

Institut für Parallele und Verteilte Systeme
Abteilung Simulation großer Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Studienarbeit Nr. 2445

Fehlertolerante Lösung von PDEs mit der Dünngitter-Kombinationstechnik

Amir Abdelaziz

Studiengang:	Informatik
Prüfer/in:	Jun.-Prof. Dr. rer. nat. Dirk Pflüger
Betreuer/in:	M. Sc. Mario Heene

Beginn am: 2. Dezember 2013

Beendet am: 3. Juni 2014

CR-Nummer: G.1.8

Kurzfassung

Diese Studienarbeit beschäftigt sich mit der fehlertoleranten Lösung von partiellen Differentialgleichungen (PDE). Untersucht werden Verfahren die auf der Dünngitter-Kombinationstechnik beruhen. Anhand von zwei PDEs, dem Laplace Problem und der Advektionsgleichung wird untersucht wie sich ein Fehlerausfall einer Teillösung auf den Fehler der Kombinationstechnik auswirkt. Des Weiteren werden Ansätze wie die Neuberechnung von kleineren Teillösungen, die Richardson-Extrapolation und deren Verknüpfung mit der Kombinationstechnik untersucht. Die dazu nötigen Methoden werden in das Dünngitter-Framework SG++ implementiert.

Inhaltsverzeichnis

1. Einleitung	9
1.1. Motivation	9
1.2. Aufgabenstellung	10
2. Definitionen, Versuchsaufbau und Ergebnisse	13
2.1. Gitter, Funktionsräume und Dünngitter-Kombinationstechnik	13
2.2. Truncated Combination Technique	17
2.3. Verfahren für das Berechnen einer Kombination im Fehlerfall	21
2.4. Beschreibung der PDEs und der Testprogramme	26
2.5. Ergebnisse der Kombinationstechnik	27
2.6. Richardson-Extrapolation	31
2.7. Verknüpfung von einfacher Kombinationstechnik und Richardson-Extrapolation . . .	35
3. Implementierung fehlertoleranter Verfahren in SG++ und Ergebnisse	39
3.1. Ansätze zur Verbesserung der Verfahren	47
4. Zusammenfassung	51
A. Anhang	53
A.1. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2}^2$	53
A.2. Kombinationen nach einem Fehler in $\tilde{f}_{n-3,n-3}^3$	53
A.3. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2}^2$ mit Richardson Extrapolation p=2 . .	55
A.4. Kombinationen nach einem Fehler in $\tilde{f}_{n-3,n-3}^3$ mit Richardson Extrapolation p=2 . .	55
A.5. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2}^2$ mit Richardson Extrapolation und verknüpfter Technik p=1	56
A.6. Kombinationen nach einem Fehler in $\tilde{f}_{n-3,n-3}^3$ mit Richardson Extrapolation und verknüpfter Technik p=1	57
A.7. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2,n-2}^2$	57
A.8. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2,n-2}^2$ mit Richardson Extrapolation p=2	59
A.9. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2,n-2}^2$ mit Richardson Extrapolation p=1	60
Literaturverzeichnis	61

Abbildungsverzeichnis

2.1.	Gitter	14
2.2.	Dünngitter-Kombinationstechnik	16
2.3.	Truncated Combination Technique $\tilde{f}_{n-t,n-t}^t$ und Fehlerfälle	19
2.4.	Inklusion Exklusion	23
2.5.	Programme für die Untersuchungen der Fehler	28
2.6.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit $\tilde{f}_{n-2,n-2}^2$	29
2.7.	Ergebnisplot: Fehler bei der Lösung der Advektionsgleichung mit $\tilde{f}_{n-2,n-2}^2$	29
2.8.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit der Richardson- Extrapolation auf $V_{n-3,n-3}$	32
2.9.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit der Richardson- Extrapolation auf $V_{n,n}$	33
2.10.	Ergebnisplot: Fehler bei der Lösung der Advektionsgleichung mit Richardson- Extrapolation und ihrer Verknüpfung mit der Kombinationstechnik, $\tilde{f}_{n-2,n-2}^2$	36
2.11.	Ergebnisplot: Fehler bei der Lösung der Advektionsgleichung mit Richardson- Extrapolation und ihrer Verknüpfung mit der Kombinationstechnik, $\tilde{f}_{n-3,n-3}^3$	37
3.1.	CombiFaultRecoveryScheme	40
3.2.	Ergebnisplot: Laplace Problem 2D faultRecoveryFast	42
3.3.	Ergebnisplot: Laplace Problem 2D faultRecoverySlow mit Neuberechnung	44
3.4.	Ergebnisplot: Laplace Problem 2D faultRecoverySlow ohne Neuberechnung	45
3.5.	Ergebnisplot: Laplace Problem 2D faultRecoveryRichardson	45
3.6.	Ergebnisplot: Laplace Problem 2D faultRecoveryBlock keine Zentrierung	47
3.7.	Ergebnisplot: Laplace Problem 2D faultRecoveryBlock mit Zentrierung	48
A.1.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit $\tilde{f}_{n-3,n-3}^3$	54
A.2.	Ergebnisplot: Fehler bei der Lösung der Advektionsgleichung mit $\tilde{f}_{n-3,n-3}^3$	54
A.3.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit der Richardson- Extrapolation auf $V_{n-2,n-2}$	55
A.4.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit der Richardson- Extrapolation auf $V_{n,n}$	56
A.5.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit $\tilde{f}_{n-2,n-2,n-2}^2$	58
A.6.	Ergebnisplot: Fehler bei der Lösung der Advektionsgleichung mit $\tilde{f}_{n-2,n-2,n-2}^2$	58
A.7.	Ergebnisplot: Fehler bei der Lösung der Laplace-Gleichung mit der Richardson- Extrapolation auf $V_{n-2,n-2,n-2}$	59
A.8.	Ergebnisplot: Fehler bei der Lösung der Advektionsgleichung mit der Richardson- Extrapolation auf $V_{n-2,n-2,n-2}$	60

Verzeichnis der Algorithmen

- 3.1. Algorithmus zur Messung der Fehler der implementierten Methoden 41
- 3.2. Bestimmung einer Richardson-Extrapolation nach einem Ausfall von Teillösungen . . 46

1. Einleitung

1.1. Motivation

Computergestützte Simulationen spielen in den Ingenieurs- und Naturwissenschaften eine wichtige Rolle. Simulationen ersetzen oder ergänzen klassische Experimente dort, wo sie nicht durchführbar, zu gefährlich oder zu teuer sind. Anwendungen finden sie z. B. bei der Untersuchung von Erdbeben, in der Astronomie, in der Medizin zur Erforschung von Proteinen oder in der Strömungssimulation. Für den Rechenaufwand spielen die konkrete Modellierung, sowie die räumliche und zeitliche Diskretisierung eine Rolle. Um auch feinskalige oder hochdimensionale Probleme zu lösen ist ein enormer Rechenaufwand nötig. [BZBP13]

So ist seit längerem eine steigende Nachfrage nach immer höherer Rechenleistung zu beobachten. Die Top500 [Top] Liste, ist ein im halbjährlichen Rhythmus aufgestelltes Ranking der schnellsten Rechner im High Performance Computing. Ihr lässt sich ein exponentielles Wachstum der Rechenleistung entnehmen. Stand in der ersten Liste von 1993 der CM-5 mit 1024 Cores und einer gemessenen Rechenleistung von ca. 59.7 GFlop/s an der Spitze, so sind es bei der Nr. 1 in der aktuellen Liste vom November 2013, dem Tianhe-2 (MilkyWay-2) - TH-IVB-FEP, rund 33,87 PFlop/s und über drei Millionen Cores.

Durch die ständige Zunahme der Komplexität der Systeme erhöht sich die Wahrscheinlichkeit, dass irgendeine der vielen Komponenten ausfällt. Einen guten Überblick über Untersuchungen zur Fehlerhäufigkeit, deren Ursachen und den Techniken fehlertoleranter Systeme liefern Egwutuoha, Levy, Selic und Chen [ELSC13]. Häufige Fehlerquellen seien neben Hardwarefehlern vor allem Softwarefehler und Benutzerinteraktionen, wie z. B. der Austausch von Hardware.

Des Weiteren stellt sich die Frage wie Fehler überhaupt erkannt werden können. Diese Arbeit wird sich nicht mit diesen Fragen beschäftigen, genauso wenig wie mit den Themen redundanter Hardware oder den verschiedenen Rollback-Recovery Verfahren. Hier seien an dieser Stelle nur kurz die Checkpoint-Restart Verfahren erwähnt zu denen sich ein Einstieg in [ELSC13] findet.

Checkpoint-Restart Verfahren basieren darauf, dass in regelmäßigen Zeitabständen der Zustand eines parallelen Programms gesichert wird. Werden Fehler diagnostiziert, werden einzelne Prozesse oder die ganze Anwendung angehalten und eine ältere Sicherung wieder eingespielt, von der aus die Berechnung weiter erfolgt. Damit diese Verfahren anwendbar sind, muss der Checkpoint die Sicherung eines fehlerfreien Zustandes sein, also insbesondere vor Auftreten von Fehlern erfolgen.

Die „Mean Time To Failure“ (MTTF) ist die durchschnittliche Zeit bis zum Auftreten des ersten Fehlers. Bei Systemen die repariert werden spricht man auch von der „Mean Time Between Failures“ (MTBF) also dem Erwartungswert der Zeit zwischen zwei aufeinander folgenden Fehlern. Die Fehlerrate ist der

1. Einleitung

Kehrwert des MTBF. Untersuchungen nach [BS06] zeigen, dass die Fehlerrate ungefähr proportional mit der Anzahl der Prozessoren steigt.

Dies bestätigt auch folgende Überlegung:

Unter der Annahme ein Fehler einer Komponente würde zu einem ganzen Systemausfall führen und alle MTBF hätten eine obere Schranke $c \in \mathbb{R}^+$ gilt folgendes:

Seien $X_1, \dots, X_n, n \in \mathbb{N}$ die Komponenten eines Systems und $M_1, \dots, M_n, n \in \mathbb{N}$ die zugehörigen MTBF Werte. Für alle $i \in \{1, \dots, n\}$ gelte $M_i \leq c \in \mathbb{R}^+$. Seien $\lambda_i = \frac{1}{M_i}, i \in \{1, \dots, n\}$ die zugehörigen Fehlerraten. Dann gilt für das ganze System

$$M = \frac{1}{\lambda} = \frac{1}{\sum_{i=1}^n \lambda_i} = \frac{1}{\sum_{i=1}^n \frac{1}{M_i}} \leq \frac{1}{n \frac{1}{c}} = \frac{c}{n}$$

Capello, Geist, Gropp, Kale, Kramer und Snir prognostizieren, dass mit den zukünftigen Exascale Systemen, Checkpoint-Restart Verfahren nicht mehr funktionieren werden, da die Zeit für Sicherung und Wiederherstellung die Mean Time To Failure des Gesamtsystems übersteigen wird [GKKS09].

Folglich müssen neue Verfahren gefunden werden, welche die effiziente Nutzung zukünftiger Systeme ermöglichen.

1.2. Aufgabenstellung

Diese Studienarbeit beschäftigt sich mit der fehlertoleranten Lösung von partiellen Differentialgleichungen (PDE) mittels der Dünngitter-Kombinationstechnik [GSZ92].

Bei der numerischen Lösung von PDEs wie z. B. bei den Finite Elemente Methoden wird der Definitionsbereich D der gesuchten Lösung f durch ein Gitter diskretisiert. Die Approximation \tilde{f} ist dann z. B. ein stückweise polynomieller und numerisch berechneter Interpolant von f , mit den Gitterpunkten als Stützstellen.

Erfüllt f bestimmte Anforderungen, wie z. B. beschränkte zweite gemischte Ableitungen, dann konvergiert die Approximation \tilde{f} , bei feiner werdender Diskretisierung, bezüglich der \mathcal{L}^p Normen, gegen f . Für einen d -dimensionalen Definitionsbereich D und $0 < p \leq \infty$ ist die \mathcal{L}^p Norm definiert durch:

$$\|f\|_{\mathcal{L}^p}^p := \begin{cases} \int_D |f(\vec{x})|^p d\vec{x} & \text{für } 0 < p < \infty \\ \max_{\vec{x} \in D} |f(\vec{x})| & \text{für } p = \infty \end{cases}$$

Werden isotrope volle Gitter verwendet, so liegt der Berechnungsaufwand, falls f genügend glatt ist, $\|f - \tilde{f}\|_{\mathcal{L}^2}$ und $\|f - \tilde{f}\|_{\mathcal{L}^\infty}$ unter eine Schranke ϵ zu bekommen für $\alpha \in \mathbb{R}^+$ in $\mathcal{O}(\epsilon^{-\alpha d})$ [BG04]. Die Komplexität steigt also exponentiell mit der Anzahl der Dimensionen des Problems. Dies nennt man auch den Fluch der Dimensionalität [Bel61].

Ist f genügend glatt, so können sogenannte Dünne Gitter [Zen91], mit einem Aufwand von $\mathcal{O}(\epsilon^{-\alpha} * |\log_2 \epsilon|^{\beta * (d-1)})$, für $\alpha, \beta \in \mathbb{R}^+$ verwendet werden. Dabei hängen α und β von dem Polynomgrad der d -dimensionalen Basisfunktionen ab, näheres dazu kann in [BG04] nachgelesen werden, d. h. der Aufwand wächst also in Abhängigkeit von der Dimension wesentlich langsamer.

Dünne Gitter sind allerdings, aufgrund ihrer hierarchischen und rekursiven Struktur, algorithmisch schwerer handzuhaben als anisotrope volle Gitter. Bei der Dünngitter-Kombinationstechnik wird eine Linearkombination der Approximationslösungen $f_{\vec{l}}, \vec{l} \in L$, aus den zugehörigen Funktionsräumen $V_{\vec{l}}, \vec{l} \in L$ verwendet um die Dünngitter-Lösung zu approximieren¹. Diese hat ein leicht schlechteres Konvergenzverhalten als das der dünnen Gitter, aber es können zumindest Standardalgorithmen verwendet werden. [Pfl10]

Ein weiterer Vorteil der Dünngitter-Kombinationstechnik ist die einfachere Parallelisierung. So können die Lösungen auf den anisotropen vollen Gittern unabhängig voneinander berechnet werden, was den Kommunikations- und Synchronisationsaufwand erheblich verringert. Fällt eine Teillösung aus, so können erfolgreich berechnete Teillösungen weiter verwendet werden. Um dies zu gewährleisten können etwa neue Koeffizienten für die Linearkombination berechnet oder kleinere Teillösungen interpoliert oder nachberechnet werden.

Zur Analyse der Fehler beschränkt sich diese Arbeit in Kapitel 2 erst einmal auf die „Truncated Combination Technique“ wie sie z. B. in [HH14] [BBNS12] und [BP12] verwendet werden.

Anhand von zwei PDEs, dem Laplace Problem und der Advektionsgleichung wird untersucht wie sich ein Fehlerausfall einer Teillösung auf den Fehler der „Truncated Combination Technique“ auswirkt. Des Weiteren werden Ansätze wie die Neuberechnung von kleineren Teillösungen, die Richardson-Extrapolation und deren Verknüpfung mit der Kombinationstechnik untersucht. Die dazu nötigen Methoden werden in das Dünngitter-Framework SG++ [Pfl10] implementiert, welches im Rahmen einer Dissertation von Dirk Pflüger, entstanden ist und in der Abteilung Simulation großer Systeme, des Institutes für Parallele und Verteilte Systeme, an der Universität Stuttgart weiterentwickelt wird.

Grundlage der Studienarbeit ist der Artikel „Robust Solutions to PDEs with multiple grids“ von Brendan Harding und Markus Hegland [HH14].

¹Dies ist nur bei bestimmten Problemen möglich. Dünngitter-Kombinationstechnik und Dünnes Gitter sind a priori nicht gleich [HGC07]. Insbesondere können für ein Problem die Dünne Gitter gegen die Lösung konvergieren, während die Dünngitter-Kombinationstechnik nicht konvergiert. Für Beispiele dazu siehe [Gar04].

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Definitionen, Versuchsaufbau und Ergebnisse: Hier werden werden, die in der Arbeit verwendeten Begriffe wie die Dünngitter-Kombinationstechnik oder die Truncated Combination Technique definiert. Es werden Fehlerabschätzungen betrachtet und das Konzept der multivariaten Richardson-Extrapolation und ihrer Verknüpfung mit der Kombinationstechnik erläutert. Es werden Verfahren zur Berechnung einer neuen Linearkombination, falls Teillösungen ausfallen, vorgestellt und die Simulationsergebnisse für den Fehler verschiedener Linearkombinationen betrachtet.

Kapitel 3 – Implementierung fehlertoleranter Verfahren in SG++ und Ergebnisse: Hier wird die Implementierung von verschiedenen Verfahren zur Berechnung einer neuen Linearkombination, falls Teillösungen ausfallen, skizziert und die Simulationsergebnisse für deren Fehler betrachtet. Am Ende des Kapitels wird noch ein Ausblick gegeben.

Kapitel 4 – Zusammenfassung fasst die Ergebnisse der Arbeit zusammen.

2. Definitionen, Versuchsaufbau und Ergebnisse

Fast alle Definitionen in diesem Kapitel sind [HH14], [HH12] und [Pfl10] entnommen und die Notation ist weitgehend ähnlich. Hier sei gleich ein Problem benannt: Wir werden ein paar Notationen verwenden, welche mathematisch nicht ganz korrekt sind. So sprechen wir z. B. von Levelvektoren und Normen auf Levelvektoren ohne einen Vektorraum zu definieren und verwenden die dafür übliche Notation für Vektoren und Normen.

2.1. Gitter, Funktionsräume und Dünngitter-Kombinationstechnik

Definition 2.1.1 (Levelvektoren, Normen, Ordnungsrelation) Ein Levelvektor \vec{l} sei ein Element aus \mathbb{N}_0^d und für alle $n \in \mathbb{N}_0$ definieren wir $\vec{n} \in \mathbb{N}_0^d$ durch

$$\vec{n} := (n, n, \dots, n)$$

Für einen Levelvektor $\vec{l} = (l_1, \dots, l_d) \in \mathbb{N}_0^d$ definieren wir die 1-Norm und die Maximumsnorm durch

$$\|\vec{l}\|_1 := \sum_{k=1}^d l_k$$

$$\|\vec{l}\|_\infty := \max\{l_1, \dots, l_d\}$$

wobei die 1-Norm auch Levelsumme genannt wird. Des Weiteren definieren wir eine Ordnungsrelation auf Levelvektoren durch

$$\forall \vec{l}, \vec{\lambda} \in \mathbb{N}_0^d : \left(\vec{l} \leq \vec{\lambda} \Leftrightarrow \forall k \in \{1, \dots, d\} : l_k \leq \lambda_k \right)$$

Die Addition ist die komponentenweise Addition und die Subtraktion wird definiert durch

$$\vec{l} - \vec{\lambda} := (\max\{l_1 - \lambda_1, 0\}, \dots, \max\{l_d - \lambda_d, 0\})$$

und eine zweistellige Verknüpfung \wedge durch

$$\forall \vec{l}, \vec{\lambda} \in \mathbb{N}_0^d : \left(\vec{l} \wedge \vec{\lambda} := (\min\{l_1, \lambda_1\}, \dots, \min\{l_d, \lambda_d\}) \right)$$

Des Weiteren definieren wir für $l, \lambda \in \mathbb{N}_0$

$$l \wedge \lambda := (l) \wedge (\lambda)$$

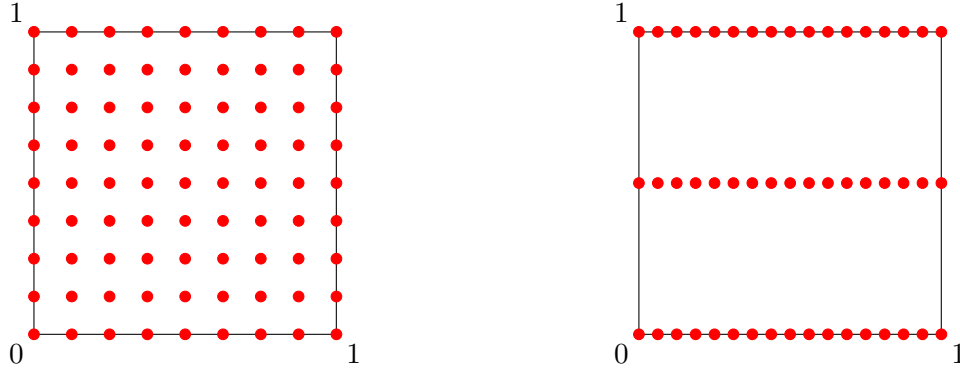


Abbildung 2.1.: Ein isotropes volles Gitter (links), ein anisotropes volles Gitter (rechts)

Definition 2.1.2 (Volles Gitter) Sei $\vec{l} = (l_1, \dots, l_d) \in \mathbb{N}_0^d$ ein Levelvektor und für beliebiges $l \in \mathbb{N}_0$ bezeichne $\Omega_l = \{kh_l \mid k \in \{0, 1, 2, 3, \dots, 2^l\}\}$ eine Diskretisierung des Einheitsintervalls mit $h_l := 2^{-l}$. Dann definieren wir ein Gitter auf dem d -dimensionalen Einheitswürfel $[0, 1]^d$ durch:

$$\Omega_{\vec{l}} := \Omega_{l_1} \times \Omega_{l_2} \times \dots \times \Omega_{l_d}$$

Sei $l_1 = l_2 = \dots = l_d = n \in \mathbb{N}_0$, dann heißt $\Omega_{\vec{l}}$ isotrop. Andernfalls heißt $\Omega_{\vec{l}}$ anisotrop.

Seien $\vec{l}, \vec{\lambda} \in \mathbb{N}_0^d$ mit $\Omega_{\vec{l}} \subseteq \Omega_{\vec{\lambda}}$ gegeben, dann sprechen wir auch davon, dass $\Omega_{\vec{l}}$ „größer“ als $\Omega_{\vec{\lambda}}$ und $\Omega_{\vec{\lambda}}$ „feiner“ als $\Omega_{\vec{l}}$ sei.

Definition 2.1.3 (Raum stückweiser linearer Funktionen) Passend zum Gitter $\Omega_{\vec{l}}$ definieren wir den zugehörigen Raum der stückweise linearen Funktionen auf $[0, 1]^d$ als:

$$V_{\vec{l}} := \text{span} \left\{ \phi_{\vec{l}, \vec{i}} : i_t = 0, \dots, 2^{l_t}, t = 1, \dots, d \right\}$$

wobei $\phi_{\vec{l}, \vec{i}}$ die üblichen d -linearen Basisfunktionen (Hut-Funktionen) sind. Sei $\vec{x} \in [0, 1]^d$ so definieren wir

$$\phi_{\vec{l}, \vec{i}}(\vec{x}) := \prod_{k=1}^d \phi_{l_k, i_k}(x_k)$$

mit

$$\phi_{l, i}(x) := \max \left\{ 1 - |2^l x - i|, 0 \right\}$$

Seien $\vec{l}, \vec{\lambda} \in \mathbb{N}_0^d$ mit $\Omega_{\vec{l}} \subseteq \Omega_{\vec{\lambda}}$, dann sprechen wir auch davon, dass $V_{\vec{l}}$ „größer“ als $V_{\vec{\lambda}}$ und $V_{\vec{\lambda}}$ „feiner“ als $V_{\vec{l}}$ sei.

