

Die Proper Generalized Decomposition zur Lösung der Wärmeleitungsgleichung

Matthias Wissel

Geboren am 18. Oktober 1988 in Aschaffenburg

1. November 2012

Bachelorarbeit Mathematik

Betreuer: Prof. Dr. Michael Griebel

INSTITUT FÜR NUMERISCHE SIMULATION

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT DER
RHEINISCHEN FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

Notation

Symbol	Bedeutung
Ω	betrachtetes Gebiet
τ	Diffusionskoeffizient
N	Anzahl der Basisfunktionen bzw. Gitterpunkte
z_1, \dots, z_N	Gitterpunkte
n	Index der Basisfunktionen bzw. Gitterpunkte
ϕ_n	Basisfunktionen
T	Anzahl der Zeitschritte
t	Index der Zeitschritte
D	Anzahl der Dimensionen
d	Index der Dimension
f	Rechte Seite
q	Iterationsschritt im Alternating Directions Algorithmus
m	Iterationsschritt in der Anreicherung der Lösung
v	Testfunktion in schwacher Formulierung
X_i	Ortsfunktionen in Darstellung für Separation in Zeit und Ort
Z_i	Zeitfunktion in Darstellung für Separation in Zeit und Ort
λ, μ, ν	Konstanten in Herleitung Separation in Zeit und Ort
U^t	Lösung in Zeitschritt t
u_i^t	Summenterm der Lösung in Zeitschritt t
$u_i^{t,(d)}$	1D-Funktion als Faktor in Summenterm i der Lösung in Zeitschritt t
a, b, c, d	Konstanten in Darstellung mit semiimplizierter Zeitdiskretisierung
\mathcal{A}	Operator in Tensorform
\mathcal{U}	Lösung in Tensorform
\mathcal{V}	Testfunktion in Tensorform
\mathcal{B}	Rechte Seite in Tensorform

Inhaltsverzeichnis

Notation	i
Inhaltsverzeichnis	iv
1. Einleitung	1
2. Herleitung des Verfahrens	5
2.1. Separation in Zeit und Ort	5
2.1.1. Abbruchkriterium	7
2.2. Semiimplizite Zeitdiskretisierung und Separation im Ort	8
2.2.1. Abbruchkriterium	10
3. Verallgemeinerung auf Tensoren	11
4. Fehler- und Laufzeitanalyse	13
4.1. Fehleranalyse	13
4.1.1. Vergleich von reduzierter und voller Tensorapproximation	13
4.1.2. Fehlerabschätzung basierend auf dem dualen Problem	14
4.1.3. Fehlerabschätzung für die niederdimensionalen Probleme	15
4.2. Komplexität und Laufzeit	15
4.2.1. Separation in Zeit und Ort	16
4.2.2. Separation für die Ortskomponente	16
5. Besonderheiten der Implementierung	17
5.1. Eigenschaften der Basisfunktionen	17
5.2. L_2 -Norm in Abbruchkriterien	18
5.3. Mehrdimensionale Integration für das L_2 -Skalarprodukt	19
5.4. Pseudocode	20
5.4.1. Approximation von Funktionen	20
5.4.2. Lösung der Poisson-Gleichung	21
5.4.3. Lösung der Wärmeleitungsgleichung	22
6. Numerische Experimente	25
6.1. Approximation von Funktionen	25
6.1.1. Funktionen mit Produktstruktur	25
6.1.2. Funktionen in beliebiger Form	28
6.2. Lösung der Poisson-Gleichung	29
7. Zusammenfassung und Ausblick	35

A. Appendix	37
A.1. Satz von Fubini	37
A.2. Green'sche Formel	37
Literaturverzeichnis	39

1. Einleitung

Immer mehr Fragestellungen und Probleme aus Physik, Technik und Wirtschaft werden heute mathematisch modelliert und beschrieben. Eine Vielzahl dieser Modelle beinhaltet neben stochastischen Prozessen und Optimierungsproblemen auch verschiedenste gewöhnliche und partielle Differentialgleichungen. Nach der Übersetzung der jeweiligen Fragestellungen in das mathematische Modell steht die Frage nach der Lösung.

Im Falle der Differentialgleichungen gibt es bereits zu vielen eine zufriedenstellende Lösung und auch klassische Wege diese zu berechnen. Hierzu werden zunehmend Computer herangezogen, da die Komplexität der Probleme das Berechnen einer analytischen Lösung immer schwieriger oder gar unmöglich machen. Allerdings stellt die steigende Komplexität auch die Computer vor immer größere Herausforderungen. Diese sind nicht allein durch immer größere Rechenleistung auszugleichen. Es müssen zusätzlich effizientere Verfahren entwickelt werden, um den Anforderungen auch weiterhin gerecht zu werden.

Das Ausmaß der benötigten Rechenleistung wird besonders bei hochdimensionalen Fragestellungen deutlich. Hierzu zählen Probleme aus der Fluidodynamik, der Quantenmechanik, aber auch der Finanzmathematik. Bei diesen Modellen tritt mit klassischen gitterbasierten Methoden der Fluch der Dimensionen auf, das heißt die Dimension wirkt sich exponentiell auf den Rechenaufwand aus. Soll etwa ein Finanzderivat auf den DAX-30 also $D = 30$, Dimensionen bewertet werden, so ergeben sich bei einem Gitter mit N Knotenpunkten pro Koordinate N^D Freiheitsgrade. Wird nun eine Diskretisierung von $N = 1000$ Knotenpunkten genutzt ergeben sich $1000^{30} = 10^{90}$ Unbekannte. Diese gigantische Zahl ist sogar größer als die Anzahl der Protonen im sichtbaren Universum, die auf 10^{80} geschätzt wird.

Aufgrund dessen werden zunehmend Verfahren gesucht die den Fluch der Dimensionen umgehen. Eine Möglichkeit bieten Separationsansätze, deren Anwendung zur Lösung von partiellen Differentialgleichungen unter anderem in [9] beschrieben wird. Die Methodik der Proper Generalized Decomposition wurde noch ohne diesen Namen erstmals in [3] im Kontext der kinetischen Gastheorie vorgestellt. Der Ansatz wurde in der Folge immer populärer und fand Anwendung in verschiedenen Bereichen. So wurde sie in [7] auf spezielle Fälle der Navier-Stokes-Gleichung sowie die Stokes- und Burgers-Gleichung angewandt. Zur Lösung von parametrischen Modellen wurde sie in [15] verwendet. Auch für klassische Probleme wie zeitabhängige partielle Differentialgleichungen kann sie angewendet werden, wie in [14] beschrieben. Die theoretischen Hintergründe wie Fehlerabschätzungen, siehe [2], ein Beweis zur Konvergenz der PGD für lineare Probleme in Hilbert-Produkt-Räumen [11] oder Methoden im Zusammenhang mit der Minimierung nichtlinearer konvexer Funktionen über reflexiven Banach-Räumen [12] wurden ebenfalls bereits betrachtet. Ein Ansatz für die effiziente Implementierung für Matrizen mit Laplaceform wurde in [10] gegeben. Eine Zusammenfassung der neuesten Fortschritte und Entwicklungen im Zusammenhang mit der Proper Generalized Decomposition gibt [5].

Problemstellung

In dieser Arbeit soll eine Lösung der hochdimensionalen Wärmeleitungsgleichung

$$\frac{\partial u}{\partial t} - \tau \Delta u = f(x_1, \dots, x_D, t) \quad \Omega_1 \times \dots \times \Omega_D \times (0, T_{max}]$$

mit entsprechenden Anfangs- und Randbedingungen

$$\begin{aligned} u(x, 0) &= g & (x_1, \dots, x_D) &\in \Omega \\ u(x, t) &= h & (x_1, \dots, x_D, t) &\in \partial\Omega \times (0, T_{max}] \end{aligned}$$

für $\Omega_i \subset \mathbb{R}$ numerisch berechnet werden. Dies ist auch mit herkömmlichen Methoden wie Finiten Differenzen oder Finiten-Elementen möglich, allerdings wird hier der Fluch der Dimensionen schnell zu einem großen Problem. Wird für eine solche Lösungsmethode ein Gitter mit $N^{(i)}$ Knoten für jedes Teilgebiet Ω_i bzw. $(0, T_{max}]$ genutzt, dann besitzt das gesamte Gebiet $\Omega \times (0, T_{max}] = \Omega_1 \times \dots \times \Omega_D \times (0, T_{max}]$ bereits $\prod_{i=1}^{D+1} N^{(i)}$ Knoten. Wir sehen, dass die Anzahl der Freiheitsgrade exponentiell mit der Dimension des Gebiets steigt.

Um diesem Phänomen entgegenzuwirken wollen wir einen alternativen Lösungsansatz vorstellen und dessen Eigenschaften untersuchen .

Lösungsansätze

Die Lösung des Problems soll mittels der Proper Generalized Decomposition (PGD) berechnet werden. Hierzu wollen wir einen Separationsansatz verwenden und Lösungsdarstellungen der Form

$$U(x, t) \approx \sum_{i=1}^M X_i(x) \cdot Z_i(t) \tag{1.1}$$

oder

$$U^t(x) \approx \sum_{i=1}^{M_t} u_i^t(x) \tag{1.2}$$

erreichen.

In der Darstellung (1.1) wird in jedem Rechenschritt die Summe um einen Produktterm angereichert und so das Residuum minimiert, bis die benötigte Genauigkeit der Lösung erreicht wird. In diesem Fall reduziert sich die Anzahl der benötigten Freiheitsgrade zu $M \times (\prod_{i=1}^D N^{(i)} + N^{(D+1)})$. Die exponentielle Steigerung der Komplexität konnte also um eins verringert werden. Hier wird die Wahl des Lösers zum Errechnen der Funktionen X_i entscheidend sein für die Komplexität. Wählen wir als Lösungsdarstellung Form (1.2) so wird in jedem Zeitschritt t eine solche Summe berechnet, die wieder in jedem Rechenschritt mit einem Produkt angereichert wird. Hier reduziert sich die Anzahl der Freiheitsgrade zu $T \times M_t \times \sum_{i=1}^D N^{(i)}$. Die Komplexität steigt also linear mit der Dimension.

Eigene Beiträge

- Herleitung der Proper Generalized Decomposition mit semiimpliziter Zeitdiskretisierung und Separationsansatz im Ort
- Ableitung der Algorithmen aus den theoretischen Herleitungen der Verfahrensvarianten sowie deren Analyse hinsichtlich der Komplexitätsklasse
- Versuche zur Approximation von Funktionen, dem Einfluss von Startwerten, der Fehlerentwicklung in den Teilbereichen des Verfahrens und der Lösung der Poisson-Gleichung sowie den Einfluss der Dimension auf die Rechenzeit

Aufbau der Arbeit

In Kapitel 2 leiten wir die Vorgehensweise der Proper Generalized Decomposition (PGD) anhand der Wärmeleitungsgleichung in zwei Varianten her. Hauptunterschied der zwei Varianten ist die Behandlung der Zeitkomponente. Es wird zum einen eine Separation in Zeit und Ort vorgestellt und zum anderen ein θ -Schema zur Behandlung der Zeitkomponente verwendet.

In Kapitel 3 führen wir eine Tensornotation ein und verallgemeinern das Verfahren auf höhere Dimensionen und allgemeinere Differentialoperatoren.

In Kapitel 4 geben wir einen Einblick in die Fehleranalyse und Komplexität des Verfahrens. Hierbei werden die zwei Varianten aus Kapitel 2 getrennt betrachtet und analysiert. Außerdem wird ein allgemeiner Ansatz für Fehleranalysen im Bereich Produktdarstellungen vorgestellt.

In Kapitel 5 gehen wir auf Schwierigkeiten und Besonderheiten bei der Implementierung des Verfahrens ein. Es werden Wege aufgezeigt die teils mehrdimensionalen Probleme auf Eindimensionale zu reduzieren sowie diese effizient zu lösen. Wir gehen auch auf die Wahl der Basiselemente im benutzten Finite-Elemente Raum ein und betrachten deren Vorteile für das Verfahren.

In Kapitel 6 werden Ergebnisse von numerischen Experimenten vorgestellt. Wir untersuchen die Auswirkung der Struktur der rechten Seiten der Gleichung sowie den Einfluss der Dimension auf die Zahl der nötigen Anreicherungsterme und die Laufzeit.

Kapitel 7 fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf mögliche Anwendungen sowie mögliche Verbesserungen des Verfahrens in verschiedenen Bereichen.

Danksagung

An dieser Stelle möchte ich allen danken, die zum Gelingen dieser Arbeit beigetragen haben. Zuerst danke ich Prof. Dr. Michael Griebel für die Vergabe des interessanten Themas, sowie Anregungen und Hilfestellungen bei Problemen. Weiterhin möchte ich mich bei Dr. Christian Rieger für die Übernahme des Zweitgutachtens bedanken.

Mein besonderer Dank gilt meinem Betreuer Alexander Hullmann für die konstruktiven Hilfestellungen, Anregungen und Erklärungen sowie seine große Geduld bei Fragen und anderen Problemen.

