

Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 292

Erkennung von Silent Faults mit der Kombinationstechnik

Marcel Hurler

Studiengang:	Informatik
Prüfer/in:	Jun.-Prof. Dr. rer. nat. Dirk Pflüger
Betreuer/in:	Jun.-Prof. Dr. rer. nat. Dirk Pflüger, M.Sc. Mario Heene
Beginn am:	17. Dezember 2015
Beendet am:	17. Juni 2016
CR-Nummer:	G.1.8

Kurzfassung

Innerhalb dieser Arbeit werden zwei Verfahren vorgestellt, um bisher unerkannte Fehler während der Simulation von Diffusions- und Advektionsproblemen mit der Dünngitter Kombinationstechnik aufzudecken. Der Schwerpunkt liegt hierbei auf der Erkennung von Fehlern anhand der zeitlichen Entwicklung einer einzelnen Kombinationslösung, ohne Vergleiche zwischen anderen Kombinationslösungen herzustellen. Die Untersuchungen zeigen, dass die beiden Verfahren im Falle eines für die Lösung relevanten Fehlers zum Zeitpunkt der Korruption ein unnatürliches Verhalten aufweisen, welches ein hohes Potential besitzt, um von einem automatisierten Erkennungsverfahren detektiert werden zu können.

Inhaltsverzeichnis

1	Einleitung und Motivation	13
2	Grundlagen	17
2.1	Dünne Gitter und Kombinations-Technik	17
3	Silent Faults und Kombinationstechnik	23
3.1	Erkennung von SDC nach Hinojosa	26
3.2	Erkennung von SDC anhand der zeitlichen Entwicklung	27
4	Verhalten der Fehlererkennungsverfahren und Ergebnisse	35
4.1	Szenario 1: Diffusion, Dirichlet, quadratische Funktion	37
4.2	Szenario 2: Advektion, Dirichlet, Sinus Funktion	42
4.3	Szenario 3: Advektion, Dirichlet, quadratische Funktion	48
4.4	Szenario 4: Advektion-Diffusion, Dirichlet, Sinus Funktion	51
4.5	Ergebnis	54
5	Zusammenfassung und Ausblick	59
	Literaturverzeichnis	61

Abbildungsverzeichnis

2.1	Diskretisierung von Funktionen	20
3.1	Entwicklung eines Advektionsprozesses über die Zeit	24
3.2	Mögliche Schritte bei der Berechnung der SGCT	24
3.3	Veranschaulichung der Abweichung vom Interpolanten	30
4.1	Szenario 1: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dün- nen Gitter	37
4.2	Szenario 1: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}	40
4.3	Szenario 1: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit	41
4.4	Szenario 1: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen . .	42
4.5	Szenario 2: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dün- nen Gitter	44
4.6	Szenario 2: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}	45
4.7	Szenario 2: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit	47
4.8	Szenario 2: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen . .	48
4.9	Szenario 3: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dün- nen Gitter	49
4.10	Szenario 3: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}	50
4.11	Szenario 3: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit	51
4.12	Szenario 3: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen . .	53
4.13	Szenario 4: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dün- nen Gitter	54
4.14	Szenario 4: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}	55
4.15	Szenario 4: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit	56
4.16	Szenario 4: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen . .	57

Tabellenverzeichnis

4.1	Szenario 1: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{54}	38
4.2	Szenario 1: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel ??) zum Zeitschritt t_{54}	39
4.3	Szenario 2: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{55}	43
4.4	Szenario 2: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel 3.1.2) zum Zeitschritt t_{55}	43
4.5	Szenario 3: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{54}	46
4.6	Szenario 3: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel 3.1.2) zum Zeitschritt t_{54}	47
4.7	Szenario 4: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{55}	52
4.8	Szenario 4: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel 3.1.2) zum Zeitschritt t_{57}	52

Verzeichnis der Algorithmen

3.1	Fehlererkennung anhand der maximalen Änderung	30
3.2	Fehlererkennung anhand der maximalen Abweichung vom Interpolanten . . .	32
3.3	Lösung einer PDE mit dem SGCT Verfahren (angelehnt an [HHH])	34

1 Einleitung und Motivation

Der Einsatz von Computern in Natur- und technischen Wissenschaften zur Simulation und Modellierung von realen Systemen birgt enorme Vorteile. So lassen sich Experimente auch dann durchführen, wenn eine reale Ausführung entweder nicht möglich ist oder aufgrund unterschiedlicher Faktoren eingeschränkt wird. Ein Problem bei der Nachbildung von realistischen Prozessen ist die enorme Zahl an Unbekannten und Abhängigkeiten, welche beachtet werden müssen, um praxisnahe Ergebnisse zu erhalten. Ein Beispiel hierfür liefert die Wetterprognose. Um das Wetter für einen kurzen Zeitraum in Stuttgart vorherzusagen, genügt es, gemittelte Werte für Luftdruck, Windgeschwindigkeit und Temperatur und deren physikalisches Zusammenspiel innerhalb von Deutschland zu kennen. Um eine vernünftige Prognose für mehrere Tage zu erstellen, benötigt man schon genauere Informationen über ein größeres Gebiet. Für eine langfristige Prognose spielen dann immer mehr zunächst unscheinbarer Prozesse wie bspw. Abstand von Sonne zur Erde oder der Klimawandel eine wichtige Rolle.

Um zeitnah Resultate zu erhalten, werden Hochleistungsrechner (HPC) eingesetzt, die mehr als zehn- bis hunderttausend Knoten gleichzeitig betreiben und insgesamt über mehrere hundert TB an Speicher verfügen. Dadurch lassen sich heutzutage Spitzengeschwindigkeiten von mehreren hundert Petaflop/s (10^{15}) erreichen, welche bis zum Jahr 2020 in den Exaflop/s Bereich (10^{18}) gesteigert werden sollen [Pro; SWA+14]. Unter den enormen Ausmaßen solcher Großrechner (Prozessorkerne, Speicher-, Netzwerkkarten, etc.) sowie der steigenden Komplexität und Integrationsdichte integrierter Schaltkreise leidet jedoch die Zuverlässigkeit der durchgeführten Berechnungen. Das Auftreten von Fehlern soll demnach in Zukunft den Normalfall anstelle einer Ausnahme darstellen. Während heutzutage im Mittel etwa ein Fehler pro Tag gemessen wird, werden im Exascale Bereich im Minutentakt Fehler erwartet. Diese Fehler äußern sich größtenteils in sog. Bitflips und sind auf unterschiedliche Einflüsse wie bspw. kosmische Strahlung oder Qualitätsmängel in Hard- und Software zurückzuführen. Je nach betroffenem Datum, kann es bei Fehlern zu unterschiedlichsten Auswirkungen kommen [EHM14b; SWA+14]. Führt der Fehler zu einem Misserfolg, also der Abweichung von einem erwarteten Endzustand, spricht man von einem Versagen (Failure) [SWA+14]. Ein großer Teil an Bitflips kann meist direkt in der Hardware erkannt und teilweise korrigiert werden, wie etwa durch Prüfsummenverfahren im Hauptspeicher. Jedoch unterstützen nicht alle Komponenten einen solchen Schutz und fehlerhafte Daten können sich im System ausbreiten [SWA+14]. Snir et. al. schlägt deshalb eine Klassifikation von Fehlern in erkannte und korrigierte Fehler (DCE), erkannte und nicht korrigierbare (DUE) sowie nicht erkannte Fehler (SE) vor. Die Auswirkungen von unerkannten Fehlern sind nicht klar ersichtlich. So kann es passieren, dass bspw. eine Berechnung aufgrund eines korrupten Wertes ein falsches Ergebnis liefert (versagt),

der zugrunde liegende Algorithmus jedoch ohne Meldung terminiert. Fehler die keine Meldung hervorrufen, also dem Benutzer zunächst verborgen bleiben, sind nach [SWA+14] maskiert (masked), andere wiederum nicht maskiert (unmasked). Ein maskierter, nicht erkannter Fehler wird auch als silent data corruption (SDC) bezeichnet und die Ursache eines SDC könnte als silent fault bezeichnet werden.

Werden Fehler erkannt, gibt es unterschiedliche Möglichkeiten diese zu beheben. Neben redundanter Berechnungen wird das sog. Checkpoint/Restart Verfahren (CR) eingesetzt, um fehlerfreie Zustände zu Speichern oder zu Laden. Werden Fehler nicht erkannt, ist es Aufgabe der Software diese zu tolerieren, aufzudecken oder bestenfalls zu korrigieren. Elliot et. al. [EHM14b] stellt hierzu Richtlinien vor, welche beschreiben, wie das Auftreten von SDC simuliert werden kann und von Algorithmen behandelt werden soll. Dabei soll Fehlern nicht durch Prüfsummenchecks in Software begegnet werden, vielmehr sollen diese als numerische Abweichungen (bspw. Rundungsfehler) gesehen, mittels natürlicher Grenzen aus der numerischen Analysis erkannt und durch Anpassung des Verfahrens toleriert werden.

Da bei der Berechnung von Hochdimensionalen Problemen ein enormer Kommunikationsaufwand entsteht, wurde im Laufe der Jahre die sogenannte Dünngitter-Kombinationstechnik (SGCT) entwickelt, welche ihren Einsatz beispielsweise bei der Simulation des Plasmas innerhalb eines Fusionskraftwerkes Anwendung findet. Weiterhin bietet die SGCT gerade im Hinblick auf die Erkennung und Korrektur von Fehlern elegante Möglichkeiten. Zum Einen treten aufgrund der Vielzahl von Kombinationsgittern unterschiedlicher Auflösungen natürliche Redundanzen auf und zum Anderen lässt sich ein befallenes Gitter durch eine modifizierte Kombination ersetzen, ohne den aktuellen Fortschritt zu verwerfen und einen früheren Zustand wiederherzustellen. Hinojosa et. al. stellt in [HHH] zwei Verfahren vor, die bei der Lösung von PDEs anhand von Vergleichen einzelner Gitterpunkte auf hierarchischer Ebene in mehreren Kombinationsgittern Ausreißer erkennen und somit die Lokalisierung eines mit SDC befallenen Gitters ermöglichen. Mit diesen Verfahren lassen sich jedoch ausschließlich jene Gitter als korrupt erkennen, deren fehlerhafte Gitterpunkte auf mindestens drei weiteren Gittern zu finden sind. Andernfalls kann keine Entscheidung getroffen werden welches Gitter befallen ist. Demnach können besonders Gitter mit sehr hohen Auflösungen in eine Dimension nicht ohne Weiteres enttarnt werden. Aus diesem Grund werden innerhalb dieser Arbeit zwei Methoden untersucht, die die Gitter ohne den Vergleich von benachbarten Teilräumen aufdecken. Die Idee der Methoden ist es, den zeitlichen Verlauf einzelner Kombinationsgitter auszunutzen, um nach Auffälligkeiten zu suchen und somit Fehler zu erkennen.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Grundlagen: Innerhalb dieses Kapitels werden die Grundlagen zum weiteren Verständnis der Arbeit erläutert. Dabei werden Begriffe, Definition und Notation erklärt, welche in den folgenden Kapiteln weiterverwendet werden.

Kapitel 3 – Silent Faults und Kombinationstechnik: In diesem Kapitel wird die Auswirkung eines Fehlers auf die SGCT beschrieben. Weiterhin werden die von Hinojosa et. al. in [HHH] vorgestellten Fehlererkennungsverfahren erläutert. Zuletzt werden die beiden, innerhalb dieser Arbeit entstandenen Fehlererkennungsverfahren beschrieben, welche die zeitlichen Entwicklungen einer einzelnen Kombinationslösung verfolgen und sich im Fehlerfall, verglichen mit früheren Zeitschritten, durch ein ungewöhnliches Verhalten äußern.

Kapitel 4 – Verhalten der Fehlererkennungsverfahren und Ergebnisse: Im vorletzten Kapitel wird das Verhalten der zuvor beschriebenen Fehlererkennungsverfahren anhand von vier unterschiedlichen Szenarien verglichen und analysiert.

Kapitel 5 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor.

2 Grundlagen

Zum weiteren Verständnis werden innerhalb dieses Kapitels die Grundlagen der dünnen Gitter sowie die Kombinationstechnik näher erläutert.

2.1 Dünne Gitter und Kombinations-Technik

Um eine Funktion f im Rechner darstellen zu können, wird aufgrund des endlichen Speicherplatzes eine Diskretisierung verlangt. Dabei wird die Funktion an unterschiedlichen Punkten sogenannter Stützstellen x ausgewertet und die zugehörigen Funktionswerte $f(x)$ gespeichert. Funktionswerte an Punkten zwischen den Stützstellen lassen sich dann durch Interpolation mit geeigneten Basisfunktionen rekonstruieren, wobei je nach Anzahl und Wahl der Stützstellen sowie der Wahl der Basisfunktionen ein unterschiedlich großer Fehler entsteht. Ein Problem bei der Darstellung von Funktionen ist der exponentielle Anstieg an notwendigen Stützstellen, um mehrdimensionale Funktionen speichern zu können. So sind im naiven Fall, bei einer Auflösung von n Punkten pro Dimension, n^d Stützstellen für eine Funktion mit d Unbekannten erforderlich. Diese Problematik ist auch als Fluch der Dimensionen [] bekannt und kann nur durch eine geschicktere Wahl der Stützstellen eingeschränkt werden.

Neben einer stochastischen Verteilung der Stützstellen welche beispielsweise bei der Monte Carlo Quadratur Anwendung findet, bieten dünne Gitter eine weitere Möglichkeit mehrdimensionale Funktionen effizient darzustellen. Diese wurden ursprünglich entwickelt um PDEs effizient zu berechnen [Hac91]. Jedoch findet ihre Anwendung ebenso in anderen Bereichen wie bspw. der Finanzmathematik [RW05] oder dem Datamining [GGT01]. Die dünnen Gitter basieren auf dem Prinzip einer ausgedünnten Tensorprodukt-konstruktion, wobei ein Interpolant durch eine gewichtete Linearkombination aus hierarchischer Basisfunktionen dargestellt wird. Die Idee der Hierarchisierung ist es an den Gitterpunkten nicht die Funktionswerte, sondern nur die sogenannten hierarchischen Überschüsse zu speichern. Gitterpunkte mit geringen Überschüssen tragen nur wenig zum Fehler des Interpolanten bei und können demnach vernachlässigt werden. Mit zunehmender Anzahl an Dimensionen wächst die Zahl der so gesparten Gitterpunkte deutlich stärker als der Interpolationsfehler.

Um den Umgang mit dünnen Gittern zu vereinfachen, wurde im Laufe der Jahre die Dünngitter-Kombinationstechnik entwickelt [GSZ]. Der Vorteil dieses Verfahrens ist, dass durch eine Kombination aus mehreren Vollgitterlösungen eine Dünngitterlösung berechnet werden kann. Dabei lassen sich Probleme auf Vollgitterlösungen einerseits leichter diskretisieren. Andererseits sind

effiziente Datenstrukturen und Algorithmen weitaus weniger komplex als bei dünnen Gittern [MP11]. Zusätzlich bietet die Kombination eine natürliche Möglichkeiten zur Parallelisierung bereit, welches den Einsatz auf HPC vereinfacht. Ein Nachteil der Kombinationstechnik äußert sich, dass nicht alle Probleme mit diesem Verfahren lösbar sind. Dabei hängt es stark von den Glattheitseigenschaften der zugrundeliegenden PDE ab, ob die einzelnen Teillösungen konvergieren. Sind diese erfüllt, so liegt der Fehler in der selben Approximationsordnung wie der einer Dünngitterlösung.

Im Folgenden werden die Begriffe Gitter, dünne Gitter und Kombinationstechnik am Beispiel der Interpolation geklärt. Weiterhin wird die Hierarchisierung und die verwendeten Definitionen und Notationen sind dabei stark an [HHH; Pfl10] angelehnt.

Um den Umgang mit späteren Begriffen und Definitionen zu vereinfachen, werden zunächst einige Schreibweisen und Notationen erläutert.

Seien $\vec{l}, \vec{k} \in \mathbb{R}^d$ zwei Vektoren, so gelten für Relationen immer komponentenweise Vergleiche also,

$$(2.1) \quad \vec{l} \sim \vec{k} \Leftrightarrow \forall i \in \{1, \dots, d\} : l_i \sim k_i.$$

Ähnlich wie bei Relationen gilt auch für Operatoren die komponentenweise Anwendung, sei also $*$ ein Operator so ist

$$(2.2) \quad \vec{l} * \vec{k} = \vec{y} = (l_1 * k_1, \dots, l_d * k_d).$$

Weiterhin ist für ein Skalar $\lambda \in \mathbb{R}$ und einen Operator $*$

$$(2.3) \quad \lambda * \vec{l} = \vec{y} = (\lambda * l_1, \dots, \lambda * l_d)$$

Für Vektoren seien die diskrete ℓ_p -Norm $|\cdot|_p$ und Maximumsnorm $|\cdot|_\infty$ wie folgt definiert:

$$(2.4) \quad |\vec{l}|_p = \left(\sum_{i=1}^d |l_i|^p \right)^{1/p} \quad \text{und} \quad |\vec{l}|_\infty = \max_{1 \leq i \leq d} |l_i|.$$

Zusätzlich soll es erlaubt sein, Multiindizes mit gleichen Einträgen kompakter zu Schreiben. Demnach sei

$$(2.5) \quad \vec{n} = (n, \dots, n) \in \mathbb{N}^d.$$

Betrachten wir eine d -dimensionale Funktion $f : \Omega \rightarrow \mathbb{R}$. Um f durch eine Funktion u zu interpolieren, müssen wir den Definitionsbereich von f zunächst diskretisieren. Der Einfachheit halber beschränken wir den für uns interessanten Bereich auf den d -dimensionalen Einheitswürfel $\Omega := [0, 1]^d$ mit dem Rand $\partial\Omega$. Wir diskretisieren Ω durch ein volles Gitter $\Omega_{\vec{i}}$ mit Maschenweiten $h_{\vec{i}} := (h_{i_1}, \dots, h_{i_d})$ und äquidistanten Gitterpunkten $x_{\vec{i}, \vec{j}} = (x_{i_1, j_1}, \dots, x_{i_d, j_d})$, wobei $x_{i_k, j_k} = j_k \cdot h_{i_k}$ mit $0 \leq j_k \leq h_{i_k}^{-1}$. Der Multiindex $\vec{i} = (i_1, \dots, i_d) \in \mathbb{N}^d$ entspricht dabei

dem Auflösungsgrad, welcher mit $h_{\vec{l}} := 2^{-\vec{l}}$ die Maschenweiten des Gitters $\Omega_{\vec{l}}$ in Richtung der einzelnen Dimensionen angibt. Sind alle $i_1 = \dots = i_d$ gleich, so spricht man von einem isotropen, ansonsten von einem anisotropen vollen Gitter. Ein gegebenes Gitter besteht somit aus insgesamt $\prod_{k=1}^d (2^{l_k} + 1)$ Gitterpunkten (ohne Rand aus $\prod_{k=1}^d (2^{l_k} - 1)$). In Abbildung 2.1b ist ein anisotropes volles Gitter einschließlich der Randpunkte mit $\vec{l} = (2, 4)$ dargestellt.