Theorem 2.1.4 Offensichtlich gilt

- i) $\forall l, \lambda \in \mathbb{N}_0 : (l \leq \lambda \Leftrightarrow \Omega_l \subseteq \Omega_\lambda)$
- ii) $\forall \vec{l}, \vec{\lambda} \in \mathbb{N}_0^d : (\vec{l} \leq \vec{\lambda} \Leftrightarrow \Omega_{\vec{l}} \subseteq \Omega_{\vec{\lambda}})$

$$\text{iii) } \forall l, \lambda \in \mathbb{N}_0 : \Omega_l \cap \Omega_\lambda = \Omega_{l \wedge \lambda}$$

$$\text{iv) } \forall \vec{l}, \vec{\lambda} \in \mathbb{N}_0^d : \Omega_{\vec{l}} \cap \Omega_{\vec{\lambda}} = \Omega_{\vec{l} \wedge \vec{\lambda}}$$

Beweis i) „ \Rightarrow “ Sei $l \leq \lambda$, dann gilt $2^{\lambda-l} \in \mathbb{N}$. Ist $k \in \mathbb{N}_0, k \leq 2^l$ dann folgt daraus, dass $k * 2^{\lambda-l} \in \{0, 1, \dots, 2^\lambda\}$ ist. Nun gilt

$$\begin{aligned} x \in \Omega_l &\Leftrightarrow \exists k \in \{0, \dots, 2^l\} : x = k * 2^{-l} = \underbrace{k * 2^{\lambda-l}}_{=u \in \{0, 1, \dots, 2^\lambda\}} * 2^{-\lambda} \\ &\Rightarrow \exists u \in \{0, \dots, 2^\lambda\} : x = u * 2^{-\lambda} \Leftrightarrow x \in \Omega_\lambda \end{aligned}$$

und somit $\Omega_l \subseteq \Omega_\lambda$.

„ \Leftarrow “ Sei $l > \lambda$. Wähle nun etwa $x \in \Omega_l$ mit $x = 2^{-l} = 2^{\lambda-l} 2^{-\lambda}$. Aus $l > \lambda$ folgt $2^{\lambda-l} \notin \mathbb{N}$ und damit auch $2^{\lambda-l} \notin \{0, \dots, 2^\lambda\}$. Nun gilt $x \notin \Omega_\lambda$ und somit $\Omega_l \not\subseteq \Omega_\lambda$.

$$\text{ii) Seien } \vec{l}, \vec{\lambda} \in \mathbb{N}_0^d. \text{ Dann gilt: } \vec{l} \leq \vec{\lambda} \stackrel{2.1.1}{\Leftrightarrow} \forall k \in \{1, \dots, d\} : l_k \leq \lambda_k \stackrel{i)}{\Leftrightarrow} \forall k \in \{1, \dots, d\} : \Omega_{l_k} \subseteq \Omega_{\lambda_k} \Leftrightarrow \prod_{k=1}^d \Omega_{l_k} \subseteq \prod_{k=1}^d \Omega_{\lambda_k} \stackrel{2.1.2}{\Leftrightarrow} \Omega_{\vec{l}} \subseteq \Omega_{\vec{\lambda}}$$

iii) Sei o.B.d.A. $l \leq \lambda$ dann gilt nach i) $\Omega_l \subseteq \Omega_\lambda$ und damit $\Omega_l \cap \Omega_\lambda = \Omega_l$. Des Weiteren gilt $l \wedge \lambda = \min\{l, \lambda\} = l$ und damit $\Omega_{l \wedge \lambda} = \Omega_l = \Omega_l \cap \Omega_\lambda$

iv) Sei

$$\vec{l} \wedge \vec{\lambda} = \vec{k} = (k_1, \dots, k_d) = (\min\{l_1, \lambda_1\}, \dots, \min\{l_d, \lambda_d\})$$

Dann gilt nach iii)

$$\forall u \in \{1, \dots, d\} : \Omega_{k_u} = \Omega_{l_u} \cap \Omega_{\lambda_u}$$

und somit für jedes $\vec{x} = (x_1, \dots, x_d) \in \Omega_{\vec{k}}$

$$\begin{aligned} \vec{x} = (x_1, \dots, x_d) \in \Omega_{\vec{k}} &\Leftrightarrow \forall u \in \{1, \dots, d\} \quad x_u \in \Omega_{k_u} \Leftrightarrow \forall u \in \{1, \dots, d\} \quad x_u \in \Omega_{l_u} \cap \Omega_{\lambda_u} \\ &\Leftrightarrow \forall u \in \{1, \dots, d\} \quad (x_u \in \Omega_{l_u} \wedge x_u \in \Omega_{\lambda_u}) \Leftrightarrow \forall u \in \{1, \dots, d\} \quad x_u \in \Omega_{l_u} \wedge \forall u \in \{1, \dots, d\} \quad x_u \in \Omega_{\lambda_u} \\ &\Leftrightarrow \vec{x} \in \Omega_{\vec{l}} \wedge \vec{x} \in \Omega_{\vec{\lambda}} \Leftrightarrow \vec{x} \in \Omega_{\vec{l}} \cap \Omega_{\vec{\lambda}} \end{aligned}$$

und damit $\Omega_{\vec{l}} \cap \Omega_{\vec{\lambda}} = \Omega_{\vec{l} \wedge \vec{\lambda}}$.

Für die Funktionsräume $V_{\vec{l}}$ lassen sich analoge Aussagen zeigen.

Definition 2.1.5 (hierarchische Inkremente) Die hierarchischen Inkremente $W_{\vec{l}}$ sind gegeben durch:

$$W_{\vec{0}} = V_{\vec{0}} \quad \forall \vec{l} \neq \vec{0} \quad V_{\vec{l}} = W_{\vec{l}} \oplus \sum_{t=1}^d V_{\vec{l} - \vec{e}_t}$$

wobei \vec{e}_t der Einheitsvektor entlang der t -ten Achse ist, sodass also $\vec{e}_t \in \mathbb{N}_0^d \wedge \|\vec{e}_t\|_1 = 1 \wedge e_{t_t} = 1$ gilt. Nun gilt:

$$V_n = \bigoplus_{l_1=0}^n \cdots \bigoplus_{l_d=0}^n W_{\vec{l}} = \bigoplus_{\|\vec{l}\|_\infty \leq n} W_{\vec{l}}$$

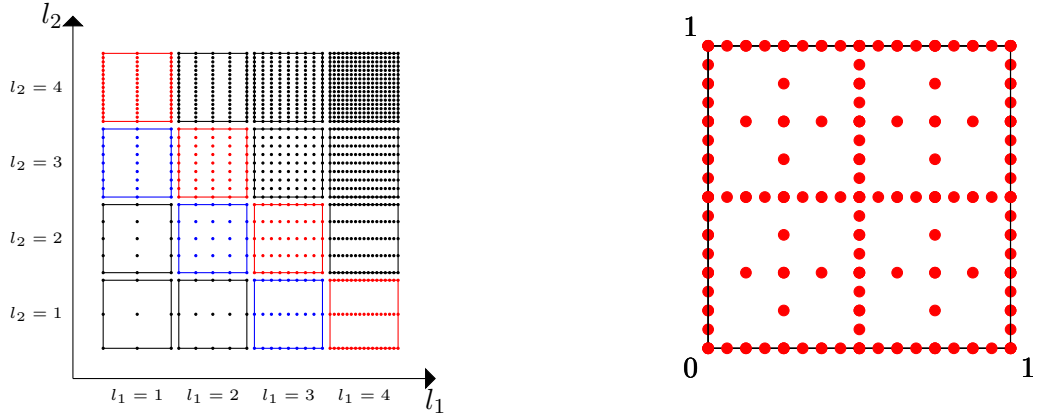


Abbildung 2.2.: Die Dünngitter-Kombinationstechnik in 2D für $n = 4$. Die einzelnen Teilgitter (links) und die Kombination (rechts). Die auf den rot dargestellten Gittern berechneten Approximationslösungen werden addiert (Koeffizient +1), die auf den blau dargestellten Gittern berechneten Approximationslösungen werden subtrahiert (Koeffizient -1). Die Menge der Gitterpunkte in der Kombination ist die Vereinigung der Gitterpunkte der einzelnen anisotropen vollen Gitter. Die Koeffizienten der Kombinationstechnik können aus der Vereinigung der roten Gitter nach dem Prinzip von Inklusion und Exklusion gewonnen werden. Die Summe der Koeffizienten der Dünngitter-Kombinationstechnik ist 1.

Definition 2.1.6 (Dünngitter-Kombinationstechnik) Für alle $\vec{l} \in \mathbb{N}_0^d$ bezeichne $f_{\vec{l}}$ die Approximationslösung der exakten Lösung f in $V_{\vec{l}}$. Sei d die Dimension des Definitionsbereiches von f . Sei $n \in \mathbb{N}$, dann heißt

$$f_n^c(\vec{x}) := \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\substack{\|\vec{l}\|_1 = n+d-1-q \\ \vec{l} \leq \vec{l}}} f_{\vec{l}}(\vec{x})$$

die Dünngitter-Kombinationstechnik. Mit zugehörigem Dünngitter-Funktionsraum:

$$V_n^s := \sum_{\substack{\|\vec{l}\|_1 = n+d-1 \\ \vec{l} \leq \vec{l}}} V_{\vec{l}} = \bigoplus_{\substack{\|\vec{l}\|_1 \leq n+d-1 \\ \|\vec{l}\|_{\infty} \leq n}} W_{\vec{l}}$$

Definition 2.1.7 (Verallgemeinerte Kombinationstechnik) Die Dünngitterkombinationstechnik lässt sich auch verallgemeinern zu

$$f_L(\vec{x}) = \sum_{\vec{l} \in L} c_{\vec{l}} f_{\vec{l}}(\vec{x})$$

wobei $L \subseteq \mathbb{N}_0^d$ gilt und $c_{\vec{l}} \in \mathbb{R}$, $\vec{l} \in L$ die Koeffizienten der Kombinationstechnik sind. Wir definieren das zugehörige Gitter durch

$$\Omega_L := \bigcup_{\vec{l} \in L} \Omega_{\vec{l}}$$

und den zugehörigen Funktionsraum als

$$V_L := \left\{ \sum_{\vec{l} \in L} c_{\vec{l}} f_{\vec{l}}(\vec{x}) \mid \forall \vec{l} \in L : (c_{\vec{l}} \in \mathbb{R} \wedge f_{\vec{l}} \in V_{\vec{l}}) \right\} = \sum_{\vec{l} \in L} V_{\vec{l}} = \sum_{\vec{l} \in \max L} V_{\vec{l}} = \bigoplus_{\vec{\lambda} \in \Lambda} W_{\vec{\lambda}}$$

mit

$$\max L := \left\{ \vec{l} \in L \mid \forall \vec{\lambda} \in L \setminus \{\vec{l}\} \exists k : \lambda_k < l_k \right\}$$

und

$$\Lambda = \left\{ \vec{\lambda} \in \mathbb{N}_0^d \mid \forall \vec{l} \in \max L : \vec{\lambda} \leq \vec{l} \right\}$$

Des Weiteren bezeichnen wir mit

$$\#(f_L) := |L|$$

die Anzahl der Teillösungen einer Kombinationstechnik.

Da f_L , eine hoffentlich bessere, Approximation von f sein soll, und die Teillösungen $f_{\vec{l}}, \vec{l} \in L$ bereits Approximationen von f waren, sind wir nur an Linearkombinationen interessiert für die gilt, dass die Summe $\sum_{\vec{l} \in L} c_{\vec{l}}$ der Koeffizienten 1 ergibt. Um die Kombination auch in der Praxis berechnen zu können, muss des Weiteren die Anzahl der Teillösungen endlich sein.

2.2. Truncated Combination Technique

[HH14] und [BBNS12] verwenden unterschiedliche Notationen für die Truncated Combination Technique. Wir werden hier beide darstellen, wobei die Darstellung nach Benk et al. für unsere Definition der Dünngitter-Kombinationstechnik angepasst wird. Die Notation von Harding und Hegland wird verwendet um die Ergebnisse dieser Arbeit besser mit den Ergebnissen in [HH14] vergleichen zu können, sie wird verallgemeinert und anschließend eine eigene allgemeinere Fehlerabschätzung gemacht.

Definition 2.2.1 (Truncated Combination Technique (T-CT) nach [BBNS12]) Sei $\vec{r} = (r_1, \dots, r_d)$ und $n \in \mathbb{N}$ gegeben, dann definieren wir die Truncated Combination Technique durch:

$$f_n^{\vec{r}}(\vec{x}) := \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\substack{\|\vec{t}\|_1 = n+d-1-q \\ \wedge \vec{r} \leq \vec{t}}} f_{\vec{t}}(\vec{x})$$

Teillösungen die aus Funktionsräumen stammen bei welchen die zugrunde liegenden Gitter nicht in jeder Dimension eine gewisse Mindestauflösung besitzen werden also aus der Kombinationstechnik ausgeschlossen.

2. Definitionen, Versuchsaufbau und Ergebnisse

Definition 2.2.2 (Truncated Combination Technique (T-CT) nach [HH14]) [HH14] definiert die Truncated Combination Technique in 2D durch

$$\tilde{f}_{(l_1, l_2)}^t := \sum_{\alpha=0}^t f_{(l_1+\alpha, l_2+t-\alpha)} - \sum_{\alpha=0}^{t-1} f_{(l_1+\alpha, l_2+t-1-\alpha)}$$

und in 3D durch

$$\tilde{f}_{(l_1, l_2, l_3)}^t := \sum_{\alpha+\beta=0}^t f_{(l_1+\alpha, l_2+\beta, l_3+t-\alpha-\beta)} - 2 \sum_{\alpha+\beta=0}^{t-1} f_{(l_1+\alpha, l_2+\beta, l_3+t-1-\alpha-\beta)} + \sum_{\alpha+\beta=0}^{t-2} f_{(l_1+\alpha, l_2+\beta, l_3+t-2-\alpha-\beta)}$$

mit

$$\sum_{\alpha+\beta=0}^t := \sum_{\alpha=0}^t \sum_{\beta=0}^{t-\alpha}$$

Wobei ab jetzt die Klammern um die Levelvektoren bei Funktionen, Funktionsräumen und Gittern der einfacheren Schreibbarkeit wegen oft weggelassen werden. Es gilt also $f_{l_1, l_2} := f_{(l_1, l_2)}$ etc.

Wir sehen, dass die beiden Definitionen für $n+2-1 = l_1 + l_2 + t$ und $\vec{r} = (r_1, r_2) = (l_1, l_2)$ in 2D und für $n+3-1 = l_1 + l_2 + l_3 + t$ und $\vec{r} = (r_1, r_2, r_3) = (l_1, l_2, l_3)$ in 3D gleich sind. Wir können also die Notation von Harding et al. verallgemeinern zu:

Definition 2.2.3 (Erweiterung von Definition 2.2.2)

$$\tilde{f}_{\vec{l}}^t := f_{\|\vec{l}\|_1+t-d+1}^{\vec{l}} = \sum_{\vec{\lambda} \in \Lambda} c_{\vec{\lambda}} f_{\vec{\lambda}}$$

für eine bestimmte Menge von Levelvektoren Λ .

Theorem 2.2.4 Die Anzahl der Teillösungen der Kombinationstechnik ist für eine feste Anzahl Dimensionen und einen festen Parameter t konstant:

$$\forall d \in \mathbb{N} \forall t \in \mathbb{N}_0 \exists c \in \mathbb{N} \forall \vec{l} \in \mathbb{N}_0^d : \#(\tilde{f}_{\vec{l}}^t) = c$$

Beweis Seien d die Anzahl der Dimensionen und t fest. Nun gilt folgendes:

$$\tilde{f}_{\vec{l}}^t = f_{\|\vec{l}\|_1+t-d+1}^{\vec{l}} = \sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\substack{\|\vec{\lambda}\|_1 = \|\vec{l}\|_1+t-q \\ \wedge \vec{l} \leq \vec{\lambda}}} f_{\vec{\lambda}}(\vec{x})$$

Die Anzahl der Terme der ersten Summe ist konstant d . Sei q nun fest. Bei der zweiten Summe werden nur Levelvektoren mit einer Levelsumme von $\|\vec{l}\|_1 + t - q$ welche größer als \vec{l} sind betrachtet. Dies entspricht dem $t - q$ maligen Inkrementieren von \vec{l} und somit dem Ziehen mit Zurücklegen ohne Reihenfolge. Ist $t < q$ so ist die zweite Summe leer. Ist $t \geq q$ so ist die Anzahl der Summanden der zweiten Summe

$$\binom{d+t-q-1}{t-q}$$

und damit konstant. Für $q = 0$ ist die zweite Summe nicht leer, da $d+t-1 \geq 0$ und $t \geq 0$ gilt. Damit haben wir mindestens eine Teillösung und $\#(\tilde{f}_{\vec{l}}^t)$ ist konstant.

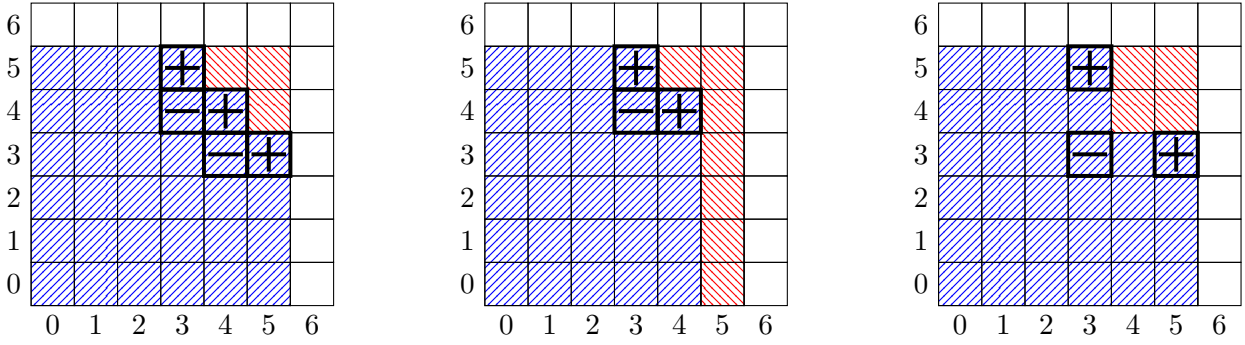


Abbildung 2.3.: Links ist $\tilde{f}_{3,3}^2$ zu sehen. Falls die Teillösung $f_{5,3}$ oder $f_{4,3}$ ausfällt kann die Kombination $f_{3,5} + f_{4,4} - f_{3,4}$ verwendet werden (Mitte). Falls $f_{4,4}$ ausfällt kann z. B. $f_{3,5}$ oder $f_{5,3}$ verwendet werden. Es kann aber auch die kleinere Teillösung $f_{3,3}$ nachberechnet, und die Kombination $f_{3,5} + f_{5,3} - f_{3,3}$ (rechts) verwendet werden. Die blau, von links unten nach rechts oben, schraffierten Quadrate deuten die hierarchischen Inkremente W_{l_1, l_2} der Kombinationen an und die rot, von links oben nach rechts unten, schraffierten Quadrate, die hierarchischen Inkremente, die im Vergleich zur Vollgitterlösung $f_{n,n}$ fehlen.

Sei $t \in \mathbb{N}_0$ nun fest, dann betrachten wir nun Kombinationen der Form $\tilde{f}_{\vec{l}_n}^t$ mit $\vec{l}_n = (n-t, n-t, \dots, n-t)$ und $n-t \geq 0$. Diese haben nicht nur konstant viele Teillösungen, was die Untersuchung hinsichtlich dem Ausfall einzelner Teillösungen im Gegensatz zur klassischen Dünngitter-Kombinationstechnik vereinfacht, sondern sind auch invariant bezüglich einer Permutation der Indizes, haben somit also eine gewisse „Symmetrie“.

In der Arbeit werden folgende Fälle näher betrachtet:

$$\begin{aligned} \tilde{f}_{n-2, n-2}^2 &= f_{n-2, n} + f_{n-1, n-1} + f_{n, n-2} \\ &\quad - f_{n-2, n-1} - f_{n-1, n-2} \end{aligned} \quad (2.1)$$

$$\begin{aligned} \tilde{f}_{n-3, n-3}^3 &= f_{n-3, n} + f_{n-2, n-1} + f_{n-1, n-2} + f_{n, n-3} \\ &\quad - f_{n-3, n-1} - f_{n-2, n-2} - f_{n-1, n-3} \end{aligned} \quad (2.2)$$

$$\begin{aligned} \tilde{f}_{n-2, n-2, n-2}^2 &= f_{n-2, n-2, n} + f_{n-2, n, n-2} + f_{n, n-2, n-2} \\ &\quad + f_{n-2, n-1, n-1} + f_{n-1, n-2, n-1} + f_{n-1, n-1, n-2} \\ &\quad - 2f_{n-2, n-2, n-1} - 2f_{n-2, n-1, n-2} - 2f_{n-1, n-2, n-2} \\ &\quad + f_{n-2, n-2, n-2} \end{aligned} \quad (2.3)$$

2. Definitionen, Versuchsaufbau und Ergebnisse

[HH14] nimmt für ein einzelnes anisotropes volles Gitter die Fehlerentwicklung

$$\epsilon_{l_1, l_2} := f - f_{l_1, l_2} = C_1(h_{l_1})h_{l_1}^p + C_2(h_{l_2})h_{l_2}^p + D(h_{l_1}, h_{l_2})h_{l_1}^p h_{l_2}^p \quad (2.4)$$

an den Gitterpunkten in 2D an. Dabei gelte, dass C_1 von h_{l_1}, x_1 und x_2 , C_2 von h_{l_2}, x_1 und x_2 und D von h_{l_1}, h_{l_2}, x_1 und x_2 abhängig ist. Ferner seien $|C_1|, |C_2|$ und $|D|$ durch eine positive Konstante κ beschränkt.

Unter diesen Bedingungen zeigt [HH14] dass

$$\left| f - \tilde{f}_{n-t, n-t}^t \right| \leq 2\kappa h_n^p + \mathcal{O}\left((h_n^p)^2\right) \quad (2.5)$$

gilt und die Konvergenz des Fehlers von $\tilde{f}_{n-t, n-t}^t$ von der gleichen Ordnung wie die des Fehlers der Vollgitterlösung $f_{n,n}$ ist, wobei der Polynomgrad der d-polynomialen Basisfunktionen $p - 1$ entspricht¹. Für den 3D Fall wird unter Annahme der Fehlerexpansion

$$\begin{aligned} \epsilon_{l_1, l_2, l_3} := f - f_{l_1, l_2, l_3} &= C_1(h_{l_1})h_{l_1}^p + C_2(h_{l_2})h_{l_2}^p + C_3(h_{l_3})h_{l_3}^p \\ &+ D_1(h_{l_1}, h_{l_2})h_{l_1}^p h_{l_2}^p + D_2(h_{l_1}, h_{l_3})h_{l_1}^p h_{l_3}^p + D_3(h_{l_2}, h_{l_3})h_{l_2}^p h_{l_3}^p \\ &+ E(h_{l_1}, h_{l_2}, h_{l_3})h_{l_1}^p h_{l_2}^p h_{l_3}^p \end{aligned} \quad (2.6)$$

eine Abschätzung für den Fehler von $\tilde{f}_{n-t, n-t, n-t}^t$ durch $\left| f - \tilde{f}_{n-t, n-t, n-t}^t \right| \leq 3\kappa h_n^p + \mathcal{O}\left((h_n^p)^2\right)$ angegeben. Dabei seien wiederum $|C_1|, |C_2|, |C_3|, |D_1|, |D_2|, |D_3|$ und $|E|$ durch eine positive Konstante κ beschränkt und von den Koordinaten des Gitterpunktes sowie den jeweiligen Maschenweiten h_{l_1}, h_{l_2} und h_{l_3} abhängig.