Schließlich möchte ich meiner Familie und meinen Freunden danken für die jahrelange Unterstützung und ihr Verständnis in allen Lebenslagen.

2. Herleitung des Verfahrens

In diesem Kapitel werden zwei Wege zur Diskretisierung der Wärmeleitungsgleichung vorgestellt die mittels eines Separationsansatzes realisiert werden können.

Im ersten Fall wird ein Separationsansatz für die Zeit und die Ortskomponente gewählt, wie auch in [5] beschrieben. Dieser besitzt folgende Darstellungsform der Lösung $U(x, t)$

$$U(x, t) \approx \sum_{i=1}^M X_i(x) \cdot Z_i(t). \quad (2.1)$$

Im zweiten Fall wird eine semiimplizite Diskretisierung der Zeitkomponente vorgenommen und der Separationsansatz wird dann in den Ortskomponenten angewandt. Hier suchen wir die Lösungen $U^t(x)$ für $t = 0, \dots, T$ in der Form

$$U^t(x) \approx \sum_{i=1}^{M_t} u_i^t(x) \quad (2.2)$$

Beide Wege sollen nun dargestellt werden.

Hierzu betrachten wir die Wärmeleitungsgleichung

$$\frac{\partial}{\partial t} u - \tau \Delta u = f(x, t) \quad \Omega \times (0, T_{max}] \quad (2.3)$$

mit entsprechenden Anfangs- und Randbedingungen

$$\begin{aligned} u(x, 0) &= g & x &\in \Omega \\ u(x, t) &= h & (x, t) &\in \partial\Omega \times (0, T_{max}] \end{aligned} \quad (2.4)$$

mit dem Diffusionskoeffizienten $\tau \in \mathbb{R}$ und $\Omega \subset \mathbb{R}^d, 1 \leq d \leq 3, T > 0$.

2.1. Separation in Zeit und Ort

Für den Separationsansatz in Zeit und Ort wollen wir M Paare der Funktionen $\{(X_i, Z_i)\}_{i=1, \dots, M}$ aus Darstellung (2.1) berechnen. Die Funktionen $\{X_i\}_{i=1, \dots, M}$ und $\{Z_i\}_{i=1, \dots, M}$ sind jeweils auf $\bar{\Omega}$ bzw. $[0, T_{max}]$ definiert und wir erhalten eine Darstellung der Lösung von der Form (2.1).

Zuerst betrachten wir die schwache Formulierung von Gleichung (2.3). Diese lautet:

Suche $u(x, t)$ das die Anfangs- und Randwerte (2.4) erfüllt, sodass

$$\int_0^T \int_{\Omega} \left(\frac{\partial}{\partial t} u(x, t) - \tau \Delta u(x, t) - f(x, t) \right) \cdot v(x, t) \, dx dt = 0 \quad (2.5)$$

für alle $v(x, t)$ in einem geeigneten Funktionenraum.

Wir gehen nun davon aus, dass die Funktionenpaare $\{(X_i, Z_i)\}_{i=1, \dots, M}$ für $i = 0, \dots, m < M$

2. Herleitung des Verfahrens

bereits berechnet wurden. Wir suchen also das Paar (X_{m+1}, Z_{m+1}) . Um dieses Paar zu berechnen wenden wir einen Alternating Directions Algorithmus an.

Dazu gehen wir im Anreicherungsschritt m von der Separationsdarstellung

$$u(x, t) \approx \sum_{i=1}^m X_i(x) \cdot Z_i(t) + W(x) \cdot Y(t) \quad (2.6)$$

aus. Außerdem können wir für die Testfunktion v folgende Form annehmen:

$$v = W \cdot Y^* + W^* \cdot Y \quad (2.7)$$

Durch einsetzen der letzten beiden Gleichungen in die schwache Formulierung erhalten wir

$$\begin{aligned} & \int_0^T \int_{\Omega} (W \cdot Y^* + Y \cdot W^*) \cdot \left(W \cdot \frac{\partial}{\partial t} Y - \tau \Delta W \cdot Y \right) dx dt = \\ & \int_0^T \int_{\Omega} (W \cdot Y^* + Y \cdot W^*) \cdot \left(f(x, t) - \sum_{i=1}^m X_i \cdot \frac{\partial}{\partial t} Z_i + \tau \sum_{i=1}^m \Delta X_i \cdot Z_i \right) dx dt \end{aligned} \quad (2.8)$$

Der Alternating Directions Algorithmus soll nun sukzessive die Funktionen W und Y verbessern, sodass diese sich dem gesuchten Funktionenpaar (X_{m+1}, Z_{m+1}) annähern.

Zuerst berechnen wir $W(x)$. Dazu nehmen wir an, dass $Y(t)$ bereits bekannt ist. Dadurch verschwindet $Y^*(t)$ in (2.7). Einsetzen in (2.8) liefert

$$\begin{aligned} & \int_{\Omega} W^* \cdot (\lambda_t W - \tau \mu_t \Delta W) dx = \\ & \int_{\Omega} W^* \cdot \left(\nu_t(x) - \sum_{i=1}^m \lambda_t^i X_i + \tau \sum_{i=1}^m \mu_t^i \Delta X_i \right) dx \end{aligned} \quad (2.9)$$

wobei

$$\begin{aligned} \lambda_t &= \int_0^T Y(t) \cdot \frac{\partial}{\partial t} Y(t) dt \\ \lambda_t^i &= \int_0^T Y(t) \cdot \frac{\partial}{\partial t} Z_i(t) dt \\ \mu_t &= \int_0^T Y^2(t) dt \\ \mu_t^i &= \int_0^T Y(t) \cdot Z_i(t) dt \\ \nu_t(x) &= \int_0^T Y(t) \cdot f(x, t) dt \quad \forall x \in \Omega \end{aligned} \quad (2.10)$$

Da diese schwache Formulierung für alle W^* erfüllt ist, kehren wir zur starken Formulierung zurück

$$\lambda_t W - \tau \mu_t \Delta W = \nu_t - \sum_{i=1}^m \lambda_t^i X_i + \tau \sum_{i=1}^m \mu_t^i \Delta X_i \quad (2.11)$$

Diese Gleichung kann durch eine beliebige passende Diskretisierungsmethode gelöst werden.

Nun wollen wir $Y(t)$ berechnen. Dazu können wir von der bereits berechneten Funktion $W(x)$ ausgehen. In diesem Fall verschwindet W^* in (2.7) und (2.8) lautet dann

$$\int_0^T \int_{\Omega} (W \cdot Y^*) \cdot \left(W \cdot \frac{\partial}{\partial t} Y - \tau \Delta W \cdot Y \right) = \int_0^T \int_{\Omega} (W \cdot Y^*) \cdot \left(f(x, t) - \sum_{i=1}^m X_i \cdot \frac{\partial}{\partial t} Z_i + \tau \sum_{i=1}^m \Delta X_i \cdot Z_i \right) dx dt \quad (2.12)$$

wobei alle ortsbezogenen Funktionen integrierbar in Ω sind. Mit

$$\begin{aligned} \lambda_x &= \int_{\Omega} W(x) \cdot \Delta W(x) dx \\ \lambda_x^i &= \int_{\Omega} W(x) \cdot \Delta X_i(x) dx \\ \mu_x &= \int_{\Omega} W^2(x) dx \\ \mu_x^i &= \int_{\Omega} W(x) \cdot X_i(x) dx \\ \nu_x(t) &= \int_{\Omega} W(x) \cdot f(x, t) dx \quad \forall t \end{aligned} \quad (2.13)$$

ergibt sich für (2.12)

$$\int_0^T Y^* \cdot \left(\mu_x \frac{\partial}{\partial t} Y - \tau \lambda_x Y - \nu_x(t) + \sum_{i=1}^m \mu_x^i \frac{\partial}{\partial t} Z_i - \sum_{i=1}^m \tau \lambda_x^i Z_i \right) dt = 0 \quad (2.14)$$

Diese Gleichung ist wieder für alle Y^* erfüllt und wir können zur starken Formulierung übergehen

$$\mu_x \frac{\partial}{\partial t} Y = \tau \lambda_x Y + \nu_x(t) - \sum_{i=1}^m \mu_x^i \frac{\partial}{\partial t} Z_i + \sum_{i=1}^m \tau \lambda_x^i Z_i \quad (2.15)$$

Diese gewöhnliche Differentialgleichung lässt sich selbst für kleine Zeitschritte mit den gegebenen Anfangsbedingungen problemlos lösen.

Diese beiden Schritte werden nun abwechselnd solange wiederholt bis das Verfahren konvergiert, d.h. beide Funktionen einen Fixpunkt erreicht haben. Ist dieser Fixpunkt erreicht werden die berechneten Funktionen zu dem neuen Anreicherungsterm, also $X_{m+1}(x) = W(x)$ und $Z_{m+1}(t) = Y(t)$.

Mit diesem Vorgehen werden nun so lange neue Anreicherungsterme berechnet bis Konvergenz vorliegt.

2.1.1. Abbruchkriterium

Um das Abbruchkriterium für den Alternating Directions Algorithmus zu beschreiben bezeichnen wir mit $Y^{(q)}(t)$ und $Y^{(q-1)}(t)$ die berechneten Funktionen $Y(t)$ im aktuellen bzw. vorherigen

Iterationsschritt. Gleiches gilt für die Ortsfunktionen $W^{(q)}(x)$ und $W^{(q-1)}(x)$. Das Abbruchkriterium lautet dann:

$$e = \|Y^{(q)}(t) \cdot W^{(q)}(x) - Y^{(q-1)}(t) \cdot W^{(q-1)}(x)\|_{L_2} < \epsilon_{ad} \quad (2.16)$$

Für das Abbruchkriterium der Anreicherung gehen wir von einer Lösungsdarstellung folgender Form aus:

$$u(x, t) \approx \sum_{i=1}^M X_i(x) \cdot Z_i(t) \quad (2.17)$$

Abhängig davon ob die exakte Lösung des Problems bekannt ist ergeben sich zwei mögliche Abbruchkriterien:

- exakte Lösung ist bekannt

$$E = \frac{\|u - u^{ref}\|_{L_2}}{\|u^{ref}\|_{L_2}} < \epsilon_{enrich} \quad (2.18)$$

- exakte Lösung ist nicht bekannt

$$E = \frac{\|\frac{\partial}{\partial t}u - \tau\Delta u - f(x, t)\|_{L_2}}{\|f(x, t)\|_{L_2}} < \epsilon_{enrich} \quad (2.19)$$

2.2. Semiimplizite Zeitdiskretisierung und Separation im Ort

Bei der semiimpliziten Zeitdiskretisierung wird ein θ -Schema für die Zeitkomponente angewandt und dann ein Separationsansatz für die Ortskomponenten genutzt. Ausgehend von einer Schrittweite h für die Zeitdiskretisierung wird für jeden Zeitschritt $t = 0, \dots, T$ eine Lösung der Form (2.2) gesucht, wobei die Funktionen u_i^t aus dieser Darstellung jeweils auf $\bar{\Omega}$ definiert sind. Erneut ausgehend von (2.3) ergibt sich mit dem Parameter $\theta \in [0, 1]$ und der Zeitschrittweite h folgende Diskretisierung der Gleichung:

$$\frac{u^{t+1} - u^t}{h} - \tau\Delta(\theta u^{t+1} + (1 - \theta)u^t) = f(x, t) \quad (2.20)$$

Die schwache Formulierung dieses Problems lautet dann:

Suche für alle $t = 0, \dots, T$ ein $u(x)$ das die Anfangs- und Randwerte (2.4) erfüllt, sodass

$$\int_{\Omega} \left(\frac{u^{t+1} - u^t}{h} - \tau\Delta(\theta u^{t+1} + (1 - \theta)u^t) - f(x, t) \right) \cdot v(x) \, dx = 0 \quad (2.21)$$

für alle $v(x)$ in einem geeigneten Funktionenraum.