Ein Interpolant $u_{\vec{l}}$ über dem Gitter $\Omega_{\vec{l}}$ soll an den Gitterpunkten die exakten Funktionswerte $f(x_{\vec{l},\vec{j}})$ zurückliefern und zwischen den Maschen eine Näherung an die korrekte Lösung berechnen. Ein solches Verhalten kann durch eine Linearkombination aus gewichteten Ansatzfunktionen $\phi_{\vec{l},\vec{j}}$ erreicht werden.

$$(2.6) \quad u_{\vec{l}}(\vec{x}) = \sum_{\vec{0} \leq \vec{j} \leq 2^{\vec{l}}} w_{\vec{l},\vec{j}} \cdot \phi_{\vec{l},\vec{j}}.$$

Dabei liefert $\phi_{\vec{l},\vec{j}}$ am Gitterpunkt $x_{\vec{l},\vec{j}}$ den Wert 1 und das Gewicht $w_{\vec{l},\vec{j}}$ bestimmt den entsprechenden Beitrag zum Interpolanten. Die Ansatzfunktionen entsprechen meist Polynomen unterschiedlicher Ordnung, jedoch beschränken wir uns im Folgenden ausschließlich auf stückweise lineare Funktionen, den sogenannten Standard-Hut-Funktionen. Diese besitzen im eindimensionalen folgende Form, und werden mittels Skalierung und Translation einem lokalen Bereich zugeordnet.

$$(2.7) \quad \phi_{l,j}(x) := \max \{ (1 - |2^l x - j|), 0 \}.$$

Für den eindimensionalen Fall sind die Ansatzfunktionen $\phi_{3,j}$, in Abbildung 2.1a, links oben dargestellt. Die d -dimensionale Hutfunktion $\phi_{\vec{l},\vec{j}}(\vec{x})$ wird dabei durch eine Tensorprodukt-Konstruktion aus d eindimensionalen Ansatzfunktionen dargestellt, welche wie folgt definiert ist.

$$(2.8) \quad \phi_{\vec{l},\vec{j}}(\vec{x}) := \prod_{k=1}^d \phi_{l_k, j_k}(x_k).$$

Durch Linearkombination lässt sich folglich der Raum aller stückweise d -linearen Funktionen $V_{\vec{l}}$ definieren, wobei $u_{\vec{l}} \in V_{\vec{l}}$ gilt.

$$(2.9) \quad V_{\vec{l}} := \text{span} \{ \phi_{\vec{l},\vec{j}} : \vec{0} \leq \vec{j} \leq 2^{\vec{l}} \}.$$

Entsprechen die Gewichte $w_{\vec{l},\vec{j}}$ genau den Funktionswerten an den Stützstellen $f(x_{\vec{l},\vec{j}})$, so spricht man von einer knotenbasierten oder nodalen Basis. Eine Funktion f lässt sich somit in der nodalen Basis durch folgende Summe darstellen.

$$(2.10) \quad f(\vec{x}) \approx u_{\vec{l}}(\vec{x}) = \sum_{\vec{0} \leq \vec{j} \leq 2^{\vec{l}}} f(x_{\vec{l},\vec{j}}) \cdot \phi_{\vec{l},\vec{j}}.$$

In Abbildung 2.1a auf der linken Seite ist der Interpolant u_3 einer Funktion (rot) bezüglich der nodalen Basis dargestellt. Auffällig ist hierbei, dass sich die Ansatzfunktionen teilweise überlappen, jedoch an allen bis auf dem entsprechenden Gitterpunkt den Wert 0 besitzen.

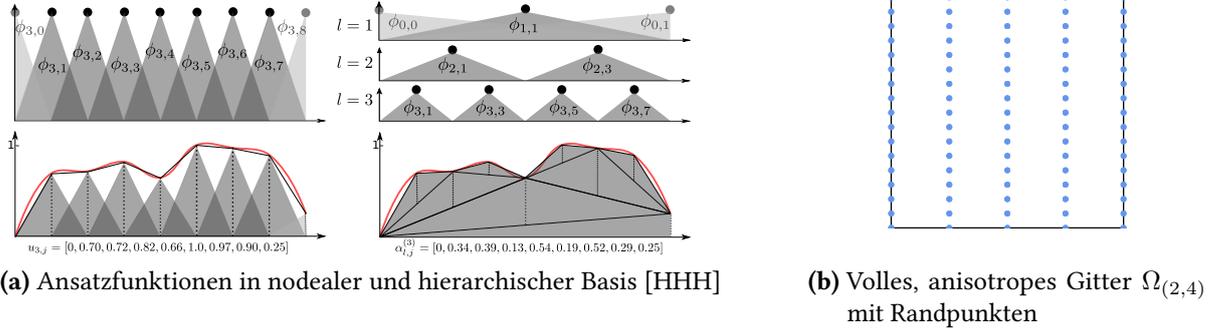


Abbildung 2.1: Diskretisierung von Funktionen

Eine Alternative zur Funktionsdarstellung mit der nodalen Basis, ist die Darstellung mit einer hierarchischen Basis. Dabei werden ähnlich wie zuvor die Werte an den Stützstellen $x_{\vec{l},\vec{j}}$ durch eine gewichtete Linearkombination aus Ansatzfunktionen dargestellt. Anders als bisher wird die Form der Ansatzfunktionen hierbei nicht abhängig vom zugrundeliegenden Auflösungsgrad der Diskretisierung definiert, sondern besitzt abhängig vom zugeordneten hierarchischen Teilraum eines Levels \vec{l} , eine spezielle, nicht überlappende Gestalt. Die hierarchischen Teilräume $W_{\vec{l}}$ sind wie folgt definiert:

$$(2.11) \quad W_{\vec{l}} := \text{span}\{\phi_{\vec{l},\vec{j}} : \vec{j} \in I_{\vec{l}}\}.$$

Wobei die Indexmenge $I_{\vec{l}}$, welche die Positionen der Ansatzfunktionen innerhalb des Teilraums $W_{\vec{l}}$ beschreibt, wie folgt definiert ist:

$$(2.12) \quad I_{\vec{l}} := \{\vec{j} : 1 \leq j_k \leq 2^{l_k} - 1, \text{ mit } j_k \text{ ungerade und } 1 \leq k \leq d\}.$$

In Abbildung 2.1a sind rechts oben, die Ansatzfunktionen der Teilräume W_1 bis W_3 abgebildet. Mithilfe der Teilräume $W_{\vec{l}}$ lässt sich folglich der Raum der stückweise d -linearen Funktionen $V_{\vec{i}}$ auf dem vollen Gitter $\Omega_{\vec{i}}$ mit Maschenweiten $h_{\vec{i}}$ durch eine direkte Summe über alle Level $\vec{l} \leq \vec{i}$ beschreiben.

$$(2.13) \quad V_{\vec{i}} := \bigoplus_{\vec{l} \leq \vec{i}} W_{\vec{l}}.$$

Ähnlich wie in (2.10) kann der Interpolant einer Funktion $u_{\vec{i}} \in V_{\vec{i}}$, auf dem vollen Gitter $\Omega_{\vec{i}}$, nun als Linearkombination aus gewichteten Ansatzfunktionen aller hierarchischer Teilräume $W_{\vec{l}}$ mit $\vec{l} \leq \vec{i}$ dargestellt werden.

$$(2.14) \quad f(\vec{x}) \approx u_{\vec{i}}(\vec{x}) = \sum_{\vec{l} \leq \vec{i}} \sum_{\vec{j} \in I_{\vec{l}}} \alpha_{\vec{l},\vec{j}}^{(\vec{i})} \cdot \phi_{\vec{l},\vec{j}}(\vec{x}).$$

Dabei beschreibt die Gleichung (2.14) die Darstellung einer Funktion anhand der hierarchischen Basis. In Abbildung 2.1a ist auf der rechten Seite der Interpolant u_3 einer Funktion in

hierarchischer Basis dargestellt. Die Gewichte $\alpha_{\vec{l},\vec{j}}^{(i)}$ sind auch als hierarchische Überschüsse bekannt, und können im eindimensionalen Fall anhand der Funktionswerte an benachbarten Stützstellen durch Anwendung des eindimensionalen Stempels, wie folgt berechnet werden:

$$(2.15) \quad \begin{aligned} \alpha_{l,j}^{(i)} &:= \left[-\frac{1}{2} \ 1 \ -\frac{1}{2}\right]_{l,j} u_i(x_{l,j}) \\ &= u_i(x_{l,j}) - \frac{u_i(x_{l,j-1}) + u_i(x_{l,j+1})}{2}. \end{aligned}$$

Die Berechnung der hierarchischen Überschüsse aus den Funktionswerten an den Stützstellen, ist als Hierarchisierung bekannt. Diese kann im mehrdimensionalen Fall durch einen d -dimensionalen Stempel, welcher ein Produkt aus d eindimensionalen Stempeln in die jeweiligen Raumrichtungen darstellt, realisiert werden.

$$(2.16) \quad \alpha_{\vec{l},\vec{j}}^{(i)} := \prod_{k=1}^d \left[-\frac{1}{2} \ 1 \ -\frac{1}{2}\right]_{l_k,j_k} u_{i_k}(x_{l_k,j_k}).$$

Anstatt den Stempel direkt auszurechnen, lässt sich eine Lösung auf einem vollen Gitter $\Omega_{\vec{i}}$ auch hierarchisieren, indem der eindimensionale Stempel auf die Werte an den Stützstellen nacheinander für jede Raumrichtung angewendet wird. Ein Beispiel des Algorithmus lässt sich für den zweidimensionalen Fall in [GSZ] finden, wobei dieser leicht auf den mehrfachen Fall erweitert werden kann. Die Umkehroperation ist als Dehierarchisierung bekannt und berechnet aus der hierarchischen eine knotenbasierte Darstellung. Ausgehend von den hierarchischen Teilräumen lassen sich schlussendlich die dünnen Gitter beschreiben.

Während eine Lösung $u_{\vec{n}}$ über dem vollen Gitter $\Omega_{\vec{n}}$ mit Maschenweiten $h_{\vec{n}}$, nach (2.14), aus der Summe über alle gewichteten Ansatzfunktionen der hierarchischen Teilräume $W_{\vec{l}}$ mit $\vec{l} \leq \vec{n}$ besteht, ergibt sich eine Lösung u_n^s auf einem dünnen Gitter Ω_n^s nur aus einem Teil der gewichteten Ansatzfunktionen. Der Raum V_n^s aller stückweise d -linearen Funktionen auf dem klassischen dünnen isotropen Gitter Ω_n^s , lässt sich nach [HHH] wie folgt definieren:

$$(2.17) \quad V_n^s := \bigoplus_{|\vec{l}|_1 \leq n+1} W_{\vec{l}}.$$

Wird eine d -dimensionale Funktion auf dem isotropen, vollen Gitter $\Omega_{\vec{n}}$ diskretisiert, müssen die Funktionswerte an allen $(n+1)^d$ Stützstellen gespeichert werden. Dadurch entsteht ein exponentieller Platzverbrauch von $\mathcal{O}(h_n^{-d})$, wobei der Approximationsfehler $\mathcal{O}(h_n^2)$ bei einer ausreichend glatten Funktion nach [Pfl10] mit zunehmend kleiner werdender Maschenweite exponential verringert wird. Im Falle einer Dünngitterlösung müssen nach [Gar13] lediglich $\mathcal{O}(h_n^{-1} \cdot \log(h_n^{-1})^{d-1})$ gespeichert werden (man bemerke, dass der exponentielle Anteil hier durch den Logarithmus abgeschwächt wird), wobei der Approximationsfehler mit $\mathcal{O}(h_n^2 \cdot \log(h_n^{-1})^{d-1})$ nur geringfügig verschlechtert wird.

Da die meisten Verfahren zur Lösung von PDEs ihre Berechnungen auf vollen Gittern durchführen und es einen enormen Aufwand mit sich bringt, diese speziell auf dünne Gitter anzupassen, wurde von im Laufe der Jahre die sogenannte Dünngitter Kombinationstechnik

(SGCT) entwickelt [GSZ]. Die Idee ist dabei durch eine Kombination aus mehreren anisotropen Vollgitterlösungen $u_{\vec{i}}$ welche, im Folgenden als Kombinationslösungen und die zugrunde liegenden Gitter als Kombinationsgitter beschrieben werden, eine isotrope Dünngitterlösung $u_n^{(c)}$ zu erzeugen. Die kombinierte Dünngitterlösung, unterscheidet sich im Allgemeinen von derjenigen, welche tatsächlich auf einem dünnen Gitter berechnet wurde. Jedoch ist der Unterschied, sofern die PDE einen gewissen Grad an Glattheit aufweist, vernachlässigbar [GSZ]. Der Kombinationsschritt der SGCT kann wie folgt durch eine gewichtete Summe der Kombinationslösungen beschrieben werden:

$$(2.18) \quad u_n^{(c)} = \sum_{\vec{i} \in \mathcal{I}} c_{\vec{i}} \cdot u_{\vec{i}}.$$

Dabei wird durch ein Gewicht $c_{\vec{i}}$ bestimmt, wie stark die zugehörige Kombinationslösung die kombinierte Dünngitterlösung beeinträchtigt. Die Indexmenge \mathcal{I} hingegen beschreibt die Auflösungen der verwendeten Kombinationsgitter, wobei durch die Wahl der verwendeten Kombinationslösungen die Güte der Approximation stark beeinflusst werden kann. Bei der klassischen Kombinationstechnik sind die beiden Parameter wie folgt bestimmt.

$$(2.19) \quad u_n^{(c)} = \underbrace{\sum_{q=0}^{d-1} (-1)^q \binom{d-1}{q}}_{=c_{\vec{i}}} \sum_{\vec{i} \in \mathcal{I}_{n,q,\tau}} u_{\vec{i}}$$

Wobei die Indexmenge $\mathcal{I}_{n,q,\tau}$ wie folgt definiert ist:

$$(2.20) \quad \mathcal{I}_{n,q,\tau} := \{ \vec{i} \in \mathbb{N}_d : |\vec{i}|_1 = |\vec{i}_{\min}|_1 + \tau - q \text{ mit } \vec{i}_{\min} = \vec{n} - \tau \cdot \vec{1} > \vec{0} \text{ und } \vec{i} \geq \vec{i}_{\min} \}$$

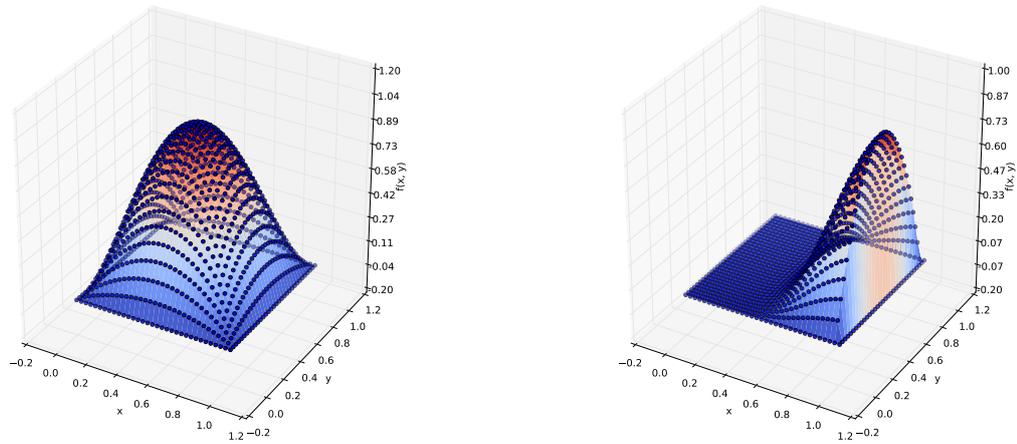
Der Parameter τ gibt dabei Ausdünnungsgrad der kombinierten Dünngitterlösung an. In den folgenden dargestellten und behandelten Kombinationsgitter, wurde τ dabei auf den Wert $n - 1$ festgelegt, was zur Folge hat, dass die Auflösung der kombinierten Lösungen folgende Bedingung erfüllen: $|\vec{i}| \in \{n, n + 1\}$. Bei der Kombination zweier Kombinationslösungen werden dabei die Werte an den Stützstellen verrechnet, welche in beiden zugehörigen Gittern auftreten. Dies kann entweder durch Interpolation auf ein Gitter mit gemeinsamer Maschenweite geschehen, oder es werden wie in [HHJP15] beschrieben direkt die übereinstimmenden hierarchischen Koeffizienten miteinander verrechnet.

3 Silent Faults und Kombinationstechnik

Innerhalb dieses Kapitels wird das Auftreten von Silent Faults bei der Lösung von PDEs mit der Dünngitter Kombinationstechnik erläutert. Dabei wird zunächst allgemein geklärt wie PDEs mit der SGCT parallel auf HPC gelöst werden und in welchen Schritten Fehler auftreten können. Weiterhin wird anhand eines Beispiels erklärt wie sich die SGCT im Falle eines Fehlers bei der Lösung von PDEs verhält und wo Möglichkeiten bestehen, Fehler zu erkennen. Darauf folgend werden die beiden von Hinojosa et. al. in [HHH] vorgestellten Fehlererkennungsverfahren beschrieben, welche bereits einen Großteil an Fehlern im Kombinationsschritt erkennen. Fehler, die mit diesen Verfahren nicht oder nicht direkt erkannt werden können, sind Teil dieser Arbeit und werden zuletzt vorgestellt.

Betrachten wir zunächst das Verfahren wie PDEs mit der Kombinationstechnik auf HPC Systemen gelöst werden können. Um das Verfahren zu veranschaulichen, wird dies am Beispiel der Advektionsgleichung $v_1 u_x + v_2 u_y = u_t$ mit zwei räumlichen und einer zeitlichen Dimension dargestellt. Die Anfangsbedingungen bilden dabei das Polynom $u(x, y, t = 0) = 16 \cdot x \cdot (x - 1) \cdot y \cdot (y - 1)$ zweiten Grades über dem Gebiet $\Omega = [0, 1]^2$ und Dirichlet Randbedingungen mit $\partial\Omega = 0$. Die Parameter v_1 und v_2 bestimmen dabei die Geschwindigkeit der Advektion in x und y Richtung und sind im Folgenden auf die Werte $v_1 = 0.1$ und $v_2 = 0$ gesetzt. In Abbildung 3.1 ist die PDE links zum initialen Zeitschritt $t = 0$ und rechts nach $T = 100$ Zeitschritten auf dem vollen Gitter $\Omega(5, 5)$ dargestellt.

Zur Lösung der PDE mithilfe der Kombinationstechnik, muss diese zunächst auf die einzelnen Kombinationsgitter diskretisiert werden und kann dann für jedes der Gitter separat für einige Zeitschritte mit einem Löser z.B. dem Upwind-Verfahren, gelöst werden. Dies ermöglicht eine parallele Berechnung der einzelnen Teillösungen, da diese während der Zeit im PDE Löser völlig unabhängig voneinander sind. Das Problem bei zeitabhängigen PDEs ist jedoch, dass die Lösungen mit der Zeit divergieren können und die kombinierte Dünngitterlösung somit verfälscht wird. Aus diesem Grund sollten die Kombinationsgitter nach einigen Zeitschritten kombiniert werden, um dann aus der kombinierten Lösung wieder die einzelnen Teillösungen gewinnen zu können [GHZ96]. Dadurch lässt sich erreichen, dass alle Kombinationsgitter auf einem gemeinsamen Stand sind, ähnlich wie zum Zeitpunkt $t = 0$. Das gesamte Lösungsverfahren kann durch ein Master/Slave Verfahren implementiert werden. Wobei die Arbeiter-Knoten die Lösung der PDE für ein einzelnes Kombinationsgitter berechnen und der Master-Knoten die



(a) PDE zum Zeitpunkt $t = 0$

(b) PDE nach $T = 100$ Zeitschritten

Abbildung 3.1: Entwicklung eines Advektionsprozesses über die Zeit

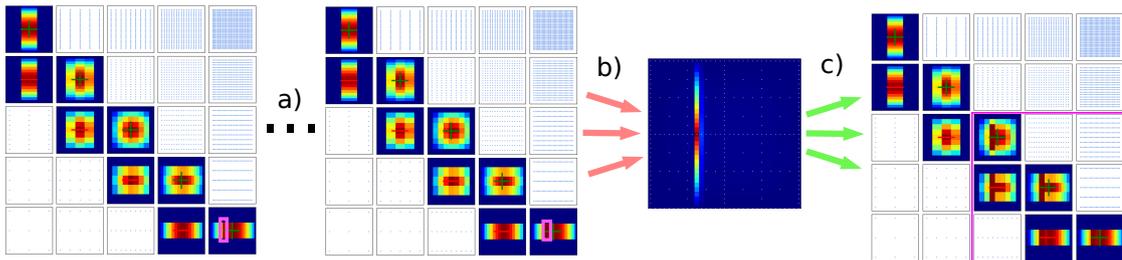


Abbildung 3.2: Mögliche Schritte bei der Berechnung der SGCT

a) stellt dabei den Berechnungsschritt, b) den Kombinationsschritt und c) den Verteilungsschritt dar. Weiterhin wurde im Gitter mit $\vec{i} = (5, 1)$ ein Fehler eingeführt, welcher sich im Verteilungsschritt auf weitere Kombinationslösungen ausbreitet (pink umrahmt).

Aufteilung der Teillösungen auf die einzelnen Arbeiter-Knoten bestimmt sowie den Kombinationsschritt einleitet. [HHH; HHLS15] In Abbildung 3.2 ist das Verfahren für einen Berechnungs-, einen Kombinations- sowie einen Verteilungsschritt dargestellt.

