Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

Vergleich von Dimensionsreduktionsmethoden für Surrogate auf Dünnen Gittern

Christopher Schnick

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Dirk Pflüger
Betreuer/in:	Michael Rehme, M.Sc.
Beginn am:	25. April 2019
Beendet am:	25. Oktober 2019

Kurzfassung

Um dem Fluch der Dimensionalität, der bei der Modellierung hochdimensionaler Probleme auftritt, entgegenzuwirken, ist eine Möglichkeit die Verwendung von Dimensionsreduktionsmethoden. Diese haben das Ziel, das orginale Modell in ein geringer-dimensionales Modell umzuwandeln und dabei nur möglichst wenig an Genauigkeit einzubüßen. Dafür gibt es viele unterschiedliche Dimensionsreduktionsmethoden, von denen einige ausgewählte Methoden wie ANOVA, Active Subspaces und die Hauptkomponentenanalyse in dieser Arbeit vorgestellt werden. Anschließend wird deren Qualität durch Anwendung auf verschiedene Problemstellungen untersucht und verglichen.

Inhaltsverzeichnis

1. Einleitung

Bei der Modellierung eines Problems mithilfe uniformer Diskretisierungen, wie z.B. uniforme isotrope Tensorproduktgitter, hängen die benötigte Laufzeit und der notwendige Speicherbedarf exponentiell von der Anzahl der Dimensionen ab. Dies führt dazu, dass bei wachsender Dimensionalität der Probleme die Berechnung zunehmend schwerer oder gar unmöglich wird, daher spricht man vom Fluch der Dimensionalität.

Um diesem Fluch der Dimensionalität entgegenzutreten, wurden viele verschiedene Methoden entwickelt, um die Auswirkungen der Komplexität, welche von der Anzahl der Dimensionen abhängt, auf Speicherbedarf und Laufzeit zu reduzieren. Dazu gehören zum Beispiel Dünne Gitter und adaptive Dünne Gitter, wie sie in [Pfl12] beschrieben werden. Diese Methoden schwächen zwar den Fluch der Dimensionalität etwas ab, stoßen aber auch irgendwann an ihre Grenzen.

Ein weiterer Ansatz ist die Reduzierung der Parameteranzahl des Modells an sich. Um dies zu bewerkstelligen, müssen die Auswirkungen der einzelnen Parameter auf das Modell analysiert und anschließend entschieden werden, welche Parameter aus dem Modell entfernt werden können und wie man diese fehlenden Parameter im orginalen Modell ersetzten könnte. Dieser Ansatz kann auch mit den gerade eben genannten Dünnen Gittern zur weiteren Abschwächung des Fluches der Dimensionalität kombiniert werden.

Ziel dieser Arbeit ist es, die Qualität verschiedener Methoden zur Konstruktion dieser dimensionsreduzierten Funktionen zu untersuchen und zu vergleichen. In dieser Arbeit werden zwei ANOVA-Methoden, Active Subspaces und eine auf der Hauptkomponentenanalyse basierende Methode behandelt. Dafür werden die verschiedenen Methoden erst definiert und dann die Implementation davon in das SG++ Framework, welches in [Pfl10] vorgestellt wird, beschrieben. Anschließend werden die Methoden auf mehrere Problemstellungen angewendet und dabei deren Qualität untersucht und miteinander verglichen. Zum Schluss wird basierend auf den Ergebnissen der verschiedenen Methoden ein Fazit gezogen.

1.1. Grundlagen

Der grundlegende Inhalt dieser Arbeit ist der Vergleich verschiedener Methoden, die aus einer Eingabefunktion mit einer bestimmen Dimension eine geringerdimensionale Funktion konstruieren und dabei versuchen, den Fehler, der dabei entsteht, möglichst gering zu halten. Dieser Sachverhalt wird in diesem Kapitel nun formal definiert.

Sei $D \in \mathbb{N}$ die Dimension der ursprünglichen Eingabefunktion, also die Anzahl der Parameter dieser Funktion. Sei außerdem $d \in \{1, \ldots, D-1\}$ die Dimension der reduzierten Funktion.

Definition 1.1.1
Sei

$$\Omega := [0,1]^D$$

der Definitionsbereich der Modellfunktion und

$$f\colon \Omega \to \mathbb{R}$$

die eigentliche Modellfunktion, welche in ihrer Dimension reduziert werden soll.

Als Definitionsbereich wird hier ohne Beschränkung der Allgemeinheit der D-dimensionale Einheitswürfel verwendet. Im Folgenden wird Ω auch als der ursprüngliche Parameterraum bezeichnet.

Definition 1.1.2
Sei

$$\Omega_R := [0,1]^d$$

der Definitionsbereich der reduzierten Modellfunktion und

$$\eta\colon \Omega_R \to \mathbb{R}$$

die reduzierte Modellfunktion.

Definition 1.1.3
Sei

$$t\colon \Omega \to \Omega_R$$

die Transformationsfunktion.

Die Funktion t überführt Elemente aus dem ursprünglichen Parameterraum Ω in den reduzierten Parameterraum Ω_R.

Die verschiedenen hier vorgestellten Dimensionsreduktionsmethoden haben das Ziel, die Funktionen t und η so zu konstruieren, dass möglichst viele Dimensionen entfernt werden können, d.h. $(D - d)$ zu maximieren, den Fehler aber möglichst gering zu halten, d.h es sollte immer noch folgende Eigenschaft gelten:

$$f(x) \approx \eta(t(x))$$

Um genau zu definieren, ab wann ein Fehler bei der Dimensionsreduktion zu groß ist, wird im folgendem Abschnitt die in dieser Arbeit verwendete Fehlermetrik, die Gesamtvarianz einer Funktion, definiert.

Definition 1.1.4
Sei

$$\langle f, g \rangle := \int_\Omega \langle f(x), g(x) \rangle \, d\mu$$

das Skalarprodukt auf L^2 und

$$\|f\|_{L^2} := \sqrt{\langle f, f \rangle} = \sqrt{\int_\Omega \|f(x)\|^2 \, dx} \tag{1.1}$$

die von dem Skalarprodukt induzierte L^2 Norm. Dann bezeichne

$$\sigma^2(f) := \|f\|_{L^2}^2 = \langle f, f \rangle = \int_\Omega \|f(x)\|_H^2 \, d\mu$$

die quadrierte L^2 Norm. Diese wird im Folgenden auch als Gesamtvarianz der Funktion f bezeichnet.

Definition 1.1.5
Sei

$$e_{rel}(\omega, t, \eta) := \frac{\sigma^2(\omega - \eta \circ t)}{\sigma^2(\omega)}$$

die relative Fehlerfunktion.

Diese gibt an, welcher Anteil der Gesamtvarianz $\sigma^2(f)$ bei der Reduktion verloren gegangen ist. Also ist $e_{rel}(\omega, t, \eta) \in [0, 1]$.

Definition 1.1.6
Sei

$$\sigma^2_{rem}(\omega, t, \eta) := 1 - e_{rel}(\omega, t, \eta)$$

der Anteil der nach der Reduktion verbleibenden Varianz an der Gesamtvarianz.

Somit können wir nun ausdrücken, dass der Fehler, der bei der Dimensionsreduktion anfällt, als akzeptabel angesehen wird, falls der Anteil der verbleibenden Varianz an der Gesamtvarianz größer als ein Mindestwert ist, d.h. $\sigma^2_{rem}(\omega, t, \eta) \geq \sigma^2_{min}$ mit $\sigma^2_{min} \in [0, 1]$.

Zusammenfassend kann also gesagt werden, dass eine Dimensionsreduktionsmethode als Eingabe eine D-dimensionale Funktion f und den akzpetablen relativen Fehler $\sigma^2_{min} \in [0, 1]$ bekommt und daraus dann eine d-dimensionale Funktion η und eine Transformationsfunktion t konstruiert, wobei das d für den Fehler minimal gewählt ist.

Zur Unterscheidung von Skalaren und Vektoren werden Vektoren unterstrichen, d.h. $\underline{a}$ bezeichnet einen Vektor. Außerdem wird der Operator $\leq$ auf Vektoren komponentenweise angewendet, d.h.:

$$\underline{a} \leq \underline{b} \Leftrightarrow a_t \leq b_t, \; 1 \leq t \leq D$$

1.2. Surrogate

In der Praxis ist es oft sehr aufwändig, die Funktion f an bestimmten Punkten auszuwerten, daher arbeiten die hier vorgestellten Methoden primär auf einem Surrogat für die ursprüngliche Eingabefunktion f.

In dieser Arbeit werden dafür primär Dünne Boundary-Gitter mit einer linearen hierarchischen Basis verwendet. Diese werden ausführlich in [Val14] beschrieben und daher werden hier nur die absolut notwendigen Definitionen übernommen, die benötigt werden, um eine Funktion damit zu interpolieren.

Definition 1.2.1
Sei

$$\phi_{l,i}(x) := \begin{cases} 1, & l = -1 \\ 1 - |x2^l - i|, & x \in [2^{-l}(i-1), 2^{-l}(i+1)] \cap [0,1] \\ 0, & sonst \end{cases}$$

eine eindimensionale Hütchenfunktion für das Level l und den Index i.

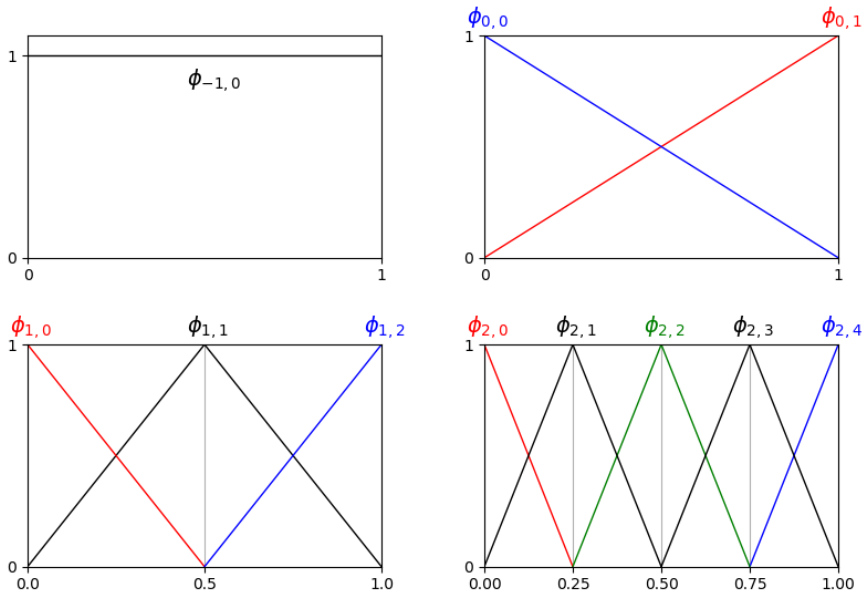

Abbildung 1.1.: Die eindimensionalen Hütchenfunktionen für die Level -1 bis 2. Für Dünne Boundary-Gitter sind nur die Hütchenfunktionen für die Level 0 bis 2 relevant, die spezielle Funktion für Level -1 wird erst später verwendet.

Im Unterschied zu normalen Dünnen Gittern bestizen Dünne Boundary-Gitter einen Levelindex 0, d.h $l \in \mathbb{N}_0$.

Definition 1.2.2
Sei

$$H_{\underline{l}} \coloneqq \left\{ \underline{i} \in \mathbb{N}_0^D \mid \begin{cases} i_t = 1, 3 \ldots, 2^{l_t} - 3, 2^{l_t} - 1, & l_t \geq 1 \\ i_t = 0, 1, & l_t = 0 \end{cases} \right\}$$

die Index-Menge für einen Multi-Level-Index $\underline{l}$ eines Dünnen Boundary-Gitters.

Mithilfe der eindimensionalen Hütchenfunktionen kann nun über den Tensorprodukt-Ansatz eine höherdimensionale Hütchenfunktion definiert werden:

Definition 1.2.3
Sei $\underline{l} = (l_1, \ldots, l_D) \in \mathbb{N}_0^D$ und $\underline{i} \in H_{\underline{l}}$. Dann ist

$$\phi_{\underline{l},\underline{i}} \coloneqq \prod_{d=1}^{D} \phi_{l_d, i_d}$$

die D-dimensionale lineare ANOVA-Ansatzfunktion für den Multi-Level-Index $\underline{l}$ und den Multi-Index $\underline{i}$.

Dafür wird die Funktion f an den Punkten des Dünnen Gitters ausgewertet und anschließend aus den Funktionswerten die sogenannten hierarchischen Überschüsse $\alpha_{\underline{l},\underline{i}}$ wie in [Val14] berechnet. Diese hierarchischen Überschüsse werden dazu verwendet, um die interpolierte Funktion als Linearkombination der Hütchen wie folgt darzustellen:

Definition 1.2.4
Sei l das Level des Dünnen Boundary-Gitters und $g \colon \Omega \to \mathbb{R}$ eine Funktion. Dann bezeichnen wir mit

$$\mathbb{S}_{g,l}^{Boundary}(x) \coloneqq \sum_{|\underline{l}|_1 \leq l} \sum_{\underline{i} \in H_{\underline{l}}} \alpha_{\underline{l},\underline{i}} \, \phi_{\underline{l},\underline{i}}(x)$$

ein Surrogat für die Funktion g, welches mit der linearen hierarchischen Basis auf einem Dünnen Boundary-Gitter mit Level l konstruiert wurde. Außerdem bezeichnen wir mit

$$\omega(x) = \mathbb{S}_{f,l}^{Boundary}(x)$$

das Surrogat für die ursprüngliche Modellfunktion f.

Bei der Interpolation entsteht ein gewisser Fehler, der bei der Fehlerberechnung der reduzierten Modellfunktion gesondert betrachtet werden muss. Der Interpolationsfehler wird bei der späteren Bewertung der Dimensionsreduktionsmethoden ausgeblendet, da er nicht Teil des Dimensionsreduktionsprozesses ist. Daher wird in den folgenden Abschnitten das Surrogat ω und nicht f als Eingabefunktion verwendet.

2. ANOVA-Methoden

ANOVA, kurz für Analysis of Variance, ist ein etabliertes Verfahren der Statistik, um wichtige Variablen eines Modells zu identifizieren, indem der Beitrag jedes Eingabeparameters zur Gesamtvarianz untersucht wird. Die ANOVA-Dekomposition zerlegt eine Modellfunktion $f(x)$ in 2^D viele Summanden, wie folgt:

$$f(x) = f_0 + \sum_{i=1}^{D} f_i(x_i) + \sum_{i<\cdots<j}^{D} f_{i,j}(x_i, x_j) + \cdots + f_{1,\ldots,D}(x_1, \ldots, x_D) \tag{2.1}$$

Dies kann zusammengefasst werden zu:

$$f(x) = f_0 + \sum_{t=1}^{D} \sum_{i_1<\cdots<i_t} f_{i_1,\ldots,i_t}(x_1, \ldots, x_t) \tag{2.2}$$

Hierbei sieht man, dass die Funktion f in $\binom{D}{0} = 1$ 0-dimensionale, also eine konstante Funktion, $\binom{D}{1} = D$ 1-dimensionale Funktionen bis zu $\binom{D}{D} = 1$ D-dimensionale Funktionen zerlegt werden. Insgesamt gibt es also 2^D Funktionsterme. Das Ziel ist schließlich zu analysieren, welche dieser 2^D Terme entfernt werden können, ohne dass dies die Funktion $f(x)$ zu stark beeinflusst.

Definition 2.0.1
Sei

$$C := \{0,1\}^D$$

die Menge aller ANOVA-Komponenten.

Eine ANOVA-Komponente ist ein Element $\underline{c} \in C$, wobei $c_i = 0$ anzeigt, dass der Funktionsterm der Komponente c entlang der i-ten Dimension konstant ist. Falls $c_i = 1$ ist, bedeutet dies, dass der Funktionsterm der Komponente entlang der i-ten Dimension aktiv sind. Für eine Dimension D gibt es ebenfalls $|C| = 2^D$ ANOVA-Komponenten.

Definition 2.0.2
Sei $\underline{c} \in C$ eine ANOVA-Komponente, die entlang der Dimensionen $i_1, \ldots, i_t$ aktiv ist. Dann ist

$$f_{\underline{c}} := f_{i_1,\ldots,i_t},\ i_k \in \{j \in \{1,\ldots,D\} \mid c_j = 1\}$$

der zugehörige Funktionsterm einer ANOVA-Komponente.

Somit gilt also, dass ein Funktionsterm f_c eine $|c|_1$-dimensionale Funktion ist. Werden der Einfachheit halber jedem Funktionsterm alle Eingabeparameter übergeben, selbst wenn diese den Funktionsterm nicht beeinflussen, kann (2.1) auch mithilfe der ANOVA-Komponenten dargestellt werden:

$$f(x) = \sum_{\underline{c} \in C} f_{\underline{c}}(x) \tag{2.3}$$

2.1. Dünne ANOVA-Gitter

Es ist möglich eine abgewandelte Variante der in [Val14] vorgestellten Dünnen Boundary-Gitter zu definieren, indem zusätzlich das Level -1 eingeführt wird, auf dem die Ansatzfunktion konstant ist. Diese Dünnen Gitter werden hier als Dünne ANOVA-Gitter bezeichnet. Mithilfe dieser Art der Dünnen Gitter kann die Eingabefunktion ω komponentenweise wie in (2.3) zerlegt werden.

Definition 2.1.1
Sei

$$\underline{l} \coloneqq (l_1, \ldots, l_D) \in (\mathbb{N}_0 \cup \{-1\})^D$$

ein Multi-Level-Index für Dünne ANOVA-Gitter mit einem Zusatz für das Level -1.

Definition 2.1.2
Sei

$$\tilde{H}_{\underline{l}} \coloneqq \left\{ \underline{i} \in \mathbb{N}_0^D \mid \begin{cases} i_t = 1, 3 \ldots, 2^{l_t} - 3, 2^{l_t} - 1, & l_t \geq 0 \\ i_t = 0, & l_t = -1 \end{cases} \right\} \tag{2.4}$$

die Index-Menge für einen Multi-Level-Index $\underline{l}$ *eines Dünnen ANOVA-Gitters.*

Die Definition der Punktemenge orientiert sich an der von normalen Dünnen Gittern. Jedoch muss bei der Berechnung der Summe des Levelvektors $\underline{l}$ auf jede Komponente 1 addiert werden, da Komponenten des Levelvektors $\underline{l}$ auch -1 sein können und diese negativen Level sich zu einer sehr negativen Levelsumme aufsummieren könnten.

Definition 2.1.3
Sei

$$\begin{aligned} \tilde{x}_{l,i} &\coloneqq i 2^{-max(l,0)} \\ \tilde{x}_{\underline{l},\underline{i}} &\coloneqq (\tilde{x}_{l_1,i_1}, \ldots, \tilde{x}_{l_D,i_D}) \\ \tilde{\Omega}_l &\coloneqq \{\tilde{x}_{\underline{l},\underline{i}} \mid \underline{l} \in \{-1, \ldots, l\}^D,\ |\underline{l} + \underline{1}|_1 \leq l + 1,\ \underline{i} \in \tilde{H}_{\underline{l}}\} \end{aligned} \tag{2.5}$$

die Menge der Gitterpunkte eines Dünnen ANOVA-Gitters mit Level l.

Indem überall eine 1 addiert wird, ist das kleinstmögliche Level, welches ein Dünnes ANOVA-Gitter haben kann -1. Würde nichts addiert, wäre das kleinstmögliche Level eines Dünnen ANOVA-Gitters $-D$. Dies wäre unpraktisch, da das minimale Level von der Dimension abhängen würde.

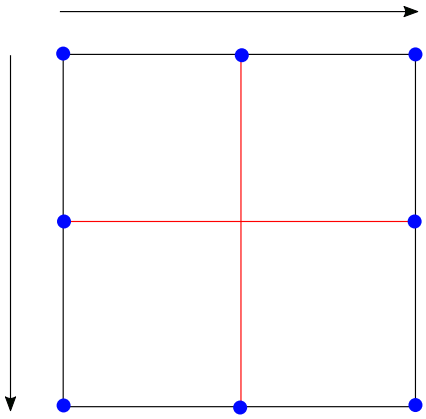

Abbildung 2.1.: Punktemenge eines Dünnen Boundary-Gitters mit $l = 1$

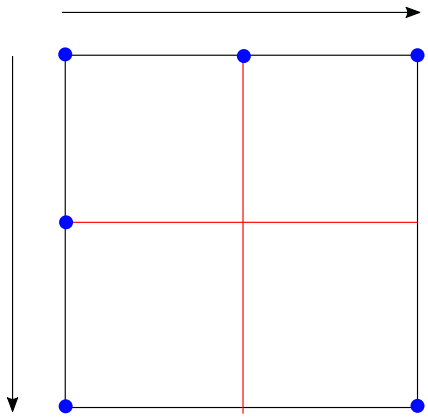

Abbildung 2.2.: Punktemenge eines Dünnen ANOVA-Gitters mit $l = 1$

Wie in Abbildung 2.2 zu sehen ist, unterscheidet sich die Punktemenge im Vergleich zu normalen Dünnen Boundary-Gittern aufgrund der leicht geänderten Level der Gitterpunkte in (2.5) geringfügig.