Zu einem festen Zeitschritt t berechnen wir nun die Funktionen der Summe (2.2). Wir gehen wie beim Separationsansatz aus Abschnitt 2.1 wieder davon aus, dass die Funktionen $\{u_i^t\}_{i=1, \dots, M}$ für $0 \leq m < M$ bereits berechnet wurden. Also haben wir in Iterationsschritt m eine Lösungsdarstellung der Form

$$U^t(x) \approx \sum_{i=0}^m u_i^t(x) + u_{m+1}^t(x) \quad (2.22)$$

Setzen wir dies in die schwache Formulierung ein und benutzen die Green'sche Formel erhalten wir

$$\int_{\Omega} u_m^{t+1} v \, dx + h\theta\tau \int_{\Omega} \nabla u_m^{t+1} \nabla v \, dx = h \int_{\Omega} f v \, dx + \int_{\Omega} U^t v \, dx \quad (2.23)$$

$$- h(1-\theta)\tau \int_{\Omega} \nabla U^t \nabla v \, dx \quad (2.24)$$

$$- \sum_{i=0}^{m_{t+1}-1} \int_{\Omega} u_i^{t+1} v \, dx \quad (2.25)$$

$$- h\theta\tau \sum_{i=0}^{m_{t+1}-1} \int_{\Omega} \nabla u_i^{t+1} \nabla v \, dx \quad (2.26)$$

Dieses Problem kann mit einem Alternating Directions Algorithmus gelöst werden. Dazu nutzen wir die Produktstruktur der einzelnen Funktionen. Für die Berechnung der Komponente (k) im Alternating Directions Algorithmus setzen wir als Testfunktion v nun

$$v = u_m^{t+1,(1)} \cdot \dots \cdot v^{(k)} \cdot \dots \cdot u_m^{t+1,(D)} \quad (2.27)$$

Damit ergibt sich folgendes 1D-Problem :

$$\begin{aligned} & \left(\prod_{\substack{j=1 \\ j \neq k}}^D a_j \right) \cdot \int_{\Omega_k} u_m^{t+1,(k)} v^{(k)} \, dx_k \\ & + h\theta\tau \sum_{j=1}^D \left(\left(\prod_{\substack{l=1 \\ l \neq j, l \neq k}}^D d_l \right) \cdot \int_{\Omega_j} \left(\frac{\partial}{\partial x_j} u_m^{t+1,(j)} \right)^2 \, dx_j \cdot \int_{\Omega_k} u_m^{t+1,(k)} v^{(k)} \, dx_k \right) \\ & = h \int_{\Omega} f \cdot u_m^{t+1,(1)} \cdot \dots \cdot v^{(k)} \cdot \dots \cdot u_m^{t+1,(D)} \, dx \\ & + \sum_{i=0}^{m_t} \left(\left(\prod_{\substack{j=1 \\ j \neq k}}^D c_{i,j} \right) \cdot \int_{\Omega_k} u_i^{t,(k)} v^{(k)} \, dx_k \right) \\ & - h(1-\theta)\tau \sum_{i=0}^{m_t} \left(\sum_{l=1}^D \left(\left(\prod_{\substack{j=1 \\ j \neq k, j \neq l}}^D c_{i,j} \right) \cdot \int_{\Omega_k} u_i^{t,(k)} v^{(k)} \, dx_k \cdot \int_{\Omega_l} \frac{\partial}{\partial x_l} u_i^{t,(l)} \frac{\partial}{\partial x_l} u_m^{t+1,(l)} \, dx_l \right) \right) \\ & - \sum_{i=0}^{m_{t+1}-1} \left(\left(\prod_{\substack{j=1 \\ j \neq k}}^D b_{i,j} \right) \cdot \int_{\Omega_k} u_i^{t+1,(k)} v^{(k)} \, dx_k \right) \\ & - h\theta\tau \sum_{i=0}^{m_{t+1}-1} \left(\sum_{l=1}^D \left(\left(\prod_{\substack{j=1 \\ j \neq k, j \neq l}}^D b_{i,j} \right) \cdot \int_{\Omega_k} u_i^{t+1,(k)} v^{(k)} \, dx_k \cdot \int_{\Omega_l} \frac{\partial}{\partial x_l} u_i^{t+1,(l)} \frac{\partial}{\partial x_l} u_m^{t+1,(l)} \, dx_l \right) \right) \end{aligned} \quad (2.28)$$

wobei die Konstanten $a_j, b_{i,j}, c_{i,j}$ und d_l wie folgt definiert sind:

$$a_j = \int_{\Omega_j} \left(u_m^{t+1,(j)} \right)^2 dx_j \quad (2.29)$$

$$b_{i,j} = \int_{\Omega_j} u_i^{t+1,(j)}(x_j) u_m^{t+1,(j)}(x_j) dx_j \quad (2.30)$$

$$c_{i,j} = \int_{\Omega_j} u_i^{t,(j)}(x_j) u_m^{t+1,(j)}(x_j) dx_j \quad (2.31)$$

$$d_l = \int_{\Omega_l} \left(u_m^{t+1,(l)} \right)^2 dx_l \quad (2.32)$$

Diese eindimensionalen Probleme können mit einer standardmäßigen Diskretisierung, wie Finiten-Elementen, gelöst werden.

2.2.1. Abbruchkriterium

Falls die exakte Lösung bekannt ist, kann das Abbruchkriterium wie bereits in Abschnitt 2.1.1 definiert werden:

$$\frac{\|u - u^{ref}\|_{L_2}}{\|u^{ref}\|_{L_2}} < \epsilon_{enrich} \quad (2.33)$$

Ist die exakte Lösung nicht bekannt so wird folgendes Abbruchkriterium angewandt:

$$\|u_m^t - u_{m-1}^t\|_{L_2} < \epsilon_{enrich} \quad (2.34)$$

Für den Alternating Directions Algorithmus ist die exakte Lösung im Allgemeinen nicht bekannt. Daher nutzen wir wieder den Fehler zwischen den Produkttermen in zwei aufeinanderfolgenden Iterationsschritten q und $q - 1$ als Kriterium. Das Abbruchkriterium lautet also:

$$\|u_m^{t,(q)} - u_m^{t,(q-1)}\|_{L_2} < \epsilon_{ad} \quad (2.35)$$

3. Verallgemeinerung auf Tensoren

Die in Kapitel 2 beschriebene Vorgehensweise soll nun analog zu [5] durch eine Tensornotation verallgemeinert werden.

Wir nehmen an, dass die diskrete Form des Problems sich formal wie folgt schreiben lässt:

$$\mathcal{V}^T \mathcal{A} \mathcal{U} = \mathcal{V}^T \mathcal{B} \quad (3.1)$$

mit

$$\begin{aligned} \mathcal{A} &= \sum_{i=1}^{m_A} A_1^i \otimes A_2^i \otimes \cdots \otimes A_D^i \\ \mathcal{B} &= \sum_{i=1}^{m_B} B_1^i \otimes B_2^i \otimes \cdots \otimes B_D^i \\ \mathcal{U} &\approx \sum_{i=1}^M u_1^i \otimes u_2^i \otimes \cdots \otimes u_D^i \end{aligned} \quad (3.2)$$

wobei A_i, B_i und u_i nur für die Koordinate x_i definiert sind. Die separierte Form ergibt sich direkt aus der schwachen Formulierung der Differentialoperatoren.

In Iteration m gehen wir davon aus, dass die Vektoren u_j^i , $1 \leq i \leq m$ und $1 \leq j \leq D$ bekannt sind. Nun suchen wir die Anreicherung der Lösung

$$\mathcal{U} = \sum_{i=1}^m u_1^i \otimes \cdots \otimes u_D^i + W_1 \otimes \cdots \otimes W_D \quad (3.3)$$

mit den neuen Vektoren W_j , $j = 1, \dots, D$. Wir gehen von folgender Form des Testfeldes aus:

$$\mathcal{V} = V_1 \otimes W_2 \otimes \cdots \otimes W_D + \cdots + W_1 \otimes \cdots \otimes W_{D-1} \otimes V_D \quad (3.4)$$

Einsetzen dieser approximativen Form und des Testfeldes in die schwache Formulierung (3.1) ergibt:

$$\begin{aligned} &\sum_{i=1}^{m_A} \sum_{j=1}^m V_1^T A_1^i u_1^j \times \cdots \times W_D^T A_D^i u_D^i + \cdots + \\ &\sum_{i=1}^{m_A} \sum_{j=1}^m W_1^T A_1^i u_1^j \times \cdots \times V_D^T A_D^i u_D^j + \sum_{i=1}^{m_A} V_1^T A_1^i W_1 \times \cdots \times V_D^T A_D^i R_D = \\ &\sum_{i=1}^{m_B} V_1^T B_1^i \times \cdots \times W_D^T B_D^i + \cdots + W_1^T B_1^i \times \cdots \times V_D^T B_D^i \end{aligned} \quad (3.5)$$

Um die Notation zu vereinfachen definieren wir:

$$\sum_{i=1}^{m_C} C_1^i \otimes \cdots \otimes C_D^i = \sum_{i=1}^{m_B} B_1^i \otimes \cdots \otimes B_D^i - \sum_{i=1}^{m_A} \sum_{j=1}^m A_1^i u_1^j \otimes \cdots \otimes A_D^i u_D^j \quad (3.6)$$

3. Verallgemeinerung auf Tensoren

wobei $m_C = m_B + m_A \times n$. Diese Summe enthält nur bekannte Felder. Damit erhalten wir aus Gleichung (3.5):

$$\begin{aligned} & \sum_{i=1}^{m_A} V_1^T A_1^i W_1 \times \cdots \times W_D^T A_D^i W_D + \cdots + \sum_{i=1}^{m_A} W_1^T A_1^i W_1 \times \cdots \times V_D^T A_D^i W_D \\ &= \sum_{i=1}^{m_C} (V_1^T C_1^i \times \cdots \times W_D^T C_D^i + \cdots + W_1^T C_1^i \times \cdots \times V_D^T C_D^i) \end{aligned} \quad (3.7)$$

Dieses stark nichtlineare Problem kann mit einem Alternating Directions Algorithmus gelöst werden. Die Idee ist, mit Startvektoren $W_j^{(0)}$, $j = 1, \dots, D$ zu beginnen oder anzunehmen, dass diese $W_j^{(q)}$ in Iterationsschritt q bereits bekannt sind und diese dann zu updaten. Dazu gibt es zwei unterschiedliche Strategien:

- Update $W_j^{(q+1)}$, $\forall j$, ausgehend von $W_1^{(q)}, \dots, W_{j-1}^{(q)}, W_{j+1}^{(q)}, \dots, W_D^{(q)}$

oder:

- Update $W_j^{(q+1)}$, $\forall j$, ausgehend von $W_1^{(q+1)}, \dots, W_{j-1}^{(q+1)}, W_{j+1}^{(q)}, \dots, W_D^{(q)}$

Letztgenannte Strategie konvergiert schneller. Dagegen hat die erste Variante den Vorteil, dass es möglich ist parallele Rechnerplattformen zu nutzen und jeden Vektor gleichzeitig zu updaten. Der Fixpunkt dieses Verfahrens ergibt nun den neuen Anreicherungsvektor $u_j^{m+1} = W_j$, $j = 1, \dots, D$.

4. Fehler- und Laufzeitanalyse

4.1. Fehleranalyse

Die Art der Lösungsdarstellung in der PGD beinhaltet zwei Möglichkeiten um die Genauigkeit zu erhöhen. Dementsprechend müssen auch zwei Fehlerquellen untersucht werden.

Zum einen bestimmt die Anzahl der Summenterme aus Darstellung (2.1) bzw. (2.2) die Genauigkeit der berechneten Lösung. Hierbei muss bemerkt werden, je mehr Terme der Summe hinzugefügt werden, desto stärker konvergiert die Lösung gegen das volle Tensorprodukt resultierend aus den Basen der verschiedenen Gebiete Ω_i bzw. $(0, T_{max}]$.

Die zweite Einflussgröße liegt in der Verfeinerung, die auf jedes Gitter der einzelnen Teilgebiete Ω_i bzw. $(0, T_{max}]$ angewendet werden müsste, um die Genauigkeit der Lösung zu verbessern.

Der Fehlerschätzer für den ersten Teil basiert auf der Lösung des sogenannten Dualen Problems wie in [2] beschrieben.

Bevor wir diesen Ansatz weiter verfolgen wollen wir aber eine etwas grundlegendere Überlegung anstellen: den Vergleich mit einer vollen statt reduzierten Separationsdarstellung.

4.1.1. Vergleich von reduzierter und voller Tensorapproximation

Wir könnten natürlich das volle Tensorprodukt zu den jeweiligen Koordinaten x_1, \dots, x_D, t benutzen und das dazugehörige diskrete System $\mathcal{A}\mathcal{U} - \mathcal{B} = 0$ der PDG lösen. Der Fehler einer so berechneten Lösung \mathcal{U} in Bezug auf eine exakte Lösung \mathcal{U}^{ex} ist abhängig vom gewählten Gitter. Wenn wir $N^{(i)}$ Knoten für jedes Gebiet $\Omega_i, i = 1, \dots, D$ und $(0, T_{max}]$ mit $\Omega = \Omega_1 \times \dots \times \Omega_D$ annehmen, so ist die Anzahl der approximierenden Funktionen $n_1 \times \dots \times n_D \times n_{D+1}$.

Benutzt man die beschriebene Verfahrensstrategie nun für den Grenzfall mit $N = n_1 \times \dots \times n_D \times n_{D+1}$ so ergibt sich theoretisch die gleiche Lösung, die wir im Fall eines vollständigen Tensorprodukts erhalten hätten.