Um zwei Gitter mit unterschiedlichen Auflösungen \vec{l}, \vec{k} zu kombinieren, bedarf es im naiven Fall einer Interpolation auf ein Gitter mit gemeinsamer Auflösung $\vec{\lambda} = \max\{\vec{l}, \vec{k}\}$, wobei die Funktionswerte dann an den Gitterpunkten direkt verrechnet werden können. Ein solches Verfahren erfordert jedoch einen hohen Berechnungs- und Kommunikationsaufwand, welcher nur durch Ausnutzung der hierarchischen Struktur effizient bewältigt werden kann [HHJP15]. Ein weiteres Problem bei der parallelen Berechnung ist die richtige Verteilung

der Kombinationsgitter auf die einzelnen Arbeiter-Knoten. Da die Laufzeit zur Lösung der PDE auf einem Kombinationsgitter stark von der Anzahl an Gitterpunkten und der Anisotropie des Gitters abhängt, entstehen unterschiedliche Anforderungen an Speicherplatz und Berechnungszeiten [HKP14]. Da auf HPC Systemen Fehler in Zukunft eher den Normalfall als die Ausnahme darstellen, müssen Algorithmen fehlertolerant implementiert werden. Eine Elegante Möglichkeit für die Kombinationstechnik bietet die fehlertolerante Erweiterung FTCT (Fault Tolerant Combination Technique) [HHL15], welche auch im Ausfall von einzelnen Gittern eine Lösung liefert, ohne dabei Gitter erneut berechnen zu müssen. Die Grundidee des Verfahrens ist es die Menge der Kombinationsgitter durch Berechnung zusätzlicher Gitter mit niedrigerer Auflösung so zu erweitern, dass im Falle eines Fehlers das betroffene Gitter durch eine angepasste Kombination ersetzt werden kann. Während die in 2 beschriebene Kombination die Gitter $u_{\vec{i}}$ mit Auflösung $\vec{i} \in \mathcal{I}_{l,0,\tau} \cup \dots \cup \mathcal{I}_{l,d,\tau}$ kombiniert, wird diese in der FTCT durch zusätzlich Auflösungen $\vec{i} \in \mathcal{I}_{l,0,\tau} \cup \dots \cup \mathcal{I}_{l,d+1,\tau} = \mathcal{I}_{l,q,\tau}^{ext}$ erweitert. Die Koeffizienten $c_{\vec{i}}$ der hinzugenommenen Gitter werden dabei ausschließlich im Fehlerfall $\neq 0$ gesetzt. Die eben beschriebenen Herausforderungen werden von Hinojosa et. al. in [HHH] in die drei Kategorien: Communication, Load Balancing und Fault Tolerance eingeteilt. Sie bilden den Schlüssel zu einer effizienten Anwendung der Kombinationstechnik auf parallelen Systemen.

Da das Verfahren grundsätzlich in zwei Schritte aufgeteilt werden kann nämlich dem Lösen der PDE auf den einzelnen Knoten und dem Kommunizieren der Lösung, während des Kombinierens und Verteilens, sind auch genau dort jene Stellen, an denen sich SDCs einschleichen können. Da die meiste Zeit für das Lösen der PDE auf den einzelnen Kombinationsgitter benötigt wird, ist hier die Chance von unbemerkt auftretenden Fehlern am höchsten [HHH]. Tritt hierbei ein Fehler auf, welcher den Wert an einem Gitterpunkt $u_{\vec{i}}(x_{i,j}^{\vec{i}})$ verfälscht, so kann dieser, abhängig von der zugrundeliegenden PDE, umliegende Gitterpunkte beeinflussen. Im Falle der zuvor beschriebenen Advektionsgleichung würde ein korrupter Wert mit der Zeit in x-Richtung wandern ohne sich dabei zu verändern, was in Abbildung 3.2 gut zu erkennen ist. Um den Fehler besser erkennen zu können, wurde dabei Skalierung der Farbwerte auf der kombinierten Lösung anders gewählt als auf den Kombinationsgittern. Anders verhält sich beispielsweise eine PDE mit einem Diffusionsterm. In diesem Fall verteilt sich der Fehler in der Zeit auf alle umliegenden Gitterpunkte und verliert dabei an Größe. Wird der Fehler frühzeitig erkannt, so reicht es aus das betroffene Gitter zu ersetzen (FTCT), ohne die gesamte Simulation unterbrechen zu müssen. Wird der Fehler nicht erkannt, befällt dieser im Kombinationsschritt die Dünngitterlösung und wird im schlimmsten Fall durch den anschließenden Verteilungsschritt auf die einzelnen Kombinationsgitter verstreut, was eine weitere Berechnung unbrauchbar macht. Dabei wird der Fehler während der Kombination zusätzlich verstärkt, da die Werte an den Gitterpunkten $u_{\vec{i}}(x_{i,j}^{\vec{i}})$ nach 2.19 mit dem Faktor $c_{\vec{i}}$ multipliziert werden [HHH]. Aus diesem Grund ist es von Nöten den Fehler spätestens vor dem Kombinationsschritt zu erkennen um entsprechende Maßnahmen einleiten zu können.

3.1 Erkennung von SDC nach Hinojosa

Hinojosa et.al. stellt hierzu in [HHH] zwei Verfahren vor, welche direkt in der Kombination die zugehörigen Kombinationsgitter $u_{\vec{t}}$ vergleichen, um so das Auftreten von SDC aufzudecken. Dabei wird die natürliche Redundanz der Kombinationsgitter ausgenutzt, um Fehler zu erkennen. Hinojosa et. al. untersucht zunächst wie groß der Unterschied zwischen den Kombinationsgittern theoretisch sein kann. Aus der error splitting assumption (ESA) folgert er, dass die Differenz zwischen demselben Gitterpunkt in zwei unterschiedlichen Kombinationsgittern beschränkt ist. Jedoch hängt diese Schranke vom zugrundeliegenden Problem ab und lässt sich höchstens abschätzen sofern alle $u_{\vec{t}}$ korrekt berechnet wurden. Aus diesem Grund sucht Hinojosa zwischen den unterschiedlichen Versionen eines Gitterpunktes nach Außenseitern welche sich stark von den anderen unterscheiden. Da sich die Gitterpunkte am einfachsten in hierarchischer Basis miteinander vergleichen lassen und so die Möglichkeit besteht, den Vergleich direkt im Kombinationsschritt zu vollziehen, stellt Hinojosa et. al. [HHH] seine beiden Verfahren bezüglich der hierarchischen Koeffizienten vor.

3.1.1 Paarweiser Vergleich

Seien $u_{\vec{s}}$ und $u_{\vec{t}}$ ein Paar von Kombinationslösungen mit zugehörigen Werten an den Gitterpunkten $\alpha_{\vec{l},\vec{j}}^{(\vec{s})}$ und $\alpha_{\vec{l},\vec{j}}^{(\vec{t})}$ in Hierarchischer Basis gegeben. Dann kann für jeden Gitterpunkt der normalisierte Unterschied wie folgt ausgedrückt werden:

$$(3.1) \quad \beta_{\vec{l},\vec{j}}^{(\vec{s},\vec{t})} := \frac{|\alpha_{\vec{l},\vec{j}}^{(\vec{s})} - \alpha_{\vec{l},\vec{j}}^{(\vec{t})}|}{\min\{|\alpha_{\vec{l},\vec{j}}^{(\vec{s})}|, |\alpha_{\vec{l},\vec{j}}^{(\vec{t})}|\}} \quad \forall \vec{l} \leq \vec{s}, \vec{t} \text{ und } \vec{0} \leq \vec{j} \leq 2^{\vec{l}}$$

Die Normalisierung wirkt dabei einerseits dem hierarchischen Abfall der Koeffizienten entgegen. Andererseits wird der Unterschied durch die Division mit dem kleineren Wert zusätzlich verstärkt. Da es genügt zu wissen ob ein Wert falsch berechnet wurde, wird für spätere Vergleiche lediglich der maximale Unterschied über alle Gitterpunkte $x_{\vec{l},\vec{j}}$ gespeichert [HHH].

$$(3.2) \quad \beta^{(\vec{s},\vec{t})} := \max_{\vec{l} \leq \vec{s}, \vec{t}} \max_{\vec{0} \leq \vec{j} \leq 2^{\vec{l}}} \beta_{\vec{l},\vec{j}}^{(\vec{s},\vec{t})}$$

Das Verfahren von Hinojosa et.al. berechnet zunächst für alle Paare (\vec{s}, \vec{t}) mit $\vec{s}, \vec{t} \in \mathcal{I}_{\vec{n}}$ und $\vec{s} \neq \vec{t}$ den zugehörigen maximalen Unterschied $\beta^{(\vec{s},\vec{t})}$. Anschließend werden die Werte der Gitterpaare miteinander verglichen und auf Ausreißer untersucht. Kommt ein Gitter in mehreren Gitterpaaren mit abnormal großen Werte (Ausreißern) vor, so kann meist davon ausgegangen werden, dass es sich um ein korruptes Gitter handelt.

3.1.2 Vergleich auf hierarchischer Ebene

Während im vorherigen Verfahren der Vergleich von Gitterpaaren eine zentrale Rolle spielt, wird im Folgenden die Idee verfolgt, Unterschiede zwischen den Versionen einer hierarchischen Koeffiziente auszunutzen, um fehlerhafte Gitter aufzudecken. Sei $\alpha_{\vec{l},\vec{j}}$ eine hierarchische Koeffiziente am Gitterpunkt $x_{\vec{l},\vec{j}}$. dann kommt $\alpha_{\vec{l},\vec{j}}$ in allen Kombinationslösungen $u_{\vec{i}}$ mit $\vec{i} \geq \vec{l}$ vor. Das bedeutet, dass es von jedem Gitterwert mindestens 1 und höchstens $|\mathcal{S}|$ unterschiedliche Versionen gibt. So kommen beispielsweise die hierarchischen Gitterwerte auf unterster Ebene mit $\vec{l} = \vec{1}$ in allen Kombinationslösungen vor. Sei $N_{\vec{l}} \in \{1, \dots, |\mathcal{S}|\}$ die Anzahl an Kombinationslösungen, welche die hierarchische Koeffiziente $\alpha_{\vec{l},\vec{j}}$ beinhalten. Aus der ESA kann gefolgert werden, dass die Varianz der $N_{\vec{l}}$ unterschiedlichen Versionen $\alpha_{\vec{l},\vec{j}}$ beschränkt ist und demnach der Unterschied zwischen den $\alpha_{\vec{l},\vec{j}}$ nicht ungemein hoch sein kann. Sollte sich demnach im Fehlerfall der Wert an einem Gitterpunkt stark ändern, so weicht dieser auch stark vom Erwartungswert ab und kann als Ausreißer eingestuft werden.

Ein Problem, welches sich in beiden Verfahren ergibt, bilden Gitterwerte $\alpha_{\vec{l},\vec{j}}$, die in der Kombination nur einmalig auftreten, also für welche $N_{\vec{l}} = 1$ gilt. Diese Gitterwerte treten in solchen Kombinationslösungen $u_{\vec{i}}$ mit $|\vec{i}|_1 = n + 1$ auf (liegen auf der Hauptdiagonalen der Kombinationstechnik, welche in Abbildung 3.2 abgebildet ist). Für diese Gitterwerte gibt es keine Redundanzen. Dies bedeutet, dass, falls eine SDC einen solchen Gitterwert beeinflusst und sich der Fehler nicht auf andere Gitterpunkte ausbreitet, so kann der Fehler mit den zuvor beschriebenen Verfahren nicht erkannt werden. Hinojosa et. al. [HHH] stellt für diesen Fall eine Möglichkeit vor, welche den hierarchischen Abfall der Koeffizienten ausnutzt, um Fehler zu erkennen. Bei hinreichend glatten Funktionen sollte der hierarchisch Überschuss mit zunehmendem Level exponentiell sinken. Ein einfacher Test welcher dies überprüft wäre ein Vergleich der hierarchischen Koeffizienten auf dem höchsten Level mit einem um m -Level darunterliegenden hierarchischen Koeffizienten zu vergleichen:

$$(3.3) \quad |\alpha_{\vec{l},\vec{j}}| < |\alpha_{(\vec{l}-\vec{e}_m),\vec{j}}|, \text{ wobei } \vec{e} \text{ der } k\text{-te Einheitsvektor ist.}$$

Ist die Ungleichung nicht erfüllt, so liegt ein Fehler vor. Hinojosa schlägt für m in [HHH] einen Wert von 3 vor.

3.2 Erkennung von SDC anhand der zeitlichen Entwicklung

Während Hinojosa et. al. die natürliche Redundanz der Kombinationsgitter ausnutzt um Fehler zu erkennen werden im Folgenden Methoden vorgestellt um Fehler ohne den Vergleich benachbarter Teilräume zu erkennen. Dabei spielt die zeitliche Entwicklung einer Kombinationslösung eine zentrale Rolle. Die zugrunde liegende Idee ist ein Kombinationsgitter, welches eine PDE $f(t, \vec{x})$ zu einem Zeitschritt t_s mit $s \in \{1, \dots, T\}$ beschreibt, mit dem selben Gitter zu früheren

Zeitpunkten t_{s-q} , mit $s > q$ zu vergleichen. Ziel ist es im Falle eines Silent Faults, ein ungewöhnliches Verhalten der Gitterpunkte möglichst frühzeitig zu erkennen, um Berechnungen mit korrupten Daten weitestgehend zu vermeiden.

3.2.1 Änderung der Gitterwerte

Bei der Untersuchung der Kombinationsgitter während der Simulation einer Wärmeleitung mit ausreichend glatten Anfangsbedingungen fällt auf, dass sich die Werte an den Gitterpunkten nicht willkürlich ändern. Vielmehr ist ein Trend zu erkennen. So ist beispielsweise im Falle einer Diffusion, der exponentielle Abfall der Gitterpunkte erkennbar. Tritt jedoch eine SDC an einem Gitterpunkt auf, ändert sich der zugehörige Wert schlagartig und führt auch im weiteren Verlauf zu abnormalen Änderungen an umliegenden Werten. Dies erlaubt eine Erkennung von Fehlern anhand einer unerwarteten Änderungsrate der einzelnen Gitterpunkte. Sei $u_{\vec{i}}^{(t_s)}(x_{\vec{i},\vec{j}})$ der Wert an einem Gitterpunkt $x_{\vec{i},\vec{j}}$ einer Kombinationslösung zum Zeitpunkt t_s . Die Steigung zwischen dem Wert $u_{\vec{i}}^{(t_s)}(x_{\vec{i},\vec{j}})$ und einem Wert $u_{\vec{i}}^{(t_{s-q})}(x_{\vec{i},\vec{j}})$ zu einem früheren Zeitschritt gibt an, wie stark sich der Wert in der Zeit verändert. Um kleinen Wertschwankungen entgegen zu wirken, kann die Änderung über die letzten $p \geq 1$ Zeitschritte gemittelt werden. Dies führt außerdem dazu, dass starke Änderungen auch noch nach p Zeitschritten gemessen werden können, da sich diese auf die gemittelte Änderung auswirken. Die gemittelte Änderung $\Delta_{\vec{i},p}^{(t_s)}(x_{\vec{i},\vec{j}})$ an einem Gitterpunkt, kann wie folgt berechnet werden.

$$(3.4) \quad \Delta_{\vec{i},p}^{(t_s)}(x_{\vec{i},\vec{j}}) := p^{-1} \cdot \sum_{k=1}^p \frac{|u_{\vec{i}}^{(t_s)}(x_{\vec{i},\vec{j}}) - u_{\vec{i}}^{(t_{s-k})}(x_{\vec{i},\vec{j}})|}{k}, \text{ mit } i > p \text{ und } s, p \in \mathbb{N}$$

Ähnlich wie in 3.1.1 interessiert uns im Folgenden nur die maximale Änderung über alle Gitterwerte, wobei pro Kombinationsgitter zu einem Zeitpunkt t_s für spätere Vergleiche somit nur ein zusätzlicher Wert gespeichert werden muss.

$$(3.5) \quad \Delta_{\vec{i},p}^{(t_s)} := \max_{\vec{0} \leq \vec{j} \leq \vec{2}^{\vec{i}}} \Delta_{\vec{i},p}^{(t_s)}(x_{\vec{i},\vec{j}})$$

Tritt ein Silent Fault auf, welcher den Wert an einem Gitterpunkt in einem Kombinationsgitter signifikant beeinflusst, ist die Änderung an diesem Gitterpunkt, verglichen mit früheren Zeitpunkten ungewöhnlich groß. Unter der Annahme, dass die Änderungen an den Gitterpunkten $\Delta_{\vec{i},p}^{(t_s)}(x_{\vec{i},\vec{j}})$ im fehlerfreien Fall etwa in der selben Größenordnung liegen¹, hebt sich eine ungewöhnlich starke Änderung an einem Gitterpunkt von den anderen Gitterpunkten ab

¹dies hat sich bei der innerhalb dieser Arbeit betrachteten Advektions- und Diffusionsproblemen herausgestellt

und bildet somit die maximale Änderung im Gitter. Um eine ungewöhnliche Entwicklung der maximalen Änderung erkennen zu können, muss nach jedem Zeitschritt $\Delta_{i,p}^{(t_s)}$ für eine Kombinationslösung berechnet und gespeichert werden. Für den Vergleich der Gitter werden weiterhin alle $p - 1$ Lösungen zu vorherigen Zeitschritten benötigt. Zur Speicherung der Historie $H_i = \{u_i^{(t_{s-p})}, \dots, u_i^{(t_{s-1})}\}$ einer Kombinationslösung $u_i^{(t_s)}$ entsteht somit ein Aufwand von $\mathcal{O}((p-1) \cdot n^d)$, wobei $u_i^{(c)}$ die kombinierte Dünngitterlösung und d die Anzahl an Unbekannten darstellt. Um den Speicherbedarf zu minimieren, bietet es sich an, lediglich für die Gitter mit den höchsten Auflösungen die Historie zu speichern, da sich dort die Gitterpunkte befinden, welche keine natürliche Redundanzen aufweisen und zumindest von zwei der von Hinojosa et al. [HHH]beschriebenen Verfahren nicht direkt erkannt werden können. Die Wahl von p hängt dabei stark von der Schrittzahl im PDE Löser ab. Da die Kombinationslösungen durch den Verteilungsschritt, welcher im Anschluss zur Kombination erfolgt, aktualisiert werden, können sich die Gitterpunkte mehr oder weniger stark ändern. Tritt eine unerwartete Änderung auf, so kann es zur fälschlichen Erkennung von Ausreißern kommen. Um dies zu vermeiden ist es sinnvoll nach jedem Kombinationsschritt alle früheren $\Delta_{i,p}^{(t_{s-q})}$ mitsamt der Historie H_i zu verwerfen und neu zu erstellen. Ein Nachteil des Verfahren ist demnach, dass nach einem Kombinationsschritt mindestens p Zeitschritte lang kein Fehler erkannt werden kann, da keine zuverlässigen Vergleichswerte vorliegen.

In 3.1 ist das Verfahren anhand eines Pseudocodes dargestellt. Dabei wird in Zeile 3 eine maximale Änderung aus vorherigen Werten prognostiziert und diese im nächsten Schritt mit der tatsächlichen Änderung verglichen. Ist der Unterschied dabei größer als ein festgelegter Schwellwert, wird der Index der fehlerhaften Lösung zurückgegeben. Die Prognose kann dabei beispielsweise durch eine lineare Regression aus den vorherigen Werten berechnet werden. Weitere Methoden um die zeitliche Entwicklung eines Wertes (hier $\Delta_p^{(t_s)}$) nach Ausreißern zu untersuchen, werden in [Hul] aufgeführt. Die vorgestellten Verfahren beruhen auf sogenannten autoregressiven Modellen, welche sich besonders dazu eignen, Prognosen für einen größeren Zeitraum zu erstellen.