Fällt eine Lösung aus Gleichung 2.1 aus. So kann die neue Kombinationstechnik g_n wie folgt gebildet werden:

$$g_n = \begin{cases} f_{n, n-2} + f_{n-1, n-1} - f_{n-1, n-2} & \text{für einen Fehler bei } f_{n-2, n} \text{ oder } f_{n-2, n-1} \\ f_{n-2, n} + f_{n-1, n-1} - f_{n-2, n-1} & \text{für einen Fehler bei } f_{n, n-2} \text{ oder } f_{n-1, n-2} \\ f_{n, n-2} \text{ oder } f_{n-2, n} & \text{für einen Fehler bei } f_{n-1, n-1} \end{cases}$$

Unter Annahme der Fehlerexpansion in Gleichung 2.4 zeigt [HH14] für g_n , dass der Fehler immer noch in $\mathcal{O}(h_n^p)$ liegt. Tritt bei einer der Teillösungen aus $\tilde{f}_{n-3, n-3}^3$ oder aus $\tilde{f}_{n-2, n-2, n-2}^2$ ein Fehler auf so bemerken Harding et al., dass für die entsprechenden Kombinationen im Fehlerfall der Betrag des Fehlers ebenfalls in $\mathcal{O}(h_n^p)$ sei.

Da wir später viele verschiedene Fälle betrachten werden, wollen wir für schwächere Bedingungen eine nicht notwendigerweise kleinste obere Schranke beweisen.

Wir nehmen nun folgende Fehlerexpansion an.

¹Bungartz et. al. verwenden hier direkt für p den Polynomgrad [BG04]

$$\epsilon_{\vec{l}} := f - f_{\vec{l}} = \sum_{K \in \mathcal{P}(\{1,2,\dots,d\}) \setminus \{\emptyset\}} C_K(\dots) \prod_{k \in K} h_{l_k}^p \quad (2.7)$$

Dabei sei d die Anzahl der Dimension und für eine beliebige Menge M bezeichne $\mathcal{P}(M)$ die Potenzmenge von M . C_K sei von den Maschenweiten h_{l_k} , $k \in K$ und den Koordinaten des Gitterpunktes abhängig. Des Weiteren existiere eine positive Konstante κ so dass $\forall K \in \mathcal{P}(\{1, 2, \dots, d\}) \setminus \{\emptyset\} : |C_K| \leq \kappa$.

Theorem 2.2.5 Seien $p \in \mathbb{N}$, $d \in \mathbb{N}$ und $t \in \mathbb{N}_0$ beliebig aber fest, es gelte Gleichung (2.7) und sei $(K_n)_{n \in \mathbb{N}}$ eine beliebige Folge von Linearkombinationen von Teillösungen mit

$$K_n = \sum_{\vec{l} \in L_n} c_{\vec{l}} f_{\vec{l}}$$

mit $\forall n \in \mathbb{N} : L_n \subseteq \mathbb{N}_0^d$. Des Weiteren gelte

$$\forall n \in \mathbb{N} \forall \vec{l} \in L_n : \left(\|\vec{l}\|_1 \leq d * (n - t) + \mu \wedge \overrightarrow{\max\{n - t, 0\}} \leq \vec{l} \wedge |c_{\vec{l}}| \leq \nu \right)$$

für $\mu \in \mathbb{N}_0$ und $\nu \in \mathbb{R}^+$ beliebig aber fest und

$$\forall n \in \mathbb{N} \sum_{\vec{l} \in L_n} c_{\vec{l}} = 1 \quad (\star)$$

dann gilt

$$|f - K_n| \in \mathcal{O}(h_n^p)$$

Beweis Sei $M = \mathcal{P}(\{1, 2, \dots, d\}) \setminus \{\emptyset\}$, sei $n \in \mathbb{N}$ beliebig aber fest, dann gilt für alle $\vec{l} \in L_n$

$$\begin{aligned} |f - f_{\vec{l}}| &= \left| \sum_{K \in M} C_K(\dots) \prod_{k \in K} h_{l_k}^p \right| \leq \sum_{K \in M} \kappa \prod_{k \in K} h_{l_k}^p \leq \kappa \sum_{K \in M} \prod_{k \in K} h_{n-t}^p \leq \kappa \sum_{K \in M} h_{n-t}^{|K|p} \\ &= \kappa \sum_{K \in M} 2^{t|K|p} h_n^{|K|p} = d * \kappa * 2^{tp} h_n^p + \mathcal{O}(h_n^{2p}) = \tilde{\kappa} h_n^p + \mathcal{O}(h_n^{2p}) \end{aligned}$$

wobei für den Fall $n < t$ die Definition von h_{n-t} auf $h_{n-t} = 2^{t-n}$ erweitert wird. Nun folgt

$$\begin{aligned} |f - K_n| &= \left| f - \sum_{\vec{l} \in L_n} c_{\vec{l}} f_{\vec{l}} \right| \stackrel{(\star)}{=} \left| \sum_{\vec{l} \in L_n} c_{\vec{l}} (f - f_{\vec{l}}) \right| \leq \sum_{\vec{l} \in L_n} |c_{\vec{l}}| (f - f_{\vec{l}}) \leq \nu \sum_{\vec{l} \in L_n} (\tilde{\kappa} h_n^p + \mathcal{O}(h_n^{2p})) \\ &\leq \nu (\tilde{\kappa} h_n^p + \mathcal{O}(h_n^{2p})) \sum_{a=0}^{\mu} \binom{d+a-1}{a} = \nu \binom{d+\mu}{\mu} \tilde{\kappa} h_n^p + \mathcal{O}(h_n^{2p}) \end{aligned}$$

2.3. Verfahren für das Berechnen einer Kombination im Fehlerfall

Fallen Teillösungen aus so verwenden Harding und Hegland in [HH12] und [HH14] folgende Strategien zur Berechnung einer neuen Linearkombination von Teillösungen.

2. Definitionen, Versuchsaufbau und Ergebnisse

Definition 2.3.1 (Projektionsoperatoren) Sei V die Menge aller Funktionen von $[0, 1]^d$ nach \mathbb{R} . Dann bezeichne für einen Levelvektor $\vec{l} \in \mathbb{N}_0^d$, $P_{\vec{l}} : V \rightarrow V_{\vec{l}}$ den zugehörigen linearen Projektionsoperator. Sei $L \subseteq \mathbb{N}_0^d$ ein Downset, d.h. es gelte

$$\forall \vec{l}, \vec{\lambda} \in \mathbb{N}_0^d : (\vec{l} \in L \wedge \vec{\lambda} \leq \vec{l} \Rightarrow \vec{\lambda} \in L)$$

dann bezeichne

$$P_L : V \rightarrow V_L, P_L = 1 - \prod_{\vec{l} \in L} (1 - P_{\vec{l}})$$

Sei $\Lambda \subseteq \mathbb{N}_0^d$ dann bezeichne $\downarrow \Lambda \subseteq \mathbb{N}_0^d$ das kleinste Downset, das Λ enthält.

[HH12] und [Heg01] halten fest, dass für die Verkettung zweier Projektionsoperatoren

$$P_{\vec{l}} P_{\vec{\lambda}} = P_{\vec{\lambda}} P_{\vec{l}} = P_{\vec{l} \wedge \vec{\lambda}}$$

gilt, und somit der lineare Operator P_L sich auch schreiben lässt als

$$P_L = 1 - \prod_{\vec{l} \in \max L} (1 - P_{\vec{l}}) = \sum_{\vec{l} \in L} c_{\vec{l}} P_{\vec{l}} \quad (2.8)$$

Haben wir also für beliebiges $L \subseteq \mathbb{N}_0^d$ eine beliebige Linearkombination $\sum_{\vec{l} \in L} c_{\vec{l}} f_{\vec{l}}$ und fallen Teillösungen $f_{\vec{\lambda}}$ für $\vec{\lambda} \in \Lambda \subseteq L$ aus, dann entfernen wir nun die entsprechenden Teillösungen, welche aus den Funktionsräumen stammen, welche feiner als ein beliebiges $V_{\vec{\lambda}}$ für $\vec{\lambda} \in \Lambda$ sind. Sei nun $K = \{ \vec{l} \in L \mid \forall \vec{\lambda} \in \Lambda : \vec{\lambda} \not\leq \vec{l} \}$ dann liefert $P_{\downarrow K}$ uns eine neue Kombinationstechnik, welche die ausgefallenen Lösungen nicht enthält, dabei müssen eventuell neue kleinere Teillösungen, d. h. aus größeren Funktionsräumen, nachberechnet werden.

Beispiel:

Sei die Dünngitter-Kombinationstechnik f_4^c wie in Abbildung 2.4 gegeben und es falle die Teillösung $f_{2,3}$ aus. Wir entfernen zusätzlich $f_{3,3}$ und $f_{2,4}$ und es gilt nun $K = \{(1, 5), (4, 2), (5, 1), (1, 4), (3, 2), (4, 1)\}$ und damit $\max \downarrow K = \{(1, 5), (4, 2), (5, 1)\}$ und damit

$$\begin{aligned} P_{\downarrow K} &= 1 - \prod_{\vec{l} \in \max \downarrow K} (1 - P_{\vec{l}}) \\ &= 1 - \left((1 - P_{(1,5)}) (1 - P_{(4,2)}) (1 - P_{(5,1)}) \right) \\ &= 1 - \left(1 - P_{(1,5)} - P_{(4,2)} - P_{(5,1)} + P_{(1,5)} P_{(4,2)} + P_{(1,5)} P_{(5,1)} + P_{(4,2)} P_{(5,1)} - P_{(1,5)} P_{(4,2)} P_{(5,1)} \right) \\ &= P_{(1,5)} + P_{(4,2)} + P_{(5,1)} - P_{(1,5)} P_{(4,2)} - P_{(1,5)} P_{(5,1)} - P_{(4,2)} P_{(5,1)} + P_{(1,5)} P_{(4,2)} P_{(5,1)} \\ &= P_{(1,5)} + P_{(4,2)} + P_{(5,1)} - P_{(1,2)} - P_{(1,1)} - P_{(4,1)} + P_{(1,1)} \\ &= P_{(1,5)} + P_{(4,2)} + P_{(5,1)} - P_{(1,2)} - P_{(4,1)} \end{aligned}$$

Fällt in der Kombination $\tilde{f}_{n-2,n-2}^2 = f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - f_{n-2,n-1} - f_{n-1,n-2}$ beispielsweise $f_{n-1,n-1}$ aus so liefert uns dieselbe Technik die Kombination $f_{n-2,n} + f_{n,n-2} - f_{n-2,n-2}$, welche

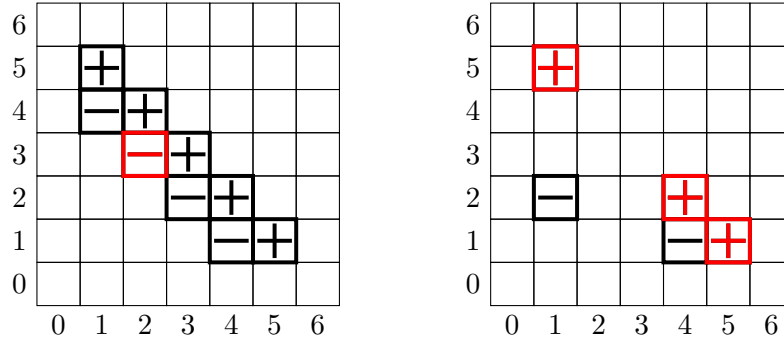


Abbildung 2.4.: Die Berechnung der neuen Linearkombination nach dem Inklusions- und Exklusionsprinzip. Links ist die ursprüngliche Kombination zu sehen, rot markiert ist die ausgefallene Teillösung. Rechts ist die neu berechnete Linearkombination zu sehen, die Teillösungen $f_{\vec{k}}$ mit $\vec{k} \in \max \downarrow K$ sind in roter Farbe markiert.

dem Fall 5 in Anhang A.1 entspricht. Die neue Teillösung $f_{n-2,n-2}$ kann Neuberechnet oder z. B. durch eine Projektion von $f_{n-2,n}$ auf $V_{n-2,n-2}$ approximiert werden.

Wir sehen, dass dieses Verfahren dem Prinzip der Inklusion und Exklusion aus der Mengenlehre auf den Gittern $\Omega_{\vec{k}}$ mit $\vec{k} \in \max \downarrow K$ entspricht.

Eine andere Möglichkeit ist, durch geschickte Anpassung der Koeffizienten der Kombinationstechnik ohne die Berechnung neuer Teillösungen oder Interpolation auszukommen, wie sie etwa in [HH14] für die T-CTs $\tilde{f}_{n-2,n-2}^2$, $\tilde{f}_{n-3,n-3}^3$ und $\tilde{f}_{n-2,n-2,n-2}^2$ untersucht wird. Fällt in $\tilde{f}_{n-2,n-2}^2$ beispielsweise $f_{n-2,n-1}$ aus, so kann die Kombination

$$f_{n-1,n-1} + f_{n,n-2} - f_{n-1,n-2}$$

gebildet werden.

Die dritte Möglichkeit ist, eine ausgefallene Teillösung durch eine Projektion einer Teillösung aus einem feineren Funktionsraum zu ersetzen. Fällt in $\tilde{f}_{n-2,n-2}^2$ beispielsweise $f_{n-2,n-1}$ aus, so kann eine neue Kombination mit

$$f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - P_{n-2,n-1} f_{n-2,n} - f_{n-1,n-2}$$

gebildet werden.

Um den Fehler einer Kombination zu messen verwenden wir die diskrete \mathcal{L}^1 Norm. Sei $f \in V$ die exakte Lösung einer PDE und f_L eine Linearkombination von Teillösungen, dann berechnen wir den Fehler für beliebiges $\vec{\lambda} \in \mathbb{N}_0^d$ in $V_{\vec{\lambda}}$ durch

$$\|f - f_L\|_1 = \sum_{\vec{x} \in \Omega_{\vec{\lambda}}} |f(\vec{x}) - f_L(\vec{x})| \quad (2.9)$$

Bei der Bewertung obiger vorgestellter Verfahren muss einerseits betrachtet werden wie groß der Fehler der neu erhaltenen Linearkombination ist und dies in Relation zum Aufwand für ihren Erhalt gesetzt werden.

2. Definitionen, Versuchsaufbau und Ergebnisse

Wir wollen nun das Verfahren der Approximation einer ausgefallenen Teillösung durch Interpolation mit dem obigen Verfahren zur Berechnung der Koeffizienten einer neuen Linearkombination und der Neuberechnung kleinerer Teillösungen hinsichtlich der Laufzeit vergleichen. Zugunsten der Interpolationslösung nehmen wir an, diese würde keinen zusätzlichen Aufwand verursachen und zeigen, dass sogar unter dieser Annahme und den Ergebnissen in Abschnitt 2.5 für den Fehler, die Neuberechnung der Koeffizienten, zumindest für die in Abschnitt 2.4 beschriebenen PDEs die bessere Strategie darzustellen scheint, und werden deswegen diese neben der Richardson-Extrapolation, siehe dazu auch Abschnitt 2.6, für unsere Implementierung in Kapitel 3 wählen.

Der Aufwand für das Verfahren mit den Projektionsoperatoren setzt sich aus der Berechnung der Koeffizienten und aus der Neuberechnung evtl. kleinerer Teillösungen zusammen.

Bisher haben wir T-CT's der Form $\tilde{f}_{\vec{l}_n}^t$ mit $\vec{l}_n = (n-t, n-t, \dots, n-t)$ und $n-t \geq 0$ für festes $t \in \mathbb{N}_0$ betrachtet. Die Summe der Anzahl der Gitterpunkte der einzelnen Teillösungen liegt wie bei der Vollgitterlösung $f_{n,n}$ in $\mathcal{O}(h_n^{-d})$, und der Fehler liegt, genauso wie bei $f_{n,n}$, falls die exakte Lösung der PDE genügend glatt ist, in $\mathcal{O}(h_n^p)$. Beide sind also in gleicher Weise von dem Fluch der Dimensionalität betroffen [HH14]. In der Praxis bieten sich daher T-CTs der Form $f_n^{\vec{r}}$ für einen festen Levelvektor \vec{r} oder etwa für $\vec{r} = (n/2, n/2, \dots, n/2)$ an [BP12]. Für diese werden auch die Algorithmen in SG++, wie in Kapitel 3 dargestellt, implementiert.

Deswegen schätzen wir nun den Aufwand für die Berechnung der Koeffizienten einer neuen Linearkombination für die Dünngitter-Kombinationstechnik (!) ab. Für die Anzahl der Teillösungen der Dünngitter-Kombinationstechnik $f_n^c = \sum_{\vec{l} \in L} c_{\vec{l}} f_{\vec{l}}$ gilt: $|L| \in \mathcal{O}((\log h_n^{-1})^{d-1}) = \mathcal{O}(n^{d-1})$ [Pfl10].

Sei $K = \{ \vec{l} \in L \mid \forall \vec{\lambda} \in \Lambda : \vec{\lambda} \not\leq \vec{l} \}$, wobei $\Lambda \subseteq L$ die Menge der Levelvektoren der ausgefallenen Teillösungen sei. Aus $\max \downarrow K = \max K \subseteq K \subseteq L$ folgt $|\max K| = |\max \downarrow K| \leq |K| \leq |L| \in \mathcal{O}(n^{d-1})$ und damit $|\max \downarrow K| \in \mathcal{O}(n^{d-1})$ und $|K| \in \mathcal{O}(n^{d-1})$. Würden wir jetzt einfach naiv wie oben das Distributivgesetz auf das Produkt $\prod_{\vec{l} \in \max \downarrow K} (1 - P_{\vec{l}})$ anwenden, hätten wir $\mathcal{O}(2^{\mathcal{O}(n^{d-1})})$ viele Summanden.

Aber es gilt

$$\forall \vec{l} \in \downarrow K : \|\vec{l}\|_1 \leq n + d - 1 \Rightarrow |\downarrow K| \leq (n + d)^d \in \mathcal{O}(n^d)$$

Zerlegen wir also das Produkt rekursiv in

$$\prod_{\vec{l} \in \max \downarrow K} (1 - P_{\vec{l}}) = (1 - P_{\vec{k}}) \underbrace{\left(\prod_{\vec{l} \in \max \downarrow K \setminus \{\vec{k}\}} (1 - P_{\vec{l}}) \right)}_{\text{schon vereinfacht}}$$

und vereinfachen wir nach jedem Schritt Ausdrücke der Form $P_{\vec{\lambda}} P_{\vec{k}}$ gleich zu $P_{\vec{\lambda} \wedge \vec{k}}$ und fassen diese zusammen, so haben wir also eine Rekursionstiefe in $\mathcal{O}(n^{d-1})$ mit jeweils maximal $\mathcal{O}(n^d)$ vielen Additionen und Multiplikationen.

Die Beträge aller Koeffizienten sind in jedem Rechenschritt durch $\mathcal{O}(2^{mn^{d-1}})$, für eine Konstante $m > 0$ beschränkt. Jede Multiplikation und Addition von Koeffizienten hat also höchstens einen Aufwand von $\mathcal{O}((n^{d-1})^2)$.

Ein Levelvektor $\vec{\lambda} = (\lambda_1, \dots, \lambda_d)$ besteht aus d Komponenten mit jeweils $\mathcal{O}(\log(n))$ Bits. Seien zwei Levelvektoren $\vec{\lambda} = (\lambda_1, \dots, \lambda_d)$ und $\vec{k} = (k_1, \dots, k_d)$ gegeben. Für alle $i \in \{1, \dots, d\}$ lässt sich $\min\{\lambda_i, k_i\}$, z. B. durch stellenweises Vergleichen, mit einem Aufwand in $\mathcal{O}(\log(n))$ berechnen und damit liegt der Aufwand für die Berechnung von $\vec{\lambda} \wedge \vec{k}$ in $\mathcal{O}(d * \log(n))$. Für die Bestimmung von K und dann $\max K = \max \downarrow K$ liefert ein paarweiser Vergleich der $\mathcal{O}(n^{d-1})$ vielen Levelvektoren jeweils einen Aufwand von $\mathcal{O}(d * \log(n) * (n^{d-1})^2)$.

Wir haben höchstens $\mathcal{O}(n^d)$ viele Daten mit einer Größe in $\mathcal{O}(n^{d-1})$ für Koeffizienten und Levelvektoren, damit ist der Speicherplatz polynomiell in n beschränkt und damit auch in polynomieller Zeit ein Zugriff möglich.

Wir haben unter anderem mindestens eine Teillösung aus f_n^c erfolgreich berechnet, denn sonst ist obiges Verfahren nicht sinnvoll anwendbar. Für alle Teillösungen aus f_n^c gilt, dass die Anzahl der Gitterpunkte in $\Omega(2^n)$ liegt. Mit einem optimalen Löser, der eine lineare Komplexität in der Anzahl der Gitterpunkte hat, haben wir also ebenfalls einen Aufwand in $\Omega(2^n)$. Sei nun also A_n der Aufwand für die Berechnung der Koeffizienten und B_n der Aufwand für das Berechnen der Teillösungen, dann gilt:

$$\frac{A_n}{B_n} \in \frac{\mathcal{O}(n^d) * \mathcal{O}(n^{d-1}) * \left(\mathcal{O}(d * \log(n)) + \mathcal{O}((n^{d-1})^2) \right) * \mathcal{O}(n^c) + \mathcal{O}(d * \log(n) * (n^{d-1})^2)}{\Omega(2^n)} \quad (2.10)$$

für eine Konstante $c \in \mathbb{N}$ und damit gilt:

$$\lim_{n \rightarrow \infty} \frac{A_n}{B_n} = 0 \quad (2.11)$$

Der Aufwand für das Berechnen der Koeffizienten fällt also, im Vergleich zum Aufwand für die Berechnung der Teillösung asymptotisch kaum ins Gewicht und wird deswegen vernachlässigt. Für den Speicherbedarf gilt die gleiche Aussage.

Nun muss nur noch der Aufwand für das Berechnen neuer Teillösungen abgeschätzt werden. Sei $I_n^d := \{ \vec{l} \in \mathbb{N}^d \mid n - d < \|\vec{l}\|_1 \leq n \}$, $J_n^d := \{ \vec{l} \in \mathbb{N}^d \mid \|\vec{l}\|_1 \leq n \}$ und für einen beliebigen Levelvektor $\vec{l} \in \mathbb{N}_0^d$ bezeichne $|\Omega_{\vec{l}}|$ die Menge der Gitterpunkte von $\Omega_{\vec{l}}$. Dann beweisen Harding und Hegland, dass für den Quotient aus den Summen der Gitterpunkte

$$L_n = \sum_{\vec{l} \in I_n^d} |\Omega_{\vec{l}}| \quad \text{und} \quad M_n = \sum_{\vec{l} \in J_n^d} |\Omega_{\vec{l}}|$$

der Grenzwert

$$\lim_{n \rightarrow \infty} \frac{M_n - L_n}{L_n} = \frac{1}{2^d - 1} \quad (2.12)$$

gilt. [HH12]

Das Verfahren bei einem Ausfall von Teillösungen einfach alle Teillösungen auf feineren Gittern zu verwerfen und die Linearkombination, wie oben beschrieben zu berechnen führt dazu, dass für alle neu zu berechnenden Teillösungen $f_{\vec{k}}$ mit $\vec{k} \in \mathbb{N}_0^d$ gilt, dass \vec{k} in $J_n^d \setminus I_n^d$ liegt. Führen wir eine Indexverschiebung um $d - 1$ durch, so können wir obige Grenzwertbetrachtungen auf unsere Definition der Dünngitter-Kombinationstechnik in Definition 2.1.6 übertragen.