2.2. Anchored-ANOVA-Dekomposition

Die Anchored-ANOVA-Dekomposition ermöglicht eine einfache Berechnung der einzelnen Funktionsterme im Vergleich zu der klassischen ANOVA-Dekomposition, welche etwas später vorgestellt wird. Dies passiert allerdings auf Kosten der Genauigkeit der Zerlegung. Bei der Anchored ANOVA Methode wird ein beliebiger Punkt $a \in \Omega$ als Ankerpunkt definiert. Anschließend werden die Eingabeparameter x mit dem Anker verschmolzen, indem für jeden Dimensionsindex i, in dem x_i inaktiv ist, d.h. $c_i = 0$ ist, der Ankerwert a_i übernommen wird. Dafür wird die Funktion $g_{a,c}(x)$ verwendet:

$$g_{a,i}(x) := \begin{cases} a, & i = 0 \\ x, & i = 1 \end{cases}$$
$$g_{a,c}(x) := (g_{a_1,c_1}(x_1), \ldots, g_{a_D,c_D}(x_D))^T$$

Für den konstanten Term f_0 wird einfach der Funktionswert an dem Ankerpunkt $\underline{a} \in \Omega$ benutzt:

$$f_0 = f(a)$$

Allgemein gilt für eine beliebige ANOVA-Komponente $\underline{c} \in C$ und einen Anker $\underline{a} \in \Omega$:

$$f_{\underline{c}}(x) = f(g_{a,c}(x)) - \sum_{\underline{c'} \leq \underline{c}} f_{c'}(g_{a,\underline{c'}}(x)) \tag{2.6}$$

Eine Zerlegung dieser Form erfüllt nicht die Bedingung, die für die klassische ANOVA-Dekomposition gelten muss. Die ANOVA-Bedingung wird etwas später behandelt.

Definition 2.2.1
Sei $\tilde{H}_{\underline{l}}$ die Index Menge eines Dünnen ANOVA-Gitters, wie bereits in (2.4) *definiert. Dann ist*

$$W_{\underline{l}}^{h} := span\{\phi_{\underline{l},\underline{i}} \mid \underline{i} \in \tilde{H}_{\underline{l}}\}$$

der hierarchische Unterraum für einen Multi-Level-Index $\underline{l}$ eines Dünnen ANOVA-Gitters mit hierarchischer linearer Basis.

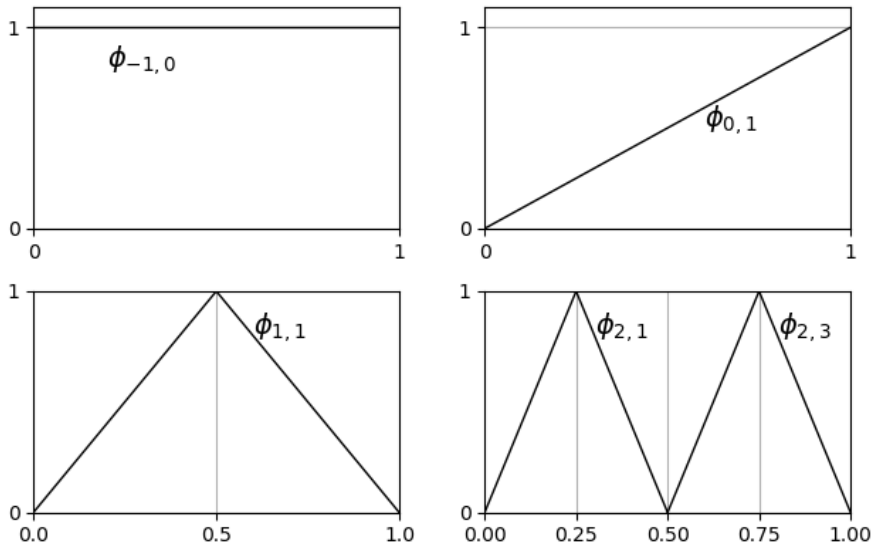

Abbildung 2.3.: Die eindimensionale hierarchische lineare ANOVA-Basis für die Level -1 bis 2

Somit kann jede Funktion $\omega \in W_{\underline{l}}$ als Linearkombination der Ansatzfunktionen ausgedrückt werden.

Definition 2.2.2
Sei $a \in \Omega$ der Anker, l das Level und $g\colon \Omega \to \mathbb{R}$ eine Funktion. Dann bezeichnen wir mit

$$\mathbb{S}_{g,l,a}^{Anchor}(x) := \sum_{|\underline{l}+\underline{1}|_1 \leq l+1} \sum_{\underline{i} \in \tilde{H}_{\underline{l}}} \alpha_{\underline{l},\underline{i}}^{h}\, \phi_{\underline{l},\underline{i}}(x)$$

ein Surrogat für die Funktion g, welches mit der linearen hierarchischen Basis auf einem Dünnen Boundary-Gitter mit Level l konstruiert wurde. Außerdem bezeichnen wir mit

$$\omega^{Anchor}(x) = \mathbb{S}_{f,l,a}^{Anchor}(x)$$

das Surrogat für die ursprüngliche Modellfunktion f mit beliebigen Level und Anker.

Hier wurde o.B.d.A für die Anchored-ANOVA-Zerlegung die hierarchische lineare ANOVA-Basis verwendet. Jedoch kann jede Basis für ein Dünnes Boundary-Gitter, wie z.B. Spline-Basen aus [Val14], durch Hinzufügen einer konstanten Ansatzfunktion $\phi_{-1,0}$ für eine Anchored-ANOVA-Zerlegung verwendet werden.

Die Koeffizienten $\alpha_{m,i}^{h}$ für die verwendete hierarchische lineare ANOVA-Basis, auch hierachische Überschüsse genannt, können dann folgendermaßen berechnet werden:

Definition 2.2.3
Sei $a \in \Omega$ der Anker und $x_{l,i} = i2^{-l}$ und $f_{l,i} = f(g_{a,d}(x_{l,i}))$. Dann sind

$$\begin{aligned} \alpha^h_{-1,0} &= f_{0,0} \\ \alpha^h_{0,1} &= f_{0,1} - f_{0,0} \\ \alpha^h_{l,i} &= f'_{l,i} - \frac{1}{2} f'_{l,i-1} - \frac{1}{2} f'_{l,i+1},\ f'_{l,i} = f_{l,i} - f_{0,0} \end{aligned} \tag{2.7}$$

die hierachischen Überschüsse.

Diese Berechnungsvorschrift orientiert sich an der Berechnung von hierarchischen Überschüssen von Dünnen Boundary-Gittern aus [Val14]. Hierbei zu beachten ist die Verwendung von $f'_{l,i}$ in der Berechnung. Diese ist notwendig, da die Ansatzfunktion $\phi_{-1,0}$ keine echte Hütchenfunktion ist und somit gesondert betrachtet werden muss. Für mehrdimensionale Dünne ANOVA-Gitter können dann die hierachischen Überschüsse mithilfe des unidirektionalen Prinzips aus [Bun96] berechnet werden, indem (2.7) für jede Dimension hintereinander angewendet wird.

2.3. Klassische ANOVA-Dekomposition

Die hier verwendete ANOVA-Dekomposition, wie sie in [Sob01] beschrieben ist, wird hier zur Differenzierung als klassische ANOVA-Dekomposition bezeichnet. Damit eine Dekomposition einer Funktion analog zu (2.2) als ANOVA-Dekomposition der Funktion bezeichnet werden kann, müssen alle Funktionsterme folgende Eigenschaft erfüllen:

$$\int_0^1 f_{\underline{c}}(x_{i_1}, \ldots, x_{i_t})\, dx_k = 0,\ k \in \{i_1, \ldots, i_t\}$$

Dies ist äquivalent zu der Aussage, dass die einzelnen Funktionsterme orthogonal zueinander sein müssen, das heißt für $\underline{c}, \underline{c}' \in C$ mit $\underline{c} \neq \underline{c}'$ muss gelten:

$$\langle f_{\underline{c}}, f_{\underline{c}'} \rangle = 0 \tag{2.8}$$

Somit gilt dann für den ersten Funktionsterm:

$$f_0 = \int_\Omega f(x)\, \mathrm{d}x$$

und für einen Funktionsterm einer beliebigen ANOVA-Komponente $\underline{c} \in C$:

$$f_{\underline{c}}(x_{i_1}, \ldots, x_{i_t}) = \int f_{i_1,\ldots,i_t}(x_{i_1}, \ldots, x_{i_t})\, \mathrm{d}x_{i_1} \ldots \mathrm{d}x_{i_t} - \sum_{\underline{c}' \leq \underline{c}} f_{\underline{c}'}(x) \tag{2.9}$$

Wie bereits erwähnt, erfordert die ANOVA-Dekomposition, dass die einzelnen Funktionsterme orthogonal zueinander seien müssen, d.h. $\langle f_c, f_{c'} \rangle = 0$. Um die Orthogonalitätsbedingung (2.8) zu erfüllen, wäre die simpelste Lösung eine orthogonale Basis zu verwenden, d.h. für jede eindimensionale Ansatzfunktion dieser Basis ϕ und ϕ' mit $\phi \neq \phi'$ müsste $\langle \phi, \phi' \rangle = 0$ gelten. Da dies

schwer zu erreichen ist, kann die Bedingung auf semi-orthogonal abgeschwächt werden, da innerhalb eines Funktionstermes einer ANOVA-Komponente keine Orthogonalität der Ansatzfunktionen benötigt wird. Somit müssen nur Ansatzfunktionen für unterschiedliche Level oder Multi-Level-Indizes orthogonal sein.

Definition 2.3.1
Seien $m,n \in \mathbb{N}_0 \cup \{-1\}, m \neq n$ zwei unterschiedliche Level. Dann ist eine Basis für ein Dünnes ANOVA-Gitter semi-orthogonal, falls für alle eindimensionalen Ansatzfunktionen $\phi_{l,i}$ der Basis gilt:

$$\langle \phi_{m,i}, \phi_{n,j} \rangle = 0,\ i \in \tilde{H}_m, j \in \tilde{H}_n$$

Demnach können bei einer semi-orthogonalen Basis zwei unterschiedliche eindimensionale Ansatzfunktionen auf dem gleichen Level nicht orthogonal zueinander sein. Die benötigte Semi-Orthogonalität hat den Nachteil, dass häufig genutzte Basen nicht verwendet werden können, da sie nicht semi-orthogonal sind.

Prewavelet-Basis

Eine Art von Basis, die die Semi-Orthogonalitätseigenschaft erfüllt, sind Neumann-Prewavelets, d.h. Prewavelets die Randwerte miteinbeziehen aus [GO95]. Jede Prewavelet-Ansatzfunktion ist eine Linearkombination von umliegenden Hütchenfunktionen des gleichen Levels. Im Unterschied zu Dirichlet-Prewavelets beziehen Neumann-Prewavelets dabei auch die Hütchenfunktionen am Rand $\phi_{l,0}$ und $\phi_{l,2^l}$ mit ein und können folgendermaßen konstruiert werden:

Definition 2.3.2
Seien $\phi_{l,i}$ die bereits definierten Hütchenfunktionen für Level l und Index i. Dann ist $\phi^p_{l,i}$ eine Prewavelet-Ansatzfunktion mit:

$$\begin{aligned}
\phi^p_{-1,0} &= \phi_{-1,0} \\
\phi^p_{0,1} &= -\phi_{0,0} + \phi_{0,1} \\
\phi^p_{1,1} &= -\phi_{1,0} + \phi_{1,1} - \phi_{1,2}
\end{aligned}$$

$$\begin{aligned}
\phi^p_{l,1} &= -\frac{12}{10}\phi_{l,0} + \frac{11}{10}\phi_{l,1} - \frac{6}{10}\phi_{l,2} + \frac{1}{10}\phi_{l,3} \\
\phi^p_{l,2^l-1} &= \frac{1}{10}\phi_{l,2^l-3} - \frac{6}{10}\phi_{l,2^l-2} + \frac{11}{10}\phi_{l,2^l-1} - \frac{12}{10}\phi_{l,2^l} \\
\phi^p_{l,i} &= \frac{1}{10}\phi_{l,i-2} - \frac{6}{10}\phi_{l,i-1} + \frac{10}{10}\phi_{l,i} - \frac{6}{10}\phi_{l,i+1} + \frac{1}{10}\phi_{l,i+2}
\end{aligned} \tag{2.10}$$

Hierzu kann angemerkt werden, dass diese Basisfunktionen sich nicht besonders zur genauen Interpolation von vielen Funktionen im Vergleich zu anderen Basisfunktionen eignen, wie später im Vergleich zu sehen ist. Da sie aber nur zur Varianzanalyse benutzt wird, ist der größere Interpolationsfehler verschmerzbar.

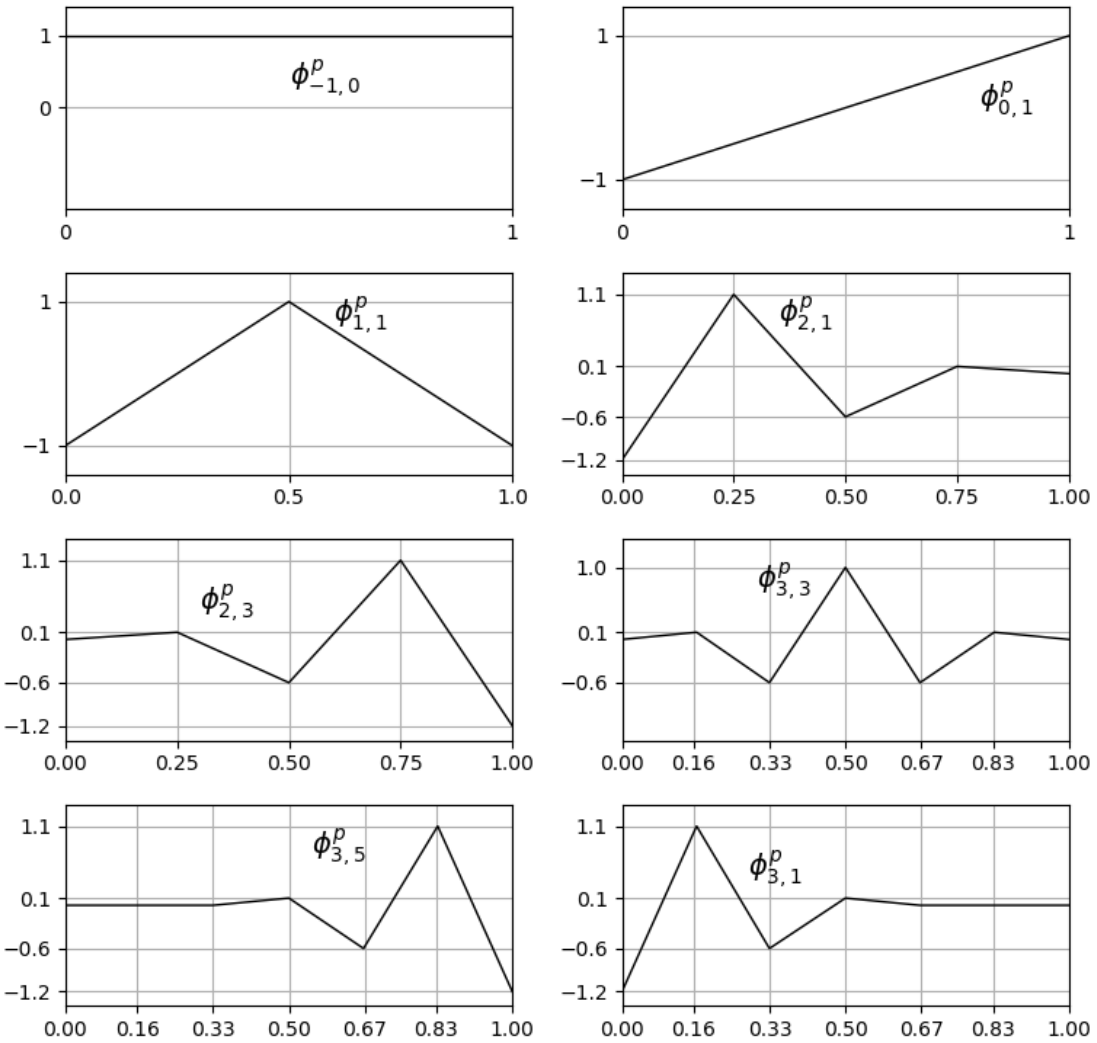

Abbildung 2.4.: Die eindimensionale hierarchische Prewavelet-Basis eines Dünnen ANOVA-Gitters für die Level -1 bis 3

Die Orthogonalitätseigenschaften der Prewavelets werden in [GO95] bewiesen. Somit gilt für zwei Level l, l' mit $l \neq l'$:

$$\langle \phi^p_{l,i}, \phi^p_{l',i} \rangle = 0 \tag{2.11}$$

Da die Ansatzfunktion $\phi^p_{-1,0} = 1$ konstant ist, folgt nach (2.11) außerdem, dass für $l \geq 0$ gilt:

$$\int_0^1 \phi^p_{l,i}(x)\, \mathrm{d}x = 0$$

Mithilfe der eindimensionalen Prewavelet-Ansatzfunktionen kann nun eine beliebig-dimensionale ANOVA-Prewavelet-Ansatzfunktion über den Tensorprodukt-Ansatz konstruiert werden:

Definition 2.3.3
Sei

$$\phi^p_{\underline{l},\underline{i}} \coloneqq \prod_{d=1}^{D} \phi^p_{l_d,i_d}$$

das D-dimensionale Prewavelet für den Multi-Level-Index $\underline{l}$ und den Multi-Index $\underline{i}$ und

$$W^p_{\underline{l}} \coloneqq span\{\phi^p_{\underline{l},\underline{i}} \mid \underline{i} \in \tilde{H}_{\underline{l}}\}$$

der hierarchische Unterraum für einen Multi-Level-Index $\underline{l}$ der mithilfe der Prewavelet-Basis auf einem Dünnen ANOVA-Gitters aufgespannt wird.

Definition 2.3.4
Sei l das Level und $g\colon \Omega \to \mathbb{R}$ eine Funktion. Dann bezeichnen wir mit

$$\mathbb{S}^{ANOVA}_{g,l}(x) \coloneqq \sum_{|\underline{l}+\underline{1}|_1 \leq l+1} \sum_{\underline{i} \in \tilde{H}_{\underline{l}}} \alpha^p_{\underline{l},\underline{i}}\, \phi^p_{\underline{l},\underline{i}}(x)$$

ein Surrogat für die Funktion g, welches mit der hierarchischen Prewavelet-Basis auf einem Dünnen ANOVA-Gitter mit Level l konstruiert wurde. Außerdem bezeichnen wir mit

$$\omega^{ANOVA}(x) = \mathbb{S}^{ANOVA}_{f,l}(x)$$

das Surrogat für die ursprüngliche Modellfunktion f mit beliebigen Level.

2.4. Diskretisierung der ANOVA-Dekomposition

Da nun behandelt wurde, wie eine Eingabefunktion mithilfe eines Dünnen ANOVA-Gitters und hierarchischer Hütchen- oder Prewavelet-Basis als Linearkombination der einzelnen Basisfunktionen ausgedrückt werden kann, ist es nun möglich, diese Linearkombination analog zu 2.3 komponentenweise zu zerlegen.

Definition 2.4.1
Sei $\tilde{H}_{\underline{l}}$ die Index Menge eines Dünnen ANOVA-Gitters, wie bereits in (2.4) *definiert. Dann ist*

$$\tilde{W}_{\underline{l}} \in \{W^h_{\underline{l}},\ W^p_{\underline{l}}\}$$

der hierarchische Unterraum für einen Multi-Level-Index $\underline{l}$ eines Dünnen ANOVA-Gitters mit hierarchischer Hütchenbasis oder hierarchischer Prewavelet-Basis.

Definition 2.4.2
Sei l das Level des Dünnen ANOVA-Gitters. Dann ist

$$A_c \coloneqq \{|\underline{l'} + \underline{1}|_1 \leq l+1 \mid \forall i \in \{1,\dots,D\}\colon (l'_i = -1 \wedge c_i = 0) \vee (l'_i \geq 0 \wedge c_i = 1)\}$$

die Menge aller Multi-Level-Indizes einer ANOVA-Komponente $\underline{c} \in C$.

Definition 2.4.3
Sei

$$\tilde{V}_c := \bigoplus_{\underline{l}' \in A_c} \tilde{W}_{\underline{l}'}$$

der Teil eines Dünnen ANOVA-Gitters für die ANOVA-Komponente c und

$$\tilde{V}_l := \bigoplus_{\underline{c} \in C} \tilde{V}_c$$

das komplette Dünne ANOVA-Gitter mit Level l.

Anstatt nun die hierarchischen Unterräume direkt zu einem Dünnen ANOVA-Gitter zusammenzufassen, werden hier alle hierarchischen Unterräume einer ANOVA-Komponente zugeordnet. Ein Beispiel dafür ist Abbildung 2.5 zu sehen.