3.2.2 Abweichung vom Interpolanten

Eine weitere Möglichkeit, Fehler anhand der zeitlichen Entwicklung eines Kombinationsgitters feststellen zu können, ist es die Glattheitseigenschaft der zugrundeliegenden PDE über die Zeit zu verfolgen. Während die korrekte Lösung einer Differentialgleichung u ausreichende Glattheit aufweisen muss, um den Fehler $|u - u_n^{(c)}|$ zwischen der kombinierten und der exakten Lösung auf ein Minimum (wie bei einer reinen Interpolation) zu beschränken [GSZ], kann es bei einem Auftreten eines Silent Faults dazu kommen, dass diese Eigenschaft verletzt wird. Tritt kein Fehler auf, so sollten sich der exakte Wert an einem Gitterpunkt einer Kombinationslösung $u_i(x_{i,j})$ nur um einen geringen Faktor von dem aus umliegenden Punkten interpolierten Wert $\tilde{u}_i(x_{i,j})$ unterscheiden. Kommt es während der Simulation jedoch zu einer SDC, so kann ein beeinträchtigter Gitterpunkt ein lokalen Extremwert darstellen, welcher stark vom

Algorithmus 3.1 Fehlererkennung anhand der maximalen Änderung

```

1: procedure MAXCHANGEOUTLIERCHECK( $u_i^{(t_s)}, H_i^{\vec{z}}$ )
2:   global  $E_{\Delta_{i,p}^{\vec{z}}} = \{\Delta_{i,p}^{\vec{z}(t_{s-q})}\}$  // Entwicklung der Änderung über die Zeit
3:
4:    $\Delta_{i,p}^{\vec{z}(t_s)} \leftarrow \text{CALCULATEMAXCHANGE}(u_i^{(t_s)}, H_i^{\vec{z}})$  // Gleichung (3.5)
5:    $\tilde{\Delta}_{i,p}^{\vec{z}(t_s)} \leftarrow \text{FORECAST}(E_{\Delta_{i,p}^{\vec{z}}})$  // Prognose aus früheren Werten
6:    $\text{diff} = |\tilde{\Delta}_{i,p}^{\vec{z}(t_s)} - \Delta_{i,p}^{\vec{z}(t_s)}|$ 
7:   if  $\text{diff} > \text{treshhold}$  then
8:      $E_{\Delta_{i,p}^{\vec{z}}} \leftarrow E_{\Delta_{i,p}^{\vec{z}}} \cup \{\Delta_{i,p}^{\vec{z}(t_s)}\}$  // Anhaltspunkt für spätere Prognosen
9:     return True
10:  else
11:     $E_{\Delta_{i,p}^{\vec{z}}} \leftarrow E_{\Delta_{i,p}^{\vec{z}}} \cup \{\tilde{\Delta}_{i,p}^{\vec{z}(t_s)}\}$  // Korrekter Wert zur Entwicklung Hinzufügen
12:    return False
13:  end if
14: end procedure

```

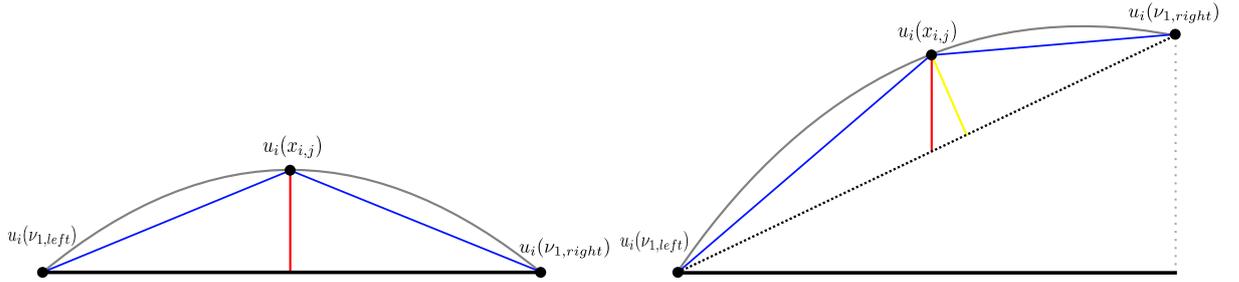


Abbildung 3.3: Veranschaulichung der Abweichung vom Interpolanten

Interpolanten abweicht und somit erkannt werden kann. Eine naheliegende Methode, den Unterschied zum interpolierten Wert an einem Gitterpunkt zu berechnen, wäre lokal das in Gleichung (2.16) beschriebene Hierarchisierungsverfahren aufzurufen. Hierbei muss jedoch im d -dimensionalen Fall alle $3 \cdot d - 1$ Nachbarn betrachtet werden, was bei N Gitterpunkten zu einer Laufzeit von $\mathcal{O}(N \cdot 3^d)$ führt. Eine weitere Möglichkeit den Interpolanten $\tilde{u}_i^{\vec{z}}(x_{i,\vec{z}})$ zu berechnen ist es ausschließlich die $2 \cdot d$ nächsten Gitterpunkte zu betrachten und diese mit inversem Abstand zum Gitterpunkt zu gewichten. Dies führt im d -dimensionalen Fall zu einer Laufzeit von $\mathcal{O}(2d \cdot N)$. Seien $\nu_{r,w} = x_{i,j \pm \vec{e}_r}^{\vec{z}}$ mit $1 \leq r \leq d$ und $w \in \{left, right\}$ die $2 \cdot d$ direkten Nachbarpunkte in Raumrichtung r eines Gitterpunktes $x_{i,\vec{z}}^{\vec{z}}$, dann lässt sich die Interpolation mittels inverser Distanz-Gewichtung (IDW) nach [Geo] wie folgt definieren.

$$(3.6) \quad \tilde{u}_i^{\vec{z}}(x_{i,\vec{z}}^{\vec{z}}) := \frac{\sum_{r,w} h_{l_r}^{-1} \cdot u_i^{\vec{z}}(\nu_{r,w})}{\sum_{r,w} h_{l_r}^{-1}}$$

Ein Vorteil, welcher durch die geringere Anzahl an betrachteten Nachbarn hervorgerufen wird, zeigt sich bei Kombinationslösungen $u_{\vec{i}}$ mit höchsten Auflösungen in eine Raumrichtung, also $\exists k$ mit $i_k = n$. Da sich die eigentliche Information einer solchen Kombinationslösung auf einem Streifen in der Mitte des Gitters Ω_i befindet, sind die zugehörigen Gitterpunkte größtenteils von Randpunkten umgeben. Gerade bei Dirichlet Randpunkten, welche sich in der Zeit nicht verändern, kann es bei einer Interpolation aus mehreren Nachbarwerten dazu kommen, dass die eigentliche Information durch die Randpunkte unterdrückt wird und somit der Unterschied zwischen Interpolant und tatsächlichem Wert verfälscht wird. Der Unterschied $\epsilon_{\vec{i}}^{(t_i)}(x)$ zu einem Zeitpunkt t_i an einem Gitterpunkt $x_{\vec{i},\vec{j}}$ kann wie folgt definiert werden.

$$(3.7) \quad \epsilon_{\vec{i}}^{(t_i)}(x_{\vec{i},\vec{j}}) := |u_{\vec{i}}(x_{\vec{i},\vec{j}}) - \tilde{u}_{\vec{i}}(x_{\vec{i},\vec{j}})|$$

Die Abweichung vom interpolierten Wert ist ein erstes Maß um lokale Extremwerte ausmachen zu können. Jedoch zeigt sich bei genauerer Betrachtung, dass abhängig von der zugrundeliegenden Steigung zwischen zwei gegenüberliegenden Nachbarn ein Unterschied zum tatsächlichen Wert mehr oder weniger aussagekräftig sein kann. In Abbildung ?? sind zwei Szenarien abgebildet. In beiden Fällen ist ein Gitterpunkt mit seinen beiden Nachbarn im eindimensionalen Fall dargestellt. Die blaue Linie ist die stückweise lineare Approximation der korrekten Lösung, welche in Grau abgebildet ist. Die gestrichelte, schwarze Linie bildet den Interpolanten zwischen den beiden Nachbarknoten, wobei der Unterschied zwischen interpoliertem und tatsächlichem Wert am Gitterpunkt in Rot gekennzeichnet ist. Im rechten Teil der Abbildung besitzt der Interpolant eine Steigung $m \neq 0$ wohingegen im linken Fall $m = 0$ gilt. Die Abweichung vom interpolierten Wert ist in beiden Fällen die selbe, jedoch ist der minimale Abstand zum Interpolanten (hier in Gelb dargestellt) in den beiden Szenarien unterschiedlich. Der minimale Abstand beschreibt, wie groß der Unterschied zwischen dem Interpolanten und dem eigentlichen Wert ist und bildet somit ein gutes Maß für ein lokalen Extremwert an einem Gitterpunkt. Unglücklicherweise liegt der Berechnung dieses minimalen Abstandes ein Optimierungsproblem zugrunde, wofür beispielsweise das Gradientenverfahren eingesetzt werden kann. Da wir jedoch nicht an einem genauen Wert interessiert sind, sondern uns mit einem "billigen" Vergleichswert zufrieden geben können, bietet es sich an, das Mittel über die Unterschiede der absoluten Steigungen zwischen Nachbarknoten und tatsächlichen Gitterwert und der absoluten Steigung zwischen Nachbarknoten und interpolierten Gitterwert, als weiteres Maß zu wählen (Man beachte, dass im rechten Fall der Abstand zum Interpolanten kleiner ist als im linken Fall, somit sind auch die Winkel zwischen Interpolanten und stückweise Linearer Funktion geringer). Das Mittel der Unterschiede der absoluten Steigungen $M_{\vec{i}}^{(t_i)}(x_{\vec{i},\vec{j}})$ lässt sich wie folgt berechnen:

$$(3.8) \quad M_{\vec{i}}^{(t_i)}(x_{\vec{i},\vec{j}}) := \frac{\sum_{r,w} h_{l_r}^{-1} \cdot (||u_{\vec{i}}(\nu_{r,w}) - u_{\vec{i}}(x_{\vec{i},\vec{j}})| - |u_{\vec{i}}(\nu_{r,w}) - \tilde{u}_{\vec{i}}(x_{\vec{i},\vec{j}})|)}{2d}$$

Folglich sei die Abweichung $\hat{\epsilon}_{\vec{i}}(x_{\vec{i},\vec{j}})$ vom Interpolanten an einem Gitterpunkt $x_{\vec{i},\vec{j}}$ das Produkt der beiden zuvor beschriebenen Maße.

$$(3.9) \quad \hat{\epsilon}_{\vec{i}}(x_{\vec{i},\vec{j}}) := M_{\vec{i}}^{(t_i)}(x_{\vec{i},\vec{j}}) \cdot \epsilon_{\vec{i}}(x_{\vec{i},\vec{j}})$$

Algorithmus 3.2 Fehlererkennung anhand der maximalen Abweichung vom Interpolanten

```

1: procedure MAXDIFFINTERPOLANTOUTLIERCHECK( $u_{\vec{i}}^{(t_s)}$ )
2:   global  $E_{\hat{\vec{e}}_{\vec{i}}} = \{\hat{\vec{e}}_{\vec{i}}^{(t_{s-q})}\}$  // Entwicklung der Abweichung über die Zeit
3:
4:    $\hat{\vec{e}}_{\vec{i}}^{(t_s)} \leftarrow \text{CALCULATEMAXDIFF}(u_{\vec{i}}^{(t_s)})$  // Gleichung (3.10)
5:    $\tilde{\vec{e}}_{\vec{i}}^{(t_s)} \leftarrow \text{FORECAST}(E_{\hat{\vec{e}}_{\vec{i}}})$  // Prognose aus früheren Werten
6:    $\text{diff} = |\tilde{\vec{e}}_{\vec{i}}^{(t_s)} - \hat{\vec{e}}_{\vec{i}}^{(t_s)}|$ 
7:   if  $\text{diff} > \text{treshhold}$  then
8:      $E_{\hat{\vec{e}}_{\vec{i}}} \leftarrow E_{\hat{\vec{e}}_{\vec{i}}} \cup \tilde{\vec{e}}_{\vec{i}}^{(t_s)}$  // Anhaltspunkt für spätere Prognosen
9:     return True
10:  else
11:     $E_{\hat{\vec{e}}_{\vec{i}}} \leftarrow E_{\hat{\vec{e}}_{\vec{i}}} \cup \{\hat{\vec{e}}_{\vec{i}}^{(t_s)}\}$  // Korrekter Wert zur Entwicklung Hinzufügen
12:    return False
13:  end if
14: end procedure

```

Ähnlich wie in den bisher vorgestellten Verfahren, soll auch in diesem Fall ein einzelner Wert eine Kombinationslösung repräsentieren. Demnach sei

$$(3.10) \quad \hat{\vec{e}}_{\vec{i}} := \max_{\vec{i} \leq \vec{i}} \max_{0 \leq j \leq 2^i} \hat{\vec{e}}_{\vec{i}}(x_{\vec{i},j})$$

ein Maß für die maximale Abweichung zwischen dem Interpolierten und tatsächlichen Wert an den Stützstellen einer Kombinationslösung $u_{\vec{i}}$.

Speichert man nach jedem Zeitschritt $\hat{\vec{e}}_{\vec{i}}$ für eine Kombinationslösung $u_{\vec{i}}$ so beobachtet man bei einer ausreichen großen SDC, dass $\hat{\vec{e}}_{\vec{i}}$ zum Zeitpunkt der SDC verhältnismäßig groß ist. Im vorherigen Verfahren wurde beschrieben, dass sich die Änderung an einem Gitterpunkt nach dem Verteilungsschritt während der SGCT unerwartet verhalten kann. Betrachtet man den Verlauf der $\hat{\vec{e}}_{\vec{i}}$ über die Zeit, so stellt man fest, dass sich auch hier der Wert nach dem Verteilungsschritt kurzzeitig unerwartet ändert. Aus diesem Grund sollten auch in diesem Fall Werte nach der Kombination nicht mit Werten davor verglichen werden. Das Verfahren ist in 3.2 ähnlich wie in 3.2.1 in Form eines Pseudocodes dargestellt. Dabei wird für eine Kombinationslösung zunächst die maximale Abweichung $\hat{\vec{e}}_{\vec{i}}^{(t_s)}$ berechnet, welche dann in Zeile 4 mit einem aus vorherigen Werten geschätzten Wert $\tilde{\vec{e}}_{\vec{i}}^{(t_s)}$ verglichen wird. Ist der Unterschied zu groß, wird die Kombinationslösung als fehlerhaft eingestuft.

3.2.3 Fehlertolerante SGCT

Um die Chance zu maximieren, einen Silent Fault während der Lösung einer PDE mit der SGCT aufzudecken, bietet es sich an, alle bisher vorgestellten Fehlererkennungsverfahren in der

Berechnung einzusetzen. Da die Kombinationsgitter wie in 3 beschrieben, während der Zeit des Lösens, auf den einzelnen Knoten eines HPC verteilt sind, entsteht durch den Austausch von Werten zwischen einzelnen Lösungen ein enormer Kommunikationsaufwand. Aus diesem Grund ist es sinnvoll, die beiden von Hinojosa et. al. vorgestellten Verfahren, welche auf dem Vergleich von Kombinationslösungen basieren, zum Zeitpunkt des Kombinationssschrittes zu vollziehen, da hierbei ohnehin Gitterwerte ausgetauscht werden müssen. Leider liegt hierzu noch kein effizienter Algorithmus vor, der dies direkt während der Kombination, beispielsweise in der von Hupp et. al. vorgestellten Reduce-Operation [HHJP15], vollzieht. Hinojosa et. al. [HHH] beschreibt, dass sich hauptsächlich der Vergleich auf hierarchischer Ebene 3.1.2 für eine solche parallele Berechnung eignen könnte. Die innerhalb dieser Arbeit vorgestellten Verfahren zur Fehlererkennung anhand der zeitlichen Entwicklung einer Kombinationslösung lassen sich hingegen direkt auf Knoten durchführen, da keine Kommunikation mit anderen Kombinationslösungen von Nöten ist. Dabei muss im Falle des zuletzt vorgestellten Verfahrens 3.2.2 lediglich ein Wert für jeden Zeitschritt pro Kombinationslösung gespeichert und nach jedem berechneten Zeitschritt $\mathcal{O}(2d \cdot N)$ Gitterwerte betrachtet werden. In 3.3 ist ein fehlertolerantes PDE Lösungsverfahren mit der SGCT beschrieben, welches alle bisher vorgestellten Erkennungsverfahren sowie die FTCT vereinigt.

Algorithmus 3.3 Lösung einer PDE mit dem SGCT Verfahren (angelehnt an [HHH])

```

1: procedure SIMULATEPDE( $u(\vec{x}, t = 0)$ ,  $T$ ,  $N_t$ ,  $\vec{n}$ ,  $\tau$ ,  $p$ )
2:    $\mathcal{J}_{\vec{n}, q, \tau}^{ext} \leftarrow \text{GENERATEEXTENDEDINDEXSET}(\vec{n}, \tau)$ 
3:    $\{c_i^{\vec{r}}\} \leftarrow \text{GENERATEFACTORS}(\vec{n}, \tau)$ 
4:    $H = \{H_i^{\vec{r}}\}$ 
5:
6:   for  $\vec{i} \in \mathcal{J}_{\vec{n}, q, \tau}^{ext}$  do
7:      $u_i^{(t_0)} \leftarrow u(\vec{x}, t = 0)$  // Diskretisierung auf Kombinationsgitter
8:      $H_i^{\vec{r}} = \emptyset \cup \{u_i^{(t_0)}\}$  // Initialisierung der Historie einer Kombilösung
9:   end for
10:
11:  for  $s = 1$  to  $T$  step  $N_t$  do // Simulation für  $T$  Zeitschritte
12:     $I_{SDC} = \emptyset$  // Menge der Indizes von fehlerhaften Kombilösungen
13:    for  $i \in \mathcal{J}_{\vec{n}, q, \tau}^{ext}$  do in parallel
14:      for  $N_t$  do
15:         $u_i^{(t_s)} \leftarrow \text{SOLVE}(u_i^{(t_{s-1})}, 1)$  // Lösen der PDE für einen Zeitschritt
16:         $dC \leftarrow \text{MAXCHANGEOUTLIERCHECK}(u_i^{(t_s)}, H_i^{\vec{r}})$  // Alg. 3.1
17:         $dD \leftarrow \text{MAXDIFFINTERPOLANTOUTLIERCHECK}(u_i^{(t_s)})$  // Alg. 3.2
18:        if  $dC$  or  $dD$  then
19:           $I_{SDC} = I_{SDC} \cup \{i\}$  // Index im Fehlerfall Speichern
20:        end if
21:         $\text{UPDATEHISTORY}(H_i^{\vec{r}}, u_i^{(t_s)}, p)$  // Historie aktualisieren
22:      end for
23:       $u_i^{(t_s)} \leftarrow \text{HIERARCHIZE}(u_i^{(t_s)})$  // Für die Kombination [HHJP15]
24:    end for
25:
26:     $I_{SDC} = I_{SDC} \cup \text{HINOJOSAOUTLIERCHECK}(\{u_i^{(t_s)}\})$ 
27:    if  $I_{SDC} \neq \emptyset$  then // Im Fehlerfall  $\rightarrow$  FTCT
28:       $\{c_i^{\vec{r}}\} \leftarrow \text{UPDATEFACTORS}(I_{SDC})$ 
29:    end if
30:     $u_n^{(c)} \leftarrow \text{COMBINE}(\{u_i^{(t_s)}\}, \{c_i^{\vec{r}}\})$  // Kombination
31:     $u_i^{(c)} \leftarrow \text{DEHIERARCHIZE}(u_n^{(c)})$ 
32:
33:    for  $i \in \mathcal{J}_{\vec{n}, q, \tau}^{ext}$  do
34:       $u_i^{(t_s)} \leftarrow \text{SAMPLE}(u_n^{(c)}, i)$  // Verteilungsschritt
35:    end for
36:  end for
37: end procedure

```

4 Verhalten der Fehlererkennungverfahren und Ergebnisse

Im Folgenden wird das Verhalten der innerhalb dieser Arbeit entstandenen Fehlererkennungsverfahren anhand der Simulation von vier 2D Diffusions- und Advektionsgleichungen, beschrieben. Die Simulation der PDEs mit Hilfe der SGCT sowie die Berechnung der Gitter und PDEs wurden in Python implementiert, wobei die Kombinationsgitter seriell berechnet und nach jedem Zeitschritt gespeichert wurden. Weiterhin wurden nach jedem Zeitschritt die beiden Erkennungsverfahren aus 3.2.1 (mit $p = 2$) und 3.2.2 aufgerufen und die entsprechenden Werte (maximale Änderung und Abweichung vom Interpolanten) abgespeichert. Um einen Vergleich zu bereits existierenden Verfahren ziehen zu können, wurden außerdem die von Hinojosa in 3.1.1 und 3.1.2 beschriebenen Erkennungsverfahren implementiert und nach jedem berechneten Zeitschritt aufgerufen. Bevor die Ergebnisse präsentiert werden, werden im Folgenden zunächst die Parameter der Simulationen beschrieben.