Folglich ist die erste Möglichkeit um die Genauigkeit der reduzierten Darstellung zu bestimmen die Berechnung von

$$\epsilon = \|\mathcal{A}\mathcal{U} - \mathcal{B}\|_2 \quad (4.1)$$

$$= \sqrt{\mathcal{U}^T \mathcal{A}^T \mathcal{A} \mathcal{U} + \mathcal{B}^T \mathcal{B} - 2\mathcal{B}^T \mathcal{A} \mathcal{U}} \quad (4.2)$$

wovon jeder Term wie folgt berechnet werden kann:

$$\begin{aligned} \mathcal{U}^T \mathcal{A}^T \mathcal{A} \mathcal{U} &= \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^{m_A} \sum_{h=1}^{m_A} \prod_{d=1}^D u_d^{iT} A_d^{kT} A_d^h u_d^j \\ \mathcal{B}^T \mathcal{B} &= \sum_{i=1}^{m_B} \sum_{j=1}^{m_B} \prod_{d=1}^D B_d^{iT} B_d^j \\ \mathcal{B}^T \mathcal{A} \mathcal{U} &= \sum_{i=1}^{m_B} \sum_{j=1}^M \sum_{k=1}^{m_A} \prod_{d=1}^D B_d^{iT} A_d^k u_d^j \end{aligned} \quad (4.3)$$

4.1.2. Fehlerabschätzung basierend auf dem dualen Problem

Dieser Abschnitt soll einen Fehlerschätzer für die Anreicherung der Lösung in der PGD einführen. Er basiert auf der Lösung eines sogenannten dualen Problems wie in [2] vorgestellt. Hier wird die exakte Lösung des dualen Problems ersetzt durch eine exaktere Lösung des Ausgangsproblems, also eine Lösung, die mehr Anreicherungsterme enthält. In unserer Situation ist diese Herangehensweise sehr effektiv, da lediglich zusätzliche Anreicherungsterme berechnet werden müssen um eine Lösung des dualen Problems zu bestimmen.

Wir gehen von einem Problem in schwacher Formulierung der Form

$$a(u, v) = b(v) \quad (4.4)$$

auf $\Omega = \Omega_1 \times \dots \times \Omega_D$ aus, welches diskret geschrieben werden kann als

$$\mathcal{V}^T \mathcal{A} \mathcal{U} = \mathcal{V}^T \mathcal{B} \quad (4.5)$$

wobei wie in Kapitel 3 gilt

$$\mathcal{A} = \sum_{i=1}^{m_A} A_1^i \otimes A_2^i \otimes \dots \otimes A_D^i, \quad (4.6)$$

$$\mathcal{B} = \sum_{i=1}^{m_B} B_1^i \otimes B_2^i \otimes \dots \otimes B_D^i. \quad (4.7)$$

$$(4.8)$$

Die Lösung \mathcal{U} in Iterationsschritt m kann dann geschrieben werden als

$$\mathcal{U}_{ap} \approx \sum_{i=1}^m u_1^i \otimes u_2^i \otimes \dots \otimes u_D^i. \quad (4.9)$$

Wir interessieren uns nun für die Funktion $o(\mathcal{U})$, die Ergebnisse des dualen Problems. Wir gehen davon aus, dass der Operator der diese Ergebnisse bestimmt linear ist und eine Summenprodukt-darstellung $\mathcal{O} = \sum_{i=1}^{m_o} O_1^i \otimes \dots \otimes O_D^i$ besitzt. Dann lässt sich das Ergebnis wie folgt auswerten:

$$o(\mathcal{U}_{ap}) = \mathcal{O} \left(\sum_{i=1}^m u_1^i \otimes u_2^i \otimes \dots \otimes u_D^i \right) \quad (4.10)$$

$$= \sum_{i=1}^m \sum_{j=1}^{m_o} \left(u_1^{iT} O_1^j \right) \cdot \dots \cdot \left(u_D^{iT} O_D^j \right). \quad (4.11)$$

Nun können wir den Fehler dieses Ergebnisses berechnen durch die Lösung der Gleichung

$$a(\Phi, v) = o(v) \quad (4.12)$$

bzw. in diskreter Form

$$\mathcal{A}^T \Phi = \mathcal{O}. \quad (4.13)$$

Für eine gute Fehlerabschätzung benötigen wir eine möglichst genaue Lösung des dualen Problems. In unserem Kontext können wir die Lösung des dualen Problems (4.13) wie folgt schreiben:

$$\Phi \approx \sum_{i=1}^{m_\Phi} \varphi_1^i \otimes \cdots \otimes \varphi_D^i \quad (4.14)$$

Jetzt können wir den Fehler des Ergebnisses über

$$o(e) = b(\Phi) - a(\mathcal{U}_{ap}, \Phi) \quad (4.15)$$

berechnen.

4.1.3. Fehlerabschätzung für die niederdimensionalen Probleme

Die Fehlerabschätzungen für die niedrigdimensionalen Probleme (2.11), (2.15) und (2.28), die im Laufe des Verfahrens gelöst werden müssen, entsprechen denen der jeweils angewandten Lösungsmethoden.

Im Falle des Separationsansatzes in Zeit und Ort benötigen wir einen Löser für die gewöhnliche Differentialgleichung der Zeitkomponente (2.15) und ein Verfahren zur Lösung der partiellen Differentialgleichung (2.11). Es müssen also die Fehlerabschätzungen zweier Verfahren gesondert betrachtet werden.

Für den Ansatz mit semiimpliziter Zeitdiskretisierung und Separationsansatz im Ort wird für (2.28) ein 1D-Finite Elemente Löser benötigt. Hier ist für die Fehlerabschätzung die Wahl des Ansatzraumes und dessen Güte zur Approximation der Funktionen darin entscheidend. Dieses Thema wird in den Standardwerken zu Finite-Elemente Methoden ausführlich behandelt. Wir verweisen deshalb an dieser Stelle auf [4].

Die Wahl des Lösers für die gewöhnliche Differentialgleichung (2.15) hängt von verschiedenen Faktoren ab. Hierauf wollen wir hier nicht weiter eingehen. Eine Vielzahl von Lösungsverfahren für gewöhnliche Differentialgleichungen, sowie deren Besonderheiten werden unter anderem in [6] ausführlich behandelt. Zur Lösung der partiellen Differentialgleichung kann ebenfalls jedes passende Verfahren herangezogen werden. Auch hier wollen wir auf eine nähere Betrachtung verzichten. Klassische Verfahren wie Finite Differenzen oder Finite Elemente werden in [4] beschrieben. Auch ein Separationsansatz oder Mehrgittermethoden stellen eine Möglichkeit dar.

4.2. Komplexität und Laufzeit

In diesem Abschnitt wollen wir die Komplexität der PGD etwas genauer betrachten. Auch hier unterscheiden wir wieder zwischen den beiden Ansätzen: Separation in Zeit und Ort und semiimpliziter Zeitdiskretisierung mit Separationsansatz im Ort.

Das Ergebnis der Laufzeitanalyse wird in beiden Fällen maßgeblich von der Form der rechten Seite f in Gleichung (2.3) bestimmt. Liegt f in Produktstruktur oder Summenproduktstruktur vor kann das L_2 -Skalarprodukt $(f, v)_{L_2}$ effizient berechnet werden, wie wir in Kapitel 5 sehen werden. Ist dies nicht der Fall muss hier eine mehrdimensionale Integration durchgeführt werden, die das Laufzeitergebnis stark beeinflusst, da hier die Dimension je nach Verfahren verschieden stark exponentiell eingeht. Wir wollen deshalb den Fall betrachten in dem die rechte Seite f in

Produktform vorliegt.

Wir beginnen mit der Analyse des Separationsansatzes in Zeit und Ort.

4.2.1. Separation in Zeit und Ort

Wählen wir den in Abschnitt 2.1 beschriebenen Ansatz ergibt sich ein Rechenaufwand von

$$\mathcal{O}(M \cdot K(P(N, D) + G(T))) \quad (4.16)$$

Dabei steht $P(N, D)$ für den Rechenaufwand der Lösung der partiellen Differentialgleichung zur Bestimmung von $W(x)$ in (2.11) abhängig von der Größe der Dimension D und der Anzahl der Diskretisierungspunkte N . $G(T)$ bezeichnet hier den Rechenaufwand zur Lösung der gewöhnlichen Differentialgleichung zur Berechnung von $Y(t)$ in Gleichung (2.15) abhängig von der Anzahl der Zeitschritte T . Mit K bezeichnen wir das Maximum über die Anzahl der nötigen Iterationsschritte in jedem Aufruf des Alternating Directions Algorithmus, also $K = \max_{m=1, \dots, M} \{K_m\}$. Schließlich bezeichnet M die Anzahl der für Konvergenz notwendigen Summenterme aus Darstellung (2.1).

Der Rechenaufwand zur Lösung der gewöhnlichen Differentialgleichung (2.15) ist abhängig von der Wahl des Verfahrens. Hierauf wollen wir nicht näher eingehen. Beschreibungen von Verfahren und deren Eigenschaften bietet wie bei der Fehlerabschätzung [6].

Dasselbe gilt für die Lösung der partiellen Differentialgleichung. Hier gibt es viele Möglichkeiten die je nach Anwendungsfall Vor- oder Nachteile mitbringen. Eine Einführung in klassische Verfahren bietet wie bereits in Abschnitt 4.1.3 erwähnt [4].

4.2.2. Separation für die Ortskomponente

Wird der in Abschnitt 2.2 beschriebene Ansatz gewählt erhalten wir einen Rechenaufwand von

$$\mathcal{O}(T \cdot M \cdot K \cdot D \cdot P(N)) \quad (4.17)$$

Wir bezeichnen dabei mit $P(N)$ den Rechenaufwand zur Lösung der 1D-Probleme (2.28) innerhalb des Alternating Directions Algorithmus, abhängig von der Anzahl der Diskretisierungspunkte N . Die Anzahl der Dimensionen bezeichnen wir mit D . K bezeichnet das Maximum über die Iterationsschritte im Alternating Directions Algorithmus. Diesmal wird das Maximum über m und t gebildet, da der Alternating Directions Algorithmus in jedem Zeitschritt t und jedem Anreicherungsschritt m benötigt wird. M bezeichnet das Maximum über die für die Konvergenz notwendigen Summanden in der Lösungsdarstellung abhängig vom Zeitschritt t , also $M = \max_{t=1, \dots, T} \{M_t\}$. Mit T wird die Anzahl der Zeitschritte durch die semiimplizite Zeitdiskretisierung bezeichnet.

Das Zustandekommen dieses Rechenaufwands wird auch im Pseudocode in Abschnitt 5.4.3 deutlich.

Wie auch im Fall des Ansatzes mit Separation in Zeit und Ort wollen wir nicht näher auf die Wahl des Lösungsverfahrens für die 1D-Probleme und deren Laufzeit eingehen.

5. Besonderheiten der Implementierung

5.1. Eigenschaften der Basisfunktionen

Nun wollen wir noch auf einige Probleme und Besonderheiten bei der Implementierung des beschriebenen Verfahrens eingehen.

In der Implementierung zu dieser Arbeit wird als Löser für die 1D-Probleme im Alternating Directions Algorithmus ein Finite Elemente Löser benutzt. Hierzu wird auf dem Gebiet Ω für jede Dimension ein Gitter mit Gitterpunkten z_n , $n = 0, \dots, N$, und dazugehörigen Basisfunktionen ϕ_n , $n = 0, \dots, N$, erzeugt. Als Basiselemente des Raumes der Testfunktionen dienen die Hutfunktionen ϕ_n definiert durch

$$\phi_n(x) = \begin{cases} 0 & \text{für } x \leq z_{n-1} \text{ oder } x \geq z_{n+1}, \\ \frac{x-z_{n-1}}{h_{n-1}} & \text{für } z_{n-1} \leq x \leq z_n, \\ \frac{z_{n+1}-x}{h_n} & \text{für } z_n \leq x \leq z_{n+1}, \end{cases} \quad (5.1)$$

wobei $h_n := z_{n+1} - z_n$. Die beiden Basisfunktionen ϕ_0 und ϕ_N sind jeweils nur auf dem vorhandenen Träger definiert, sodass anschaulich halbe Hutfunktionen entstehen.

Aus der Wahl der Testfunktionen ergeben sich nun einige nützliche Eigenschaften.

