56.04900
28	1.00000	0.09091	0.85714	0.38462	0.85714	39.50000

Tabelle C.8: Testdatensatz für den Engine Emission Datensatz, Teil 1

Beobachtung	x_1	x_2	x_3	x_4	x_5	y
29	0.49153	0.54545	0.14286	0.07692	0.38571	49.34200
30	0.79658	0.06061	0.28571	0.23077	0.20000	10.85300
31	0.08468	0.40909	0.71429	0.76923	0.45714	73.30100
32	0.93225	0.04546	0.42857	0.46154	0.44286	20.32600
33	0.59315	0.42424	0.85714	0.15385	0.74286	70.17600
34	0.30504	0.87879	0.85714	0.46154	1.00000	122.70000
35	0.54234	0.57576	0.14286	0.53846	0.58571	67.87500
36	0.11874	0.66667	0.42857	0.92308	0.51429	88.45900
37	0.71189	0.34848	1.00000	0.15385	0.61429	46.50400
38	0.88126	0.33333	0.71429	0.53846	0.65714	37.26600
39	0.00000	0.03030	1.00000	1.00000	0.11429	6.50450
40	0.10162	0.30303	0.14286	0.76923	0.00000	12.69900
41	0.76270	0.51515	0.28571	0.69231	0.41429	30.17300
42	0.13568	0.98485	0.14286	0.30769	0.07143	20.04100
43	0.42378	0.10606	0.28571	0.53846	0.22857	38.14100
44	0.61009	0.56061	0.28571	0.69231	0.30000	31.90400
45	0.83045	0.71212	0.42857	0.30769	0.54286	41.79600
46	0.33892	0.86364	0.28571	0.07692	0.72857	92.81400
47	0.69496	0.18182	0.71429	0.69231	0.40000	31.74400
48	0.15261	0.48485	0.14286	0.30769	0.67143	95.23200
49	0.20342	0.90909	0.57143	0.69231	0.94286	138.25000

Tabelle C.9: Testdatensatz für den Engine Emmission Datensatz, Teil 2

Beobachtung	x_1	x_2	x_3	x_4	x_5	y
1	0.00	0.38	0.13	0.63	0.06	14.30
2	0.75	1.00	0.63	0.06	0.44	11.08
3	0.81	0.06	0.06	0.31	0.63	14.36
4	0.13	0.19	0.31	0.25	0.69	10.01
5	0.38	0.56	0.75	0.19	1.00	11.19
6	0.94	0.25	0.56	0.00	0.25	2.70
7	0.69	0.69	0.19	0.88	0.88	19.55
8	0.44	0.88	0.00	0.44	0.19	19.68
9	0.25	0.31	0.88	1.00	0.50	9.65
10	0.56	0.13	1.00	0.56	0.81	9.63
11	0.31	0.50	0.81	0.13	0.13	7.94
12	0.06	0.75	0.44	0.50	0.75	16.98
13	0.63	0.44	0.25	0.81	0.00	19.51
14	0.88	0.81	0.69	0.75	0.31	22.88
15	0.19	0.94	0.94	0.69	0.38	18.01
16	0.50	0.00	0.38	0.94	0.56	10.20
17	1.00	0.63	0.50	0.38	0.94	17.68

Tabelle C.10: Versuchsplan für den Exhaust Manifold Datensatz (DS3)

Beobachtung	x_1	x_2	x_3	x_4	x_5	y
1	0.00	0.26	0.47	0.21	0.42	8.61
2	0.63	0.42	0.05	0.26	0.11	15.48
3	0.84	0.63	0.53	0.42	0.00	20.72
4	0.68	0.00	0.37	0.68	0.26	14.93
5	0.11	0.79	0.26	0.89	0.16	17.59
6	0.47	0.95	0.21	0.11	0.37	12.54
7	0.58	0.89	0.42	0.95	0.84	19.15
8	0.16	0.11	0.16	0.47	0.89	12.47
9	0.32	0.68	0.00	0.58	0.53	19.32
10	0.21	0.47	0.74	0.00	0.21	3.57
11	0.37	0.16	0.84	0.84	0.05	10.57
12	0.53	0.05	1.00	0.32	0.58	6.19
13	1.00	0.53	0.11	0.79	0.63	21.30
14	0.05	0.58	0.89	0.74	0.74	13.05
15	0.89	0.21	0.32	0.05	0.79	6.30
16	0.95	0.84	0.95	0.53	0.32	21.19
17	0.74	0.74	0.79	0.16	0.95	12.04
18	0.79	0.32	0.68	1.00	0.47	12.53
19	0.26	1.00	0.63	0.37	0.68	18.08
20	0.42	0.37	0.58	0.63	1.00	14.76