Harding und Hegland schlagen vor alle kleineren Teillösungen, unabhängig davon ob sie gebraucht werden, gleich im Voraus mitzuberechnen [HH12]. Unter der Annahme eines optimalen Löser mit einer Komplexität die linear in der Anzahl der Gitterpunkte ist und unter Berücksichtigung von Gleichung 2.11 und 2.12 ergibt sich bereits für $d = 2$ die Prognose, dass sich das Verfahren für große n , falls für den Fehler e_n im Vergleich zu dem Fehler \tilde{e}_n für das Verfahren einer Interpolation $e_n < \frac{3}{4}\tilde{e}_n$ gilt, lohnt.

2.4. Beschreibung der PDEs und der Testprogramme

In dieser Arbeit untersuchen wir zwei PDEs. Zum einen das Laplace Problem $\Delta f = 0$ in 2 Dimensionen mit der exakten Lösung $f(x, y) = \sin(\pi y) \frac{\sinh(\pi(1-x))}{\sinh(\pi)}$ auf $[0, 1]^2$ und in 3 Dimensionen mit der exakten Lösung $f(x, y, z) = \sin(\pi z) \sin(\pi y) \frac{\sinh(\sqrt{2}\pi(1-x))}{\sinh(\pi)}$ auf $[0, 1]^3$, jeweils mit Dirichlet Randbedingungen. Wir diskretisieren mit Finiten Elementen zweiter Ordnung und verwenden ein CG-Verfahren als Löser.

Die zweite PDE ist die Advektionsgleichung $\frac{\partial f}{\partial t} + \vec{a} * \nabla f = 0$ mit $\vec{a} = (1, 1)$, der exakten Lösung $f(x, y, t) = \sin(2\pi(x - t)) \sin(2\pi(y - t))$ auf $[0, 1]^2$ und periodischen Randbedingungen in 2D und mit $\vec{a} = (1, 1, 1)$, der exakten Lösung $f(x, y, z, t) = \sin(2\pi(x - t)) \sin(2\pi(y - t)) \sin(2\pi(z - t))$ auf $[0, 1]^3$ und periodischen Randbedingungen in 3D. Wir starten mit jeweils einer Anfangsbedingung zum Zeitpunkt $t = 0$ und lösen das Problem auf jedem Gitter bis zum Zeitpunkt $t = 0.5$, ganz am Ende wird die Kombinationslösung gebildet. Für die Zeitschritte wird ein Runge-Kuttaverfahren vierter Ordnung und eine Diskretisierung 1. Ordnung für die Raumkoordinaten verwendet.

Die Advektionsgleichung beschreibt beispielsweise den Transport eines gelösten Stoffes in einer Flüssigkeit durch eine Strömung. Das Laplace Problem entspricht dem stationären Fall der homogenen Wärmeleitungsgleichung.

Abbildung 2.5 zeigt die Struktur der Programme für die Untersuchungen der beiden PDEs. Für die Parallelisierung wurde die MPI Implementierung MPICH2 in Version 1.4.1 verwendet. Um die beiden PDEs zu lösen verwenden wir DUNE (Distributed and Unified Numerics Environment) [PB], mit den DUNE-Core Modules in Version 2.2 und DUNE-FEM [dev10] in Version 1.30. Das Dünngitter-Framework SG++ liefert uns die Tools um die Kombinationstechnik zu realisieren. Dort sind bereits

die Klassen `CombiSchemeBasis` und `ParallelCombiGrid` vorhanden. Die Basisklasse `CombiSchemeBasis` ist in Abbildung 3.1 dargestellt. Sie ist im wesentlichen ein Container für die Levelvektoren und Koeffizienten einer Kombination $\sum_{\vec{l} \in L} c_{\vec{l}} f_{\vec{l}}$ welche in den Member Variablen `levels_vector_` und `coefficients_` gespeichert werden. Über die einzelnen Methoden können die Koeffizienten bzw. die Levelvektoren gesetzt oder gelesen werden. Der Konstruktor von `ParallelCombiGrid` bekommt ein Objekt der Klasse `CombiSchemeBasis` übergeben. Er alloziert mehrdimensionale Arrays welche die Funktionswerte der einzelnen Teillösungen $f_{\vec{l}}, \vec{l} \in L$ an den jeweiligen Gitterpunkten der zugehörigen Gitter $\Omega_{\vec{l}}$ speichern und stellt Methoden zur Auswertung der Linearkombination auf $[0, 1]^d$ bereit.

Die Klassen `Transport` und `Poisson` hängen von den DUNE Bibliotheken ab und basieren auf den Beispielen in [dev10]. Diese wurden für unsere spezifische Problemstellung angepasst und lösen die entsprechenden PDE's für eine gewählte Diskretisierung. Über Konfigurationsdateien werden Einstellungen wie etwa die maximale Schrittweite des Runge-Kutta Verfahrens oder der Wert für das Residuum festgelegt ab dem der CG-Löser abbricht.

Die Klasse `createSchemes` liefert uns für ein $n \in \mathbb{N}$ die Levelvektoren und Koeffizienten der T-CTs, der Richardson-Extrapolationen (siehe Abschnitt 2.6), der Verknüpfung beider Techniken (siehe Abschnitt 2.7) und der Kombinationen falls eine Teillösung ausfällt. Im Anhang sind alle untersuchten Kombinationen aufgelistet. Des Weiteren liefert sie die Information auf welchem Gitter der Fehler berechnet werden soll und wie ggf. interpoliert werden soll.

In der „combiutils.hpp“ sind die Funktionen für die Kommunikation der Prozesse, die Berechnung des Fehlers auf einem Raum $V_{\vec{l}}$ und Interpolationen implementiert.

Die Einsprungspunkte für die beiden Programme für das Laplace bzw. das Advektionsproblem, befinden sich in den Dateien „combi_laplace.cpp“ bzw. „combi_transport.cpp“.

2.5. Ergebnisse der Kombinationstechnik

Wir betrachten nun im Falle der T-CTs $\tilde{f}_{n-2,n-2}^2$, $\tilde{f}_{n-3,n-3}^3$, und $\tilde{f}_{n-2,n-2,n-2}^2$, verschiedene Kombinationen falls genau eine Teillösung ausfällt. Aus Theorem 2.2.5 folgt, dass wir für alle untersuchten Kombinationen, wie sie im Anhang aufgelistet werden, für das Laplace-Gleichung eine Konvergenz in $\mathcal{O}(h_n^2)$ und für die Advektionsgleichung in $\mathcal{O}(h_n)$ erwarten. Der Fehler wird in allen Ergebnissen dieser Arbeit wie in Gleichung 2.9 dargestellt berechnet.

Abbildung 2.6 zeigt den Fehler für die Laplace-Gleichung in 2D für die Vollgitterlösung $f_{n,n}$, $\tilde{f}_{n-2,n-2}^2$ und verschiedene Kombinationen nach einem Ausfall einer Teillösung der T-CT für $n \in \{5, \dots, 10\}$. Zu erkennen ist, dass sich wie erwartet bei allen Kombinationen, mit der Halbierung der Maschenweite, gleichzeitig der Fehler ungefähr viertelt und sich die Fehler der einzelnen Kombinationen näherungsweise um einen konstanten Faktor unterscheiden. Der schlechteste Fall ist die Fehlerkombination 3, die aus nur einer Teillösung, nämlich $f_{n,n-2}$, besteht.

Die Fehler für die Kombinationen $\tilde{f}_{n-2,n-2}^2$ (case 0) und $f_{n-2,n} + f_{n,n-2} - f_{n-2,n-2}$ (case 5) liegen sehr nahe an der, der Vollgitterlösung $f_{n,n}$. Ein Grund dafür ist, dass wie Abbildung 2.3 zeigt, nur die hierarchischen Inkremente $W_{n,n}$, $W_{n-1,n}$ und $W_{n,n-1}$ bzw. zusätzlich $W_{n-1,n-1}$ fehlen, während z. B.

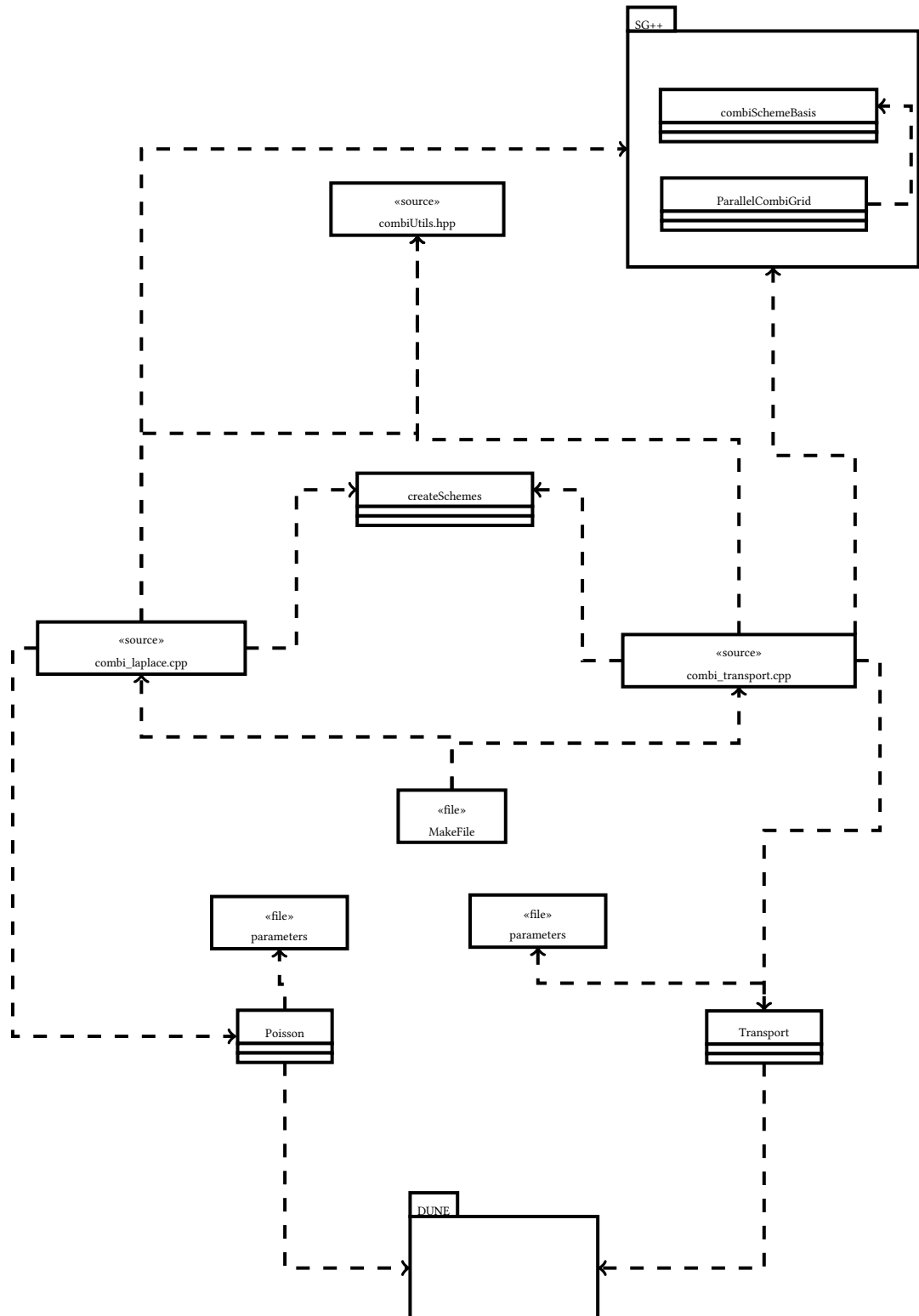


Abbildung 2.5.: Schematische Darstellung des Aufbaus der Programme für die Untersuchungen der Fehler verschiedener Linearkombinationen von Teillösungen.

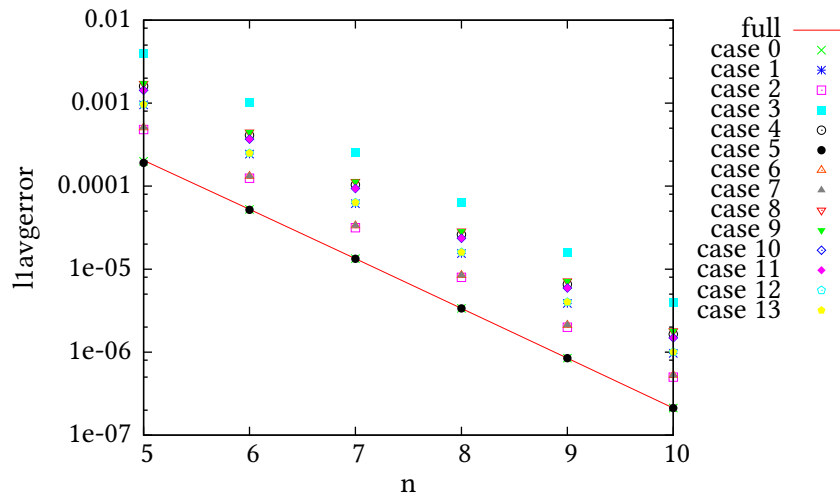


Abbildung 2.6.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Kombinationstechnik $\tilde{f}_{n-2,n-2}^2$ und den Linearkombinationen wie sie in Anhang A.1 auf Seite 53 beschrieben sind, falls eine Teillösung ausfällt. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

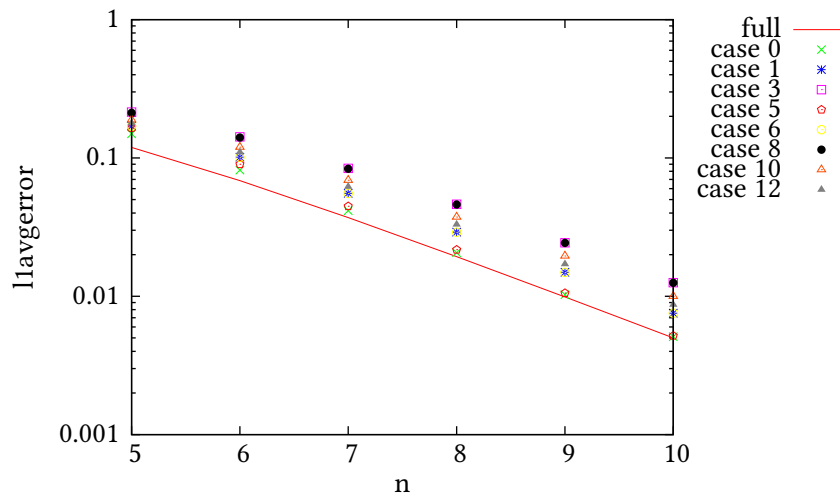


Abbildung 2.7.: Dargestellt sind die Fehler bei der Lösung der Advektionsgleichung mit der Kombinationstechnik $\tilde{f}_{n-2,n-2}^2$ und den Linearkombinationen wie sie in Anhang A.1 auf Seite 53 beschrieben sind, falls eine Teillösung ausfällt. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

2. Definitionen, Versuchsaufbau und Ergebnisse

in Fall 2 die Anzahl der fehlenden hierarchischen Inkremente in $\mathcal{O}(n)$ liegt. Des Weiteren sind auch die einzelnen Beiträge der fehlenden hierarchischen Inkremente in Fall 0 und 5 vergleichsweise klein. Dies liegt daran, dass für die Beiträge der hierarchischen Inkremente bezüglich der \mathcal{L}^∞ Norm die Abschätzungen in [Zen91] gelten, und diese also mit steigender Levelsumme gegen 0 konvergieren. Während in Fall 0 und 5 die Levelsumme der fehlenden hierarchischen Inkremente nach unten durch $2n - 2$ abgeschätzt werden kann, fehlt in Fall 2 etwa auch $W_{n,1}$ mit verhältnismäßig hohem Beitrag. Dass die beiden besten Fälle tatsächlich einen leicht geringeren Fehler als den der Vollgitterlösung $f_{n,n}$ haben, lässt sich vermutlich auf Besonderheiten des Problems und des verwendeten numerischen Verfahrens zurückführen. Dass die exakte Lösung nicht symmetrisch ist lässt sich auch aus einem Vergleich der Fehler für die Kombinationen 1 und 2 bzw. 3 und 4 ablesen, deren Levelvektoren jeweils bis auf eine Permutation der Indizes identisch sind.

Fällt $f_{n-2,n}$ aus so kann entweder Kombination 1 mit $f_{n,n-2} + f_{n-1,n-1} - f_{n-1,n-2}$ oder Kombination 3 mit $f_{n,n-2}$ verwendet werden. Fällt $f_{n,n-2}$ aus so kann entweder Kombination 2 mit $f_{n-2,n} + f_{n-1,n-1} - f_{n-2,n-1}$ oder Kombination 4 mit $f_{n-2,n}$ verwendet werden. In beiden Fällen sind die Linearkombinationen mit jeweils drei Teillösungen die bessere Wahl.

Fällt beispielsweise $f_{n-2,n-1}$ aus, so kann aus den verbleibenden Teillösungen entweder die Kombination 1 mit $f_{n,n-2} + f_{n-1,n-1} - f_{n-1,n-2}$ gebildet werden oder aber eine der beiden Teillösungen $f_{n-2,n}$ oder $f_{n-1,n-1}$, wie in den beiden Fällen 6 und 10, in $V_{n-2,n-1}$ projiziert werden. Für einen Ausfall von $f_{n-1,n-2}$ lassen sich analog die Kombinationen 2, 7, und 11 bilden. Dabei ist zu sehen, dass in drei der vier Interpolationsfälle der Fehler sogar schlechter ist als die entsprechende alternative Kombination mit nur drei Teillösungen. Auch der Aufwand für eine Interpolation spricht gegen diese, im Vergleich zu den Fällen 1 und 2 bei denen nur die Koeffizienten der Linearkombination angepasst werden müssen.

Da der Fehler davon abhängt, welche Teillösung für die Projektion verwendet wird, wirft dies die Frage auf, wie für eine beliebige PDE entsprechende Teillösungen ausgewählt werden. Liegen keine anderen Informationen vor, so ist es naheliegend für einen Ausfall von $f_{n-2,n-1}$ etwa die Linearkombination $\frac{1}{2}(f_{n-2,n} + f_{n-1,n-1})$, wie in Fall 12, und für einen Ausfall von $f_{n-1,n-2}$ die Linearkombination $\frac{1}{2}(f_{n-1,n-1} + f_{n,n-2})$, wie in Fall 13, für eine Projektion zu verwenden. Dies kann etwa auch auf beliebige intakte Teillösungen, aus den zugehörigen feineren Funktionsräumen, in beliebiger Dimension erweitert werden.

In unserem Fall entspricht der Fehler der Kombination 12 dabei ungefähr dem arithmetischen Mittel der Fehler der Kombinationen 6 und 10, und der Fehler von Kombination 13 ungefähr dem arithmetischen Mittel der Fehler der Kombinationen 7 und 11. Damit sind beide schlechter als die jeweiligen Kombinationen 1 und 2 mit jeweils drei Teillösungen. Sowohl bei Ausfall von $f_{n-2,n-1}$ als auch von $f_{n-1,n-2}$ wären jeweils noch die Fälle 3 und 4 denkbar gewesen. Diese sind aber noch schlechter als alle obigen aufgezählten Kombinationen im Fehlerfall.

Zuletzt wollen wir noch einen Fehler bei $f_{n-1,n-1}$ betrachten. Möglich sind hier wieder die Fälle 3 und 4 mit jeweils einer Teillösung. Weitere Möglichkeiten sind Kombination 5 mit $f_{n-2,n} + f_{n,n-2} - f_{n-2,n-2}$ Kombination 8 mit $f_{n-2,n} + f_{n,n-2} - P_{n-2,n-2}f_{n-2,n}$ oder Kombination 9 mit $f_{n-2,n} + f_{n,n-2} - P_{n-2,n-2}f_{n,n-2}$. Erstaunlicherweise ist die Kombination $f_{n-2,n}$ sogar geringfügig besser als die beiden Interpolationsfälle. Der Fehler von Kombination 5 ist ungefähr um

den Faktor 8 besser, als der Fehler der beiden Kombinationen 8 und 9, dafür muss aber die Teillösung $f_{n-2,n-2}$ berechnet werden.

Abbildung 2.7 zeigt analog die Fehler für die Vollgitterlösung $f_{n,n}$, $\tilde{f}_{n-2,n-2}^2$ und die oben besprochenen Kombinationen für die Advektionsgleichung. Da die exakte Lösung $\sin(2\pi(x-t))\sin(2\pi(y-t))$ ist, sind die Ergebnisse für Kombinationen, deren Levelvektoren bis auf eine Permutation der Indizes identisch sind, gleich und nicht in dem Plot enthalten. Die Resultate entsprechen weitgehend den Ergebnissen für die Laplace-Gleichung. So sind die Fälle 0 und 5 die besten und die Interpolationsfälle schneiden schlechter ab, als die entsprechenden Kombinationen mit nur drei Teillösungen bzw. mit der Neuberechnung der kleineren Teillösung $f_{n-2,n-2}$. Unterschiedlich ist, dass sich bei Halbierung der Maschenweite, wie erwartet, der Fehler ungefähr halbiert und dass die Vollgitterlösung $f_{n,n}$ erkennbar besser als alle anderen Kombinationen abschneidet.

Für die T-CT $\tilde{f}_{n-2,n-2}^2$ aus fünf Teillösungen haben wir relativ ausführlich die einzelnen Fälle aufgezählt, andere Kombinationen bei Ausfall einer Teillösung wären durchaus möglich gewesen. Da die T-CT $\tilde{f}_{n-3,n-3}^3$ aus sieben Teillösungen und die T-CT $\tilde{f}_{n-2,n-2,n-2}^2$ aus zehn Teillösungen besteht und für diese noch mehr Möglichkeiten denkbar sind, wollen wir hier nicht mehrere Dutzend Fallunterscheidungen machen und schließen diesen Abschnitt mit der Bemerkung, dass die Abbildungen A.1, A.2, A.5, und A.6 für das Advektionsproblem und das Laplace Problem in 2D und 3D ebenfalls die erwartete Fehlerabnahme zeigen. Für das Laplace Problem im Fall der T-CT $\tilde{f}_{n-3,n-3}^3$, zeigt sich ein etwas besseres Abschneiden der beiden Kombinationen mit Projektionsoperatoren.

Den Ergebnissen und den Betrachtungen in Abschnitt 2.3 lässt sich nun entnehmen, dass die Interpolation im Vergleich zur Neuberechnung von Koeffizienten und kleinerer Teillösungen, die schlechtere Strategie zu sein scheint.

2.6. Richardson-Extrapolation

In Theorem 2.2.5 haben wir obere Schranken, aber keine untere Schranke für den Fehler bewiesen. Die Idee der Richardson-Extrapolation ist es, durch geeignete Wahl der Koeffizienten einer Linearkombination von Teillösungen ein besseres Fehlerverhalten zu erreichen. Dabei wird ausgenutzt, dass sich die ersten Terme der Fehlerexpansion aufheben.

[HH14] geht dazu in 2D von einem punktweisen Fehler von

$$\epsilon_{l_1,l_2} = f - f_{l_1,l_2} = e^{p,0}h_{l_1}^p + e^{0,p}h_{l_2}^p + e^{2p,0}h_{l_1}^{2p} + e^{0,2p}h_{l_2}^{2p} + \dots \quad (2.13)$$

aus, wobei die Funktionen $e^{m,n}$ nur von Koordinaten des Gitterpunktes abhängen.