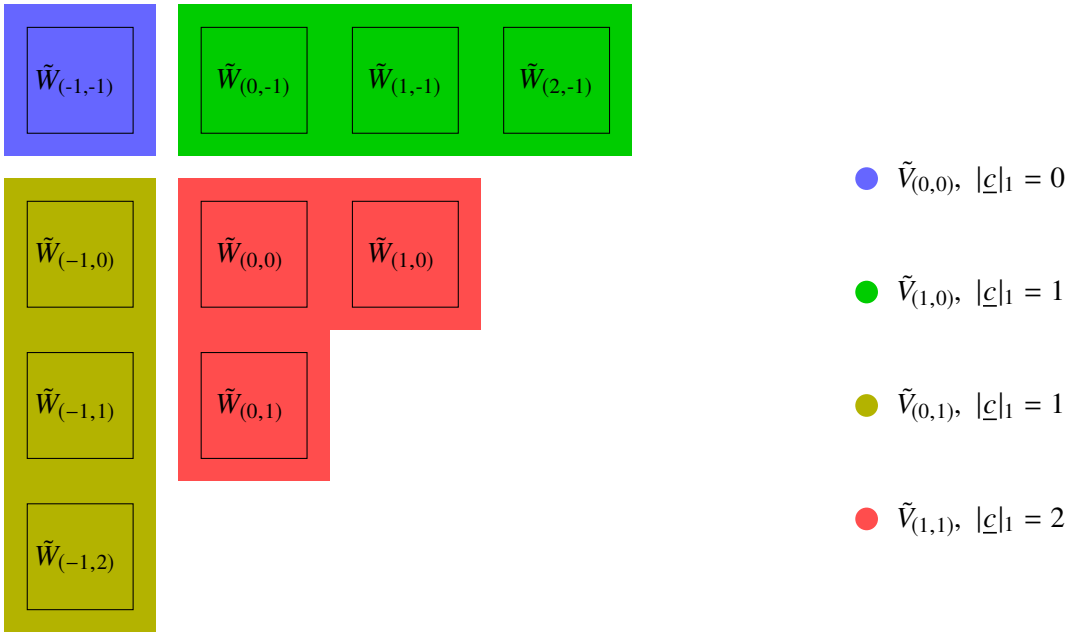

Abbildung 2.5.: Beispielhafte Aufteilung des Funktionenraumes $\tilde{V}$ eines Dünnen ANOVA-Gitters mit $l = 2$ in die hierarchischen Unterräume $\tilde{V}_c$ der einzelnen ANOVA-Komponenten aus $C = \{(0,0),(0,1),(1,0),(1,1)\}$.

Diese Art der Dekomposition kann nun auf die Surrogate beider ANOVA-Methoden angewendet werden. Zum einen auf die Anchored-ANOVA-Dekomposition:

$$\begin{aligned}\omega^{\text{Anchor}}(x) &= \sum_{|\underline{l}+\underline{1}|_1 \leq l+1} \sum_{\underline{i} \in \vec{H}_{\underline{l}}} \alpha^h_{\underline{l},\underline{i}}\, \phi_{\underline{l},\underline{i}}(x) \\ &= \sum_{\underline{c} \in C} \sum_{\underline{l}' \in A^l_{\underline{c}}} \sum_{\underline{i} \in \vec{H}_{\underline{l}'}} \alpha^h_{\underline{l},\underline{i}}\, \phi_{\underline{l},\underline{i}}(x) \\ &= \sum_{\underline{c} \in C} \omega^{\text{Anchor}}_{\underline{c}}(x),\ \omega^{\text{Anchor}}_{\underline{c}} \in W^h_{\underline{l}}\end{aligned} \tag{2.12}$$

Dadurch dass verschiedene Anker a verwendet werden können, kann diese Dekomposition variiert werden, da sich in der Folge die einzelnen $\alpha^h_{\underline{l},\underline{i}}$ unterscheiden. Dies ist, wie später im Vergleich zu sehen, sehr nützlich, da so die Gesamtvarianz von ω^{Anchor} verschieden stark auf die einzelnen Funktionsterme ω^{Anchor}_c ausgelagert werden kann und somit bei einer guten Ankerwahl der Dimensionsreduktionsprozess ein besseres Ergebnis lierfert.

Zum anderen kann auch die klassische ANOVA-Dekomposition angewendet werden:

$$\begin{aligned}\omega^{\text{ANOVA}}(x) &= \sum_{|\underline{l}+\underline{1}|_1 \leq l+1} \sum_{\underline{i} \in \vec{H}_{\underline{l}}} \alpha^p_{\underline{l},\underline{i}}\, \phi^p_{\underline{l},\underline{i}}(x) \\ &= \sum_{\underline{c} \in C} \sum_{\underline{l}' \in A^l_{\underline{c}}} \sum_{\underline{i} \in \vec{H}_{\underline{l}'}} \alpha^p_{\underline{l},\underline{i}}\, \phi^p_{\underline{l},\underline{i}}(x) \\ &= \sum_{\underline{c} \in C} \omega^{\text{ANOVA}}_{\underline{c}}(x),\ \omega^{\text{ANOVA}}_{\underline{c}} \in W^p_{\underline{l}}\end{aligned} \tag{2.13}$$

Hier sind demnach die $\omega_{\underline{c}}$ die diskretisierten Funktionsterme der jeweiligen ANOVA-Dekomposition.

2.5. Varianzanalyse

Die Dekompositionen (2.12) und (2.13) können nun aus dem Blickwinkel der Sensitivitätsanalyse, wie sie in [Sob01] beschrieben ist, betrachtet werden. Dafür definiert man Sensitivitätsindizes wie folgt:

Definition 2.5.1
Sei $\underline{c} \in C$ eine ANOVA-Komponente. Dann ist der Sensitivitätsindex $S_{\underline{c}}$ der Komponente gegeben durch

$$S_{\underline{c}} \in [0,1]$$

Die Anzahl der aktiven Dimensionen, also $|\underline{c}|_1$, ist die Ordnung dieses Sensitivitätsindex. Für $\underline{c} \in C$ mit $|\underline{c}|_1 = 1$ ist z.B. der entsprechende Sensitivitätsindex erster Ordnung. Außerdem gilt:

$$\sum_{\underline{c} \in C} S_{\underline{c}} = 1$$

Mithilfe dieser Sensitivitätsindizes kann man nun untersuchen, welche Kombination an aktiven Parametern einen großen oder kleinen Einfluss auf die Gesamtvarianz haben und dementsprechend einige Funktionsterme von ANOVA-Komponenten aus der ursprünglichen Dekomposition zu entfernen. Je größer der Sensitivitätsindex, desto größer ist der Beitrag des Funktionsterms der ANOVA-Komponente zu der Gesamtvarianz.

Definition 2.5.2
Sei $i \in \{1, \dots, D\}$ ein Dimensionsindex und $C^i = \{\underline{c} \in C \mid c_i = 1\}$ die Menge der in der i-ten Dimension aktiven ANOVA-Komponenten. Dann ist

$$S_{T_i} \coloneqq \sum_{\underline{c} \in C_i} S_{\underline{c}}$$

der totale Effektindex von i.

Dieser gibt an, wie groß der Beitrag der i-ten Dimension zur Gesamtvarianz ist.

Nun kann damit begonnen werden, die einzelnen Funktionsterme der gerade diskretisierten Dekomposition auf ihre Varianz zu untersuchen, d.h. Sensitivitätsindizes für jede ANOVA-Komponente zu definieren und zu bestimmen.

Satz 1
Seien $\underline{c}, \underline{c}' \in C$ mit $\underline{c} \neq \underline{c}'$ zwei unterschiedliche ANOVA-Komponenten. Seien hier außerdem $\omega_{\underline{c}}\omega_{\underline{c}'} \in span\{\phi^p_{\underline{l},\underline{i}} \mid \underline{i} \in \tilde{H}_{\underline{l}}\}$ zwei Funktionsterme der diskretisierten klassischen ANOVA-Dekomposition. Dann gilt für die Varianz der Funktionensterme:

$$\sigma^2(\omega_{\underline{c}} + \omega_{\underline{c}'}) = \sigma^2(\omega_{\underline{c}}) + \sigma^2(\omega_{\underline{c}'})$$

Proof 1
Da die Basisfunktionen ϕ^p der verwendeten Basis semi-orthogonal sind, gilt $\langle \omega_{\underline{c}}, \omega_{\underline{c}'} \rangle = 0$. Daraus folgt:

$$\begin{aligned}
\sigma^2(\omega_{\underline{c}} + \omega_{\underline{c}'}) =& \|\omega_{\underline{c}} + \omega_{\underline{c}'}\|^2_{L^2} \\
=& \langle \omega_{\underline{c}} + \omega_{\underline{c}'}, \omega_{\underline{c}} + \omega_{\underline{c}'} \rangle \\
=& \langle \omega_{\underline{c}}, \omega_c \rangle + \langle \omega_{\underline{c}'}, \omega_{\underline{c}'} \rangle + \langle \omega_{\underline{c}}, \omega_{\underline{c}'} \rangle + \langle \omega_{\underline{c}'}, \omega_{\underline{c}} \rangle \\
=& \|\omega_{\underline{c}}\|^2_{L^2} + \|\omega_{\underline{c}'}\|^2_{L^2} + \langle \omega_{\underline{c}}, \omega_{\underline{c}'} \rangle + \langle \omega_{\underline{c}'}, \omega_{\underline{c}} \rangle \\
=& \|\omega_{\underline{c}}\|^2_{L^2} + \|\omega_{\underline{c}'}\|^2_{L^2} \\
=& \sigma^2(\omega_{\underline{c}}) + \sigma^2(\omega_{\underline{c}'})
\end{aligned}$$

Daraus folgt, dass man die Gesamtvarianz der Funktion ω in die Varianz der einzelnen Funktionsterme ω_c folgendermaßen zerlegen kann:

$$\sigma^2(\omega^{\text{ANOVA}}) = \sigma^2\left(\sum_{\underline{c} \in C} \omega^{\text{ANOVA}}_{\underline{c}}\right) = \sum_{\underline{c} \in C} \sigma^2(\omega^{\text{ANOVA}}_{\underline{c}})$$

Somit können die Sensitivitätsindizes für die klassische ANOVA-Methode wie folgt definiert werden:

Definition 2.5.3
Sei $\underline{c} \in C$ eine ANOVA-Komponente. Dann ist der Sensitivitätsindex $S_{\underline{c}}$ der Komponente gegeben durch

$$S_{\underline{c}}^{ANOVA} := \frac{\sigma^2(\omega_{\underline{c}}^{ANOVA})}{\sigma^2(\omega^{ANOVA})}, \quad \sum_{\underline{c} \in C} S_{\underline{c}}^{ANOVA} = 1$$

Die Anchored-ANOVA-Dekomposition erfüllt diese Orthogonalitätsbedingung nicht und somit werden die Sensitivitätsindizes für die Anchored-ANOVA-Methode anders definiert:

Definition 2.5.4
Sei $\underline{c} \in C$ eine ANOVA-Komponente. Dann ist der Sensitivitätsindex der Komponente gegeben durch

$$S_{\underline{c}}^{Anchor} := \frac{\sigma^2(\omega_{\underline{c}}^{Anchor})}{\sum_{\underline{c} \in C} \sigma^2(\omega_c^{Anchor})}, \quad \sum_{\underline{c} \in C} S_{\underline{c}}^{Anchor} = 1$$

Somit spiegeln die Sensitivitätsindizes der klassischen ANOVA-Dekomposition den echten Anteil der Gesamtvarianz einer Komponente wieder, die Sensitivitätsindizes der Anchored-ANOVA-Dekomposition jedoch nicht. Diese sind zwar auch indikativ dafür, wie groß der Beitrag einer Komponente ist, jedoch kann aus den Sensitivitätsindizes der Anchored-ANOVA-Dekomposition kein Rückschluss auf den Gesamtvarianzanteil gezogen werden.

2.6. Varianten

Sind jetzt die Sensitivitätsindizes für jede ANOVA-Komponente bestimmt, muss entschieden werden, wie nun die Funktion durch die Entfernung einiger Funktionsterme in der Dimension reduziert werden kann. Dafür werden zwei Varianten vorgestellt. Bei der der parameterbasierten Variante beseitigt man bestimmte Parameter komplett, d.h alle Funktionsterme, die einen dieser Parameter enthalten, werden entfernt. Bei der effektiven Dimensionsreduktion werden alle Funktionsterme, die eine bestimmte Anzahl an aktiven Parametern überschreiten, entfernt.

2.6.1. Parameterbasierte Dimensionsreduktion

Bei der parameterbasierten Dimensionsreduktion versucht man, die Anzahl der Eingabeparameter für das Surrogat zu verkleinern, indem jeder einzelne Eingabeparameter auf seinen Beitrag zur Gesamtvarianz untersucht und gegebenenfalls ganz aus dem Modell entfernt wird. Dafür definiert man eine Transformationsfunktion, welche die inaktiven Parameter eliminiert:

Definition 2.6.1
Sei o.B.d.A $R = \{d_1, \ldots, d_r\}$ mit $d_1 < \ldots < d_r$ eine geordnete Menge an Dimensionsindizes, die entfernt werden sollen.

Dann ist T_R eine $(D \times d) - Matrix$ mit

$$(T_R)_{i,j} := \begin{cases} 1, & d_i = j \\ 0, & sonst \end{cases}$$

Mithilfe dieser Matrix können nun Eingabevektoren $x \in [0,1]^D$ in geringer-dimensionale Vektoren $x' \in [0,1]^d$ überführt werden:

Definition 2.6.2
Sei

$$t_R^{ANOVA}(x) \coloneqq T_R\, x$$

die Transformationsfunktion.

Um nun das d-dimensionale Surrogat richtig zu konstruieren, muss die ursprüngliche Modellfunktion entlang des Definitionsbereichs des Surrogats ausgewertet werden. Dies geschieht mittels der invertierten Transformationsfunktion $(t^{ANOVA})^{-1}(x) = T_R^{-1}\, x$, die leicht berechnet werden kann.

Definition 2.6.3
Sei $R \subseteq \{1,\ldots,D\}$ die Menge an Dimensionsindizes, die entfernt werden sollen und somit $d = D - |R|$. Sei außerdem $\gamma = \omega \circ (t_R^{ANOVA})^{-1}$ die Auswertungsfunktion und l das Level. Dann ist

$$\eta_R^{ANOVA}(x) \coloneqq \mathbb{S}_{\gamma,l}^{Boundary}(x)$$

das reduzierte Surrogat, aus dem alle inaktiven Dimensionen R entfernt wurden.

Das Surrogat $\mathbb{S}_\gamma$ kann mithilfe eines Dünnen Gitters mit einer beliebigen Basis erstellt werden, d.h. die ANOVA-Prewavelet-Basis wird hier nicht mehr benötigt. Dies ist vorteilhaft, da, wie schon vorher erwähnt, die ANOVA-Prewavelet-Basis im Vergleich zu vielen anderen Basen einen größeren Interpolationsfehler verursacht und nur zur Varianzanalyse gut geeignet ist. In dieser Arbeit wird für das reduzierte Surrogat die hierarchische Hütchenbasis auf einem Dünnen Boundary-Gitter verwendet.

Abbildung 2.6.: Beispielhafte Reduktion der Dimensionen, indem $\tilde{V}_{(1,0)}$ und $\tilde{V}_{(1,1)}$ entfernt werden. Die resultierende Funktion $\omega(x, y) = \omega_{(0,0)} + \omega_{(0,1)}(y)$ ist echt eindimensional, da hier der Parameter x nicht verwendet wird und somit entfernt werden kann.

Algorithmus 2.1 Pseudocode eines rekursiven Algorithmus für die parameterbasierte Dimensionsreduktion. Eingabe ist die Funktion ω, welche ein Surrogat auf einem Dünnen ANOVA-Gitter für die Eingabefunktion ist und der minimale Anteil der Gesamtvarianz $\sigma^2_{min} \in [0,1]$, den die reduzierte Funktion mindestens abdecken soll.
Mit jedem Rekursionsschritt wird eine Dimension entfernt.

function REDUCE(ω, σ^2_{min})
  $R = \emptyset$
  $\sigma^2_{current} \leftarrow 1.0$
  $reduceRec(f, R, \sigma^2_{current}, \sigma^2_{min})$
  return $(t^{ANOVA}_R, \eta^{ANOVA}_R)$
end function

function REDUCEREC($\omega, R, \sigma^2_{current}, \sigma^2_{min}$)
  $r \leftarrow 0$
  $m \leftarrow \infty$
  for $i \in \{1, \ldots, D\} \setminus R$ **do**
    if $S_{T_i} < m$ **then**
      $m \leftarrow S_{T_i}$
      $r \leftarrow i$
    end if
  end for

  $R_{new} = R \cup \{r\}$
  $\sigma^2_{now} \leftarrow \sigma^2_{rem}(\omega, t^{ANOVA}_{R_{new}}, \eta^{ANOVA}_{R_{new}})$
  if $\sigma^2_{now} < \sigma^2_{min}$ **then**
    return
  else
    $reduceRec(\omega, R_{new}, \sigma^2_{now}, \sigma^2_{min})$
  end if
end function

2.6.2. Effektive Dimensionsreduktion

Im Unterschied zur parameterbasierten Dimensionsreduktion, bei der versucht wird, die Anzahl der Eingabeparameter für das Surrogat zu verkleinern, bleibt bei der effektiven Dimensionsreduktion die ursprüngliche der Anzahl verschiedenen Eingabeparameter gleich, d.h. es werden keine Parameter vollständig eliminiert. Nur die Komplexität der einzelnen Funktionsterme wird hier gesenkt.

Definition 2.6.4
Sei

$$C_i := \{\underline{c} \in C \mid |c|_1 = i\}$$

die Menge aller ANOVA-Komponenten mit i aktiven Dimensionen. Hier wird i auch als ANOVA-Ordnung bezeichnet, da i angibt, wie viele nicht-konstante Dimensionen eine ANOVA-Komponente besitzt.

Dann kann die Funktion ω wie nach aufsteigender ANOVA-Ordnung zerlegt werden:

$$\omega^{\text{ANOVA}}(x) = \sum_{i=0}^{D} \sum_{\underline{c} \in C_i} \omega_{\underline{c}}^{\text{ANOVA}}(x)$$

Sei jetzt $k \in \{0, \ldots, D-1\}$ die maximale Ordnung, also die maximale Parameteranzahl, die ein Funktionsterm haben darf. Dann ist

$$\omega_k^{\text{ANOVA}}(x) := \sum_{i=0}^{k} \sum_{\underline{c} \in C_i} \omega_{\underline{c}}^{\text{ANOVA}}(x)$$

die ANOVA-Dekomposition bis zur k-ten Ordnung. Es wurden also alle Funktionsterme mit einer Ordnung, die größer als k ist, entfernt. Der Varianzverlust, der beim Entfernen der Funktionsterme der klassischen ANOVA-Dekomposition entsteht, lässt sich wie folgt berechnen:

$$\begin{aligned} \sigma^2(\omega^{\text{ANOVA}} - \omega_k^{\text{ANOVA}}) &= \sigma^2\Big(\sum_{i=0}^{D} \sum_{\underline{c} \in C_i} \omega_{\underline{c}}^{\text{ANOVA}} - \sum_{i=0}^{k} \sum_{\underline{c} \in C_i} \omega_{\underline{c}}^{\text{ANOVA}}\Big) \\ &= \sum_{i=k+1}^{D} \sum_{\underline{c} \in C_i} \sigma^2(\omega_{\underline{c}}^{\text{ANOVA}}) \end{aligned} \tag{2.14}$$

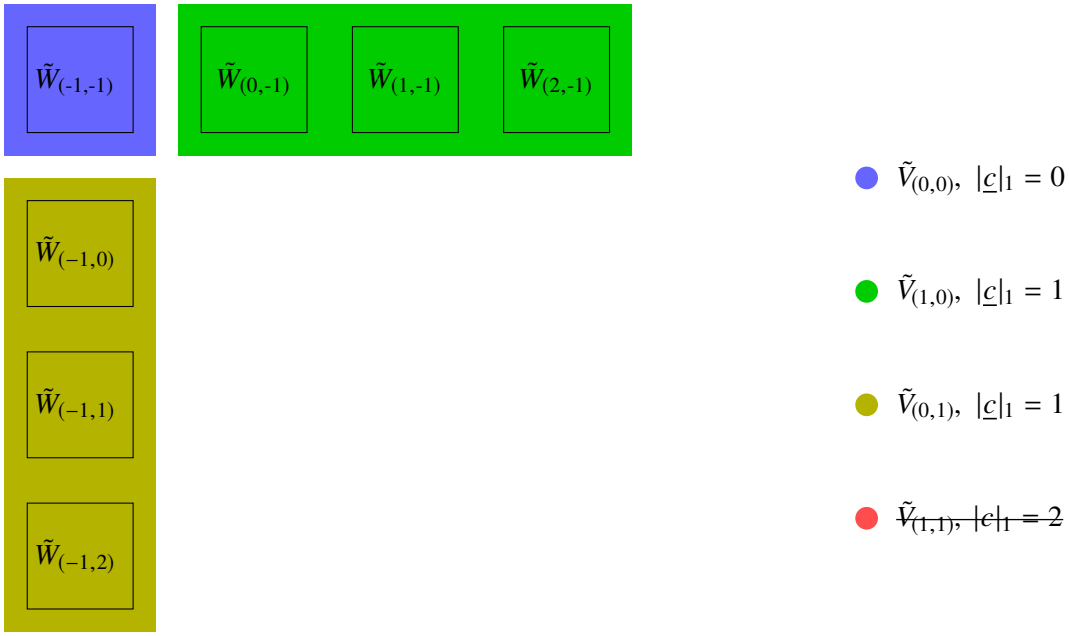

Abbildung 2.7.: Beispielhafte Reduktion der effektiven Dimensionen unter Verwendung der ANOVA-Ordnung $k = 1$. Alle hierarchischen Unterräume $\tilde{W}_{\underline{l}}$ von ANOVA-Komponenten $\underline{c} \in C$ mit $|c|_1 > k$ werden entfernt, in diesem Fall die ANOVA-Komponente $(1, 1)$. Die resultierende Funktion $\omega(x, y) = \omega_{(0,0)} + \omega_{(0,1)}(x) + \omega_{(0,1)}(y)$ ist effektiv eindimensional, da alle Teilfunktionen maximal eindimensonal sind.