Zum Ersten ergibt sich durch den kompakten Träger der einzelnen Hutfunktionen nur eine dünnbesetzte Matrix im zu lösenden linearen Gleichungssystem, da $\phi_i \cdot \phi_j = 0$ falls $|i - j| \geq 2$ für $i, j = 0, \dots, N$. Dies reduziert den Aufwand zur Lösung des linearen Gleichungssystems enorm. Weiterhin können auch die Berechnungen der 1D-Integrale, die für die Konstanten notwendig sind, effizient durchgeführt werden, da sich diese analytisch berechnen lassen. Für die Konstanten werden immer Integrale vom Produkte zweier 1D-Funktionen berechnet. Diese Funktionen haben jeweils eine Darstellung über die Basisfunktionen ϕ_n . Betrachten wir zwei 1D-Funktionen g und h mit einer Darstellung

$$g(x) = \sum_{n=0}^N \alpha_n \phi_n(x) \quad \text{und} \quad h(x) = \sum_{n=0}^N \beta_n \phi_n(x) \quad (5.2)$$

so ergibt sich für deren Produkt durch die kompakten Träger folgende einfache Form

$$\begin{aligned} g(x) \cdot h(x) &= \sum_{n=0}^N \alpha_n \beta_n \phi_n(x)^2 + \sum_{n=0}^{N-1} (\alpha_n \beta_{n+1} + \alpha_{n+1} \beta_n) \phi_n(x) \phi_{n+1}(x) \\ &= \sum_{n=0}^{N-1} (\alpha_n \beta_n \phi_n(x)^2 + (\alpha_n \beta_{n+1} + \alpha_{n+1} \beta_n) \phi_n(x) \phi_{n+1}(x)) + \alpha_N \beta_N \phi_N(x)^2 \end{aligned} \quad (5.3)$$

Wenn wir nun das Integral eines solchen Produkts berechnen möchten können wir wegen der Additivität des Integrals die einzelnen Summanden integrieren und später aufsummieren. Da wir sowohl Produkte von normalen Funktionen, als auch Produkte von deren Ableitungen, für die

Konstanten berechnen müssen, werden folgende vier 1D-Integrale benötigt, die jeweils analytisch berechnet werden können:

$$\int_{\Omega} \phi_n(x)^2 dx = \begin{cases} \frac{z_{n+1}-z_n}{3} & \text{für } n = 0, \\ \frac{z_{n+1}-z_{n-1}}{3} & \text{für } n = 1, \dots, N-1, \\ \frac{z_n-z_{n-1}}{3} & \text{für } n = N, \end{cases} \quad (5.4)$$

$$\int_{\Omega} \phi_n(x)\phi_{n+1}(x)dx = \frac{z_{n+1} - z_n}{6} \quad \text{für } n = 0, \dots, N-1,$$

$$\int_{\Omega} \left(\frac{\partial}{\partial x} \phi_n(x) \right)^2 dx = \begin{cases} \frac{1}{z_{n+1}-z_n} & \text{für } n = 0, \\ \frac{1}{z_{n+1}-z_n} + \frac{1}{z_n-z_{n-1}} & \text{für } n = 1, \dots, N-1, \\ \frac{1}{z_n-z_{n-1}} & \text{für } n = N, \end{cases} \quad (5.5)$$

$$\int_{\Omega} \frac{\partial}{\partial x} \phi_n(x) \frac{\partial}{\partial x} \phi_{n+1}(x) dx = \frac{1}{z_n - z_{n+1}} \quad \text{für } n = 0, \dots, N-1.$$

5.2. L_2 -Norm in Abbruchkriterien

Als Abbruchkriterien für die Anreicherung der Lösung und des Alternating Directions Algorithmus muss eine L_2 -Norm auf dem Gebiet Ω berechnet werden. Da hierfür keine mehrdimensionale Integration verwendet werden soll, wollen wir hier eine Rechenvorschrift beschreiben, welche die Summenproduktstruktur der Lösung nutzt um die Normen effizient zu berechnen.

Wir beginnen mit dem Abbruchkriterium für den Alternating Directions Algorithmus. Das Kriterium nutzt die Produkte aus dem Alternating Directions Algorithmus im aktuellen Rechenschritt q und dem vorherigen $q-1$ und lautet wie in Abschnitt 2.2.1 beschrieben

$$\|u^{(q)} - u^{(q-1)}\|_{L_2} \quad (5.6)$$

Setzen wir nun die Produktstruktur der beiden Terme $u^{(q)} = \prod_{i=1}^D u^{i,(q)}$ und $u^{(q-1)} = \prod_{i=1}^D u^{i,(q-1)}$ ein ergibt sich

$$\left\| \prod_{i=1}^D u^{i,(q)} - \prod_{i=1}^D u^{i,(q-1)} \right\|_{L_2}^2 = \int_{\Omega} \left(\prod_{i=1}^D u^{i,(q)} - \prod_{i=1}^D u^{i,(q-1)} \right)^2 dx \quad (5.7)$$

$$= \int_{\Omega} \left(\prod_{i=1}^D u^{i,(q)} \right)^2 dx - 2 \cdot \int_{\Omega} \prod_{i=1}^D u^{i,(q)} u^{i,(q-1)} dx \quad (5.8)$$

$$+ \int_{\Omega} \left(\prod_{i=1}^D u^{i,(q-1)} \right)^2 dx \quad (5.9)$$

Diese Terme können mit dem Satz von Fubini auf Produkte von eindimensionalen Integralen reduziert werden. Damit erhalten wir für (5.6) insgesamt

$$\prod_{i=1}^D \int_{\Omega_i} \left(u^{i,(q)} \right)^2 dx_i - 2 \cdot \prod_{i=1}^D \int_{\Omega_i} u^{i,(q)} u^{i,(q-1)} dx_i + \prod_{i=1}^D \int_{\Omega_i} \left(u^{i,(q-1)} \right)^2 dx_i \quad (5.10)$$

Mit den Basisintegralen aus (5.4) und Formel (5.3) können wir diese Terme effizient berechnen.

Dieselbe Vorgehensweise können wir auch für das Abbruchkriterium der Anreicherung der Lösung anwenden. Wir erhalten also

$$\|u_m - u_{m-1}\|_{L_2}^2 = \left\| \sum_{k=1}^m \prod_{i=1}^D u_k^{(i)} - \sum_{k=1}^{m-1} \prod_{i=1}^D u_k^{(i)} \right\|_{L_2}^2 \quad (5.11)$$

$$= \int_{\Omega} \left(\sum_{k=1}^m \prod_{i=1}^D u_k^{(i)} \right)^2 dx - 2 \cdot \int_{\Omega} \left(\sum_{k=1}^m \prod_{i=1}^D u_k^{(i)} \right) \cdot \left(\sum_{k=1}^{m-1} \prod_{i=1}^D u_k^{(i)} \right) dx \quad (5.12)$$

$$+ \int_{\Omega} \left(\sum_{k=1}^{m-1} \prod_{i=1}^D u_k^{(i)} \right)^2 dx \quad (5.13)$$

Wenden wir nun wieder den Satz von Fubini an ergibt sich

$$\sum_{k=1}^m \sum_{l=1}^m \prod_{i=1}^D \int_{\Omega_i} u_k^{(i)} u_l^{(i)} dx_i - 2 \cdot \sum_{k=1}^m \sum_{l=1}^{m-1} \prod_{i=1}^D \int_{\Omega_i} u_k^{(i)} u_l^{(i)} dx_i + \sum_{k=1}^{m-1} \sum_{l=1}^{m-1} \prod_{i=1}^D \int_{\Omega_i} u_k^{(i)} u_l^{(i)} dx_i \quad (5.14)$$

Diese Terme können wir wieder mit (5.4) und (5.3) effizient berechnen.

5.3. Mehrdimensionale Integration für das L_2 -Skalarprodukt

Ein Problem bei der effizienten Implementierung der PGD stellt auf der rechten Seite das L_2 -Skalarprodukt dar. Hier muss ein mehrdimensionales Integral berechnet werden, sofern die rechte Seite f der Wärmeleitungsgleichung nicht bereits in einer Produktform bzw. Summenproduktform vorliegt.

Liegt f in einer Produkt- oder Summenproduktform vor, so können wir mit Hilfe des Satzes von Fubini und den Eigenschaften des Integrals das mehrdimensionale Integral auf mehrere 1D-Integrale reduzieren und diese wie oben beschrieben effizient berechnen.

Liegt also $f : \mathbb{R}^D \rightarrow \mathbb{R}$ in der Form

$$f(x) = \sum_{i=0}^{m_f} f_i^{(1)}(x_1) \cdot \dots \cdot f_i^{(D)}(x_D) \quad (5.15)$$

vor, so lässt sich das L_2 -Skalarprodukt mit der Testfunktion v wie folgt berechnen

$$(f, v)_{L_2} = \int_{\Omega} f \cdot v dx \quad (5.16)$$

$$= \int_{\Omega} \left(\sum_{i=0}^{m_f} f_i^{(1)}(x_1) \cdot \dots \cdot f_i^{(D)}(x_D) \right) \cdot v dx \quad (5.17)$$

Wird nun noch die Form von v im Alternating Directions Algorithmus zu Dimension d in Anreicherungsstschritt $m + 1$ berücksichtigt erhalten wir

$$(f, v)_{L_2} = \int_{\Omega} \left(\sum_{i=0}^{m_f} f_i^{(1)} \cdot \dots \cdot f_i^{(D)} \right) \cdot u_{m+1}^{(1)} \cdot \dots \cdot v^{(d)} \cdot \dots \cdot u_{m+1}^{(D)} dx \quad (5.18)$$

$$= \sum_{m=0}^{m_f} \left[\left(\prod_{\substack{k=0 \\ k \neq d}}^D \beta_{i,k} \right) \int_{\Omega_d} f_i^{(d)} \cdot v^{(d)} dx_d \right] \quad (5.19)$$

wobei $\beta_{i,k} = \int_{\Omega_k} f_i^{(k)} \cdot u_{m+1}^{(k)} dx_k$.

Liegt f nicht in Produkt- oder Summenproduktform vor, so wurde für diese Arbeit die *Cuhre-Routine* der *Cuba-Library* [13] verwendet um wie schon für die 1D-Integrale Basiselemente zu berechnen, die anschließend in den jeweiligen Schritten zur Berechnung des L_2 -Skalarprodukts genutzt werden können.

Wir berechnen also zu Beginn des Algorithmus für alle Indizes $i_1, \dots, i_D, i_d = 1, \dots, N$ den Ausdruck

$$\int_{\Omega} f(x) \cdot \phi_{i_1}(x_1) \cdot \dots \cdot \phi_{i_D}(x_D) dx \quad (5.20)$$

Das L_2 -Skalarprodukt kann nun anschließend über folgende Vorschrift berechnet werden:

$$(f, v)_{L_2} = \sum_{i_1=1}^N \gamma_{i_1} \left(\sum_{i_2=1}^N \gamma_{i_2} \left(\dots \left(\sum_{i_D=1}^N \gamma_{i_D} \int_{\Omega} f(x_1, \dots, x_D) \cdot \phi_{i_1}(x_1) \cdot \dots \cdot \phi_{i_D}(x_D) dx \right) \dots \right) \right) \quad (5.21)$$

Dabei seien γ_{i_d} die Koeffizienten für die Darstellung in der Hutfunktionenbasis der 1D-Funktionen der Testfunktion v .

5.4. Pseudocode

In diesem Abschnitt wollen wir die Algorithmen der PGD für die Approximation von Funktionen, zur Lösung der Poisson-Gleichung und zur Lösung der Wärmeleitungsgleichung in den beiden betrachteten Varianten in Pseudocode angeben.

5.4.1. Approximation von Funktionen

Wir beginnen mit der Approximation von Funktionen. Als Ausgangspunkt dient uns hier die Gleichung

$$u = f \quad \Omega \quad (5.22)$$

$$u = 0 \quad \partial\Omega \quad (5.23)$$

Aus dieser kann analog zu der Vorgehensweise in Kapitel 2 bei der Wärmeleitungsgleichung das jeweils zu lösende 1D-Problem hergeleitet werden. Dazu betrachten wir die schwache Formulierung des Problems, setzen die Summenproduktform der Lösung u sowie die Testfunktion v für Komponente d ein. Es ergibt sich dann Algorithmus 1.

Algorithm 1 PGD zur Approximation von Funktionen

```

1: Berechne Basiselemente (5.4)
2:  $m = 0$ 
3: while  $\|u_m - u_{m-1}\|_{L_2} > \varepsilon_{enrich}$  und  $m < itmax_{enrich}$  do
4:    $q = 0$ 
5:   Initialisiere  $u_m^{(q)}$ ;
6:   while  $\|u_m^{(q)} - u_m^{(q-1)}\|_{L_2} > \varepsilon_{ad}$  und  $q < itmax_{ad}$  do
7:     for  $d = 1; d < D; d++$  do
8:       Löse
9:       
$$\prod_{\substack{j=1 \\ j \neq d}}^D c_j \int_{\Omega_d} u_m^{d,(q)} v^d dx_d = \int_{\Omega} f \cdot u_m^{1,(q)} \cdot \dots \cdot v^d \cdot \dots \cdot u_m^{D,(q)} dx$$


$$- \sum_{i=0}^{m-1} \prod_{\substack{j=1 \\ j \neq d}}^D b_{i,j} \int_{\Omega_d} u_i^{d,(q)} v^d dx_d$$

10:      wobei  $c_j = \int_{\Omega_j} (u_m^{j,(q)})^2 dx_j$ 
11:      und  $b_{i,j} = \int_{\Omega_j} u_i^j u_m^{j,(q)} dx_j$ 
12:     end for
13:      $q = q + 1$ 
14:   end while
15:    $m = m + 1$ 
16: end while

```

5.4.2. Lösung der Poisson-Gleichung

In diesem Abschnitt soll die Poisson-Gleichung

$$-\Delta u = f \quad \Omega \quad (5.24)$$

$$u = 0 \quad \partial\Omega \quad (5.25)$$

gelöst werden. Um die 1D-Probleme für den Alternating Directions Algorithmus herzuleiten gehen wir wieder über zur schwachen Formulierung des Problems. Anschließend benutzen wir die Green'sche Formel um einen symmetrischen Operator zu erhalten. Das einsetzen der Summenproduktstruktur der Lösung u sowie der Testfunktion v für Komponente d liefert dann die benötigte Gleichung. Algorithmus 2 beschreibt das Verfahren.