Sei $L_n^d = \{ \vec{l} \in \mathbb{N}_0^d \mid \|\vec{l}\|_1 = n \}$. Wir verallgemeinern die Fehlerexpansion in 2.13 nun zu

$$\epsilon_{\vec{l}} = f - f_{\vec{l}} = \left(\sum_{j=1}^d e^j h_{l_j}^p \right) + \left(\sum_{k=2}^{\infty} \sum_{\vec{\lambda} \in L_k^d} e^{\vec{\lambda}} \prod_{u=1}^d h_{l_u}^{\lambda_u p} \right) \quad (2.14)$$

wobei die Terme e^j und $e^{\vec{\lambda}}$ wieder nur von den Koordinaten des Gitterpunktes abhängig sind.

2. Definitionen, Versuchsaufbau und Ergebnisse

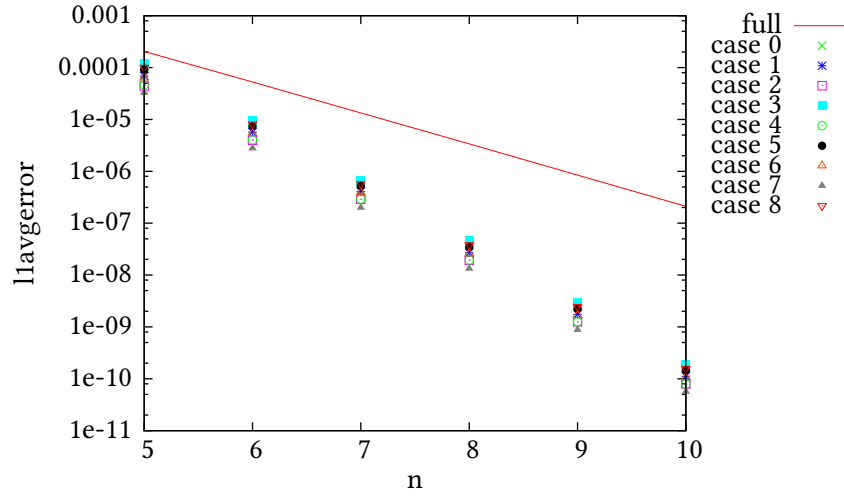


Abbildung 2.8.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Richardson-Extrapolation auf $V_{n-3,n-3}$. Die einzelnen Fälle sind in Anhang A.4 auf Seite 55 beschrieben. Der Fehler für $f_{n,n}$ wird auf $V_{n,n}$ und für alle anderen Fälle auf $V_{n-3,n-3}$ berechnet.

Bei der klassischen Richardson-Extrapolation werden zwei Teillösungen durch die Formel

$$\frac{2^p}{2^p - 1} P_{(n,n,\dots,n)} f_{(n+1,n+1,\dots,n+1)} + \frac{1}{1 - 2^p} f_{(n,n,\dots,n)}$$

kombiniert. Wobei $P_{(n,n,\dots,n)}$ die Projektion bzw. Interpolation wie in Definition 2.3.1 bezeichnet.

Harding schlägt, statt der klassischen, die multivariate Richardson-Extrapolation vor. Diese hat besonders in höheren Dimensionen und bei Extrapolationen höherer Ordnung einen geringeren Aufwand pro Nutzen [BGR94]. Sei z. B. die Kombination $\tilde{f}_{n-2,n-2}^2$ gegeben, so suchen wir nun Koeffizienten so dass folgende Gleichung gilt:

$$a_1 f_{n-2,n} + a_2 f_{n-1,n-1} + a_3 f_{n,n-2} + a_4 f_{n-2,n-1} + a_5 f_{n-1,n-2} = f + 0e^{p,0} + 0e^{0,p} + \mathcal{O}(h_n^{2p}) \quad (2.15)$$

was zu dem Gleichungssystem

$$\begin{aligned} a_1 + a_2 + a_3 + a_4 + a_5 &= 1 \\ a_1 h_{n-2}^p + a_2 h_{n-1}^p + a_3 h_n^p + a_4 h_{n-2}^p + a_5 h_{n-1}^p &= 0 \\ a_1 h_n^p + a_2 h_{n-1}^p + a_3 h_{n-2}^p + a_4 h_{n-1}^p + a_5 h_{n-2}^p &= 0 \end{aligned}$$

führt.[HH14]

Da Gleichung 2.13 nur an den Gitterpunkten gilt, ist die Auslöschung der ersten Fehlerterme im Allgemeinen auch nur auf dem größten gemeinsamen Gitter garantiert [HH14]. Dies deckt sich auch mit den Ergebnissen der Experimente. So zeigt Abbildung 2.8, die Abnahme des Fehlers um den Faktor

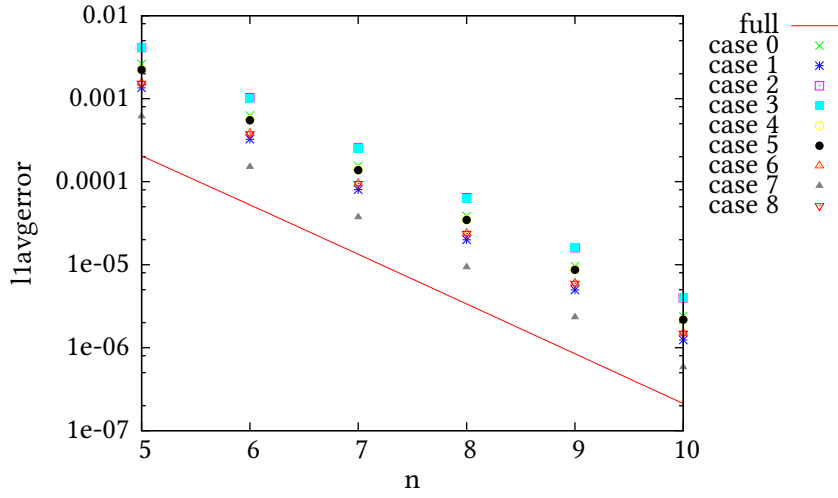


Abbildung 2.9.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Richardson-Extrapolation auf $V_{n,n}$. Die einzelnen Fälle sind in Anhang A.4 auf Seite 55 beschrieben. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

16 bei Halbierung von h_n , falls auf $V_{n-3,n-3}$ extrapoliert und um einen Faktor 4 in Abbildung 2.9, falls auf $V_{n,n}$ extrapoliert wird.

Das obige lineare Gleichungssystem ist unterbesetzt, eine Möglichkeit die zusätzlichen Freiheitsgrade los zu werden, wäre noch zusätzliche Gleichungen z. B. die Symmetriebedingungen $a_1 = a_3$ und $a_4 = a_5$ hinzuzufügen, eine andere Möglichkeit wäre weitere Fehlerterme zu eliminieren. In höheren Dimensionen verschärft sich das Problem noch. [HH14]

Ein weiteres Problem bei der Verwendung aller Teillösungen ist im Falle einer Truncated Combination Technique von der Form $f_n^{\vec{r}}$, dass auf $\Omega_{\vec{r}}$, und bei der klassischen Dünngitter-Kombinationstechnik sogar auf $\Omega_{(1,1,\dots,1)}$ extrapoliert werden würde, so dass im Allgemeinen für alle anderen Punkte keine Auslöschung der Fehlerterme eintritt, und der Aufwand für eine verbesserte Konvergenz dadurch ins Leere läuft, dass sie an den meisten Punkten nicht gilt.

Deswegen versuchen wir wie folgt mit $(d+1)$ Teillösungen auszukommen:

Sei nun $L_n = \{ \vec{l}_{n_0}, \vec{l}_{n_1}, \dots, \vec{l}_{n_d} \} \subseteq \mathbb{N}_0^d$ mit $\vec{l}_{n_0} = (n, n, \dots, n)$ und für $k \geq 1$ sei $\vec{l}_{n_k} = \vec{l}_{n_0} + \vec{e}_k$, wobei $\vec{e}_k \in \mathbb{N}_0^d \wedge \|\vec{e}_k\|_1 = 1 \wedge e_{kk} = 1$, d. h. e_k ist der Einheitsvektor entlang der k -ten Achse, gilt.

Wir wollen nun die Koeffizienten $a_{\vec{l}}$, $\vec{l} \in L_n$ so bestimmen das folgende Gleichung erfüllt ist:

$$\sum_{\vec{l} \in L_n} a_{\vec{l}} f_{\vec{l}}(\vec{x}) = f + \mathcal{O}(h_n^{2p}) + \sum_{j=1}^d 0e^j$$

Nun können wir ein lineares Gleichungssystem aufstellen.

2. Definitionen, Versuchsaufbau und Ergebnisse

$$\begin{aligned}
 & \left(\begin{array}{cccccc|c} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ & \frac{1}{2^p} & & & & & 0 \\ & & \frac{1}{2^p} & & & & 0 \\ & & & \ddots & & & \vdots \\ & 1 & & & \frac{1}{2^p} & & 0 \\ & & & & & \frac{1}{2^p} & 0 \end{array} \right) \Rightarrow \left(\begin{array}{cccccc|c} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ & \frac{1-2^p}{2^p} & & & & & -1 \\ & & \frac{1-2^p}{2^p} & & & & -1 \\ & & & \ddots & & & \vdots \\ & 0 & & & \frac{1-2^p}{2^p} & & -1 \\ & & & & & \frac{1-2^p}{2^p} & -1 \end{array} \right) \\
 & \Rightarrow \left(\begin{array}{cccccc|c} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ & 1 & & & & & \frac{2^p}{2^p-1} \\ & & 1 & & 0 & & \frac{2^p}{2^p-1} \\ & & & \ddots & & & \vdots \\ & 0 & & & 1 & & \frac{2^p}{2^p-1} \\ & & & & & 1 & \frac{2^p}{2^p-1} \end{array} \right)
 \end{aligned}$$

Es gilt also $a_{l_{n_0}} \rightarrow 1 - d * \frac{2^p}{2^p-1}$ und $a_{l_{n_k}} \rightarrow \frac{2^p}{2^p-1}$ für $1 \leq k \leq d$. So bekommen wir z. B. für $p=2$ in 2D die Kombination

$$\frac{4}{3}P_{n,n}f_{n+1,n} + \frac{4}{3}P_{n,n}f_{n,n+1} - \frac{5}{3}f_{n,n} \quad (2.16)$$

und in 3D die Kombination

$$\frac{4}{3}P_{n,n,n}f_{n+1,n,n} + \frac{4}{3}P_{n,n,n}f_{n,n+1,n} + \frac{4}{3}P_{n,n,n}f_{n,n,n+1} - 3f_{n,n,n} \quad (2.17)$$

Der Vorteil, dieser Auswahl von Teillösungen für die Richardson-Extrapolation ist, dass sie für jedes beliebige d durchführbar ist. Fallen bei einer Kombinationstechnik Lösungen aus, so kann Gleichung 2.16 bzw. 2.17 verschoben werden [HH14, vgl.], so dass die ausgefallenen Teillösungen nicht verwendet werden. Beispiele sind Fall 6, 7 und 8 aus Abbildung 2.8 oder die Fälle 1 bis 4 aus Abbildung A.7. Wir haben genau $(d+1)$ Teillösungen für $(d+1)$ Gleichungen verwendet und das Gleichungssystem ist eindeutig lösbar. Im Gegensatz zur klassischen Kombinationstechnik ersparen wir uns den Aufwand die Teillösung $f_{(n+1,n+1,\dots,n+1)}$ mit $(2^{n+1} + 1)^d$ Unbekannten zu lösen und ersetzen sie durch d Teillösungen mit jeweils nur $(2^n + 1)^{d-1}(2^{n+1} + 1)$ Unbekannten.

Sei a_n die Anzahl der Punkte der klassischen Richardson-Extrapolation und b_n die obiger Kombination dann gilt für den Grenzwert des Quotienten:

$$\begin{aligned}
 \lim_{n \rightarrow \infty} \frac{a_n}{b_n} &= \lim_{n \rightarrow \infty} \frac{(2^{n+1} + 1)^d + (2^n + 1)^d}{d(2^n + 1)^{d-1}(2^{n+1} + 1) + (2^n + 1)^d} = \lim_{n \rightarrow \infty} \frac{(2^{n+1} + 1)^d + (2^n + 1)^d}{(2^n + 1)^{d-1}(d(2^{n+1} + 1) + (2^n + 1))} \\
 &\geq \lim_{n \rightarrow \infty} \frac{(2^{n+1} + 1)^d}{(2^n + 1)^{d-1}(d+1)(2^{n+1} + 1)} = \lim_{n \rightarrow \infty} \frac{(2^{n+1} + 1)^{d-1}}{(2^n + 1)^{d-1}(d+1)} = \lim_{n \rightarrow \infty} \frac{1}{d+1} \left(\frac{(2^{n+1} + 1)}{(2^n + 1)} \right)^{d-1} \\
 &= \frac{2^{d-1}}{d+1}
 \end{aligned}$$

In den Abbildungen 2.10, 2.11, A.3, A.4, A.7, und A.8 sind die weiteren Ergebnisse für die Richardson-Extrapolation in 2D und 3D für Advektions- und Laplace-Gleichung dargestellt.

2.7. Verknüpfung von einfacher Kombinationstechnik und Richardson-Extrapolation

[HH14] schlägt eine Verknüpfung der multivariaten Richardson-Extrapolation mit der Kombinationstechnik vor.

Sei $\tilde{f}_{n-2,n-2}^2$ gegeben $p = 1$ und $f_{n-2,n-2}^e = -2P_{n-2,n-2}f_{n-2,n} + 5P_{n-2,n-2}f_{n-1,n-1} - 2P_{n-2,n-2}f_{n,n-2}$ eine Extrapolation auf $V_{n-2,n-2}$, so liefert die Kombination

$$P_L f_{n-2,n-2}^e + (\tilde{f}_{n-2,n-2}^2 - P_{n-2,n-2} \tilde{f}_{n-2,n-2})$$

, wobei $P_L f_{n-2,n-2}^e$ die bilineare Interpolation auf den Dünngitterraum von $\tilde{f}_{n-2,n-2}^2$ ist, mit $L = \{(l_1, l_2) \in \mathbb{N}^2 \mid l_1 + l_2 \leq 2n - 2 \wedge l_1, l_2 \leq n\}$, eine Approximation zweiter Ordnung für $V_{n-2,n-2}$ und eine Approximation erster Ordnung auf

$$\left(\bigoplus_{l_2 \leq n-2} W_{n,l_2} \right) \oplus \left(\bigoplus_{l_2 \leq n-1} W_{n-1,l_2} \right) \oplus \left(\bigoplus_{l_1 \leq n-2} W_{l_1,n} \right) \oplus \left(\bigoplus_{l_1 \leq n-2} W_{l_1,n-1} \right)$$

Fallen Teillösungen aus so verwendet [HH14] im Fehlerfall folgende Kombination

$$P_L g_{n-2,n-2}^e + (g_L - P_{n-2,n-2} g_L)$$

wobei $g_{n-2,n-2}^e$ eine Extrapolation von erfolgreich berechneten Teillösungen auf $V_{n-2,n-2}$ und g_L eine Kombination von erfolgreich berechneten Teillösungen ist.[HH14]

In den Abbildungen 2.10 und 2.11 sind für die Advektionsgleichung in 2D die Fehler der Vollgitterlösung $f_{n,n}$, als auch die der Richardson-Extrapolation und der kombinierten Technik für einen Ausfall von Teillösungen aus $\tilde{f}_{n-2,n-2}^2$ bzw. $\tilde{f}_{n-3,n-3}^3$ zu sehen. Der Fehler der kombinierten Technik wird immer in dem Dünngitterraum ausgerechnet, welcher von den Funktionsräumen der noch intakten Teillösungen aufgespannt wird. Wir sehen, dass der Fehler geringfügig schlechter als der, der Richardson-Extrapolation ist. Dies entspricht nicht ganz den Ergebnissen aus [HH14] in welchen der Fehler der kombinierten Technik geringfügig besser als der, der Richardson-Extrapolation ist.

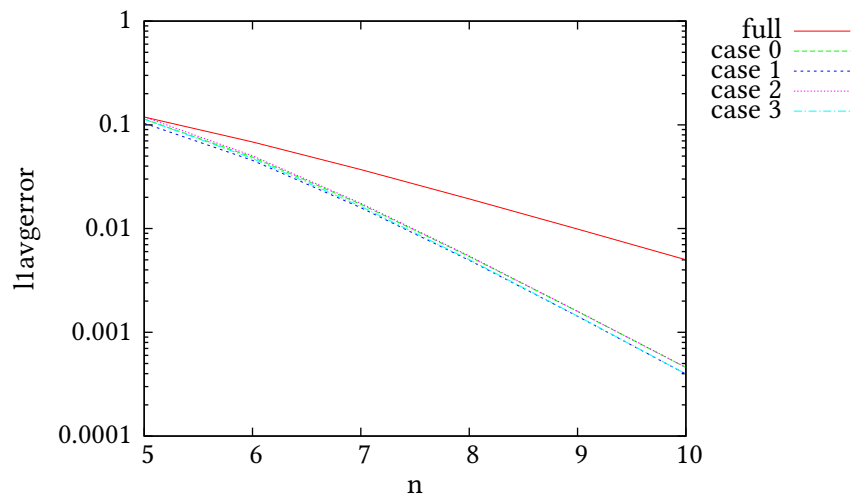


Abbildung 2.10.: Dargestellt sind die Fehler bei der Lösung der Advektionsgleichung für Linearkombinationen nach einem Ausfall von Teillösungen aus $\tilde{f}_{n-2,n-2}^2$. Die einzelnen Fälle sind in Anhang A.5 auf Seite 56 beschrieben. Fall 0 und Fall 1 sind Richardson-Extrapolationen auf $V_{n-2,n-2}$ und ihr Fehler wird auf $V_{n-2,n-2}$ berechnet. Fall 2 und 3 entsprechen der Kombination der Richardson-Extrapolation auf $V_{n-2,n-2}$ und der einfachen Linearkombination und ihre Fehler werden auf dem Raum V_L , mit L wie im Anhang beschrieben, berechnet. Der Fehler der Vollgitterlösung $f_{n,n}$ wird auf $V_{n,n}$ berechnet.

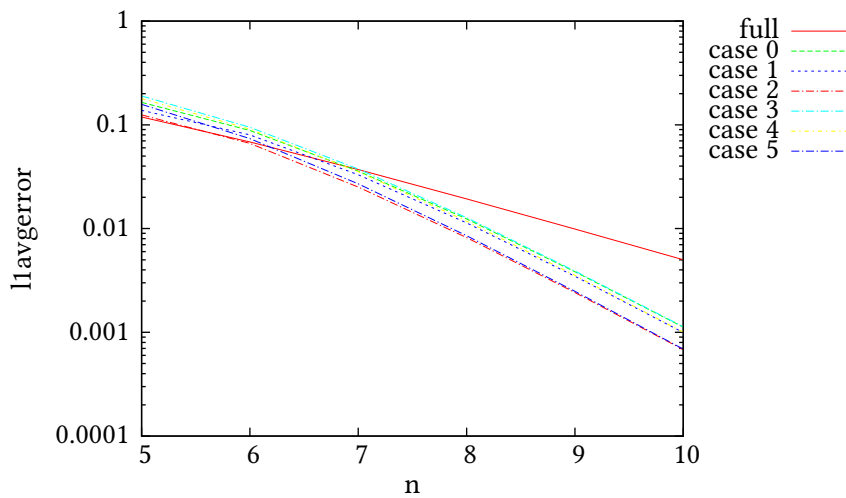


Abbildung 2.11.: Dargestellt sind die Fehler bei der Lösung der Advektionsgleichung für Linearkombinationen nach einem Ausfall von Teillösungen aus $\tilde{f}_{n-3,n-3}^3$. Die einzelnen Fälle sind in Anhang A.6 auf Seite 57 beschrieben. Fall 0, 1 und 2 sind Richardson-Extrapolationen auf $V_{n-3,n-3}$ und ihr Fehler wird auf $V_{n-3,n-3}$ berechnet. Fall 3, 4 und 5 entsprechen der Kombination der Richardson-Extrapolation auf $V_{n-3,n-3}$ und der einfachen Linearkombination und ihre Fehler werden auf dem Raum V_L , mit L wie im Anhang beschrieben, berechnet. Der Fehler der Vollgitterlösung $f_{n,n}$ wird auf $V_{n,n}$ berechnet.

3. Implementierung fehlertoleranter Verfahren in SG++ und Ergebnisse

In diesem Kapitel soll die Implementierung von Verfahren zur Berechnung einer neuen Linearkombination, falls Teillösungen bei der Berechnung von T-CTs der Form $f_n^{\vec{T}}$ ausfallen, und die Ergebnisse für deren Fehler dargestellt werden.

In Abbildung 3.1 auf der nächsten Seite ist die Klasse CombiFaultRecoveryScheme dargestellt. Diese wurde von der Klasse CombiSchemeBasis abgeleitet, welche wie in Abschnitt 2.4 bereits beschrieben, die Koeffizienten und Levelvektoren einer Linearkombination kapselt. CombiFaultRecoveryScheme hat zwei Konstruktoren. Der erste erzeugt T-CTs der Form

$$\sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q} \sum_{\substack{\|\vec{T}\|_1 = \text{levelSumme} - q \\ \text{minLevels} \leq \vec{T}}} f_{\vec{T}}(\vec{x})$$

Wollen wir also f_n^c nach Definition 2.1.6 erhalten, so müssen wir den Konstruktor also durch „CombiFaultRecoveryScheme (dim, n+d-1, $\vec{1}$)“ aufrufen.

Der zweite Konstruktor erzeugt T-CTs der Form $\tilde{f}_{\vec{l}_n}^t$ mit $\vec{l}_n = (n-t, n-t, \dots, n-t)$. Da die Klasse von CombiSchemeBasis abgeleitet ist, lässt sich jede beliebige andere Linearkombination über die Methoden updateScheme und SetCoef setzen.

Fallen nun Teillösungen aus, so können über die Methoden faultRecoveryFast, faultRecoverySlow, faultRecoveryBlock und faultRecoveryRichardson, neue Kombinationen berechnet werden. Dazu müssen allen Methoden über den Parameter failedGrids die Indizes der Levelvektoren in der Membervariablen levels_vector_ übergeben werden, deren zugehörigen Teillösungen ausgefallen sind. Die Methoden berechnen, dann die neue Linearkombination und diese resultiert in einer entsprechenden Anpassung der beiden Attribute levels_vector_ und coefficients_.