Tabelle C.11: Testdatensatz für den Exhaust Manifold Datensatz (DS3)

Beobachtung	x_1	x_2	x_3	x_4	x_5	x_6	y
1	71.00	16.80	21.00	2.00	1.00	0.98	56.75
2	15.00	15.60	21.80	1.00	2.00	1.30	57.65
3	29.00	14.40	25.00	2.00	1.00	1.14	53.97
4	85.00	14.40	21.80	2.00	3.00	0.66	58.77
5	29.00	12.00	21.00	3.00	2.00	0.82	56.34
6	57.00	12.00	23.40	1.00	3.00	0.98	56.85
7	85.00	13.20	24.20	3.00	2.00	1.30	56.68
8	71.00	18.00	25.00	1.00	2.00	0.82	58.45
9	43.00	18.00	22.60	3.00	3.00	1.14	55.50
10	15.00	16.80	24.20	2.00	3.00	0.50	52.77
11	43.00	13.20	22.60	1.00	1.00	0.50	57.36
12	57.00	15.60	23.40	3.00	1.00	0.66	59.64

Tabelle C.12: Versuchsplan für den Kolbensschlagdatensatz (DS4)

Beobachtung	x_1	x_2	x_3	x_4	x_5	x_6	y
1	70.15	13.82	23.91	1.30	1.10	0.74	58.51
2	15.00	14.98	22.37	1.89	2.05	0.52	54.65
3	63.79	16.04	21.16	1.10	1.16	1.19	56.22
4	21.36	16.34	21.32	1.65	2.86	0.90	54.80
5	62.37	15.33	24.96	1.59	1.53	0.70	57.49
6	15.71	14.53	23.34	2.54	2.76	1.07	55.28
7	20.66	12.66	21.85	2.25	2.35	0.56	54.11
8	26.31	14.22	24.52	2.13	2.15	0.95	54.38
9	74.39	16.14	21.08	2.41	2.27	0.73	57.59
10	41.87	16.95	23.18	2.15	1.89	1.23	53.89
11	43.28	13.16	24.80	2.07	2.56	0.65	54.53
12	31.26	15.48	23.63	1.22	2.68	1.21	56.19
13	17.83	13.52	21.61	1.28	2.21	1.17	56.53
14	67.32	14.68	22.13	2.58	2.66	0.61	57.17
15	52.47	14.73	23.51	2.27	2.07	0.98	54.56
16	31.97	15.18	21.24	2.17	1.08	0.66	55.70
17	73.69	12.20	23.71	1.71	2.41	0.52	58.43
18	65.20	15.43	24.07	2.74	1.91	0.56	58.71
19	79.34	14.12	24.76	2.29	1.71	0.59	59.58
20	75.81	14.93	24.23	1.83	3.00	0.84	57.02
21	39.75	13.72	24.03	2.03	2.88	1.18	54.57
22	78.64	15.38	21.69	2.09	2.74	1.03	56.43
23	65.91	14.07	22.33	1.75	2.62	1.15	55.61
24	51.77	13.67	21.20	1.16	2.72	0.57	55.83
25	58.13	12.91	21.04	1.99	2.60	0.82	55.39
26	22.78	16.09	24.92	1.95	2.01	1.11	54.92
27	57.42	15.89	22.94	1.32	1.97	1.08	55.93
28	29.85	12.96	24.19	2.33	1.22	1.15	53.39