Die einzelnen Simulationen wurden für $T = 100$ Zeitschritte gelöst, wobei die Kombinationslösungen nach jedem 10ten Zeitschritt kombiniert wurden ($N_t = 10$). Zur Lösung der PDEs wurde das Upwind Verfahren eingesetzt, welches richtungsabhängige Ableitungen verwendet und sich so besonders für Advektionsprozesse eignet. Die Advektions-Diffusionsgleichung lässt sich nach [Meh14] wie folgt definieren:

$$(4.1) \quad u_t = Re^{-1} \cdot Pr^{-1} \cdot (u_{xx} + u_{yy}) - v_1 \cdot u_x - v_2 \cdot u_y$$

Die Werte Re (Reynoldszahl) sowie Pr (Prandtlzahl) bestimmen dabei den Diffusionsanteil. Die Geschwindigkeit der Advektion in x - und y - Richtung wird, wie bereits in Kapitel 3 beschrieben, durch die Konstanten v_1 und v_2 ausgedrückt. In den im Folgenden beschriebenen Simulationen wurde der Anteil der Diffusion durch $Re = 1000$ und $Pr = 0.3$ festgelegt sowie die Geschwindigkeit der Advektion auf $v_1 = 0.1$ und $v_2 = 0$ gesetzt. Wird nur ein einzelner der beiden Prozesse Diffusion und Advektion separat betrachtet, so sind die Werte des nicht zugehörigen Prozesses, auf 0 gesetzt. Die kombinierte Lösung besitzt in allen Fällen eine Auflösung von $\vec{l} = (5, 5)$. Demnach wurden für jeden Zeitschritt $|\mathcal{S}_{5,q}| = 9$ Kombinationslösungen nacheinander berechnet. Da ein explizites Verfahren zur Lösung der PDEs gewählt wurde, kann der Zeitschritt nicht beliebig groß gewählt werden. Die maximale Länge eines Zeitschrittes

kann nach [Meh14] wie folgt aus dem Quadrat der Maschenweiten sowie dem Diffusionsanteil berechnet werden.

$$(4.2) \quad dt < \frac{Re \cdot Pr \cdot h_5^2}{4}.$$

Die Länge eines Zeitschritts wurde in den folgenden Szenarien auf 80% der oberen Schranke gesetzt. In den folgenden Versuchen wurden vier unterschiedliche Anfangsbedingungen betrachtet. Dabei wurden die PDEs für Dirichlet Randbedingungen berechnet mit $\partial\Omega = 0$. Als initiale Lösung wurde zum Einen die bereits in 3 vorgestellte quadratische Funktion und zum Anderen die in [HHH] verwendete Sinusfunktion eingesetzt. Die Sinusfunktion $f(x, y) = \sin(2\pi x) \cdot \sin(2\pi y)$ besitzt im Gebiet Ω vier Extremwerte und weist an den Rändern $\partial\Omega$, ähnlich wie die quadratische Funktion, den Wert 0 auf. Um eine SDC künstlich zu erzeugen, wurde in den Simulationen der Wert an einem Gitterpunkt $x_{\vec{i},\vec{j}}$ einer Kombinationslösung $u_{\vec{i}}$, ähnlich wie in [EHM14a; HHH] beschrieben, wie folgt beeinträchtigt:

- $\tilde{u}_i(x_{\vec{i},\vec{j}}) = u_i(x_{\vec{i},\vec{j}}) \cdot 10^e$

Dabei stellt $\tilde{u}_i(x_{\vec{i},\vec{j}})$ den korrupten Wert am Gitterpunkt dar und e bestimmt die Größenordnung des eingeführten Fehlers. Wie bereits in Hinojosa et. al. beschrieben, betragen auch hier die Größenordnungen des Fehlers $e \in \{-300, -0.5, +5\}$. Der Fehler wurde nach der Berechnung eines Zeitschrittes eingeführt. Dadurch wird den Verfahren die Möglichkeit geboten, den Wert anhand eines einzelnen korrupten Wertes zu erkennen, ohne einen solchen Fall komplett ausschließen zu müssen. Wird der Fehler andernfalls während des Lösungsschrittes eingeführt, so breitet sich dieser durch die Advektion und/oder Diffusion direkt auf Nachbarwerte aus.

Die folgenden vier Szenarien beschreiben die Parameter der einzelnen Simulationen. In allen Fällen sind dabei die kombinierte Lösung zum finalen Zeitpunkt sowie die Kombinationsgitter vor der ersten Kombination, nach Auftreten des Fehlers(zum Zeitschritt t_{60}), abgebildet. Weiterhin sind Plots zu den beiden in dieser Arbeit vorgestellten Verfahren dargestellt. Ein Teil der Ergebnisse, welche mithilfe der von Hinojosa et. al. in Kapitel 3.1.1 und 3.1.2 vorgestellten Verfahren berechnet wurden, sind anhand von zwei Tabellen aufgeführt.

- Diffusion am Beispiel der Funktion 2. Grades mit Dirichlet Randbedingungen $\partial\Omega = 0$. Der Fehler mit der Größenordnung von 10^5 wurde in Kombinationslösung $u_{\vec{i}}$ mit Auflösung $\vec{i} = (5, 1)$ am Punkt $x_{(5,1),(9,1)} = (0.28125, 0.5)$ eingeführt.
- Advektion am Beispiel einer 2D Sinusfunktion mit Dirichlet Randbedingungen $\partial\Omega = 0$. Der Fehler mit Größenordnung $10^{0.5}$ wurde in Kombinationslösung $u_{\vec{i}}$ mit Auflösung $\vec{i} = (3, 3)$ am Punkt $x_{(3,3),(3,7)} = (0.375, 0.875)$ eingeführt.
- Advektion am Beispiel der Funktion 2. Grades mit mit Dirichlet Randbedingungen $\partial\Omega = 0$. Der Fehler mit Größenordnung $10^{0.5}$ wurde in Kombinationslösung $u_{\vec{i}}$ mit Auflösung $\vec{i} = (5, 1)$ am Punkt $x_{(5,1),(16,1)} = (0.5, 0.5)$ eingeführt.

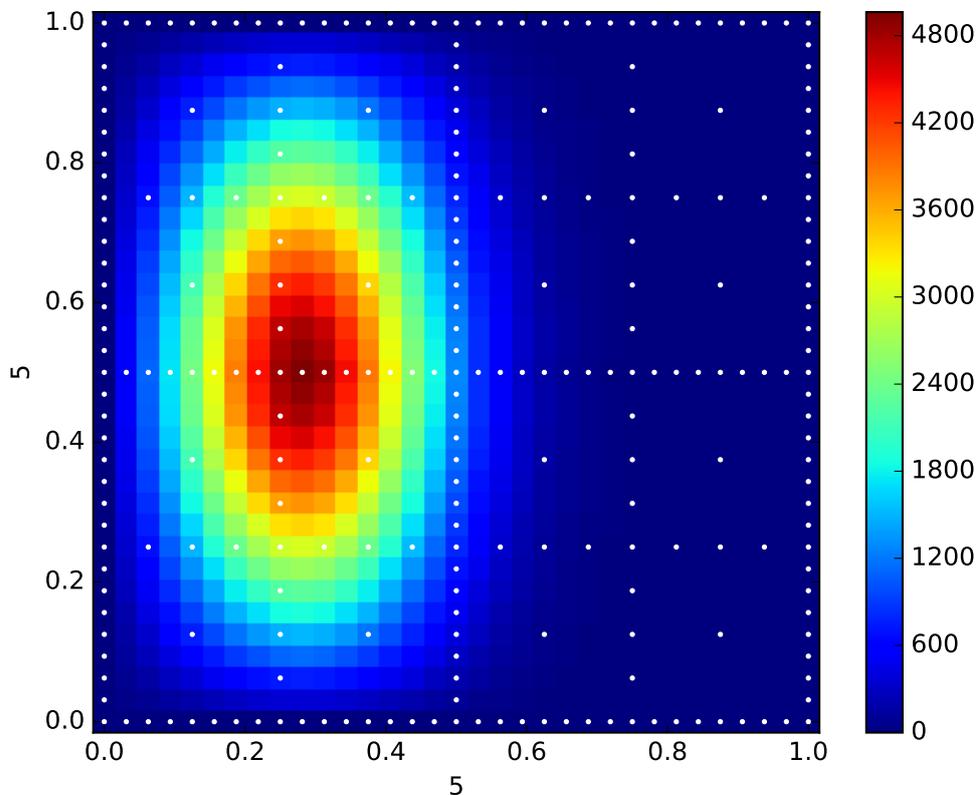


Abbildung 4.1: Szenario 1: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dünnen Gitter

- Advektion-Diffusion am Beispiel einer 2D Sinusfunktion mit Dirichlet Randbedingungen $\partial\Omega = 0$. Der Fehler mit Größenordnung 10^{-300} wurde in Kombinationslösung $u_{\vec{i}}$ mit Auflösung $\vec{i} = (3, 3)$ am Punkt $x_{(3,3),(3,7)} = (0.375, 0.875)$ eingeführt.

4.1 Szenario 1: Diffusion, Dirichlet, quadratische Funktion

Das erste Szenario beschreibt den günstigsten Fall um einen Fehler erkennen zu können. Dabei besitzt der Fehler zum Einen eine Größenordnung, welche der Größenordnung der korrekten Werten an den Gitterpunkten um ein Vielfaches entspricht, und zum Anderen breitet sich der Fehler durch die Diffusion unmittelbar auf alle umliegenden Gitterpunkte aus, was besonders in Abbildung 4.2 anhand Kombinationslösung $u_{(5,1)}$ (unten rechts), oder in der finalen Lösung 4.1 zu beobachten ist. Dementsprechend reagieren die Fehlererkennungsverfahren wie erwartet

4 Verhalten der Fehlererkennungsverfahren und Ergebnisse

Gitterpaar	Maximaler Unterschied
[1 5], [2 4]	0.000749698801013
[1 5], [3 3]	0.00328856105235
[1 5], [4 2]	0.002004974735
[1 5], [5 1]	5.31205967474e-16
[1 5], [1 4]	0.00124504992858
[1 5], [2 3]	0.00316844613388
[1 5], [3 2]	0.00196336401258
[2 4], [3 3]	0.0030604437861
[2 4], [4 2]	0.00248577650005
[2 4], [5 1]	69282.2855014
[2 4], [1 4]	0.000620860386294
[2 4], [2 3]	0.00322139401551
[2 4], [3 2]	0.00244414581115
[3 3], [4 2]	0.0030604437861
[3 3], [5 1]	200096.434067
[3 3], [1 4]	0.00249813748055
[3 3], [2 3]	0.00195604300867
[3 3], [3 2]	0.00195604300867
[4 2], [5 1]	586923.849458
[4 2], [1 4]	0.00187547289261
[4 2], [2 3]	0.00244414581115
[4 2], [3 2]	0.00322139401553
[5 1], [1 4]	1.71856743155e-06
[5 1], [2 3]	69285.1627798
[5 1], [3 2]	200120.392602
[1 4], [2 3]	0.00237811719255
[1 4], [3 2]	0.00183386754807
[2 3], [3 2]	0.00192208082973

Tabelle 4.1: Szenario 1: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{54}

4.1 Szenario 1: Diffusion, Dirichlet, quadratische Funktion

Zeitschr.	Erkanntes Gitter, Punkt	Verglichene Gitter	Verglichene Werte
55	[5 1], [0.25 0.5]	[2 4], [3 3], [4 2], [5 1], [2 3], [3 2], [4 1]	[1.92000000e-01 1.91000000e-01 1.91000000e-01 1.32896120e+04 1.92000000e-01 1.91000000e-01 1.91000000e-01]
	[5 1], [0.125 0.5]	[3 3], [4 2], [5 1], [3 2], [4 1]	[3.30000000e-02 3.30000000e-02 -6.64467700e+03 3.30000000e-02 3.30000000e-02]
	[5 1], [0.375 0.5]	[3 3], [4 2], [5 1], [3 2], [4 1]	[5.60000000e-02 5.60000000e-02 -6.64465400e+03 5.60000000e-02 5.60000000e-02]
	[5 1], [0.1875 0.5]	[4 2], [5 1], [4 1]	[1.10000000e-02 -6.64469900e+03 1.10000000e-02]
	[4 1], [0.1875 0.5]	[4 2], [5 1], [4 1]	[1.10000000e-02 -6.64469900e+03 1.10000000e-02]
	[5 1], [0.3125 0.5]	[4 2], [5 1], [4 1]	[1.40000000e-02 6.64472400e+03 1.40000000e-02]
	[4 1], [0.3125 0.5]	[4 2], [5 1], [4 1]	[1.40000000e-02 6.64472400e+03 1.40000000e-02]

Tabelle 4.2: Szenario 1: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel ??) zum Zeitschritt t_{54}

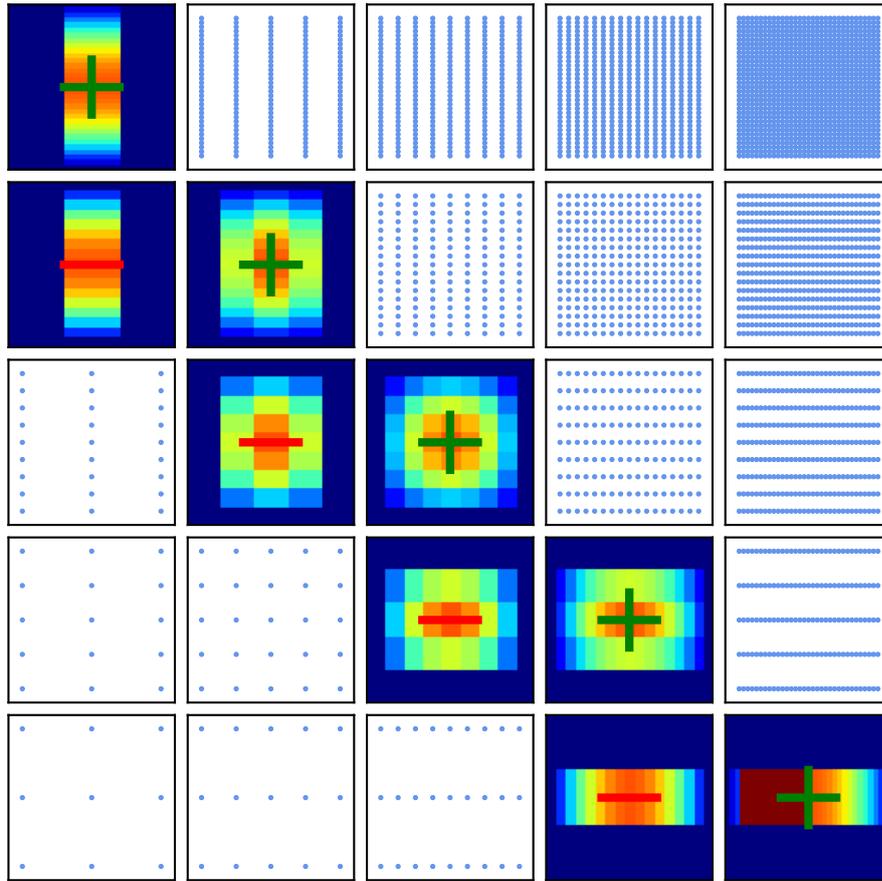


Abbildung 4.2: Szenario 1: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}

auf den Fehler. Anhand des Verlaufs der maximalen Änderung in Abbildung 4.4 lässt sich beispielsweise erkennen, dass sich bis zum Zeitschritt t_{54} die Änderung am Kombinationsgitter zunächst fast konstant verhält. Lediglich zu den Zeitpunkten der Kombinationen ist der kurzzeitige Anstieg erkennbar, welcher bereits in 3.2.1 beschrieben wurde. Zum Zeitpunkt t_{54} ist dann ein enormer Anstieg am lila Plot der Kombinationslösung $u_{(5,1)}$ erkennbar, welcher durch die logarithmische Achseneinteilung der Ordinate zusätzlich unterdrückt wird. Am Zeitschritt t_{60} erkennt man, dass der Fehler durch den Kombinationsschritt auf die weiteren Lösungen übertragen wird, wobei Lösungen, welche die höchsten Auflösungen in y -Richtung aufweisen, zunächst verschont bleiben bis der Fehler sich auch über das Zentrum des Gebiets Ω hinweg ausgebreitet hat. Die maximale Abweichung vom Interpolanten, abgebildet in Abbildung 4.3 verhält sich nahezu gleich wie die maximale Änderung. Einzig die Größenordnung des Ausschlags zum Zeitpunkt des Fehlers ist doppelt so groß, was auf die Multiplikation der beiden Maße aus 3.9 zurückzuführen ist. Da der Fehler auf dem Gitter mit höchstem Level in x Richtung eingeführt wurde und kein anderes Gitter denselben Punkt beinhaltet, ist es den beiden Verfahren von Hinojosa et. al. erst zum nachfolgenden Zeitschritt t_{55} möglich, den Fehler

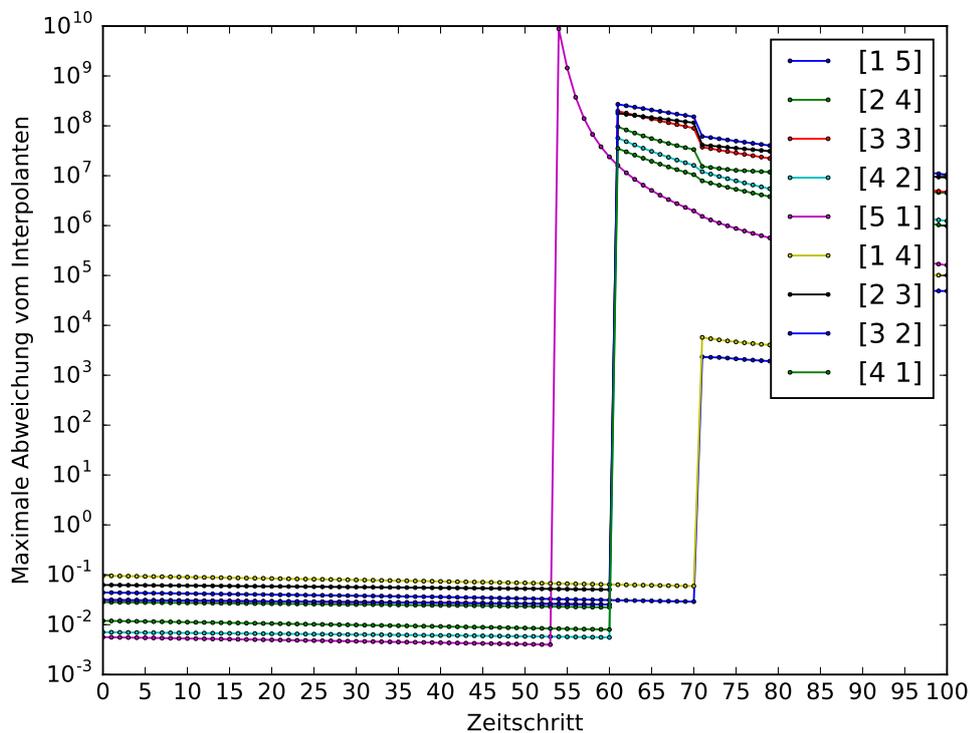


Abbildung 4.3: Szenario 1: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit

zu enttarnen. Dabei erkennt man in der ersten Tabelle in 4.1, welche in der linken Spalte die Auflösung des verglichenen Gitterpaars und in der rechten Spalte den maximalen Unterschied eines Gitterwertes in beiden Kombinationslösungen darstellt 3.2, dass in fast allen Einträgen, in welchen die Kombinationslösung $u_{(5,1)}$ mit einer anderen Kombinationslösung verglichen wurde, der maximale Unterschied enorm groß ist. Die weitere Tabelle beschreibt das zweite Verfahren von Hinojosa et. al. 3.1.2, wobei jede der Zeilen einen erkannten Ausreißer zum Zeitschritt t_5^5 darstellt. Die erste Spalte gibt hierbei das erkannte Gitter sowie den korrupten Gitterpunkt an. In der nächsten Spalte sind die verglichenen Kombinationslösungen aufgelistet, wobei die zugehörigen Werte am korrupten Gitterpunkt in der letzten Spalte aufgelistet sind. Vergleicht man die korrupten Gitterpunkte in den jeweiligen Zeilen, so stellt man fest, dass sich der Fehler auf die benachbarten Punkte in x -Richtung ausgebreitet hat (man bemerke, dass sich der Fehler nicht auf den Rand ausbreiten kann, da dieser nach jedem berechneten Schritt erneut auf 0 gesetzt wird) und vom Verfahren erkannt werden kann.