Da die Dünngitter-Kombinationstechnik f_n^c aus $\mathcal{O}(n^{d-1})$ Teillösungen besteht, gibt es für jede der Methoden $\mathcal{O}(2^{cn^{d-1}})$ verschiedene Fälle, für eine Konstante $c > 0$. Diese können selbst für kleine n nicht mehr alle durchprobiert werden. Statt dessen werden, wie in Algorithmus 3.1 dargestellt, für jede Levelsumme zwischen 9 und 13, bei einem Ausfall von ein bis vier zufällig ausgewählten Teillösungen, für jedes Verfahren die Fehler für eine gewisse Anzahl an Samples berechnet, und deren Durchschnitt gebildet.

faultRecoveryfast berechnet die neue Linearkombination durch das Prinzip der Inklusion und Exklusion wie bereits in Abschnitt 2.3 beschrieben. Der Fehler für die Dünngitter-Kombinationstechnik liegt, falls die exakte Lösung f genügend glatt, in $\mathcal{O}(h_n^2 n^{d-1})$ [Pfl10]. Die Ergebnisse für f_n^c und die

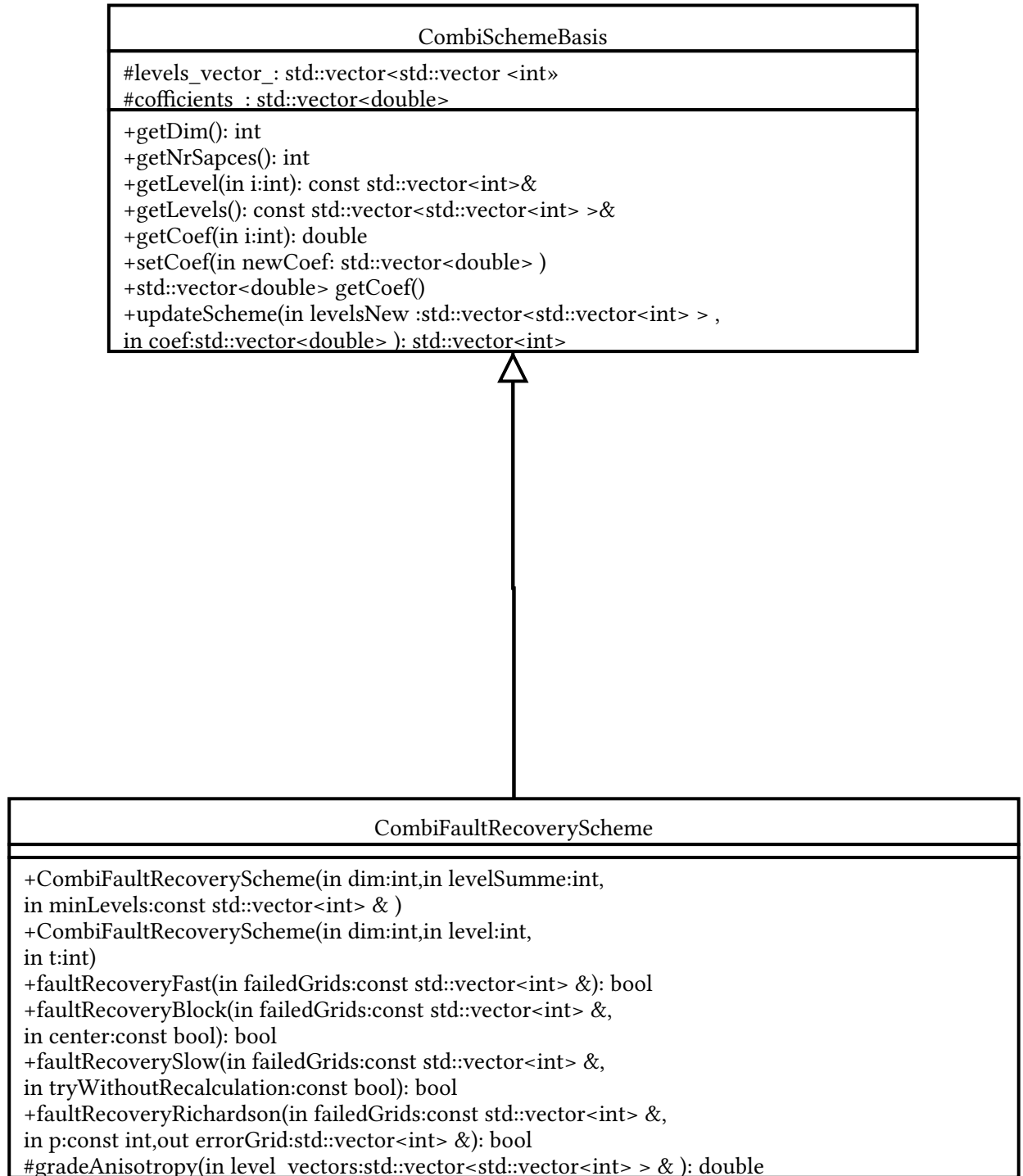


Abbildung 3.1.: CombiFaultRecoverScheme

Algorithmus 3.1 Algorithmus zur Messung der Fehler der implementierten Methoden

```
procedure TEST
  solutions  $\leftarrow$  calculateAllSolutionsWithLevelSumFrom9To13()
  for all  $levelSum \in \{9, \dots, 13\}$  do
    for all  $amountFaultyGrids \in \{1, \dots, 4\}$  do
      for all  $m \in METHODS$  do
        error( $m$ )  $\leftarrow$  0
        successfullyCalculated( $m$ )  $\leftarrow$  0
      end for
      for all  $samples \in \{1, \dots, levelSum * amountFaultyGrids\}$  do
        failedGrids  $\leftarrow$  getRandomFaultyGrids( $levelSum, amountFaultyGrids$ )
        for all  $m \in METHODS$  do
          CombiFaultRecoveryScheme  $cFRS(2, levelsum, \{1, 1\})$ 
          success  $\leftarrow$  cFRS.calcNewSchemeWithMethod(...)
          if success then
            successfullyCalculated( $m$ )  $\leftarrow$  successfullyCalculated( $m$ ) + 1
            error( $m$ )  $\leftarrow$  calcErrorFromCombinationWithMethod(...) + error( $m$ )
          end if
        end for
      end for
      for all  $m \in METHODS$  do
        error( $m$ )  $\leftarrow$   $\frac{error(m)}{successfullyCalculated(m)}$ 
        SAVEToFile( $levelSum, m, error(m)$ )
      end for
    end for
  end for
end procedure
```

entsprechenden Linearkombination bei Ausfällen von Teillösungen sind in Abbildung 3.2 zu sehen und verhalten sich so wie wir es für eine Konvergenz in $\mathcal{O}(h_n^2 n^{d-1})$ erwarten würden.

Hierbei fällt auf, dass bei Ausfall einer Teillösung nicht alle Teillösungen aus den zugehörigen feineren Funktionsräumen verworfen werden müssen. Dies nützt die Methode `faultRecoverySlow` aus. In dem Beispiel der Dünngitter-Kombinationstechnik f_4^c , wie in Abschnitt 2.3 beschrieben, hätten bei einem Ausfall der Teillösung $f_{2,3}$ das Entfernen von einer der beiden Teillösungen $f_{3,3}$ und $f_{2,4}$ ausgereicht. Da wir 2 Teillösungen verworfen hatten, würden wir in diesem Fall einen größeren Fehler der neu berechneten Kombination erwarten, als für die neu berechneten Kombinationen, falls wir nur eine Teillösung verwerfen würden. So bemerkt [HH12], dass es im Allgemeinen, in d Dimensionen, passieren kann, dass viele hundert Teillösungen zu viel verworfen werden und sich das Problem dementsprechend verschärft.

Sei nun eine Kombination $f_L = \sum_{\vec{l} \in L} c_{\vec{l}} f_{\vec{l}}$ gegeben und sei $\Lambda \subset L$ die Menge der Levelvektoren der ausgefallenen Teillösungen. `faultRecoverySlow` berechnet nun die Menge $K = \{ \vec{l} \in L \mid \exists \vec{\lambda} \in \Lambda : \vec{\lambda} \leq \vec{l} \}$ der Levelvektoren aller Teillösungen, aus den zugehörigen feineren

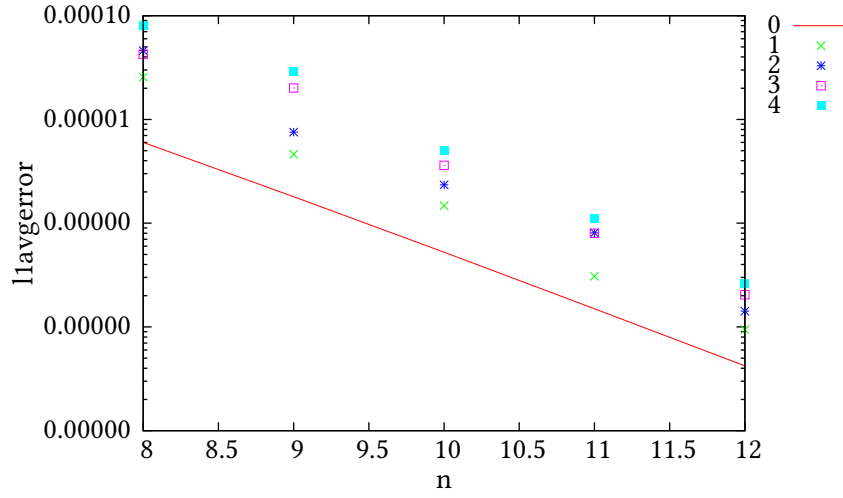


Abbildung 3.2.: Das Laplace Problem wird in 2D für die Dünngitter-Kombinationstechnik f_n^c gelöst. Die Ergebnisse für die Fehler, der mit der Methode faultRecoveryFast berechneten Linearkombinationen, werden für einen Ausfall von bis zu vier Teillösungen dargestellt. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

Funktionsräumen, und versucht so wenig Teillösungen zu verwerfen wie möglich. Dazu wird für alle $M \in \mathcal{P}(K)$, eine neue Linearkombination durch $P_{\downarrow(L \setminus M)}$ berechnet und überprüft ob die Bedingung $\forall \vec{\lambda} \in \Lambda : c_{\vec{\lambda}} = 0$ erfüllt ist, also keine der ausgefallenen Teillösungen in der neu berechneten Linearkombination verwendet wird. Unter diesen Lösungen wird diejenige ausgewählt, welche eine minimale Anzahl an Teillösungen verwirft. Gibt es mehrere derartige Linearkombinationen wird diejenige ausgewählt, welche möglichst „isotrop“ ist. Sei eine Linearkombination $f_G = \sum_{\vec{g} \in G} c_{\vec{g}} f_{\vec{g}}$ gegeben, dann wird die Bewertungsfunktion $\tilde{\phi}(f_G)$ minimiert, welche durch

$$\tilde{\phi}(f_G) := \frac{1}{\sum_{\vec{g} \in G} |c_{\vec{g}}|} \sum_{\vec{g} \in G} |c_{\vec{g}}| \phi(f_{\vec{g}})$$

mit

$$\phi(f_{\vec{g}}) := \frac{1}{\|g\|_1} \sqrt{\sum_{i \in \{1, \dots, d\}} \left(g_i - \frac{\|g\|_1}{d} \right)^2} \quad (3.1)$$

gegeben ist. Ist die Bewertungsfunktion für mehrere Linearkombination gleich wird eine beliebige ausgewählt.

Wir berechnen also für jeden Levelvektor $\vec{g} \in G$, als Vektor in \mathbb{R}^n aufgefasst, einen gewichteten euklidischen Abstand $\phi(\vec{g})$ zum Vektor $\left(\frac{\|g\|_1}{d}, \frac{\|g\|_1}{d}, \dots, \frac{\|g\|_1}{d} \right)$ und haben damit ein Metrik für die

Anisotropie des Gitters $\Omega_{\vec{g}}$. Die Idee dahinter, ist dass Teillösungen, zu den zugehörigen stark anisotropen Gittern, für dieselben relativ schlechte Approximationen liefern, vgl. dazu etwa Gleichung 2.7, 2.13, 2.14 oder etwa [BP12]. Diese werden durch die Bewertungsfunktion „bestraft“. Die Gewichtung der einzelnen Teillösungen wird auch in der Bewertungsfunktion berücksichtigt und damit nicht automatisch Linearkombinationen mit vielen Teillösungen verworfen werden, wird ein gewichtetes arithmetisches Mittel gebildet.

Hat der Parameter `tryWithoutRecalculation` den Wert `true`, wird versucht eine Kombination, nach obigen Kriterien, zu finden für die keine neuen Teillösungen berechnet werden müssen. Wird eine solche nicht gefunden ist die Lösung dieselbe, wie für den Fall, dass für `tryWithoutRecalculation` `false` übergeben worden wäre.

Hier sei gleich erwähnt, dass der Aufwand für die Berechnung von `faultRecoverySlow` auch wirklich langsam ist. Sei d die Dimension unseres Problems. Die klassische Dünngitter-Kombinationstechnik $f_n^c = f_n^{\vec{1}} = \sum_{\vec{l}_n \in L_n} c_{\vec{l}_n} f_{\vec{l}_n}$ mit $n \in \mathbb{N}$ bestehen z. B. aus Levelvektoren \vec{l}_n mit $n \leq \|\vec{l}_n\|_1 \leq n + d - 1$. Fällt also eine Teillösung $f_{\vec{m}}$ mit $\|\vec{m}\|_1 = n \wedge \vec{m} \in L_n$ aus, so gilt für die Menge $K_{\vec{m}} = \{ f_{\vec{l}_n} \mid \vec{l}_n \in L_n \wedge \vec{m} \leq \vec{l}_n \}$ der Teillösungen, aus den zugehörigen feineren Funktionsräumen,

$$|K_{\vec{m}}| = \sum_{q=0}^{d-1} \binom{d+q-1}{q} = \binom{2d-1}{d-1} \quad (3.2)$$

Ist die Wahrscheinlichkeit für den Ausfall der einzelnen Teillösungen $f_{\vec{l}_n}$ mit $\vec{l}_n \in L_n$ und $n \in \mathbb{N}$ unabhängig und zum Parameter $q \in [0, 1]$ Bernoulli verteilt, so gilt für den Erwartungswert der Anzahl der ausgefallenen Teillösungen $E(X) = q |L_n|$. Allerdings ist dies noch optimistisch geschätzt. Denn eigentlich hängt die Ausfallwahrscheinlichkeit auf einem Knoten von der Zeitdauer der Berechnung ab und mit n wächst auch der Aufwand die einzelnen Teillösungen $f_{\vec{l}_n}, \vec{l}_n \in L_n$ zu berechnen, so dass die Ausfallwahrscheinlichkeit q unter anderem von n abhängig ist. Dies bedeutet, dass wir eventuell mit einer oberen Schranke für den Aufwand im **average case (!)** von `faultRecoverySlow` rechnen müssen, welche nach unten durch $\Omega(2^{\lfloor K|cn^{d-1} \rfloor})$ für eine Konstante c abgeschätzt werden kann. Aufgrund der Vollständigkeit wurde dieses Verfahren dennoch implementiert und dessen Ergebnisse in den Abbildungen 3.3 und 3.4 dargestellt.

Die Ergebnisse legen ebenfalls eine Konvergenz in $\mathcal{O}(h_n^2 n^{d-1})$ nahe. Beide Ergebnisse sind nahezu identisch. Dies liegt daran, dass das Verfahren der Inklusion und Exklusion auf Gittern, wie in Abschnitt 2.3 beschrieben, nach einem Ausfall von Teillösungen aus f_n^c in 2D nur dann keine neuen Levelvektoren produziert, falls sich alle verworfenen Teillösungen an den „Enden“ der „Diagonalen“, vgl. dazu etwa Abbildung 2.2, befinden und die Wahrscheinlichkeit dafür mit wachsendem n fällt. Im Vergleich zu `faultRecoveryFast` sind die Ergebnisse leicht besser, dafür ist der Aufwand aber auch größer. Beide langsamen Verfahren benötigten in 2D, genauso wie `faultRecoverFast`, für die Berechnung der Fehlerkombination in allen Fällen unter Verwendung eines Prozesses auf einem Rechner mit „Intel Xeon E7540“ Prozessoren, eine durchschnittliche CPU Time von weit unter einer Sekunde. Dabei wurde die CPU Time mit „`clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &ts)`“ aus der C++ Standardbibliothek gemessen. Für `faultRecoveryFast` war dies auch in 3D der Fall, während für `faultRecoverySlow` in 3D bereits für die Levelsumme 9 und vier ausgefallenen Teillösungen eine

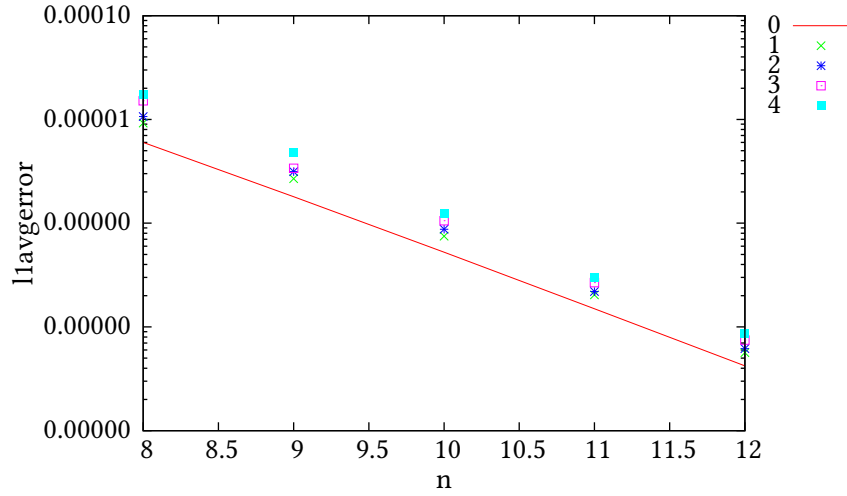


Abbildung 3.3.: Das Laplace Problem wird in 2D für die Dünngitter-Kombinationstechnik f_n^c gelöst. Für einen Ausfall von bis zu vier Teillösungen werden mit der Methode faultRecoverySlow neue Linearkombinationen berechnet. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

CPU Time in der Größenordnung von mehreren Minuten gemessen wurde, so dass diese Messungen abgebrochen wurden.

Die Methode faultRecoveryRichardson bekommt neben dem Parameter failedGrids, noch die Ordnung der Diskretisierung über den Parameter p übergeben und berechnet eine multivariate Richardson-Extrapolation aus $d+1$ Teillösungen wie in Abschnitt 2.6 beschrieben. Sei also $L = \{ \vec{l}_0, \vec{l}_1, \dots, \vec{l}_d \} \subseteq \mathbb{N}_0^d$ und für $k \geq 1$ sei $\vec{l}_k = \vec{l}_0 + \vec{e}_k$, wobei $\vec{e}_k \in \mathbb{N}_0^d \wedge \|\vec{e}_k\|_1 = 1 \wedge e_{kk} = 1$, d. h. e_k ist der Einheitsvektor entlang der k -ten Achse, gilt. Nun wird durch Bestimmung von \vec{l}_0 die Kombinationstechnik so gewählt, dass keine ausgefallene Teillösung für die Extrapolation verwendet wird. Es wird also eine Art „Referenz-Linearkombination“ solange verschoben, bis eine passende Lösung gefunden ist. Unter denjenigen Linearkombinationen, die diese Eigenschaft erfüllen, werden diejenigen gewählt, welche $\|\vec{l}_0\|_1$ maximieren und unter diesen wiederum eine derjenigen ausgewählt welche Gleichung 3.1, wie in Algorithmus 3.2 dargestellt, minimiert. Im Parameter errorGrid wird \vec{l}_0 , d. h. der Levelvektor für das Gitter zurück gegeben auf dem die Fehlerauslöschung der ersten Terme der Fehlerexpansion auftritt.

Bei der Vollgitterlösung $f_{n,n}$ oder T-CTs der Form $\tilde{f}_{(n-t,n-t)}^t$ würden wir für die klassische bzw. multivariate Richardson-Extrapolation, welche für eine Teillösung $f_{(l_1,l_2)}$ die Fehlerterme die von $h_{l_1}^p$ und $h_{l_2}^p$ abhängig sind auslöscht, einen Fehler in $\mathcal{O}(h_n^{2p})$ erwarten. Für die, wie oben beschrieben, berechneten Richardson-Extrapolationen, nach einem Ausfall von Teillösungen der Dünngitter-Kombinationstechnik f_n^c würden wir also dementsprechend erwarten, dass der durchschnittliche Fehler über alle Samples in $\mathcal{O}(h_n^p)$ liegt.

Die Ergebnisse von faultRecoveryRichardson sind in der Abbildung 3.5 dargestellt. Die Ergebnisse entsprechen im Groben dem Erwarteten, wobei im Plot auch Sprünge zu erkennen sind.

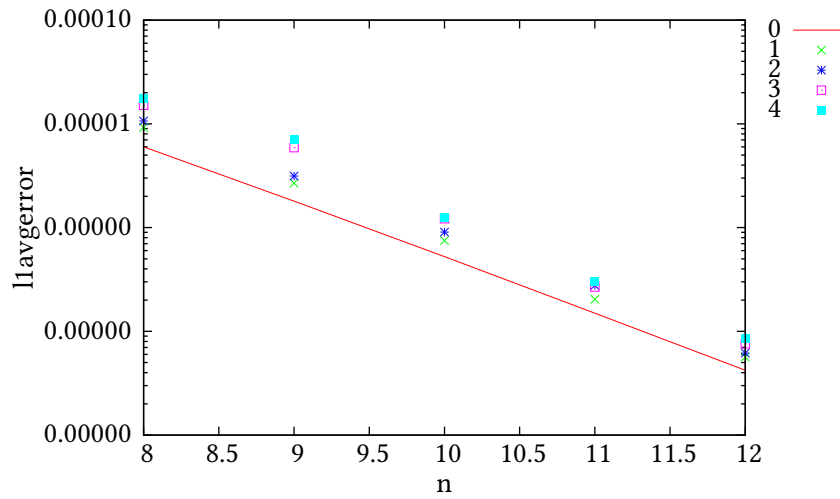


Abbildung 3.4.: Das Laplace Problem wird in 2D für die Dünngitter-Kombinationstechnik f_n^c gelöst. Für einen Ausfall von bis zu vier Teillösungen werden mit der Methode `faultRecoverySlow` neue Linearkombinationen berechnet. Dabei wird versucht **ohne neue Levelvektoren** auszukommen. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

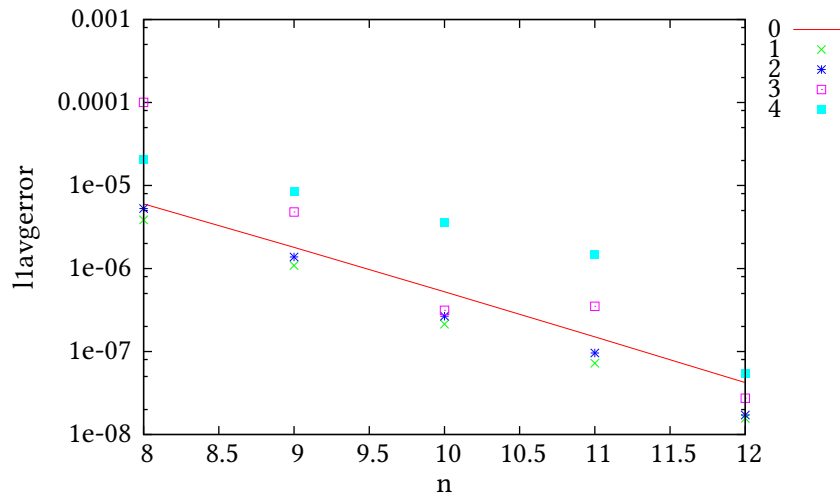


Abbildung 3.5.: Das Laplace Problem wird in 2D für die Dünngitter-Kombinationstechnik f_n^c gelöst. Für einen Ausfall von bis zu vier Teillösungen werden mit der Methode `faultRecoveryRichardson` neue Linearkombinationen berechnet. Für alle Fälle, d. h. für jedes Sample wurde der Fehler auf dem Funktionsraum ausgerechnet, in welches extrapoliert wurde.