Tabelle C.13: Testdatensatz für den Kolbensschlagdatensatz (DS4), Teil 1

Beobachtung	x_1	x_2	x_3	x_4	x_5	x_6	y
29	35.51	15.84	22.45	2.31	2.94	1.28	54.33
30	56.72	12.81	22.70	1.48	1.57	0.60	57.04
31	49.65	12.40	21.44	2.47	2.98	1.09	54.75
32	30.56	14.37	23.10	2.72	2.49	0.75	55.05
33	61.67	13.57	24.35	2.64	2.45	1.11	55.07
34	48.94	16.65	24.11	1.14	1.73	1.06	56.12
35	53.89	14.17	23.42	1.00	1.44	1.13	55.92
36	68.03	16.29	24.64	2.52	1.34	1.27	54.07
37	66.62	13.62	22.25	2.11	1.24	0.86	56.17
38	77.22	15.69	23.87	2.70	2.23	1.16	56.08
39	82.17	16.44	24.47	1.53	2.58	0.99	57.90
40	56.01	15.03	22.01	2.98	2.54	1.25	55.53
41	81.46	13.31	22.54	2.78	2.90	0.69	59.44
42	80.05	16.55	22.17	1.87	1.61	1.12	56.60
43	63.08	12.00	22.41	2.37	1.83	0.97	55.03
44	27.02	13.06	22.62	1.08	2.52	1.01	56.12
45	47.53	13.87	25.00	2.43	2.31	0.87	54.69
46	71.57	14.47	21.36	2.23	1.77	1.22	54.45
47	25.61	13.26	21.12	2.68	1.55	1.24	54.16
48	55.30	17.00	23.75	1.91	2.37	0.90	55.15
49	33.38	14.88	24.84	2.88	1.59	1.02	55.50
50	38.33	14.83	21.00	1.69	2.13	0.98	54.90
51	16.41	12.45	24.72	1.18	1.38	0.88	55.92
52	80.76	13.21	24.15	1.06	2.03	0.79	59.40
53	45.40	16.19	22.74	2.62	1.46	0.94	55.31
54	46.11	16.24	23.26	1.26	2.47	0.67	55.52
55	34.09	15.74	24.60	1.12	2.25	0.54	55.31
56	53.18	15.64	21.73	1.77	1.65	0.76	56.16

Tabelle C.14: Testdatensatz für den Kolbenschlagdatensatz (DS4), Teil 2

Beobachtung	x_1	x_2	x_3	x_4	x_5	x_6	y
57	24.19	15.94	22.05	2.05	1.51	0.83	55.06
58	34.80	12.56	21.40	1.85	1.42	0.73	54.96
59	85.00	14.32	22.78	1.63	2.33	1.29	56.74
60	44.70	14.63	24.39	1.81	1.04	1.20	53.83
61	36.92	12.76	23.95	2.94	2.70	0.93	55.92
62	37.63	13.46	23.30	2.45	1.63	0.50	55.78
63	29.14	14.58	21.97	1.02	1.87	0.69	55.87
64	60.96	15.54	21.53	1.46	2.43	0.86	56.25
65	18.54	16.80	24.31	2.35	2.64	0.72	54.02
66	51.06	15.99	24.68	2.86	2.84	0.77	55.88
67	43.99	15.28	22.66	2.49	2.17	0.80	54.92
68	54.60	12.35	24.43	2.80	1.14	0.63	57.68
69	76.52	12.30	21.57	1.24	1.69	0.92	57.89
70	72.98	16.75	23.46	2.96	1.40	0.81	59.71
71	75.10	13.11	21.93	2.56	1.06	1.00	56.09
72	27.73	16.39	23.99	1.73	1.26	0.58	56.26
73	41.16	12.05	22.21	1.34	2.82	0.77	55.29
74	64.49	13.01	23.38	1.93	2.19	0.71	56.50
75	24.90	14.02	23.55	1.55	2.96	0.62	54.04
76	36.21	16.85	22.58	1.36	1.12	0.96	56.13
77	59.55	13.41	24.56	1.67	1.85	1.03	55.39
78	23.48	12.15	23.83	2.60	1.95	0.68	54.98
79	84.29	15.79	23.06	2.01	1.20	0.65	60.82
80	77.93	14.78	22.49	1.20	1.36	0.55	60.92
81	19.24	13.77	22.82	3.00	1.75	0.85	56.71
82	69.44	12.51	23.02	2.90	2.09	1.19	55.71
83	68.74	15.13	23.14	1.04	2.92	0.94	57.26
84	46.82	12.86	22.09	1.57	1.93	1.26	54.57