Ein Problem, welches sowohl hier als auch noch in den folgenden Szenarien auftreten wird, stellen die von der Methode `outlier_test`, aus der Bibliothek `statsmodels`, erkannten Ausreißer dar. Soll ein Ausreißer in einer Menge von drei Werten erkannt werden, so sind die zurückgelieferten Ergebnisse teilweise unbrauchbar. Wird die Methode beispielsweise mit den Werten

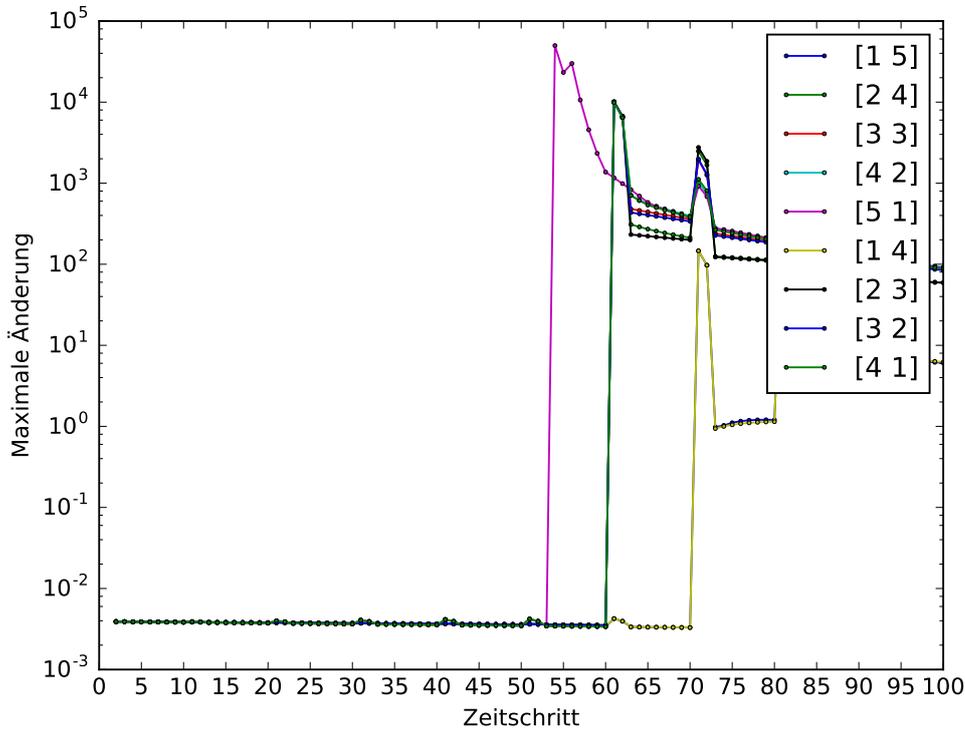


Abbildung 4.4: Szenario 1: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen

(0.16, 1, 0.21) aufgerufen, so wird standardgemäß kein Wert als Ausreißer erkannt und die beiden Letzten Werte als gleich wahrscheinlich gewertet. Weiterhin werden negative Werte völlig ignoriert. Werden demnach die Werte (2, 2, -2, 2) übergeben, wird der Wert -2 nicht als Ausreißer erkannt sondern es sind alle mit völliger Sicherheit keine Ausreißer. Es fehlte innerhalb dieser Arbeit an der Zeit, sich mit dem Verfahren genauer zu beschäftigen, da die Methode jedoch im Modul `linear_model` aufzufinden ist, liegt es nahe, dass diese in irgendeiner Weise versucht, mittels linearer Regression Ausreißer aufzuspüren, was zumindest den zuerst beschriebenen Fall im Ansatz erklären könnte.

4.2 Szenario 2: Advektion, Dirichlet, Sinus Funktion

Die zweite Simulation beschäftigt sich mit einer reinen Advektion, wobei der Fehler anders als zuvor deutlich kleiner ist und demnach nicht so einfach erkennbar sein sollte. Der Fehler wurde in diesem Fall ähnlich wie im vorherigen Szenario an einem Gitterwert ohne mögliche Vergleichswerte eingeführt. Durch die Advektion wirkt sich der Fehler in dieser Simulation ausschließlich in Positive x -Richtung aus, was auch anhand der Kombinationslösung $u_{(3,3)}$

Gitterpaar	Maximaler Unterschied
[1 5], [2 4]	0.0470068960746
[1 5], [3 3]	0.288155110779
[1 5], [4 2]	0.0311125836777
[1 5], [5 1]	0.0181809584681
[1 5], [1 4]	0
[1 5], [2 3]	0.0470068960746
[1 5], [3 2]	0.0478357318853
[2 4], [3 3]	1.93317017642
[2 4], [4 2]	2.82061891264
[2 4], [5 1]	3.59802413044
[2 4], [1 4]	0.0470068960746
[2 4], [2 3]	0
[2 4], [3 2]	1.30170261492
[3 3], [4 2]	0.971560775127
[3 3], [5 1]	1.30055911807
[3 3], [1 4]	0.288155110779
[3 3], [2 3]	1.93317017642
[3 3], [3 2]	0
[4 2], [5 1]	0.363035500314
[4 2], [1 4]	0.0311125836777
[4 2], [2 3]	2.82061891264
[4 2], [3 2]	0.971560775127
[5 1], [1 4]	0.0181809584681
[5 1], [2 3]	3.59802413044
[5 1], [3 2]	1.30055911807
[1 4], [2 3]	0.0470068960746
[1 4], [3 2]	0.0478357318853
[2 3], [3 2]	1.30170261492

Tabelle 4.3: Szenario 2: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{55}

Zeitschr.	Erkanntes Gitter, Punkt	Verglichene Gitter	Verglichene Werte
55	[3 3], [0.375 0.25]	[3 3], [4 2], [3 2]	[0.014 0.007 0.014]
	[3 3], [0.375 0.75]	[3 3], [4 2], [3 2]	[-0.014 -0.007 -0.014]

Tabelle 4.4: Szenario 2: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel 3.1.2) zum Zeitschritt t_{55}

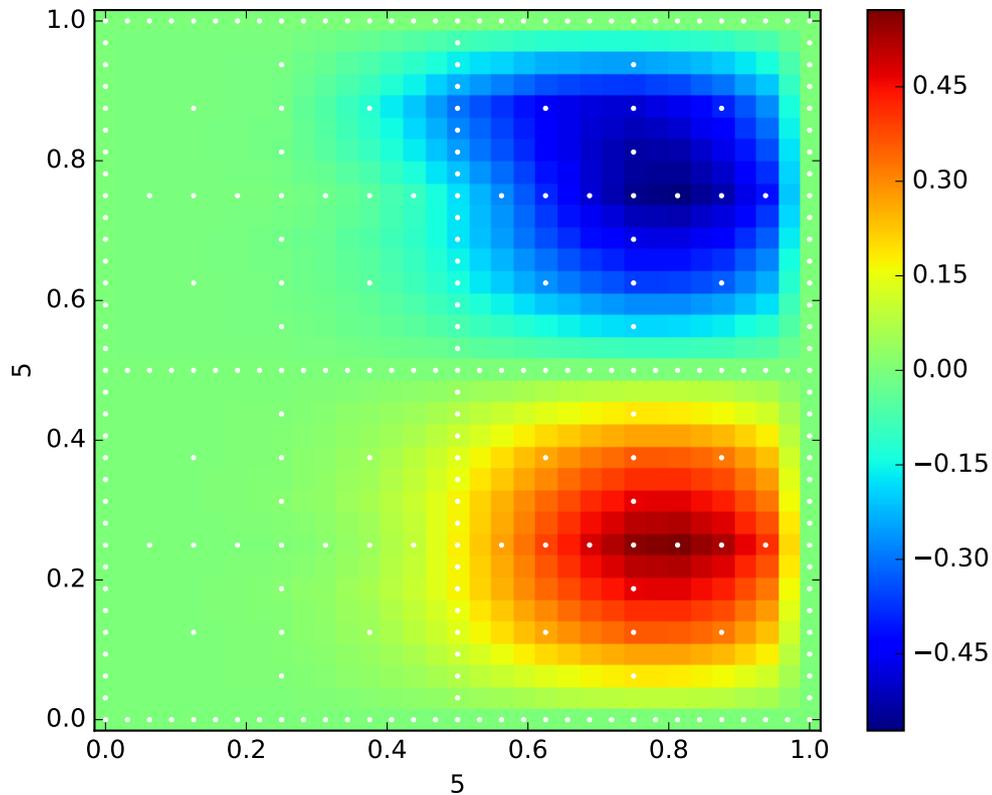


Abbildung 4.5: Szenario 2: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dünnen Gitter

in Abbildung 4.6 und in der finalen Lösung zu erkennen ist. Die Streuung um den Fehler entsteht dabei lediglich durch die Interpolation auf die Vollgitterlösung. Demnach wird der Fehler in jedem Schritt auf weniger Nachbarn übertragen als beim zuvor beschriebenen Diffusionsprozess. Dies erklärt auch die geringere Anzahl an der, mit dem von Hinojosa et. al. vorgestellten Verfahren, erkannten Ausreißer in Tabelle 4.4. Während die Größe des Fehlers um ein Vielfaches kleiner ist als im vorherigen Szenario, fallen entsprechend die maximalen Unterschiede der korrupten Kombinationslösung in Tabelle 4.3 deutlich geringer aus. Trotzdem ist der Unterschied im Vergleich zu einem nicht betroffenen Gitterpaar noch groß genug, um möglicherweise erkannt werden zu können. Betrachtet man die zeitliche Entwicklung der beiden Fehlererkennungsmethoden aus 3.1 und 3.9, so erkennt man, dass die Werte der Kombinationslösungen $u_{(1,4)}$ und $u_{(1,5)}$ (verdeckt durch den Plot der Lösung $u_{(1,4)}$) plötzlich ungewöhnlich stark ansteigen. Da in beiden Lösungen die eigentliche Information auf dem Streifen in y -Richtung, in der Mitte des Gitters $\Omega_{(1,4)}$ beziehungsweise $\Omega_{(1,5)}$ konzentriert ist und die zugehörigen Werte $x_{(1,4),(1,j_2)}$, sowie die Randwerte zu Beginn der Simulation den

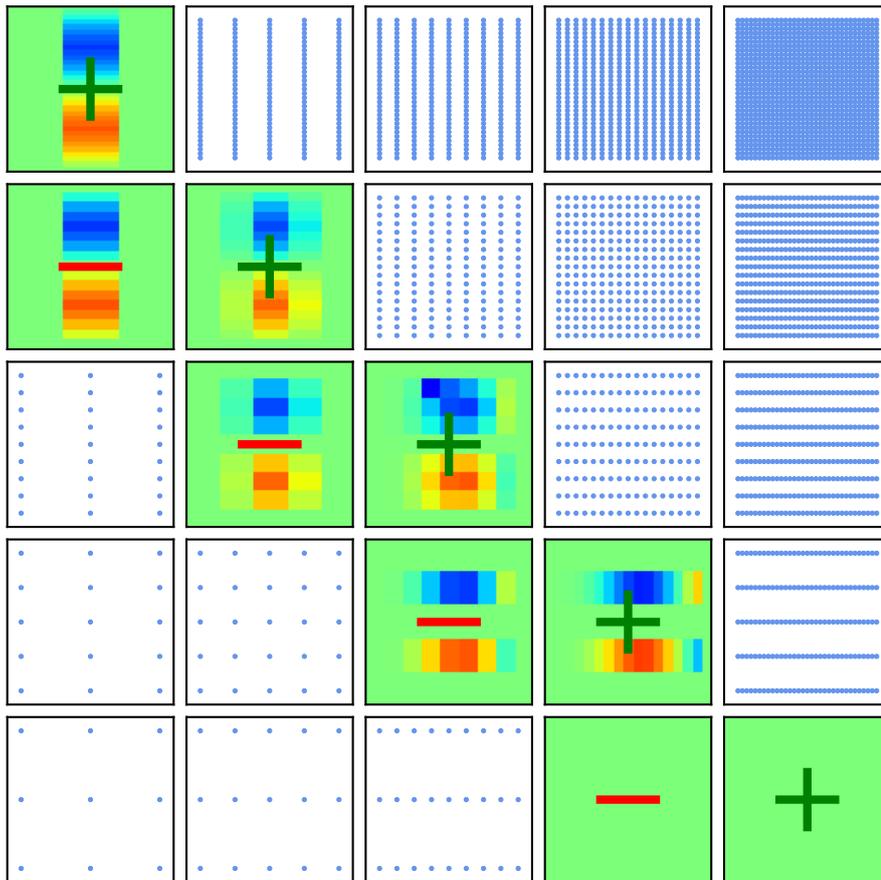


Abbildung 4.6: Szenario 2: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}

Wert 0 erhalten, kann sich in den ersten Zeitschritten keine bis auf durch Rechenfehler bedingte Änderung einstellen. Nach der Kombination ändern sich die Werte demnach schlagartig, da andere Kombinationslösungen auf die Information benachbarter Werte zugreifen und sich somit in der Zeit weiterentwickeln konnten. Die unterschiedliche Entwicklung der Gitter erklärt zudem die vielen *false positive* Funde des zweiten Verfahrens von Hinojosa et. al., welche sich hauptsächlich zu Beginn der Simulation ereigneten. Der Fehler zum Zeitschritt t_{54} in der Kombinationslösung $u_{(3,3)}$ lässt sich in den beiden Plots, welche die zeitliche Entwicklung der maximalen Änderung sowie die maximale Abweichung vom Interpolanten beschreiben, anhand des Ausschlags erkennen, wobei dieser im Vergleich zum zuvor beschriebenen Fall wie erwartet deutlich geringer ausfällt.

4 Verhalten der Fehlererkennungsverfahren und Ergebnisse

Gitterpaar	Maximaler Unterschied
[1 5], [2 4]	0.0394436973646
[1 5], [3 3]	0.0529054410588
[1 5], [4 2]	0.0518546882635
[1 5], [5 1]	2.01368903162
[1 5], [1 4]	0
[1 5], [2 3]	0.0394436973646
[1 5], [3 2]	0.0529054410588
[2 4], [3 3]	0.0496732799062
[2 4], [4 2]	0.0542186901441
[2 4], [5 1]	12.7606193937
[2 4], [1 4]	0.0394436973646
[2 4], [2 3]	0
[2 4], [3 2]	0.0245523747695
[3 3], [4 2]	0.142929437559
[3 3], [5 1]	13.8190820436
[3 3], [1 4]	0.0529054410588
[3 3], [2 3]	0.0496732799062
[3 3], [3 2]	0
[4 2], [5 1]	4706.5140516
[4 2], [1 4]	0.0518546882635
[4 2], [2 3]	0.0542186901441
[4 2], [3 2]	0.142929437559
[5 1], [1 4]	2.01368903162
[5 1], [2 3]	12.7606193937
[5 1], [3 2]	13.8190820436
[1 4], [2 3]	0.0394436973646
[1 4], [3 2]	0.0529054410588
[2 3], [3 2]	0.0245523747695

Tabelle 4.5: Szenario 3: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{54}

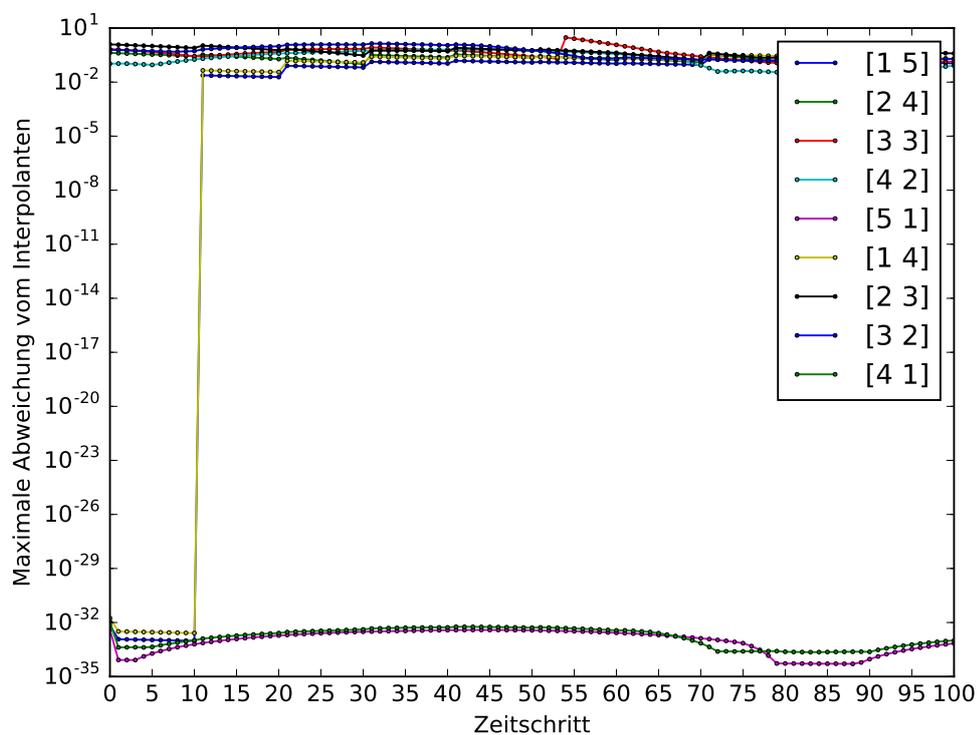


Abbildung 4.7: Szenario 2: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit

Zeitschr.	Erkanntes Gitter, Punkt	Verglichene Gitter	Verglichene Werte
54	[5 1], [0.375 0.5]	[3 3], [4 2], [5 1], [3 2], [4 1]	[-0.045 -0.052 -0.672 -0.045 -0.052]
	[5 1], [0.625 0.5]	[3 3], [4 2], [5 1], [3 2], [4 1]	[0.058 0.06 -0.557 0.058 0.06]
	[5 1], [0.4375 0.5]	[4 2], [5 1], [4 1]	[0. -0.617 0.]
	[4 1], [0.4375 0.5]	[4 2], [5 1], [4 1]	[0. -0.617 0.]
	[5 1], [0.5625 0.5]	[4 2], [5 1], [4 1]	[0.014 -0.604 0.014]
	[4 1], [0.5625 0.5]	[4 2], [5 1], [4 1]	[0.014 -0.604 0.014]

Tabelle 4.6: Szenario 3: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel 3.1.2) zum Zeitschritt t_{54}

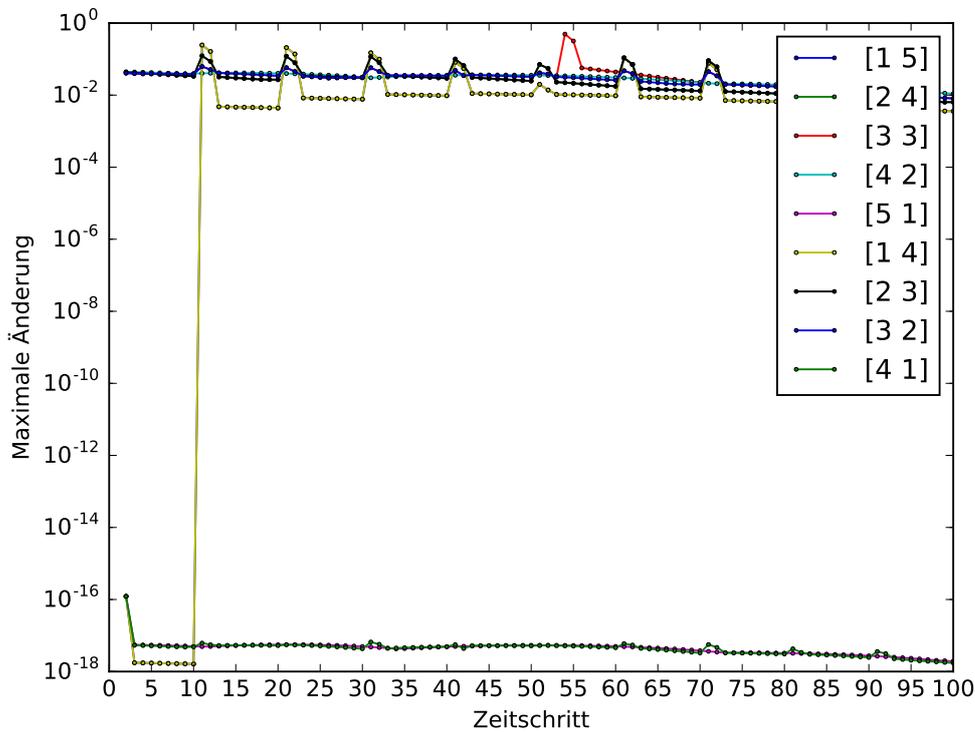


Abbildung 4.8: Szenario 2: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen

4.3 Szenario 3: Advektion, Dirichlet, quadratische Funktion

Dieser Fall beschreibt eine Advektion, wobei der Fehler, anders als in den bisherigen Fällen, in der Mitte der der Kombinationslösung $u_{(5,1)}$, oder genauer gesagt, am Punkt $x_{(5,1),(16,1)} = (0.5, 0.5)$, eingeführt wurde. Genau wie im vorherigen Fall wandert der Fehler dabei mit der Zeit ausschließlich in x -Richtung, was anhand des Kombinationsgitters $u_{(5,1)}$ zum Zeitpunkt t_60 , in Abbildung 4.10, erkannt werden kann (Das grüne Pluszeichen ist genau in der Mitte des Gitters). Betrachtet man die finale, kombinierte Lösung in Abbildung 4.9 so ist kein Fehler erkennbar, da sich dieser nach $T = 60$ Zeitschritte bereits außerhalb des Randes befindet. Da der Fehler in der Mitte des Gitters eingeführt wurde, sollte sich der Fehler mit den von Hinojosa et. al. [HHH] beschriebenen Verfahren leicht erkennen lassen. Wie erwartet lässt sich in anhand der maximalen Unterschiede der Lösungspaare in Tabelle 4.1 erkennen, dass in allen Fällen, in welchen die Kombinationslösung $u_{(5,1)}$ verglichen wurde, sich der zugehörige Wert stark von den anderen Werten unterscheidet. Auffällig ist dabei, dass der maximale Unterschied zwischen den beiden Kombinationslösungen $u_{(4,2)}$ und $u_{(5,1)}$ im Verhältnis zu den weiteren maximalen Unterschieden enorm groß ist. Das zweite von Hinojosa et. al. vorgestellte

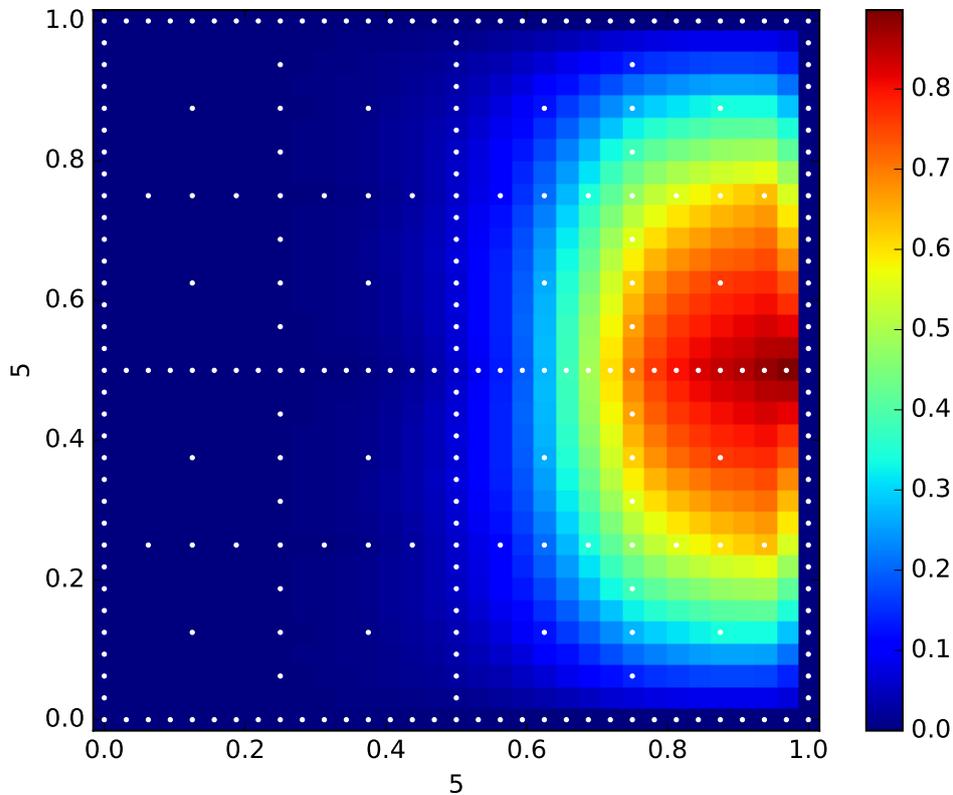


Abbildung 4.9: Szenario 3: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dünnen Gitter

Verfahren erkennt zwar einige Fehler, jedoch wird der eigentliche Fehler in der Mitte des Gitters nicht erkannt (Tabelle 4.2). Dieses Verhalten lässt sich wie folgt erklären: Eine Änderung am Wert eines Gitterpunkts $x_{i,j}^{\vec{s}}$, welcher dem hierarchischen Koeffizienten $\alpha_{\vec{l},k}$ mit einem Level \vec{l} entspricht, hat zur Folge, dass durch die Hierarchisierung alle angrenzenden Koeffizienten auf nächster Ebene mit $\vec{s} = \vec{l} + \vec{e}_k, 1 \leq k \leq d$ (direkte Nachfolger) beeinflusst werden. Einfach gesprochen, müssen, falls ein Gitterwert geändert wird, die Nachfolger auf hierarchischer Ebene diese Änderung ausgleichen, um am entsprechenden Gitterpunkt weiterhin den korrekten Wert beschreiben zu können. Da Hinojosa et. al. [HHH] nicht die Gitterwerte, sondern deren hierarchische Überschüsse vergleicht, können Änderungen auch an anderen Gitterpunkten erkannt werden. Der Grund, weshalb der Fehler am eigentlichen Gitterpunkt nicht erkannt wird, liegt an der verwendeten Methode `outlier_test`, welche den Ausreißer standardgemäß als nicht wahrscheinlich genug einstuft. Eine andere Wahl des Schwellwertes könnte dies beheben, jedoch auch die Anzahl an falsch Positiven erhöhen. Betrachtet man den zeitlichen Verlauf der innerhalb dieser Arbeit vorgestellten Verfahren, so lässt sich erkennen, dass in

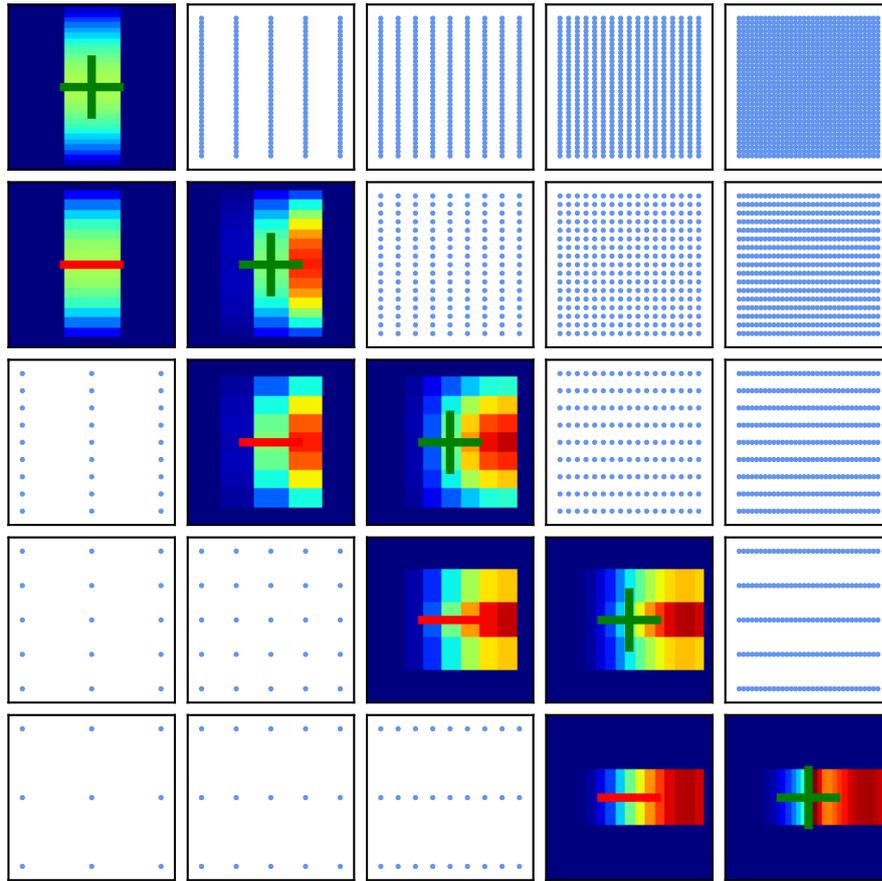


Abbildung 4.10: Szenario 3: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}

beiden Plots der Abbildungen 4.12 und 4.11 zum Zeitschritt t_{54} ein Ausschlag des Wertes der Kombinationslösung $u_{(5,1)}$ erkennbar ist. In Falle der maximalen Änderung verhalten sich die Werte zu den Zeitschritten zwischen den Kombinationsgittern relativ vorhersehbar, wobei im Falle der maximalen Abweichung zwischen den Zeitschritten t_{10} und t_{30} der Werte der Kombinationslösungen $u_{(2,4)}$ und $u_{(2,3)}$ ungewöhnlich stark schwankt. Auch zu Beginn am Zeitpunkt t_5 tritt bei einigen Kombinationslösungen, unter anderem auch bei derjenigen, welche den Fehler enthält, eine Änderung im generellen Trend auf. Jedoch muss erwähnt werden, dass diese Schwankungen relativ gesehen zum Ausschlag am Zeitpunkt des Fehlers, deutlich geringer ausfallen, als dass es das Schaubild darstellt. Weiterhin lässt sich, ähnlich wie im ersten Szenario 4.1 beschreiben, beobachten, wie nach dem Kombinationsschritt zum Zeitschritt t_{60} der Fehler auf die anderen Gitter übertragen wird.

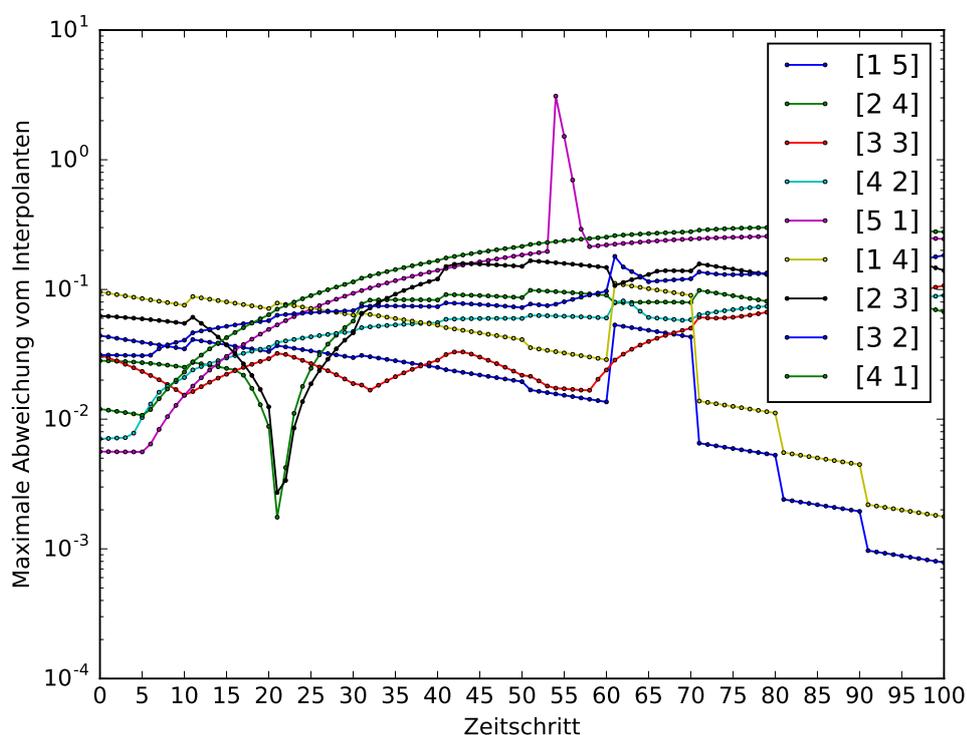


Abbildung 4.11: Szenario 3: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit

4.4 Szenario 4: Advektion-Diffusion, Dirichlet, Sinus Funktion

Das letzte der vorgestellten Szenarien beschreibt ein Advektions-Diffusions Problem, wobei der eingeführte Fehler, mit Faktor 10^{-300} , den Wert am Gitterpunkt nahezu 0 werden lässt. Da durch die Diffusion und die Dirichlet Randbedingungen bereits alle Gitterwerte nach dem Nullpunkt streben, beschreibt dieser Fall somit eine enorme Herausforderung für die in Kapitel 3 vorgestellten Fehlererkennungsverfahren. Anhand der in 4.13 abgebildeten, finalen Dünngitterlösung $u_{\frac{1}{5}}^{(c)}$, lässt sich demnach mit bloßem Auge kein Fehler erkennen. Im Kombinationsgitter $u_{(3,3)}$ hingegen lässt sich nach längerem Betrachten eine winzige Unregelmäßigkeit der Farben links oberhalb des Pluszeichens feststellen. Betrachten wir im Folgenden das Verhalten der Fehlererkennungsverfahren. Ein Teil der Ergebnisse des von Hinojosa et. al. in [HHH] vorgestellten Verfahrens zur Fehlererkennung, anhand des maximalen Unterschieds der hierarchischen Koeffizienten, sind in 4.7 zum Zeitschritt t_{55} dargestellt. In der Tabelle lässt sich erkennen, dass in fast allen Fällen, in welchen die Kombinationslösung $u_{(3,3)}$ mit einer anderen Kombinationslösung verglichen wurde, der maximale Unterschied zu den größten

4 Verhalten der Fehlererkennungsverfahren und Ergebnisse

Gitterpaar	Maximaler Unterschied
[1 5], [2 4]	0.483292747543
[1 5], [3 3]	0.751416760327
[1 5], [4 2]	0.724206139409
[1 5], [5 1]	0.855138816367
[1 5], [1 4]	0.464307982013
[1 5], [2 3]	0.745481536477
[1 5], [3 2]	0.733714455075
[2 4], [3 3]	2.12565808431
[2 4], [4 2]	4.04533815876
[2 4], [5 1]	1.94473241355
[2 4], [1 4]	0.0265122996604
[2 4], [2 3]	2.795378754
[2 4], [3 2]	1.3922328808
[3 3], [4 2]	5.49133729185
[3 3], [5 1]	4.30796149298
[3 3], [1 4]	0.212414180707
[3 3], [2 3]	4.37862906274
[3 3], [3 2]	2.62779304414
[4 2], [5 1]	13.8571708134
[4 2], [1 4]	0.177488725452
[4 2], [2 3]	0.691732381742
[4 2], [3 2]	1.75915744205
[5 1], [1 4]	0.266904803603
[5 1], [2 3]	10.1763748386
[5 1], [3 2]	4.38467670855
[1 4], [2 3]	0.192018043962
[1 4], [3 2]	0.183982110574
[2 3], [3 2]	1.07558883171

Tabelle 4.7: Szenario 4: Ein Teil der verglichenen Lösungspaare mit ihren zugehörigen maximalen Unterscheide (Kapitel 3.1.1) zum Zeitschritt t_{55}

Zeitschr.	Erkanntes Gitter, Punkt	Verglichene Gitter	Verglichene Werte
57	[3 3], [0.375 0.5]	[3 3], [4 2], [5 1], [3 2], [4 1]	[0. 0. 0. 0. 0.]

Tabelle 4.8: Szenario 4: Die Ergebnisse des Vergleichs auf hierarchischer Ebene (Kapitel 3.1.2) zum Zeitschritt t_{57}

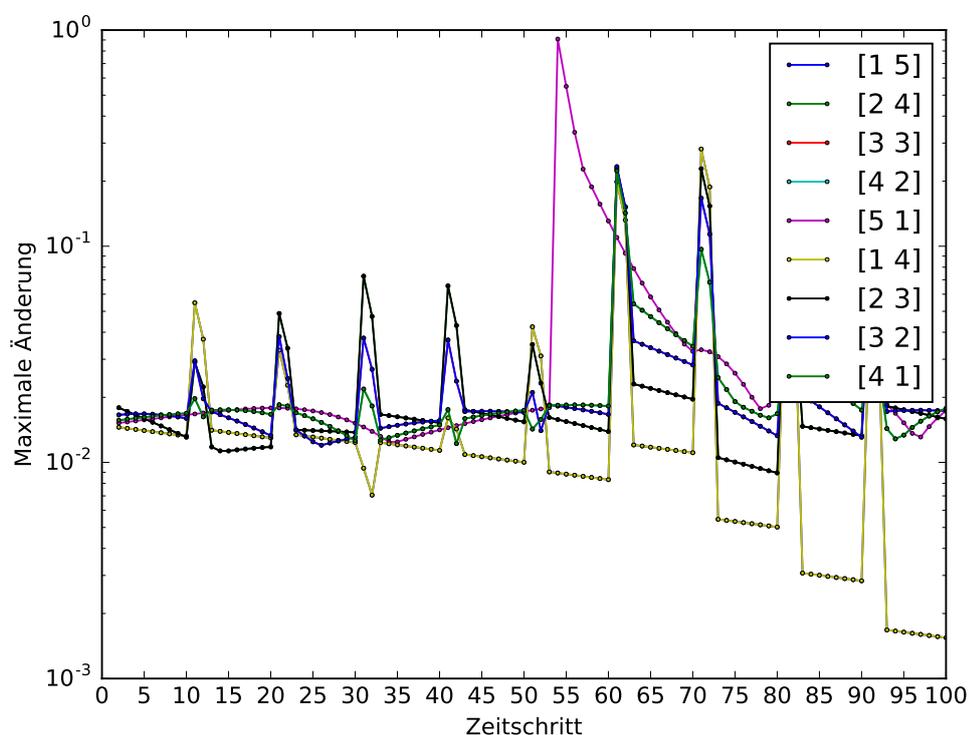


Abbildung 4.12: Szenario 3: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen

Unterschieden zählt, jedoch weitere Lösungspaare existieren, welche einen noch größeren Wert aufweisen. Betrachtet man die nachfolgende Tabelle 4.8, welche das Ergebnis des zweiten von Hinojosa et. al. vorgestellten Verfahrens, zum Zeitschritt t_{57} beschreibt, so erkennt man, dass in der Kombinationslösung $u_{(3,3)}$ tatsächlich ein Fehler erkannt wurde, wobei vor allem im nachfolgenden Zeitschritt vermehrt Fehler in der Kombinationslösung $u_{(3,3)}$ festgestellt wurden. Zu früheren Zeitschritten wurden jedoch keine Fehler erkannt, lediglich zum und vor dem Auftreten des Fehlers wurden viele falsch Positive Fehler erkannt. Das Verfahren zur Erkennung eines Fehlers anhand der maximalen Abweichung vom Interpolanten, welches in Abbildung 4.15 dargestellt ist, reagiert ähnlich, wie im zweiten Szenario, jedoch ist hierbei kein Anzeichen einer fehlerhaften Kombinationslösung ersichtlich, was auf den zu kleinen Fehler und demnach auch der zu geringen Abweichung vom Interpolanten zurückzuführen ist. Das Verfahren zur Fehlererkennung anhand der maximalen Änderung an einem Gitterwert innerhalb einer Kombinationslösung hingegen schlägt zum Zeitpunkt des Fehlers geringfügig aus und zeigt somit, dass es zumindest theoretisch möglich ist, einen kleinen Fehler frühzeitig zu erkennen.