Algorithmus 3.2 Bestimmung einer Richardson-Extrapolation nach einem Ausfall von Teillösungen

```

procedure FAULTRECOVERYRICHARDSON(failedGrids, p, out errorGrid)
    levelsum  $\leftarrow n$ 
    success  $\leftarrow false$ 
    while levelsum  $\leq n + d - 1$  do
        minGrade  $\leftarrow$  DOUBLE_MAX
        grade  $\leftarrow$  DOUBLE_MAX
        for all  $\vec{l}_0 \in levels\_vector\_do$ 
            if  $\|\vec{l}_0\|_1 == levelsum$  then
                for all  $k \in \{1, \dots, d\}$  do
                     $\vec{l}_k \leftarrow \vec{l}_0 + e_k$ 
                end for
                valid  $\leftarrow$  isValid( $\vec{l}_0, \dots, \vec{l}_d$ ) // Existieren alle Teillösungen ?
                if valid then
                    grade  $\leftarrow$  getGrade( $\vec{l}_0, \dots, \vec{l}_d$ )
                    if grade < minGrade then
                        minGrade  $\leftarrow$  grade
                        result  $\leftarrow (\vec{l}_0, \dots, \vec{l}_d)$ 
                        success  $\leftarrow true$ 
                    end if
                end if
            end if
        end for
        levelsum  $\leftarrow levelsum + 1$ 
    end while
    updateCombiFaultRecoveryScheme(result,  $(1 - d * \frac{2^p}{2^p-1}, \frac{2^p}{2^p-1}, \dots, \frac{2^p}{2^p-1})$ )
    errorGrid  $\leftarrow \vec{l}_0$ 
    return success
end procedure

```

Als letztes wurde noch ein Verfahren implementiert, dass, ähnlich wie bei der Methode faultRecoveryRichardson, durch Verschiebung von T-CTs der Form $\tilde{f}_{l_n}^t$ mit $\vec{l}_n = (n - t, n - t, \dots, n - t)$ eine Linearkombination im Fehlerfall berechnet. Dabei, wird zuerst t maximiert und dann Gleichung 3.1 minimiert. Über den Parameter center kann eine zulässige Linearkombination so eingeschränkt werden, dass sie zentriert wird, d. h. es werden für jede Levelsumme die Menge der intakten Teillösungen bestimmt welche die kleinste Anisotropie, nach Gleichung 3.1, aufweisen und mindestens eine dieser Teillösungen muss in der neu berechneten Linearkombination enthalten sein.

Die Ergebnisse sind in den Abbildungen 3.6 und 3.7 dargestellt. Diese Methode liefert im Vergleich zu den anderen Verfahren eine wesentlich schlechtere Konvergenz, da zu viele Gitter verworfen werden. Zudem ist die Komplexität ähnlich hoch wie im Falle von faultRecoverySlow und war bereits in 3D, deswegen nicht mehr anwendbar.

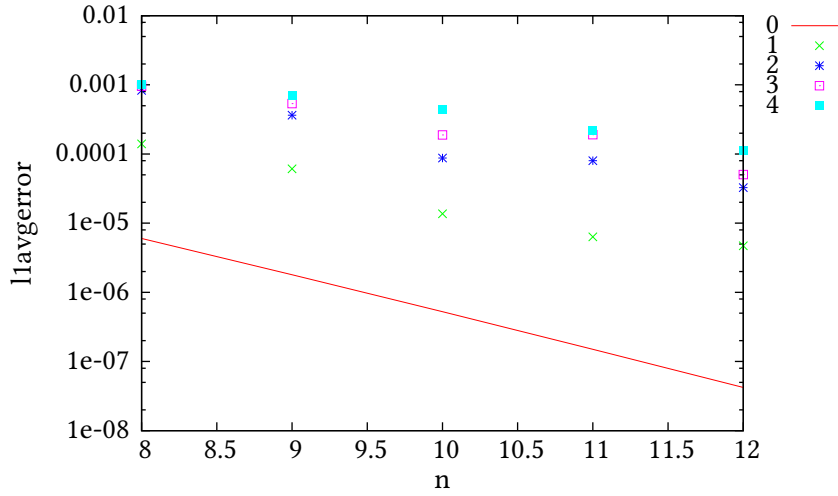


Abbildung 3.6.: Das Laplace Problem wird in 2D für die Dünngitter-Kombinationstechnik f_n^c gelöst. Für einen Ausfall von bis zu vier Teillösungen werden mit der Methode `faultRecoveryBlock`, `center=false` neue Linearkombinationen berechnet. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

Die Rückgabewerte der Methoden geben an, ob die neue Kombination erfolgreich berechnet werden konnte. So kann z. B. bei `faultRecoveryRichardson` der Fall eintreten, dass keine zulässige Translation der „Referenz-Linearkombination“ gefunden werden konnte oder bei `faultRecoveryFast`, dass nach dem Verwerfen feinerer Teillösungen keine mehr übrig sind. Ist dies der Fall, so werden die Membervariablen für Levelvektor und Koeffizientenvektor nicht modifiziert.

3.1. Ansätze zur Verbesserung der Verfahren

In diesem Abschnitt sollen einige Vorschläge zur Verbesserung obiger Verfahren gemacht werden, welche aufgrund der beschränkten Bearbeitungszeit dieser Arbeit nicht untersucht wurden. Fallen bei einer Kombinationstechnik $f_L = \sum_{\vec{l} \in L} c_{\vec{l}} f_{\vec{l}}$ die Teillösungen $\vec{\lambda} \in \Lambda \subseteq L$ aus, so müssen wie bereits besprochen nicht alle feineren Teillösungen $f_{\vec{k}}$ mit $\vec{k} \in K = \{ \vec{l} \in L \mid \exists \vec{\lambda} \in \Lambda : \vec{\lambda} \leq \vec{l} \}$ verworfen werden. Statt wie in Kapitel 3 für alle Mengen $M \in \mathcal{P}(K)$ die neue Linearkombination durch $P_{\downarrow(L \setminus M)}$ zu berechnen und zu überprüfen ob die Bedingung $\forall \vec{\lambda} \in \Lambda : c_{\vec{\lambda}} = 0$ erfüllt ist, kann der Suchraum auch eingeschränkt werden, und dadurch, statt einer exponentiellen, wieder eine polynomielle Komplexität in n erreicht werden. Eine Möglichkeit wäre z. B. solange sukzessive Teillösungen zu entfernen bis die Bedingung $\forall \vec{\lambda} \in \Lambda : c_{\vec{\lambda}} = 0$ erfüllt ist. Wir hätten dann also eine Menge $\{M_1, M_2, \dots, M_t\}$ mit $M_1 \subsetneq M_2 \subsetneq \dots \subsetneq M_t \subseteq K$ und $M_1 = \{ \vec{k}_1 \}$ für ein $\vec{k}_1 \in K$, so dass für alle $1 \leq i < t$ gilt: $M_{i+1} = M_i \cup \{ \vec{k}_{i+1} \}$ mit $\vec{k}_{i+1} \in K \setminus \bigcup_{j=1}^i \{ \vec{k}_j \}$. Für die Wahl der Teillösung \vec{k}_i welche im i -ten Schritt entfernt wird, sind prinzipiell auch Heuristiken denkbar. Die Anzahl der Schritte ist dabei für eine Dünngitter-Kombinationstechnik f_n^c durch $\mathcal{O}(n^{d-1})$ beschränkt und unter Berücksichtigung des Aufwandes für die Berechnung der Koeffizienten wie in Abschnitt

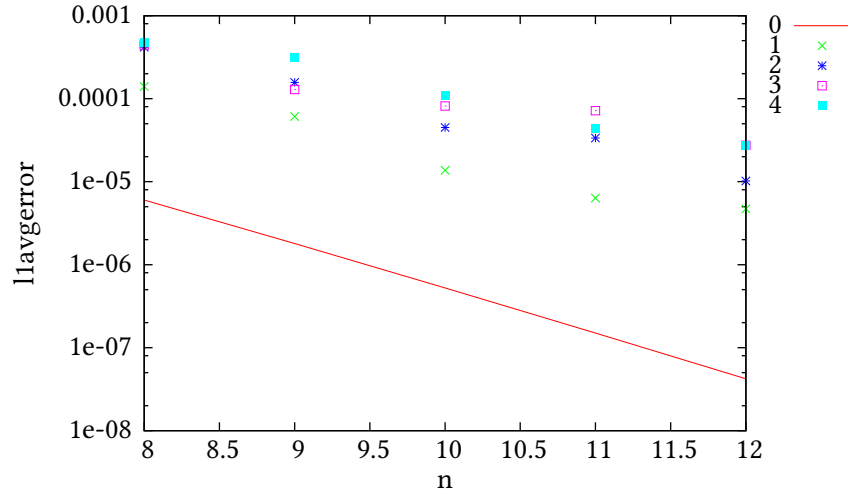


Abbildung 3.7.: Das Laplace Problem wird in 2D für die Dünngitter-Kombinationstechnik f_n^c gelöst. Für einen Ausfall von bis zu vier Teillösungen werden mit der Methode `faultRecoveryBlock, center=true` neue Linearkombinationen berechnet. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

2.3 beschrieben ist der Gesamtaufwand polynomiell in n beschränkt. Unter der Annahme, dass die Anzahl der verworfenen Teillösungen eine Schätzung für die Verschlechterung des Fehlers gegenüber der ursprünglichen Linearkombination ist, würden wir hier erwarten schlechter als die Methode `faultRecoverySlow` abzuschneiden.

Eine zweite Möglichkeit, die unabhängig davon unter gewissen Bedingungen möglich ist, ist lokal nach Teillösungen zu suchen, die verworfen werden. Für die Verkettung $P_{\vec{l}_1}P_{\vec{l}_2} = P_{\vec{l}_1 \wedge \vec{l}_2}$ zweier Projektionsoperatoren $P_{\vec{l}_1}$ und $P_{\vec{l}_2}$ gilt $\vec{l}_1 \geq \vec{l}_1 \wedge \vec{l}_2 \leq \vec{l}_2$.

Seien nun z.B. $\vec{\lambda}_1, \vec{\lambda}_2 \in \Lambda$ gegeben, es gelte $K_1 = \{ \vec{l} \in L \mid \vec{\lambda}_1 \leq \vec{l} \}, K_2 = \{ \vec{l} \in L \mid \vec{\lambda}_2 \leq \vec{l} \}$ und sei des Weiteren $K_1 \cap K_2 = \emptyset$, dann gilt

$$\forall \vec{k}_1 \in K_1 \forall \vec{k}_2 \in K_2 \forall \vec{l} \in L : \left(P_{\vec{l}}P_{\vec{k}_1} = P_{\vec{k}_1}P_{\vec{l}} \neq P_{\vec{\lambda}_2} \wedge P_{\vec{l}}P_{\vec{k}_2} = P_{\vec{k}_2}P_{\vec{l}} \neq P_{\vec{\lambda}_1} \right) \quad (3.3)$$

Dies bedeutet, dass sich das Entfernen bzw. Hinzufügen von Teillösungen $f_{\vec{k}_2}$ mit $\vec{k}_2 \in K_2$ nicht auf den nach Gleichung 2.8 berechneten Koeffizienten $c_{\vec{\lambda}_1}$ der zugehörigen Teillösung $f_{\vec{\lambda}_1}$ auswirkt, genauso wie sich auch das Entfernen bzw. Hinzufügen von Teillösungen $f_{\vec{k}_1}$ mit $\vec{k}_1 \in K_1$ nicht auf den nach Gleichung 2.8 berechneten Koeffizienten $c_{\vec{\lambda}_2}$ der zugehörigen Teillösung $f_{\vec{\lambda}_2}$ auswirkt. Eine mögliche Strategie wäre also die Menge Λ der Levelvektoren ausgefallener Teillösungen wie folgt zu partitionieren: Sei also $\Lambda = \dot{\bigcup}_{i=1}^t \Lambda_i$ und für $1 \leq i \leq t$ sei $K_i = \{ \vec{l} \in L \mid \exists \vec{\lambda} \in \Lambda_i : \vec{\lambda} \leq \vec{l} \}$ und es gelte $K_i \cap K_j = \emptyset$ für alle $i \neq j$ mit $1 \leq i, j \leq t$. Dann können die Anzahl der ausgefallenen Teillösungen für jedes $K_i, i \in \{1, \dots, t\}$ unabhängig von einander gegebenenfalls unter Beachtung von Gleichung 3.1 minimiert werden. Die neu berechnete Linearkombination entspricht der, welche auch die Methode `faultRecoverySlow` liefert, allerdings kann sich der Rechenaufwand

auch gegebenenfalls sogar verschlechtern, etwa dann wenn die Partition aus genau einem Element besteht.

Eine andere Möglichkeit wäre, eine Richardson-Extrapolation die auch Fehlerterme höherer Ordnung auslöscht gegebenenfalls unter Verwendung der Verknüpfung mit der Kombinationstechnik zu verwenden.

4. Zusammenfassung

In dieser Arbeit wurden Verfahren zur fehlertoleranten Lösung von partiellen Differentialgleichungen mittels Linearkombinationen von Teillösungen auf anisotropen vollen Gittern untersucht. Dabei wurden im Wesentlichen die Ergebnisse aus [HH14] nachvollzogen.

Zuerst wurden die benötigten Definitionen wie die Dünngitter-Kombinationstechnik oder die Truncated Combination Technique und Fehlerabschätzungen eingeführt. Für das Laplace Problem und die Advektionsgleichung wurden in 2D und 3D, unter Verwendung von Truncated Combination Techniques mit einer konstanten Anzahl an Teillösungen untersucht, wie sich ein Ausfall einer Teillösung auf den Fehler der Kombinationstechnik auswirkt.

Die grundlegende Idee ist dabei, dass bei einem Ausfall einer Teillösung der Kombinationstechnik diese angepasst werden kann. So konnten etwa einzelne Teillösungen verworfen werden und dabei die Koeffizienten der Linearkombination angepasst werden oder eine ausgefallene Teillösung durch eine Interpolation approximiert werden. Eine andere Möglichkeit war noch nicht verwendete Teillösungen zu berechnen oder diese durch eine Interpolation zu erhalten.

Dabei wurde festgestellt, dass sich die Interpolationen hinsichtlich des Kosten/ Nutzen Verhältnisses nicht lohnten und es besser war vorsorglich alle kleineren Teillösungen im Voraus zu berechnen und diese bei Bedarf zu verwenden.

In Abschnitt 2.6 wurde dann die Idee der multivariaten Richardson-Extrapolation erläutert, welche darauf basiert Charakteristika von Fehlerexpansionen auszunutzen, welche unter bestimmten Voraussetzungen für bestimmte PDEs gelten, um durch Auslöschung von Fehlertermen eine bessere Konvergenz zu erreichen und die Ergebnisse für diese dargestellt. In Abschnitt 2.7 wurde dann die Verknüpfung beider Verfahren erläutert, welche eine Approximation höherer Ordnung auf einem gröberen und eine Approximation geringerer Ordnung auf einem feineren Gitter aufweist.

Zum Schluss wurden in Kapitel 3, die Implementierung, von Verfahren zur Berechnung einer neuen Linearkombination von Teillösungen im Fehlerfall, anhand der vorherigen Ergebnisse, in das Dünngitter Framework SG++ vorgestellt. Dabei wurde festgestellt, dass das Verwerfen von allen feineren Teillösungen und die anschließende Berechnung einer neuen Kombination durch das Prinzip von Inklusion und Exklusion, sowie die Richardson-Extrapolation die besten Ergebnisse lieferten.

A. Anhang

A.1. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2}^2$

case 0 : $f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - f_{n-2,n-1} - f_{n-1,n-2}$

case 1 : $f_{n,n-2} + f_{n-1,n-1} - f_{n-1,n-2}$

case 2 : $f_{n-2,n} + f_{n-1,n-1} - f_{n-2,n-1}$

case 3 : $f_{n,n-2}$

case 4: $f_{n-2,n}$

case 5: $f_{n-2,n} + f_{n,n-2} - f_{n-2,n-2}$

case 6 : $f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - P_{n-2,n-1}f_{n-2,n} - f_{n-1,n-2}$

case 7 : $f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - f_{n-2,n-1} - P_{n-1,n-2}f_{n,n-2}$

case 8 : $f_{n-2,n} + f_{n,n-2} - P_{n-2,n-2}f_{n-2,n}$

case 9 : $f_{n-2,n} + f_{n,n-2} - P_{n-2,n-2}f_{n,n-2}$

case 10 : $f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - P_{n-2,n-1}f_{n-1,n-1} - f_{n-1,n-2}$

case 11 : $f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - f_{n-2,n-1} - P_{n-1,n-2}f_{n-1,n-1}$

case 12 : $f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - P_{n-2,n-1}(\frac{1}{2}(f_{n-2,n} + f_{n-1,n-1})) - f_{n-1,n-2}$

case 13 : $f_{n-2,n} + f_{n-1,n-1} + f_{n,n-2} - f_{n-2,n-1} - P_{n-1,n-2}(\frac{1}{2}(f_{n-1,n-1} + f_{n,n-2}))$

A.2. Kombinationen nach einem Fehler in $\tilde{f}_{n-3,n-3}^3$

case 0: $f_{n-3,n} + f_{n-2,n-1} + f_{n-1,n-2} + f_{n,n-3} - f_{n-3,n-1} - f_{n-2,n-2} - f_{n-1,n-3}$

case 1: $f_{n-2,n-1} + f_{n-1,n-2} + f_{n,n-3} - f_{n-2,n-2} - f_{n-1,n-3}$

case 2: $f_{n-3,n} + f_{n-2,n-1} + f_{n-1,n-2} - f_{n-3,n-1} - f_{n-2,n-2}$

case 3: $f_{n-1,n-2} + f_{n,n-3} - f_{n-1,n-3}$

case 4: $f_{n-3,n} + f_{n-2,n-1} + f_{n-3,n-1}$

case 5: $f_{n-3,n} + f_{n-2,n-1} + f_{n-1,n-2} + f_{n,n-3} - P_{n-3,n-1}f_{n-3,n} - f_{n-2,n-2} - f_{n-1,n-3}$

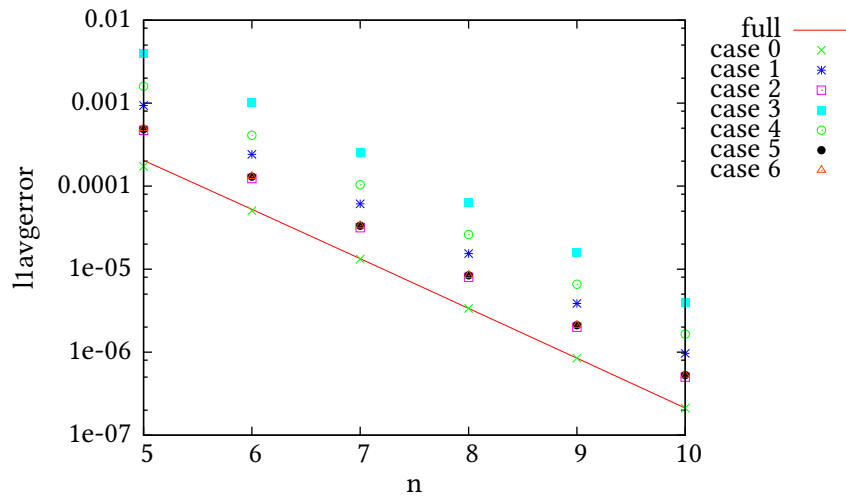


Abbildung A.1.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Kombinationstechnik $\tilde{f}_{n-3,n-3}^3$ und den Linearkombinationen wie sie in Anhang A.2 beschrieben sind, falls eine Teillösung ausfällt. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

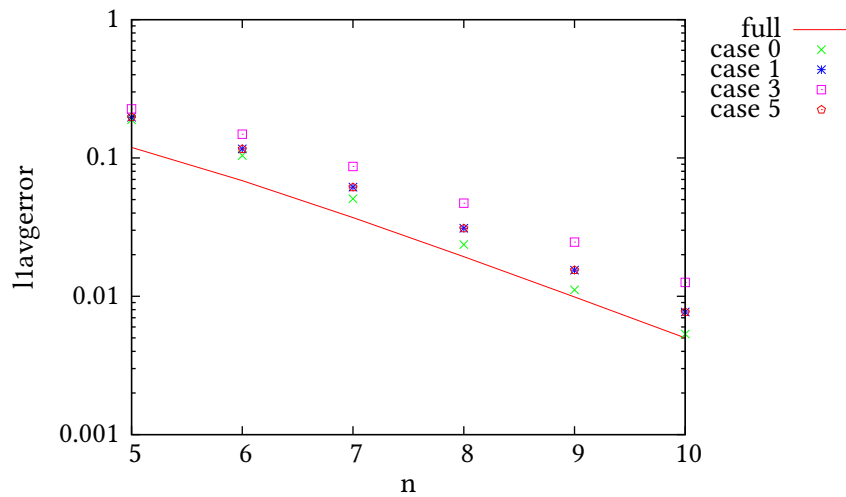


Abbildung A.2.: Dargestellt sind die Fehler bei der Lösung der Advektionsgleichung mit der Kombinationstechnik $\tilde{f}_{n-3,n-3}^3$ und den Linearkombinationen wie sie in Anhang A.2 beschrieben sind, falls eine Teillösung ausfällt. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

A.3. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2}^2$ mit Richardson Extrapolation p=2

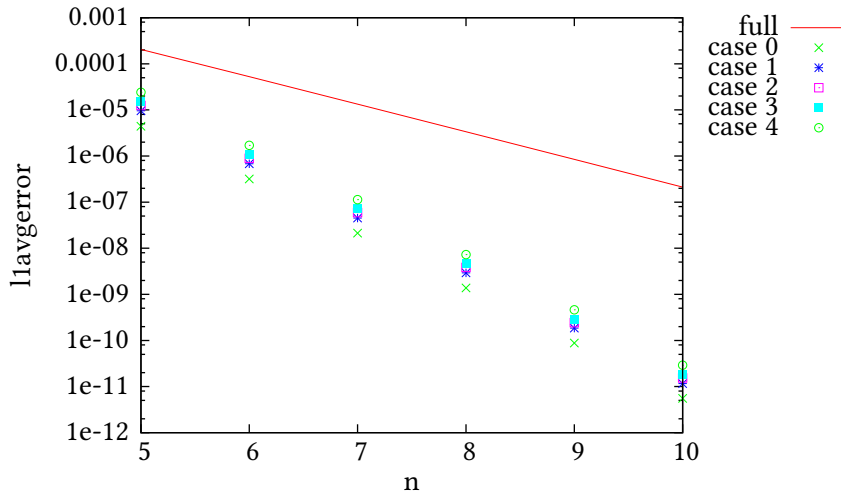


Abbildung A.3.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Richardson-Extrapolation auf $V_{n-2,n-2}$. Die einzelnen Fälle sind in Anhang A.3 beschrieben. Der Fehler für $f_{n,n}$ wird auf $V_{n,n}$ und für alle anderen Fälle auf $V_{n-2,n-2}$ berechnet.

case 6: $f_{n-3,n} + f_{n-2,n-1} + f_{n-1,n-2} + f_{n,n-3} - f_{n-3,n-1} - f_{n-2,n-2} - P_{n-1,n-3}f_{n,n-3}$

full: $f_{n,n}$

A.3. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2}^2$ mit Richardson Extrapolation p=2

case 0: $\frac{-16}{9}f_{n-2,n} + \frac{23}{9}f_{n-1,n-1} + \frac{-16}{9}f_{n,n-2} + f_{n-2,n-1} + f_{n-1,n-2}$

case 1: $\frac{-4}{9}f_{n-2,n} + \frac{17}{9}f_{n-1,n-1} + \frac{4}{9}f_{n,n-2}$

case 2: $\frac{4}{3}f_{n-1,n-1} + \frac{4}{3}f_{n,n-2} + \frac{-5}{3}f_{n-1,n-2}$

case 3: $\frac{4}{3}f_{n-2,n} + \frac{4}{3}f_{n-1,n-1} + \frac{-5}{3}f_{n-2,n-1}$

case 4: $\frac{20}{6}f_{n-2,n} + \frac{20}{6}f_{n,n-2} + \frac{-17}{6}f_{n-2,n-1} + \frac{-17}{6}f_{n-1,n-2}$