Findet man nun eine Ordnung $k < D$, sodass die Varianz $\sigma^2(\omega^{\text{ANOVA}} - \omega_k^{\text{ANOVA}})$, die durch das Weglassen von Funktionenstermen mit einer größeren Ordnung als k Dimensionen verloren geht, akzeptabel ist, kann man so die effektive Dimension der Funktion ω auf von D auf k reduzieren. Dadurch kann die Komplexität von $O(n^D)$ auf $O(n^k)$ reduziert werden, da wir nun eine Summe an Funktionenstermen mit maximal k Eingabeparametern haben. Die ursprüngliche Anzahl an verschiedenen Eingabeparametern D bleibt aber bei $\omega_k^{\text{ANOVA}}(x)$ unverändert.

Hier wurde nur die effektive Dimensionsreduktion für ein Surrogat, welches bereits mithilfe der klassischen ANOVA-Methode zerlegt wurde, gezeigt. Die effektive Dimensionsreduktion kann jedoch ebenfalls auf eine Anchored-ANOVA-Zerlegung angewendet werden, jedoch kann dann die verlorere Varianz nicht so einfach wie in (2.14) berechnet werden.

Diese Methode wird der Vollständigkeit halber hier beschrieben, da oft in der Literatur, wie z.B in [Gar13] [Feu10], mit ANOVA-Dimensionsreduktion implizit die effektive Dimensionsreduktionsvariante gemeint ist. Jedoch wird diese Methode nicht in den späteren Vergleich mitaufgenommen, weil sie sich nicht mit den anderen hier vorgestellen Methoden vergleichen lässt, da, wie vorher beschrieben, im Gegensatz zu allen anderen Methoden kein Parameter vollständig eliminiert wird.

3. Achsenfreie Methoden

Die folgenden zwei Methoden berechnen eine Orthonormalbasis mit Gewichten für jeden einzelnen Basisvektor dieser Orthonormalbasis, um auszudrücken, wie aktiv die Funktion entlang dieser Richtung ist. Anschließend wird eine Orthogonalprojektion mittels der aktivsten d Basisvektoren und einem Stützvektor durchgeführt. Die aktivsten Basisvektoren sind die Vektoren mit der höchsten Gewichtung. Zum Schluss wird die Transformationsfunktion konstruiert, die Elemente aus dem Parameterraum erst auf einen Projektionsraum und anschließend in den Parameterraum eines d-dimensionalen Surrogats abbildet.

Definition 3.0.1
Sei $V = \{v_1, v_2, \ldots, v_D\}$ eine Menge an orthogonalen Richtungsvektoren mit $|v_i| = 1$. Sei außerdem $\gamma\colon \{v_1, v_2, \ldots, v_D\} \to \mathbb{R}$ eine Gewichtungsfunktion, die jedem Richtungsvektor ein Gewicht zuordnet. Dann ist

$$B \coloneqq \left(v_{i_1}\ v_{i_2}\ \ldots\ v_{i_{D-1}}\ v_{i_D} \right),\ \gamma(v_{i_1}) \geq \ldots \geq \gamma(v_{i_D})$$

eine Orthonormalbasis, wobei die Reihenfolge der Spaltenvektoren andeutet, in welche Richtungen eine Funktion am aktivsten ist.

Der Projektionsraum ist ebenfalls ein D-dimensionaler Raum, welcher später zur Konstruktion des reduzierten Surrogats benutzt wird. Im Idealfall ist er ein Unterraum von Ω, jedoch ist dies, wie später zu sehen ist, nicht immer möglich.

Definition 3.0.2
Sei

$$m \coloneqq \begin{pmatrix} m_1 \\ \ldots \\ m_D \end{pmatrix}$$

der Stützvektor des Projektionsraumes.

Dieser Stützvektor ist notwendig, da nur basierend auf den aktiven Richtungen in B kein genaues reduzierte Surrogat erstellt werden kann. Hat eine Funktion in einem nur sehr kleinen Bereich in Ω eine Änderungsrate ungleich null, muss der Stützvektor so bestimmt werden, dass er in diesem kleinen aktiven Bereich liegt, da sonst das reduzierte Surrogat nicht akkurat erstellt werden kann.

Unter Zuhilfenahme einer Projektionsfunktion werden Elemente aus Ω in einem Raum abgebildet, der durch den Stützvektor m und die ersten d Basisvektoren von B gegeben ist. Dafür wird eine Matrix definiert, bei der die letzten $D - d$ Basisvektoren aus B entfernt wurden:

Definition 3.0.3
Sei

$$B_d \coloneqq \begin{pmatrix} v_1 & v_2 & \dots & v_{d-1} & v_d & \underline{0} & \dots & \underline{0} \end{pmatrix}$$

eine $(D \times D)$-Matrix, welche die ersten d Spaltenvektoren der Orthonormalbasis B enthält und die restlichen Spalten durch Nullvektoren ersetzt.

Der Projektionsraum P wird durch die ersten d Basisvektoren von der Orthonormalbasis B aufgespannt, ist aber in seinen Ausmaßen begrenzt. Mithilfe einer Projektionsfunktion p werden Elemente aus Ω nach P abgebildet. Dabei wird der Eingabevektor x relativ zum Stützvektor in die neue Basis überführt, die durch die Matrix B_d gegeben ist.

Definition 3.0.4
Sei

$$\begin{aligned} p\colon [0,1]^D &\to P \\ x &\mapsto m + B_d^T\,(x - m) \end{aligned}$$

die Projektionsfunktion.

Wird auf dem Parameterraum Ω eine Orthogonalprojektion ausgeführt, entsteht ein Raum in der Form eines sogenannten Zonotops. Dieser Raum $p(\Omega)$ ist nicht immer ein Unterraum von Ω, sondern ein Unterraum von $\mathbb{R}^D$ in der Form eines d-dimensionalen Zonotops. Dieses d-dimensionale Zonotop ist ein konvexes Polygon, wobei dessen genaue Form von der Projektionsfunktion p abhängt. Da der Definitionsbereich der reduzierten Funktion Ω_R aber ein d-dimensionaler Einheitswürfel ist, muss der Raum $p(\Omega)$ angepasst werden. Dafür wird das kleinstmögliche d-dimensionale Rechteck gebildet, welches das ganze Zonotop enthält. Anschließend wird dieses Rechteck auf einen Einheitswürfel skaliert. Abbildung 3.1 illustriert diesen Sachverhalt mit einem simplen zweidimensionalen Beispiel.

Das kleinstmögliche d-dimensionale Rechteck in $\mathbb{R}^D$ ist der Projektionsraum P. Der Projektionsraum erfüllt also die Eigenschaft, dass jedes Element des ursprünglichen Definitionsbereiches Ω nach der Projektion mit $p(x)$ in dem Projektionsraum liegt, d.h. $\forall x \in \Omega\colon p(x) \in P$.

Definition 3.0.5
Sei

$$E \coloneqq \{0,1\}^D$$

die Menge aller Eckpunkte in Ω.

Diese Eckpunkte werden benötigt, um die genauen Ausmaße des Projektionsraumes zu bestimmen. Es ist leicht zu sehen, dass die projezierten Ecken $p(E)$ die Eckpunkte des Zonotops bilden. Liegen nun alle Ecken nach der Projektion mit p in P, d.h. $p(E) \subseteq P$, liegen auch alle anderen Punkte aus $p(\Omega)$ in P. Das heißt:

$$\forall e \in E\colon p(e) \in P \Rightarrow \forall x \in \Omega\colon p(x) \in P$$

Somit kann nun der Projektionsraum konstruiert werden, indem ein minimales d-dimensionales Rechteck konstruiert wird, welches alle projezierten Eckpunkte enthält:

Definition 3.0.6
Sei

$$P \coloneqq \left\{ m + \sum_{i=1}^{d} \beta_i v_i \mid \beta_i \in (\beta_i^-, \beta_i^+) \right\} \tag{3.1}$$

der Projetionsraum mit

$$\begin{aligned} \beta_i^+ &\coloneqq \max\{\langle (e-m), v_i \rangle \mid e \in E\} \\ \beta_i^- &\coloneqq \min\{\langle (e-m), v_i \rangle \mid e \in E\} \end{aligned} \tag{3.2}$$

Im Idealfall würden alle $x \in \Omega$ aus der ursprünglichen Definitionsmenge mithilfe der Projektionsfunktion $p(x)$ in einen Unterraum von Ω abgebildet. Wie aber in 3.1 zu sehen ist, gibt es Fälle, in denen dies nicht zutrifft.

Für dieses Problem existiert keine allgemeine Lösung, jedoch finden sich einige Ansätze. Der hier verwendete Ansatz ist, die ursprüngliche Funktion auch außerhalb des ursprünglichen Definitionsbereiches auszuwerten, um so ein akkurates Surrogat zu erstellen. Welcher Funktionswert für einen Eingabewert außerhalb des Definitionsbereiches sinnvoll ist, hängt von dem Modell ab. Das heißt, zur Dimensionsreduktion mit einer achsenfreien Methode muss die Eingabefunktion darauf ausgelegt werden, auch außerhalb von Ω ausgewertet werden zu können.

Die genauen Ausmaße des Projektionsraumes entlang jeder Dimension können mit folgendem Satz einfach berechnet werden:

Satz 2
Es gilt:

$$\begin{aligned} \beta_i^+ &= \sum_{j=1}^{D} \begin{cases} (1-m_j)(v_i)_j, & (v_i)_j \geq 0 \\ -m_j(v_i)_j, & (v_i)_j < 0 \end{cases} \\ \beta_i^- &= \sum_{j=1}^{D} \begin{cases} (1-m_j)(v_i)_j, & (v_i)_j < 0 \\ -m_j(v_i)_j, & (v_i)_j \geq 0 \end{cases} \end{aligned}$$

Proof 2
(3.2) *lässt sich auch folgendermaßen ausdrücken:*

$$\langle (e-m), v \rangle = \sum_{i=j}^{D} \begin{cases} (1-m_j)(v_i)_j, & e_i = 1 \\ -m_j(v_i)_j, & e_i = 0 \end{cases}$$

Um nun diesen Wert über alle möglichen Eckpunkte zu minimieren oder zu maximieren, muss jeder Summenterm minimal oder maximal gewählt werden.

Für diesen Projektionsraum kann nun ein d-dimensionales Surrogat erstellt werden, indem die Modellfunktion an einigen Punkten des Projektionsraumes ausgewertet wird und damit schließlich die reduzierte Funktion η interpoliert wird.

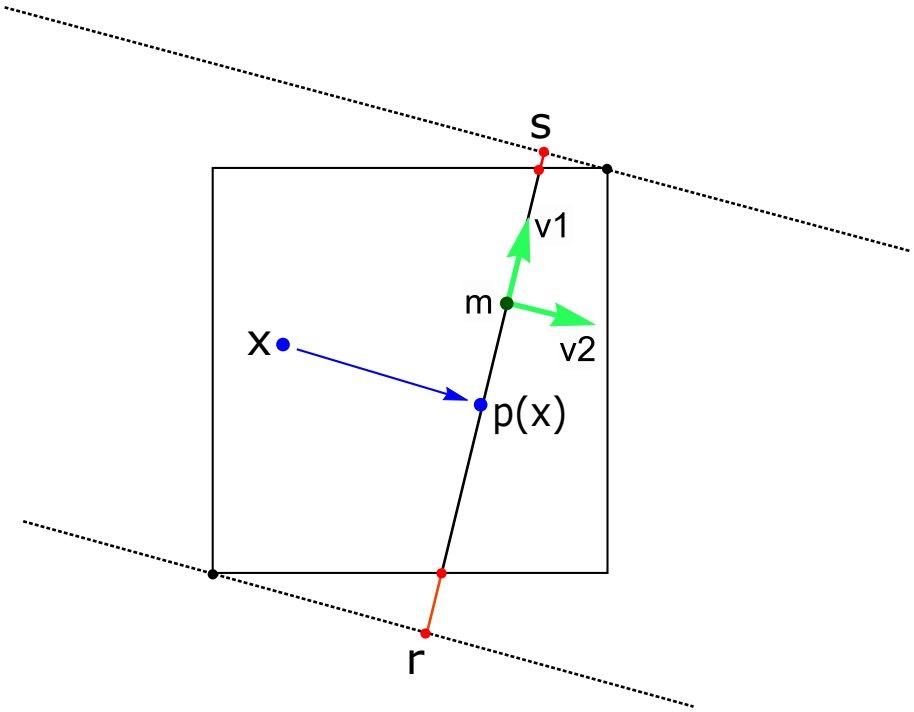

Abbildung 3.1.: Beispielhafte orthogonale Projektion vom ursprünglichen Parameterraum Ω, der durch den schwarzen Würfel begrenzt ist, auf den Projektionsraum P, der durch den Mittelpunkt m und den ersten Basisvektor v_1 gegeben ist. Der zweite Basisvektor wird bei dieser Projektion verworfen, also gilt $d = 1$. Der Raum P ist hier ein eindimensionales Rechteck in $\mathbb{R}^2$, aber kein Unterraum von Ω. Der Projektionsraum ist die Strecke von r nach s. Während die Projektion von Punkten wie x zu $p(x)$ problemlos verläuft, können bestimmte Punkte auf die rot markierten Abschnitte projeziert werden. Diese liegen dann außerhalb des neuen Unterraums und müssen speziell behandelt werden. Das eindimensionale Surrogat wird dann zwischen r und s gelegt. Hier bei korrespondiert r mit der Nullkoordinate des Surrogats. Die Einskoordinate befindet sich bei s.

Dafür muss noch eine Abbildung von dem Projektionsraum auf die Definitionsmenge von η erstellt werden:

Definition 3.0.7
Sei

$$p_s := m + \sum_{i=1}^{d} \beta_i^- v_i$$

der Punkt aus P, der zur Nullkoordinate des Surrogats zugeordnet wird. Dann ist

$$s\colon P \to \Omega_R$$
$$p \mapsto p',\ p'_i = \frac{\langle (p - p_s), v_i \rangle}{\beta_i^+ - \beta_i^-}$$

eine Funktion, welche Punkte aus dem Projektionsraum P auf die Definitionsmenge des reduzierten Surrogats Ω_R abbildet.

Damit kann die Transformationsfunktion definiert werden, welche die Elemente der urpsprünglichen Definitionsmenge Ω auf die Definitionsmenge des Surrogates Ω_R abbildet:

Definition 3.0.8
Sei

$$t(x) \coloneqq s(p(x))$$

die Transformationsfunktion.

Um nun das d-dimensionale Surrogat richtig zu konstruieren, muss die ursprüngliche Modellfunktion entlang des Projektionsraumes ausgewertet werden. Das Surrogat kann mittels eines Dünnen Gitters mit einer beliebigen Basis konstruiert werden.

Definition 3.0.9
Sei $\gamma = \omega \circ (t^{-1})$ die Auswertungsfunktion. Dann ist

$$\eta(x) \coloneqq \mathbb{S}_{\gamma,l}^{Boundary}(x)$$

die reduzierte Funktion, welche ein Surrogat ist.

Hier wird wieder o.B.d.A das Surrogat für die reduzierte Funktion mithilfe einer hierarchischen linearen Basis auf einem Dünnen Boundary-Gitter konstruiert.

3.1. Explorative Dimensionsreduktion

Die folgenden Methoden sind zwar inder Lage, mithilfe von Gewichten die Richtungsvektoren gemäß ihrer Wichtigkeit bei der Dimensionsreduktion ordnen, jedoch können bei dieser Gewichtung keine Aussagen darüber getroffen werden, wie gut eine reduzierte Funktion im Bezug zu der verwendeten Fehlermetrik, hier der verlorenen Gesamtvarianz, abschneidet. Um den eigentlichen Fehler, der bei einer Reduktion anfällt zu messen, muss also die Funktion solange explorativ reduziert werden, bis der gemessene Fehler, also die verlorene Varianz, zu groß wird.

Hierbei muss angemerkt werden, dass die berechneten Richtungsvektoren und Gewichte keinen Zusammenhang zu dem gemessenem Fehler besitzen. Das heißt, es ist durchaus möglich, dass es für eine Funktion eine andere Orthonormalbasis B' geben kann, für die es einige $d < D$ gibt, bei der der Fehler kleiner ist. Anders ausgedrückt, es ist nicht garantiert, dass die achsenfreien Methoden eine Funktion immer optimal bezüglich der verlorenen Varianz reduzieren. Wie gut verschiedene Methoden abschneiden, wird an einer späteren Stelle untersucht.

Algorithmus 3.1 Pseudocode für die explorative Dimensionsreduktion. Eingaben sind die Funktion ω, der Stützvektor m, die bereits mit einer achsenfreien Methode bestimmte Orthonormalbasis B und die Mindestvarianz σ^2_{min}.

```
function REDUCEWITHMINVARIANCE(ω, m, B, σ²_min)
    σ²_start ← σ²(ω)
    t_prev ← createTransformation(f, m, B, D)
    η_prev ← ω
    for d := D − 1, . . . , 1 do
        // Erstellt die Transformationsfunktion wie oben beschrieben
        t ← createTransformation(f, m, B, d)
        // Erstellt das dimensionsreduzierte Surrogat ebenfalls wie beschrieben
        η ← reduceToDimension(f, B, d, t)
        σ²_now ← σ²_rem(ω, t, η)
        if σ²_now < σ²_min then
            return (η_prev, t_prev)
        else
            η_prev ← η
            t_prev ← t
        end if
    end for
    return (η_prev, t_prev)
end function
```

3.2. Dimensionsreduktion mithilfe der Hauptkomponentenanalyse

Um mithilfe der Hauptkomponentenanalyse, im englischen auch Principal Component Analysis (PCA) genannt, eine Eingabefunktion in ihrer Dimension zu reduzieren, wird die Eingabefunktion ω als Wahrscheinlichkeitsdichte interpretiert und dann aus dieser eine Anzahl an Samples gezogen. Mit der Hauptkomponentenanalyse werden dann Eigenvektoren und Eigenwerte der Kovarianzmatrix der gezogenen Samples berechnet. Die Eigenvektoren bilden dann in umgekehrter Reihenfolge die einzelnen Basisvektoren der Orthonormalbasis.

Um Samples aus einer beliebigen Funktion ω zu ziehen, muss diese in eine Wahrscheinlichkeitsdichtefunktion ω' umgewandelt werden:

Definition 3.2.1
Sei

$$\begin{aligned} \omega_+ \colon [0,1]^D &\to \mathbb{R}_+ \\ x &\mapsto \omega(x) - \min \omega \end{aligned}$$

eine positiv reelle Funktion und

$$\xi := \int_\Omega \omega_+(x)\, dx$$

das Integral dieser Funktion. Dann ist

$$\omega' \colon [0,1]^D \to \mathbb{R}_+$$
$$x \mapsto \begin{cases} \frac{\omega_+(x)}{\xi}, & \xi > 0 \\ 1, & \xi = 0 \end{cases}$$

eine positive reelle Funktion mit $\int_\Omega \omega'(x)\, dx = 1$*, also eine Wahrscheinlichkeitsdichtefunktion. Der zweite Fall für* $\xi = 0$ *ist nötig, da* ω *konstant sein kann und somit* $\omega'(x) = 0$ *ist.*

Um dann Samples aus ω', also einer potentiell beliebigen D-dimensionalen Wahrscheinlichkeitsdichtefunktion zu ziehen, wird der Metropolis-Hastings Algorithmus verwendet. Dieser wird in [Has70] genauer beschrieben und ein Pseudocode des Algorithmus ist in 3.2 zu sehen.