Algorithm 2 PGD zur Lösung der Poisson-Gleichung

```

1: Berechne Basiselemente (5.4) und (5.5)
2:  $m = 0$ 
3: while  $\|u_m - u_{m-1}\|_{L_2} > \varepsilon_{enrich}$  und  $m < itmax_{enrich}$  do
4:    $q = 0$ 
5:   Initialisiere  $u_m^{(q)}$ ;
6:   while  $\|u_m^{(q)} - u_m^{(q-1)}\|_{L_2} > \varepsilon_{ad}$  und  $q < itmax_{ad}$  do
7:     for  $d = 1; d < D; d++$  do
8:       Löse
9:       
$$\sum_{i \neq d}^D \left( \left( \prod_{j \neq i, j \neq d}^D c_j \right) \int_{\Omega_i} \left( \frac{\partial}{\partial x_i} u_m^{i(q)} \right)^2 dx_i \int_{\Omega_d} u_m^{d,(q)} v^{(d)} dx_d \right)$$


$$+ \left( \prod_{j \neq d}^D c_j \right) \int_{\Omega_d} \frac{\partial}{\partial x_d} u_m^{d,(q)} \frac{\partial}{\partial x_d} v^{(d)} dx_d$$


$$= \int_{\Omega} f \cdot u_m^{1,(q)} \cdot \dots \cdot v^d \cdot \dots \cdot u_m^{D,(q)} dx$$


$$- \sum_{l=0}^{m-1} \sum_{i \neq d}^D \left( \left( \prod_{j \neq i, j \neq d}^D c_{l,j} \right) \int_{\Omega_i} \frac{\partial}{\partial x_i} u_l^{i,(q)} \frac{\partial}{\partial x_i} u_m^{i,(q)} dx_i \int_{\Omega_d} u_l^{d,(q)} v^d dx_d \right)$$


$$+ \left( \prod_{j \neq d}^D c_{l,j} \right) \int_{\Omega_d} \frac{\partial}{\partial x_d} u_l^{d,(q)} \frac{\partial}{\partial x_d} v^d dx_d$$

       wobei  $c_j = \int_{\Omega_j} \left( u_m^{j,(q)} \right)^2 dx_j$ 
       und  $c_{l,j} = \int_{\Omega_j} u_l^{j,(q)} u_m^{j,(q)} dx_j$ 
10:     end for
11:      $q = q + 1$ 
12:   end while
13:    $m = m + 1$ 
14: end while

```

5.4.3. Lösung der Wärmeleitungsgleichung

In diesem Abschnitt zeigen wir den Pseudocode zur Lösung der Wärmeleitungsgleichung durch die PGD.

Wir beginnen mit dem Verfahren aus Abschnitt 2.1. Dieses wird von Algorithmus 3 beschrieben.

Algorithm 3 PGD zur Lösung der Wärmeleitungsgleichung mit Separation in Zeit und Ort

```

1:  $m = 0$ 
2: while  $\|u_m - u_{m-1}\|_{L_2} > \varepsilon_{enrich}$  und  $m < itmax_{enrich}$  do
3:    $q = 0$ 
4:   Initialisiere  $W^{(q)}$  und  $Y^{(q)}$ 
5:   while  $\|W^{(q)} \cdot Y^{(q)} - W^{(q-1)} \cdot Y^{(q-1)}\|_{L_2} > \varepsilon_{ad}$  und  $q < itmax_{ad}$  do
6:     Löse (2.11)
7:     Löse (2.15)
8:      $q = q + 1$ 
9:   end while
10:   $m = m + 1$ 
11: end while

```

Den Pseudocode für die Variante aus Abschnitt 2.2 beschreibt Algorithmus 4.

Algorithm 4 PGD zur Lösung der Wärmeleitungsgleichung mit Separation im Ort

```

1: Berechne Basiselemente (5.4) und (5.5)
2: for  $t = 0; t < T; t ++$  do
3:    $m_t = 0$ 
4:   while  $\|u_m^t - u_{m-1}^t\|_{L_2} > \varepsilon_{enrich}$  und  $m_t < itmax_{enrich}$  do
5:      $q = 0$ 
6:     Initialisiere  $u_m^{t,(q)}$ 
7:     while  $\|u_m^{t,(q)} - u_m^{t,(q-1)}\|_{L_2} > \varepsilon_{ad}$  und  $q < itmax_{ad}$  do
8:       for  $d = 1; d < D; d ++$  do
9:         Löse (2.28)
10:      end for
11:       $q = q + 1$ 
12:    end while
13:     $m_t = m_t + 1$ 
14:  end while
15: end for

```

6. Numerische Experimente

In diesem Kapitel wollen wir einige numerische Experimente mit der Proper Generalized Decomposition betrachten. Zuerst befassen wir uns mit der Approximation von Funktionen durch die PGD. Im Anschluss wollen wir die Lösung der Poisson-Gleichung berechnen. Dies entspricht im Rechenaufwand der Lösung der Wärmeleitungsgleichung in einem Zeitschritt für die semi-implizite Zeitdiskretisierung oder der Berechnung der Funktion $W(x)$ in (2.11), was in beiden Fällen den Großteil des Rechenaufwandes darstellt.

Als Diskretisierung für die 1D-Probleme wählen wir $N = 100$ uniform verteilte Gitterpunkte auf dem Gebiet $[0, 1]$. Als Abbruchkriterium für den Alternating Directions Algorithmus wählen wir den absoluten Fehler $\epsilon_{ad} = 0.0001$, sowie für die Anreicherung der Lösung den relativen Fehler $\epsilon_{enrich} = 0.01$.

6.1. Approximation von Funktionen

Wir beginnen mit Funktionen die eine Produktstruktur aufweisen. Diese eignen sich besonders gut für den im Verfahren verwendeten Separationsansatz.

6.1.1. Funktionen mit Produktstruktur

Für die Funktionen mit Produktstruktur betrachten wir zuerst als einfaches Beispiel die Funktion $f_1 : \mathbb{R}^2 \rightarrow \mathbb{R}$ definiert durch $f_1(x, y) = x^2 \cdot y^2$ auf dem Einheitsquadrat, also $\Omega = [0, 1]^2$. Wir wollen zunächst die Entstehung der Lösung durch das Verfahren anhand der Abbildungen nachvollziehen. Anschließend widmen wir uns der Fehlerentwicklung im Alternating Directions Algorithmus und der Anreicherung der Lösung sowie der Abhängigkeit der Fehlerentwicklung von der Initialisierung der Funktionen.

6.1.1.1. Entstehung der Lösung

Die Lösung unserer Approximationsaufgabe wird durch die Proper Generalized Decomposition schrittweise gebildet. Jeder Summenterm in der Anreicherung unserer Lösung entsteht durch ein Produkt von eindimensionalen Funktionen. Um die einzelnen Komponenten im Alternating Directions Algorithmus zu berechnen benötigen wir einen Startwert bzw. Startfunktionen. Die Implementierung zu dieser Arbeit benutzt dazu konstante Funktionen $u_m^{(0)}(x) = a$ mit $a \in \mathbb{R}$. In wieweit der Startwert Einfluss auf das Verfahren nimmt wird später (siehe Abschnitt 6.1.1.2) noch behandelt.

In unserem Beispiel wird zuerst, also erster Summenterm, erster Schritt des Alternating Directions Algorithmus der ersten Komponente, die Funktion $u_0^{1,(1)}(x) = 1/3 x^2$ approximiert, wie auch in Abbildung 6.1 zu sehen ist. Die zweite Komponente approximiert dann die Funktion $u_0^{2,(1)}(y) = 3y^2$ wie Abbildung 6.2 zeigt.

Das aus dem Alternating Directions Algorithmus resultierende Produkt $u_0(x, y)$, zu sehen in Abbildung 6.3, bildet den ersten Summenterm der approximierten Lösung.

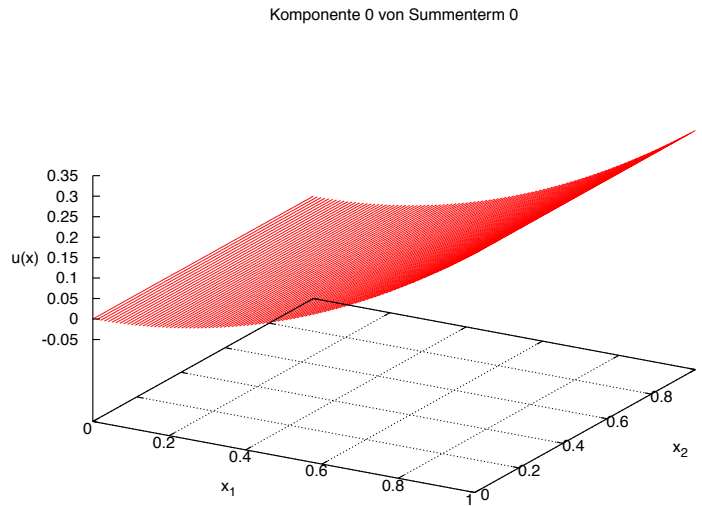


Abbildung 6.1.: Approximation 1. Komponente von $f_1(x, y) = x^2 \cdot y^2$

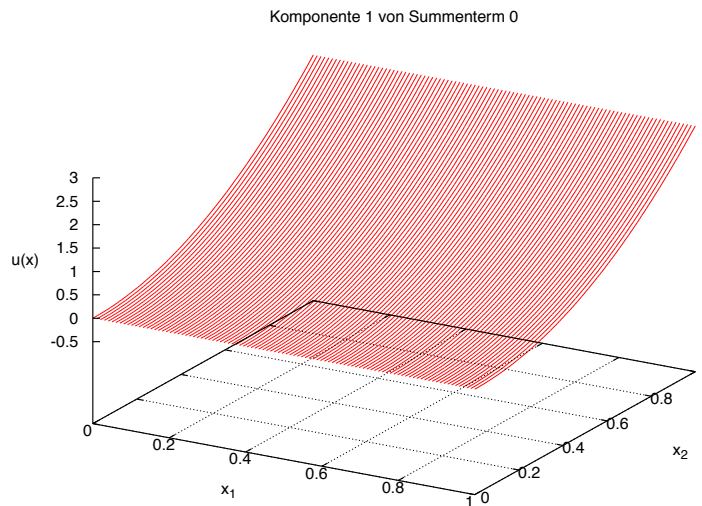


Abbildung 6.2.: Approximation 2. Komponente von $f_1(x, y) = x^2 \cdot y^2$

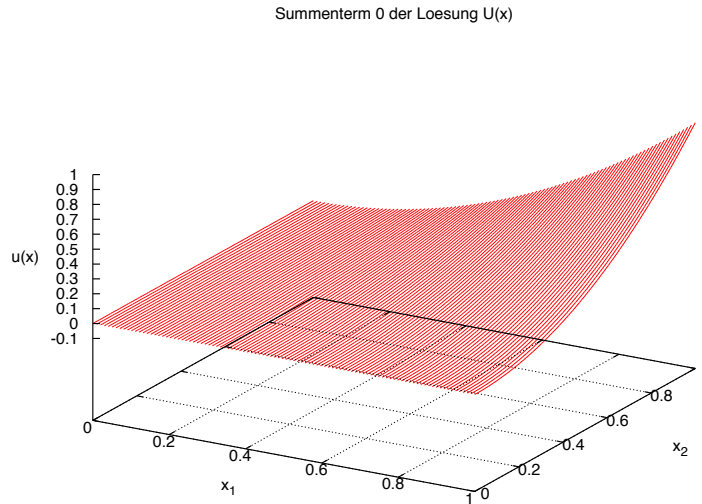


Abbildung 6.3.: Summenterm 0 von $f_1(x, y) = x^2 \cdot y^2$

Da der Fehler bisher noch zu groß ist, werden nun in gleicher Weise weitere 1D-Funktionen und Summenterme berechnet. Diese verbessern die Lösung sukzessive bis die benötigte Genauigkeit erreicht ist. Die am Ende errechnete Lösung zeigt Abbildung 6.4.

6.1.1.2. Fehlerentwicklung und Abhängigkeit von der Initialisierung

Nun wollen wir uns noch der Fehlerentwicklung unseres Beispiels widmen.