A.4. Kombinationen nach einem Fehler in $\tilde{f}_{n-3,n-3}^3$ mit Richardson Extrapolation p=2

case 0: $\frac{-64}{45}f_{n-3,n} + \frac{19}{45}f_{n-2,n-1} + \frac{19}{45}f_{n-1,n-2} + \frac{-64}{45}f_{n,n-3} + f_{n-3,n-1} + f_{n-2,n-2} + f_{n-1,n-3}$

case 1: $\frac{-2}{9}f_{n-3,n} + \frac{13}{18}f_{n-2,n-1} + \frac{13}{18}f_{n-1,n-2} + \frac{-2}{9}f_{n,n-3}$

case 2: $\frac{-16}{9}f_{n-2,n-1} + \frac{23}{9}f_{n-1,n-2} + \frac{-16}{9}f_{n,n-3} + f_{n-2,n-2} + f_{n-1,n-3}$

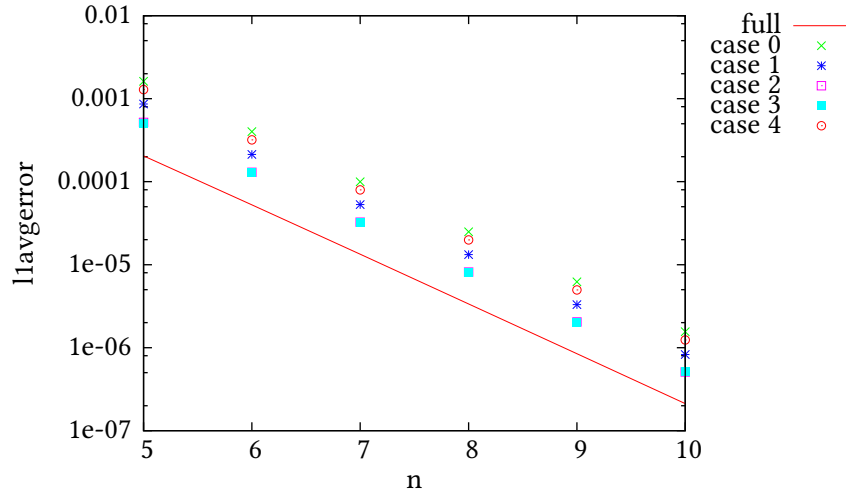


Abbildung A.4.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Richardson-Extrapolation auf $V_{n,n}$. Die einzelnen Fälle sind in Anhang A.3 beschrieben. Für alle Fälle wird der Fehler auf $V_{n,n}$ berechnet.

$$\text{case 3: } \frac{-16}{9}f_{n-3,n} + \frac{23}{9}f_{n-2,n-1} + \frac{-16}{9}f_{n-1,n-2} + f_{n-3,n-1} + f_{n-2,n-2}$$

$$\text{case 4: } \frac{-4}{9}f_{n-2,n-1} + \frac{17}{9}f_{n-1,n-2} + \frac{-4}{9}f_{n,n-3}$$

$$\text{case 5: } \frac{-4}{9}f_{n-3,n} + \frac{17}{9}f_{n-2,n-1} + \frac{-4}{9}f_{n-1,n-2}$$

$$\text{case 6: } \frac{4}{3}f_{n-1,n-2} + \frac{4}{3}f_{n,n-3} + \frac{-5}{3}f_{n-1,n-3}$$

$$\text{case 7: } \frac{4}{3}f_{n-2,n-1} + \frac{4}{3}f_{n-1,n-2} + \frac{-5}{3}f_{n-2,n-2}$$

$$\text{case 8: } \frac{4}{3}f_{n-3,n} + \frac{4}{3}f_{n-2,n-1} + \frac{-5}{3}f_{n-3,n-1}$$

A.5. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2}^2$ mit Richardson Extrapolation und verknüpfter Technik p=1

$$\text{case 0: } -2P_{n-2,n-2}f_{n-2,n} + 5P_{n-2,n-2}f_{n-1,n-1} - 2P_{n-2,n-2}f_{n,n-2}$$

$$\text{case 1: } 2P_{n-2,n-2}f_{n-2,n} + 2P_{n-2,n-2}f_{n-1,n-1} - 3P_{n-2,n-2}f_{n-2,n-1}$$

$$\text{case 2: } P_L P_{n-2,n-2}(-2f_{n-2,n} + 5f_{n-1,n-1} - 2f_{n,n-2}) + \tilde{f}_{n-2,n-2}^2 - P_{n-2,n-2}\tilde{f}_{n-2,n-2}^2 \text{ mit } L = \{(n-2, n), (n-1, n-1), (n, n-2)\}$$

$$\text{case 3: } P_L P_{n-2,n-2}(2f_{n-2,n} + 2f_{n-1,n-1} - 3f_{n-2,n-1}) + (f_{n-2,n} + f_{n-1,n-1} - f_{n-2,n-1}) - P_{n-2,n-2}(f_{n-2,n} + f_{n-1,n-1} - f_{n-2,n-1}) \text{ mit } L = \{(n-2, n), (n-1, n-1)\}$$

A.6. Kombinationen nach einem Fehler in $\tilde{f}_{n-3,n-3}^3$ mit Richardson Extrapolation und verknüpfter Technik p=1

case 0: $P_{n-3,n-3}(-2f_{n-3,n} + 5f_{n-2,n-1} - 2f_{n-1,n-2})$

case 1: $P_{n-3,n-3}(2f_{n-3,n} + 2f_{n-2,n-1} - 3f_{n-3,n-1})$

case 2: $P_{n-3,n-3}(2f_{n-2,n-1} + 2f_{n-1,n-2} - 3f_{n-2,n-2})$

case 3: $P_L P_{n-3,n-3}(-2f_{n-3,n} + 5f_{n-2,n-1} - 2f_{n-1,n-2})$
 $+ (f_{n-3,n} + f_{n-2,n-1} - f_{n-1,n-2} - f_{n-3,n-1} - f_{n-2,n-2})$
 $- P_{n-3,n-3}(f_{n-3,n} + f_{n-2,n-1} - f_{n-1,n-2} - f_{n-3,n-1} - f_{n-2,n-2})$

mit $L = \{ (n-3, n), (n-2, n-1), (n-1, n-2) \}$

case 4: $P_L P_{n-3,n-3}(2f_{n-3,n} + 2f_{n-2,n-1} - 3f_{n-3,n-1}) + (f_{n-3,n} + f_{n-2,n-1} - f_{n-3,n-1})$
 $- P_{n-3,n-3}(f_{n-3,n} + f_{n-2,n-1} - f_{n-3,n-1})$

mit $L = \{ (n-3, n), (n-2, n-1) \}$

case 5: $P_L P_{n-3,n-3}(2f_{n-2,n-1} + 2f_{n-1,n-2} - 3f_{n-2,n-2}) + (f_{n-2,n-1} + f_{n-1,n-2} - f_{n-2,n-2})$
 $- P_{n-3,n-3}(f_{n-2,n-1} + f_{n-1,n-2} - f_{n-2,n-2})$

mit $L = \{ (n-2, n-1), (n-1, n-2) \}$

A.7. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2,n-2}^2$

full: $f_{n,n,n}$

case 0:

$$f_{n-2,n-2,n} + f_{n-2,n,n-2} + f_{n,n-2,n-2} + f_{n-2,n-1,n-1} + f_{n-1,n-2,n-1} + f_{n-1,n-1,n-2}$$

$$- 2f_{n-2,n-2,n-1} - 2f_{n-2,n-1,n-2} - 2f_{n-1,n-2,n-2} + f_{n-2,n-2,n-2}$$

case 1:

$$0f_{n-2,n-2,n} + f_{n-2,n,n-2} + f_{n,n-2,n-2} + f_{n-2,n-1,n-1} + f_{n-1,n-2,n-1} + f_{n-1,n-1,n-2}$$

$$- 1f_{n-2,n-2,n-1} - 2f_{n-2,n-1,n-2} - 2f_{n-1,n-2,n-2} + f_{n-2,n-2,n-2}$$

case 4:

$$f_{n-2,n-2,n} + f_{n-2,n,n-2} + f_{n,n-2,n-2} + 0f_{n-2,n-1,n-1} + f_{n-1,n-2,n-1} + f_{n-1,n-1,n-2}$$

$$- 1f_{n-2,n-2,n-1} - 1f_{n-2,n-1,n-2} - 2f_{n-1,n-2,n-2} + f_{n-2,n-2,n-2}$$

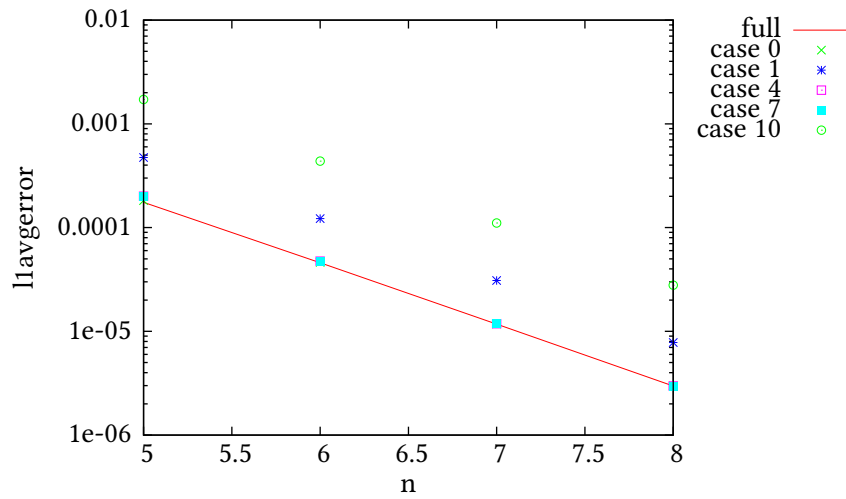


Abbildung A.5.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Kombinationstechnik $\tilde{f}_{n-2,n-2,n-2}^2$ und den Linearkombinationen wie sie in Anhang A.7 beschrieben sind, falls eine Teillösung ausfällt. Für alle Fälle wird der Fehler auf $V_{n,n,n}$ berechnet.

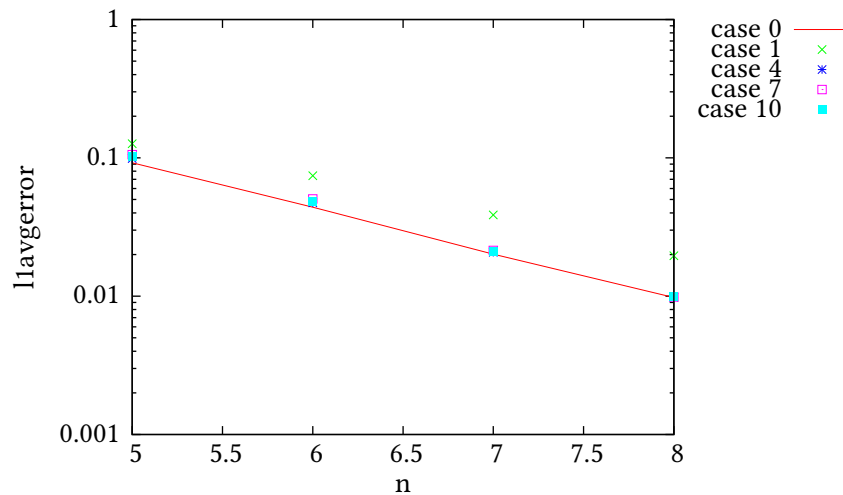


Abbildung A.6.: Dargestellt sind die Fehler bei der Lösung der Advektionsgleichung mit der Kombinationstechnik $\tilde{f}_{n-2,n-2,n-2}^2$ und den Linearkombinationen wie sie in Anhang A.7 beschrieben sind, falls eine Teillösung ausfällt. Die Berechnung der Vollgitterlösung $f_{n,n,n}$ ist zu rechenaufwändig und wurde nicht berechnet. Für alle Fälle wird der Fehler auf $V_{n,n,n}$ berechnet.

A.8. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2,n-2}^2$ mit Richardson Extrapolation p=2

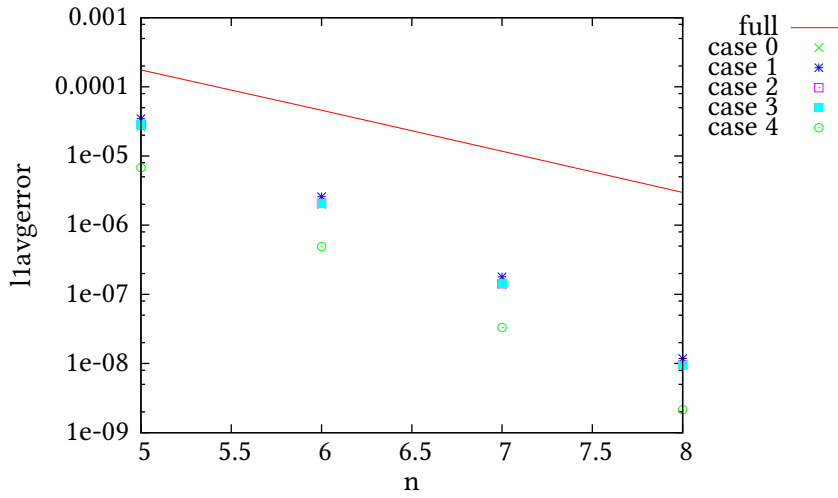


Abbildung A.7.: Dargestellt sind die Fehler bei der Lösung der Laplace-Gleichung mit der Richardson-Extrapolation auf $V_{n-2,n-2,n-2}$. Die einzelnen Fälle sind in Anhang A.8 beschrieben. Der Fehler für $f_{n,n,n}$ wird auf $V_{n,n,n}$ und für alle anderen Fälle auf $V_{n-2,n-2,n-2}$ berechnet.

case 7:

$$f_{n-2,n-2,n} + f_{n-2,n,n-2} + f_{n,n-2,n-2} + 0f_{n-2,n-1,n-1} + 0f_{n-1,n-2,n-1} + f_{n-1,n-1,n-2} \\ + 0f_{n-2,n-2,n-1} - 1f_{n-2,n-1,n-2} - 1f_{n-1,n-2,n-2} - 1f_{n-2,n-2,n-2}$$

case 10:

$$f_{n-2,n-2,n} + f_{n-2,n,n-2} + f_{n,n-2,n-2} + f_{n-2,n-1,n-1} + 0f_{n-1,n-2,n-1} + f_{n-1,n-1,n-2} \\ - 1f_{n-2,n-2,n-1} - 1f_{n-2,n-1,n-2} - 2f_{n-1,n-2,n-2} + 0f_{n-2,n-2,n-2}$$

A.8. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2,n-2}^2$ mit Richardson Extrapolation p=2

$$\text{case 0: } \frac{-8}{9}f_{n-2,n-2,n} + \frac{-8}{9}f_{n-2,n,n-2} + \frac{-8}{9}f_{n,n-2,n-2} + \frac{11}{9}f_{n-2,n-1,n-1} + \frac{11}{9}f_{n-1,n-2,n-1} + \frac{11}{9}f_{n-1,n-1,n-2}$$

$$\text{case 1: } \frac{4}{3}f_{n-1,n-2,n-2} + \frac{4}{3}f_{n-2,n-1,n-2} + \frac{4}{3}f_{n-2,n-2,n-1} - 3f_{n-2,n-2,n-2}$$

$$\text{case 2: } \frac{4}{3}f_{n-1,n-2,n-1} + \frac{4}{3}f_{n-2,n-1,n-1} + \frac{4}{3}f_{n-2,n-2,n} - 3f_{n-2,n-2,n-1}$$

$$\text{case 3: } \frac{4}{3}f_{n-1,n-1,n-2} + \frac{4}{3}f_{n-2,n,n-2} + \frac{4}{3}f_{n-2,n-1,n-1} - 3f_{n-2,n-1,n-2}$$

$$\text{case 4: } \frac{4}{3}f_{n,n-2,n-2} + \frac{4}{3}f_{n-1,n-1,n-2} + \frac{4}{3}f_{n-1,n-2,n-1} - 3f_{n-1,n-2,n-2}$$

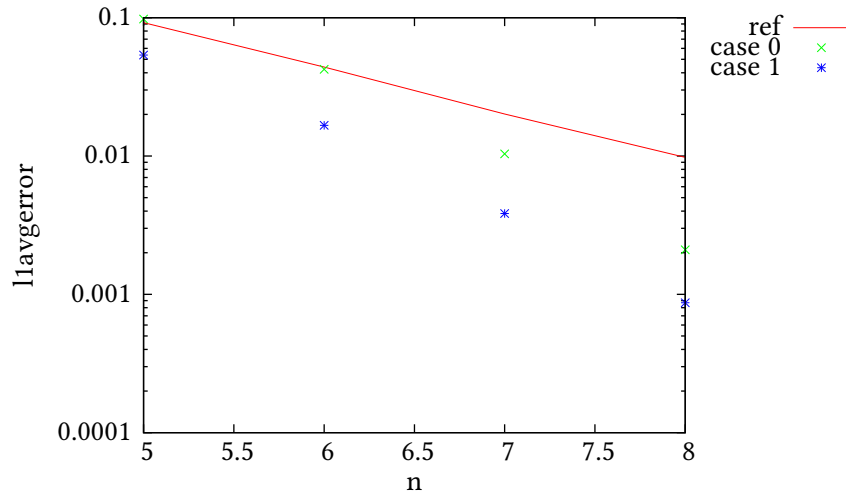


Abbildung A.8.: Dargestellt sind die Fehler bei der Lösung der Advektionsgleichung mit der Richardson-Extrapolation auf $V_{n-2,n-2,n-2}$. Die einzelnen Fälle sind in Anhang A.9 beschrieben. Die Berechnung der Vollgitterlösung $f_{n,n,n}$ ist zu rechenaufwändig, deswegen entspricht „ref“ dem Fall 0 aus Anhang A.6, die Fehler für Fall 0 und Fall 1 werden auf $V_{n-2,n-2,n-2}$ berechnet.

A.9. Kombinationen nach einem Fehler in $\tilde{f}_{n-2,n-2,n-2}^2$ mit Richardson Extrapolation p=1

case 0: $2f_{n-2,n-2,n-1} + 2f_{n-2,n-1,n-2} + 2f_{n-1,n-2,n-2} - 5f_{n-2,n-2,n-2}$

case 1: $2f_{n-2,n-2,n} + 2f_{n-2,n-1,n-1} + 2f_{n-1,n-2,n-1} - 5f_{n-2,n-2,n-1}$

Literaturverzeichnis

- [BBNS12] J. Benk, H.-J. Bungartz, A.-E. Nagy, S. Schraufstetter. Variants of the Combination Technique for Multi-Dimensional Option Pricing. *Progress in Industrial Mathematics at ECMI 2010*, S. 231–237, 2012. doi:10.1007/978-3-642-25100-9_27. URL http://dx.doi.org/10.1007/978-3-642-25100-9_27. (Zitiert auf den Seiten 11 und 17)
- [Bel61] R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton Univ. Pr., Princeton, NJ, 1961. (Zitiert auf Seite 10)
- [BG04] H.-J. Bungartz, M. Griebel. Sparse grids. *ANU*, 13:147, 2004. doi:10.1017/s0962492904000182. URL <http://dx.doi.org/10.1017/S0962492904000182>. (Zitiert auf den Seiten 10, 11 und 20)
- [BGR94] H. Bungartz, M. Griebel, U. R de. Extrapolation, Combination, And Sparse Grid Techniques For Elliptic Boundary Value Problems, 1994. (Zitiert auf Seite 32)
- [BP12] J. Benk, D. Pfl ger. Hybrid Parallel Solutions of the Black-Scholes PDE with the Truncated Combination Technique. *In Proceedings of the HPCS conference*, S. 678–683, 2012. (Zitiert auf den Seiten 11, 24 und 43)
- [BS06] G. A. G. Bianca Schroeder. A large-scale study of failures in highperformance computing systems, 2006. (Zitiert auf Seite 10)
- [BZBP13] H.-J. Bungartz, S. Zimmer, M. Buchholz, D. Pfl ger. *Modellbildung und Simulation: Eine anwendungsorientierte Einf hrung (eXamen.press) (German Edition)*. Springer Spektrum, 2013. URL <http://www.amazon.com/Modellbildung-Simulation-anwendungsorientierte-Einf%C3%BChrung-eXamen-press-ebook/dp/B00FZFKTCE%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechnikie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3DB00FZFKTCE>. (Zitiert auf Seite 9)
- [dev10] T. D.-F. developers. *Introduction to Dune-Fem*. Abteilung f r Angewandte Mathematik, Universit t Freiburg, Hermann-Herder-Str. 10, D-79104 Freiburg, Germany, 2010. URL <http://dune.mathematik.uni-freiburg.de/doc/dune-fem-howto-1.3.0.pdf>. (Zitiert auf den Seiten 26 und 27)
- [ELSC13] I. P. Egwuotuoha, D. Levy, B. Selic, S. Chen. A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *J Supercomput*, 65(3):1302–1326, 2013. doi:10.1007/s11227-013-0884-0. URL <http://dx.doi.org/10.1007/s11227-013-0884-0>. (Zitiert auf Seite 9)

- [Gar04] J. Garcke. *Maschinelles Lernen durch Funktionsrekonstruktion mit verallgemeinerten dünnen Gittern*. Dissertation, Institut für Numerische Simulation, Rheinische Friedrich–Wilhelms–Universität Bonn, 2004. (Zitiert auf Seite 11)
- [GKKS09] A. Geist, S. Kale, B. Kramer, M. Snir. Toward exascale resilience, 2009. (Zitiert auf Seite 10)
- [GSZ92] M. Griebel, M. Schneider, C. Zenger. A Combination Technique For The Solution Of Sparse Grid Problems, 1992. (Zitiert auf Seite 10)
- [Heg01] M. Hegland. Adaptive Sparse Grids, 2001. (Zitiert auf Seite 22)
- [HGC07] M. Hegland, J. Garcke, V. Challis. The combination technique and some generalisations. *Linear Algebra and its Applications*, 420(2-3):249–275, 2007. doi:10.1016/j.laa.2006.07.014. URL <http://dx.doi.org/10.1016/j.laa.2006.07.014>. (Zitiert auf Seite 11)
- [HH12] B. Harding, M. Hegland. A robust combination technique. *ANZIAM Journal*, 2012. (Zitiert auf den Seiten 13, 21, 22, 26 und 41)
- [HH14] B. Harding, M. Hegland. Robust Solutions to PDEs with Multiple Grids. In J. Garcke, D. Pflüger, Herausgeber, *Sparse Grids and Applications - Munich 2012*, Band 97 von *Lecture Notes in Computational Science and Engineering*, S. 171–193. Springer International Publishing, 2014. doi:10.1007/978-3-319-04537-5_7. URL http://dx.doi.org/10.1007/978-3-319-04537-5_7. (Zitiert auf den Seiten 11, 13, 17, 18, 20, 21, 23, 24, 31, 32, 33, 34, 35 und 51)
- [PB] A. D. C. E. R. K. M. N. M. O. O. S. Peter Bastian, Markus Blatt. *The Distributed and Unified Numerics Environment (DUNE) Grid Interface HOWTO*. URL <http://www.dune-project.org/dune.html>. (Zitiert auf Seite 26)
- [Pfl10] D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Dissertation, 2010. (Zitiert auf den Seiten 11, 13, 24 und 39)
- [Top] Top500. <http://www.top500.org/statistics/perfdevel/>. (Zitiert auf Seite 9)
- [Zen91] C. Zenger. Sparse Grids. In W. Hackbusch, Herausgeber, *Parallel Algorithms for Partial Differential Equations*, Band 31 von *Notes on Numerical Fluid Mechanics*, S. 241–251. Vieweg, 1991. (Zitiert auf den Seiten 11 und 30)

Alle URLs wurden zuletzt am 02. 06. 2014 geprüft.

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