Algorithmus 3.2 Pseudocode des hier verwendeten Metropolis-Hastings-Algorithmus. Eingaben sind die Wahrscheinlichkeitsdichtefunktion ω', die Anzahl der Samples n, die Anzahl an Iterationen pro Sample i und die Standardabweichung der Schrittweite σ. Die Funktion $rand()$ zieht zufällig ein Element aus einer definierten Wahrscheinlichkeitsverteilung.

```
function metropolisHastings(ω', n, i, σ)
    for j := 1, ..., n do
        x ← rand(U(0,1))^D
        p ← ω'(x)
        for k := 1, ..., i do
            x_p ← (r_1)
            for d := 1, ..., D do
                r ← rand(N(0, σ^2))
                (x_p)_d ← (x_p)_d + r
                α ← ω'(x_p)/p
                if (α > 1) ∨ (rand(U(0,1)) < α) then
                    x ← x_p
                    p ← αp
                end if
            end for
        end for
    end for
end function
```

Der Algorithmus generiert ein Sample, indem er immer wieder zufällig ein neues Sample generiert und anschließend entscheidet, ob das neue Sample übernommen werden soll. Dies passiert, wenn das neue Sample eine höhere Wahrscheinlichkeit hat, gezogen zu werden oder mit einer geringen Wahrscheinlichkeit auch zufällig, um Stillstand zu verhindern.

Definition 3.2.2
Sei

$$\boldsymbol{X} \coloneqq \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_{n-1} \\ X_n \end{pmatrix} \tag{3.3}$$

ein n-dimensionaler Vektor an D-dimensionalen Zufallsvariablen, die aus der Wahrscheinlichkeitsdichtefunktion ω' gezogen wurden.

Nachdem nun die Samples gezogen wurden, werden die einzelnen Hauptkomponenten der gezogenen Zufallsvariablen ermittelt. Dafür müssen als erstes die einzelnen Zufallsvariablen bezüglich des Erwartungswertes zentriert werden.

Definition 3.2.3
Sei

$$\boldsymbol{Y} \coloneqq \boldsymbol{X} - E(\boldsymbol{X})$$

ein Vektor an Zufallsvariablen, der bezüglich des Erwartungswertes von $\boldsymbol{X}$ zentriert ist. Also gilt $E(\boldsymbol{Y}) = 0$ und

$$\Sigma \coloneqq \boldsymbol{Y}^T \boldsymbol{Y}$$

die Kovarianzmatrix von $\boldsymbol{Y}$. Diese ist eine symmetrische und positiv semidefinite $n \times n$-Matrix.

Definition 3.2.4
Sei $\Lambda = diag(\lambda_1, ..., \lambda_D)$ die Matrix der Eigenwerte von C.

Sei W eine orthogonale $(D \times D)$-Matrix mit den Spaltenvektoren $w_1, \dots, w_D$, welche die jeweiligen Eigenvektoren zu dem jeweiligen Eigenwert λ_i bilden:

$$W \coloneqq \begin{pmatrix} w_1 & w_2 & \dots & w_{D-1} & w_D \end{pmatrix}$$

Die Kovarianzmatrix Σ kann nun, da sie positiv semidefinit ist, wie folgt zerlegt werden:

$$\Sigma = W \Lambda W^T$$

Der i-te Spaltenvektor von W zeigt die Richtung der i-ten Hauptkomponenten an. Der Eigenwert λ_i der i-ten Hauptkomponente gibt die Varianz der gezogenen Samples entlang des i-ten Spaltenvektors an, also $Var(w_i) = \lambda_i$.

Aufgrund der Konstruktion der Wahrscheinlichkeitsdichte zeigen die so berechneten Hauptkomponenten aber genau die inaktiven Richtungen der ursprünglichen Funktion an. Denn besteht entlang eines Richtungsvektors einer Hauptkomponente eine hohe Varianz der gezogenen Samples, heißt das, dass die ursprüngliche Funktion entlang dieser Richtung keine große Änderungsrate besitzt. Daher muss die Reihenfolge der Hauptkomponenten invertiert werden, um die schon erwähnte Orthonormalbasis B zu konstruieren, die dann zur Dimemsionsreduktion verwendet werden kann.

Definition 3.2.5
Sei $\gamma_{PCA-Func}(w_i) = \lambda_{D-i}$ *die Gewichtungsfunktion und* $\gamma_{PCA-Func}(w_{i_1}) \geq \ldots \geq \gamma_{PCA-Func}(w_{i_D})$.

Dann ist

$$B_{PCA-Func} = \begin{pmatrix} w_{i_1} & w_{i_2} & \ldots & w_{i_{D-1}} & w_{i_D} \end{pmatrix}$$

die neue Basis, die zur Reduktion verwendet wird.

Definition 3.2.6
Sei

$$m_{PCA-Func} = E(\boldsymbol{X})$$

der Stützvektor des Projektionsraumes.

3.3. Active Subspaces

Die Methode der Active Subspaces, wie in [PC14] beschrieben, versucht wichtige, d.h. aktive, Richtungen in dem Parameterraum zu finden. Dafür werden die Richtungen der größten Änderungen gesucht. Der Active Subspace besteht aus aus den Eigenvektoren mit den größten dazugehörenden Eigenwerten einer Matrix, die das durschnittliche äußere Produkt des Gradienten ist.

Definition 3.3.1
Sei

$$\rho \colon [0,1]^D \to \mathbb{R}_+, \int_\Omega \rho \, dx = 1$$

eine Wahrscheinlichkeitsdichtefunktion, die die Verteilung der Input-Parameter beschreibt.

Die Funktion ρ wird der Vollständigkeit halber aufgeführt, da sie in den klassischen Active-Subspace-Methoden verwendet wird. Diese wird aber in dieser Arbeit nicht verwendet, da alle anderen Methoden nichts Derartiges besitzen und somit nur der Vergleich mit den anderen Methoden komplizierter werden würde. Wir nehmen daher an, dass alle Parameter uniform verteilt sind.

Definition 3.3.2
Sei

$$C \coloneqq \int_\Omega (\nabla\omega)(\nabla\omega)^T \rho \, dx$$

das durschnittliche äußere Produkt des Gradienten.

Die Matrix C kann auf verschiedene Weisen berechnet werden. In dieser Arbeit werden als zwei Alternativen eine einfache Monte-Carlo-Berechnung aus [PC14] und eine Quadratur-Berechnung mit Dünnen Gittern verwendet. Für Letztere wird zuerst die Matrix $(\nabla\omega)(\nabla\omega)^T$ aufgestellt und anschließend jedes Element der Matrix, also jede Kombination an Richtungsableitungen, mihtilfe eines Dünnen Gitters interpoliert und anschließend mittels einer Quadraturmethode auf diesem Dünnen Gitter das Integral des Matrixelements berechnet.

Definition 3.3.3
Sei $\Lambda = diag(\lambda_1, ..., \lambda_D)$ die Matrix der Eigenwerte von C.

Sei W eine orthogonale $(D \times D)$-Matrix mit den Spaltenvektoren $w_1, \ldots, w_D$, welche die jeweiligen Eigenvektoren zu dem jeweiligen Eigenwert λ_i bilden:

$$W := \begin{pmatrix} w_1 & w_2 & \ldots & w_{D-1} & w_D \end{pmatrix}$$

Die Matrix C kann nun, da sie positiv semidefinit ist, wie folgt zerlegt werden:

$$C = W\Lambda W^T$$

Definition 3.3.4
Sei $\gamma_{AS}(w_i) = \lambda_i$ die Gewichtungsfunktion und $\gamma_{AS}(w_{i_1}) \geq ... \geq \gamma_{AS}(w_{i_D})$.

Dann ist

$$B_{AS} = \begin{pmatrix} w_{i_1} & w_{i_2} & \ldots & w_{i_{D-1}} & w_{i_D} \end{pmatrix}$$

die neue Basis, die zur Reduktion verwendet wird.

Zur Konstruktion des Projektionsraumes wird noch der Stützvektor m benötigt. Die klassische Active-Subspace-Methode berechnet aber nur aktive Richtungen und somit wird der gleiche Stützvektor wie bei der Hauptkomponentenmethode verwendet.

Definition 3.3.5
Sei

$$m_{AS} = m_{PCA-Func}$$

der Stützvektor des Projektionsraumes.

Der Stützvektor kann auch anders berechnet werden, jedoch läuft es am Ende darauf hinaus, dass dafür eine Form von Monte-Carlo-Sampling verwendet wird.

4. Implementation und Vergleich

Teil dieser Arbeit war es, alle beschriebenen Dimensionsreduktionsmethoden in das SG++ Framework zu implementieren. Da SG++ bereits viele Funktionalitäten, wie die Erstellung, Hierarchisierung und generelle Verwendung von dünnen Güttern enthält, musste so nur noch Funktionalität wie Dünne ANOVA-Gitter und die Prewavelet-Basis hinzugefügt werden. Dies gestaltete sich etwas kompliziert, da sich Dünne ANOVA-Gitter in vielen Aspekten fundamental anders verhalten als normale Dünne Gitter oder auch Dünne Gitter mit Randunterstützung. Sie unterscheiden sich in der Konstruktion der Punktemenge und daher unterscheiden sich alle Algorithmen auf Dünnen ANOVA-Gittern von denen für normale Dünne Gitter.

Bevor die hier vorgestellen Methoden zur Dimensionsreduktion miteinander bei Anwendung auf verschiedene Modelle verglichen werden können, um zu entscheiden, welche der Methoden sich für eine bestimmte Problemstellung am besten eignet, müssen die Eigenheiten jeder Methode berücksichtigt werden. Das heißt, dass bestimmte Problemstellungen in der Auswahl der Methode durch ihre Eigenheiten beschränkt sind.

Ein Aspekt ist die Achsenorientierung des Modells. Sind aktive Richtungen nicht entlang der Achsen des Koordinatensystems ausgerichtet, kann mit den ANOVA-Methoden kein befriedigendes Ergebnis erzielt werden. Es sollte entweder eine andere Methode verwendet werden, oder das Modell durch eine Modifizierung entlang den Achsen ausgerichtet werden.

Ein weiterer Aspekt der achsenfreien Methoden ist, dass die Modellfunktion zur Konstruktion des dimensionsreduzierten Surrogats im Gegensatz zu den ANOVA-Methoden oft außerhalb ihres ursprünglichen Definitionsbereichs ausgewertet werden muss. Daher muss bei Verwendung einer achsenfreien Methode die Modellfunktion so erweitert werden, dass sie auch außerhalb von Ω akkurat ausgewertet werden kann.

Insgesamt werden fünf Methoden betrachtet: ANOVA, Anchored-ANOVA, PCA, AS-MC und AS-QUAD. Mit PCA ist hier die auf der Hauptkomponenten basierende Methode gemeint. AS-MC ist die Active-Subspace-Methode, bei der die Active-Subspace-Matrix C mithilfe einer Monte-Carlo-Methode berechnet wird. AS-QUAD ist ebenfalls eine Active-Subspace-Methode, bei der die einzelnen Einträge der Active-Subspace-Matrix C mithilfe von Quadraturverfahren auf Dünnen Gittern berechnet werden. Mit ANOVA ist die klassische ANOVA-Methode gemeint.

Eine Fehlerquelle bei der PCA-Methode ist das Sampling der Funktion mithilfe des Metropolis-Hastings-Algorithmus. Die Anzahl an Samples und Iterationen pro Sample sollten, wenn sie groß genug gewählt sind, die Ergebnisse nicht sehr stark beeinflussen, jedoch kann die Schrittweite σ die Qualität beeinflussen. Daher werden die PCA-Ergebnisse für verschiedene Werte für σ untersucht. Für die Active-Subspaces-Methoden, die den Gradienten mithilfe eines Differenzenquotienten berechnen, werden ebenfalls unterschiedliche Parameter untersucht. Die Anzahl der Samples der AS-MC-Methode beträgt in diesem Kapitel immer 10000. Für die Anchored-ANOVA-Methode werden zum besseren Vergleich verschiedene Anker verwendet.

4.1. Strategien

Bei der Dimensionsreduktion werden hier zwei verschiedene Strategien verwendet, nämlich die statische- und dynamische Dimensionsreduktionsstrategie. Die statische Strategie eignet sich für Funktionen, bei denen bereits bekannt ist, auf welche Dimension die Funktion ohne größeren Fehler reduziert werden kann. Bei dieser wird einfach angegeben, wieviele Dimensionen d erhalten werden sollen. Anschließend werden die inaktivsten $D - d$ Dimensionen ohne Rücksicht auf die verlorene Gesamtvarianz entfernt. Diese Strategie wird außerdem im Vergleichskapitel verwendet um ausschließlich den Fehler, den verschiedene Methoden bei der Reduktion auf eine einheitliche Dimension hervorrufen, zu vergleichen.

Die dynamische Strategie, welche als Eingabe den minimalen Anteil der Gesamtvarianz, die erhalten werden soll, σ^2_{min} bekommt, ist die überall in dieser Arbeit implizit beschriebene Strategie. Diese eignet sich für alle Eingabefunktionen, bei denen die kleinste akzeptable Dimension noch nicht bekannt ist und diese somit dynamisch bestimmt werden muss. Alle bereits gezeigten Algorithmen verwenden die dynamische Strategie und es ist trivial, diese Algorithmen in statische Algorithmen umzuwandeln.

4.2. Vergleich der achsenfreien Methoden

Um zunächst die achsenfreien Methoden miteinander zu vergleichen, wird folgende zweidimensionale Beispielfunktion definiert:

$$f_1(x, y) = max(1 - |5x' - 2.5|, 0),\ x' = x\cos(0.15\pi) - y\sin(0.15\pi)$$

Diese Funktion ist eine ursprünglich eindimensionale Hütchenfunktion, die um ca. 27 Grad rotiert wurde. Sie ist so gewählt, dass eine achsenfreie Methode erkennen sollte, dass die Funktion auf eine eindimensionale Funktion reduzierbar ist. Somit wird hier eine statische Dimensionsreduktion durchgeführt mit $D = 2$ und $d = 1$. Die beiden ANOVA-Methoden werden hier nicht angewendet, da sie sich, wie schon erwähnt, für Probleme dieser Art nicht eignen und somit auch keine gute Dimensionsreduktion durchführen könnten.

Definition 4.2.1
Sei l das Level des Dünnen Boundary-Gitters. Dann ist in diesem Abschnitt

$$\omega_1(x) = \mathbb{S}^{Boundary}_{f_1,l}(x)$$

das Surrogat für die ursprüngliche Modellfunktion, welches mithilfe eines Dünnen Boundary-Gitters mit Level l konstruiert wurde.

Da in der Praxis, wie schon vorher erwähnt, die Modellfunktion nicht so einfach wie in diesem Beispiel auszuwerten ist und meistens nur ein Surrogat konstruiert werden kann, wird hier ebenfalls ein Surrogat für f_1 erstellt, nämlich ω_1, wobei das Level l, und somit auch die Anzahl der Gitterpunkte variiert. In diesem Beispiel werden für das Dünne Gitter die Level $l \in \{0, \ldots, 10\}$ untersucht.

Ein Problem, welches hier auftaucht, ist, dass die Funktion f_1 nicht besonders gut mithilfe eines Dünnen Gitters und dessen Basisfunktionen interpoliert werden kann, da die Gitterpunkte auch entlang der einzelnen Achsen ausgerichtet sind und somit die achsenfreie Funktion f_1 nicht entlang

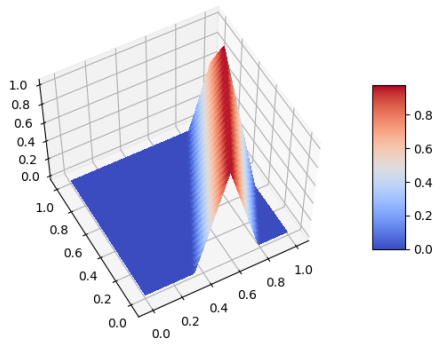

Abbildung 4.1.: Die Funktion f_1.

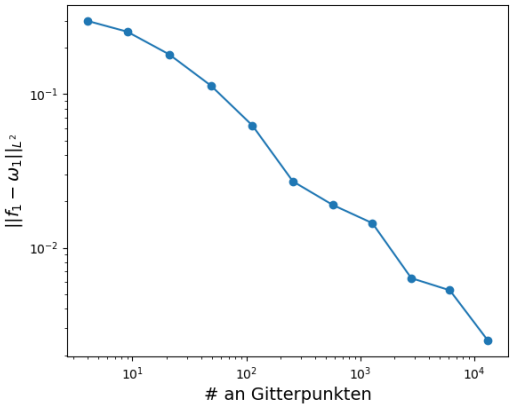

Abbildung 4.2.: Interpolationsfehler des Surrogats ω_1 für $l = 0, \ldots, 10$.

ihres Verlaufs auswerten. Der L^2 Fehler, wie er in Abbildung 4.2 zu sehen ist, ist zwar akzeptabel, jedoch muss im Hinterkopf behalten werden, dass sich dieser Fehler im Dimensionsreduktionsprozess eventuell verstärken kann. Die Auswirkungen des Interpolationsfehlers werden später betrachtet.

Außerdem kann die Entwicklung des Projektionsraumes über die verschiedenen Level des Dünnen Gitters betrachtet und auch methodenübergreifend verglichen werden. Der Projektionsraum wird mithilfe von (3.1) konstruiert und wie zu sehen ist, sind Teile des Projektionsraumes außerhalb des Definitionsbereiches Ω der Modellfunktion f_1. Dies bereitet bei dieser Beispielfunktion jedoch keine Probleme, da f_1 in ganz $\mathbb{R}^2$ ausgewertet werden kann und außerdem korrekte Werte außerhalb von Ω für eine möglichst genaue Konstruktion des dimensionsreduzierten Surrogats η liefert (In diesem Fall ist es immer der Wert 0).

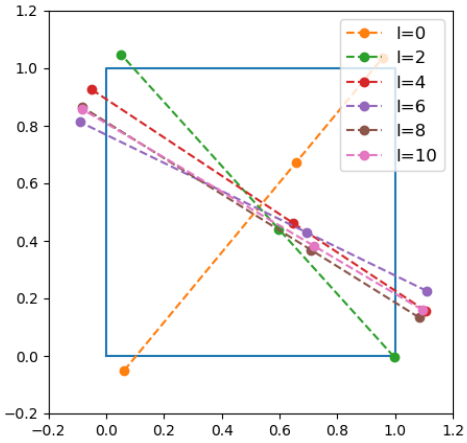

Abbildung 4.3.: Projektionsräume der PCA-Methode mit $l = 0, 2, \ldots, 10$.

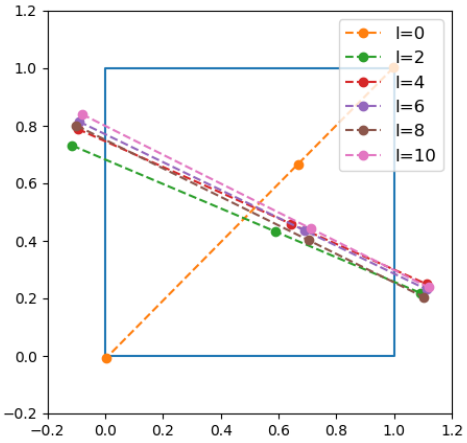

Abbildung 4.4.: Projektionsräume der AS-MC-Methode mit $l = 0, 2, \ldots, 10$.

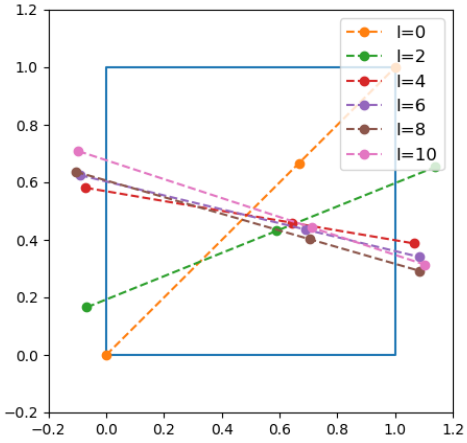

Abbildung 4.5.: Projektionsräume der AS-QUAD-Methode mit $l = 0, 2, \ldots, 10$.

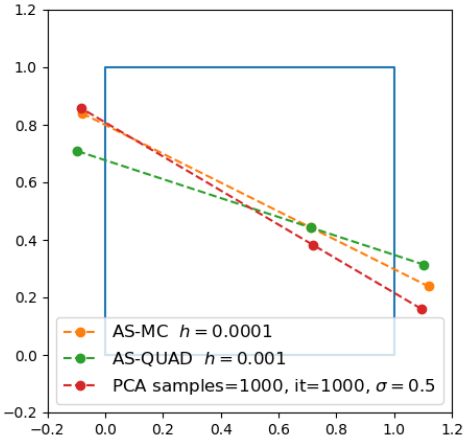

Abbildung 4.6.: Projektionsräume der drei Methoden für $l = 10$.