In unserem einfachen Beispiel benötigen wir nur sehr wenige Summenterme für die gewünschte Genauigkeit. Die Fehlerentwicklung zeigt Abbildung 6.5. Die genauen Werte finden sich in der folgenden Tabelle:

Tabelle 6.1.: Fehlerentwicklung Anreicherung

Iterationsschritt	Fehler
1	1
2	1
3	2.15477e-05

Hierbei muss bemerkt werden, dass der Fehler im 1. Iterationsschritt immer mit dem Maß in der L_2 -Norm des ersten Summenterms initialisiert wurde, da im ersten Schritt der Anreicherung

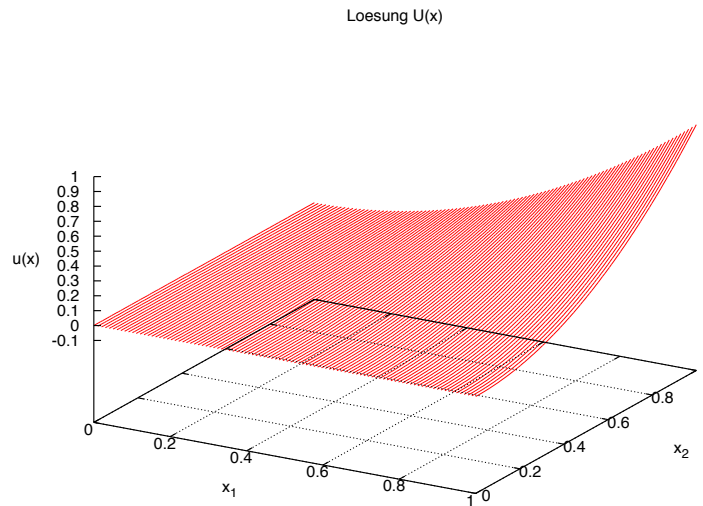


Abbildung 6.4.: Approximierte Lösung zu $f_1(x, y) = x^2 \cdot y^2$

noch nicht zwei aufeinanderfolgende Summenterme für das Abbruchkriterium zur Verfügung stehen.

Jetzt wollen wir den Einfluss unserer Initialisierung im Alternating Directions Algorithmus auf die Fehlerentwicklung untersuchen. Auf die Fehlerentwicklung der Anreicherung hat die Initialisierung nur dann einen Einfluss, wenn ein manueller Abbruch des Alternating Directions Algorithmus notwendig wird (ein Beispiel hierfür zeigt 6.6). Dies kann passieren, wenn das Verfahren gegen ein lokales Minimum konvergiert, anstatt gegen das gesuchte globale Minimum. Nach dem Abbruch wird dieser Fehler durch die Residuumsminimierung in den Anreicherungsschritten wieder ausgeglichen.

Um den Einfluss der Initialisierung auf den Alternating Directions Algorithmus zu untersuchen wurde das Verfahren mit den Werten $a = 1$ (Fehlerentwicklungen in Abbildungen 6.6, 6.7 und 6.8) und $a = 10$ (Fehlerentwicklung in Abbildungen 6.9, 6.10 und 6.11) gestartet. Die Abbildungen zeigen, dass es in den Fehlerentwicklungen keinen relevanten Unterschied für die verschiedenen Startwerte gibt.

Auch die Wahl $a = 100$ erzeugt keine Veränderungen wie Tabelle 6.2 zeigt.

6.1.2. Funktionen in beliebiger Form

In diesem Abschnitt betrachten wir eine rechte Seite f mit beliebiger Struktur. Als Beispiel dient uns die Funktion $f_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ definiert durch $f_2(x, y) = \exp(-x^2 \cdot y^2)$ auf dem Gebiet $\Omega = [-1, 1]^2$. An diesem Beispiel wollen wir die schrittweise Anreicherung der Funktion mit dem Residuum vergleichen. In den Abbildungen 6.12, 6.14 und 6.16, sind die einzelnen Summenterme der berechneten Lösung zu sehen. Die Abbildungen 6.13, 6.15 und 6.17, zeigen das Residuum

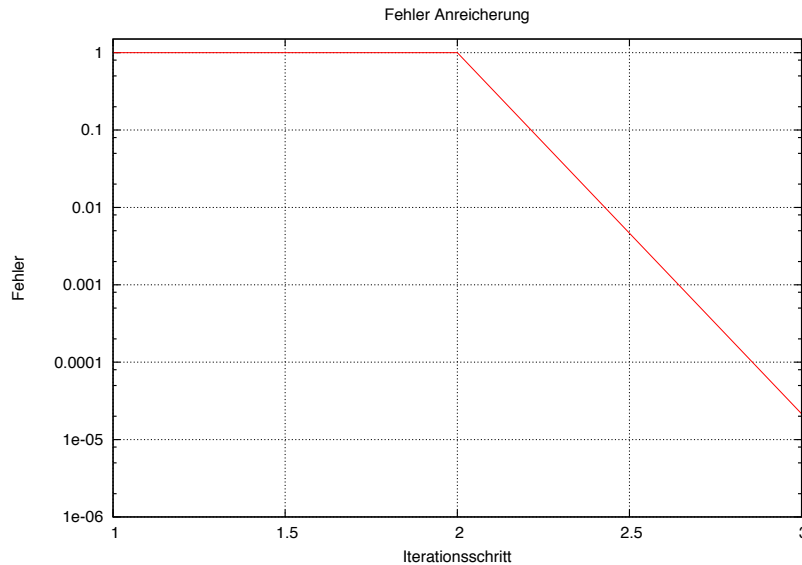


Abbildung 6.5.: Fehlerentwicklung der Anreicherung für $f(x, y) = x^2 y^2$

im jeweiligen Anreicherungsschritt m , also $f(x, y) - \sum_{i=0}^m u_i(x, y)$.

In den beiden Abbildungen 6.18 und 6.19 sind die approximierte Lösung und das verbliebene Residuum zu sehen. Da dieses Residuum klein genug ist, greift unser Abbruchkriterium und wir erhalten eine Lösung mit drei Summentermen.

6.2. Lösung der Poisson-Gleichung

In diesem Abschnitt möchten wir die Lösung der Poisson-Gleichung

$$u = f \quad \Omega \quad (6.1)$$

$$u = 0 \quad \partial\Omega \quad (6.2)$$

berechnen. Wir wählen als rechte Seite f die Funktion $f: \mathbb{R}^D \rightarrow \mathbb{R}$ definiert durch

$$f(x_1, \dots, x_D) = \prod_{i=1}^D x_i^2. \quad (6.3)$$

und als Gebiet $\Omega = [0, 1]^D$.

Die Lösung für $D = 2$ ist in Abbildung 6.20 dargestellt.

Abschließend wollen wir noch die Abhängigkeit zwischen der Anzahl der Dimensionen D , der benötigten Anzahl an Anreicherungstermen M und der benötigten Rechenzeit untersuchen. Dies ist in Tabelle 6.3 dargestellt. Aus diesen Daten wird deutlich, dass kein direkter Zusammenhang zwischen diesen beiden Größen besteht. Auch wenn wir vielleicht erwartet hätten eine steigende

6. Numerische Experimente

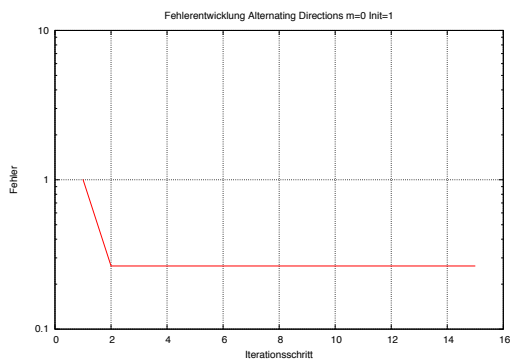


Abbildung 6.6.: Fehlerentwicklung AD $m=0$
 $a=1$

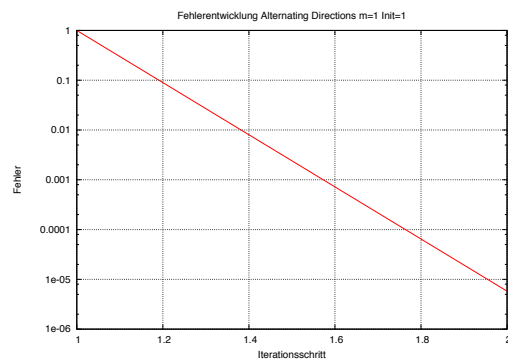


Abbildung 6.7.: Fehlerentwicklung AD $m=1$
 $a=1$

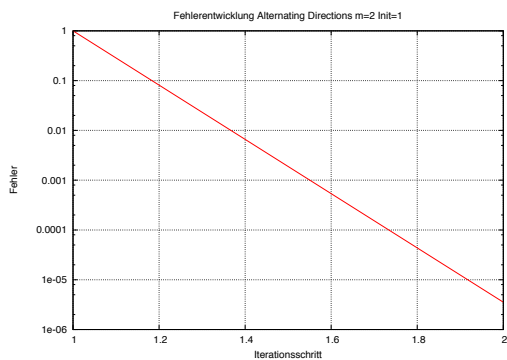


Abbildung 6.8.: Fehlerentwicklung AD $m=2$
 $a=1$

Anzahl an Anreicherungstermen für höhere Dimensionen zu benötigen, so ist dies nicht der Fall. Die benötigte Rechenzeit deckt sich mit den Ergebnissen zur Zahl der benötigten Anreicherungsterme. Die Messungen der Rechenzeit wurden auf einem Computer mit Intel Core i5 1.7GHz Prozessor und 4GB Arbeitsspeicher durchgeführt.

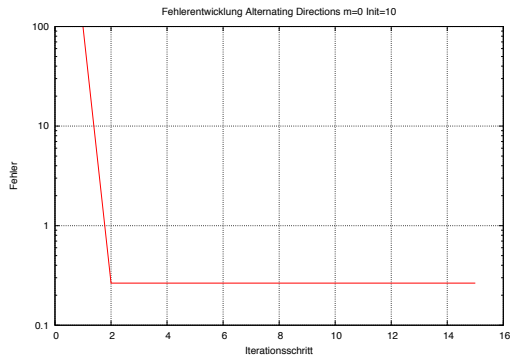


Abbildung 6.9.: Fehlerentwicklung AD $m=0$
 $a=10$

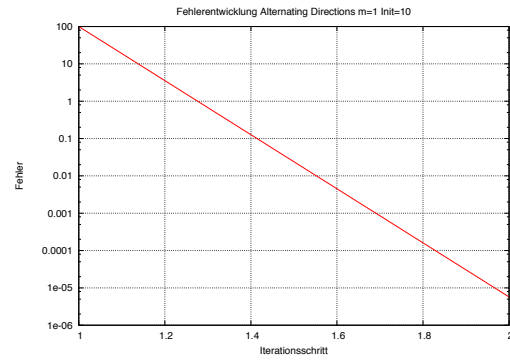


Abbildung 6.10.: Fehlerentwicklung AD $m=1$
 $a=10$

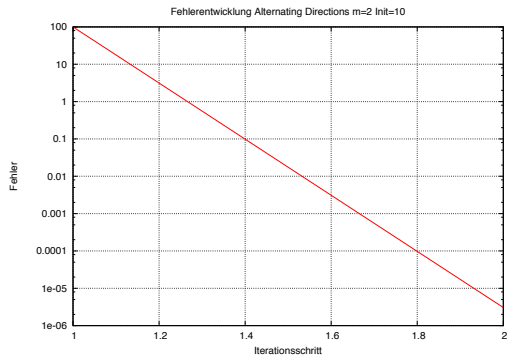


Abbildung 6.11.: Fehlerentwicklung AD $m=2$
 $a=10$

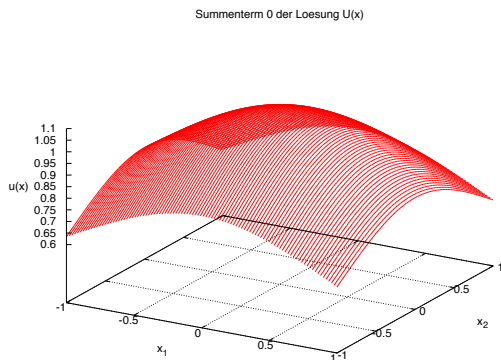


Abbildung 6.12.: 1. Anreicherungsterm

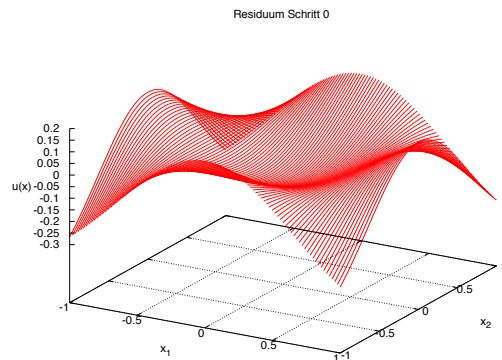


Abbildung 6.13.: Residuum nach 1. Schritt

Tabelle 6.2.: Fehlerentwicklung für Startwert $a=100$

Anreicherungs-schritt					
m=0		m=1		m=2	
Iterationsschritt	Fehler	Iterationsschritt	Fehler	Iterationsschritt	Fehler
1	10000	1	10000	1	10000
2	0.264943	2	5.72711e-06	2	3.05855e-06
3	0.264943				
4	0.264943				
5	0.264943				
6	0.264943				
7	0.264943				
8	0.264943				
9	0.264943				
10	0.264943				
11	0.264943				
12	0.264943				
13	0.264943				
14	0.264943				
15	0.264943				