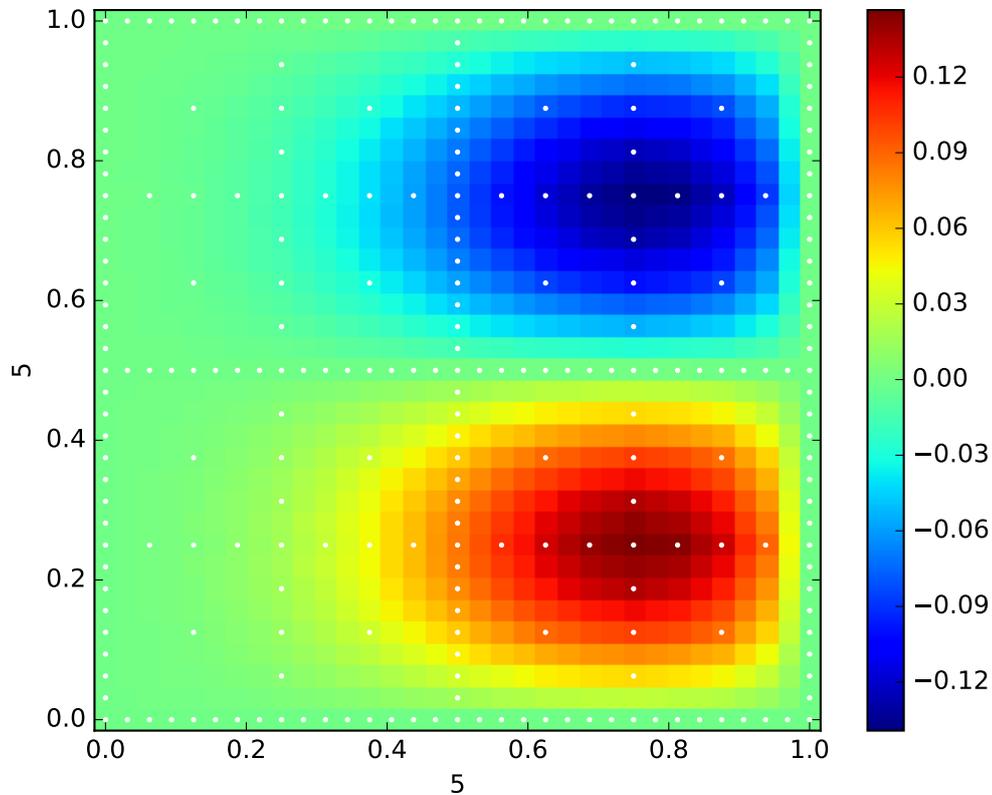


Abbildung 4.13: Szenario 4: Kombinierte Lösung nach $T = 100$ Zeitschritten auf dem Dünnen Gitter

4.5 Ergebnis

Die zuvor beschriebenen Untersuchungen zeigen, dass sich die innerhalb dieser Arbeit vorgestellten Verfahren grundsätzlich zur Erkennung eines Fehlers eignen. Demnach konnte in allen durchgeführten Simulationen ein ungewöhnlicher Anstieg der betrachteten Werte, in mindestens einem der beiden Ergebnisplots festgestellt werden. Lediglich im zuletzt beschriebenen Szenario, in welchem der fehlerhafte Wert nahe am Nullpunkt lag und umliegende Punkte ebenfalls kleine Werte aufwiesen, konnte kein Ausschlag anhand der maximalen Abweichung vom Interpolanten festgestellt werden. Da der eingeführte Fehler die finale Lösung jedoch nur unwesentlich beeinträchtigt hat, könnte dies auch bedeuten, dass das Verfahren toleranter gegenüber kleinen Unstimmigkeiten ist und somit ein möglicher Erkennungsalgorithmus eventuell weniger falsch positive Werte zurückliefert. Bemerkenswert ist, dass sofern ein Ausschlag der Werte erkennbar war, dieser direkt zum Zeitpunkt des Fehlers sichtbar wurde, wohingegen die von Hinojosa et. al. vorgestellten Verfahren besonders bei Fehlern an Gitterpunkten auf

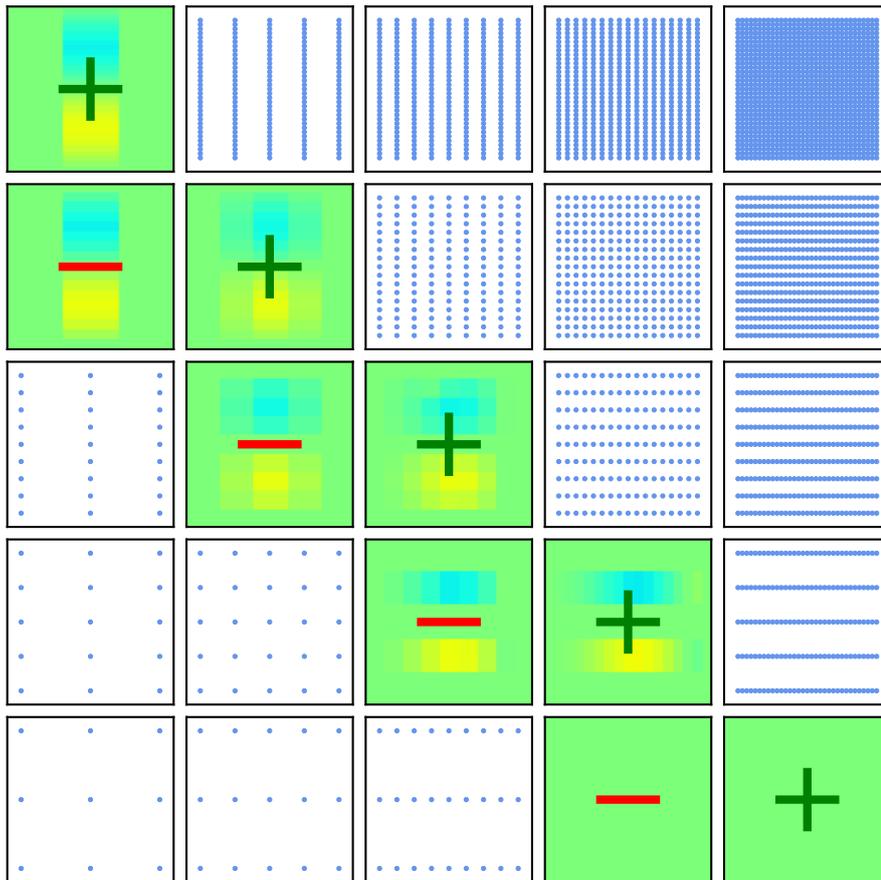


Abbildung 4.14: Szenario 4: Kombinationsgitter vor der Kombination zum Zeitschritt t_{60}

hohem hierarchischen Level, ohne Vergleichswerte, diesen erst zum darauffolgenden oder wie im letzten Fall beschrieben bis zu drei Zeitschritte später, meldeten. Vergleicht man die Ausschläge der beiden Verfahren zur Fehlererkennung, anhand der zeitlichen Entwicklung einer Kombinationslösung, so stellt man fest, dass der Ausschlag in den Plots des Verfahrens: maximalen Abweichung vom Interpolanten, relativ zum Ausschlag in Plots des Verfahrens: maximale Änderung der Gitterwerte größer ist. Dies liegt vorallem daran, dass durch das Produkt der beiden Maße in (3.9) eine Abweichung doppelt gewichtet wird.

4 Verhalten der Fehlererkennungsverfahren und Ergebnisse

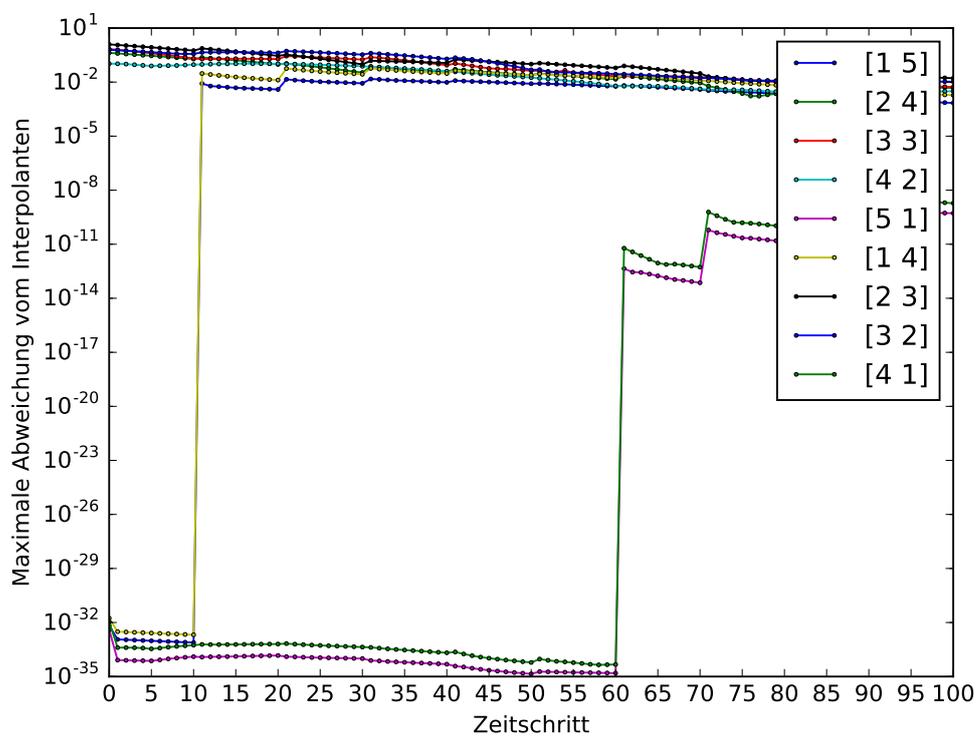


Abbildung 4.15: Szenario 4: Plots zur maximalen Abweichung vom Interpolanten (Kapitel 3.2.2) der einzelnen Kombinationslösungen über die Zeit

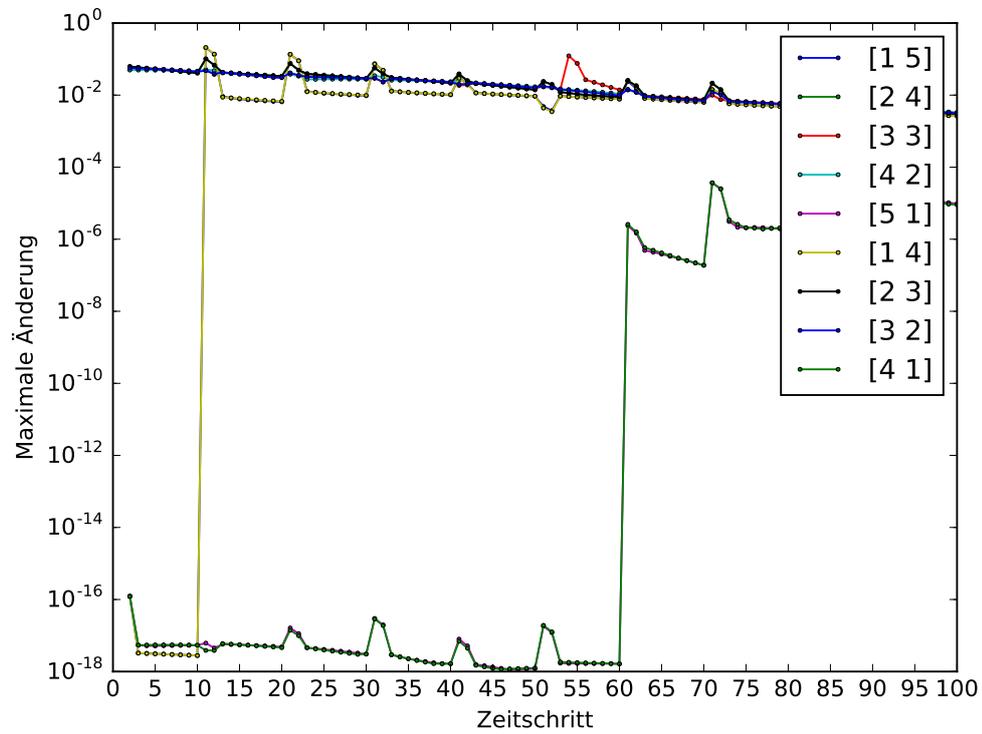


Abbildung 4.16: Szenario 4: Plots zur maximalen Änderung an den Gitterwerten, gemittelt über die letzten $p = 2$ Zeitschritte (Kapitel 3.2.1), für alle Kombinationslösungen

5 Zusammenfassung und Ausblick

Innerhalb dieser Arbeit wurden zwei neue Methoden zur Fehlererkennung während einer Simulation einer PDE mit der Dünnmitter Kombinationstechnik vorgestellt und anhand von vier unterschiedlichen Szenarien evaluiert. Die zugrundeliegende Idee der Methode ist es, den zeitlichen Verlauf einzelner Kombinationlösungen auszunutzen um Fehler, wie Silent Faults aufzuspüren. Während sich die Gitterwerte eines Kombinationsgitters bei einem ausreichend glatten Advektions- und/oder Diffusionsproblem nicht wie im Fehlerfall schlagartig ändern sollten, ist es eine Möglichkeit nach solchen unnatürlichen Veränderungen der Werte an den Gitterpunkten zu suchen um aufgetretene Fehler zu enttarnen.

Zu Beginn der Arbeit werden die Grundlagen der Gitter, dünnen Gitter und die Kombinationstechnik beschrieben. Dabei werden Notationen und Definitionen erläutert, welche in folgenden Kapiteln verwendet werden.

Im Anschluss wird speziell auf die Auswirkung der Silent Faults auf die Kombinationstechnik eingegangen. Weiterhin werden die von Hinojosa et. al. in [HHH] vorgestellten Fehlererkennungsverfahren erläutert, welche hauptsächlich auf dem Vergleich von Kombinationslösungen beruhen. Darauf folgend werden die innerhalb dieser Arbeit entwickelten Methoden zur Fehlererkennung anhand der zeitlichen Entwicklung einer Kombinationslösung, vorgestellt und beschrieben.

Zuletzt werden die zuvor beschriebenen Verfahren anhand von vier unterschiedlichen Advektions und Diffusionsproblemen analysiert und mit zwei der von Hinojosa et. al. [HHH] vorgestellten Verfahren verglichen.

Ausblick

Während es innerhalb dieser Arbeit aufgrund der eingeschränkten Bearbeitungszeit lediglich möglich war, die Fehlereerkennungsmethoden zu beschreiben und deren grundsätzliches Verhalten zu untersuchen, konnten die in Kapitel 3 vorgestellten automatisierten Fehlererkennungsverfahren weder ausreichend getestet noch optimiert werden. Dies wäre demnach ein nächster Schritt, zumal die Ergebnisse der vorgestellten Methoden vielversprechend für eine automatische Erkennung ausfielen. Weiterhin könnten somit Zahlen für *precision* und *recall* erhoben werden, was einen genaueren Vergleich zu den von Hinojosa et. al. vorgestellten

Verfahren ermöglicht. Ein weiterer Punkt ist die Untersuchung der Methoden, im Falle von höherdimensionalen Problemen. Während hier ausschließlich zweidimensionale PDEs betrachtet wurden, kann es sein, dass sich die Verfahren im mehrdimensionalen Fall anders verhalten. Da die Berechnung der Gitter und der Kombinationstechnik, welche innerhalb dieser Arbeit implementiert wurden, auch d -dimensionale Fälle unterstützt, sind bis auf die Lösung der PDEs im mehrdimensionalen Fall keine großartigen Änderungen am Quelltext nötig.

Literaturverzeichnis

- [EHM14a] J. Elliott, M. Hoemmen, F. Mueller. „Evaluating the impact of SDC on the GMRES iterative solver“. In: *Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS* (2014), S. 1193–1202 (zitiert auf S. 36).
- [EHM14b] J. Elliott, M. Hoemmen, F. Mueller. „Resilience in Numerical Methods: A Position on Fault Models and Methodologies“. In: (2014). arXiv: [arXiv:1401.3013v1](https://arxiv.org/abs/1401.3013v1) (zitiert auf S. 13, 14).
- [Gar13] J. Garcke. „Sparse grids in a nutshell“. In: *Lecture Notes in Computational Science and Engineering* 88 (2013), S. 57–80 (zitiert auf S. 21).
- [Geo] Geographic Information Training Alliance. *GITTA*. URL: http://www.gitta.info/ContiSpatVar/de/html/Interpolatio%7B%5C_%7DlearningObject2.xhtml (zitiert auf S. 30).
- [GGT01] J. Garcke, M. Griebel, M. Thess. „Data mining with sparse grids“. In: *Computing (Vienna/New York)* 67.3 (2001), S. 225–253 (zitiert auf S. 17).
- [GHZ96] M. Griebel, W. Huber, C. Zenger. „Numerical Turbulence Simulation On A Parallel Computer Using The Combination Method“. In: *Flow simulation on high performance computers II* (1996), S. 34–47 (zitiert auf S. 23).
- [GSZ] M. Griebel, M. Schneider, C. Zenger. „GRID PROBLEMS i X i X“. In: (), S. 1–19 (zitiert auf S. 17, 21, 22, 29).
- [Hac91] W. Hackbusch. *Parallel algorithms for partial differential equations : proceedings of the Sixth GAMM-Seminar, Kiel, January 19-21, 1990*. 31. 1991, 255 p. (Zitiert auf S. 17).
- [HHH] A. P. Hinojosa, B. Harding, M. Hegland. „SDC-Resilient Algorithms Using the Sparse Grid Combination Technique“. In: (), S. 1–21 (zitiert auf S. 11, 14, 15, 18, 20, 21, 23–27, 29, 33, 34, 36, 48, 49, 51, 59).
- [HHJP15] P. Hupp, M. Heene, R. Jacob, D. Pflüger. „Global Communication Schemes for the Numerical Solution of High-Dimensional {PDEs}“. In: *Parallel Computing* (2015) (zitiert auf S. 22, 24, 33, 34).
- [HHLS15] B. Harding, M. Hegland, J. Larson, J. Southern. „Fault Tolerant Computation with the Sparse Grid Combination Technique“. In: *SIAM Journal on Scientific Computing* 37.3 (Jan. 2015), S. C331–C353. URL: <http://epubs.siam.org/doi/10.1137/140964448> (zitiert auf S. 24, 25).

- [HKP14] M. Heene, C. Kowitz, D. Pflüger. „Load balancing for massively parallel computations with the sparse grid combination technique“. In: *Advances in Parallel Computing* 25 (2014), S. 574–583 (zitiert auf S. 25).
- [Hul] D. Hull. *Outlier Analysis - Hull.PDF* (zitiert auf S. 29).
- [Meh14] M. Mehl. „Numerische Simulation“. In: (2014) (zitiert auf S. 35, 36).
- [MP11] A. Muraras, D. Pfl. „Compact Data Structure and Scalable“. In: *Evaluation* (2011), S. 25–34 (zitiert auf S. 18).
- [Pfl10] D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems Spatially Adaptive Sparse Grids for High-Dimensional Problems Pflüger*. 2010 (zitiert auf S. 18, 21).
- [Pro] Prometeus GmbH. *No Title*. URL: <http://www.top500.org/> (zitiert auf S. 13).
- [RW05] C. Reisinger, G. Wittum. „Efficient hierarchical approximation of high-dimensional option pricing problems“. In: *SIAM Journal on Scientific Computing* 29.1 (2005), S. 440–458. arXiv: [arXiv:0710.0491v1](https://arxiv.org/abs/0710.0491v1). URL: <http://eprints.maths.ox.ac.uk/242> (zitiert auf S. 17).
- [SWA+14] M. Snir, R. W. Wisniewski, J. a. Abraham, S. V. Adve, S. Bagchi, P. Balaji, J. Belak, P. Bose, F. Cappello, B. Carlson, A. a. Chien, P. Coteus, N. a. DeBardeleben, P. C. Diniz, C. Engelmann, M. Erez, S. Fazzari, A. Geist, R. Gupta, F. Johnson. „Addressing failures in exascale computing.“ In: *International Journal of High Performance Computing Applications* 28.2 (2014), S. 129–173. URL: [10.1177/1094342014522573](https://doi.org/10.1177/1094342014522573) <http://search.ebscohost.com/login.aspx?direct=true%7B%5C%7Ddb=bth%7B%5C%7DAN=96010112%7B%5C%7Dsite=ehost-live> (zitiert auf S. 13, 14).

Alle URLs wurden zuletzt am 16. 06. 2016 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