Um die Übersicht zu behalten wurden nur die Projektionsräume für jedes zweite Level aufgezeichnet. Wie bei allen Methoden zu sehen ist, sind für kleine Level durch einen großen Interpolationsfehler von ω_1, wie in Abbildung 4.2 zu erkennen ist, die Projektionsräume nicht zu gebrauchen. Für größere Level konvergieren die Projektionsräume alle gegen den methodenspezifischen Projektionsraum, jedoch mit unterschiedlicher Geschwindigkeit. Außerdem unterscheiden sich, wie in Abbilding 4.6 zu sehen ist, die konstruierten Projektionsräume geringfügig voneinander. Die AS-QUAD-Methode konvergiert eher langsam, während die AS-MC- und PCA-Methode schneller konvergieren. Wie in Abbildung 4.5 zu sehen ist, schwankt selbst für größere Level im Vergleich zu den zwei anderen Methoden der Projektionsraum noch ziemlich stark.
Diese Tatsachen schlagen sich schlussendlich auch im Fehler der reduzierten Funktion nieder, der als nächstes untersucht wird, um zu sehen, wie die verschiedenen Methoden unter realen Bedingungen für die Beispielfunktion abschneiden. Da der Dimensionsreduktionsprozess nur mit dem interpolierten Surrogat ω_1 arbeitet, kann sich der Interpolationsfehler, der in Abbildung 4.2 zu sehen ist, hierbei verstärken.

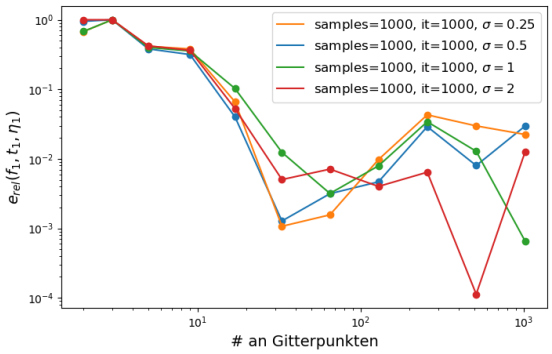

Abbildung 4.7.: Relativer Fehler der PCA-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen MH-Parametern.

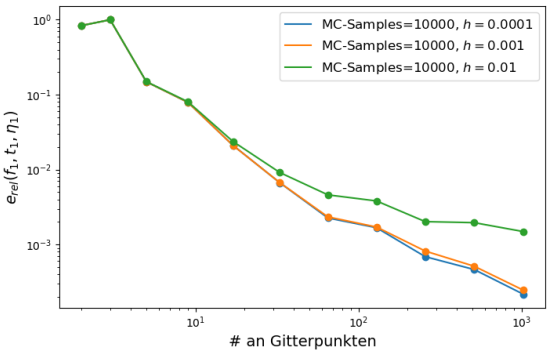

Abbildung 4.8.: Relativer Fehler der AS-MC-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen Differenzenquotienten.

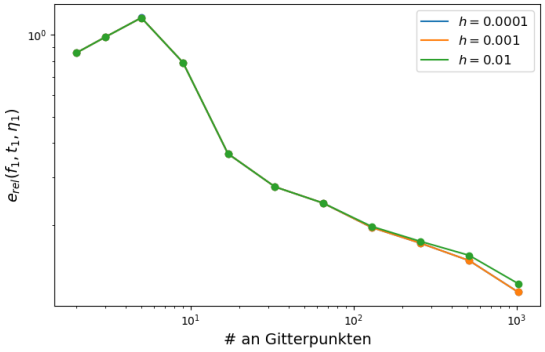

Abbildung 4.9.: Relativer Fehler der AS-QUAD-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen Differenzenquotienten.

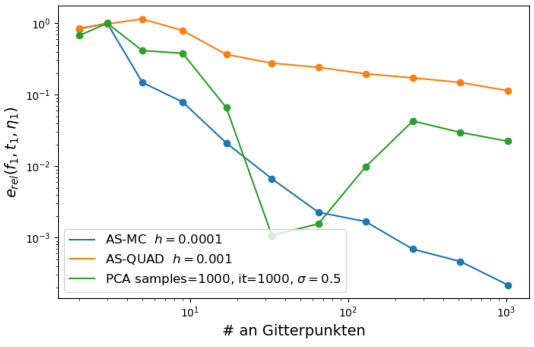

Abbildung 4.10.: Relativer Fehler der drei Methoden mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und den jeweils besten Parametern.

Wie in Abbildung 4.7 zu sehen ist, schwankt der relative Fehler bei der PCA-Methode stark im Gegensatz zu den beiden Active-Subspace-Methoden in Abbildung 4.8 und 4.9, bei denen es nur sehr kleine Schwankungen im Fehler gibt. In Abbildung 4.10 erkennt man gut, dass die AS-MC-Methode hinsichtlich der verlorenen Gesamtvarianz am besten abschneidet. Erst danach folgen die PCA-Methode und die AS-QUAD-Methode, welche vergleichsweise große Fehler produzieren.
Da in diesem Beispiel die gesamte Modellfunktion sehr schnell auswertet werden kann, ist man in der Lage ebenfalls zu untersuchen, wie die Methoden ohne den Interpolationsfehler abschneiden. Folglich werden hier die Methoden direkt auf die Modellfunktion angewendet, d.h. $\omega_1 = f_1$ gesetzt. Somit wird der Interpolationsfehler bei der Berechnung eliminiert und die Methoden können auf ihre Stabilität überprüft werden.

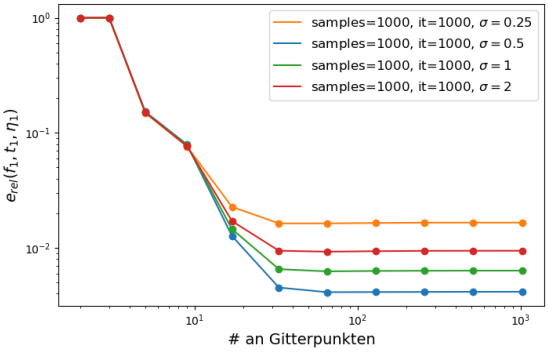

Abbildung 4.11.: Relativer Fehler der PCA-Methode mit exakter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen MH-Parametern.

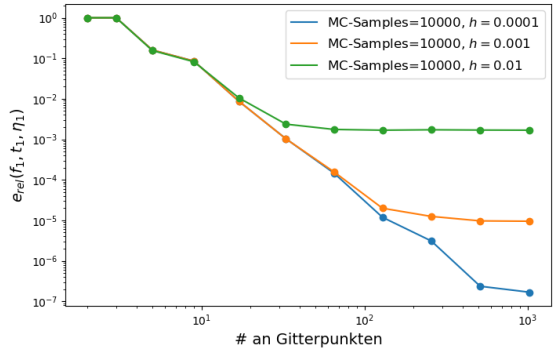

Abbildung 4.12.: Relativer Fehler der AS-MC-Methode mit exakter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen Differenzenquotienten.

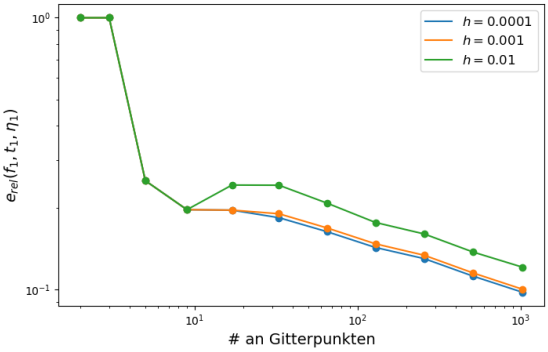

Abbildung 4.13.: Relativer Fehler der AS-QUAD-Methode mit exakter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen Differenzenquotienten.

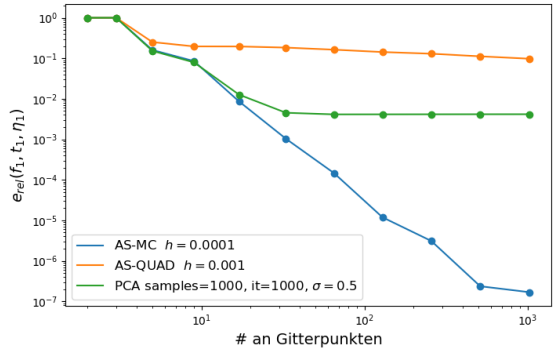

Abbildung 4.14.: Relativer Fehler der drei Methoden mit exakter Eingabefunktion für $l = 0, 2, \ldots, 10$ und den jeweils besten Parametern.

Bei der PCA-Methode existieren hier keine Schwankungen mehr. Stattdessen konvergiert die Methode relativ schnell gegen einen relativen Fehlerwert. Im Vergleich zu der AS-MC-Methode in Abbildung 4.14 ist dieser Fehlerwert aber viel größer. Der Grund für das schlechte Abschneiden der PCA-Methode ist auf die Art und Weise zurückzuführen, wo und wie Samples mit dem Metropolis-Hastings-Algorithmus gezogen werden. Es werden nur im Definitionsbereich Samples gezogen und dadurch wird, wie in Abbildung 4.1 zu sehen ist, das Hütchen am Rand von Ω schräg abgeschnitten.

Daher wird die Hütchenfunktion in der Randregion des Definitionsbereiches nur an einer Seite des Huts gesampled und somit werden leicht verfälschte Hauptkomponenten berechnet. Dieses Problem lässt sich im Prinzip auch nicht lösen, da nicht einfach ermittelt werden kann, wo die Eingabefunktion am Rand abgeschnitten wird. Daher muss dies bei der Anwendung der PCA-Methode beachtet werden.

Bei der PCA-Methode existieren hier keine Schwankungen mehr. Stattdessen konvergiert die Methode relativ schnell gegen einen relativen Fehlerwert. Im Vergleich zu der AS-MC-Methode in Abbildung 4.14 ist dieser Fehlerwert aber viel größer.

Der Grund für das schlechte Abschneiden der PCA-Methode ist auf die Art und Weise zurückzuführen, wo und wie Samples mit dem Metropolis-Hastings-Algorithmus gezogen werden. Es werden nur im Definitionsbereich Samples gezogen und dadurch wird, wie in Abbildung 4.1 zu sehen ist, das Hütchen am Rand von Ω schräg abgeschnitten. Daher wird die Hütchenfunktion in der Randregion des Definitionsbereiches nur an einer Seite des Huts gesampled und somit werden leicht verfälschte Hauptkomponenten berechnet. Dieses Problem lässt sich im Prinzip auch nicht lösen, da nicht einfach ermittelt werden kann, wo die Eingabefunktion am Rand abgeschnitten wird. Daher muss dies bei der Anwendung der PCA-Methode beachtet werden.

Die AS-MC-Methode schneidet sehr gut ab, da sie nicht das Problem der PCA-Methode besitzt und bei dieser Beispielfunktion sehr einfach die Richtung der größten Änderung mithilfe fast exakter Gradienten durch ein sehr kleines h finden kann.

Die AS-QUAD-Methode schneidet auch hier sehr schlecht ab. Dies kann auf den etwas anderen Projektionsraum der AS-QUAD-Methode im Vergleich zur AS-MC-Methode zurückgeführt werden. Diese Abweichung liegt darin begründet, dass die einzelnen Richtungsableitungen der Funktion f_1 sich nicht sehr gut mithilfe eines dünnen Gitters interpolieren lassen und somit die Quadratur auf bereits fehlerbehafteten Daten ausgeführt wird. Diese kleine Abweichung verstärkt sich hier, da die Hütchenfunktion f_1 sehr steil ist und je größer die Änderungsrate der Modellfunktion entlang des Projektionsraumes ist, desto stärker fallen schon kleine Abweichungen bei der Berechnung der Basis zur Dimensionsreduktion ins Gewicht.

4.3. Vergleich aller Methoden

Um jetzt alle fünf Methoden miteinander zu vergleichen, wird folgende zweidimensionale Beispielfunktion verwendet:

$$f_2(x, y) = \sin(\frac{3}{2}\pi x)e^{-\frac{1}{2}y}$$

Diese Funktion ist, wie in Abbildung 4.15 zu sehen, entlang der Achsen ausgerichtet. Daher können hier zusätzlich die beiden ANOVA-Methoden angewendet werden. Jedoch ist diese Funktion keine echte eindimensionale Funktion und somit kann man diese Funktion nicht ohne Fehler reduzieren. Es wird ebenfalls eine statische Dimensionsreduktion durchgeführt mit $D = 2$ und $d = 1$. Für alle anderen Methoden muss wieder, wie auch im ersten Beispiel, ein Surrogat verwendet werden. Demnach wird hier ebenfalls ein Surrogat für f_2 erstellt, nämlich ω_2. Dafür wird die Funktion f_2 mithilfe eines dünnen Boundary-Gitters mit hierarchischer linearer Basis interpoliert, wobei das Level l, und somit auch die Anzahl der Gitterpunkte variiert. In diesem Beispiel werden für das Dünne Gitter wieder die Level $l \in \{0, \ldots, 10\}$ untersucht.

Definition 4.3.1
Sei

$$\omega_2(x) = \mathbb{S}^{Boundary}_{f_2,l}(x)$$

das Surrogat für f_2, welches mithilfe der hierarchischen linearen Basis auf einem Dünnen Boundary-Gitter mit Level l konstruiert wurde. Sei außerdem

$$\omega_2^{ANOVA}(x) = \mathbb{S}^{ANOVA}_{f_2,l}(x)$$

das Surrogat für die ursprüngliche Modellfunktion f_2, welches mithilfe der hierarchischen Prewavelet-Basis auf einem Dünnen ANOVA-Gitter mit Level l konstruiert wurde und

$$\omega_2^{Anchor}(x) = \mathbb{S}^{Anchor}_{f_2,l,a}(x)$$

das Surrogat für die ursprüngliche Modellfunktion f_2, welches mithilfe der hierarchischen linearen Basis auf einem Dünnen ANOVA-Gitter mit Level l und Anker a konstruiert wurde.

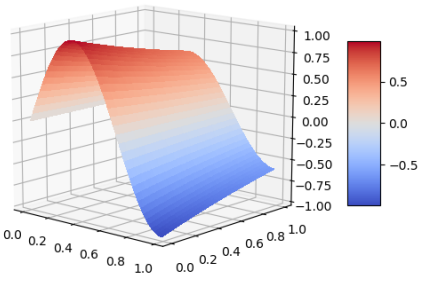

Abbildung 4.15.: Die Funktion f_2.

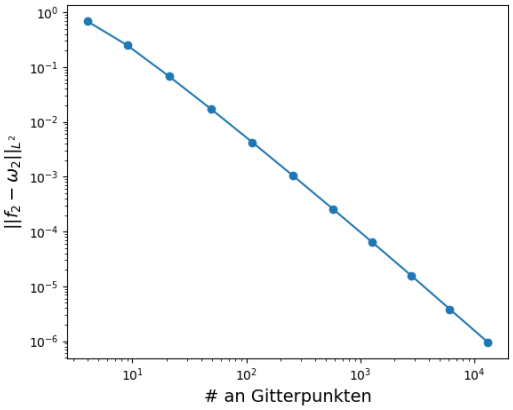

Abbildung 4.16.: Interpolationsfehler des Surrogats ω_2 für $l = 0, \ldots, 10$.

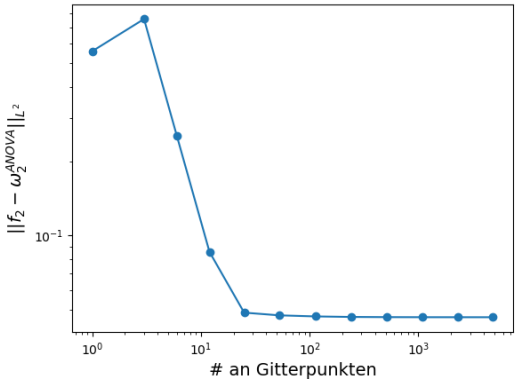

Abbildung 4.17.: Interpolationsfehler des Surrogats ω_2^{ANOVA} für $l = -1, \ldots, 10$.

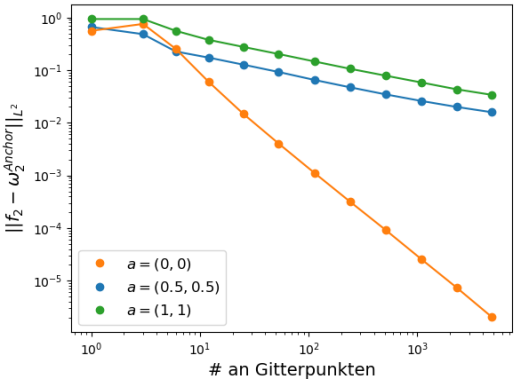

Abbildung 4.18.: Interpolationsfehler des Surrogats ω_2^{Anchor} für $l = 0, \ldots, 10$ und verschiedene Anker

Im Vergleich zu ersten Beispielfunktion ist der L^2 Fehler, wie er in Abbildung 4.16 zu sehen ist, viel geringer als der Fehler für die erste Beispielfunktion, da die Funktion f_2 durch ihre Orientierung entlang der Achsen viel besser zu interpolieren ist. Weil der Fehler für höhere Level sehr gering ist, fällt der Unterschied zwischen Ergebnissen, die die achsenfreien Methoden auf dem Surrogat ω_2 und der orginalen Funktion f_2 berechnen, sehr gering aus und wird aus diesem Grund hier ausgeblendet. Stattdessen wird der Fokus primär auf das Abschneiden im Vergleich zu den ANOVA-Methoden gelegt.

Die ANOVA-Methoden arbeiten ebenfalls mit Surrogaten, nur werden hier Dünne ANOVA-Gitter verwendet. Der Interpolationsfehler der klassischen ANOVA-Methode ist ziemlich groß, da die verwendete Prewavelet-Basis sich nicht sonderlich gut zur genauen Interpolation einer Funktion eignet. Bei der Anchored-ANOVA-Methode zeigt sich ein großer Unterschied bei den Interpolationsfehlern unter Verwendung verschiedener Anker. Auf die Tatsache, dass ein anderer Anker als der Nullanker schlechte Interpolationsergebnisse liefert, wird in dem nächsten Beispiel genauer eigegangen.

Die Entwicklung des Projektionsraumes über die verschiedenen Level des Dünnen Gitters kann hier nur für die achsenfreien Methoden betrachtet werden, da die ANOVA-Methoden nicht mit Projektionsräumen arbeiten.

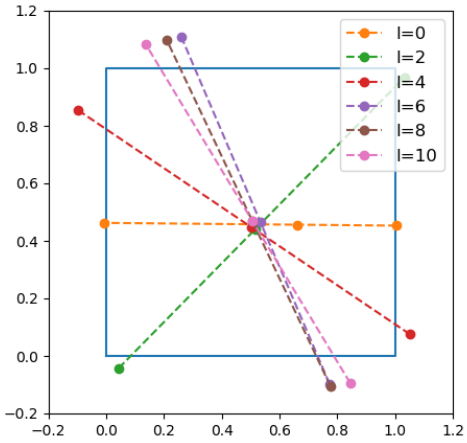

Abbildung 4.19.: Projektionsräume der PCA-Methode mit $l = 0, 2, \ldots, 10$.

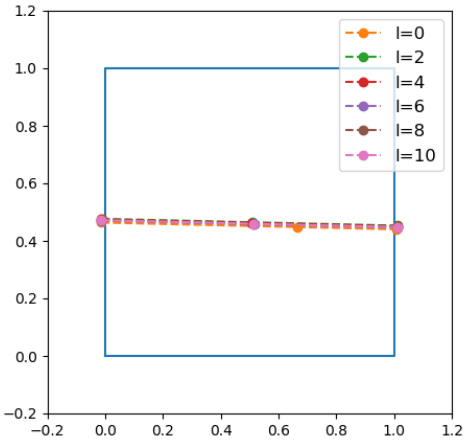

Abbildung 4.20.: Projektionsräume der AS-MC-Methode mit $l = 0, 2, \ldots, 10$.

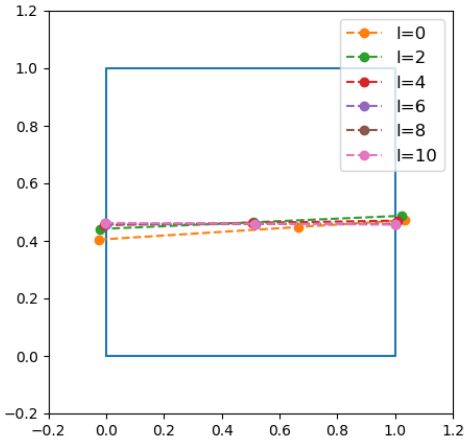

Abbildung 4.21.: Projektionsräume der AS-QUAD-Methode mit $l = 0, 2, \ldots, 10$.

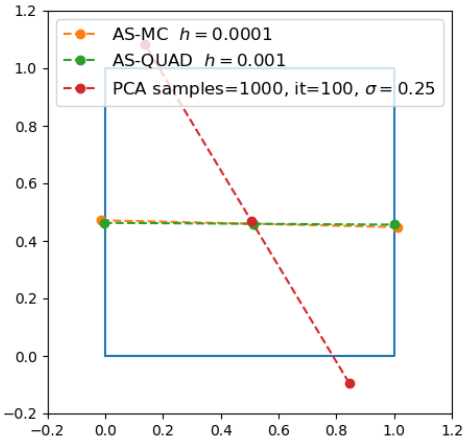

Abbildung 4.22.: Projektionsräume der drei Methoden für $l = 10$.