Tabelle C.15: Testdatensatz für den Kolbensschlagdatensatz (DS4), Teil 3

Beobachtung	x_1	x_2	x_3	x_4	x_5	x_6	y
85	19.95	15.23	23.79	1.44	1.67	1.28	55.97
86	83.59	13.97	21.28	2.84	1.99	0.89	59.15
87	17.12	15.59	22.29	2.76	1.18	1.04	55.78
88	48.23	14.27	21.65	2.92	1.30	0.53	58.64
89	40.45	16.49	21.48	2.66	1.81	0.61	56.24
90	82.88	12.71	23.59	2.19	1.48	1.07	56.17
91	39.04	14.42	24.27	1.40	1.79	0.82	55.56
92	42.58	13.92	21.81	1.51	1.32	1.10	54.79
93	32.68	12.25	22.90	1.79	2.29	1.05	54.75
94	50.35	12.10	23.22	1.97	1.28	1.30	52.66
95	22.07	13.36	22.98	1.61	1.02	0.78	55.50
96	28.43	16.70	21.77	2.82	2.39	1.14	55.26
97	58.84	15.08	23.67	2.39	1.00	0.91	55.68
98	60.25	16.60	22.86	2.21	2.80	0.51	55.57
99	70.86	16.90	21.89	1.42	2.11	0.64	58.34
100	72.27	12.61	24.88	1.38	2.78	1.24	56.82

Tabelle C.16: Testdatensatz für den Kolbenschlagdatensatz (DS4), Teil 4

ANHANG D

DATENSATZ AUS KAPITEL 6

Lauf Nr.	WR	ZT	S1	S2	S3
1	-1.00	-1.00	17.37	24.63	23.74
2	-1.00	0.00	22.69	30.42	29.46
3	-1.00	1.00	27.65	36.36	35.39
4	0.00	-1.00	17.63	22.93	21.86
5	0.00	0.00	21.55	28.39	26.94
6	0.00	1.00	26.11	33.59	32.48
7	1.00	-1.00	17.39	21.79	19.57
8	1.00	0.00	21.74	26.65	24.16
9	1.00	1.00	25.82	31.28	28.82

Tabelle D.1: Datensatz zur Rückfederung aus Kapitel 6, Computerexperimente

Lauf Nr.	WR	ZT	θ	Lauf Nr.	WR	ZT	θ
1	-1.00	0.00	29.97	21	0.00	0.00	28.52
2	-1.00	0.00	31.07	22	0.00	0.00	26.70
3	-1.00	0.00	30.57	23	0.00	0.00	27.69
4	-1.00	0.00	30.59	24	0.00	0.00	27.45
5	-1.00	0.00	31.45	25	0.00	0.00	26.66
6	-1.00	0.00	31.45	26	0.00	0.00	26.66
7	-1.00	1.00	37.86	27	0.00	0.00	26.45
8	-1.00	1.00	39.39	28	0.00	1.00	32.36
9	-1.00	1.00	39.42	29	0.00	1.00	32.39
10	0.00	-1.00	22.20	30	0.00	1.00	31.92
11	0.00	-1.00	22.83	31	1.00	-1.00	21.13
12	0.00	-1.00	22.73	32	1.00	-1.00	21.08
13	0.00	-1.00	22.82	33	1.00	-1.00	20.90
14	0.00	-1.00	22.81	34	1.00	0.00	25.48
15	0.00	-1.00	22.86	35	1.00	0.00	25.22
16	0.00	0.00	27.74	36	1.00	0.00	25.47
17	0.00	0.00	28.16	37	1.00	1.00	28.73
18	0.00	0.00	27.67	38	1.00	1.00	29.49
19	0.00	0.00	28.42	39	1.00	1.00	29.03
20	0.00	0.00	28.54				

Tabelle D.2: Datensatz zur Rückfederung aus Kapitel 6, echte Experimente