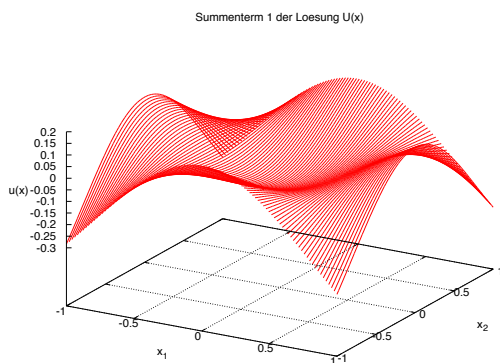


Abbildung 6.14.: 2. Anreicherungs-term

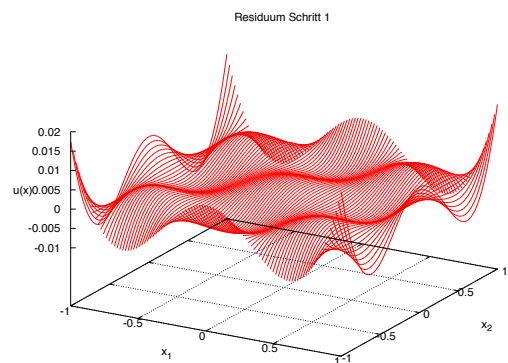


Abbildung 6.15.: Residuum nach 2 Schritten

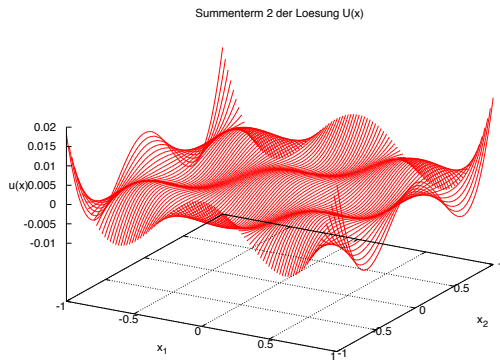


Abbildung 6.16.: 3. Anreicherungsterm

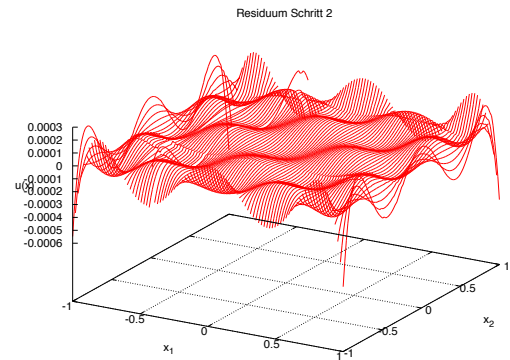


Abbildung 6.17.: Residuum nach 3 Schritten

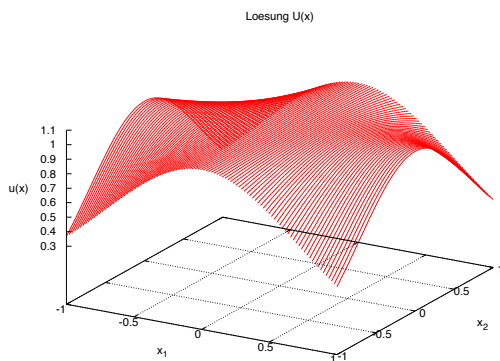


Abbildung 6.18.: Approximierte Lösung

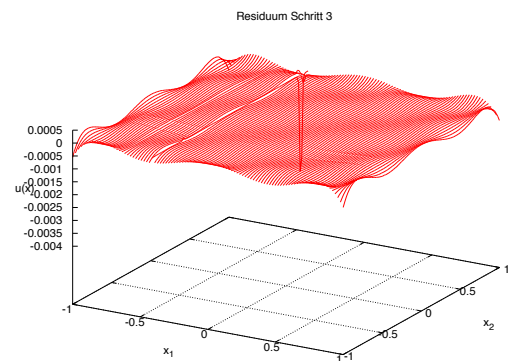


Abbildung 6.19.: Residuum nach 4 Schritten

Tabelle 6.3.: Zusammenhang Dimensionen, Anreicherungstermen und Rechenzeit

Dimensionen	Anreicherungsterme	Rechenzeit in Sekunden
2	6	0.594
3	9	0.935
4	10	1.011
5	14	1.401
6	15	1.620
7	15	1.640
8	7	0.771
9	7	0.769
10	7	0.826

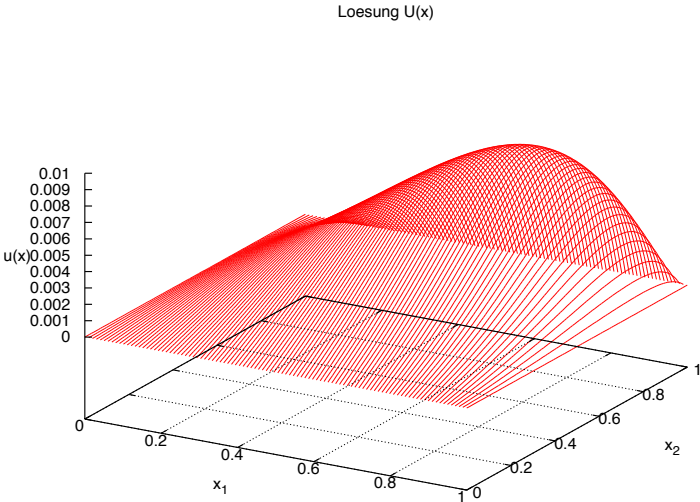


Abbildung 6.20.: Lösung der Poisson-Gleichung

7. Zusammenfassung und Ausblick

Resultate dieser Arbeit

In dieser Arbeit haben wir das Verfahren der Proper Generalized Decomposition am Beispiel der Wärmeleitungsgleichung hergeleitet. Wir stellten zwei verschiedene Varianten des Verfahrens vor, die wir im Weiteren genauer betrachtet haben. Wir haben die für die einzelnen Fälle notwendigen Abbruchkriterien hergeleitet und später auch eine Methode für deren effiziente Berechnung vorgestellt.

Im weiteren Verlauf wurde das Verfahren der PGD durch eine Tensornotation verallgemeinert. Dies ermöglicht eine wesentlich abstraktere Betrachtung des Verfahrens und gibt die Motivation, es auch auf andere Differentialoperatoren anzuwenden.

Im darauffolgenden Kapitel haben wir die Eigenschaften des Verfahrens untersucht. So haben wir zwei verschiedene Arten eines Fehlerschätzers für die Anreicherung der Lösung vorgestellt und sind auf die Fehlerentwicklung der niederdimensionalen Probleme eingegangen.

Im Anschluss daran wurden die Laufzeiten für die zwei vorgestellten Varianten des Algorithmus hergeleitet. Hier ist besonders zu erwähnen, dass unter der Annahme die rechte Seite f liegt in Produktstruktur vor, das Verfahren eine lediglich lineare Zunahme des Aufwands bezüglich der Dimension aufweist.

Wir haben außerdem gesehen wie einige Probleme bei der praktischen Umsetzung des Verfahrens gelöst werden können um eine effiziente Implementierung zu ermöglichen. Hier haben wir auch das Verfahren in Pseudocode zusammengefasst.

Schließlich haben wir das Verfahren in der Anwendung untersucht. Es hat sich gezeigt, dass die PGD ein Verfahren mit enormem Potential für die Lösung von hochdimensionalen Problemen darstellt. Allerdings bleibt die Problematik des Fluchs der Dimensionen insofern bestehen, dass eine mehrdimensionale Integration durchgeführt werden muss, sofern die rechte Seite f nicht in Produktstruktur vorliegt.

Ausblick

Das vorgestellte Verfahren der Proper Generalized Decomposition wurde in dieser Arbeit auf die Wärmeleitungsgleichung angewendet. Es kann aber auch zur Lösung diverser anderer Differentialoperatoren eingesetzt werden und eignet sich dabei vor allem für hochdimensionale Probleme. Da diese immer häufiger auftreten sind die weiteren Anwendungsmöglichkeiten sehr zahlreich. Dazu gehören etwa parametrische Modelle, in denen Parameter als zusätzliche Koordinaten betrachtet werden. So kann das Modell einmal berechnet werden und später muss nur noch eine kleine Nachberechnung für den gewünschten Parameter durchgeführt werden. In diesem Zusammenhang werden auch Echtzeit-Simulationen von Modellen die als zu Komplex galten wieder interessant. Ein Verfahren wie die PGD, das die Möglichkeit bietet die Dimension nur linear in die Komplexität eingehen zu lassen eröffnet hier sowohl neue Möglichkeiten in der Anwendung, als auch die erneute Betrachtung von bereits bestehenden Anwendungen.

Ein weiterer Anknüpfungspunkt ist eine Verbesserung des eigentlich Verfahrens etwa hinsichtlich

Adaptivität, sodass im Alternating Directions Algorithmus nur diejenigen Komponenten geupdated werden, für die es auch notwendig ist. Dies ist unter anderem Gegenstand der aktuellen Forschung.

A. Appendix

A.1. Satz von Fubini

Satz A.1. Fubini

Für $f \in \mathcal{L}_\infty(\mathbb{R}^{m+n})$ gilt:

i) $f(x, \bullet) \in L_1(\mathbb{R}^n)$ für \mathcal{L}_m -f.a. $x \in \mathbb{R}^m$,
 $f(\bullet, y) \in L_1(\mathbb{R}^m)$ für \mathcal{L}_n -f.a. $y \in \mathbb{R}^n$;

ii) $x \mapsto \int_{\mathbb{R}^n} f(x, y) dy$ ist \mathcal{L}_m -integrierbar,
 $y \mapsto \int_{\mathbb{R}^m} f(x, y) dx$ ist \mathcal{L}_n -integrierbar;

iii) $\int_{\mathbb{R}^{m+n}} f d(x, y) = \int_{\mathbb{R}^m} [\int_{\mathbb{R}^n} f(x, y) dy] dx = \int_{\mathbb{R}^n} [\int_{\mathbb{R}^m} f(x, y) dx] dy$.

Beweis: Ein Beweis des Satzes findet sich in vielen Standardwerken der Analysis, wie [8].

A.2. Green'sche Formel

Satz A.2. Green'sche Formel

Es sei $\Omega \subset \mathbb{R}^D$ ein beschränktes Lipschitz-Gebiet und es sei $u \in C^2(\Omega) \cap C^1(\bar{\Omega})$. Ist $v \in C^1(\bar{\Omega})$, so gilt die **erste Green'sche Formel**

$$\int_{\Omega} v \Delta u \, dx = - \int_{\Omega} \nabla v \cdot \nabla u \, dx + \int_{\partial \Omega} v \partial_{\nu} u \, ds_x.$$

Ist $v \in C^2(\Omega) \cap C^1(\bar{\Omega})$, so gilt die **zweite Green'sche Formel**

$$\int_{\Omega} v \Delta u - u \Delta v \, dx = \int_{\partial \Omega} v \partial_{\nu} u - u \partial_{\nu} v \, ds_x.$$

Dabei beschreibt $\partial_{\nu} u$ die Ableitung in Richtung der Normalen.

Beweis: Ein Beweis dieses Satzes findet sich ebenfalls in vielen Standardwerken der Analysis, wie [1].

Literaturverzeichnis

- [1] *H. Amann and J. Escher. Analysis 3. Birkhäuser, 2 edition, 2001.*
- [2] *A. Ammar, F. Chinesta, P. Diez, and A. Huerta. An error estimator for separated representations of highly multidimensional models. Computer Methods in Applied Mechanics and Engineering, 2010.*
- [3] *A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory a new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Preprint, 2006.*
- [4] *D. Braess. Finite Elemente Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie. Springer, 4. edition, 2007.*
- [5] *F. Chinesta, A. Ammar, and E. Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. Preprint, 2009.*
- [6] *P. Deuffhard and F. Bornemann. Numerische Mathematik 2 Gewöhnliche Differentialgleichungen. de Gruyter, 3. edition, 2001.*
- [7] *A. Dumon, C. Allery, and A. Ammar. Proper general decomposition (pgd) for the resolution of navier–stokes equations. Journal of Computational Physics, 2010.*
- [8] *J. Elstrodt. Maß- und Integrationstheorie. Springer, 6 edition, 2009.*
- [9] *Lawrence C. Evans. Partial Differential Equations, volume 19 of Graduate studies in mathematics. American Mathematical Society, 1 edition, 1998.*
- [10] *A. Falcó, L. Hilario, N. Montés, and M. C. Mora. Numerical strategies for the galerkin-proper generalized decomposition method. Preprint, 2011.*
- [11] *A. Falcó and A. Nouy. A proper generalized decomposition for the solution of elliptic problems in abstract form by using a functional eckart-young approach. Preprint, 2010.*
- [12] *A. Falcó and A. Nouy. Proper generalized decomposition for nonlinear convex problems in tensor banach spaces. Preprint, 2011.*
- [13] *T. Hahn. Cuba - a library for multidimensional numerical integration. Computer Physics Communications, 168(2):78–95, 2005.*
- [14] *A. Nouy. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Comput. Methods Appl. Mech. Engrg., 2010.*
- [15] *E. Pruliere, F. Chinesta, and A. Ammar. On the deterministic solution of multidimensional parametric models using the proper generalized decomposition. Mathematics and Computers in Simulation, 2010.*