Wie in Abbildung 4.19 zu sehen ist, konstruiert hier die PCA-Methode ganz andere Projektionsräume im Vergleich zu den Active-Subspace-Methoden, welche hier fast identische Projektionsräume erstellen. Dieser starke Unterschied sollte sich dann ebenfalls in der anschließenden Fehlerauswertung zeigen.

Nun kann erneut untersucht werden, wie die verschiedenen Methoden unter realen Bedingungen für die Beispielfunktion abschneiden. Da der Dimensionsreduktionsprozess nur mit dem interpolierten Surrogaten ω_2 arbeitet, müssen wieder die Interpolationsfehler beachtet werden.

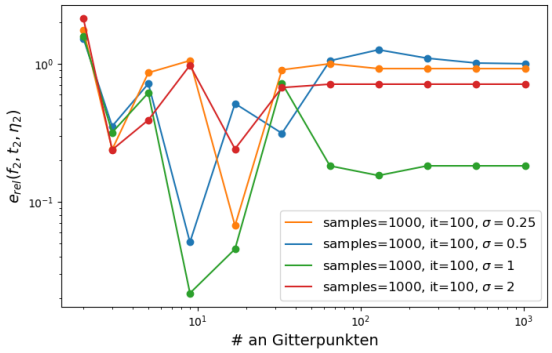

Abbildung 4.23.: Relativer Fehler der PCA-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen MH-Parametern.

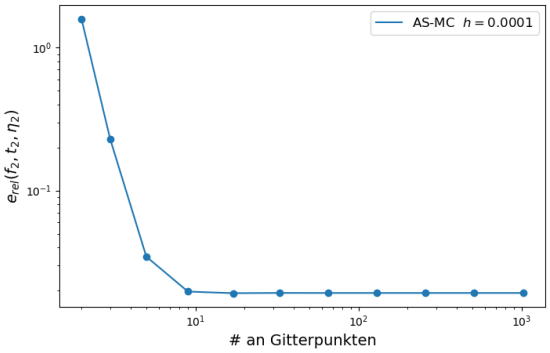

Abbildung 4.24.: Relativer Fehler der AS-MC-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen Differenzenquotienten.

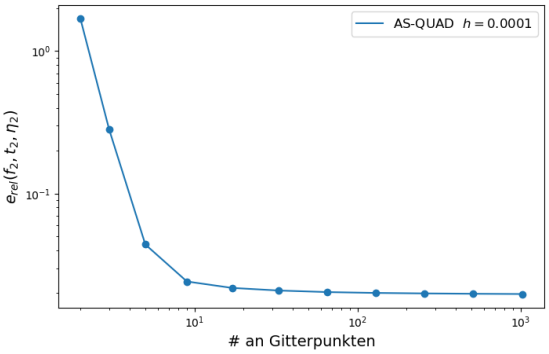

Abbildung 4.25.: Relativer Fehler der AS-QUAD-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen Differenzenquotienten.

Abbildung 4.26.: Relativer Fehler der klassischen ANOVA-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$.

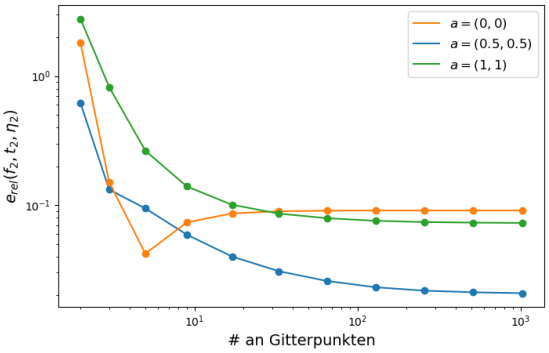

Abbildung 4.27.: Relativer Fehler der Anchored-ANOVA-Methode mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und verschiedenen Ankern.

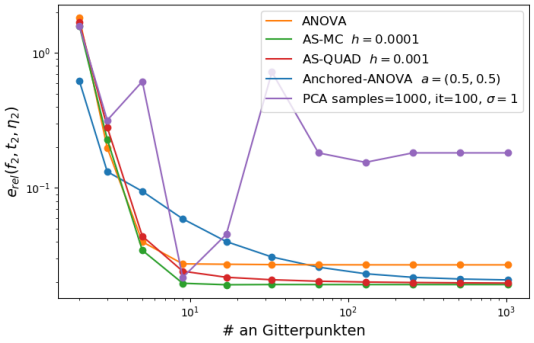

Abbildung 4.28.: Relativer Fehler aller fünf Methoden mit interpolierter Eingabefunktion für $l = 0, 2, \ldots, 10$ und den jeweils besten Parametern.

Wie man beim Vergleich der Methoden erkennen kann, ist die PCA-Methode für diese Funktion komplett unbrauchbar. Die beiden AS-Methoden liefern ziemlich gleichwertige Ergebnisse. Im Gegensatz zum ersten Beispiel ist hier die AS-QUAD-Methode genauso gut wie die AS-MC-Methode. Dies ist auf die Tatsache zurückzuführen, dass hier die einzelnen Richungsableitungen der Funktion ziemlich genau mit einem Dünnen Gitter interpoliert werden können und daher die Quadraturergebnisse ebenfalls sehr genau sind.

In Abbildung 4.28 ist zu sehen, dass sich alle Methoden außer der PCA-Methode gegen einen relativen Fehlerwert konvergieren.

4.4. Anker bei der dynamischen Dimensionsreduktion

Die Verwendung von verschiedenen Ankerpunkten bei der Anchored-ANOVA-Methode ermöglicht es, die gleiche Funktion auf verschiedene Arten zu zerlegen. In den vorherigen Beispielen, in denen die Funktion auf genau eine Dimension reduziert wurde, verändert der Anker nur die Genauigkeit des Surrogats geringfügig. Bei der dynamischen Dimensionsreduktion ist die richtige Wahl des Ankers jedoch sehr wichtig, wie mit folgender Beispielfunktion demonstriert wird:

$$f_3(x, y) = \min(x/0.05, 1)\min(y/0.05, 1)(1 + 0.01\sin(4\pi x))$$

Wie in Abbildung 4.29 zu sehen ist, sind die Funktionswerte entlang der X-Achse und Y-Achse gleich Null. Das führt dazu, dass unter Verwendung des Standardankers $a = (0, 0)$ die Gesamtvarianz der Funktion komplett durch die ANOVA-Komponente $(1, 1)$ ausgedrückt wird. Wählt man hingegen einen Anker, wie z.B $a = (0.0, 0.5)$ kann dieses Problem vermieden werden und der große Anteil der Gesamtvarianz der ANOVA-Komponente $(1, 0)$ zugeordnet werden, wie in Abbildung 4.30 zu sehen ist.

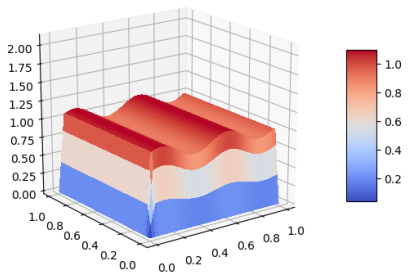

Abbildung 4.29.: Die Funktion f_3.

Anker a=(0.0,0.0)

	$c_1 = 0$	$c_1 = 1$
$c_0 = 0$	0.0	0.0
$c_0 = 1$	0.0	1.0

Anker a=(0.0,0.5)

	$c_1 = 0$	$c_1 = 1$
$c_0 = 0$	0.0	0.0
$c_0 = 1$	0.985	0.015

Abbildung 4.30.: Relative Verteilung der Gesamtvarianz auf die einzelnen ANOVA-Komponenten für die zwei Anker

Dies ist zwar ein ziemlich extremes Beispiel, jedoch verdeutlicht es sehr gut die Bedeutung der Wahl des Ankers. Das Problem dabei ist nur, dass nicht einfach ermittelt werden kann, welcher Anker optimale Ergebnisse lierfert. Also müssen verschiedene Anker, von denen angenommen wird, dass sie sich gut eignen, ausprobiert werden, was den Dimensionsreduktionsprozess langwieriger gestaltet.

4.5. Dynamische Dimensionsreduktion

Um das Abschneiden der Methoden bei der dynamischen Dimensionsreduktion miteinander zu vergleichen, wird folgende siebendimensionale Beispielfunktion definiert:

$$f_4(x_0, \ldots, x_6) = 2x_0x_1 + 3x_2 - e^{1.5x_3} + \sin(\pi x_4) + 0.01x_5 + e^{-0.1x_6}$$

Diese besteht aus vielen Termen, wobei jeder Term die Funktion unterschiedlich stark beeinflusst. Mithilfe der Methoden wird nun versucht, diese Funktion dynamisch in ihrer Dimension zu reduzieren, d.h. man probiert, einen minimalen Varianzanteil zu bewahren. In der Praxis, wie z.B. in [Hol08], wird als akzeptabler Anteil der verbleibenden Gesamtvarianz oft $\sigma^2_{min} = 0.99$ verwendet. Um zu untersuchen, wie sich verschiedene minimale Varianzanteile auf die Ergebnisse auswirken, werden hier die Methoden für $\sigma^2_{min} \in \{0.90, 0.95, 0.99\}$ gestestet. Außerdem wird für das Level der Surrogate $l = 8$ verwendet.

	$\sigma^2_{min} = 0.90$		$\sigma^2_{min} = 0.95$		$\sigma^2_{min} = 0.99$	
	R	σ^2_{rem}	R	σ^2_{rem}	R	σ^2_{rem}
Anchored-ANOVA $a = \underline{0}$	{0, 5, 6}	0.925	{5, 6}	0.999	{5,6}	0.999
Anchored-ANOVA $a = \underline{0.5}$	{0, 1, 4, 5, 6}	0.917	{0, 1, 5, 6}	0.962	{5,6}	0.999
Anchored-ANOVA $a = \underline{1}$	{4, 5, 6}	0.964	{4, 5, 6}	0.967	{5,6}	0.999
ANOVA	{0, 1, 4, 5, 6}	0.908	{4, 5, 6}	0.974	{5, 6}	0.999

Abbildung 4.31.: Ergebnisse der ANOVA-Methoden für $l = 8$ und verschiedene minimale Varianzanteile für die Funktion f_4. Für jedes Ergebnis werden die Indizes der entfernen Dimensionen R und der verbleibende Anteil der Varianz σ^2_{rem} aufgelistet.

Wie in Abbildung 4.31 zu sehen ist, variieren die Ergebnisse vor allem für kleinere σ^2_{min}. Während für $\sigma^2_{min} = 0.99$ für jede Methode die gleichen Parameter entfernt werden, kann für $\sigma^2_{min} = 0.9$ die Dimension auf vier und im besten Fall auf sogar zwei reduziert werden. Es zeigt sich wieder, dass die richtige Wahl des Ankers eine entscheidende Rolle spielt

	$\sigma^2_{min} = 0.90$		$\sigma^2_{min} = 0.95$		$\sigma^2_{min} = 0.99$	
	d	σ^2_{rem}	d	σ^2_{rem}	d	σ^2_{rem}
PCA	6	0.913	7	1.0	7	1.0
AS-MC	2	0.962	2	0.962	3	0.992
AS-QUAD	4	0.958	4	0.958	5	0.999

Abbildung 4.32.: Ergebnisse der restlichen Methoden für $l = 8$ und verschiedene minimale Varianzanteile für die Funktion f_4. Für jedes Ergebnis werden die Anzahl der Dimensionen der reduzierten Funktion d und der verbleibende Anteil der Varianz σ^2_{rem} aufgelistet.

Wie in Abbildung 4.32 zu sehen ist, unterscheiden sich die Methoden sehr deutlich in ihren Ergebnissen. Die PCA-Methode schafft es nur, maximal eine Dimension zu entfernen. Dies kann wieder auf die abweichende Konstruktion des Projektionsraumes zurückgeführt werden, welche schon in einem vorherigen Beispiel gezeigt wurde. Die AS-MC-Methode schafft es, mit einem Varianzanteil von 0.992 vier Dimensionen zu entfernen, was deutlich besser als alle anderen Methoden ist.

Somit schneidet auch hier wieder die AS-MC-Methode am besten ab, da sie durchgehend am meisten Dimensionen entfernen konnte. Darauf folgen die ANOVA-Methoden und die AS-QUAD-Methode, welche zwar etwas weniger Dimensionen entfernen konnten, jedoch immer noch akzeptabel abschneiden. Das Schlusslicht bildet wieder die PCA-Methode.

4.6. Hochdimensionale Modelle

Die PCA-Methode benötigt für n Samples und i Iterationen pro Sample im schlechtesten Fall (ni) Funktionsauswertungen. Für hochdimensionale Dünne Gitter mit höherem Level braucht die PCA-Methode somit viel Rechenzeit, da bei höherdimensionalen Parameterräumen mehr Samples nötig sind, um die Genauigkeit beizubehalten. Alternativ kann natürlich eine geringere Anzahl an Samples oder Iterationen gewählt werden, was aber die Genauigkeit der Methode beeinträchtigt.

Die AS-MC-Methode braucht für n Samples durch die Berechnung des Differenzenquotienten genau (nD) Funktionsauswertungen und kann daher viel mehr Samples als die PCA-Methode erstellen, da die Anzahl an Iterationen meistens viel größer als die Dimension ist, jedoch muss auch, wie bei der PCA-Methode die Anzahl an Samples für größere Dimensionen gesteigert werden.

Die AS-QUAD-Methode benötigt genau (gD) Funktionsauswertungen, wobei g die Anzahl der Gitterpunkte ist, um für D Dimensionen genau $D(D-1)$ Dünne Gitter zu erstellen. Für jede Kombination an Richtungsableitungen muss ein Dünnens Gitter erstellt und anschließend eine Quadratur darauf ausgeführt werden. Somit darf das Level für diese vielen Dünnen Gitter nicht zu groß gewählt werden. Ab wann also die AS-MC-Methode im Bezug auf die Laufzeit schneller als die AS-Quad-Methode ist, hängt also von der Dimension, der Anzahl der MC-Samples und des Levels des Surrogats ab.

Die ANOVA-Methoden haben bei hohen Dimensionen vor allem das Problem, dass die Anzahl der ANOVA-Komponenten exponentiell wächst und somit die Varianz für 2^D seperate Funktionsterme berechnet und anschließend die totalen Effektindizes berechnet werden müssen.

Zusammenfassend kann gesagt werden, dass sich für hochdimensionale Modelle laufzeittechnisch die ANOVA-Methoden und die AS-MC-Methode am besten eignen. Die PCA-Methode braucht zu viele Funktionsauswertungen um schon eine relativ kleine Menge an Samples zu generieren. Die AS-Quad-Methode arbeitet für große D mit sehr vielen Dünne Gittern und ist daher ebenfalls vergleichsweise langsam.

4.7. Fazit

Nachdem alle Methoden verschiedenen Tests unterworfen und die Ergebnisse ausgewertet wurden, kann nun basierend auf den Ergebnissen ein Fazit gezogen werden. Hierbei muss beachtet werden, dass die richtige Wahl der Dimensionsreduktionsmethode aber auch stark von der Beschaffenheit der Modellfunktion abhängt. Für das Beispiel f_2 unterscheiden sich die ANOVA- und AS-Methoden nur geringfügig, während für f_4 die Ergebnisse sehr stark variieren.

Die Active-Subspace-Methoden eignen sich, basierend auf den hier gezeigten Resultaten, am besten für die Dimensionsreduktion. Falls die Eingabefunktion, und somit auch die einzelnen Richtungsableitungen, ziemlich genau durch ein Dünnes Gitter interpoliert werden können und außerdem die Dimension der Eingabefunktion nicht zu groß ist, ist die AS-QUAD-Methode die beste Wahl, da sie weniger Funktionsauswertungen als die AS-MC-Methode für gleichgute Ergebnisse braucht. Kann hingegen die Funktion nur mit größerem Fehler mithilfe eines Dünnen Gitters interpoliert werden oder ist die Dimension hochdimensional, eignet sich die AS-MC-Methode besser.

Die PCA-Methode lieferte durchgehend schlechtere Ergebnisse als die Active-Subspace-Methoden ab und benötigt außerdem durch das Samplen vergleichsweise eine wesentlich größere Menge an Funktionsauswertungen. Daher sollte die PCA-Methode nicht angewendet werden. Der einzigste Pluspunkt ist die Berechnung des Stützvektors m für den Projektionsraum. Der Stützvektor ist der gebildete Mittelwert der generierten Samples der PCA-Methode und da dieser auch von den Active-Subspace-Methoden gut verwendet werden kann, ergänzt die PCA-Methode die Active-Subspace-Methoden und ist somit nicht komplett irrelevant.

Die Anchored-ANOVA-Methode hat gleichzeitig den Vor- und Nachteil der freien Wahl des Ankers. Diese ist vorteilhaft, da, wie gezeigt wurde, durch die richtige Wahl des Ankers bessere Ergebnisse erzielt werden können. Jedoch ist es nicht immer offensichtlich, welcher Anker die beste Wahl ist, wenn z.B eine hochdimensionale Funktion wie in dem letzten Anwendungsbeispiel reduziert werden soll. In Folge dessen muss ein guter Anker oft mithilfe des Trial and Error Prinzips bestimmt werden.

Als Alternative zu der Anchored-ANOVA-Methode bietet sich die klassische ANOVA-Methode an. Diese liefert, auf Grund der Art der Dekomposition, ähnliche Ergebnisse wie die Anchored-ANOVA-Methode mit guter Ankerwahl. Um somit ähnlich gute Ergebnisse ohne die Wahl eines Ankers zu erreichen, kann die klassische ANOVA-Methode angewendet werden.

Schlussendlich kann gesagt werden, dass bei guter Wahl der Dimensionsreduktionsmethode und dessen Parameter der Fluch der Dimensionalität für ein geeignetes Problem abgeschwächt werden kann. Somit kann die Laufzeitkomplexität von einigen hochdimensionalen Problemen, die ohne Dimensionsreduktion gar nicht oder nur sehr schwer lösbar waren, auf ein annehmbares Maß reduziert werden.

A. Appendix

A.1. Prewavelet ANOVA-Basis

Liegen die hierachischen Überschüsse $\alpha^h_{l,i}$ für eine hierarchische lineare ANOVA-Basis vor, kann damit begonnen werden herzuleiten, wie diese in die hierarchische Prewavelet ANOVA-Basis umgewandelt werden kann. Die Herleitung orientiert sich an [Feu10].

Wie schon in (2.10) gezeigt, lässt sich eine Prewavelet Ansatzfunktion für Level l und Index i lässt sich als Linearkombination der umliegenden Hütchenfunktionen ϕ des gleichen Levels darstellen:

Definition A.1.1
Sei

$$\beta_{l,i,j}$$

der Koeffizient, mit dem die $(i+j)$-te Hütchenfunktion des Levels l auf die Prewavelet-Ansatzfunktion für Level l und Index i aufsummiert wird. Dann kann eine Prewavelet-Ansatzfunktion $\phi^p_{l,i}$ wie folgt definiert werden:

$$\phi^p_{l,i} = \beta_{l,i,-2}\ \phi_{l,i-2} + \beta_{l,i,-1}\ \phi_{l,i-1} + \beta_{l,i,0}\ \phi_{l,i} + \beta_{l,i,+1}\ \phi_{l,i+1} + \beta_{l,i,+2}\ \phi_{l,i+2}$$

Ein Spezialfall dieser Darstellung sind die Ansatzfunktionen für $l \in \{-1, 0, 1\}$, da es auf diesen Levels weniger als 5 Hütchenfunktionen gibt:

$$\phi^p_{-1,0} = -\phi^n_{0,0} + \phi^n_{0,1} = 1$$
$$\beta_{-1,0,-2} = 0,\ \beta_{-1,0,-1} = 0,\ \beta_{-1,0,0} = 1,\ \beta_{-1,0,+1} = 1,\ \beta_{-1,0,+2} = 0$$

$$\phi^p_{0,1} = -\phi^n_{0,0} + \phi^n_{0,1}$$
$$\beta_{0,1,-2} = 0,\ \beta_{0,1,-1} = -1,\ \beta_{0,1,0} = 1,\ \beta_{0,1,+1} = 0,\ \beta_{0,1,+2} = 0$$

$$\phi^p_{1,1} = -\phi^n_{1,0} + \phi^n_{1,1} - \phi^n_{1,2}$$
$$\beta_{1,1,-2} = 0,\ \beta_{1,1,-1} = -1,\ \beta_{1,1,0} = 1,\ \beta_{1,1,+1} = -1,\ \beta_{1,1,+2} = 0$$

Dazu kommen noch die echten Prewavelet-Ansatzfunktionen für $l \geq 2$. Hierbei gibt es auch Speziallfälle, nämlich für die Prewavelets die am linken und rechten Rand liegen und den Index $i = 1$ oder $i = 2^l - 1$ besitzen:

$$\phi^p_{l,1} = -\frac{12}{10}\phi^n_{l,0} + \frac{11}{10}\phi^n_{l,1} - \frac{6}{10}\phi^n_{l,2} + \frac{1}{10}\phi^n_{l,3}$$
$$\beta_{l,1,-2} = 0,\ \beta_{l,1,-1} = -\frac{12}{10},\ \beta_{l,1,0} = \frac{11}{10},\ \beta_{l,1,+1} = -\frac{6}{10},\ \beta_{l,1,+2} = \frac{1}{10}$$

$$\phi^p_{l,2^l-1} = \frac{1}{10}\phi^n_{l,2^l-3} - \frac{6}{10}\phi^n_{l,2^l-2} + \frac{11}{10}\phi^n_{l,2^l-1} - \frac{12}{10}\phi^n_{l,2^l}$$
$$\beta_{l,2^l-1,-2} = \frac{1}{10},\ \beta_{l,2^l-1,-1} = -\frac{6}{10},\ \beta_{l,2^l-1,0} = \frac{11}{10},\ \beta_{l,2^l-1,+1} = -\frac{12}{10},\ \beta_{l,2^l-1,+2} = 0$$

$$\phi^p_{l,i} = \frac{1}{10}\phi^n_{l,i-2} - \frac{6}{10}\phi^n_{l,i-1} + \frac{10}{10}\phi^n_{l,i} - \frac{6}{10}\phi^n_{l,i+1} + \frac{1}{10}\phi^n_{l,i+2}$$
$$\beta_{l,i,-2} = \frac{1}{10},\ \beta_{l,i,-1} = -\frac{6}{10},\ \beta_{l,i,0} = \frac{10}{10},\ \beta_{l,i,+1} = -\frac{6}{10},\ \beta_{l,i,+2} = \frac{1}{10}$$

Falls für ein $\beta_{l,i,j}$ entweder $i \notin \tilde{H}_l$ oder $(i+j) \notin \{0,\dots,2^l\}$ gilt, ist hier $\beta_{l,i,j} = 0$.

Wie die Neumann-Prewavelet-Ansatzfunktionen für die Level -1 bis 3 aussehen, wurde bereits in 2.4 gezeigt.

Es wird nur der eindimensionale Fall betrachtet, da mithilfe des unidirektionalen Prinzips aus [Bun96] die Umwandlung auch für mehrdimensionale Fälle äquivalent durchgeführt werden kann. Betrachtet man den Algorithmus zur Umwandlung einer nodalen Basis mit $\alpha^n_{l,i}$ in eine lineare hierarchische Basis mit $\alpha^h_{l,i}$, wie sie in [Pfl10] beschrieben ist, sieht man, dass dieser rekursiv arbeitet und die hierarschischen Überschüsse für jedes Level folgendermaßen berechnet:

$$\alpha^h_{l,i} = t_{l,i} - \frac{1}{2}t_{l,i-1} - \frac{1}{2}t_{l,i+1} \tag{A.1}$$

Dabei werden temporäre Werte für das nächsthöhere Level propagiert, die die ursprünglichen $\alpha^n_{l,i}$ beschreiben:

Definition A.1.2
Sei $l \geq 1$. Dann ist

$$t_{l,i} = \alpha^n_{l,i} + t_{l+1,2i} \tag{A.2}$$

Für den propagierten temporären Wert gilt $t_{l,i} = 0$ falls l größer als das maximale Level ist. Außerdem ist $\alpha^n_{l,i} = 0$, falls l nicht das maximale Level ist.

Da Prewavelet-Ansatzfunktionen nur eine Linearkombination der umliegenden Hütchenfunktionen sind, kann eine nodale Basis für ein Level $l \geq 1$ leicht mithilfe folgender zwei Gleichungen auch mithilfe Prewavelet-Ansatzfunktionen ausgedrückt werden, die an allen ungeraden Indizes des Levels ansetzen:

Satz 3
Sei $l \geq 1$ und i ungerade. Dann gilt für den Wert der nodalen Ansatzfunktion $\phi^n_{l,i}$:

$$\alpha^n_{l,i} = \beta_{l,i,0}\ \alpha^p_{l,i} + \beta_{l,i-2,+2}\ \alpha^p_{l,i-2} + \beta_{l,i+2,-2}\ \alpha^p_{l,i+2} \tag{A.3}$$

Proof 3
Dies ist ziemlich offensichtlich, da es bei ungeraden Indizes drei Prewavelet-Ansatzfunktionen gibt, die ungleich 0 sind. Die Ansatzfunktion für den Index an sich ($\phi^p_{l,i}$) und die rechte und linke Ansatzfunktionen ($\phi^p_{l,i-2}$ und $\phi^p_{l,i+2}$).

Satz 4
Sei $l \geq 1$ und i gerade. Dann gilt für den Wert der nodalen Ansatzfunktion $\phi^n_{l,i}$:

$$\alpha^n_{l,i} = \beta_{l,i+1,-1}\ \alpha^p_{l,i+1} + \beta_{l,i-1,+1}\ \alpha^p_{l,i-1} \tag{A.4}$$

Proof 4
In diesem Fall gibt es am Index i keine Ansatzfunktion $\phi^p_{l,i}$, da i gerade ist. Jedoch gibt es an den jeweiligen Nachbarindizes Ansatzfunktionen ($\phi^p_{l,i+1}$ und $\phi^p_{l,i-1}$), da die Nachbarindizes ungerade sind.

Da eine hierarchische Basis konstruiert wird und es daher auf höheren Leveln mit $l \geq 1$ nur bei ungeraden Indizes eine Ansatzfunktion gibt, gilt folgender Satz:

Satz 5
Sei $l \geq 1$ und i ungerade. Dann ist

$$\begin{aligned}
\alpha^h_{l,i} &= t_{l,i} - \frac{1}{2}t_{l,i-1} - \frac{1}{2}t_{l,i+1} \\
&= \alpha^n_{l,i} + t_{l+1,2i} - \frac{1}{2}t_{l,i-1} - \frac{1}{2}t_{l,i+1} \\
&= \beta_{l,i,0}\ \alpha^p_{l,i} + \beta_{l,i-2,+2}\ \alpha^p_{l,i-2} + \beta_{l,i+2,-2}\ \alpha^p_{l,i+2} + t_{l+1,2i} - \frac{1}{2}t_{l,i-1} - \frac{1}{2}t_{l,i+1} \\
&= \beta_{l,i,0}\ \alpha^p_{l,i} + \beta_{l,i-2,+2}\ \alpha^p_{l,i-2} + \beta_{l,i+2,-2}\ \alpha^p_{l,i+2} + t_{l+1,2i} - \frac{1}{2}\alpha^n_{l,i-1} - \frac{1}{2}\alpha^n_{l,i+1} \\
&+ t_{l+1,2i} - \frac{1}{2}t_{l+1,2(i-1)} - \frac{1}{2}t_{l+1,2(i+1)}
\end{aligned} \tag{A.5}$$

Proof 5
Einsetzen von (A.2)*,* (A.3) *und* (A.4) *in* (A.1)*.*

Definition A.1.3
Sei

$$r_{l,i} = \alpha^h_{l,i} - t_{l+1,2i} + \frac{1}{2}t_{l+1,2(i-1)} + \frac{1}{2}t_{l+1,2(i+1)} \tag{A.6}$$

der Wert für Level l und Index i für die rechte Seite des lineare Gleichungssystems. Dieser Wert besteht nur aus Termen aus A.5, die bereits gegeben sind.

Bringt man nun einige Terme aus A.5 auf die linke Seite und ersetzt diese durch A.6 erhält man folgende Gleichung:

$$r_{l,i} = \beta_{l,i,0}\ \alpha^p_{l,i} + \beta_{l,i-2,+2}\ \alpha^p_{l,i-2} + \beta_{l,i+2,-2}\ \alpha^p_{l,i+2} - \frac{1}{2}\alpha^n_{l,i-1} - \frac{1}{2}\alpha^n_{l,i+1} \tag{A.7}$$

Satz 6
Sei $l \geq 1$ und $i \in \tilde{H}_l$. Dann gilt:

$$\begin{aligned} r_{l,i} =& \beta_{l,i,0}\ \alpha^p_{l,i} + \beta_{l,i-2,+2}\ \alpha^p_{l,i-2} + \beta_{l,i+2,-2}\ \alpha^p_{l,i+2} \\ & -\frac{1}{2}(\beta_{l,i,-1}\ \alpha^p_{l,i} + \beta_{l,i-2,+1}\ \alpha^p_{l,i-2}) - \frac{1}{2}(\beta_{l,i,+1}\alpha^p_{l,i} + \beta_{l,i+2,-1}\alpha^p_{l,i+2}) \\ =& (\beta_{l,i,0} - \frac{1}{2}\beta_{l,i,+1} - \frac{1}{2}\beta_{l,i,-1})\alpha^p_{l,i} \\ & +(\beta_{l,i+2,-2} - \frac{1}{2}\beta_{l,i+2,-1})\alpha^p_{l,i+2} \\ & +(\beta_{l,i-2,+2} - \frac{1}{2}\beta_{l,i-2,+1})\alpha^p_{l,i-2} \end{aligned} \tag{A.8}$$

Proof 6
Einsetzen von (A.4) *in* (A.7)*, da i ungerade ist und daher $i-1$ und $i+1$ gerade ist. Anschließend können die α^p ausgeklammert werden.*

Satz 7
Sei $l \geq 1$ und i gerade. Dann lässt sich $t_{l,i}$ wie folgt berechnen:

$$t_{l,i} = \beta_{l,i+1,-1}\ \alpha^p_{l,i+1} + \beta_{l,i-1,+1}\ \alpha^p_{l,i-1} + t_{l+1,2i} \tag{A.9}$$

Proof 7
Einsetzen von (A.4) *in* (A.2)*, da i gerade ist.*

Für $l \geq 2$ und die Indizes $i \in \{3, 5, \ldots, 2^l - 5, 2^l - 3\}$, die nicht am Rand liegen, gilt nach A.8:

$$\begin{aligned} r_{l,i} &= (1 - (\frac{1}{2}(-\frac{6}{10})) - (\frac{1}{2}(-\frac{6}{10})))\alpha^p_{l,i} + (\frac{1}{10} - (\frac{1}{2}(-\frac{6}{10})))\alpha^p_{l,i+2} + (\frac{1}{10} - (\frac{1}{2}(-\frac{6}{10})))\alpha^p_{l,i-2} \\ &= \frac{16}{10}\alpha^p_{l,i} + \frac{4}{10}\alpha^p_{l,i+2} + \frac{4}{10}\alpha^p_{l,i-2} \end{aligned}$$

Für $l \geq 2$ und den Index $i = 1$ am linken Rand gilt nach (A.8):

$$\begin{aligned} r_{l,1} &= (\frac{11}{10} - (\frac{1}{2}(-\frac{6}{10})) - (\frac{1}{2}(-\frac{12}{10})))\alpha^p_{l,1} + (\frac{1}{10} - (\frac{1}{2}(-\frac{12}{10})))\alpha^p_{l,3} + 0 \\ &= \frac{20}{10}\alpha^p_{l,1} + \frac{7}{10}\alpha^p_{l,3} \end{aligned}$$

Für $l \geq 2$ und den Index $i = 2^l - 1$ am rechten Rand gilt nach (A.8):

$$\begin{aligned} r_{l,2^l-1} &= (\frac{11}{10} - (\frac{1}{2}(-\frac{12}{10})) - (\frac{1}{2}(-\frac{6}{10})))\alpha^p_{l,2^l-1} + 0 + (\frac{1}{10} - (\frac{1}{2}(-\frac{12}{10})))\alpha^p_{l,2^l-3} \\ &= \frac{20}{10}\alpha^p_{l,2^l-1} + \frac{7}{10}\alpha^p_{l,2^l-3} \end{aligned}$$

Aus diesen Gleichungen lässt sich nun für $l \geq 2$ folgendendes tridiagonales lineares Gleichungssystem aufstellen, welches leicht mit einem entsprechenden Algorithmus gelöst werden kann:

$$\begin{pmatrix} \frac{20}{10} & \frac{7}{10} & 0 & 0 & \cdots & 0 \\ \frac{4}{10} & \frac{16}{10} & \frac{4}{10} & 0 & \cdots & 0 \\ 0 & \frac{4}{10} & \frac{16}{10} & \frac{4}{10} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \frac{4}{10} & \frac{16}{10} & \frac{4}{10} \\ 0 & 0 & 0 & 0 & \frac{7}{10} & \frac{20}{10} \end{pmatrix} \begin{pmatrix} \alpha^p_{l,1} \\ \alpha^p_{l,3} \\ \alpha^p_{l,5} \\ \vdots \\ \alpha^p_{l,2^l-3} \\ \alpha^p_{l,2^l-1} \end{pmatrix} = \begin{pmatrix} r_{l,1} \\ r_{l,3} \\ r_{l,5} \\ \vdots \\ r_{l,2^l-3} \\ r_{l,2^l-1} \end{pmatrix} \tag{A.10}$$

Nachdem das LGS (A.10) gelöst wurde, müssen für die nächste Iteration noch die Werte $t_{l,i}$ berechnet werden. Ein Sonderfall von (A.9) sind die Werte $t_{l,0}$ und $t_{l,2^l}$ für den Rand:

$$\begin{aligned} t_{l,0} &= \beta_{l,1,-1}\, \alpha^p_{l,1} + t_{l+1,0} = -\frac{12}{10}\, \alpha^p_{l,1} + t_{l+1,0} \\ t_{l,2^l} &= \beta_{l,2^l-1,+1}\, \alpha^p_{l,2^l-1} + t_{l+1,2^{(l+1)}} = -\frac{12}{10}\, \alpha^p_{l,2^l-1} + t_{l+1,2^{(l+1)}} \end{aligned}$$

Für $t_{l,i}$, die nicht am Rand liegen, gilt nach (A.9):

$$t_{l,i} = \beta_{l,i+1,-1}\, \alpha^p_{l,i+1} + \beta_{l,i-1,+1}\, \alpha^p_{l,i-1} + t_{l+1,2i} = -\frac{6}{10}\, \alpha^p_{l,i+1} - \frac{6}{10}\, \alpha^p_{l,i-1} + t_{l+1,2i}$$

Jetzt können die $\alpha^p_{l,i}$ des nächsten niedrigeren Levels mithilfe der gerade berechneten $t_{l,i}$ berechnet werden. Ist das nächste Level 1, muss nur eine Gleichung aufgestellt werden.

Mit (A.8) kann dann die Gleichung für $\alpha^p_{1,1}$ aufgestellt werden:

$$\begin{aligned} & r_{1,1} = (\beta_{1,1,0} - \frac{1}{2}\beta_{1,1,-1} - \frac{1}{2}\beta_{1,1,1})\, \alpha^p_{1,1} \\ \Leftrightarrow & \alpha^h_{1,1} - t_{2,2} + \frac{1}{2}t_{2,0} + \frac{1}{2}t_{2,4} = (1 + \frac{1}{2} + \frac{1}{2})\alpha^p_{1,1} \\ \Leftrightarrow & \alpha^p_{1,1} = \frac{\alpha^h_{1,1} - t_{2,2} + \frac{1}{2}t_{2,0} + \frac{1}{2}t_{2,4}}{2} \end{aligned}$$

Mit (A.9) kann dann $t_{1,0}$ und $t_{1,2}$ folgendermaßen berechnet werden:

$$\begin{aligned} t_{1,0} &= \beta_{1,1,-1}\, \alpha^p_{1,1} + t_{2,0} = -\alpha^p_{1,1} + t_{2,0} \\ t_{1,2} &= \beta_{1,1,+1}\, \alpha^p_{1,1} + t_{2,4} = -\alpha^p_{1,1} + t_{2,4} \end{aligned}$$

Jetzt müssen nur noch $\alpha^p_{-1,0}$ und $\alpha^p_{0,1}$ berechnet werden. Dafür kann eine Gleichung für den linken Rand aufgestellt:

$$\begin{aligned} & \beta_{-1,0,0}\, \alpha^p_{-1,0} + \beta_{0,1,-1}\, \alpha^p_{0,1} = \alpha^h_{0,0} - t_{1,0} \\ \Leftrightarrow\ & \alpha^p_{-1,0} - \alpha^p_{0,1} = \alpha^h_{-1,0} - t_{1,0} \end{aligned} \tag{A.11}$$

Für den rechten Rand gilt:

$$\begin{aligned} \beta_{-1,0,1}\, \alpha^p_{-1,0} + \beta_{0,1,0}\, \alpha^p_{0,1} &= \alpha^h_{0,1} - t_{1,2} \\ \Leftrightarrow \alpha^p_{-1,0} + \alpha^p_{0,1} &= \alpha^h_{-1,0} + \alpha^h_{0,1} - t_{1,2} \end{aligned} \tag{A.12}$$

Löst man nun die Gleichung (A.11) nach $\alpha^p_{-1,0}$ auf, indem man (A.12) einsetzt, erhält man:

$$\begin{aligned} \alpha^p_{-1,0} - \alpha^p_{0,1} &= \alpha^n_{0,1} - t_{1,2} \\ \Leftrightarrow 2\alpha^p_{-1,0} &= 2\alpha^h_{-1,0} + \alpha^h_{0,1} - t_{1,0} - t_{1,2} \\ \Leftrightarrow \alpha^p_{-1,0} &= \alpha^h_{-1,0} + \frac{\alpha^h_{0,1} - t_{1,0} - t_{1,2}}{2} \end{aligned}$$

Löst man nun die Gleichung (A.12) nach $\alpha^p_{0,1}$ auf, indem man (A.11) einsetzt, erhält man:

$$\begin{aligned} \alpha^p_{-1,0} + \alpha^p_{0,1} &= \alpha^n_{0,1} - t_{1,2} \\ \Rightarrow\ 2\alpha^p_{0,1} &= \alpha^h_{0,1} + t_{1,0} - t_{1,2} \\ \Rightarrow \alpha^p_{0,1} &= \frac{\alpha^h_{0,1} + t_{1,0} - t_{1,2}}{2} \end{aligned}$$

Zusammenfassend kann gesagt werden, dass um die Koeffizienten einer eindimensionalen hierarchischen Prewavelet-Basis des Levels l zu berechnen für jedes $2 \leq l' \leq l$ ein tridiagonales Gleichungssystem mit $2^{l'-1}$ Einträgen gelöst werden muss. Die Fälle für $l' \leq 2$ müssen gesondert betrachtet und berechnet werden.

Literaturverzeichnis

[Bun96] Bungartz, H.J.: A unidirectional approach for d-dimensional finite element methods for higher order on sparse grids. (1996), 12

[Feu10] Feuersänger, Christian: *Sparse Grid Methods for Higher Dimensional Approximation*, Mathematisch–Naturwissenschaftliche Fakultät der der Rheinischen Friedrich–Wilhelms–Universität Bonn, Diss., 2010

[Gar13] Garcke, Jochen: Sparse grids in a nutshell. In: *Sparse grids and applications*, Springer, 2013, S. 57–80

[GO95] Griebel, M. ; Oswald, Peter: Tensor Product Type Subspace Splittings and Multilevel Iterative Methods for Anisotropic Problems. In: *Advances in Computational Mathematics* 4 (1995), 12, S. 171–206. http://dx.doi.org/10.1007/BF02123478. – DOI 10.1007/BF02123478

[Has70] Hastings, W. K.: Monte Carlo sampling methods using Markov chains and their applications. In: *Biometrika* 57 (1970), 04, Nr. 1, 97-109. http://dx.doi.org/10.1093/biomet/57.1.97. – DOI 10.1093/biomet/57.1.97. – ISSN 0006–3444

[Hol08] Holtz, Markus: *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*, Mathematisch–Naturwissenschaftliche Fakultät der der Rheinischen Friedrich–Wilhelms–Universität Bonn, Diss., 2008

[PC14] Paul Constantine, David G.: Computing active subspaces with Monte Carlo. (2014). https://arxiv.org/pdf/1408.0545v2.pdf

[Pfl10] Pflüger, Dirk: *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut http://www5.in.tum.de/pub/pflueger10spatially.pdf. – ISBN 9783868535556

[Pfl12] Pflüger, Dirk: Spatially Adaptive Refinement. In: Garcke, Jochen (Hrsg.) ; Griebel, Michael (Hrsg.): *Sparse Grids and Applications*. Springer (Lecture Notes in Computational Science and Engineering), 243–262

[Sob01] Sobolá, I. M.: Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates. In: *Math. Comput. Simul.* 55 (2001), Februar, Nr. 1-3, 271–280. http://dx.doi.org/10.1016/S0378-4754(00)00270-6. – DOI 10.1016/S0378–4754(00)00270–6. – ISSN 0378–4754

[Val14] Valentin, Julian: *Hierarchische Optimierung mit Gradientenverfahren auf Dünngitterfunktionen*. ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart_fi/MSTR-3629/MSTR-3629.pdf. Version: 2014

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift