

DIPLOMARBEIT

Eine adaptive Hamilton-Jacobi Level-Set-Reinitialisierung für Zweiphasenströmungen

angefertigt am

Institut für Numerische Simulation

vorgelegt der

Mathematisch-Naturwissenschaftlichen Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Juli 2008

von

Markus Bürger

aus

Coesfeld

Inhaltsverzeichnis

1	Einleitung	1
1.1	Numerische Simulation von Zweiphasenströmungen	1
1.2	Die Problemstellung	4
1.3	Der Lösungsansatz	5
1.4	Zu dieser Arbeit	7
1.5	Dank	8
2	Grundlagen	9
2.1	Diskretisierungen und Gittertypen	9
2.1.1	Strukturierte Gitter	10
2.1.2	Unstrukturierte Gitter	17
2.2	Ghost-Cell-Immersed-Boundary-Methode	18
2.2.1	Dirichlet-Bedingungen	19
2.2.2	Neumann-Bedingungen	22
2.2.3	Ergebnis	23
2.3	Anforderungen an das Gitter	23
2.3.1	Komplexe Geometrien	23
2.3.2	Lokale Verfeinerung	24
2.4	Level-Set-Methode	25
2.4.1	Grenzflächen	26
2.4.2	Die Level-Set-Funktion	27
2.4.3	Level-Set-Reinitialisierung	30
3	Adaptivität	33
3.1	Verfahren mit octrees	35
3.1.1	Geometrieerzeugung mit constructive solid geometry	35
3.1.2	Approximation von Oberflächen mit octrees	36
3.1.3	Fully threaded trees	36
3.2	Numerik auf octrees	40
3.3	Mehrgitter	42
3.3.1	Mehrgitter auf composite grids	44
3.4	Gebietszerlegungsmethoden	48
3.4.1	Mortar-Methoden	49

3.4.2	Mortar-Finite-Elemente-Methode	50
3.4.3	Mortar-Mimetische-Finite-Differenzen-Methode	53
4	Implementierung der adaptiven Level-Set-Methode auf kartesischen Gittern	57
4.1	Numerische Diffusion bei Level-Set-Methoden	57
4.1.1	Das Problem am Beispiel von Strömungssimulationen	58
4.1.2	Klassische Lösungsansätze	60
4.2	Das Zweiparameterproblem	68
4.2.1	Ein Zweigitterverfahren	70
4.2.2	Kartesische grid patches	75
4.2.3	Das patch grid	76
4.2.4	Datenstruktur	78
4.2.5	Der Algorithmus	79
4.3	Numerik	80
4.3.1	Transferoperatoren	83
4.3.2	Randbedingungen	84
5	Numerische Ergebnisse	85
5.1	Diskretisierungsfehler	87
5.1.1	Prolongation und Restriktion	87
5.1.2	Differentialoperatoren	89
5.2	Reinitialisierung	90
5.2.1	Numerische Diffusion	91
5.2.2	Erzeugung einer Abstandsfunktion und Erhaltung der Nullkontur .	97
5.3	Dynamische Systeme mit Konvektion	100
5.4	Test in NaSt3DGPF	101
6	Zusammenfassung und Ausblick	117
	Literaturverzeichnis	121

Kapitel 1

Einleitung

1.1 Numerische Simulation von Zweiphasenströmungen

Eines der wohl wichtigsten Hilfsmittel im Bereich der natur- und ingenieurwissenschaftlichen Forschung ist die numerische Simulation. Sie liefert rasch aussagekräftige Ergebnisse über Phänomene, die weder experimentell ermittelt noch als geschlossene Lösung berechnet werden können. So erlauben numerische Simulationen zum Beispiel Aussagen über die Entstehung des Weltalls und seine Entwicklung nach dem Urknall, die Entstehung von Galaxien, Sternen und Planeten. Aber nicht nur der Blick in die Vergangenheit wird ermöglicht, sondern auch Prognosen für zukünftige Ereignisse. Schon zu einer Selbstverständlichkeit ist die sichere Vorhersage des Wetters geworden, aber auch längerfristige globale Klimavorhersagen werden aufgrund numerischer Simulationen gemacht. An besonderer Stelle stehen auch die Prognosen von Naturkatastrophen wie Tsunamis, Erdbeben, Hochwasser und Wirbelstürmen, die der Frühwarnung der Bevölkerung dienen. Im technischen Umfeld wird numerische Simulation ebenfalls sehr vielfältig eingesetzt. Beispielsweise in der Materialforschung oder der Entwicklung von effizienteren und ökologischeren Antriebs- und Energiegewinnungstechnologien.

Neben dem wissenschaftlichen Zweig spielt numerische Simulation aber auch in der Industrie eine ganz wesentliche Rolle. Weil Simulationen schneller und weniger kostenintensiv sind als Experimente an realen Modellen, erfreuen sie sich einer immer größer werdenden Beliebtheit. Als Beispiel sei die Automobilindustrie erwähnt: Kein Fahrzeug wird gebaut, ohne dass es zuvor zahlreiche Simulationsversuche auf dem Computer durchlaufen hat. So optimieren Ingenieure anhand von Simulationsergebnissen unter anderem den Luftwiderstand, das Fahrverhalten auf der Straße, die Fahrzeug-Akustik und die Haltbarkeit der verbauten Materialien, bevor ein erster Prototyp gebaut wird. Nicht zuletzt machen numerische Simulationen auch virtuelle Crash-Tests an Fahrzeugen möglich.

Die numerische Simulation hat mit der rasanten Entwicklung von Computern stark zugenommen. Durch immer schneller werdende und kostengünstigere Computerchips können Simulationen immer umfangreicher und genauer werden. Die technischen Grenzen der numerischen Simulation rücken durch die schnell voranschreitende Entwicklung von Computersystemen in die Ferne, was den Raum für neuere und höhere Ansprüche an die Leistung von Simulationen schafft.

Ein wichtiges Teilgebiet der numerischen Simulation ist die numerische Strömungssimulation. Diese befasst sich mit der Modellierung und Berechnung von Flüssigkeiten und Gasen – sogenannten Fluiden – und grenzt sich somit von anderen großen Bereichen wie der numerischen Strukturmechanik und der Moleküldynamik ab. Wichtige Anwendungen der Strömungssimulation liegen in der Aero- und Hydrodynamik, aber auch in großskaligen Wetter- und Klimavorhersagen [1] sowie der Berechnung von Strömungen in porösen Medien [34]. Auch die Wechselwirkung von Fluiden mit Festkörperstrukturen [21] fallen in das Forschungs- und Anwendungsgebiet der Strömungssimulation. Man unterscheidet einphasige Strömungen, an denen nur ein Fluid beteiligt ist, von zwei- oder mehrphasigen Strömungen mit der Beteiligung zweier oder mehrerer Fluide, die sich untereinander nicht mischen, wobei also keines ihrer Moleküle von einem Fluid in das andere wechselt. Die Trennschicht der beiden Phasen bildet eine in der Zeit veränderliche Oberfläche. Sie wird daher auch freie Oberfläche genannt. Wichtige Anwendungen



Abbildung 1.1: Dynamik von Wassertropfen

der Zweiphasenströmungssimulation sind sogenannte Beschichtungsvorgänge, bei denen dünne Fluidfilme, zum Beispiel Lacke, auf feste Oberflächen appliziert werden. Dabei ist es von großem Interesse, die Schichten möglichst gleichmäßig und frei von Lufteinschlüssen aufzutragen. Die numerische Simulation hilft hierbei, diese Vorgänge zu optimieren. Eine andere Anwendung der Zweiphasenströmungssimulation im kleinskaligen Bereich ist die Tröpfchendynamik (siehe dazu auch Abbildung 1.1¹). Sie untersucht die Entstehung, die Bewegung und die Agglomeration – das Zusammenschließen – von kleinsten Flüssigkeitströpfchen, die von Gasen umgeben sind. Einen Nutzen daraus zieht beispielsweise die Pharmaindustrie bei der Entwicklung von neuen Herstellungsverfahren für Inhalate und von Medikamenten, die in Form von Sprays appliziert werden. An anderer Stelle dient die Tröpfchensimulation dem Verständnis der Bildung von Kraftstoff-Luft-Gemischen bei

¹Bildherkunft: NASA

Verbrennungsprozessen. Hier steht ein großes wirtschaftliches und ökologisches Interesse im Vordergrund. So kann die numerische Zweiphasenströmungssimulation doch zu einer Optimierung des Wirkungsgrades von Verbrennungsprozessen in Motoren, Kraftwerken, Flugzeugen und Raketenantrieben beitragen. Im großskaligen Bereich sind der Schiffbau, die Ozeanologie, die weiträumige Simulation von Gewässern sowie die Klimaforschung als Beispiele für Anwendungen der Zweiphasenströmungssimulation zu nennen.

Zweiphasenströmungen sind anspruchsvolle und komplexe fluiddynamische Phänomene, da neben den bei herkömmlichen einphasigen Strömungen auftretenden Kräften ferner Kelvin-Helmholtz- und Rayleigh-Taylor-Instabilitäten (siehe Ambrose und David [2], Oran und Boris [20] und Dimonte [24]) mit regularisierenden Oberflächenspannungen in nichtlinearem Zusammenspiel auf das Fluid einwirken. Bisher ist es nicht gelungen, die Phänomene von Zweiphasenströmungen ausschließlich mithilfe analytischer Methoden zu beschreiben. Daher werden auch hier numerische Verfahren zur Berechnung eingesetzt. Erste Versuche, die freie Oberfläche in Strömungssimulationen zu repräsentieren, reichen bis in die 60er Jahre zurück. Sie verwenden Partikel, die die Zellen des Gitters derart markieren, dass eine Zuordnung zu einem der zwei Fluide möglich wird (Marker-and-Cell-Methode). Spätere Entwicklungen in den 70er Jahren machen von einer skalaren Funktion Gebrauch, die jeder Gitterzelle die Volumenanteile der beiden Fluide zuordnet (Volume-of-Fluid-Methode). Die Schwierigkeit dieser Methode liegt in der Rekonstruktion der freien Oberfläche aus den einzelnen Volumenanteilen. Ein ganz anderes Verfahren hingegen ist die in den 80er Jahren von Stanley Osher und James A. Sethian [36] entwickelte Level-Set-Methode. Sie repräsentiert die freie Oberfläche implizit durch die Nullkontur einer skalaren Funktion – der sogenannten Level-Set-Funktion.

Die Numerik als Teilgebiet der Mathematik befasst sich mit der Entwicklung und Analyse von Algorithmen und Methoden zur Lösung von sowohl diskreten als auch kontinuierlichen mathematischen Problemen. Auch wenn die Numerik durch das Wirken von Archimedes, der für die Berechnung sowohl von Flächen wie auch für die Kreiszahl π Algorithmen lieferte, bis in die Antike zurückverfolgt werden kann, gewann sie doch erst durch die Entwicklung von Computern ab den 1930er Jahren dramatisch an Bedeutung. Diese Umstände verdankt sie vor allem der Pionierarbeit von Konrad Zuse und John von Neumann. Die Numerik bildet die aus Modellen hergeleiteten kontinuierlichen Gleichungen auf diskrete Gleichungen ab. Deren Lösungen liegen in endlichdimensionalen Funktionenräumen, welche Unterräume der Lösungsräume der kontinuierlichen Probleme sind. Die Analyse solcher Lösungsansätze untersucht den Fehler, der durch die Diskretisierung entsteht – den sogenannten Diskretisierungsfehler. Er wird stets in Bezug auf die Maschenweite h der Diskretisierung betrachtet. Die Lösung eines stabilen numerischen Verfahrens konvergiert gegen die Lösung des kontinuierlichen Problems, wenn der Fehler für kleiner werdende Maschenweite h gegen Null geht. Die Maschenweite einer Diskretisierung ist jedoch aus technischen Gründen – etwa durch die nur endliche Speicherkapazität und Rechengeschwindigkeit eines Computers – nach unten beschränkt. Das heißt, h kann sehr kleine Werte, aber nicht den Wert Null annehmen. Es ist daher

die Aufgabe der Numerik, Verfahren auf eine schnelle Konvergenz gegen die kontinuierliche Lösung für das kleiner werdende h hin zu optimieren. Ferner kann ein numerisches Verfahren nur dann brauchbare Ergebnisse liefern, wenn die Stabilität des Verfahrens sichergestellt ist. Aus diesen Aspekten lassen sich interessante Problemstellungen für numerische Simulationen von Zweiphasenströmungen mit der Level-Set-Methode formulieren.

1.2 Die Problemstellung

Der vorhergehende Abschnitt erwähnt den Begriff der Stabilität von numerischen Verfahren. Dieser oder mehr noch sein Gegensatz – die Instabilität – soll in Bezug auf die Level-Set-Methode erläutert und die Konsequenzen für Zweiphasenströmungssimulationen dabei herausgestellt werden. Wie schon erwähnt, identifiziert die Level-Set-Methode die freie Oberfläche implizit über die Nullniveaumenge einer skalaren Funktion. Diese Nullniveaumenge nennt man auch Nullkontur oder Nulllevel. Da die Fluidparameter wie Dichte und Viskosität über die freie Oberfläche hinweg starke Sprünge haben, ist aus Gründen der Stabilität (des Strömungslösers) eine Regularisierung dieser Parameter in einer Umgebung der Nullkontur notwendig. Ferner wird die Oberflächenkraft über ein Deltafunktional im Bereich der Nullkontur aktiviert. Damit die Regularisierung und das Deltafunktional korrekt ausgewertet werden können, wählt man als Level-Set-Funktion eine sogenannte vorzeichenbehaftete Abstandsfunktion. Diese ordnet jedem Punkt im Raum seinen Abstand zur Nullkontur zu, mit positivem Vorzeichen, falls der Punkt auf der einen, und mit negativem Vorzeichen, falls der Punkt auf der anderen Seite der Nullkontur liegt. Damit die Nullkontur stets die freie Oberfläche der Fluide repräsentiert, wird die Level-Set-Funktion in jedem Zeitschritt des Strömungslösers gemäß dem Geschwindigkeitsfeld im Gebiet transportiert. Dieser Transport zerstört im Allgemeinen die gewünschte Abstandseigenschaft. Durch das iterative Lösen einer Gleichung vom Hamilton-Jacobi-Typ kann die Abstandseigenschaft der Level-Set-Funktion wiederhergestellt werden. Dieser Prozess heißt Reinitialisierung. Aus analytischer Sicht ist die Reinitialisierung für kontinuierliche Level-Set-Funktionen unproblematisch. Im Diskreten jedoch zeigt sich, dass die Genauigkeit der Reinitialisierung nicht nur vom Diskretisierungsfehler abhängt, sondern auch deutlich unter Instabilitäten des Verfahrens leidet. Diese zeigen sich in dem wohl bekanntesten Problem – der Massenerhaltung beziehungsweise der numerischen Diffusion: Die Reinitialisierung verursacht eine künstliche Verschiebung der Nullkontur und somit eine Verschiebung der freien Oberfläche. Dadurch wird auf künstliche Weise Fluidmasse von der einen Phase in die andere transportiert. Man bezeichnet dieses Phänomen auch als numerische Diffusion. Daraus ergeben sich mehrere Probleme: Der künstliche Massentransport stört die Massenbilanz der beiden Phasen. Ferner wird die Energie- und Kräftebilanz gestört, indem sich durch die Verschiebung der freien Oberfläche lokal die Fluidparameter wie Dichte und Viskosität ver-

ändern. Neben diesen Effekten, die zu einer Instabilität des gesamten Strömungssystems führen können, besitzt die numerische Diffusion weitere unerwünschte Eigenschaften, die ein physikalisch inkorrektes Verhalten der Fluide verursachen. Sie zeigen sich zum Beispiel bei dünnen Fluidfilmen, die auseinanderreißen oder an ihrer Ausbreitung gehindert werden. Oft kann man auch beobachten, dass sich an filigranen Strukturen der freien Oberfläche Satellitentröpfchen abschnüren. Da die künstliche Verschiebung der Nullkontur stets in Krümmungsrichtung der freien Oberfläche erfolgt, verlieren Fluidtröpfchen an Masse oder erleiden sogar den totalen Massenverlust; Bereiche auf der Oberfläche mit starken Krümmungen neigen dazu, künstlich aufzuweichen. Zusammenfassend lässt sich sagen, dass die Instabilitäten der Level-Set-Reinitialisierung Störungen an der freien Oberfläche verursachen, die zum einen die physikalischen und topologischen Strukturen der Oberfläche selbst zerstören und zum anderen die Stabilität des Strömungslösers negativ beeinflussen.

Es gibt klassische Lösungsansätze, die die oben genannten Probleme reduzieren: Durch die Wahl von Diskretisierungen höherer Ordnung sowohl für die räumlichen Differentialoperatoren (WENO/ENO) wie auch für die Zeitschrittverfahren (Runge-Kutta) lässt sich eine deutlich bessere Massenerhaltung erzielen (siehe Croce [18]). Darüber hinaus stabilisiert eine geeignete Regularisierung der Vorzeichenfunktion in der Reinitialisierungsgleichung den Prozess der Reinitialisierung wesentlich (siehe Croce [18]).

Es lässt sich nun eine essentielle Aussage formulieren: Mittels einer geeignet regulierten Vorzeichenfunktion kann die künstliche Verschiebung der Nullkontur auf ein Maximum von nur einer Gittermaschenweite eingeschränkt werden.

1.3 Der Lösungsansatz

Ziel ist es nun, eine im Hinblick auf die Massenerhaltung beziehungsweise numerische Diffusion hin optimierte Erweiterung des in das Strömungsprogramm NaSt3DGPF integrierte Level-Set-Verfahrens zu konstruieren.

Die numerische Diffusion der Level-Set-Reinitialisierung zeigt sich in einer künstlichen Verschiebung der Nullkontur. Durch klassische Lösungsansätze kann diese künstliche Verschiebung auf den Rahmen von einer Gittermaschenweite beschränkt werden. Eine erste Idee, die numerische Diffusion zu reduzieren, ist, die Maschenweite zu verkleinern. Eine globale Verfeinerung des Gitters verursacht jedoch enorm hohen Aufwand in Rechenzeit und Speicher. Zu beachten ist ferner die Tatsache, dass numerische Diffusion ein intrinsisches Problem der Level-Set-Methode ist und auf Instabilitäten der Level-Set-Reinitialisierung – genauer gesagt des Hamilton-Jacobi-Systems – zurückzuführen ist. So kann numerische Diffusion auftreten, obwohl die Diskretisierungsgenauigkeit des Gitters an sich dem Strömungsproblem angemessen ist. Darüber hinaus betrifft der Effekt numerischer Diffusion zwar die Level-Set-Funktion in globaler Weise, ist aber lediglich in einer Umgebung um die Nullkontur – also der freien Oberfläche im Strömungsproblem

– von Interesse. Eine zweite Idee führt daher zu einer adaptiven Verfeinerung der Level-Set-Funktion. Die Gitter der Strömungsgrößen können unverändert bestehen bleiben, weshalb der Strömungslöser weiterhin auf dem globalen groben Gitter operiert. Es bezeichnen H die Maschenweite des globalen groben Gitters und h die feine Maschenweite des adaptiven Gitters. Da für den Transport der Level-Set-Funktion nur Geschwindigkeitswerte des globalen groben Gitters zur Verfügung stehen, kann dieser – ungeachtet der adaptiven Feinheit der Level-Set-Funktion – nur mit der Genauigkeit des groben Gitters durchgeführt werden. Reinitialisiert man die Level-Set-Funktion jedoch auf dem adaptiven Gitter, so profitiert man von der kleineren Maschenweite h , da sich die numerische Diffusion nun im Rahmen einer substantiell kleineren Maschenweite h anstatt H bewegt.

Eine offene Frage ist die nach einem Verfeinerungskriterium. Gesucht ist also eine Regel, nach der die groben Gitterzellen der Level-Set-Funktion verfeinert werden müssen. Ein guter Ansatz dabei ist die lokale Verfeinerung in einer Umgebung der Nullkontur.

Da dieses adaptive Verfahren eine Erweiterung des Strömungslösers NaSt3DGPF sein soll, wird großer Wert auf die Wiederverwendbarkeit von bestehenden Finite-Differenzen-Schemata gelegt. Das in dieser Arbeit entwickelte adaptive Verfahren macht daher von Gittern mit kartesischen Eigenschaften Gebrauch. Dazu werden im Rahmen dieser Arbeit fully threaded trees als spezielle Form von octrees implementiert, an denen gezeigt wird, dass sie sich hervorragend zur adaptiven Verfeinerung der Umgebung von freien Oberflächen und komplexen Gebietsrändern eignen. In dieser Arbeit wird beschrieben, wie der in der Hamilton-Jacobi-Iteration der Reinitialisierungsgleichung vorkommende Differentialoperator ∇ auf octrees berechnet werden kann [39]. Ein anderes Verfahren zur konsistenten Diskretisierung mittels finiter Differenzen auf Gittern mit hängenden Knoten, welches auf der Mortar-Methode basiert, ist die sogenannte Mortar-Mimetische-Finite-Differenzen-Methode. Im Rahmen dieser Arbeit ist diese Methode für die Verwendung auf octrees und beliebigen kartesischen Gittern mit hoher Irregularität formuliert worden. Ferner soll die Erweiterung möglichst modular gehalten sein, damit das Programm NaSt3DGPF nur so wenig wie möglich modifiziert werden muss. Ein Verfahren, dass diese Bedingungen erfüllen kann, setzt gemäß dem Verfeinerungskriterium auf grobe Gitterzellen feine grid patches auf; diese grid patches sind kartesische strukturierte Gitter der Maschenweite h . Es resultiert ein blockstrukturiertes Gitter, das aus dem globalen groben Gitter mit der Maschenweite H und zahlreichen kartesischen strukturierten Gittern mit der feinen Maschenweite h – den sogenannten grid patches – besteht. Dieses auf blockstrukturierten Gittern basierende adaptierende Verfahren für die Level-Set-Reinitialisierung ist im Rahmen dieser Arbeit entwickelt und implementiert worden. Die Parallelisierung des Verfahrens erfolgte unter Verwendung der C/C++ Bibliothek Message Passing Interface (MPI). Dieses führt in folgender Weise zu einer bezüglich Speicher und Rechenaufwand optimierten Verbesserung des herkömmlichen Level-Set-Algorithmus in NaSt3DGPF: An Stelle der Reinitialisierung nach dem Transportschritt wird die Level-Set-Funktion auf die feinen grid patches übertragen. Essentielles Merk-

mal, welches zu den hervorragenden Verbesserungen der Massenerhaltung führt, ist die Durchführung der Level-Set-Reinitialisierung auf den feinen grid patches. Dabei kann die Feingittermaschenweite h signifikant kleiner gewählt werden als die Grobgittermaschenweite H . Daraus resultiert, dass sich die Nullkontur maximal um die kurze Distanz h – nicht mehr um die große Distanz H – künstlich verschiebt. Die nun erzeugte Level-Set-Funktion mit Abstandseigenschaft wird jetzt auf das globale grobe Gitter übertragen und steht dem Strömungslöser wieder zur Verfügung. Zur ausführlichen Analyse der im Rahmen dieser Arbeit entwickelten adaptiven Level-Set-Reinitialisierung werden neue Testverfahren konstruiert und bekannte Testfälle herangezogen. Die Resultate dieser Tests zeigen, dass das neue adaptive Verfahren dem herkömmlichen Verfahren mit einem globalen feinen Gitter bezüglich Massenverlust durch numerische Diffusion in Relation zu Speicheraufwand und Laufzeit weitaus überlegen ist.

1.4 Zu dieser Arbeit

Diese Arbeit hat folgenden Aufbau:

Kapitel 2 beginnt mit einer Übersicht über verschiedene Gittertypen unter dem Gesichtspunkt der Approximation von beliebigen Geometrien und Oberflächen. Es erläutert ferner ihre Konstruktion, Handhabung und Eignung für verschiedene Probleme und Diskretisierungen. Zusätzlich wird eine Methode zur impliziten Einbettung von Gebietsrändern – die Ghost-Cell-Immersed-Boundary-Methode – vorgestellt. Es folgt eine Motivation für adaptive Gitter. Den Schluss dieses Grundlagenkapitels bildet die Darstellung der Level-Set-Methode.

Kapitel 3 befasst sich mit adaptiven Diskretisierungen. Unter dem Aspekt der Erweiterung der Level-Set-Methode von NaSt3DGPF geht es verstärkt auf strukturierte und kartesische Gitter ein. So beschreibt das Kapitel Gitter mit Baumstruktur – speziell octrees – und zeigt auf, wie Ränder von Geometrien und andere beliebige Oberflächen durch octrees adaptiert werden können. Eine sehr effiziente Implementierung von octrees – den sogenannten fully threaded trees – wird vorgestellt und ein Interpolationsschema zur Berechnung von Gradienten einer Funktion auf octrees angegeben. Die Vorstellung eines Mehrgitterverfahrens auf composite grids liefert eine weitere elegante Diskretisierung von Gebieten mit beliebigen Rändern. Der letzte Abschnitt zeigt die Anwendung der Mortar-Mimetische-Finite-Differenzen-Methode auf octrees und anderen kartesischen Gittern mit großen Irregularitäten beziehungsweise zahlreichen hängenden Knoten.

Kapitel 4 motiviert die Entwicklung einer adaptiven Level-Set-Methode auf kartesischen Gittern und beginnt dabei mit der ausführlichen Darstellung des Problems der numerischen Diffusion anhand von Beispielen aus der Zweiphasenströmungsmechanik. Nach der kurzen Vorstellung klassischer Methoden, die die Robustheit der Level-Set-Reinitialisierung verbessern, folgt eine detaillierte Beschreibung aller Komponenten des neuen Verfahrens.

Kapitel 5 beinhaltet alle Tests und Analysen der adaptiven Level-Set-Reinitialisierung. Nach ausführlichen Untersuchungen einzelner Komponenten folgt die Analyse des neuen Verfahrens an Zweiphasenströmungssimulationen, wobei auch Aspekte wie Speicher und Rechenaufwand untersucht werden.

Kapitel 6 schließt diese Arbeit mit einer kurzen Zusammenfassung und gibt einen Ausblick auf zukünftige Entwicklungen adaptiver Level-Set-Techniken.

1.5 Dank

An dieser Stelle möchte ich mich bei allen bedanken, die zum Entstehen dieser Arbeit beigetragen haben.

Mein Dank gilt besonders Herrn Prof. Dr. Michael Griebel für das sehr interessante Thema, seine hilfreichen Anregungen und die ausgezeichneten Arbeitsbedingungen am Institut.

Herrn Prof. Dr. Sören Bartels danke ich für die Übernahme des Zweitgutachtens.

Roberto Croce hat die Betreuung meiner Arbeit übernommen. Seine große Diskussionsbereitschaft, zahlreichen Ideen und hilfreichen Anregungen waren mir eine große Unterstützung. Herzlichen Dank dafür.

Des weiteren möchte ich mich bei Susanne Claus, Peter Hahnen, Tim Osadnik und Allan Zulficar für die gemeinsame Zeit im Büro bedanken.

Besonderer Dank gilt meiner Frau und meinen Kindern, meinen Eltern und Schwiegereltern für ihre große Unterstützung, ihr Vertrauen und ihre Geduld.

Kapitel 2

Grundlagen

Im Rahmen dieser Arbeit soll ein robusteres und stabileres Level-Set-Verfahren für Zweiphasenströmungssimulationen als Erweiterung des Programms NaSt3DGPF erarbeitet werden. Dazu ist es zu allererst notwendig, einen Überblick über verschiedene Verfahren und Gittertypen zu gewinnen, um darauf aufbauend eine adäquate Methode zu entwickeln. In diesem Grundlagenkapitel werden unterschiedliche Gittertypen unter dem Aspekt ihrer Erzeugung und der auf ihnen einsetzbaren Methoden und Algorithmen vorgestellt. Es soll ferner erläutert werden, inwieweit verschiedene Gitter komplexere Gebietsränder und Oberflächen approximieren können. Der letzte Abschnitt in diesem Kapitel gewährt einen Einblick in Level-Set-Methoden, der zum Verständnis der in dieser Arbeit entwickelten Methode beiträgt.

2.1 Diskretisierungen und Gittertypen

Die Simulation von realen Vorgängen in der Natur und technischen Abläufen in Industrie und Wirtschaft mithilfe von numerischen Mitteln erfordert immer wieder dieselben Vorgehensweisen, um analysierbare und auf das Ursprungsproblem rückführbare Ergebnisse zu erhalten.

Modellierung

Zunächst muss unter Berücksichtigung der wesentlichen und unwesentlichen Eigenschaften des Vorbilds aus diesem ein Modell entwickelt werden. Dieses Modell kann nun mit den Mitteln der Mathematik genau in Form von partiellen Differentialgleichungen zusammen mit ihren Randbedingungen beschrieben werden.

Diskretisieren der Gleichung

Damit die Numerik ihren Beitrag zum Lösen der neu gewonnenen Gleichungen leisten kann, müssen diese zunächst diskretisiert werden. Das heißt, dass man für jede Funktion und jeden Differentialoperator einen diskreten Repräsentanten finden muss. Unter den unzähligen Möglichkeiten, diesen Schritt zu vollziehen, sollen an dieser Stelle zwei wesentliche Methoden erwähnt werden.

Bei der einen Methode wird der diskrete Repräsentant einer Funktion dadurch gewonnen, dass die kontinuierliche Funktion an diskreten Punkten eines numerischen Gitters ausgewertet wird. Die zugehörigen Differentialoperatoren gehen bei dieser Art der Diskretisierung in sogenannte Differenzenoperatoren über, nachdem die Grenzwertbildung der Differenzenquotienten aufgehoben und auf endlich kleine - finite - Abstände, nämlich die der Gitterpunkte, reduziert wurde. Diese Methode trägt daher den Namen Finite-Differenzen-Methode.

Die andere Methode reduziert die Funktionen der Differentialgleichung nicht auf die Punkte eines Gitters, sondern stellt sie als Linearkombination von Basisfunktionen eines diskreten Unterraumes dar. Deren Träger werden dabei von den Zellen des numerischen Gitters definiert und finite Elemente genannt. Diese Methode nennt man deshalb auch Finite-Elemente-Methode. Für die Formulierung der diskreten Gleichung bedient sich die Finite-Elemente-Methode der integralen Darstellung der Differentialgleichung, das heißt, die Integrale werden über die diskreten finiten Elemente definiert und können so analytisch ausgewertet werden.

Diskretisieren des Gebiets

Mit der Diskretisierung der Gleichung stark verknüpft ist die Diskretisierung des Gebiets, auf dem die Lösung der Differentialgleichung gesucht wird. Hierbei wird das vom Modell vorgegebene Gebiet durch eine endliche Menge von Punkten oder finiten Elementen ersetzt, wodurch ein numerisches Gitter entsteht. Die Wahl eines geeigneten Gitters hängt von folgenden Aspekten ab:

- Geometrie des Gebiets
- Randbedingungen
- verwendetes numerisches Verfahren
- Zeitaufwand der Gittergenerierung
- Zeitaufwand für Gitteroperationen
- Speicheraufwand des Gitters

Die nächsten Abschnitte enthalten einen Überblick über die verschiedenen Gittertypen. Eine umfassende und vertiefende Übersicht zum Thema Gitter und Gittergenerierung findet sich in [31].

2.1.1 Strukturierte Gitter

Unter der Vielzahl von verschiedenen Gittertypen zählen die strukturierten zu den einfachsten Formen. Unter ihnen gibt es zwei wesentliche Ausprägungen: kartesische und

kurvilineare Gitter. Beiden gemein ist die Eigenschaft, dass sie ungeachtet ihrer physikalischen Gestalt stets auf ein logisches, rechteckiges Gitter mit einfacher Indizierung in sogenannter i-j-k-Struktur – wie in Abbildung 2.1a zu sehen – abgebildet werden können.

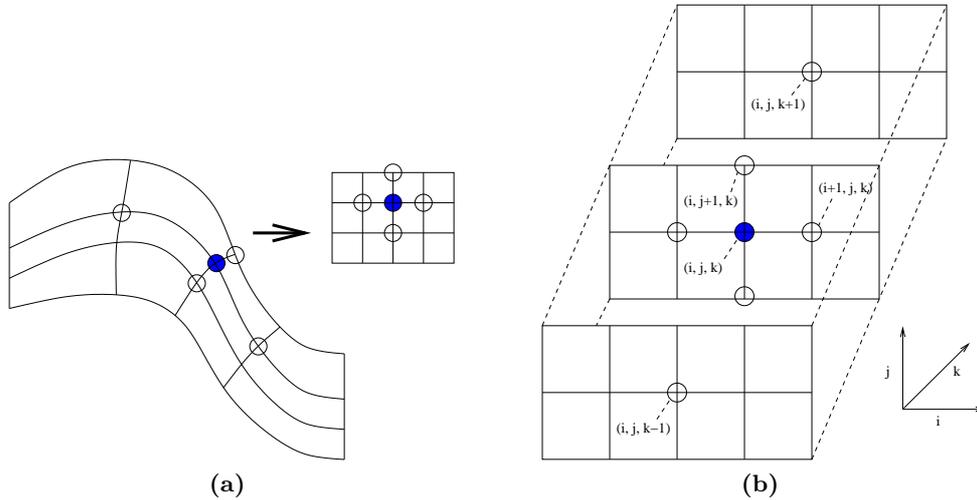


Abbildung 2.1: (a) Ein kurvilineares, strukturiertes Gitter mit seiner logischen, rechteckigen Struktur. (b) Ein isotropes, strukturiertes Gitter mit i-j-k-Indizierung.

Durch eine derartige Struktur lassen sich auch die Zuordnungen zwischen Punkten des Gitters und Punkten des physikalischen Gebiets leicht nachvollziehen. Ebenso einfach ist auch das Auffinden von Nachbarpunkten eines Punktes durch simples In- oder Dekrementieren des jeweiligen Indexes (siehe auch Abbildung 2.1b). Aufgrund dieser Eigenschaften eignen sich strukturierte Gitter hervorragend für Anwendungen mit Finite-Differenzen-Methoden.

Kartesische Gitter

In einem kartesischen Gitter werden alle Punkte auf Achsen angeordnet, die entweder parallel oder orthogonal zueinander verlaufen. Sie lassen sich daher auch als Tensor ihrer einzelnen Koordinatenachsen äußerst platzsparend speichern, wie die Abbildung 2.2 verdeutlicht. Sind die Abstände der Punkte auf den Achsen der jeweiligen Raumrichtungen, wie in Abbildung 2.3a immer gleich, so spricht man von isotropen Gittern; ansonsten wie in Abbildung 2.3b von lokal anisotropen Gittern. In beiden Fällen lassen sich durch die sehr regelmäßige Struktur und die Orthogonalität der Gitterachsen numerische Ausdrücke, die nativ in kartesischen Koordinaten vorliegen - wie etwa finite Differenzen - auf dem Gitter besonders leicht formulieren.

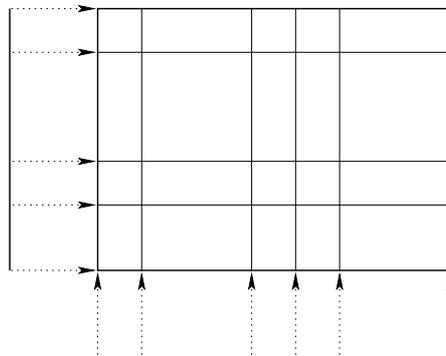


Abbildung 2.2: Ein kartesisches Gitter als Tensor seiner Koordinatenachsen

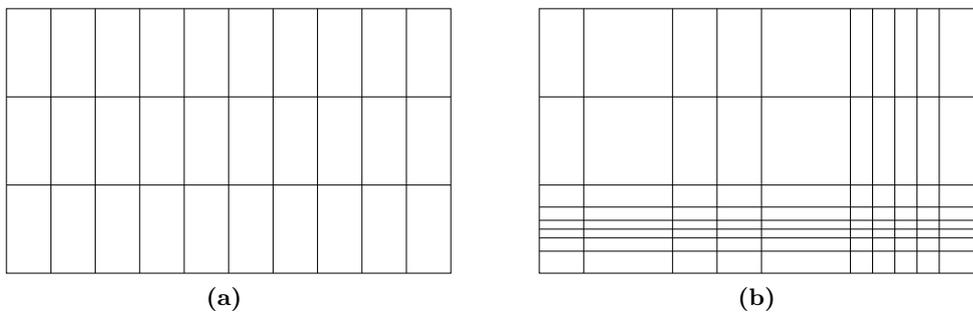


Abbildung 2.3: (a) Ein lokal isotropes Gitter. (b) Ein lokal anisotropes Gitter.

Kurvilineare Gitter

Während kartesische Gitter lediglich rechteckige¹ Gebiete abdecken, sind kurvilineare Gitter in der Lage, fast beliebige Gebiete, welche dem Rechteck topologisch sehr ähnlich sind, exakt abzudecken. Man kann sich dies als Vorgang vorstellen, bei dem ein Rechteck verbogen und verzerrt wird, bis es die gewünschte Form annimmt. Kurvilineare Gitter unterscheiden sich logisch nicht von kartesischen Gittern, weshalb ein numerisches Computerprogramm, welches eine Gleichung auf einem kartesischen Gitter löst, ebenso auch auf kurvilinearen Gittern operieren kann. Es muss jedoch die Differentialgleichung von kartesischen Koordinaten auf das entsprechende Koordinatensystem des kurvilinearen Gitters umformuliert werden.

Beispiel 2.1: Der Laplaceoperator auf dem kartesischen Gitter

$$\Delta f = f_{\xi\xi} + f_{\eta\eta} , \text{ mit } \xi = x \text{ und } \eta = y$$

¹In zwei Dimensionen. Entsprechend Quader und Hyperwürfel in drei und beliebigen Dimensionen.

wird auf einem zylindrischen Gitter zu

$$\Delta f = \frac{f_{\xi\xi}}{\eta^2} + f_{\eta\eta} + \frac{f_{\eta}}{\eta}, \text{ mit } \xi = \theta \text{ und } \eta = r.$$

Erzeugung strukturierter Gitter

Ist das vom Gitter abzudeckende Gebiet analytisch definiert, wie zum Beispiel durch einen Zylinder oder durch lokale Koordinatentransformationen beschrieben, so lässt sich aus diesen Informationen auf einfache Art und Weise durch Umparametrisieren ein logisch orthogonales Gitter auf ein kurvilineares Gitter transformieren. Ein Beispiel für die Erzeugung eines zylindrischen Gitters illustriert die Abbildung 2.4.

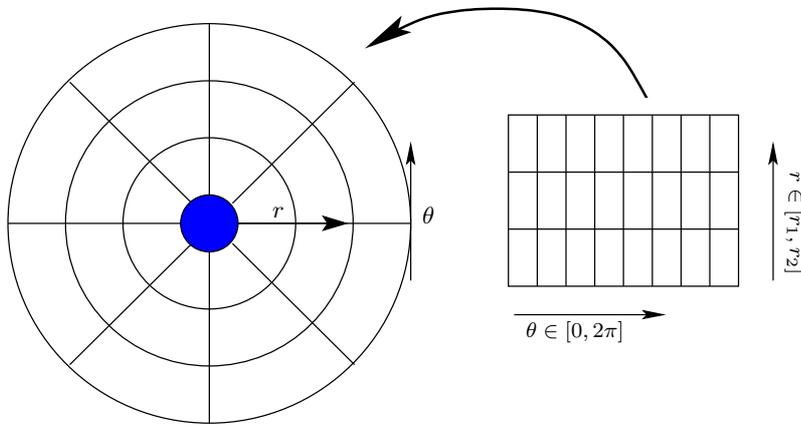


Abbildung 2.4: Ein zylindrisches Gitter entsteht aus einem kartesischen Gitter

Hat man es mit geometrisch sehr komplexen Gebietsrändern zu tun, so ist man besonders daran interessiert, dass das Gitter diesem Gebietsrand möglichst gut angepasst ist. Eine einfache und bezüglich Speicher und Zeitaufwand kostengünstige Methode, welche sich der kartesischen Gitter bedient, macht es mittels der Markierung von Zellen möglich, die Geometrie mit mäßiger Genauigkeit in das kartesische Gitter einzubetten: Man nehme das durch ein Rechteck umschriebene Gebiet und überdecke dieses mit einem kartesischen Gitter der gewünschten Auflösung. Nun markiert man alle externen Zellen (Voxel), die außerhalb des Gebiets liegen, so dass die numerischen Ausdrücke der Gleichung nur noch auf diesen Gebietszellen definiert sind und ausgewertet werden. Diese Vorgehensweise und ein fertiges dreidimensionales Gitter werden in den Abbildungen 2.5 und 2.6 gezeigt. Ein wesentlicher Nachteil dieser Methode gegenüber dem Vorteil der geringen Kosten ist jedoch die Ungenauigkeit der groben Approximierung des Gebietsrandes durch lineare, achsenparallele Strecken beziehungsweise Flächen.

Die Tatsache, dass sich komplexe Geometrien nur grob in kartesische Gitter einbetten lassen, führt zu der Überlegung, kurvilineare Gitter zu verwenden, da deren Strukturen

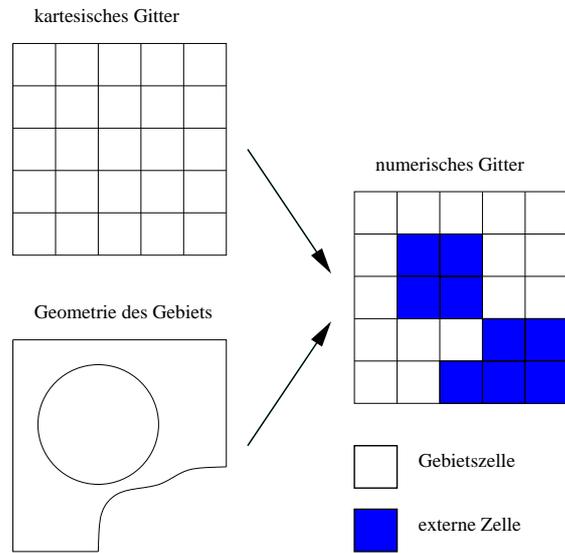


Abbildung 2.5: Ein kartesisches Gitter mit eingebetteter Geometrie

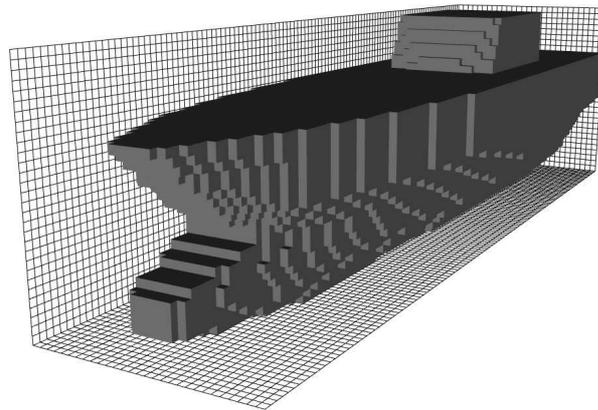


Abbildung 2.6: Markierte Zellen (Voxel) betten die Geometrie eines Schiffs in ein kartesisches Gitter ein.

nicht nur linear und auf Achsenparallelität beschränkt, sondern grundsätzlich in ihrer räumlichen Orientierung frei sind und krummlinig sein dürfen. Eine interessante Technik zur Erzeugung von Gittern, die sich der Gebietsgeometrie anpassen - sogenannter boundary fitted grids - bedient sich der Level-Set-Methode. Eine detailliertere Beschreibung

der Level-Set-Methode gibt das Kapitel 2.4 auf Seite 25. Diese Technik verfolgt das Ziel, strukturierte, an Körper mit fast beliebiger Geometrie angepasste Gitter zu erzeugen, wobei diese Körper durchaus spitze Ecken oder eine Oberfläche mit stark variierender Krümmung besitzen dürfen. Für diesen Zweck macht die Technik von der Lösung einer Hamilton-Jacobi-Gleichung Gebrauch, durch die eine Front - die Grenzfläche zwischen Körper und Gebiet - Stück für Stück ausgebreitet wird und so zum Aufbau des Gitters beiträgt.

Veranschaulicht man sich nun diese Art der Gittergenerierung in zwei Dimensionen, indem man sich den Rand der Region, die nicht zum Gebiet gehört, als Kurve $\gamma(t = 0)$ in \mathbb{R}^2 vorstellt, so stimmt die mit der Zeit t voranschreitende Front $\gamma(t)$ also zum Zeitpunkt $t = 0$ mit dem Rand überein. Die Ausbreitung der Front $\gamma(t)$ mit $t > 0$ geschieht, indem man jeden Punkt x auf der Kurve $\gamma(0)$ mit der Geschwindigkeit $F(\kappa)$ in Abhängigkeit der Krümmung $\kappa(x)$ in Richtung der Normalen ν fortbewegt. Nun erhält man für äquidistante Zeitschritte eine Schar von Kurven

$$\gamma(i\Delta t), \text{ mit Zeitschritt } \Delta t \text{ und } i \in \mathbb{N},$$

die den Gebietsrand schalenartig umgeben: die randangepassten Schalen.

Die Punkte auf der Kurve unterliegen der Bewegungsgleichung $\nu \cdot x_t = F(\kappa)$. Betrachtet man nun die Geschwindigkeitsfunktion $F(\kappa) = 1 - \epsilon \kappa$ mit einer nichtnegativen Konstanten ϵ , so ergibt sich, siehe Sethian [46, 47], für die Krümmung die Entwicklungsgleichung

$$\kappa_t = \epsilon \kappa_{\alpha\alpha} + \epsilon \kappa^3 - \kappa^2. \quad (2.1)$$

Für $\epsilon = 0$ wird aus dieser Reaktions-Diffusions-Gleichung eine reine Reaktionsgleichung, deren Lösung schon in endlicher Zeit Singularitäten besitzen kann, wenn die Krümmung der Anfangskurve $\gamma(0)$ an irgendeiner Stelle negativ ist. Dies hat zur Folge, dass die sich ausbreitende Front ihre Glattheit verliert und spitze Ecken ausbildet, an denen die Normale ν nicht mehr definiert ist. Die Abbildung 2.7a verdeutlicht dieses Problem. Wählt man nun für die Konstante ϵ einen Wert > 0 , so sorgt der diffusive Anteil $\epsilon \kappa_{\alpha\alpha}$ in Gleichung 2.1 stets für die Glattheit der sich ausbreitenden Front (siehe auch Abbildung 2.7b). Ein vollständiges numerisches Gitter entsteht nun durch die Konstruktion von transversalen Linien. Dafür werden anfänglich N Knoten auf dem Rand des Gebietes platziert, um welches ein Gitter erzeugt werden soll. Diese Knoten werden wie in Abbildung 2.8 mit ihren N Positionen $X_i = (x_i, y_i) \in \mathbb{R}^2, i \in \{1 \dots N\}$ mit jedem Zeitschritt der Frontausbreitung mit der Geschwindigkeit $F(\kappa)$ in Richtung der äußeren Normalen ν transportiert. Dafür werden für jeden Zeitschritt die N gewöhnlichen Differentialgleichungen

$$\frac{dX_i}{dt} = F(\kappa) \nu, \text{ mit } \nu = \frac{\nabla \phi}{|\nabla \phi|} \text{ und } i \in \{1 \dots N\}$$

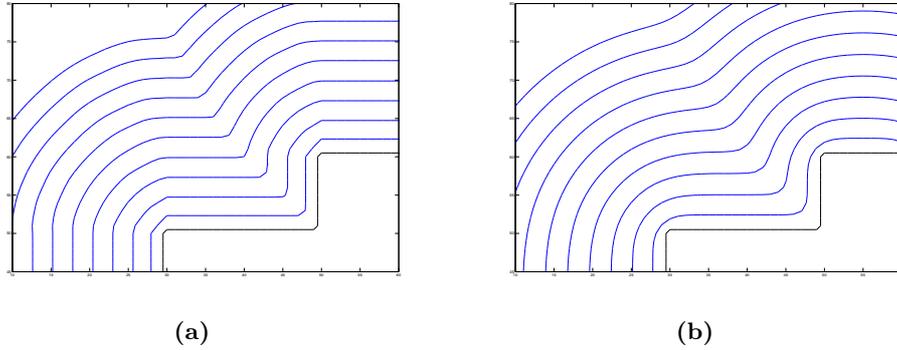


Abbildung 2.7: (a) Eine sich mit der konstanten Geschwindigkeit 1 ausbreitende Front mit singulären Krümmungen. (b) Eine sich mit der Geschwindigkeit $1 - 0.25 \kappa$ ausbreitende Front bleibt stets glatt.

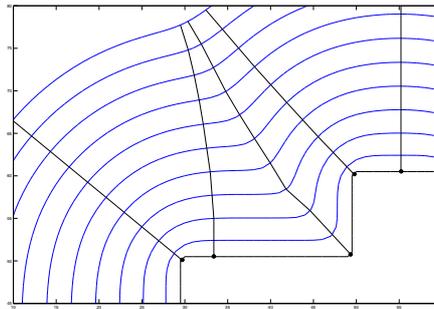


Abbildung 2.8: Erzeugung transversaler Linien durch den Transport von Punkten auf dem Rand in Normalenrichtung.

gelöst. Aufgrund dieses Verfahrens verlaufen die transversalen Linien immer orthogonal zu den randangepassten Schalen. Jedoch erzielt man mit diesem Verfahren im Allgemeinen keine zufriedenstellenden Ergebnisse, da sich zum Beispiel bei nichtkonvexem Gebietsrand $\gamma(0)$ die transversalen Linien beliebig nahe kommen können und so die Größe und auch das Seitenverhältnis der Gitterzellen entarten. Um numerisch brauchbare Gitter produzieren zu können, wird man daher zu weiteren Verfahren greifen müssen, die die Anordnung der Punkte eines Gitters in dieser Hinsicht nachträglich verbessern. In [47] werden diese Verfahren sowie weitere Aspekte dieser Methode beschrieben, Gitter zu erzeugen.

Eine mögliche Anwendung für kurvilineare Gitter wird in 3.3.1 auf Seite 44 beschrieben.

2.1.2 Unstrukturierte Gitter

Ein wesentliches Unterscheidungsmerkmal zwischen strukturierten und unstrukturierten Gittern ist die Tatsache, dass unstrukturierte Gitter keine Nachbarschaftbeziehungen ihrer Punkte als intrinsische Eigenschaft mit sich führen (vergleiche Abschnitt 2.1.1 auf Seite 10). Vielmehr sind spezielle Datenstrukturen zur Speicherung von Nachbarpunkten, Nachbarelementen und begrenzender Kanten unerlässlich.

Als Beispiel für unstrukturierte Gitter sollen hier zwei besondere Klassen von Gittern vorgestellt werden. Zum einen handelt es sich um Triangulierungen, die häufig im Zusammenhang mit Finite-Elemente-Methoden und Gebieten mit komplexerer Geometrie verwendet werden, sowie um quadrees und octrees.

Triangulierungen

Durch sogenannte Triangulierungen wird das Gebiet in Dreiecke beziehungsweise Tetraeder zerlegt. Sie entstehen durch eine anfängliche Wahl der Gitterpunkte und dem Delaunay-Voronoi-Verfahren, welches die Zuordnung von Punkten zu Gitterzellen vornimmt. Ein solches Dreiecksnetz für zwei Raumdimensionen ist in Abbildung 2.9 zu

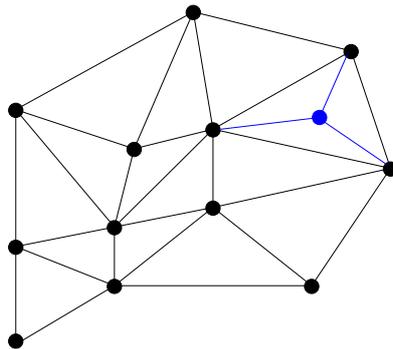


Abbildung 2.9: Ein unstrukturiertes Dreiecksgitter (Triangulierung). In blau: Ein weiterer Punkt mit Kanten wird hinzugefügt

sehen. Ihr großer Vorteil gegenüber strukturierten Gittern ist ihre geometrische Flexibilität, das heißt, durch die geeignete Wahl ihrer Gitterpunkte auf dem durchaus komplexen Gebietsrand entstehen Triangulierungen, die der Randgeometrie beliebig genau angepasst sein können. Ferner besteht die Möglichkeit, mit wenig Aufwand weitere Punkte auf dem Gebietsrand oder im Gebiet hinzuzufügen, um somit die Anpassung an den Rand oder die Auflösung des Gitters lokal zu verbessern (siehe Abbildung 2.9), womit sich Triangulierungen besonders gut mit adaptiven Algorithmen zusammen verwenden lassen.

Quadrees und octrees

Im Gegensatz zu Triangulierungen besitzen die Gitter der quadrees (in zwei Dimensionen) und octrees (in drei Dimensionen) eine stärkere innere Struktur, bedingt durch die Selbstähnlichkeit und die Hierarchie ihrer Gitterzellen. Wesentliches Merkmal eines octrees ist die Baumstruktur, die die Beziehungen der Gitterzellen zueinander definiert: Es gibt genau eine Zelle mit der größten Maschenweite; sie wird durch die Wurzel des Baumes repräsentiert. Diese Zelle kann - wie grundsätzlich jede Zelle eines octrees - in vier (bei quadrees) sowie in acht (bei octrees) Tochterzellen gleichmäßig unterteilt sein. Diese Tochterzellen finden sich in der Baumstruktur als Nachkommen wieder. Eine nicht weiter unterteilte Zelle nennt man auch Blatt des Baumes oder Blattzelle. Ferner wird jeder Zelle ein Level zugeordnet, welches Auskunft über den Grad der Verfeinerung gibt. Ausgehend von der Wurzel mit dem Level 0 gilt: Hat eine Zelle c das Level l , dann haben die Tochterzellen das Level $l + 1$. Die Abbildung 2.10 verdeutlicht dieses Prinzip.

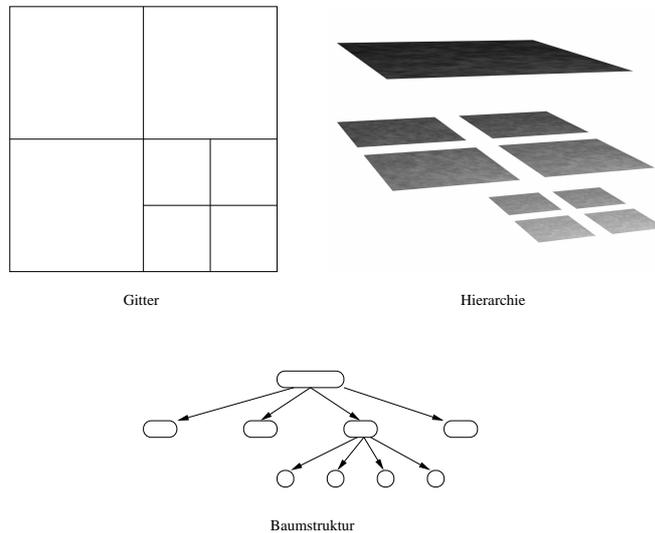


Abbildung 2.10: Das Gitter eines quadrees, seine Hierarchie und die Baumstruktur

2.2 Ghost-Cell-Immersed-Boundary-Methode

In den vorangegangenen Abschnitten wurde gezeigt, welche Möglichkeiten für die Wahl der Gitter einer Diskretisierung zur Verfügung stehen. Dabei wurde deutlich, dass kartesische Gitter besonders leicht zu erzeugen und der numerische Algorithmus, der auf dem Gitter operiert, vergleichsweise einfach zu implementieren sind. Zur Einbettung der Geometrie ist man dabei allerdings auf die grobe und nicht sehr flächentreue Voxelisierung (siehe Abbildung 2.6 auf Seite 14) angewiesen. Möchte man das Gitter der Geometrie

anpassen, jedoch nicht auf strukturierte Gitter verzichten, so greift man auf boundary fitted grids zurück und nimmt die höheren Kosten der Gittergenerierung kurvilinearere Gitter und Transformationen der Gleichungen (siehe Abschnitt 2.1.1 auf Seite 12) in Kauf. Eine effiziente Methode, die die geringen Kosten von kartesischen Gittern und die gute Approximation der Geometrie vereint, beschreiben Tseng und Ferziger in [55] als ghost cell immersed boundary method (GCIBM). Diese Methode definiert den Rand beliebiger Geometrien in indirekter Weise, indem sie entsprechende Zellen des Gitters als Ghostzellen definiert und somit die benötigten Randbedingungen der eingebetteten Geometrie herstellt.

Die Immersed-Boundary-Methode ist von Peskin [38] eingeführt worden, ihre Anwendbarkeit auf Probleme mit beliebiger Geometrie unter signifikant geringerem Rechenaufwand als andere Methoden und ohne Einbuße der numerischen Genauigkeit demonstrieren Fadlun, Verzicco, Orlandi, Mohd-Yusof und Haworth in [23, 42]. Besondere Vorteile der Methode sind

- geringer Aufwand für Speicher und CPU,
- einfache Gittererzeugung; besonders im Vergleich zu kurvilinearen, boundary fitted grids und Triangulierungen,
- entbehrliche Manipulation einmal erzeugter Gitter selbst bei zeitabhängiger Geometrie,
- die Möglichkeit, Geometrien mit fast beliebiger Form zu behandeln.

Zur Anwendung der GCIBM müssen zunächst die entsprechenden Zellen des Gitters als Ghostzellen identifiziert werden. Dazu definiert man alle Zellenkanten, die vom Rand der Geometrie geschnitten werden, als immersed boundary, was das Gebiet nun in zwei Domänen aufteilt: die physische und die Ghostzellen-Domäne. Die Ghostzellen selbst sind nur diejenigen Zellen der Ghostzellen-Domäne, die an Zellen der physischen Domäne angrenzen. Abbildung 2.11 verdeutlicht das Prinzip der Domänen. Das Ziel der GCIBM ist es nun, die Variablen in den Ghostzellen so zu setzen, dass sie implizit einen Gebietsrand mit der geforderten Randbedingung definieren.

2.2.1 Dirichlet-Bedingungen

In den Ghostzellen werden die Variablen ϕ in Form von Polynomen ausgedrückt, um dann an den Ghostzellenpunkten ausgewertet zu werden. Obwohl höhere Polynomgrade eine höhere Genauigkeit des Verfahrens versprechen, sieht man von der Verwendung von Polynomen höheren Grades ab, da die Freiheitsgrade der Ghostzellen durch Extrapolation gewonnen werden und Interpolationspolynome höheren Grades außerhalb ihrer Stützstellen zu starken Oszillationen neigen. In der Regel kommen lineare und quadratische Polynome zum Einsatz, was die Genauigkeit zweiter Ordnung des Schemas erhält.

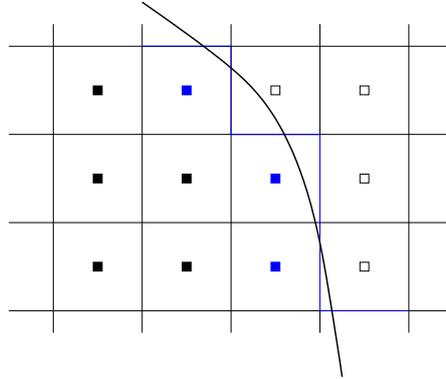


Abbildung 2.11: Schema einer GCI-MB: Die vom Rand einer Geometrie geschnittenen Kanten (blau) definieren die immersed boundary; sie ordnet jede Gitterzelle der physischen Domäne (\square) oder der Ghostzellen-Domäne (\blacksquare) mit ihren Ghostzellen (\blacksquare) zu.

Lineare Rekonstruktion

Ein einfacher Weg in zwei Dimensionen besteht darin, ein Dreieck aus den zwei nächsten benachbarten Punkten (\times) der Ghostzelle G und dem Punkt O auf dem Rand der Geometrie zu konstruieren, an dem die Randbedingung erfüllt sein soll (siehe auch Ab-

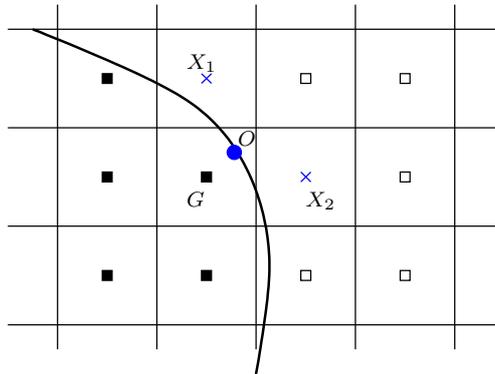


Abbildung 2.12: Die Variable der Ghostzelle G wird durch lineare Extrapolation mittels des Dreiecks $\Delta(X_1, X_2, O)$ gewonnen.

bildung 2.12). Dabei kann der Punkt O entweder als Mittelpunkt des Geometrierandes innerhalb der Ghostzelle gewählt werden oder so, dass die Gerade \overline{GO} orthogonal zum Rand steht. Eine lineare Interpolation $\phi = a_0 + a_1x + a_2y$ in zwei Dimensionen ermöglicht nun die Berechnung des Ghostzellenwertes als gewichtete Kombination seiner benachbarten Werte in den Knoten X_1, X_2 und O . Der Koeffizientenvektor a wird nun

aus

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = B^{-1} \begin{pmatrix} \phi(O) \\ \phi(X_1) \\ \phi(X_2) \end{pmatrix} \quad (2.2)$$

gewonnen mit

$$B = \begin{bmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{bmatrix}. \quad (2.3)$$

Wesentliche Nachteile ergeben sich jedoch, wenn der Punkt O sehr dicht bei den Punkten X_1 oder X_2 liegt, was sich trotz algebraischer Korrektheit in großen, negativen Koeffizienten zeigt und letztendlich das Verfahren numerisch instabil werden lässt. Abhilfe schaffen hier zwei Ansätze: In [45] ziehen Majumdar, Iaccarino und Durbin an Stelle des eigentlichen Ghostzellenpunktes G sein Bild I als Interpolationspunkt heran, wobei I der an dem Geometrierand gespiegelte Punkt G ist. Der Wert des Ghostzellenpunktes ergibt sich dann aus $\phi_G = 2\phi_0 - \phi_I$.

Ein anderer Ansatz eignet sich, falls einer der nächsten Punkte G' der physikalischen Domäne so dicht an O liegt, dass $\overline{OG'} < 0.1\Delta x$ mit der Zellgröße Δx , jedoch weit von den Ghostzellenpunkten entfernt ist. Die Lösung dieses Ansatzes besteht nun darin, den Punkt O zum nächstgelegenen Punkt G' zu verschieben. Dazu haben Gibou, Fedkiw, Cheng und Kang [25] gezeigt, dass mit dieser Methode die Poissongleichung auf unregelmäßigen Gebieten noch mit einer Genauigkeit zweiter Ordnung gelöst werden kann.

Quadratische Rekonstruktion

Die meisten Finite-Volumen-Verfahren zweiter Ordnung nehmen eine quadratische Variation ihrer Variablen in Wandnähe an. In zwei Dimensionen wird daher die Variable ϕ als

$$\phi = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2$$

ausgedrückt und ihre sechs Koeffizienten analog zum linearen Fall mittels der fünf nächsten Punkte $X_{1,\dots,5}$ der physikalischen Domäne und dem Punkt O auf dem Geometrierand berechnet. Die Matrix B aus Gleichung 2.2 wird nun durch die 6×6 -Matrix

$$B = \begin{bmatrix} 1 & x_0 & y_0 & x_0^2 & x_0y_0 & y_0^2 \\ 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ & & & \dots & & \\ 1 & x_5 & y_5 & x_5^2 & x_5y_5 & y_5^2 \end{bmatrix} \quad (2.4)$$

ersetzt. Wiederum werden die Ghostzellenknoten entweder direkt ausgewertet oder über ihr Spiegelbild I berechnet.

2.2.2 Neumann-Bedingungen

Die GCIBM lässt sich ebenfalls mit Neumann-Randbedingungen implementieren. Exemplarisch gelten soll als Randbedingung

$$\left. \frac{\partial \phi}{\partial n} \right|_{\Omega} = 0 \quad (2.5)$$

mit der Ableitung von ϕ in Normalenrichtung n

$$\begin{aligned} \frac{\partial \phi}{\partial n} &= \frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y \\ n &= \begin{pmatrix} n_x \\ n_y \end{pmatrix}. \end{aligned}$$

Sei θ der Winkel der Normalen des Geometrierandes im Punkt O gegen die x -Achse beziehungsweise

$$n = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix},$$

dann kann der Koeffizientenvektor a wieder aus einem Gleichungssystem

$$\begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ a_m \end{bmatrix} = B^{-1} \begin{bmatrix} \frac{\partial \phi}{\partial n} \\ \phi_1 \\ \cdot \\ \phi_m \end{bmatrix} \quad (2.6)$$

berechnet werden.

Lineare Rekonstruktion

Im linearen Fall lautet die Matrix B

$$B = \begin{bmatrix} 0 & -\sin(\theta) & \cos(\theta) \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{bmatrix}. \quad (2.7)$$

Quadratische Rekonstruktion

Im quadratischen Fall lautet die Matrix B

$$B = \begin{bmatrix} 0 & -\sin(\theta) & \cos(\theta) & -2\sin(\theta)x_0 & \cos(\theta)x_0 - \sin(\theta)y_0 & 2\cos(\theta)y_0 \\ 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \end{bmatrix}. \quad (2.8)$$

2.2.3 Ergebnis

Tseng und Ferziger verifizieren in [55] die Genauigkeit und Anwendbarkeit der GCIBM anhand von Strömungsproblemen sowohl in zwei als auch in drei Dimensionen und erzielen damit eine globale Genauigkeit zweiter Ordnung. Darüber hinaus zeigen sie, dass die Implementierung der GCIBM in bereits bestehende Simulationsprogramme, die auf strukturierten, orthogonalen Gittern basieren, mit nur geringem Aufwand durchgeführt werden kann und der Rechenaufwand dieser Methode im Vergleich zu anderen Methoden mit unstrukturierten boundary fitted grids sehr gering ausfällt.

2.3 Anforderungen an das Gitter

Die Genauigkeit von numerischen Simulationen hängt zum einen von der Ordnung des verwendeten Verfahrens ab. Eine andere wichtige Rolle aber spielt die Diskretisierung mit ihrem zugrundeliegenden Gitter. Inwieweit eine Diskretisierung die physikalische Wirklichkeit wiedergibt, übermitteln der Diskretisierungsfehler. Die Konstruktion eines Gitters richtet sich daher nach den geometrischen Eigenschaften des zu simulierenden physikalischen Vorbilds und berücksichtigt allgemeine Kenntnisse über die Lösungen. So wird das Gitter in Teilen des Gebiets, in denen die Lösungen hohe Gradienten besitzen und stark variieren, eine höhere Punktdichte besitzen als in Regionen, in denen die Lösungen sehr glatt sind.

Diese Arbeit widmet sich ganz besonders der Behandlung von freien Oberflächen in Zweiphasenströmungssimulationen. Die wesentlichen Probleme und Herausforderungen stellen sich dann, wenn die freie Oberfläche starke Krümmungen und feine Strukturen und Details aufweist. In Bezug auf das Gitter, welches diese besonderen Eigenschaften adäquat diskretisieren soll, kann das Problem verallgemeinert werden, so dass nicht nur von freien Oberflächen, sondern umfassender von komplexen Geometrien gesprochen werden kann. Nun soll im Folgenden erläutert werden, was komplexe Geometrien sind und welche Anforderungen an ein numerisches Gitter gestellt werden, das für numerische Zwecke mit komplexen Geometrien dienen soll.

2.3.1 Komplexe Geometrien

Ein Objekt besitzt eine komplexe Geometrie, wenn es nicht zu den primitiven Grundobjekten wie zum Beispiel Würfel, Quader, Kugel, Kegel und Tetraeder gehört sowie nicht aus nur wenigen dieser Objekte zusammengesetzt werden kann. Eine weitere Eigenschaft, die komplexe Geometrien aufweisen, sind große Skalenunterschiede der vorhandenen Strukturen. Das bedeutet, dass das Objekt in den äußeren Abmessungen groß dimensioniert ist, aber dennoch kleine Strukturen mit hoher Detaildichte besitzt. Ferner ist die Topologie eines Objekts von großer Bedeutung, so dass ein Objekt aus durchaus mehreren oder gar vielen verschiedenen Zusammenhangskomponenten bestehen und

seine Oberfläche starke Variationen in ihrer Krümmung aufweisen darf (siehe Abbildungen 2.13 und 2.14).

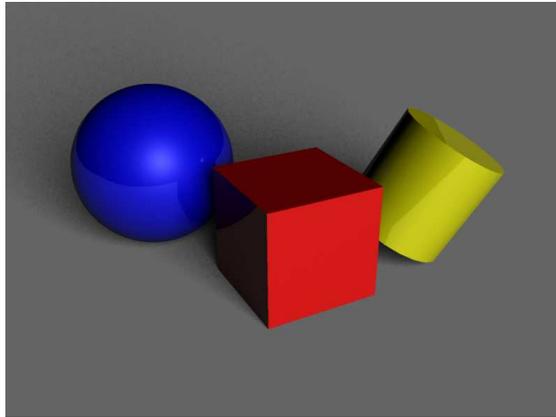


Abbildung 2.13: Geometrisch primitive Objekte: Sphäre, Würfel, Zylinder



Abbildung 2.14: Der Bonner Bahnhof: ein Beispiel für ein Objekt mit komplexer Geometrie. Er besitzt große äußere Abmessungen und viele kleine Details.

2.3.2 Lokale Verfeinerung

Für ein Gitter, das Strukturen mit komplexen Geometrien auflösen soll, muss dessen Auflösung also fein genug sein, um die Details nicht zu verlieren und die Topologie der Strukturen nicht zu verändern (siehe hierzu Abbildung 2.15). Da komplexe Geometrien nicht nur detailreich, sondern auch groß in ihren Abmessungen sein können, ist es ineffizient, auf dem gesamten Gitter, vor allem in detailarmen Regionen, eine hohe

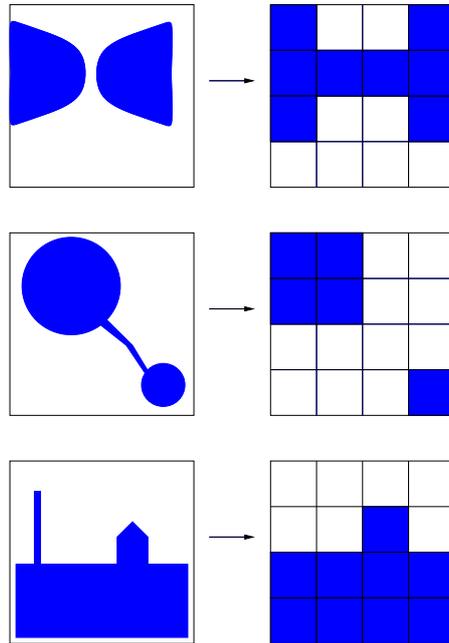


Abbildung 2.15: Bei zu geringer Auflösung wird die Topologie verfälscht, und Details gehen auf dem Gitter verloren (von oben nach unten): Aus zwei Zusammenhangskomponenten wird eine, aus einer Zusammenhangskomponente werden zwei, ein Detail verschwindet.

Auflösung zu wählen. Dieses Vorgehen würde nicht nur den Speicheraufwand, sondern auch die Laufzeit der auf dem Gitter operierenden, numerischen Algorithmen vergrößern.

Das Ziel ist es nun, mittels Adaptivität die Auflösung des Gitters lokal so fein zu wählen, dass mit ihm alle wichtigen Details und Eigenschaften der Geometrien und der Lösungen erfasst und ein stabiles Verhalten der numerischen Algorithmen gewährleistet werden, jedoch in anderen Regionen grob genug, dass die Kosten durch Speicher- und Laufzeitaufwand nicht unnötig groß sind (siehe auch Abbildung 2.16).

2.4 Level-Set-Methode

Die Level-Set-Methode ist seit ihrer Einführung im Jahr 1988 durch Osher und Sethian [48, 36] erfolgreich in vielen Gebieten eingesetzt worden. Sie stellt ein Universalwerkzeug dar, deren Anwendungsmöglichkeiten mannigfaltig sind: So wird sie beispielsweise in der Computergraphik, im Bereich computational geometry und robotic eingesetzt. Im medizinischen Umfeld benutzt man die Level-Set-Methode zum automatischen Segmentieren von Gebieten in digitalen Röntgenbildern oder von Volumendaten aus Computer- und Magnetresonanztomographen, so dass ein rechnergestütztes Auffinden von anatomo-

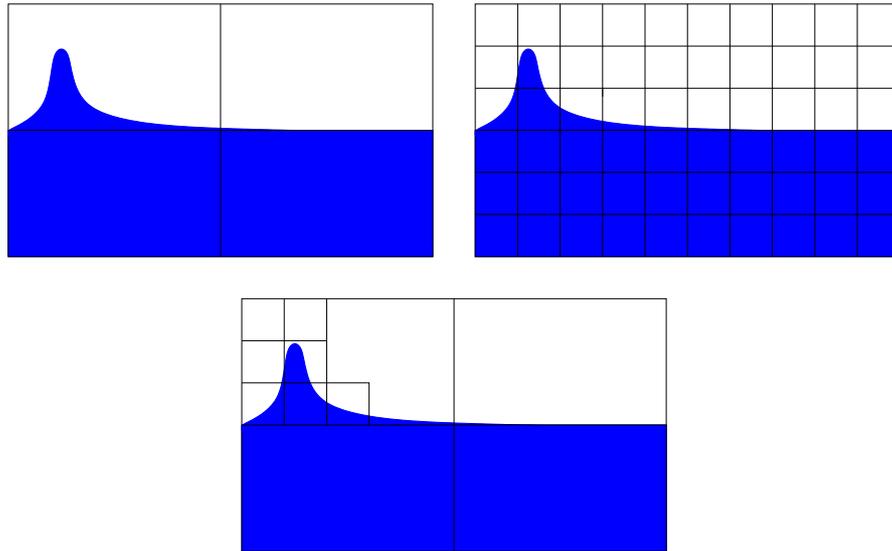


Abbildung 2.16: Ein Beispiel für ein adaptives Gitter: Das Gitter ist zu grob, um die Details der Geometrie aufzulösen (links). Das Gitter ist fein genug für die Details, besitzt aber überflüssige Punkte, in denen die Geometrie auch grob beschreibbar ist (rechts). Ein adaptives Gitter besitzt feine und grobe Gitterzellen (unten).

misch bedeutungsvollen Objekten wie Organen, Knochen, Weichteilen, Tumoren oder Ähnlichem möglich wird. Andere große Bereiche sind der Einsatz in der Optimierung von kürzesten Wegen oder statischen Tragwerken und in der Simulation von Wellen, Flammen und sich ausbreitenden Fronten.

2.4.1 Darstellung und Transport von Grenzflächen

Für die Simulation von Zweiphasenströmungen wird ein Hilfsmittel benötigt, welches die Zuordnung eines Fluids zu einer der beiden Phasen ermöglicht. Es gibt sogenannte Volume-Tracking-Methoden wie die häufig eingesetzte Volume-of-Fluid-Methode. Sie wurde 1979 von Hirt und Nichols [29] entwickelt und kommt in vielen Strömungslösern zur Zweiphasensimulation zum Einsatz. Ein anderes Paradigma der Phasentrennung machen sich dagegen die Surface-Tracking-Methoden zunutze. Sie benutzen Markerpartikel entlang der Oberfläche, um diese im Gebiet zu definieren. Sehr bekannt geworden ist die Surface-Marker-Methode von S. Chen, D.B. Johnson und P.E. Raad [16]. Ferner findet die in diesem Abschnitt näher beschriebene Level-Set-Methode zum Verfolgen der Grenzfläche, welche zu den sogenannten Front-Capturing-Methoden gehört, Verwendung in Zweiphasensimulationen. Einen ausführlichen Überblick über verschiedene Verfahren zum Verfolgen von Grenzflächen in Zweiphasenströmungen gibt Croce in Abschnitt 4.1

in [18]. Im Rahmen dieser Arbeit wird die Level-Set-Methode zum Gegenstand weiterer Untersuchungen und Entwicklungen.

2.4.2 Die Level-Set-Funktion

Die Grenzfläche zwischen den Phasen wird bei Level-Set-Methoden als Niveaumenge einer auf dem gesamten Gebiet definierten, Lipschitz-stetigen Funktion - der Level-Set-Funktion definiert:

Definition 2.1 [LEVEL-SET-FUNKTION]

Die Level-Set-Funktion ϕ ist auf dem gesamten Gebiet Ω definiert und ordnet zu jedem Zeitpunkt t einem Punkt $x \in \Omega$ ein Level beziehungsweise eine Kontur zu.

$$\begin{aligned} \Omega &\subset \mathbb{R}^n \\ \phi : \Omega \times \mathbb{R}^+ &\rightarrow \mathbb{R} \\ (x, t) &\mapsto \phi(x, t) \end{aligned} \tag{2.9}$$

Weitere Erläuterungen bringt der einfacheren Wortwahl halber der Fall $\Omega \subset \mathbb{R}^2$. Die Idee ist nun, die Grenzfläche als Kontur $\{\phi = 0\}$ zum Wert 0 zu verstehen; die sich zeitlich fortpflanzende Grenzfläche $\Gamma(t)$ wird also aufgefasst als implizit gegeben durch die Definition:

Definition 2.2 [GRENZFLÄCHE]

Die Grenzfläche Γ zum Zeitpunkt t sei

$$\Gamma(t) := \{x \mid \phi(x, t) = 0\}. \tag{2.10}$$

Insbesondere gilt also

$$\phi(\vec{x}(t), t) = 0, \text{ für alle } \vec{x}(t) \in \Gamma(t). \tag{2.11}$$

Nach dem Differenzieren unter Anwendung der Kettenregel erhält man

$$\phi_t + \nabla\phi(\vec{x}(t), t) \cdot \vec{x}'(t). \tag{2.12}$$

Die Eigenschaft 2.11 auf der Nullkontur der Level-Set-Funktion lässt bisher noch viel Spielraum für eine tatsächliche Definition der Funktion auf dem gesamten Gebiet Ω zu. Damit man eine möglichst glatte Funktion erhält, die zudem noch die Unterscheidung zwischen den beiden Seiten der Grenzfläche erlaubt, stellt man noch weitere Bedingungen:

Zusätzlich soll die Level-Set-Funktion die Eigenschaft besitzen, auf verschiedenen Seiten von Γ ein unterschiedliches Vorzeichen zu haben. In der Praxis wählt man für die Level-Set-Funktion eine vorzeichenbehaftete Abstandsfunktion, das heißt, der Wert der Funktion entspricht vom Betrag her dem kleinsten Abstand zur Grenzfläche, und das

Vorzeichen hängt davon ab, ob sich ein Punkt auf der einen oder der anderen Seite der Grenzfläche befindet. Die Level-Set-Funktion unterliegt daher der sogenannten Eikonalgleichung

$$|\nabla\phi| = 1, \quad (2.13)$$

was die Level-Set-Funktion zu einer Lipschitz-stetigen Funktion macht. Das Prinzip

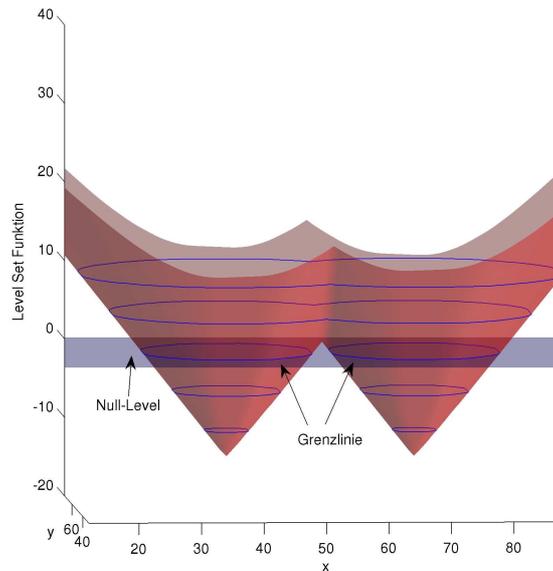


Abbildung 2.17: Eine Level-Set-Funktion in zwei Dimensionen: Auf der Level-Set-Funktion (rote Fläche) sind die Konturlinien (blau) zu den Werten (-10, -5, 0, 5, 10) eingezeichnet. Auf einer Konturlinie haben alle Punkte den gleichen Level-Set-Wert und somit den gleichen Abstand zu der zu repräsentierenden Grenzfläche/Grenzlinie (hier zwei Kreise), auf der die Level-Set-Funktion den Wert 0 annimmt (blau markierte Ebene).

der Repräsentation von Grenzflächen mithilfe einer Abstandsfunktion verdeutlichen die Abbildungen 2.17 und 2.18.

Freie Oberflächen in der Strömungssimulation

Da die im Strömungsgebiet herrschende Strömungsgeschwindigkeit $v(x, t)$ im Allgemeinen von 0 verschieden ist, verändert sich somit auch die Geometrie der Grenzfläche der Phasen. Unter der Voraussetzung, dass kein Partikel des Fluids von der einen Phase in die andere wechseln kann, muss die substantielle Ableitung der Level-Set-Funktion verschwinden.

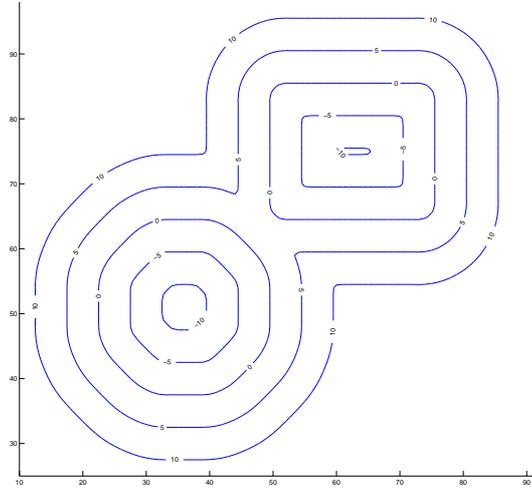


Abbildung 2.18: Die Konturen einer Level-Set-Funktion in zwei Dimensionen: Die Grenzfläche/Grenzlinie (hier ein Kreis und ein Rechteck) wird durch die Kontur zum Wert 0 definiert. Auf einer Konturlinie haben alle Punkte den gleichen Abstand zur Grenzfläche/Grenzlinie.

Definition 2.3 [SUBSTANTIELLE ABLEITUNG]

Sei $v(x, t)$ das Geschwindigkeitsfeld, dann ist die substantielle Ableitung von ϕ wie folgt definiert:

$$\frac{D}{Dt}\phi := \frac{\partial}{\partial t}\phi + (v \cdot \nabla)\phi. \quad (2.14)$$

Daraus folgt eine Gleichung für den Transport der Level-Set-Funktion ϕ mit dem Strömungsfeld v :

$$\begin{aligned} \frac{D}{Dt}\phi &\stackrel{!}{=} 0 \\ \Leftrightarrow \frac{\partial}{\partial t}\phi &= -(v \cdot \nabla)\phi. \end{aligned} \quad (2.15)$$

Für die Numerik einer Strömungssimulation ist die vorzeichenbehaftete Abstandseigenschaft der Level-Set-Funktion von großer Bedeutung, da nur so die Oberflächenspannung und die Regularisierung der Fluidparameter korrekt berechnet werden können. Nach dem Transportschritt 2.15 erfüllt die Level-Set-Funktion im Allgemeinen nicht mehr die Eikonalgleichung, so dass die Abstandseigenschaft nun gestört ist. Ziel ist nun, aus der gestörten Level-Set-Funktion unter Beibehaltung der Nullkontur wieder eine vorzeichenbehaftete Abstandsfunktion zu generieren. Es gibt zahlreiche Methoden, die dieses Ziel

verfolgen. Besonders nennenswert sind darunter die Fast-Marching- und Narrow-Band-Methode von Sethian [48]; sie rekonstruieren die Abstandsfunktion in expliziter Weise unter Verwendung sehr effizienter, auf Binärbäumen basierender Algorithmen. Eine andere Methode beruht auf der Beobachtung von J.-M. Morel und wurde von Sussman, Smereka und Osher [52] entwickelt. Dieser Ansatz führt zur sogenannten Level-Set-Reinitialisierung.

2.4.3 Level-Set-Reinitialisierung

Definition 2.4 [REINITIALISIERUNG]

Die Reinitialisierung ist ein iterativer Prozess, der schrittweise die Abstandseigenschaft der gestörten Abstandsfunktion ϕ wiederherstellt. Die Iteration besteht darin, dass Anfangswertproblem der Gleichung vom Hamilton-Jacobi-Typ

$$\begin{aligned} \frac{\partial}{\partial \tau} d(x, \tau) &= \text{sign}(\phi) \cdot (|\nabla d| - 1) \\ d(x, 0) &= \phi(x) \end{aligned} \tag{2.16}$$

bis zu einem Zeitpunkt $\tau = T_{\text{end}}$ zu lösen.

Dabei ist die Signumfunktion wie folgt definiert:

$$\text{sign}(\phi) := \begin{cases} -1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi = 0 \\ +1 & \text{if } \phi > 0 \end{cases}.$$

Bei dieser Reinitialisierung wird die Funktion d aus Gleichung 2.16 mit der gestörten Abstandsfunktion ϕ initialisiert und bis zu einem Zeitpunkt $\tau_{\text{stationär}} > 0$ berechnet, an dem d stationär wird (also $\frac{\partial}{\partial \tau} d = 0$). Es ist leicht ersichtlich, dass d im stationären Zustand gerade die Eikonalgleichung und somit die Abstandseigenschaft erfüllt, während die Signumfunktion in Gleichung 2.16 für den Informationsfluss von der Grenzfläche aus in das Gebiet hinein sorgt, die Grenzfläche zu $\phi = 0$ jedoch nicht verändert. Auf diese Weise wird d wieder sukzessive an einer Abstandsfunktion ausgerichtet, so dass man anschließend mittels einer Substitution $\phi(x) \leftarrow d(x, \tau_{\text{stationär}})$ wieder eine neue Abstandsfunktion ϕ erhält. Eine wichtige Frage jedoch ist die nach dem Zeitpunkt T_{end} , an dem die Lösung von Gleichung 2.16 einer Abstandsfunktion hinreichend ähnlich ist. Dabei genügt es in der Praxis, die Abstandseigenschaft in einem schmalen Band der Breite 2ϵ um die Grenzschicht herum herzustellen. Denn in dem in dieser Arbeit verwendeten Zweiphasenmodell für inkompressible Navier-Stokes-Gleichungen (siehe Kapitel 5 in [18]) werden die in einer ϵ -Umgebung der Grenzschicht regularisierte Delta- und Heaviside'sche Gewichtsfunktion zum Einsetzen der über die Grenzschicht wechselnden Fluidparameter

wie Dichte und Viskosität und der Oberflächenspannung verwendet. Nach der anfänglichen globalen Berechnung von $|\nabla d| = 1$ zu Beginn einer Simulation wird anschließend nur noch $|\nabla d| = 1$ für das Band $|d| < \epsilon$ verlangt.

Gleichung 2.16 in der Form einer Transportgleichung (vergleiche Gleichung 2.15)

$$\frac{\partial}{\partial \tau} d(x, \tau) + \vec{c} \cdot \nabla d = \text{sign}(\phi), \text{ mit } \vec{c} = \text{sign}(\phi) \frac{\nabla d}{|\nabla d|} \quad (2.17)$$

zeigt, dass in dieser nichtlinearen Differentialgleichung die Charakteristiken von der Nullkontur aus in Richtung der Normalen verlaufen, weswegen d zunächst in unmittelbarer Nähe der Grenzschicht die Abstandseigenschaft erhält. Aufgrund dieses günstigen Umstandes werden nur wenige Zeitschritte der Gleichung 2.16 benötigt, um d im Band $|d| < 2\epsilon$ zu reinitialisieren. Die genaue Anzahl der benötigten Zeitschritte kann daher eingegrenzt werden, da aus Gleichung 2.17 hervorgeht, dass sich die Information mit der Geschwindigkeit vom Betrag 1 in beide Richtungen von der Grenzschicht wegbewegt. Es ergibt sich also für die Anzahl der nötigen Zeitschritte $iter_{max}$ beziehungsweise für die Endzeit T_{end} :

$$iter_{max} = \frac{\epsilon}{\Delta t}, \text{ mit Zeitschrittweite } \Delta t$$

$$T_{end} = \epsilon.$$

Kapitel 3

Adaptivität

Ziel dieser Arbeit ist die Entwicklung eines verbesserten Level-Set-Verfahrens von zwei-phasigen Strömungen, deren freie Oberfläche sowohl topologisch als auch geometrisch sehr komplex sein kann (zur Beschreibung von komplexen Geometrien siehe Abschnitt 2.3.1). Ferner weist der Abschnitt 2.3.2 auf Seite 24 auf die Möglichkeit der Adaptivität hin, so dass eine den räumlichen Eigenschaften des Problems angepasste Diskretisierungsgenauigkeit erreicht werden kann. Es wird aber nicht nur der Diskretisierungsfehler, sondern auch die Stabilität eines Verfahrens von der lokalen Maschenweite beeinflusst. In Abschnitt 4.1 wird dieser Aspekt in Bezug auf die Numerik der Level-Set-Methode erläutert. Im Hinblick auf die Erweiterung des Programms NaSt3DGPF sind bereits die Vorzüge von strukturierten und insbesondere kartesischen Gittern genannt worden. Dieses Kapitel stellt nun gängige adaptive Techniken vor, die auf strukturierten beziehungsweise kartesischen Gittern basieren und sich prinzipiell als Erweiterungslösung in NaSt3DGPF integrieren lassen.

Je nachdem, welche Art von Gitter in einer Diskretisierung Verwendung findet, gibt es viele Methoden, einer Diskretisierung die gewünschte Adaptivität zu verleihen. Im Falle von Triangulierungen geschieht dies durch das Hinzufügen von Knoten und Kanten, so dass einzelne Dreiecke gezielt in mehrere zerlegt werden. Bei kartesischen Gittern werden in der Regel zusätzliche Freiheitsgrade an kanonischen Positionen in fest definierten Abständen zu den Grobgitterpunkten eingefügt, wodurch das Gitter nun zu einem unstrukturierten Gitter - etwa einem octree - wird. Ein unerwünschter Effekt bei Adaptivität ist im Allgemeinen die Entstehung sogenannter hängender Knoten. Durch diese wird das Gleichgewicht von Freiheitsgraden im Gitter und daher auch die Konsistenz lokal gestört.

Definition 3.1 [HÄNGENDER KNOTEN]

Ein hängender Knoten ist ein Freiheitsgrad, der einer Ecke einer Gitterzelle, nicht aber einer Ecke einer der anliegenden Gitterzellen angehört. Ein hängender Knoten verursacht ein Ungleichgewicht zwischen Zellen und Freiheitsgraden eines Gitters und stört somit lokal dessen Konsistenz.

Ein Beispiel für hängende Knoten zeigt die Abbildung 3.1. Abbildung 3.2 macht deutlich, dass es im Falle von Triangulierungen durch Einfügen von weiteren Kanten mög-

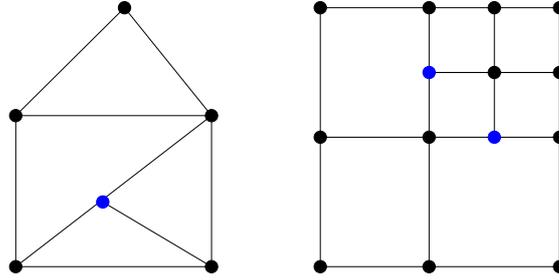


Abbildung 3.1: Hängende Knoten (blau) in einem triangulierten und einem kartesischen Gitter.

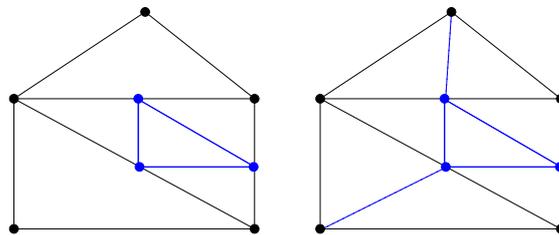


Abbildung 3.2: Knoten (blau) werden einer Triangulierung hinzugefügt (links). Diese teilen die Kante eines Dreiecks und werden somit zu hängenden Knoten. Durch Hinzufügen weiterer Kanten (gestrichelte Linien) werden korrespondierende Dreiecke unterteilt, so dass nun jedes Dreieck genau drei Knoten in seinen Eckpunkten besitzt.

lich ist, hängende Knoten zu eliminieren. Die Numerik von NaSt3DGPF basiert auf der Finite-Differenzen-Methode und operiert daher auf strukturierten kartesischen Gittern. Für die Erweiterung des bestehenden NaSt3DGPF-Programms im Hinblick auf Adaptivität soll daher verstärkt auf kartesische Gitter geachtet werden. In den folgenden Abschnitten wird daher nicht mehr auf Triangulierungen eingegangen. Im Gegensatz zum oben geschilderten Fall bei Triangulierungen können jedoch hängende Knoten auf unstrukturierten kartesischen Gittern nicht beseitigt werden. Ein Verfahren mit octrees bedarf daher einer besonderen Behandlung von hängenden Knoten. Abschnitt 3.2 schildert die Kompensation des von hängenden Knoten verursachten Ungleichgewichts zwischen Gitterzellen und Freiheitsgraden durch spezielle Interpolationsverfahren. Zwei weitere bekannte Verfahren mit adaptivem Charakter sind die Mehrgitter- und Gebietszerlegungsmethoden. Sie erlauben die Diskretisierung mithilfe strukturierter Gitterkomponenten und umgehen somit das für octrees typische Problem mit den hängenden Knoten.

Eine der zentralen Fragen im Zusammenhang mit Adaptivität ist die Frage nach einem geeigneten Verfeinerungskriterium, welches die Information darüber gibt, an welcher Stelle im Gitter eine Verfeinerung stattfinden soll. Für die Behandlung von freien Ober-

flächen ist es sinnvoll, in der Nähe dieser fein aufzulösen. Wird zudem die Oberfläche durch eine Abstandsfunktion beschrieben, wie bei Level-Set-Methoden üblich, so wird in den Bereichen des Gebiets verfeinert, an denen die Level-Set-Funktion vom Betrag her kleiner ist als ein bestimmter Grenzwert - die Bandbreite (siehe Abbildung 3.3a). Eine konkrete Wahl für ein geeignetes Verfeinerungskriterium wird in Abschnitt 4.2.1

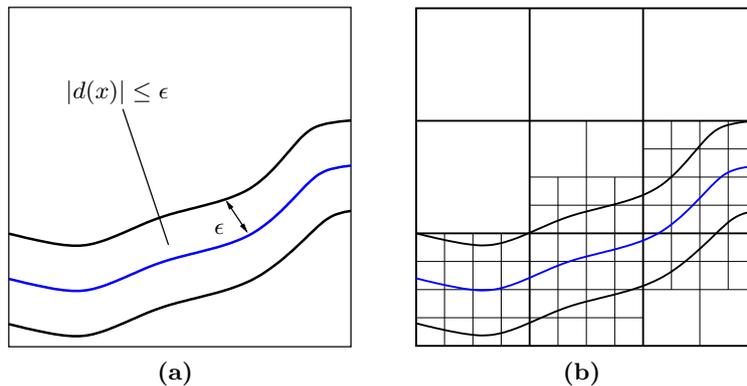


Abbildung 3.3: (a) Die Abstandsfunktion d der freien Oberfläche definiert ein schmales Band der Breite 2ϵ , in dem die Verfeinerung stattfinden soll. (b) Eine Verfeinerung mit octrees: Die größten Zellen bilden jeweils die Wurzel eines octrees mit dem Level 0. Die feinsten Zellen haben das Level 2.

erarbeitet.

3.1 Verfahren mit octrees

Ein erster Ansatz zur lokalen Verfeinerung der kartesischen Gitter basiert auf sogenannten octrees in drei sowie quadrees in zwei Dimensionen (zur Definition siehe Abschnitt 2.1.2 auf Seite 18). Die Vorgehensweise ist einfach: Jede Blattzelle, deren Verfeinerungslevel noch nicht dem maximalen Level entspricht, wird verfeinert, wenn für sie das Verfeinerungskriterium erfüllt ist. Es lassen sich aber nicht nur freie Oberflächen in Strömungssimulationen, sondern auch allgemein geometrische Strukturen wie Gebietsränder oder Strömungshindernisse mit octrees approximieren. Ein beliebtes Mittel zur Generierung solcher Strukturen ist die im Folgenden vorgestellte constructive solid geometry.

3.1.1 Geometrierzeugung mit constructive solid geometry

Zur Definition von geometrischen Daten bedient man sich in der Computergraphik und im Bereich computer aided design (CAD) der Modellierung mittels constructive solid geometry (CSG, Konstruktive Festkörpergeometrie). Diese Methode erlaubt es dem

Anwender, aus primitiven Objekten (vergleiche Abbildung 2.13 auf Seite 24) durch mengentheoretische Operationen neue Objekte mit komplexerer Geometrie zu erzeugen. Als Operationen stehen dabei Vereinigung, Schnitt und Differenz zur Verfügung. Abbildung 3.4a zeigt das Funktionsprinzip von CSG.

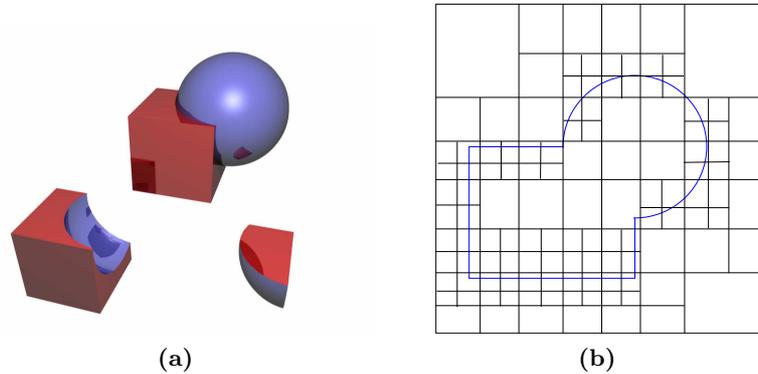


Abbildung 3.4: (a) Die primitiven Objekte Würfel und Sphäre werden durch eine einmalige CSG-Operation zu neuen Objekten transformiert. Oben Vereinigung, links Differenz und rechts Schnitt. (b) Gitter mit Octree-Struktur approximieren den Rand eines CSG-Konstruktes (Rechteck vereinigt mit Sphäre) in zwei Dimensionen.

3.1.2 Approximation von Oberflächen mit octrees

Nun soll getestet werden, wie gut sich Gitter mit Octree-Struktur dazu eignen, die Details der Geometrie von Oberflächen zu diskretisieren. Dazu sei noch einmal festgestellt, dass das Gitter die Oberfläche sehr fein, das übrige Gebiet jedoch nur grob diskretisieren soll. Ein einfaches, daraus abgeleitetes Verfeinerungskriterium könnte daher lauten: Jede Blattzelle, deren Level noch nicht dem maximalen Level entspricht, würde verfeinert, wenn sie sowohl Tochterzellen auf der einen als auch auf der anderen Seite der Fläche besäße. Das heißt, eine Verfeinerung einer Zelle findet genau dann statt, wenn ihre Tochterzellen die Oberfläche besser aufzulösen vermögen als die nicht verfeinerte Zelle selbst. Abbildung 3.4b zeigt ein zweidimensionales Beispiel. Eine reale Implementierung der CSG-Methode zur Konstruktion von Gittern mittels octrees in drei Dimensionen wurde anhand einiger CSG-Konstrukte getestet. Die Abbildung 3.5 zeigt Schnitte durch dreidimensionale octrees, die mit dem oben genannten Algorithmus den Rand der CSG-Objekte stellvertretend für eine komplexe freie Oberfläche approximieren.

3.1.3 Fully threaded trees

Für einen numerischen Algorithmus ist es notwendig, zu einer gegebenen Gitterzelle ihre Nachbarschaftsbeziehungen zu kennen. Das bedeutet, dass während einer Rechnung

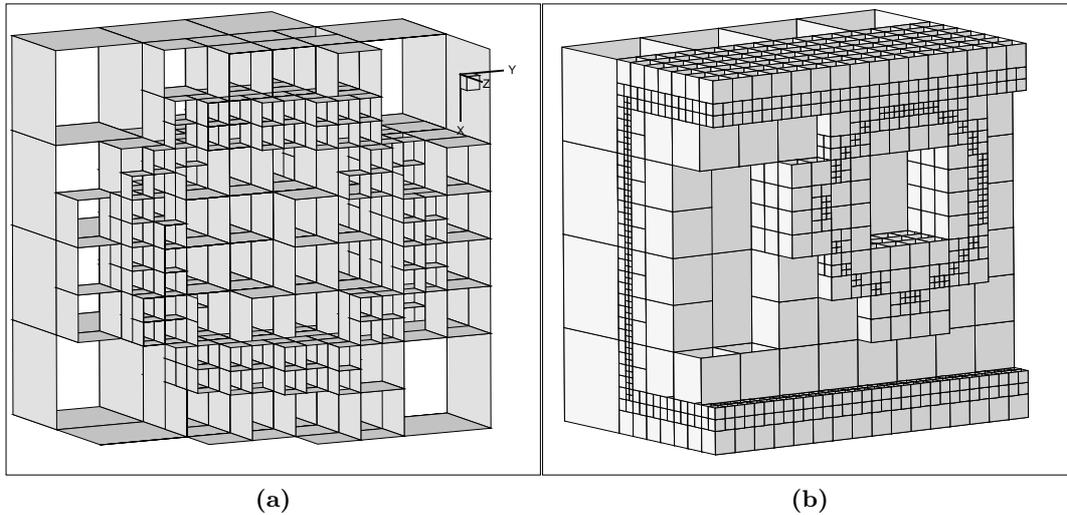


Abbildung 3.5: Kartesische Gitter werden mit octrees verfeinert, so dass sie den Rand von CSG-Objekten approximieren. In Bild (a): Eine Sphäre. Bild (b): Ein Quader, aus dem ein Zylinder ausgeschnitten wurde. Zu sehen sind jeweils die Gitterzellen eines octrees, aus dem die vordersten 40% ausgeblendet wurden; der Rand der Geometrie wurde der Übersichtlichkeit halber nicht dargestellt.

häufig auf direkte Nachbarzellen, die Mutterzelle oder die Tochterzellen zugegriffen wird. Zum Zweck einer effizienten Implementierung von octrees ist es das Ziel, diese Zugriffe in möglichst kurzer Zeit zu gewährleisten. Da die Konnektivitäten in einer gewöhnlichen Baumstruktur nur zwischen Tochter- und Mutterzelle verlaufen (vergleiche Abbil-

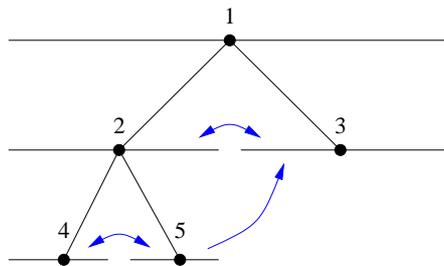


Abbildung 3.6: Nachbarschaftsbeziehungen (blaue Pfeile) und direkte Konnektivität (schwarze Linien) in einem binären Baum in einer Dimension (analog quadrees/octrees in zwei/drei Dimensionen): Damit die Zellen 2 und 4 ihre Nachbarn erreichen, muss der Baum über ein Level traversiert werden. Zelle 5 findet ihren Nachbarn 3 nur über zwei Level hinweg.

dung 3.6), muss der Baum zum Auffinden von Nachbarn über mehrere Level traversiert

werden. Im schlimmsten Fall erreicht der Suchvorgang sogar die Baumwurzel. Im Mittel jedoch gibt Khokhlov in [33] die Anzahl der zu traversierenden Level mit $\simeq 2$ an. Um einer Entartung des Baumes entgegenzuwirken, wird häufig die Irregularität - das ist die Differenz der Level benachbarter Zellen - auf 1 beschränkt. Das heißt, für zwei Zellen i und j , die über ihre Kanten oder diagonal benachbart sind, soll $|l(i) - l(j)| \leq 1$ mit $l(i)$ Level von i gelten. Obwohl diese Einschränkungen nicht von fundamentaler Notwendigkeit sind, erleichtern sie die Berechnung von Differenzenoperatoren, wie beispielsweise ∇_d als Diskretisierung von ∇ , erheblich (siehe Abschnitt 3.2 auf Seite 40). Aus Gründen der Effizienz werden herkömmliche quadrees und octrees im Allgemeinen nicht für numerische Zwecke eingesetzt. Einen anderen Ansatz bieten dagegen fully threaded trees (FTT), wie sie Khokhlov in [33] beschreibt: Wie schon im Fall von strukturierten kartesischen Gittern ist das Rechengebiet im Allgemeinen wieder ein Quader. Exemplarisch wird jedoch davon ausgegangen, dass dieser Quader im Besonderen ein Würfel der Kantenlänge L ist; dies ist keine Einschränkung, da sich das Prinzip der FTT analog auf quaderförmige Gebiete erweitern lässt. Das gesamte Gebiet wird nun mit der Wurzel des FTT identifiziert und in Zellen unterschiedlicher Größe $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots, \frac{1}{2^{l_{max}}}$ von L unterteilt mit dem höchsten Level l_{max} . Damit ein numerischer Algorithmus nun effizient arbeiten kann, sollten zu jeder Zelle i folgende Informationen einfach und schnell zugänglich sein:

$$\begin{aligned}
 Lv(i) & \text{ — das Level der Zelle} \\
 S(i) & \text{ — } true/false: \text{ Zelle ist verfeinert oder ein Blatt} \\
 P(i) & \text{ — Zeiger auf die Mutterzelle} \\
 Ch(i, j) & \text{ — Zeiger auf die Tochterzellen } j = 1, \dots, 8 \\
 Nb(i, j) & \text{ — Zeiger auf die Nachbarn } j = 1, \dots, 6 \\
 D(i) & \text{ — die eigentlichen Nutzdaten} \\
 r(i) & \text{ — der Ortsvektor des Zellmittelpunktes}
 \end{aligned} \tag{3.1}$$

Bestimmt durch das Level $Lv(i)$ ergibt sich für die Ausdehnung der Zelle i : $\Delta_i = 2^{-Lv(i)}L$. Ferner ist zu beachten, dass die Nachbarschaftsbeziehungen von Zellen nicht symmetrisch sind; so hat in Abbildung 3.6 die Zelle 5 die Zelle 3 zum Nachbarn, die Zelle 3 jedoch hat Zelle 2 als Nachbarn. Anders formuliert soll gelten: Der j -te Nachbar von i hat entweder dieselbe Größe wie i , $\Delta_{Nb(i,j)} = \Delta_i$, oder ist zweimal so groß, $\Delta_{Nb(i,j)} = 2\Delta_i$. Im ersten Fall ist die Zelle j ein Blatt oder eine verfeinerte Zelle, im letzteren Fall kann j lediglich eine Blattzelle sein. Die konkrete Implementierung eines FTT könnte nun zu jeder Zelle neben den Nutzdaten auch alle Zeiger, Koordinaten und flags speichern gemäß obenstehender Liste 3.1, was einem unvertretbar großen Overhead von 18 words¹ entspräche. Ferner kann ein so konstruierter Baum nicht im Parallelbetrieb

¹Die Bezeichnung word steht hier für einen nicht näher definierten Speicherbedarf – in der Regel jedoch 4 byte. Der exakte Speicherbedarf ist abhängig von Hardware, Betriebssystem, Programmiersprache und Compiler.

modifiziert werden, da Nachbarn einer Zelle wiederum Zeiger auf diese Zelle enthalten und beim Entfernen der Zelle umgebogen werden müssten. Grundsätzlich sind vom Entfernen einer Zelle auch immer andere Zellen mitbetroffen. Ein Konflikt tritt vor allem auf, sobald zeitgleich zwei benachbarte Zellen aus dem Baum entfernt oder erzeugt werden. Im Parallelbetrieb ist man jedoch vor derartigen Konflikten nicht geschützt.

Eine auch für parallele Algorithmen geeignete lauffzeit- und speichereffiziente Implementierung von FTT stützt sich auf folgende drei Beobachtungen:

1. Wenn eine Zelle verfeinert wird, so werden ihre acht Tochterzellen simultan erzeugt. Sie können daher zusammenhängend im Speicher abgelegt und über einen einzigen (anstatt acht) Zeiger referenziert werden.
2. Da alle acht Tochterzellen (Geschwister) zusammenhängend abgespeichert werden, sind ihre Nachbarschaftsbeziehungen implizit bekannt, und es müssen keine weiteren Zeiger zwischen Geschwistern gespeichert werden.
3. Nachbarzellen sind entweder Geschwister oder Tochterzellen eines Nachbarn der Mutterzelle. Es gibt nur eine begrenzte Anzahl an Nachbarschaftsbeziehungen zwischen Geschwistern von benachbarten Mutterzellen. Daher kann ein Nachbar einer Tochterzelle ohne Suche unter den Tochterzellen des Nachbarn gefunden werden.

Die FTT-Struktur, wie sie für diese Arbeit implementiert wurde, zeigt Abbildung 3.7 an einem zweidimensionalen Beispiel. Alle acht Tochterzellen sind zu einem sogenannten oct gruppiert, welcher neben den acht Zellen noch weitere, diesen Zellen gemeine Informationen beinhaltet: einen Zeiger P auf die Mutterzelle, das Level L_v des octs, den Ortsvektor r der Mutterzelle und ein array $OctNb(6)$ mit Zeigern auf die Mutterzellen von benachbarten octs. Diese Implementierung benötigt 17 words pro oct beziehungsweise maximal drei words pro Zelle an Speicherplatz und reduziert den Mehraufwand damit signifikant. Dank dieser Struktur befindet sich die Hälfte aller Nachbarn einer Zelle im selben oct. Die übrigen Nachbarn erreicht man nun über benachbarte octs der Mutterzelle, deren Zeiger der oct selbst vorhält und indem lediglich eine Addition oder Subtraktion eines Index vorgenommen werden muss, um schließlich den gewünschten Nachbarn aufzusuchen. Daher beläuft sich die durchschnittliche Anzahl der zu traversonierenden Level während einer Nachbarsuche auf $\simeq \frac{1}{2}$; das ist um den Faktor 4 besser als bei herkömmlichen octrees im Durchschnitt. Ferner wird aus Abbildung 3.7 deutlich, warum diese besondere Art der Implementierung eines FTT den problemlosen Einsatz von parallelen Algorithmen ermöglicht. Durch einfaches Löschen des assoziierten octs, auf den $OctCh$ (in der zweidimensionalen Abbildung $BinCh$) zeigt, und durch Setzen des Wertes nil für diesen Zeiger, kann auf einfache Weise eine verfeinerte Zelle wieder vergrößert werden. Es fallen keine weiteren Änderungen an anderen Zellen oder octs an. Ähnlich einfach ist das Verfeinern einer Zelle, indem ein neuer oct erzeugt und sein array $OctNb(6)$ ($BinNb(2)$ in der zweidimensionalen Abbildung) mit den Zeigern auf

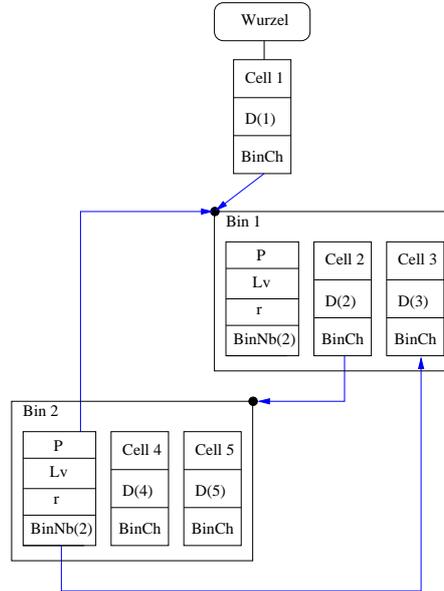


Abbildung 3.7: Der zweidimensionale Beispielbaum aus Abbildung 3.6 als FTT implementiert: Die Tochterzellen werden zu einem bin (oct in drei Dimensionen) gebündelt. Jeder bin speichert zusätzlich einen Zeiger P auf die Mutterzelle, das Level Lv des bins, den Ortsvektor r der Mutterzelle und zwei Zeiger (sechs in drei Dimensionen) auf Mutterzellen von benachbarten bins im array $BinNb(2)$.

Nachbarn der zu verfeinernden Zelle initialisiert wird. Wiederum sind in keiner anderen Zelle oder keinem weiteren oct Änderungen vorzunehmen.

3.2 Numerik auf octrees

Abhängig von der Wahl des Gitters müssen nun die zu berechnenden Größen und die Operatoren der Gleichung auf diesem Gitter dargestellt werden. Zunächst ist es Ziel dieser Arbeit, den Level-Set-Algorithmus, wie er bereits in NaSt3DGPF auf kartesischen Gittern implementiert ist, so zu erweitern, dass aus ihm eine Version für adaptive Gitter wird. So wird die Level-Set-Funktion weiterhin als in den Zellenmittelpunkten lokalisiert angenommen. Genauer heißt das im Fall von octrees, dass jede Blattzelle in ihrem Mittelpunkt einen Wert der Level-Set-Funktion speichert. Als Operator gilt es nun den Gradienten der Level-Set-Funktion ∇d zu diskretisieren, wobei nun erstmals die Probleme von hängenden Knoten (siehe Definition auf Seite 33) deutlich werden: In Bezug auf Nachbarschaftsbeziehungen der Freiheitsgrade fehlen den Knoten, deren Nachbarn keine Blätter sind oder auf einem geringeren Level stehen, korrespondierende Nachbarn (siehe auch Abbildung 3.8). Dadurch wird insbesondere die Auswertung von Differenzenster-

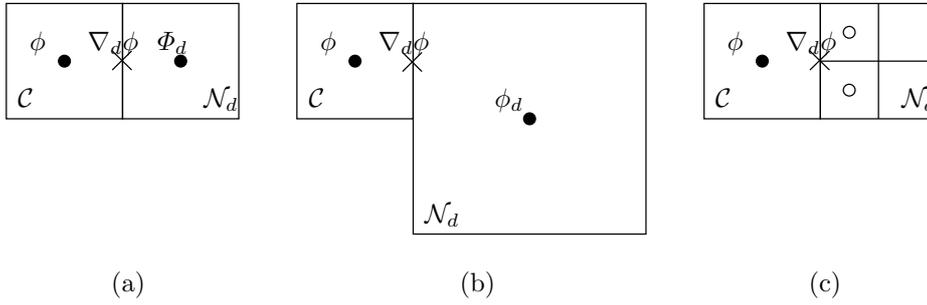


Abbildung 3.8: Es sind drei Fälle bei der Berechnung des Gradienten zu unterscheiden: (a) Zellen auf gleichem Level. (b) Fein-grob-Grenze. (c) Grob-fein-Grenze.

nen erschwert. Zu diesem Problem gibt Popinet in [39] eine Methode zur konsistenten Interpolation von $\nabla\phi$ auf hängenden Knoten in octrees an: Es sei angenommen, dass der Gradient als lineare Funktion von ϕ in den Zellenmittelpunkten

$$\nabla_d\phi = \alpha_d\phi + \beta_d$$

mit den Konstanten α und β dargestellt werden kann, wobei die Funktionen α und β in ϕ der anliegenden Zellen linear sind. Nun müssen drei Fälle unterschieden werden (vergleiche Abbildung 3.8). Falls der Nachbar \mathcal{N}_d einer Zelle in Richtung d dasselbe Level hat und ein Blatt ist, ergibt sich für $\nabla_d\phi = \frac{(\phi_d - \phi)}{h}$ mit ϕ_d als Wert von ϕ im Mittelpunkt von \mathcal{N}_d . Mit obenstehender Notation ergibt sich $\alpha_d = -\frac{1}{h}$ und $\beta_d = \frac{\phi_d}{h}$. Wie die Situation

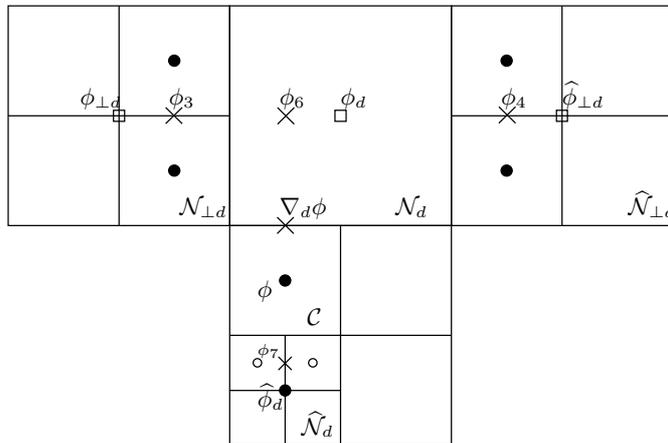


Abbildung 3.9: Eine Interpolation zweiter Ordnung für die Berechnung des Gradienten an einem Fein-grob-Übergang.

für den Fall aussieht, dass \mathcal{N}_d auf niedrigerem Level steht (vergleiche Abbildung 3.8b),

zeigt die Abbildung 3.9. Um die Genauigkeit der Ordnung 2 beizubehalten, wird nun eine Drei-Punkt-Interpolation durchgeführt, indem $\nabla_d\phi$ mittels einer durch ϕ_6 , ϕ und entweder ϕ_7 oder $\widehat{\phi}_d$ verlaufenden Parabel approximiert wird. Aufgrund der vorgeschriebenen Beschränkung der Irregularität von octrees hat $\widehat{\mathcal{N}}_d$ dasselbe Level wie \mathcal{C} . Falls $\widehat{\mathcal{N}}_d$ ein Blatt ist, erhält man für $\nabla_d\phi$

$$h\nabla_d\phi = -\frac{\phi}{3} - \frac{\widehat{\phi}_d}{5} + \frac{8}{15}\phi_6, \quad (3.2)$$

wobei der Wert $\widehat{\phi}_d$ im Mittelpunkt von $\widehat{\mathcal{N}}_d$ benutzt worden ist. Falls $\widehat{\mathcal{N}}_d$ jedoch kein Blatt ist, wird ein interpolierter Wert ϕ_7 in $\widehat{\mathcal{N}}_d$ durch Mittelung der Werte seiner Tochterzellen konstruiert, die an \mathcal{C} angrenzen (durch \circ in Abbildung 3.9 gekennzeichnet). Dann ist der Gradient gegeben durch

$$h\nabla_d\phi = -\frac{2}{9}\phi - \frac{8}{27}\phi_7 + \frac{14}{27}\phi_6. \quad (3.3)$$

Es bleibt noch die Aufgabe, den Wert ϕ_6 zu ermitteln: Dieser muss durch ϕ_d und Werte in senkrechter Richtung zu d benachbarter Zellen $\widehat{\mathcal{N}}_{\perp d}$ und $\mathcal{N}_{\perp d}$ interpoliert werden. $\widehat{\mathcal{N}}_{\perp d}$ und $\mathcal{N}_{\perp d}$ haben wegen der beschränkten Irregularität dasselbe Level wie die Zelle \mathcal{N}_d . Sind diese jedoch keine Blattzellen, ergeben sich ϕ_3 und ϕ_4 , indem wieder Mittelungen der durch \bullet in Abbildung 3.9 gekennzeichneten Werte vorgenommen werden. Folgende Fälle resultieren:

$$\phi_6 = \begin{cases} \frac{15}{16}\phi_d - \frac{3}{32}\widehat{\phi}_{\perp d} + \frac{5}{32}\phi_{\perp d} & \text{falls } \widehat{\mathcal{N}}_{\perp d} \text{ und } \mathcal{N}_{\perp d} \text{ Blätter sind} \\ \frac{5}{6}\phi_d - \frac{1}{14}\widehat{\phi}_{\perp d} + \frac{5}{21}\phi_3 & \text{falls } \widehat{\mathcal{N}}_{\perp d} \text{ ein Blatt ist} \\ \phi_d - \frac{1}{7}\phi_4 + \frac{1}{7}\phi_{\perp d} & \text{falls } \mathcal{N}_{\perp d} \text{ ein Blatt ist} \\ \frac{8}{9}\phi_d - \frac{1}{9}\phi_4 + \frac{2}{9}\phi_3 & \text{in allen anderen Fällen} \end{cases} \quad (3.4)$$

Wie oben erwähnt, kann $\nabla_d\phi$ als lineare Funktion von ϕ beschrieben werden; die entsprechenden Werte α_d und β_d ergeben sich dann aus 3.2, 3.3 und 3.4. Im dritten Fall hat die Nachbarzelle \mathcal{N}_d dasselbe Level wie \mathcal{C} , ist jedoch keine Blattzelle (vergleiche Abbildung 3.8.c). Der Gradient wird in diesem Falle einfach als negatives Mittel der Gradienten berechnet, die durch die an die Zelle \mathcal{C} angrenzenden Tochterzellen von \mathcal{N}_d erzeugt werden. Wiederum werden diese Gradienten durch die schon im vorherigen Fall angewendete Interpolationsmethode gewonnen. Eine Erweiterung dieses Schemas auf drei Dimensionen geschieht analog zur oben beschriebenen Methode für zwei Dimensionen.

3.3 Mehrgitter

Das vorhergehende Kapitel hat die Verwendung von adaptiven Gittern zur adäquaten Auflösung detaillierter Strukturen motiviert. Die Suche nach konkreten Diskretisierungen

beziehungsweise Verfahren und Methoden, die adaptiven Charakter haben, führt unter anderem zu den sogenannten Mehrgittermethoden. Sie erlauben das effiziente Lösen von Gleichungssystemen auch auf sehr fein aufgelösten Gittern (zum Beispiel $\mathcal{O}(N)$ für das Poisson-Problem in zwei Dimensionen mit der Anzahl N an Freiheitsgraden). Ferner ist die Mehrgittermethode kompatibel mit adaptiver Gitterverfeinerung, indem das Gebiet von einer Hierarchie von Gittern unterschiedlicher Maschenweite überdeckt wird (siehe Abbildung 3.10). Brandt [9, 10] hat aufgrund dieser Idee die sogenannte

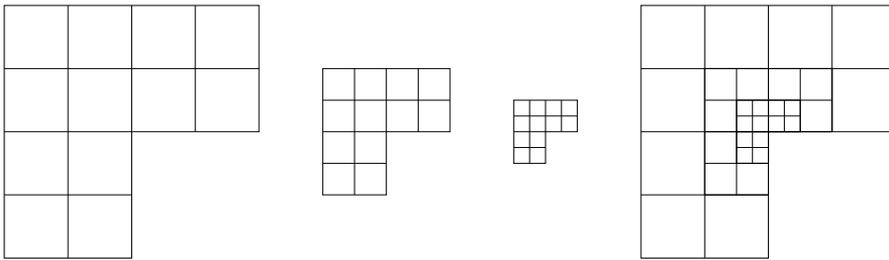


Abbildung 3.10: Eine Gitterhierarchie: Die ersten drei Gitter (von links) haben unterschiedliche Maschenweiten. Ein feineres Gitter deckt nur einen kleinen Teil des nächst größeren Gitters ab. Zusammengesetzt ergeben sie das adaptive Gitter (rechts).

multilevel adaptive technique (MLAT) eingeführt. An dieser Stelle sollen die Grundzüge eines Mehrgitterverfahrens kurz vorgestellt werden (siehe auch Trottenberg, Oosterlee und Schüller [56] sowie Stüben [32]): Aus der Beobachtung, dass herkömmliche iterative Gleichungslöser den Fehler zwar nach wenigen Iterationen stark glätten, diesen aber nicht wesentlich verringern, resultiert das Glättungsprinzip: Viele iterative Gleichungslöser haben einen starken Glättungseffekt auf den Fehler.

Das zweite Prinzip ist das Grobgitterprinzip: Ein glatter Fehler kann auch auf einem groben Gitter gut approximiert werden. Eine Grobgitterprozedur verursacht substantiell weniger Kosten als eine Feingitterprozedur.

Da dieses Prinzip nicht nur für Fehler, sondern auch für Korrekturgrößen wie Defekt und Korrektor gilt, spricht man auch vom Prinzip der Grobgitterkorrektur.

Neben einem iterativen Löser, der als Glätter dient, gibt es noch einen exakten Löser und die Transferoperatoren als Hauptkomponenten eines Mehrgitteralgorithmus. Letztere transferieren den Defekt vom feinen aufs gröbere Gitter – die Restriktion – beziehungsweise den Korrektor vom größeren aufs feine Gitter – die Interpolation. Ein Mehrgitteralgorithmus arbeitet nun wie folgt: Zu Lösen sei das lineare Problem auf dem feinen Gitter mit Maschenweite h

$$L_h u_h = f_h$$

mit dem linearen Operator L_h , der Lösung u_h und der rechten Seite f_h . Wenige Iterationen eines Löser glätten den Fehler und liefern eine glatte Approximation \hat{u}_h an die

Lösung u_h . Der Defekt

$$d_h = f_h - L_h \hat{u}_h$$

wird auf das nächst gröbere Gitter transferiert

$$d_H = I_h^H d_h.$$

Nun wird das neue System

$$L_H \hat{v}_H = d_H$$

auf dem groben Gitter gelöst. Dies geschieht entweder mit einem exakten Löser oder rekursiv mit dem Mehrgitteralgorithmus. Der gewonnene Korrektor \hat{v}_H kann nun mittels Interpolation auf das feinere Gitter übertragen werden

$$\hat{v}_h = I_H^h \hat{v}_H$$

und dort zu einer Korrektur

$$\hat{u}_h \leftarrow \hat{u}_h + \hat{v}_h$$

beitragen. Anschließend wird ein zweites Mal der Glätter auf

$$L_h \hat{u}_h = f_h$$

angewendet. Eine Formulierung dieses Algorithmus in Pseudo-Code zeigt Algorithmus 1. Für eine detaillierte Beschreibung zur Entwicklung von Mehrgitteralgorithmen siehe Brandt [11].

3.3.1 Mehrgitter auf composite grids

Das vorhergehende Grundlagenkapitel stellt die wichtigsten Arten von Gittern vor und zeigt die Vorteile von strukturierten Gittern, die im Wesentlichen in der kostengünstigen Gittergenerierung und der einfachen Implementierung numerischer Algorithmen auf diesen Gittern liegen (vergleiche Abschnitt 2.1.1 auf Seite 10). Andererseits bieten unstrukturierte Gitter die größte Flexibilität im Umgang mit der Geometrie, da ihre Freiheitsgrade fast beliebig auf dem Rand der Geometrie positioniert werden können. Eine Möglichkeit, die Vorteile strukturierter Gitter auch im Fall beliebiger Geometrien, wie sie auch bei freien Oberflächen von Strömungssimulationen vorkommen, nutzen zu können, besteht darin, das globale Gitter lokal aus einzelnen, strukturierten Gitterkomponenten zu einem sogenannten composite grid zusammenzusetzen. Abbildung 3.11 zeigt ein composite grid, bestehend aus einem Hauptgitter und einer der Geometrie angepassten Gitterkomponente. Im Fall von zeitabhängigen Geometrien beschränken sich erneute Anpassungen des Gitters eventuell auf einzelne lokale Gitterkomponenten - im einfachsten Fall Translationen, Rotationen und Skalierungen - und lassen den globalen Teil des Gitters unbeeinflusst.

Algorithmus : $\text{MGC}(A_h, v_h, f_h, l, \nu^1, \nu^2)$

Input : Operator A_h , Startwert v_h , Rechte Seite f_h , Level l, ν^1, ν^2

Output : $v_h \approx A_h^{-1} f_h$

$R := \|f_h - A_h v_h\|$

while $R > \epsilon$ **do** solange keine Konvergenz erreicht

if $l = l_{\min}$ **then** l ist das größte Level

$v_h := A_h^{-1} f_h;$ /* Löse direkt */

else

$v_h := (S^1)^{\nu^1}(A_h, v_h, f_h);$ /* Glätte ν^1 mal mit S^1 */

$d_h := (f_h - A_h v_h);$ /* Berechne den Defekt */

$f_{2h} := I_h^{2h} d_h;$ /* Restriktion von d_h als rechte Seite */

$\text{MGC}(A_{2h}, v_{2h}, f_{2h}, l + 1, \nu^1, \nu^2)$

$v_h := v_h + I_{2h}^h v_{2h};$ /* Feingitterkorrektur */

$v_h := (S^2)^{\nu^2}(A_h, v_h, f_h);$ /* Glätte ν^2 mal mit S^2 */

$R := \|f_h - A_h v_h\|$

Algorithmus 1 : Ein Multigrid-V-Zyklus

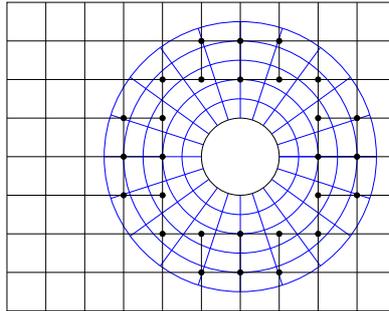


Abbildung 3.11: Ein composite grid mit einer strukturierten, kurvilinearen Gitterkomponente (blau). Die Freiheitsgrade des Hauptgitters in der Überlappregion sind mit • markiert.

Die Lösung verschiedener partieller Differentialgleichungen auf composite grids haben u.a. Starius [50, 51], Kreiss [35], Reyna [43], Atta und Vadyak [19], Benek et al. [30], Henshaw [27] and Berger [5] untersucht, während Chesshire [17] ein auf composite grids basierendes Programm zur Gittergenerierung (CMPGRD) entwickelt hat. Im Folgenden soll nun anhand eines Modellproblems eine Mehrgittermethode zur Lösung von elliptischen partiellen Differentialgleichungen, wie sie in Henshaw und Chesshire [28] beschrieben wird, vorgestellt werden.

Zu lösen sei das folgende

Modellproblem 3.1:

$$Lu := c_{xx} \frac{\partial^2 u}{\partial x^2} + c_{xy} \frac{\partial^2 u}{\partial y \partial x} + c_{yy} \frac{\partial^2 u}{\partial y^2} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} + c_c u = f \text{ in } \Omega, \quad (3.5)$$

$$Bu := b_x \frac{\partial u}{\partial x} + b_y \frac{\partial u}{\partial y} + b_c u = g \text{ auf } \partial\Omega \quad (3.6)$$

in zwei Dimensionen.

Diskretisierung

Die Diskretisierung besteht nun darin, ein Gitter zu erzeugen, das sich aus einem oder mehreren strukturierten Gittern - den Gitterkomponenten - zusammensetzt. Dabei ist darauf zu achten, dass sich die Gitter an ihren Rändern überlappen (vergleiche Abbildung 3.11). Obwohl nun einige Punkte in den Überlappregionen verworfen werden, betrachtet man jede Gitterkomponente für sich als logisch rechteckig. Dabei kann jeder Punkt einer Gitterkomponente einem der folgenden vier Typen zugeordnet werden:

1. ein innerer Punkt, an dem die Gleichung 3.5 angewendet wird,
2. ein Randpunkt, der $\partial\Omega$ zugeordnet ist und an dem die Randbedingung 3.6 zu erfüllen ist,
3. ein Punkt, an dem die Lösung mittels Interpolation an die Lösung anderer Gitterkomponenten gekoppelt ist; die Menge dieser Punkte wird auch Interpolationsrand genannt,
4. ein verworfener Punkt in einer Überlappregion.

Unter vielen Möglichkeiten, eine diskrete Approximierung an die Gleichung 3.5 mit ihrer Randbedingung 3.6 zu finden, wählen Henshaw und Chesshire die Abbildungsmethode und gehen wie folgt vor: Jede Gitterkomponente ist gerade so konstruiert, dass sie mit einer glatten Abbildung auf das Einheitsquadrat mit den Koordinaten (r, s) abgebildet werden kann, was zu einer Transformation der Gleichungen 3.5 und 3.6 führt:

$$Lu := c_{rr} \frac{\partial^2 u}{\partial r^2} + c_{rs} \frac{\partial^2 u}{\partial s \partial r} + c_{ss} \frac{\partial^2 u}{\partial s^2} + c_r \frac{\partial u}{\partial r} + c_s \frac{\partial u}{\partial s} + c_c u = f \text{ in } \Omega,$$

$$Bu := b_r \frac{\partial u}{\partial r} + b_s \frac{\partial u}{\partial s} + b_c u = g \text{ auf } \partial\Omega$$

In diesen Gleichungen hängen die Koeffizienten von den Ableitungen der Abbildung zwischen den Gittern ab. Nun kann die Lösung einer partiellen Differentialgleichung auf einem composite grid als die Lösung von Gleichungen auf einer Sequenz von Einheitsquadraten betrachtet werden. Dazu müssen die Lösungen auf den einzelnen Quadraten

mittels Interpolation auf dem Interpolationsrand gekoppelt werden. Die diskrete Lösung am Punkt (i, j) auf dem Gitter k heie $v_k(i, j)$. So folgen diskrete Approximationen an 3.5 und 3.6 in der Form

$$L_k v_k = f_k, \quad (3.7)$$

$$B_k v_k = g_k. \quad (3.8)$$

Ein Punkt (i, j, k) des Interpolationsrandes wird bei der Abbildungsmethode aus Punkten einer anderen Gitterkomponente k' interpoliert. Dazu ist es notwendig, die Position (i', j') des Punktes (i, j, k) im Gitter k' zu kennen. Durch Standardinterpolation ergibt sich v_k durch

$$v_k(i, j) = \sum_{i', j'} \alpha_{kk'}(i, j, i', j') v_{k'}(i', j'). \quad (3.9)$$

Die Gleichungen 3.7, 3.8 und 3.9 knnen zu einem linearen Gleichungssystem

$$\mathbf{A} \mathbf{v} = \mathbf{f} \quad (3.10)$$

zusammengefasst werden; darin wird \mathbf{v} als composite mesh function und \mathbf{A} als composite mesh operator bezeichnet.

Iterative Lser

Obwohl die Matrix \mathbf{A} des linearen Systems 3.10 im Allgemeinen nicht symmetrisch ist, lassen sich dennoch viele Standardverfahren zum iterativen Lsen des Systems anwenden. Jedoch stellt Linden (zu finden in Stben und Trottenberg [32]) heraus, dass Iterationen, basierend auf Schwartz' Methode, langsamer als konkurrierende Verfahren sind und stark von der Breite der berlappregion abhngen. Daher sei es sinnvoller, die Gleichungen 3.10 simultan zu lsen. Zu diesem Zweck stellen Henshaw und Chesshire in [28] eine Mehrgittermethode fr die Anwendung auf composite grids vor:

Das einmal erzeugte composite grid M kann durch Anwendung von Vergrberungsalgorithmen (siehe Stben und Trottenberg [32]), zum Beispiel der Standardvergrberung, zu einer Sequenz von Gittern $M^l, l = 1, 2, 3, \dots$ mit sukzessiv wachsender Maschenweite h erweitert werden. Der Einfachheit halber sei vorausgesetzt, dass die Anzahl der Gitterkomponenten auf jedem Level l gleich ist. Es bezeichne M_k^l die k -te Gitterkomponente des composite grids M^l auf dem Level l von M . Analog zu Gleichung 3.10 erhlt man die Gittergleichung

$$\mathbf{A}^l \mathbf{v}^l = \mathbf{f}^l \quad (3.11)$$

mit \mathbf{v}^l als composite mesh function. Nun lsst sich die Gleichung 3.11 mittels eines Mehrgitterlsers lsen, wie ihn zum Beispiel der Algorithmus 1 auf Seite 45 beschreibt. Fr die bei der Mehrgitteriteration verwendeten Gltter geben Henshaw und Chesshire [28]

zwei Möglichkeiten an:

$$S^l := \left\{ \begin{array}{l} \text{Glätte auf erster Gitterkomponente} \\ \text{Glätte auf zweiter Gitterkomponente} \\ \quad \vdots \\ \text{Glätte auf letzter Gitterkomponente} \\ \text{Interpoliere} \end{array} \right.$$

In dieser Variante wird zunächst auf allen Gitterkomponenten unabhängig voneinander geglättet und anschließend die Interpolationsbedingung ausgeführt. In der Praxis jedoch zeigt sich, dass das iterative Verfahren schneller konvergiert, wenn die Interpolation, welche die Gitterkomponenten untereinander koppelt, nach jedem einzelnen Glättungsschritt auf einer Gitterkomponente durchgeführt wird. Der entsprechende Glätter ist wie folgt konstruiert:

$$S^l := \left\{ \begin{array}{l} \text{Glätte auf erster Gitterkomponente} \\ \text{Interpoliere} \\ \text{Glätte auf zweiter Gitterkomponente} \\ \text{Interpoliere} \\ \quad \vdots \\ \text{Glätte auf letzter Gitterkomponente} \\ \text{Interpoliere} \end{array} \right.$$

Dabei ist der Aufwand der Interpolationen im Vergleich zum Aufwand der eigentlichen Glättungen gering. Als eigentliche Glätter auf den einzelnen Gitterkomponenten stehen alle herkömmlichen Glätter wie zum Beispiel Gauss-Seidel, Red-Black, Zebra line zur Auswahl (siehe auch Stüben und Trottenberg [32] und Trottenberg, C. Oosterlee, A. Schüller [56]).

3.4 Gebietszerlegungsmethoden

Ein Verfahren, das Gleichungen auf unterschiedlich diskretisierten Teilgebieten löst, ist die Gebietszerlegungsmethode. In Bezug auf Adaptivität lässt sie sich wie folgt einsetzen: Das gesamte Gebiet wird gemäß den Anforderungen an die lokale Geometrie und Diskretisierungsgenauigkeit derart in Teilgebiete zerlegt, dass jedes dieser Teilgebiete mit einem eigenen strukturierten Gitter der gewünschten Maschenweite überdeckt werden kann. Nun kann ein Randwertproblem, welches auf dem gesamten Gebiet definiert ist, in mehrere Randwertprobleme auf den einzelnen Teilgebieten mit den zugehörigen Gittern aufgeteilt und gelöst werden. Dabei sind die einzelnen Teilprobleme unabhängig

voneinander lösbar, wodurch sich Gebietszerlegungsmethoden hervorragend für die Implementierung auf Parallelrechnern eignen (siehe auch Smith, Bjørstad und Gropp [49]). Im Wesentlichen unterscheidet man zwei Arten von Gebietszerlegungsmethoden:

- Bei den sogenannten nicht überlappenden Methoden überdecken sich benachbarte Teilgebiete lediglich an den gemeinsamen Rändern. Spezielle Formen der nicht überlappenden Gebietszerlegungsmethoden sind die Mortar-Methoden (siehe Abschnitt 3.4.1). Vertiefende Erläuterungen zu nicht überlappenden Gebietszerlegungsmethoden geben Jinchao Xu und Jun Zou in [60].
- Zu den sogenannten überlappenden Gebietszerlegungsmethoden zählen die additive und multiplikative Schwarz-Methode (siehe Michael Griebel und Peter Oswald [26] sowie Andrea Toselli und Olof Widlund [54]), bei denen die benachbarten Regionen über den gemeinsamen Rand hinaus überlappen.

Eine allgemeine Einführung in Gebietszerlegungsmethoden geben Chan und Mathew in [14], weiterführende Informationen über Gebietszerlegungsmethoden liefern Quarteroni und Valli in [40] sowie Jinchao Xu in seinem sehr bekannten Artikel [59]. Der folgende Abschnitt beschreibt eine spezielle Form der nicht überlappenden Gebietszerlegungsmethoden – die Mortar-Methoden – und stellt heraus, inwieweit sich Mortar-Methoden für eine Erweiterung der Level-Set-Reinitialisierung in NaSt3DGPf eignen. Ferner skizziert der Abschnitt 3.4.3 die Konstruktion einer Mortar-Mimetischen-Finiten-Differenzen-Methode, die eine adaptive Diskretisierung basierend auf kartesischen Gitterkomponenten ermöglicht.

3.4.1 Mortar-Methoden

Auf den Gebietszerlegungsmethoden basierend, erlauben die Mortar-Methoden äußerst flexible Diskretisierungen dahingehend, dass für jedes Teilgebiet eine eigene Diskretisierung unabhängig von den anderen Teilgebieten gewählt werden kann. Dazu ist es nicht einmal notwendig, dass die Diskretisierungen an den Gebietsgrenzen zueinander konform sind, weshalb große Unterschiede zwischen den Maschenweiten und auch hängende Knoten erlaubt sind (siehe auch Abbildung 3.12). Mortar-Methoden kommen daher oft zum Einsatz, wenn sich das zu lösende Problem räumlich in zueinander nicht konforme Teilprobleme aufteilen lässt oder auf einer Kopplung von unterschiedlichen physikalischen Modellen beruht. Auch bei Kontaktproblemen, bei denen sich die Gitter verschieden diskretisierter Festkörper berühren oder gegeneinander verschieben, spielen Mortar-Methoden eine wichtige Rolle. Ferner können oft heterogene Probleme in homogene zerlegt und diese nach getrennter Diskretisierung mittels Mortar-Methoden wieder gekoppelt werden.

Das Ziel der Betrachtung von Mortar-Methoden in dieser Arbeit ist eine Antwort auf die Frage nach der Anwendbarkeit dieses Verfahrens für eine verbesserte Level-Set-

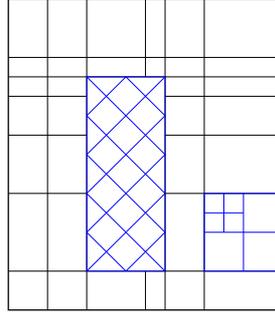


Abbildung 3.12: Eine Gebietszerlegung mit unterschiedlichen, nichtkonformen Diskretisierungen (schwarz, blau) auf den Teilgebieten.

Methode. Auf der Suche nach geeigneten Diskretisierungen beziehungsweise Gittertypen als Grundlage einer Erweiterung oder Erneuerung des konventionellen Level-Set-Algorithmus in NaSt3DGPF bieten Mortar-Methoden die Aussicht, einerseits weiterhin mit kartesischen Gittern arbeiten zu können und andererseits Diskretisierungsfehler und numerische Instabilitäten durch Adaptivität – etwa durch octrees (siehe Abschnitt 3.1) oder Triangulierungen – zu reduzieren und gleichzeitig eine praktikable Methode anzuwenden, um größere Irregularitäten und hängende Knoten leicht handhaben zu können. Dazu wird im folgenden Abschnitt eine kurze Einführung in die Mortar-Technik mit finiten Elementen gegeben und anschließend gezeigt, wie diese in Form von mimetischen finiten Differenzen an die Bedürfnisse der Level-Set-Methode innerhalb des Programms NaSt3DGPF angepasst werden können.

3.4.2 Mortar-Finite-Elemente-Methode

Eine detaillierte Einführung in Mortar-Methoden – genauer Mortar-Finite-Element-Methode – gibt Wohlmuth in [58] anhand des Modellproblems:

$$\begin{aligned} -\nabla \cdot (a \nabla u) + bu &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{auf } \partial\Omega, \end{aligned} \tag{3.12}$$

wobei a eine positiv definite Matrix, $a_{ij} \in L^\infty(\Omega)$, $1 \leq i, j \leq d$, $f \in L^2(\Omega)$, $0 \leq b \in L^\infty(\Omega)$ und $\Omega \subset \mathbb{R}^d$ mit $d \in \{2, 3\}$ ein beschränktes polygonales Gebiet ist. Ferner soll gelten, dass die Koeffizienten a_{ij} und b auf den Elementen der Triangulierung konstant sind. Sei Ω nun in K nicht überlappende Teilgebiete Ω_k zerlegt:

$$\bar{\Omega} = \bigcup_{k=1}^K \bar{\Omega}_k. \tag{3.13}$$

Diese Zerlegung sei geometrisch konform; das heißt, dass der Schnitt zweier Teilgebiete $\partial\Omega_l \cap \partial\Omega_k$ mit $k \neq l$ entweder leer, ein Knoten, eine gemeinsame Kante oder Fläche ist.

Ist dieser Schnitt eine $(d - 1)$ -dimensionale Hyperfläche, so wird er Interface genannt. Jedes Teilgebiet erhält eine Triangulierung $\tau_{k;h_k}$ mit der maximalen Maschenweite h_k . Nun wird ein Finite-Elemente-Raum der konformen P_{n_k} -Elemente auf Ω_k mit $\tau_{k;h_k}$ und $n_k \geq 1$ eingeführt, die den homogenen Dirichlet-Randbedingungen auf $\partial\Omega \cap \partial\Omega_k$ genügen. Dieser Raum werde mit $X_{h_k;n_k}$ bezeichnet. Ebenfalls werden die Interfaces $\partial\Omega_l \cap \partial\Omega_k$ mit einer $(d - 1)$ -dimensionalen Triangulierung versehen. Diese leitet sich entweder von $\tau_{k;h_k}$ oder $\tau_{k;h_l}$ ab, die im Allgemeinen nicht miteinander koinzidieren. Für jedes dieser Interfaces γ_m , $1 \leq m \leq M$ existieren Indizes $1 \leq l < k \leq K$ mit $\bar{\gamma}_m = \partial\Omega_l \cap \partial\Omega_k$. Die Triangulierungen $\mathcal{S}_{m;h_m}$ der Interfaces γ_m sind Kanten in zwei beziehungsweise Flächen in drei Dimensionen von $\tau_{k;h_l}$ oder $\tau_{k;h_k}$; per Definition stammt $\mathcal{S}_{m;h_m}$ von der sogenannten Non-Mortar-Seite ab. Die gegenüberliegende Seite wird daher auch Mortar-Seite genannt. Ferner bezeichnen $\Omega_{n(m)}$ und $\Omega_{\bar{n}(m)}$ die Teilgebiete auf der Non- beziehungsweise der Mortar-Seite von γ_m . Im Allgemeinen sind $\tau_{k;h_l}$ und $\tau_{k;h_k}$ nicht zueinander konform, wobei grob zwei Fälle unterschieden werden (siehe auch Abbildung 3.13):

- 1) $h_l \approx h_k$: Gitter sind gegeneinander verschoben; tritt häufig bei zeitabhängigen und Kontaktproblemen auf,
- 2) $h_l \ll h_k$: häufig bei stark heterogenen Problemen.

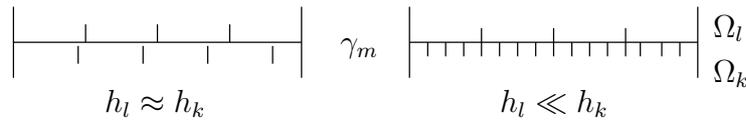


Abbildung 3.13: Unterschiedliche, nichtkonforme Triangulierungen an einem Interface in zwei Dimensionen.

In beiden Fällen liefert jedoch die Mortar-Finite-Elemente-Methode eine optimale A-priori-Schranke für den Diskretisierungsfehler in der Energienorm [12, 13]. Beachtenswert dabei ist, dass die Konstante in der Fehlerabschätzung in zwei Dimensionen weder von dem Verhältnis der Maschenweiten h_l/h_k noch von dem Grad der Verschiebung der Gitter abhängt. In drei Dimensionen hingegen geht h_l/h_k in die Abschätzung mit ein [4, 7]. Es stehen im Wesentlichen zwei Ansätze zur Verfügung, um eine Approximation u_h als Lösung eines diskreten Variationsproblems zu erhalten. Der erste ist von C. Bernardi, Y. Madai und A.T. Patera in [12, 13] eingeführt und definiert das Variationsproblem auf V_h , einem Unterraum des Produktraumes der Elemente, der die schwache Kontinuitätsbedingung auf dem Interface erfüllt. V_h ist somit gegeben durch

$$V_h := \{v \in L^2(\Omega) \mid v|_{\Omega_k} \in X_{h_k;n_k}, 1 \leq k \leq K, \int_{\gamma_m} [v]\mu d\sigma = 0, \mu \in M_{h_m}(\gamma_m), 1 \leq m \leq M\},$$

wobei der Raum der Testfunktionen $M_{h_m}(\gamma_m)$ – das ist der Raum der Lagrange-Multiplikatoren – durch

$$M_{h_m}(\gamma_m) := \{ \mu \in L^2(\gamma_m) \mid \mu = w|_{\gamma_m}, w \in X_{h_{n(m)}; n_{n(m)}}, \\ \mu|_e \in P_{n_m-1}(e), \text{ falls } e \in \mathcal{S}_{m; h_m} \text{ einen Endpunkt von } \gamma_m \text{ enthält} \} \quad (3.14)$$

mit $n_m := n_{n(m)}$ gegeben ist. Im dreidimensionalen Fall muss dieser Raum in einer Umgebung von $\partial\gamma_m$ modifiziert werden; diese Einführung beschränkt sich jedoch auf den zweidimensionalen Fall. $M_{h_m}(\gamma_m)$ kann als modifizierter Spurraum der Kodimension 2 gesehen werden, der mit der Triangulierung $\tau_{n(m); h_{n(m)}}$ assoziiert ist und somit ausschließlich von der Non-Mortar-Seite abhängt. M_h bezeichne das globale Produkt $\prod_{m=1}^M M_{h_m}(\gamma_m)$ als Unterraum von $L^2(\mathcal{S})$ mit $\bar{\mathcal{S}} := \cup_{m=1}^M \bar{\gamma}_m$.

Die nichtkonforme Formulierung der Mortar-Finite-Elemente-Methode als Variationsproblem lautet nun: Finde ein $u_h \in V_h$, so dass

$$a(u_h, v_h) = (f, v_h)_0, \quad v_h \in V_h; \quad (3.15)$$

(siehe auch [12, 13]). In Gleichung 3.15 ist die Bilinearform $a(\cdot, \cdot)$ definiert als

$$a(v, w) := \sum_{k=1}^K \int_{\Omega_k} a \nabla v \cdot \nabla w + b v w \, dx, \quad v, w \in \prod_{k=1}^K H^1(\Omega_k).$$

C. Bernardi, Y. Madai und A.T. Patera zeigen in [12, 13] die gleichmäßige Elliptizität von $a(\cdot, \cdot)$ auf $V_h \times V_h$ und die Bestapproximation und Konsistenz von optimaler Ordnung. Ferner existiert eine eindeutige Lösung von 3.15. Ausreichende Glattheit und H^2 -Regularität der schwachen Lösung von 3.12 vorausgesetzt, gelten [12, 13, 8] folgende Abschätzungen für den Diskretisierungsfehler in der gebrochenen H^1 -Norm und in der L^2 -Norm:

$$\|u - u_h\|_1^2 \leq C \sum_{k=1}^K h_k^{2n_k} |u|_{n_k+1; \Omega_k}^2 \\ \|u - u_h\|_0^2 \leq h^2 C \sum_{k=1}^K h_k^{2n_k} |u|_{n_k+1; \Omega_k}^2$$

mit $\|v\|_1^2 := \sum_{k=1}^K \|v\|_{1; \Omega_k}^2$, $v \in \prod_{k=1}^K H^1(\Omega_k)$, wobei $\|\cdot\|_{s; D}$ und $|\cdot|_{s; D}$ für die H^s -Norm beziehungsweise ihre Seminorm auf der offenen Menge $D \subset \Omega$ stehen. Dabei sorgen die Einschränkungen auf den Interfaces dafür, dass der Konsistenzfehler mindestens so gut ist wie die Summe der Bestapproximationsfehler auf den Teilgebieten. Wird hingegen V_h gegen den uneingeschränkten Produktraum

$$X_h := \{ v \in L^2(\Omega) \mid v|_{\Omega_k} \in X_{h_k; n_k}, 1 \leq k \leq K \}$$

ausgetauscht, so ist der Konsistenzfehler nicht mehr in der Maschenweite beschränkt.

Ein zweiter Ansatz wurde von Belgacem [3] eingeführt und von Wohlmuth [57] weitergehend untersucht. Der wesentliche Unterschied zum nichtkonformen Ansatz besteht darin, dass die Randbedingungen auf dem Interface nicht durch den globalen Raum V_h selbst erfüllt werden, sondern mittels Lagrange-Multiplikatoren aus 3.14 realisiert werden. Ein anfängliches Minimierungsproblem führt zu folgendem Sattelpunktsproblem: Finde $(u_h, \lambda_h) \in (X_h, M_h)$, so dass

$$\begin{aligned} a(u_h, v) + b(v, \lambda_h) &= (f, v)_0, & v \in X_h \\ b(u_h, \mu) &= 0, & \mu \in M_h, \end{aligned} \quad (3.16)$$

mit der über die duale Paarung auf dem Interface definierten Bilinearform

$$b(v, \mu) := \sum_{m=1}^M \langle [v], \mu \rangle_{\gamma_m}, \quad v \in \prod_{k=1}^K H^1(\Omega_k), \quad \mu \in \prod_{m=1}^M \left(H^{\frac{1}{2}}(\gamma_m) \right)'$$

und $[v] := v|_{\Omega_{n(m)}} - v|_{\Omega_{\bar{n}(m)}}$. In diesen Ausdrücken steht $\left(H^{\frac{1}{2}}(\gamma_m) \right)'$ für den Dualraum von $H^{\frac{1}{2}}(\gamma_m)$ und $\langle \cdot, \cdot \rangle$ für die duale Paarung. Offensichtlich stimmen die Lösungen des positiv definiten Variationsproblems 3.15 und der ersten Komponente des Sattelpunktsproblems 3.16 überein. Der diskrete Lagrange-Multiplikator λ_h hingegen approximiert den Fluss. Für ihn ergeben sich A-priori-Fehlerabschätzungen für $\lambda - \lambda_h$ aus der Approximationseigenschaft von M_h . F. Ben Belgacem hat in [3] solche A-priori-Abschätzungen in der $H_{00}^{1/2}$ -Dualnorm ermittelt; diese Norm ist definiert durch

$$\|\mu\|_{(H_{00}^{\frac{1}{2}}(\mathcal{S}))'}^2 := \sum_{m=1}^M \|\mu\|_{(H_{00}^{\frac{1}{2}}(\gamma_m))'}^2 := \sum_{m=1}^M \sup_{v \in H_{00}^{\frac{1}{2}}(\gamma_m)} \frac{\langle v, \mu \rangle_{\gamma_m}^2}{\|v\|_{H_{00}^{\frac{1}{2}}(\gamma_m)}^2}$$

mit $\mu \in (H_{00}^{1/2}(\mathcal{S}))'$ und $(H_{00}^{1/2}(\mathcal{S}))' := \prod_{m=1}^M (H_{00}^{1/2}(\gamma_m))'$. Die entsprechende Fehlerabschätzung lautet schließlich

$$\|\lambda - \lambda_h\|_{M'}^2 \leq C \sum_{k=1}^K h_k^{2n_k} |u|_{1+n_k; \Omega_k}^2$$

3.4.3 Mortar-Mimetische-Finite-Differenzen-Methode

Nach dem Überblick über das Prinzip der Mortar-Methoden soll nun die Level-Set-Reinitialisierung in NaSt3DGPF auf möglichst effiziente Weise verbessert werden. Möglich wäre in Anlehnung an Berndt, Lipnikov und Shashkov [6] folgende Vorgehensweise: Der Diskretisierung liegt ein kartesisches Gitter zugrunde, welches lokal mittels octrees verfeinert wird (siehe hierzu Abschnitte 3 und 3.1) ab Seite 33). Verzichtet man auf eine

1-Irregularität, erspart man sich das Einfügen zusätzlicher Freiheitsgrade, die der Regularität dienen, aber nicht direkt zur Auflösung des Problems beitragen; dieser Vorteil geht jedoch mit einer hochgradigen Irregularität und einer Vielzahl an hängenden Knoten einher. Beispielhaft werden hier Diskretisierungen mit zwei festen Maschenweiten behandelt (vergleiche Abbildung 3.14); das im Folgenden vorgestellte Schema kann jedoch in analoger Weise auf beliebige Gitter mit Octree-Struktur erweitert werden. Gerade an diesen inkonsistenten Stellen soll mit Mortar-Methoden ein konsistentes Diskretisierungsschema aufgestellt werden. Betrachte man das Gebiet als Vereinigung zweier nicht überlappender Teilgebiete, deren einer Teil mit Viereckselementen der groben Maschenweite H trianguliert ist. Das andere Teilgebiet sei mit Viereckselementen der Maschenweite $h = H/2^v$ mit der Irregularität v trianguliert (siehe Abbildung 3.14). Innerhalb der beiden Teilge-

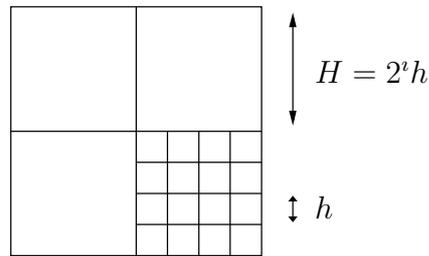


Abbildung 3.14: Ein einfacher quadtree in zwei Dimensionen. Alle Blattzellen haben entweder die Maschenweite H oder h .

bierte wird ein gewöhnliches Differenzen-Schema für die Diskretisierung des Gradienten $\nabla\phi$ der Level-Set-Funktion ϕ verwendet und auf dem Interface zwischen den Teilgebieten mittels Lagrange-Multiplikatoren die schwache Stetigkeit von ϕ herbeigeführt. In [6] haben Berndt, Lipnikov, Shashkov, Wheeler und Yotov eine Mortar-Methode mit mimetischen finiten Differenzen, die sogenannte Mortar-Mimetic-Finite-Difference-Method oder kurz MFD, für ein Modellproblem vorgestellt. Dieses Verfahren wird nun analog auf die Hamilton-Jacobi-Gleichung 2.16 der Level-Set-Reinitialisierung angewendet, wobei das Hauptaugenmerk auf dem Finite-Differenzen-Schema des Gradienten $\nabla\phi$ und ganz besonders auf der Lösung des Stetigkeitsproblems auf dem Interface liegen wird.

Sei das Gebiet Ω nun zerlegt in die nicht überlappenden Teilgebiete Ω_H und Ω_h , also $\overline{\Omega} = \overline{\Omega}_H \cup \overline{\Omega}_h$. Ferner sei $\Gamma_{H,h} = \partial\Omega_H \cap \partial\Omega_h$ das Interface zwischen den Gebieten und τ_H und τ_h seien die Triangulierungen der Teilgebiete. Auf dem Interface $\Gamma_{H,h}$ seien nun drei unterschiedliche Triangulierungen definiert: $\tau_H|_\Gamma$ und $\tau_h|_\Gamma$ erhält man jeweils durch Assoziation von τ_H und τ_h mit $\Gamma_{H,h}$; λ sei eine eigenständige Triangulierung von $\Gamma_{H,h}$ und muss nicht zwingend mit $\tau_H|_\Gamma$ oder $\tau_h|_\Gamma$ übereinstimmen. λ sei jedoch in diesem Fall über die Triangulierung der Seite mit der größeren Maschenweite H definiert und mit $\lambda := \tau_H|_\Gamma$ bezeichnet. Damit auch auf dem Interface schwache Stetigkeit von ϕ erzwungen werden kann, wird der Raum Λ der Lagrange-Multiplikatoren auf $\Gamma_{H,h}$ eingeführt, dessen

Finite-Elemente-Unterraum Λ^d mit der Triangulierung λ assoziiert ist. Λ^d sei daher als Raum der unstetigen, stückweise linearen Funktionen definiert, die für jeden Knoten der Triangulierung λ über mehrere Freiheitsgrade – zwei Knoten in zwei Dimensionen und drei Knoten in drei Dimensionen – verfügen.

Nun werden die Freiheitsgrade der skalaren Funktion ϕ in gewohnter Weise in den Zellenmittelpunkten von τ_H und τ_h und zusätzlich in den Mittelpunkten der $(d-1)$ -dimensionalen Triangulierung $\tau_h|_\Gamma$ und $\tau_H|_\Gamma$ positioniert, was die Räume $X_{\tau_h|_\Gamma}$ und $X_{\tau_H|_\Gamma}$ liefert. Die Abbildung 3.15a zeigt ein Beispiel in zwei Dimensionen. Ohne Be-

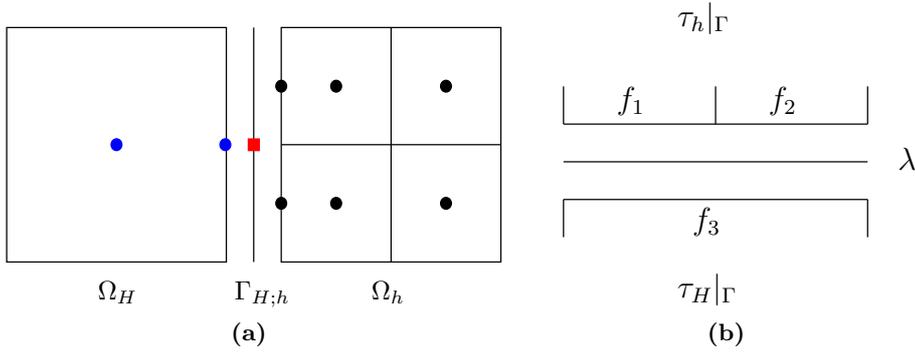


Abbildung 3.15: (a) Die Freiheitsgrade der beiden Teilgebiete (• und •) liegen in den Zellmittelpunkten und den Mittelpunkten der mit dem Interface $\Gamma_{H,h}$ assoziierten Triangulierungen $\tau_h|_\Gamma$ und $\tau_H|_\Gamma$. Der Mortar-Raum Λ definiert zusätzliche Freiheitsgrade (■) auf den Mittelpunkten der Triangulierung $\lambda := \tau_h|_\Gamma$. (b) Triangulierungen am Interface: $\tau_h|_\Gamma$ besteht aus den Elementen f_1 und f_2 , $\tau_H|_\Gamma$ aus dem Element f_3 .

schränkung der Allgemeinheit wird angenommen, dass $\tau_h|_\Gamma = \{f_1, f_2\}$ und $\tau_H|_\Gamma = \{f_3\}$, wie in Abbildung 3.15b dargestellt. Eine schwache Formulierung der Stetigkeit der Level-Set-Funktion ϕ auf dem Interface lautet nun

$$\langle [\phi_X], \mu \rangle_{\Gamma_{H,h}} = 0 \quad \forall \mu \in \Lambda^d, \quad (3.17)$$

wobei ϕ_X das Finite-Elemente-Pendant von $\phi \in X_{\tau_h|_\Gamma} \cup X_{\tau_H|_\Gamma}$ und $[\phi_X]$ der Sprung von ϕ_X auf dem Interface ist. Nun werden lineare Projektionsoperatoren $R_h : \Lambda^d \rightarrow X_{\tau_h|_\Gamma}^d$ und $R_H : \Lambda^d \rightarrow X_{\tau_H|_\Gamma}^d$ eingeführt; diese werden implizit durch

$$\begin{aligned} \langle [\phi_X], \mu \rangle_{\Gamma_{H,h}} &= \langle \phi_h, R_h \mu \rangle_{\Gamma_{H,h}} - \langle \phi_H, R_H \mu \rangle_{\Gamma_{H,h}} \\ &\quad \forall \mu \in \Lambda^d, \phi_h \in X_{\tau_h|_\Gamma}, \phi_H \in X_{\tau_H|_\Gamma}, \phi \in X_{\tau_h|_\Gamma} \cup X_{\tau_H|_\Gamma} \end{aligned} \quad (3.18)$$

definiert. Konkret können nun die Projektoren mit

$$R_h = \frac{1}{2|f_3|} \begin{bmatrix} |f_1| & |f_1| + 2|f_2| \\ 2|f_1| + |f_2| & |f_2| \end{bmatrix} \quad \text{und} \quad R_H = \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix} \quad (3.19)$$

angegeben werden. Analog dazu können durch Lösen eines Interpolationsproblems auch Projektoren für den Fall größerer Irregularität und sogar für den dreidimensionalen Fall ausgerechnet werden. Sei nun die Irregularität $\iota = 2$ und $\tau_h|_\Gamma = \{f_1, f_2, f_3, f_4\}$ und $\tau_H|_\Gamma = \{f_5\}$, dann folgt in zwei Dimensionen zum Beispiel

$$R_h = \frac{1}{2|f_5|} \begin{bmatrix} |f_1| & |f_1| + 2(|f_2| + |f_3| + |f_4|) \\ 2|f_1| + |f_2| & |f_2| + 2(|f_3| + |f_4|) \\ 2(|f_1| + |f_2|) + |f_3| & |f_3| + 2|f_4| \\ 2(|f_1| + |f_2| + |f_3|) + |f_4| & |f_4| \end{bmatrix} \quad \text{und} \quad R_H = \frac{1}{2} \begin{bmatrix} 1 & 1 \end{bmatrix}. \quad (3.20)$$

In diesem Abschnitt wird deutlich, dass die Mortar-Mimetische-Finite-Differenzen-Methode lediglich eine andere Formulierung für ein Interpolationsproblem bei finiten Differenzen mit hängenden Knoten ist (vergleiche Abschnitt 3.2).

Kapitel 4

Eine Implementierung der adaptiven Level-Set-Methode auf kartesischen Gittern

In diesem Kapitel wird zunächst auf die motivierende Problemstellung aus der Einleitung detaillierter eingegangen. Es wird dabei im zweiten Abschnitt der Kern des Problems herausgestellt und ein Ansatz zur Entwicklung einer verbesserten Level-Set-Methode vorgestellt. Der dritte Abschnitt beschreibt die konkrete Vorgehensweise bei der Entwicklung der adaptiven Level-Set-Methode auf kartesischen Gittern und ihre Implementierung in NaSt3DGPF. Alle Aspekte, die die Numerik betreffen, werden im letzten Abschnitt erläutert.

4.1 Numerische Diffusion bei Level-Set-Methoden

Die Einleitung dieser Arbeit führt in die Problematik der konventionellen Level-Set-Methode im Zusammenhang mit Zweiphasenströmungsproblemen ein. Da sich diese Arbeit mit der Entwicklung eines robusteren Verfahrens befasst, ist es nötig, einen tieferen Blick in die Numerik von Level-Set-Methoden zu werfen. Dieser Abschnitt erläutert die bekanntesten Problematiken und deren Lösungsansätze, wie sie bereits in vielen Programmen implementiert werden. Ausführlich geht Croce [18] in Kapitel 5 darauf ein.

Während das Prinzip der Level-Set-Methode ein hohes Maß an Effizienz bei zweiphasigen Strömungssimulationen verspricht, bedarf die numerische Behandlung dieser Methode jedoch viel Sorgfalt, um entstehende numerische Fehler kontrollieren zu können.

Ein Hauptproblem der Level-Set-Methode liegt in der Reinitialisierung der Level-Set-Funktion. Ihre numerische Ungenauigkeit zeigt sich in einer Verschiebung der Nullkontur - also der in der Strömungssimulation eigentlich zu berechnenden freien Oberfläche. Dieser Effekt bewirkt daher einen künstlichen Massenverlust einer Phase (beziehungsweise Massenzuwachs der anderen Phase) in einer Zweiphasensimulation und wird oft auch als numerische Diffusion bezeichnet. Im Abschnitt 2.4 des Grundlagenkapitels wird der Prozess der Reinitialisierung erklärt. Anhand der Gleichung 2.16 in Definition 2.4 und Gleichung 2.15 (siehe Seite 30) wird deutlich, dass die Reinitialisierung im Analytischen exakt sein muss. Probleme treten erst im Zusammenhang mit einer Diskretisierung auf.

Das ist einerseits der unvermeidbare Diskretisierungsfehler, dem weitestgehend durch die Wahl von diskreten Operatoren höherer Ordnung sowohl für die räumlichen als auch die zeitlichen Differentialoperatoren der Hamilton-Jacobi-Gleichung begegnet wird. In seiner Arbeit [18] zeigt Croce, dass WENO-Schemata für die räumlichen Operatoren beziehungsweise Runge-Kutta-Verfahren dritter Ordnung für die Zeitdiskretisierung adäquate Mittel sind, den Diskretisierungsfehler der Reinitialisierung erheblich zu reduzieren.

4.1.1 Das Problem am Beispiel von Strömungssimulationen

In den vorhergehenden Abschnitten wurde das Phänomen der numerischen Diffusion bei Level-Set-Methoden theoretisch erläutert. Die numerische Diffusion kann auf Diskretisierungsfehler und auf Instabilitäten des numerischen Verfahrens zurückgeführt werden. Bekannte Lösungsansätze reduzieren die künstliche Verschiebung der Nullkontur der Level-Set-Funktion auf die Distanz von maximal einer Maschenweite, indem Finite-Differenzen-Schemata und Zeitschrittverfahren höherer Ordnung sowie eine spezielle Glättung der Signumfunktion beziehungsweise das subcell fixing angewendet werden. Da die Level-Set-Methode zumeist ein Hilfsmittel in einer weiteren Anwendung ist, wird in diesem Abschnitt das Problem der numerischen Diffusion im Zusammenhang mit zweiphasigen Strömungssimulationen betrachtet, wobei die aktuelle Implementation von NaSt3DGPF zugrunde liegt.

Die Maschenweite H als zentraler Parameter

Das Simulationsprogramm NaSt3DGPF ist ein Löser für strömungsmechanische Probleme auf Grundlage der inkompressiblen Navier-Stokes-Gleichungen. Die Diskretisierung basiert auf lokal anisotropen, kartesischen Gittern zusammen mit finiten Differenzen für die Ortsdiskretisierung; die freie Oberfläche bei zweiphasigen Strömungen wird durch die Nullkontur einer Level-Set-Funktion repräsentiert. Zentraler Parameter dieser Diskretisierung ist die Maschenweite H . Jeder diskrete Zeitschritt im Strömungslöser ist mit einer festen Sequenz von Instruktionen verbunden. So erfolgt nach dem Berechnen des neuen Geschwindigkeitsfeldes der Transport der Level-Set-Funktion (siehe Gleichung 2.15 auf Seite 29) und erzwingt damit die Reinitialisierung, um wieder eine Abstandsfunktion zu erhalten. Das bedeutet, dass sich in jedem Zeitschritt des Strömungslösers die freie Oberfläche um die Weite H verschieben kann. Dieser Effekt wird begünstigt durch starke Krümmungen, nicht differenzierbare (Lipschitz-stetige) Stellen und Topologieveränderungen der freien Oberfläche. Die folgenden Abbildungen verdeutlichen exemplarisch die Probleme: Besitzt die freie Oberfläche wie der in Abbildung 4.1 gezeigte Würfel an den Kanten und Ecken starke oder gar singuläre Krümmungen, führt die künstliche Verschiebung der Nullkontur im Level-Set-Algorithmus zum Ausglätten der Kanten. Ebenso zeigt die Abbildung 4.2 eine Störung an den Lipschitz-stetigen Stellen, die wieder mit einer künstlichen Glättung dieser Stellen einhergeht. Ein anderes Phänomen tritt bei dünnen

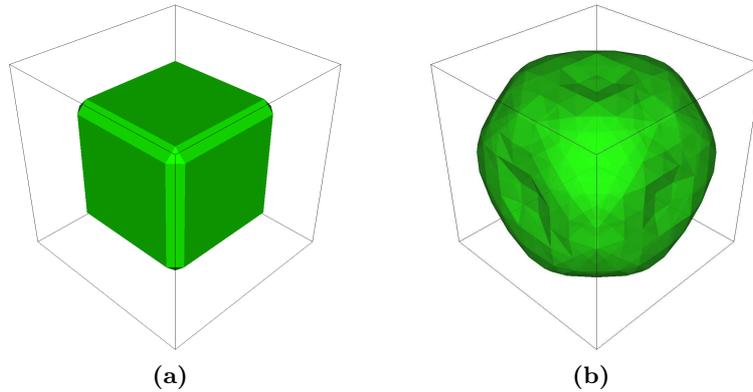


Abbildung 4.1: Durch zahlreiche Iterationen der Reinitialisierung beginnen die Ecken und Kanten des Würfels aufzuweichen.

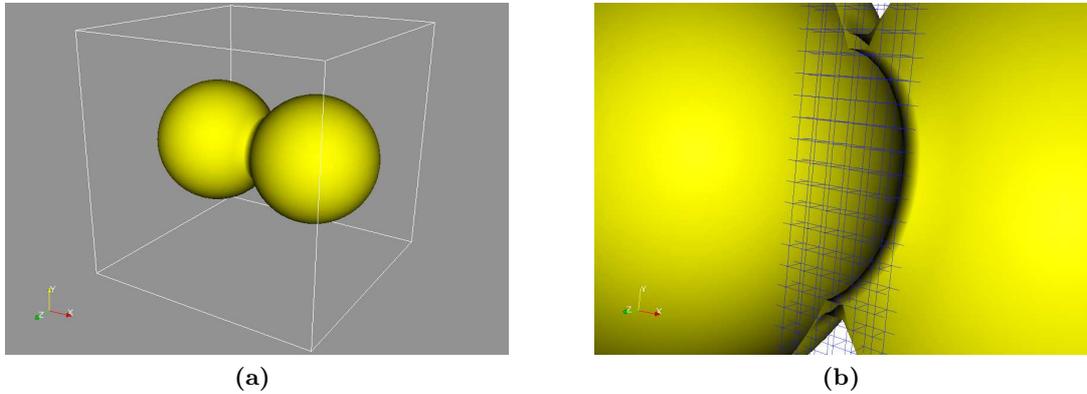


Abbildung 4.2: Zwei Sphären durchdringen sich: (a) Die freie Oberfläche hat an dem Schnittkreis eine Lipschitz-stetige (aber nicht differenzierbare) Stelle. (b) Schon nach wenigen Iterationen der Reinitialisierung wird die freie Oberfläche gestört, so dass sich der ursprüngliche Schnittkreis an die nach außen hin nächstgelegenen Gitterpunkte anschmiegt. Zu sehen ist ein Schnitt der freien Oberfläche vor der Reinitialisierung und nach mehreren Iterationen. Zur Illustration ist ein Teil des Gitters dargestellt.

Filmen auf (siehe Abbildung 4.3). Hier kann eine künstliche Verschiebung der freien Oberfläche zu einem Abriss des Filmes führen (vergleiche Abbildungen 4.3a und 4.3b). In diesem Fall wird die Geometrie und Topologie der freien Oberfläche derart zerstört, dass die Strömungssimulation keine Rückschlüsse mehr auf die Realität bieten kann und unbrauchbar wird. Besonders erwähnenswert an dieser Stelle ist die Tatsache, dass die Auflösung des numerischen Gitters durchaus fein genug sein kann, um die geometrischen Strukturen und die dünnen Filme darstellen zu können. Auch kann die gegebene Auflösung zum adäquaten Lösen der Strömungsgleichungen hinreichend fein sein. Dennoch

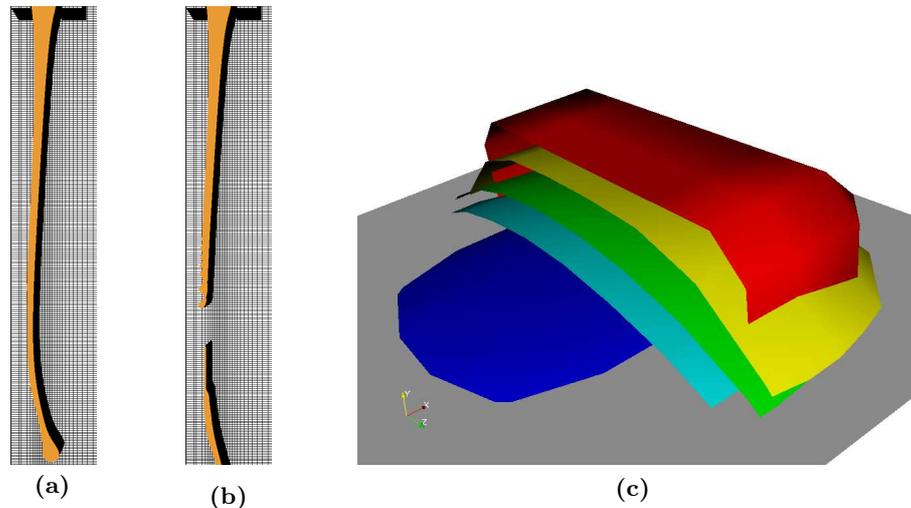


Abbildung 4.3: Ein dünner Fluidfilm (a) beginnt zu reißen (b). Bild (c): Ein Fluid (rot) kollabiert unter Schwerkraft. Nach mehreren Zeitschritten ist durch die ungünstige Verschiebung der freien Oberfläche ein großer Teil der ursprünglichen Fluidmenge verschwunden. Zu sehen ist die freie Oberfläche des Fluids zu verschiedenen Zeitschritten (rot, gelb, grün, cyan, blau) im Schnittbild.

kann das Scheitern der Simulation einzig an dem Level-Set-Algorithmus liegen. Das gleiche Problem zeigt auch Abbildung 4.3c; eine Fluidmenge in ihrer ursprünglichen Form als Quader wird unter Gravitation auf den Boden gedrückt, wobei der Quader kollabiert und sich auszubreiten beginnt. Physikalisch korrekt ist eine gleichmäßige Verteilung des Fluids auf dem Boden. Statt dessen verschwindet scheinbar die gesamte Masse. Ferner verdeutlicht der in Abbildung 4.4 gezeigte Befüllungsprozess, dass eine künstliche Verschiebung der freien Oberfläche die Ausbreitung des Flüssigkeitsfilms auf dem Boden des Gefäßes verhindern kann: Das Gefäß füllt sich nicht.

4.1.2 Klassische Lösungsansätze

T.Y. Hou hat beobachtet, dass die numerische Diffusion während der Reinitialisierung durch eine Verschiebung der Nullkontur in Normalenrichtung bewirkt wird, deren Geschwindigkeit proportional zur Krümmung dieser Kontur ist. Diese Beobachtung von T.Y. Hou hat Chang, Hou, Merriman und Osher in [15] veranlasst, der eigentlichen Reinitialisierung einen zweiten Schritt hinzuzufügen, der den Massenverlust wieder rückgängig macht. Zu diesem Zweck wird im zweiten Schritt eine Reinitialisierungsgleichung

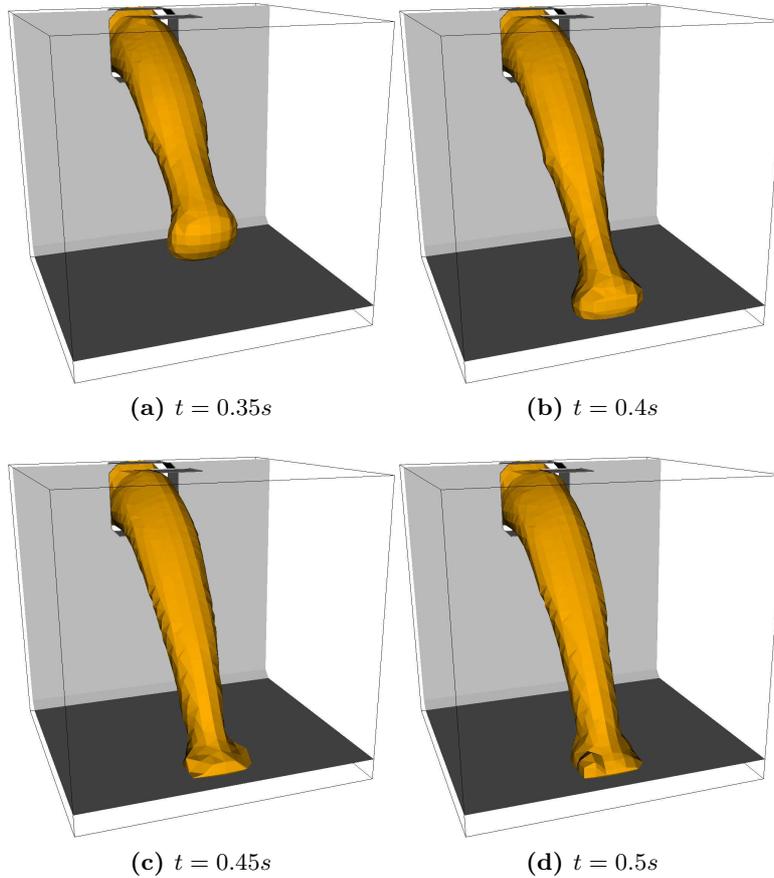


Abbildung 4.4: Ein Bassin wird mit einem Fluid befüllt. Zu sehen ist die Ausbreitung des Fluids zu bestimmten Zeitpunkten: In (a) hat der Strahl noch keinen Kontakt mit den Wänden; in (b) berührt die freie Oberfläche den Boden des Bassins. Das Auftreffen des Fluids bildet einen dünnen Film, der durch die künstliche Verschiebung der freien Oberfläche verschwindet (siehe (c) und (d)), ehe er sich ausbreiten und das Bassin befüllen kann.

vom gestörten Hamilton-Jakobi-Typ

$$\frac{\partial}{\partial \tau} d + (A_0 - A(\tau)) (-P + \kappa) |\nabla d| = 0 \tag{4.1}$$

$$d(x, \tau = 0) = \phi(x, t)$$

durchgeführt. Dabei ist A_0 die Masse, die vor der ersten Reinitialisierung von der Nullkontur definiert worden war, und $A(\tau)$ die aktuelle Masse, definiert durch $d(x, \tau)$. Ferner ist κ die Krümmung der Konturen, also $\kappa = \nabla \cdot \nabla d$, und P ist eine in [15] definierte, positive Konstante, die zur Stabilisierung des zweiten Reinitialisierungsprozesses in 4.1

beitragen soll. Wie auch schon bei der eigentlichen Reinitialisierung 2.16 wird in 4.1 die Iteration in der künstlichen Zeit τ bis zu einem näherungsweise stationären Zustand durchgeführt. Dieser ist erreicht, wenn $(A_0 - A(\tau))$ erfüllt ist. Dies bedeutet, dass die durch die Nullkontur von $d(x)$ gegebene Masse nach der zweiten Reinitialisierung 4.1 theoretisch bis auf Maschinengenauigkeit wiederhergestellt werden kann. Eine konkrete Implementierung dieses Verfahrens hat Croce in [18] durchgeführt. Als Diskretisierungsschemata verwendet Croce WENO fünfter Ordnung für den Ort und Runge-Kutta dritter Ordnung in der Zeit. Er stellt fest, dass sich dieses Verfahren im Einsatz mit dem Simulationsprogramm NaSt3DGPF instabil verhält, sobald die freie Oberfläche eine stark variierende Krümmung aufweist. Im Rahmen dieser Arbeit sollen aber gerade die durch diese Topologien verursachte numerische Diffusion im Vordergrund stehen. Daher findet der oben vorgestellte Lösungsansatz im Folgenden keine Verwendung mehr.

Subcell fixing

Der oben genannte Ansatz beruht auf einer Beobachtung und bewirkt, dass die von der Reinitialisierung zuvor verursachten Störungen wieder korrigiert werden. Weitaus elegantere Methoden setzen sich zum Ziel, diese Störungen der Nullkontur gar nicht erst entstehen zu lassen. Eine spezielle Konstruktion von Upwind-Schemata zur verbesserten Erhaltung der Nullkontur stellen G. Russo and P. Smereka in [44] vor. Sie modifizieren das von M. Sussman, P. Smereka, und S. Osher in [53] dargestellte Verfahren und beschreiben die Entwicklung eines „echten“ Upwind-Schemas: Anhand eines eindimensionalen Beispiels kann verdeutlicht werden, dass ein herkömmliches Upwind-Verfahren für die Diskretisierung der räumlichen Operatoren der Reinitialisierung

$$\phi_i^{n+1} = \phi_i^n + \Delta t S(\phi_i^0) G_i(\phi), \quad (4.2)$$

wobei

$$G_i(\phi) = \begin{cases} \max(|a_+|, |b_-|) - 1, & \text{falls } \phi_i^0 > 0 \\ \max(|a_-|, |b_+|) - 1, & \text{falls } \phi_i^0 < 0 \end{cases} \quad (4.3)$$

mit

$$\begin{aligned} a &\equiv D_x^- \phi_i = (\phi_i - \phi_{i-1}) / \Delta x \\ b &\equiv D_x^+ \phi_i = (\phi_{i+1} - \phi_i) / \Delta x \end{aligned} \quad (4.4)$$

und mit der geglätteten Signumfunktion

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + \Delta x^2}}$$

und $h_+ := \max(h, 0)$, $h_- := \min(h, 0) \forall h \in \mathbb{R}$ dazu neigt, die Nullkontur zum nächstgelegenen Gitterpunkt zu verschieben. Um dies zu verdeutlichen, sei angenommen, dass

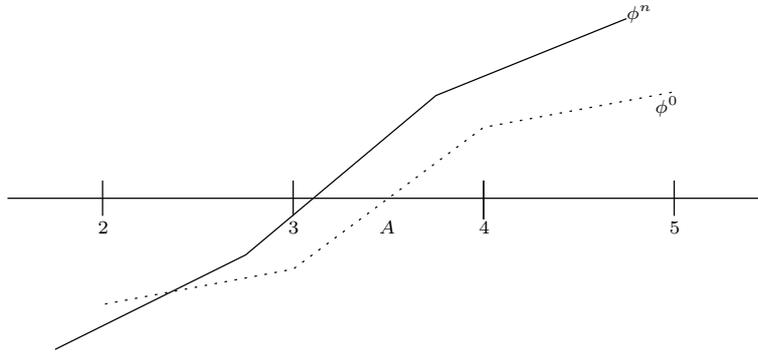


Abbildung 4.5: Die gestörte Level-Set-Funktion: gepunktete Linie. Sie hat in Punkt A den Nulldurchgang. Die Level-Set-Funktion nach einigen Iterationen: durchgezogene Linie.

ϕ^0 einen Nulldurchgang zwischen den Punkten 3 und 4 hat, also $\phi_3^0 < 0$ und $\phi_4^0 > 0$, wie in Abbildung 4.5 zu sehen. Typisch für Upwind-Schemata ist ihre Eigenschaft, die Differenzensterne nach der Charakteristik der Gleichung zu wählen. Da die Charakteristiken vom Nullniveau in normaler Richtung verlaufen (siehe Gleichung 2.17 in Abschnitt 2.4.3 auf Seite 30), wird in unmittelbarer Nähe der Nullkontur über diese hinweg differenziert und die Upwind-Eigenschaft verletzt. Dass das oben erwähnte Upwind-Verfahren zwangsweise über die Nullkontur hinweg differenziert, machen Russo und Smereka an folgendem Beispiel deutlich: Angenommen, man berechnet ϕ_i bei $i = 4$, so ergibt sich gemäß (4.4)

$$a = \frac{\phi_4 - \phi_3}{\Delta x}, a = \frac{\phi_5 - \phi_4}{\Delta x},$$

das heißt für dieses Beispiel $a > 0$, $b > 0$ und $\phi_4^0 > 0$. So liefert 4.2

$$\phi_4^{n+1} = \phi_4^n + \frac{|\phi_4^n - \phi_3^n|}{\Delta x} \Delta t$$

die Erkenntnis, dass die Upwind-Eigenschaft darin verletzt ist, dass ϕ_4^{n+1} von dem Wert ϕ_3^n abhängt, der jedoch auf der anderen Seite der Nullkontur liegt. Um ein neues Upwind-Schema zu konstruieren, modifizieren Russo und Smereka den Ausdruck 4.3 und die Signumfunktion aus Gleichung 4.2 wie folgt: In der Nähe der Nullkontur soll

$$G_i(\phi) = |D_x^{up} \phi_i| - 1 \tag{4.5}$$

gelten. Dabei sei die Upwind-Differenz $D_x^{up} \phi_i$, wie sie Abbildung 4.6 illustriert (wenn Punkt i an der Nullkontur liegt), gegeben durch ϕ_i/D_i mit D_i als Approximation der Abstandsfunktion im Punkt i auf Basis von ϕ^0 . Da die Charakteristiken stets von der

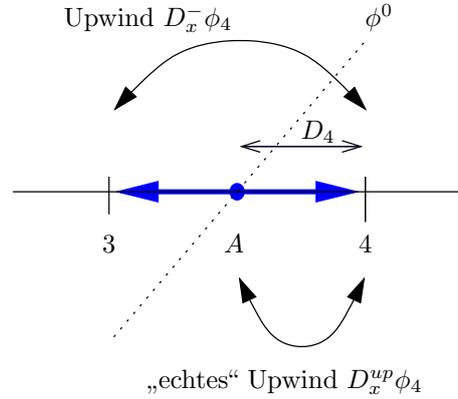


Abbildung 4.6: Die Charakteristiken (blau) weisen vom Nullniveau weg. Die übliche Upwind-Differenz $D_x^- \phi_4$ greift über die Nullstelle hinweg und erhält Information vom Wert ϕ_3 ; die Charakteristik bei Punkt 3 hat eine andere Orientierung als die bei Punkt 4. Die modifizierte Upwind-Differenz $D_x^{up} \phi_4$ behebt dieses Problem.

Nullkontur wegweisen, kann folgendes Upwind-Schema hergeleitet werden:

$$D_x^{up} = \begin{cases} \frac{\phi_i}{|D_i|} & , \text{ falls } \phi_i^0 \phi_{i-1}^0 < 0 \\ -\frac{\phi_i}{|D_i|} & , \text{ falls } \phi_i^0 \phi_{i+1}^0 < 0 \end{cases} \quad (4.6)$$

mit D_i als Approximation an die Abstandsfunktion im Punkte i . Als mögliche Wahl von D_i geben Russo und Smereka

$$D_i = \Delta x \frac{2\phi_i^0}{|\phi_{i+1}^0 - \phi_{i-1}^0|} \quad (4.7)$$

an. Zusammen mit einer geglätteten Signumfunktion

$$S = \begin{cases} \frac{D_i}{\Delta x} & , \text{ falls } \phi_i^0 \phi_{i-1}^0 < 0 \text{ oder } \phi_i^0 \phi_{i+1}^0 < 0 \\ \text{sign}(\phi_i^0) & , \text{ sonst} \end{cases} \quad (4.8)$$

sowie mittels Einsetzen von 4.6 und 4.7 in 4.2 ergibt sich durch

$$\phi_i^{n+1} = \begin{cases} \phi_i^n - \frac{\Delta t}{\Delta x} (\text{sign}(\phi_i^0) |\phi_i^n| - D_i) & , \text{ falls } \phi_i^0 \phi_{i-1}^0 < 0 \text{ oder } \phi_i^0 \phi_{i+1}^0 < 0 \\ \phi_i^n - \Delta t \text{sign}(\phi_i^0) G(\phi)_i & , \text{ sonst} \end{cases} \quad (4.9)$$

ein Upwind-Schema, welches Konsistenz mit der Nullkontur gewährleistet, indem es die Information über ihre Position nutzt und diese somit fixieren kann; man spricht auch vom sogenannten subcell fixing. In [44] können Russo und Smereka zeigen, dass eine mit subcell fixing modifizierte Reinitialisierung eine Fehlerschranke hat, die unabhängig von der Anzahl der Iterationsschritte ist.

Bemerkung: Im Falle einer stark gestörten Abstandsfunktion können sehr kleine Gradienten von ϕ^0 dazu führen, dass der Nenner in 4.7 gegen 0 geht. Eine robustere Variante sieht dafür eine Regularisierung vor. Die neue Distanz ist nun wie folgt definiert:

$$D_i = \Delta x \frac{\phi_i^0}{\Delta \phi_i^0} \text{ mit} \quad (4.10)$$

$$\Delta \phi_i^0 = \max \{ |\phi_{i+1}^0 - \phi_{i-1}^0| / 2, |\phi_{i+1}^0 - \phi_i^0|, |\phi_i^0 - \phi_{i-1}^0|, \epsilon \}. \quad (4.11)$$

Bemerkung: Eine adäquate Courant-Friedrichs-Lewy-Bedingung (CFL-Bedingung) für das oben konstruierte Schema lautet $\Delta t < \Delta x$ (siehe auch Gleichung 2.17 auf Seite 31 und folgenden Abschnitt). Die räumlichen Ableitungen in der Nähe der Nullkontur werden jedoch mit einer effektiven Maschenweite von $|D_i|$ anstatt Δx berechnet. Damit aufgrund der Tatsache $|D_i| \leq \Delta x$ bei konstantem Zeitschritt Δt dennoch Stabilität gewährleistet werden kann, kompensiert die Signumfunktion S aus 4.8 die kleinere Maschenweite. Das kann wie folgt verstanden werden: Sei c die Geschwindigkeit der Charakteristik, also $c = 1$ bei der Reinitialisierung, dann muss aufgrund der CFL-Bedingung $c \Delta t < \Delta x$ durch die Zeitschrittweite Δt erfüllt werden. Das subcell fixing liefert lokal die Maschenweite $\widetilde{\Delta x} = |D_i|$ sowie mithilfe der Signumfunktion S aus 4.8 lokal die Geschwindigkeit der Charakteristik $\widetilde{c} = |D_i| / \Delta x$. Die CFL-Bedingung lautet dann

$$\widetilde{c} \Delta t < \widetilde{\Delta x} \Leftrightarrow \frac{|D_i|}{\Delta x} \Delta t < |D_i|,$$

was äquivalent zur oben genannten und im übrigen Gebiet gültigen CFL-Bedingung $\Delta t < \Delta x$ ist.

Bemerkung: Eine Erweiterung des subcell fixing auf zwei oder mehr Dimensionen ist denkbar einfach; hier ein zweidimensionales Beispiel aus [44]: Gleichung 4.2 bleibt erhalten, 4.3 wird durch

$$G_{i,j}(\phi) = \begin{cases} \sqrt{\max(a_+^2, b_-^2) + \max(c_+^2, d_-^2)}, & \text{falls } \phi_{i,j}^0 > 0 \\ \sqrt{\max(a_-^2, b_+^2) + \max(c_-^2, d_+^2)}, & \text{falls } \phi_{i,j}^0 < 0 \end{cases} \quad (4.12)$$

ersetzt und 4.4 um

$$\begin{aligned} c &\equiv D_y^- \phi_{i,j} = (\phi_{i,j} - \phi_{i,j-1}) / \Delta x \\ d &\equiv D_y^+ \phi_{i,j} = (\phi_{i,j+1} - \phi_{i,j}) / \Delta x \end{aligned} \quad (4.13)$$

ergänzt. Die neue Distanz $D_{i,j}$ ergibt sich aus

$$D_{i,j} = \frac{2\Delta x \phi_{i,j}^0}{\left[(\phi_{i+1,j}^0 - \phi_{i-1,j}^0)^2 + (\phi_{i,j+1}^0 - \phi_{i,j-1}^0)^2 \right]^{1/2}} \quad (4.14)$$

oder einer robusteren Variante analog zu 4.10 und 4.11.

In den bisher vorgestellten Methoden wird der entstandene Massenverlust korrigiert (Chang, Hou, Merriman und Osher in [15]), wohingegen bestimmte Finite-Differenzen-Schemata für die Operatoren und Zeitschrittverfahren höherer Ordnung (Croce [18], Russo und Smereka [44]) Diskretisierungsfehler effizient minimieren können. Neben der Minimierung des Diskretisierungsfehlers steht die Stabilität der numerischen Methoden im Vordergrund. Daher soll im Folgenden diskutiert werden, welchen Einfluss die Diskretisierung auf das Stabilitätsverhalten der Reinitialisierung hat beziehungsweise unter welchen Bedingungen eine gestörte Abstandsfunktion noch reinitialisierbar ist.

Stabilität - die Courant-Friedrichs-Lewy-Bedingung

Die Gradienten einer sehr stark gestörten Abstandsfunktion entarten, so dass die Courant-Friedrichs-Lewy-Bedingung (CFL-Bedingung) nicht ohne Weiteres eingehalten werden kann. Diese besagt, dass bei einem expliziten Zeitschrittverfahren zur Lösung von zeitabhängigen Differentialgleichungen eine notwendige Voraussetzung für die Stabilität des Verfahrens darin liegt, die Zeitschrittweite derart zu begrenzen, dass die Charakteristik der Gleichung innerhalb eines Zeitschritts um nicht mehr als eine räumliche Maschenweite voranschreitet (siehe auch R. Courant, K. Friedrichs and H. Lewy [41]).

Der im Folgenden vorgestellte Ansatz analysiert die Reinitialisierung 2.16 unter dem Gesichtspunkt der CFL-Bedingung. In [37] wird dieser Sachverhalt wie folgt erläutert: An Stelle der Signumfunktion $\text{sign}(d)$ der gestörten Abstandsfunktion d wird in der Numerik eine regularisierte Form

$$s_h(d) := \frac{d}{\sqrt{d^2 + h^2}} \quad (4.15)$$

verwendet. Diese konvergiert gegen $\text{sign}(d)$ für große d und ist hinreichend glatt für $|d| < 2h$ (siehe Abbildung 4.7). Nun funktioniert das Verfahren gut, solange die Störung

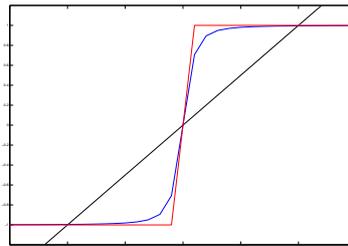


Abbildung 4.7: Die Regularisierung der Signumfunktion: In schwarz die Funktion d , in rot $\text{sign}(d)$ und in blau die regularisierte Signumfunktion $s_h(d)$.

der Abstandseigenschaft hinreichend klein ist. Bei größeren Störungen treten zwei Probleme auf. Ist der Gradient von d zu klein, wird $s_h(d)$ ebenfalls klein und damit auch

die Ausbreitungsgeschwindigkeit der Charakteristik, so dass die Anzahl der Iterationen in 2.16 steigt. Wenn hingegen der Gradient von d zu groß wird, kann das dazu führen, dass die Funktion d in der Nähe der Nullniveaumenge ihr Vorzeichen ändert und diese somit ungewollt verschiebt. Der Grund für dieses Phänomen wird in [37] anhand folgenden Beispiels erklärt: Die gestörte eindimensionale Abstandsfunktion d^0 nehme im Intervall $[x_i, x_{i+1}]$ den Wert 0 an, und es gelte $d_i^0 < 0 < d_{i+1}^0$. Bei Upwind erster Ordnung für die Ortsdiskretisierung und dem expliziten Euler-Verfahren für die Zeitdiskretisierung ergibt sich nach dem ersten Zeitschritt

$$\begin{aligned} d_i^1 &= d_i^0 + s_i \Delta\tau \left(1 - \frac{d_{i+1}^0 - d_i^0}{\Delta x} \right) \\ d_{i+1}^1 &= d_{i+1}^0 + s_{i+1} \Delta\tau \left(1 - \frac{d_{i+1}^0 - d_i^0}{\Delta x} \right), \end{aligned}$$

wobei $-1 < s_i < 0$ und $0 < s_{i+1} < 1$ Approximationen an die Signumfunktion $\sigma_{\Delta X}(d_i)$ beziehungsweise $\sigma_{\Delta X}(d_{i+1})$ sind. Das obige Schema ist monoton, das heißt, d_i^1 ist nicht fallend in d_i^0 und $d_{i\pm 1}^0$, solange die CFL-Bedingung $\Delta\tau < \Delta x$ erfüllt ist. Daher gilt $d_i^1 < 0 < d_{i+1}^1$, wenn $\frac{d_{i+1}^0 - d_i^0}{\Delta x} \leq 1$ ist, und zwar unabhängig von der Wahl von $\Delta\tau$, s_i und s_{i+1} . Ist hingegen $\frac{d_{i+1}^0 - d_i^0}{\Delta x} > 1$, so kann es vorkommen, dass zusammen mit der Approximation von $\text{sign}(d)$ durch die regularisierte Signumfunktion $s_{\Delta x}(d)$ nach 4.15 die Funktion d ihr Vorzeichen wechselt. Das folgende Beispiel aus [37] verdeutlicht diesen Sachverhalt: Sei $\Delta\tau = c \Delta x$, $d_i = -m \Delta x$ und $d_{i+1} = -n \Delta x$, dann folgt

$$\begin{aligned} d_i^1 &= m \Delta x \left[-1 + \frac{c(m+n-1)}{\sqrt{m^2+1}} \right] \\ d_{i+1}^1 &= n \Delta x \left[-1 + \frac{c(m+n-1)}{\sqrt{n^2+1}} \right]. \end{aligned}$$

Nun lassen sich – ganz gleich, wie klein c auch sein mag – m und n so wählen, dass d_i^1 oder d_{i+1}^1 ihr Vorzeichen wechseln und der Nulldurchgang das Intervall $[x_i, x_{i+1}]$ verlässt. Um sicher zu stellen, dass dieser Fall nicht eintritt, gelte

$$\frac{\Delta\tau}{\Delta x} \leq \frac{|d_k|}{\Delta x |s_k| \left| 1 - \frac{d_{k+1} - d_k}{\Delta x} \right|} \quad \text{für } k \in \{i, i+1\},$$

ohne jedoch die Zeitschrittweite $\Delta\tau$ weiter einzuschränken, als die CFL-Bedingung vorgibt. Dies führt zu einer neuen Wahl der Approximation an die Signumfunktion:

$$s_k := \frac{d_k}{\sqrt{d_k^2 + \left(1 - \left| \frac{d_{k+1} - d_k}{\Delta x} \right| \right)^2 \Delta x^2}}.$$

Diese hängt nun lokal von der Steigung von d ab. Ebenso gilt obiges Schema auch für höhere Dimensionen, und man erhält unter Verwendung einer Approximation $D(d)$ für ∇d :

$$\sigma(d) := \frac{d}{\sqrt{d^2 + (1 - |D(d)|)^2 \Delta x^2}} .$$

Für eine regularisierte Form wird 4.15 geglättet und liefert die im Folgenden ausschließlich verwendete geglättete und regularisierte Signumfunktion:

$$\sigma_h(d) := \frac{d}{\sqrt{d^2 + |D(d)|^2 h^2}} . \quad (4.16)$$

Mithilfe von 4.16 lässt sich die Verschiebung der Nullkontur derart eingrenzen, dass diese die Gitterpunkte nicht überschreitet; es findet also eine Verschiebung nur noch maximal bis zu den Gitterpunkten statt, deren Distanz zur gestörten Abstandsfunktion kleiner als eine Gittermaschenweite ist. Zusammenfassend lässt sich sagen, dass die Wahl von 4.16 als Approximation der Signum-Funktion das Problem der Verschiebung der Nullkontur über Zellgrenzen hinweg löst, wenn d steil ist, sowie die Konvergenz beschleunigt, wenn d flach ist. Im Folgenden werden alle numerischen Berechnungen, soweit nicht anders angegeben, mit der geglätteten, regularisierten Signumfunktion σ_h aus 4.16 durchgeführt.

Bemerkung: Die Ursache für die Verletzung der CFL-Bedingung liegt nicht nur in einer zu stark gestörten Abstandsfunktion, sondern auch in der fehlerhaften Auswertung des Gradienten auf dem kartesischen Gitter. Das heißt, dass ein Finite-Differenzen-Schema umso ungenauer wird, je weniger die Ausrichtung des Gradienten mit den Gitterachsen übereinstimmt (siehe Abbildung 4.9a). Ferner ist die Abstandsfunktion im Allgemeinen nicht differenzierbar, sondern lediglich Lipschitz-stetig. Solche Lipschitz-stetigen Stellen treten in der Regel außer bei Topologieänderungen nicht auf der Nullkontur selbst auf (siehe Abbildung 4.9b). Differenzenverfahren produzieren an Gitterpunkten in unmittelbarer Nähe dieser Stellen Fehler der Größenordnung $\mathcal{O}(1)$. Ein weiteres Problem ergibt sich, wenn der Differenzenstern wie in Abbildung 4.10 über Topologieänderungen hinweggreift.

4.2 Das Zweiparameterproblem

Der vorherige Abschnitt zeigt, welche unerwünschten Auswirkungen die numerische Diffusion bei Level-Set-Methoden im Bereich der Zweiphasensimulation haben und wie mit den in dieser Arbeit vorgestellten Methoden das Ausmaß der künstlichen Verschiebung der Nullkontur direkt auf die Maschenweite H beschränkt werden kann. In den folgenden Abschnitten werden die Ideen und der Weg zu einer verbesserten Level-Set-Methode vorgestellt.

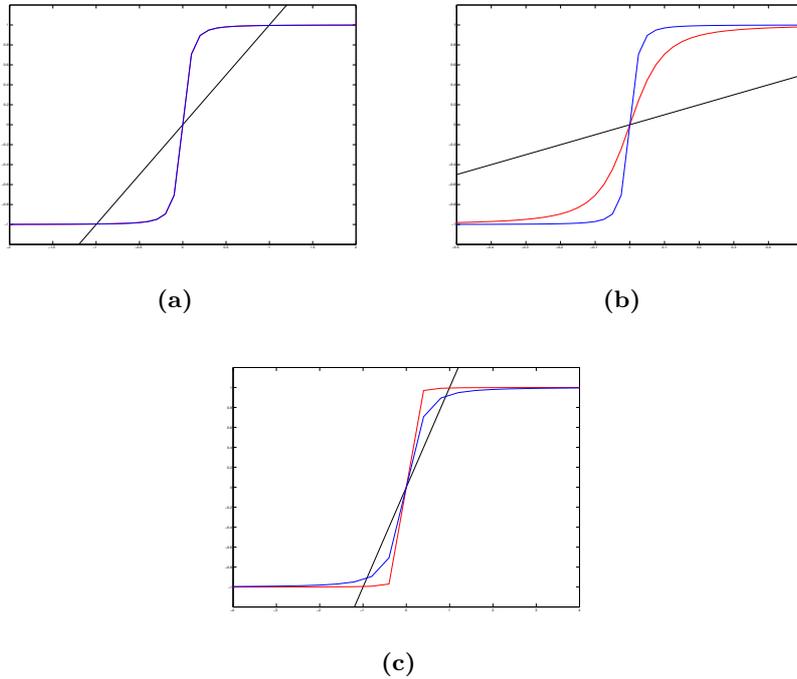


Abbildung 4.8: Die Glättung wirkt auf die regularisierte Signumfunktion in Abhängigkeit von ∇d : In dieser Darstellung ist d in schwarz, die regularisierte Signumfunktion $s_h(d)$ in rot und die geglättete, regularisierte Signumfunktion $\sigma_h(d)$ in blau eingezeichnet. Im Fall (a) ist $\nabla d = 1$, und s_h ist mit σ_h identisch. Im Fall (b) ist $\nabla d < 1$, und σ_h gleicht die zu flache s_h aus. Im Fall (c) ist $\nabla d > 1$, und σ_h ist gegenüber s_h abgeflacht, so dass die CFL-Bedingung trotz des großen Gradienten von d nicht verletzt wird.

Um die Level-Set-Methode zu verbessern, wird ein zweiter Parameter h mit $h \ll H$ eingeführt, der substantiell kleiner ist als H und die Level-Set-Reinitialisierung derart auf den Parameter h überführt, dass die Verschiebung der Nullkontur nur noch im Rahmen von h abläuft. Die Idee dahinter beruht auf der Konstruktion eines zweiten Gitters mit der Maschenweite h , so dass der bisherige Level-Set-Algorithmus vom groben Gitter der Maschenweite H entkoppelt und auf das feinere Gitter übertragen werden kann. Folgende Punkte sollen dabei berücksichtigt werden:

- Die neue Technik soll modular zu NaSt3DGPF hinzugefügt werden.
- Änderungen an NaSt3DGPF selbst sind möglichst gering zu halten.
- Das neue Gitter soll im Wesentlichen kartesisch sein, damit alte Algorithmen weiterhin verwendet werden können.

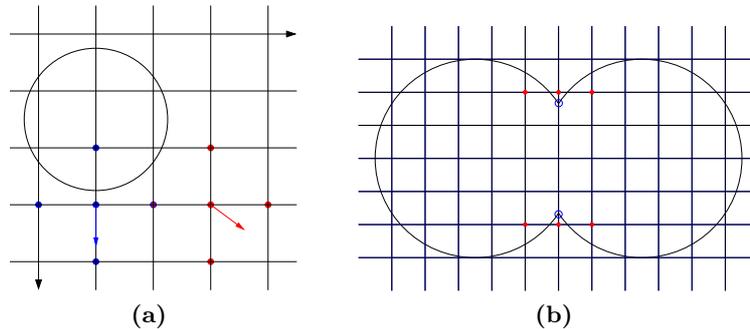


Abbildung 4.9: (a) Eine Abstandsfunktion in zwei Dimensionen: Die Nullkontur entspricht einem Kreis. Die Pfeile deuten zwei Gradienten der Abstandsfunktion an; die Punkte in jeweils gleicher Farbe sind die Gitterpunkte, die bei zentralen Differenzen zur Berechnung des Gradienten herangezogen werden. Blau: Gradient liegt auf einer Gitterachse. Rot: Gradient ist weder zur x - noch zur y -Achse parallel. (b) Eine Topologieänderung mit Lipschitz-stetiger Stelle (\circ) in der Nullkontur durch das Verschmelzen zweier Kreise. Eine partielle Ableitung mit zentralen Differenzen (\bullet) respektiert die Topologie nicht.

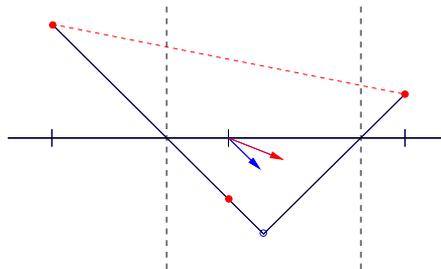


Abbildung 4.10: Eine eindimensionale Abstandsfunktion: Drei Freiheitsgrade (senkrechte Markierungen $|$) liegen auf unterschiedlichen Seiten der Nullkontur (senkrechte gestrichelte Linie). Die zentralen Differenzen im mittleren Freiheitsgrad (rote Punkte) greifen über mehr als einen Nulldurchgang hinweg. Es resultiert ein schlecht berechneter Gradient (roter Pfeil) im Vergleich zum wahren Gradienten (blauer Pfeil).

- Aus Kostengründen wird ein adaptives Gitter einem globalen Gitter vorgezogen.

Der folgende Abschnitt gibt Aufschluss über die Umsetzung der Idee des Zweiparameterproblems.

4.2.1 Ein Zweigitterverfahren

Gittertypen

Zur Erläuterung der in dieser Arbeit vorgestellten Methode seien hier noch einmal kurz die unterschiedlichen Diskretisierungen beziehungsweise Gittertypen (siehe Ab-

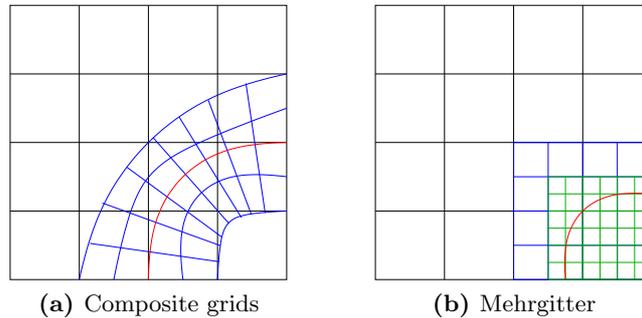


Abbildung 4.11: (a) Composite grids: Grobes Gitter mit Maschenweite H und kurvilineares Gitter (blau) mit Maschenweite h in einem schmalen Band um die Nullkontur (rot). (b) Ein klassisches Mehrgitterverfahren: Das grobe Gitter mit der Maschenweite H (schwarz) wird im Bereich der Nullkontur (rot) durch eine Sequenz von ineinander verschachtelten, kartesischen Gittern mit kleiner werdenden Maschenweiten ergänzt. Das feinste Gitter hat die Maschenweite h .

schnitt 2.1) und numerischen Verfahren bei adaptiven Gittern (siehe Kapitel 3) aufgeführt. Es stehen folgende Grundtypen von Gittern und Diskretisierungen zur Verfügung:

- Strukturierte Gitter
 - kurvilinear
 - kartesisch
- Unstrukturierte Gitter
 - Triangulierungen
 - octrees

Im Abschnitt 2.1.1 wird gezeigt, dass mithilfe von Level-Set-Funktionen kurvilineare Gitter erzeugt werden können. Auf diese Weise lassen sich sehr einfach Gitter mit einer kleinen Maschenweite h in dem gewünschten, schmalen Band um die Nullkontur der Level-Set-Funktion generieren. Da zur Konstruktion des Gitters die Level-Set-Funktion selbst herangezogen wird, stehen darüber hinaus die Abbildungen zwischen dem kurvilinearen Gitter und seinem korrespondierenden kartesischen Gitter zur Verfügung, wodurch die Transformation der bestehenden Differentialoperatoren auf das neue Gitter und damit eine Erweiterung von NaSt3DGPF ermöglicht wird (vergleiche Beispiel 2.1 und Abbildungen 2.4 und 4.11a). In Abschnitt 3.3.1 wird ferner ein Verfahren beschrieben, mit dem das ursprünglich grobe und das feine kurvilineare Gitter im Bereich der Nullkontur in einer Art Mehrgitteralgorithmus auf composite grids zur Lösung der Gleichung beitragen. Abschnitt 3.4 hingegen beschreibt den Gebrauch der beiden Gitter im Zusammenhang mit Gebietszerlegungsmethoden. Mit einem Lösungsansatz auf rein kartesischen Gittern

basierend, lässt sich das Zweiparameterproblem ebenfalls realisieren. Dabei kommt eine Sequenz von kartesischen Gittern mit unterschiedlichen Maschenweiten zum Einsatz, bei der die feinen Gitter einen wesentlich kleineren Teil des Gebiets abdecken als das globale grobe Gitter. Dies führt zu einem klassischen Mehrgitterverfahren (siehe Abbildung 4.11b). Allerdings kann eine topologisch komplexe Nullkontur dazu führen, dass die feinen Gitter ebenfalls das gesamte Gebiet oder den größten Teil davon abzudecken haben, was den adaptiven Effekt zunichte macht. Ein voll adaptives Mehrgitterverfahren hingegen arbeitet prinzipiell auf unstrukturierten Gittern wie octrees und Triangulierungen. Letztere sind bezüglich der einfachen Erweiterung von NaSt3DGPF, dessen Numerik auf finiten Differenzen beruht, ungeeignet. Dadurch könnten die in NaSt3DGPF bereits integrierten Finite-Differenzen-Schemata nicht weiter verwendet werden und eine weitaus aufwendigere Implementierung der Level-Set-Reinitialisierung mit finiten Elementen erfordern. Auf der anderen Seite kann auf octrees mit Finite-Differenzen-Schemata operiert werden, wenn dafür aber die Behandlung von hängenden Knoten in Kauf genommen wird. Der Abschnitt 3.2 stellt Interpolationsverfahren zur Rekonstruktion von hängenden Knoten in Finite-Differenzen-Schemata auf ctrees vor. Ferner lassen sich hängende Knoten leichter handhaben, wenn ihre Anzahl pro Gitterzellenkante auf 1 beschränkt ist; das heißt, durch die Wahl einer 1-Irregularität des octrees sind im Allgemeinen nicht nur die Grob- und Feingittermaschenweiten H und h vertreten, sondern wegen $h \ll H$ auch viele Zwischengrößen. Belässt man es bei zwei Maschenweiten h und H und nimmt eine große Irregularität in Kauf, liefert der Abschnitt 3.4.3 über die Mortar-Mimetische-Finite-Differenzen-Methode einen möglichen Lösungsansatz.

Verfeinerungskriterium

Um ein effizientes adaptives Verfahren konstruieren zu können, benötigt man neben einem Überblick über verschiedene Gittertypen eine Auswahl an Verfeinerungskriterien, welche Vorschriften zur Verfeinerung und Vergrößerung des Gitters liefern. Betrachtet man das Verfeinerungskriterium als Abbildung, dann ordnet sie zu jedem Zeitpunkt t jeder Grobgitterzelle $c \in G_H$ ein Element der Menge $\{0, 1\}$ zu:

$$\begin{aligned} \text{refine} : G_H \times \mathbb{R}^+ &\rightarrow \{0, 1\} \\ (c, t) &\mapsto \begin{cases} 1, & c \text{ ist zu verfeinern} \\ 0, & c \text{ ist nicht zu verfeinern} \end{cases} \end{aligned}$$

Im Falle von Level-Set-Methoden für Strömungssimulationen mit zwei Phasen soll die Adaptivität gerade dafür sorgen, dass die Reinitialisierung der Level-Set-Funktion die Nullkontur nicht stört und ferner die Abstandseigenschaft in einer Umgebung um die Nullkontur möglichst präzise rekonstruiert wird.

Sei ϵ der Parameter, der die Ausdehnung dieser Umgebung festlegt, dann sei das Verfei-

nerungskriterium refine_ϵ wie folgt definiert:

$$\begin{aligned} \text{refine}_\epsilon : G_H \times \mathbb{R}^+ &\rightarrow \{0, 1\} \\ (c, t) &\mapsto \begin{cases} 1, & |\phi(P(c))| \leq \epsilon \\ 0, & |\phi(P(c))| > \epsilon \end{cases}, \end{aligned} \quad (4.17)$$

wobei $\phi(x)$ die Level-Set-Funktion im Punkte x und $P(c)$ die absolute Position der Gitterzelle c ist. Zu diesem Zweck lässt sich bei kartesischen Gittern $P(c)$ mit den Koordinaten des Freiheitsgrades in c und im Fall einer Triangulierung zum Beispiel mit den Koordinaten des Zellenmittelpunktes oder des Schwerpunktes identifizieren. Dieses Kriterium refine_ϵ garantiert, dass die Nullkontur und ihre Umgebung der Breite ϵ durch ein feines Gitter G_h der Maschenweite h diskretisiert werden. Es ist jedoch Vorsicht geboten, da der Begriff „Breite“ nicht im euklidischen Sinn zu verstehen ist, sondern sich auf den Wert der Level-Set-Funktion bezieht. Das bedeutet, dass die Umgebung verzerrt wird, sobald die Abstandseigenschaft der Level-Set-Funktion gestört ist. Eine Möglichkeit, dieses Problem zu umgehen, beruht auf der Annahme, dass $\nabla\phi$ innerhalb der Umgebung entlang jeder Normalen annähernd konstant ist. Abbildung 4.12 zeigt dazu ein Beispiel.

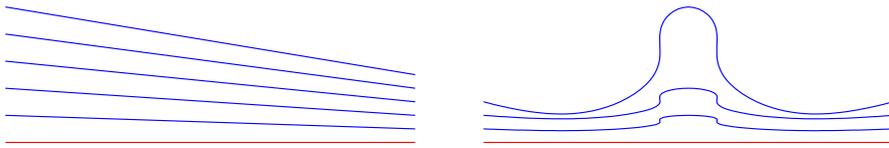


Abbildung 4.12: Die Konturen gestörter Abstandsfunktionen: Die Abstandseigenschaft zur Nullkontur (rot) ist gestört ($|\nabla\phi| \neq 1$). Im linken Bild ist $\nabla\phi$ entlang jeder Normalen konstant, im rechten Bild nicht.

Eine entsprechende lokale Skalierung der Bandbreite ϵ mit $|\nabla\phi|$ führt zu einer Umgebung mit annähernd euklidischer Breite ϵ . Das korrigierte Verfeinerungskriterium für gestörte Abstandsfunktionen lautet nun $\widehat{\text{refine}}_\epsilon$:

$$\begin{aligned} \widehat{\text{refine}}_\epsilon : G_H \times \mathbb{R}^+ &\rightarrow \{0, 1\} \\ (c, t) &\mapsto \begin{cases} 1, & |\phi(P(c))| \leq \epsilon |\nabla\phi(P(c))| \\ 0, & |\phi(P(c))| > \epsilon |\nabla\phi(P(c))| \end{cases}. \end{aligned} \quad (4.18)$$

Die Abbildungen (4.13) verdeutlichen den Unterschied der beiden Verfeinerungskriterien refine_ϵ und $\widehat{\text{refine}}_\epsilon$. Bei einer stark gestörten Abstandsfunktion kann refine_ϵ kein adäquates Gitter erzeugen.

Abschließend wird der oben genannte Begriff der Umgebung genauer definiert:

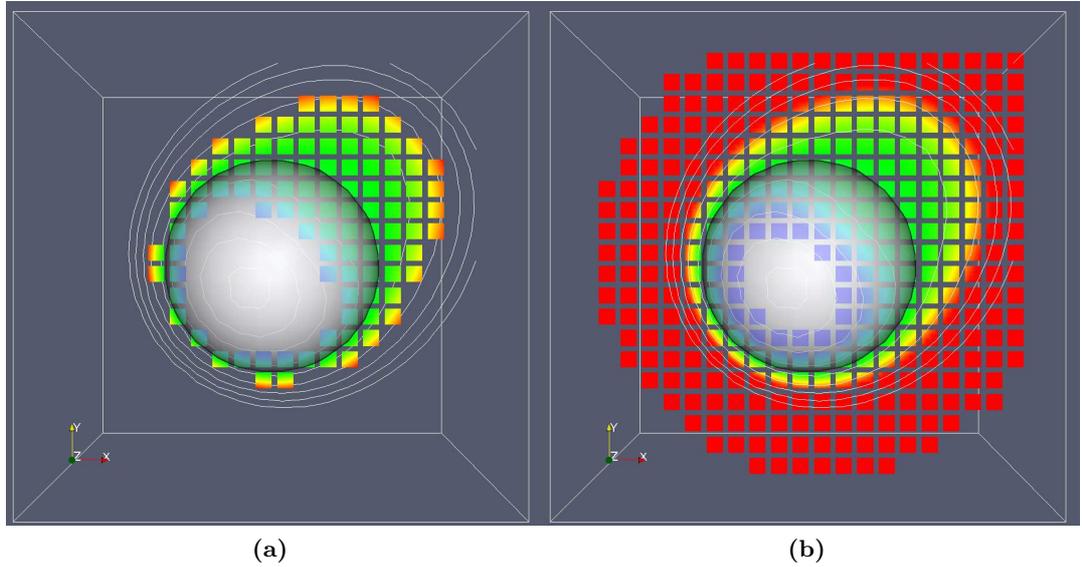


Abbildung 4.13: Die Verfeinerungskriterien refine_ϵ und $\widehat{\text{refine}}_\epsilon$ im Vergleich. In (a) findet refine_ϵ und in (b) $\widehat{\text{refine}}_\epsilon$ Verwendung. Zu sehen ist die Nullkontur als Sphäre mit Radius 1 und eine Schnittfläche durch die Level-Set-Funktion mit mehreren Konturen aus dem Intervall $[-1, +1]$. Zum Vergleich ist ein Schnitt durch die grid patches dargestellt.

Definition 4.1 [DAS RELEVANTE GEBIET]

Das relevante Gebiet $\Omega_r^\epsilon \subset \Omega \subset \mathbb{R}^3$ sei definiert als:

$$\Omega_r^\epsilon := \{x \in \Omega \mid |d(x)| \leq \epsilon\},$$

wobei d eine Abstandsfunktion zur Level-Set-Nullkontur ist.

Die Charakteristiken der Hamilton-Jacobi-Iteration

Das Schema der Adaptivität für die Level-Set-Methode bewirkt, wie in Abbildung 4.14 veranschaulicht, dass das Gitter in einer Umgebung um die Nullkontur der Level-Set-Funktion – dem relevanten Gebiet – verfeinert wird. Ein Algorithmus, der der Wiederherstellung der Abstandseigenschaft der Level-Set-Funktion dient, löst iterativ die Gleichung 2.16 vom Hamilton-Jacobi-Typ, deren Charakteristiken in Richtung der äußeren und inneren Normalen der Nullkontur verlaufen. Die Information im Gebiet fließt demnach von der Nullkontur weg; insbesondere vom fein diskretisierten zum grob diskretisierten Gebiet. Das bedeutet einerseits, dass der Algorithmus prinzipiell keine Information aus dem grob diskretisierten Gebiet bezieht und andererseits Störungen, die durch Inkonsistenzen am Übergang vom feinen zum groben Gitter entstehen, keine Auswirkung

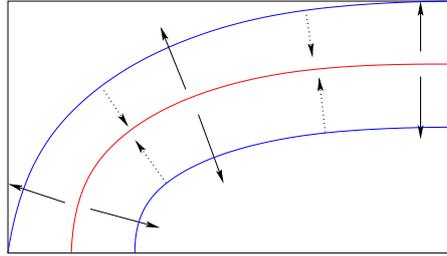


Abbildung 4.14: Der Informationsfluss bei der Level-Set-Reinitialisierung: Die Nullkontur (rot) liegt in ihrem relevanten Gebiet (im Bild durch blaue Linien angedeutet), in der das Gebiet fein diskretisiert wird. Durchgezogene Pfeile zeigen die Richtung der Charakteristiken und gestrichelte Pfeile deuten die Richtung an, aus der sich Upwind-Schemata die Information holen.

auf die Nullkontur und das relevante Gebiet haben. Dieses Phänomen wird zusätzlich durch die Verwendung von Upwind-Schemata ausgenutzt (siehe Abbildung 4.14).

In dieser Arbeit soll eine verbesserte Methode zur Reinitialisierung einer Level-Set-Funktion entwickelt werden, deren Diskretisierung gemäß der Aufzählung auf Seite 69 sowohl adaptiv ist als auch von kartesischen Gittern Gebrauch macht. Zusammen mit den vorhergehenden Überlegungen resultiert daraus die Idee, die groben Gitterzellen c mit kleinen kartesischen Gittern der feinen Maschenweite h – im folgenden grid patches genannt – zu versehen, falls $\widehat{\text{refine}}_\epsilon(c, t)$ den Wert 1 annimmt. Der zugehörige Algorithmus besitzt dabei die Grundzüge eines Mehrgitterverfahrens. Die folgenden Abschnitte dieses Kapitels beschreiben die konkrete Entwicklung dieser Methode.

4.2.2 Kartesische grid patches

Die vorhergehenden Erläuterungen zu Gittertypen und Verfeinerungskriterien und die in der Aufzählung auf Seite 69 genannten Punkte führen zu der Wahl von feinen kartesischen grid patches, die bei Bedarf, das heißt, wenn es das Verfeinerungskriterium für Adaptivität vorschreibt, auf eine grobe Gitterzelle aufgesetzt werden. Im Folgenden soll die neue Gitterstruktur genauer definiert werden. Die wesentlichen Merkmale lauten:

- Jedes grid patch ist ein strukturiertes kartesisches Gitter.
- Alle grid patches besitzen dieselbe Struktur, das heißt, die Anzahl der Freiheitsgrade auf den entsprechenden Raumachsen ist bei allen grid patches die gleiche.
- Nicht notwendigerweise, aber der Einfachheit halber ist die Anzahl der Freiheitsgrade im grid patch auf jeder Achse gleich. Sie ist eine Potenz von 2. Ein grid patch in drei Dimensionen besitzt also insgesamt $(2^l)^3$ Freiheitsgrade mit der Irregularität l .

- Jedes Patch überdeckt exakt eine Gitterzelle des groben Gitters.
- Ein grid patch besitzt für jede Raumrichtung (x, y, z) isotrope Maschenweiten $(dx/2^i, dy/2^i, dz/2^i)$, wobei (dx, dy, dz) die Ausdehnungen der assoziierten Grobgitterzelle sind.

4.2.3 Das patch grid

Adaptivität entsteht nun, indem eine zu verfeinernde Grobgitterzelle c mit einem grid patch assoziiert wird, wobei die Zelle c , geometrisch betrachtet, zu ihrem grid patch kongruent ist. Die Gesamtheit aller Grobgitterzellen und grid patches heie patch grid. Abbildung 4.15a verdeutlicht das Prinzip eines patch grids. Ferner zeigt die Abbil-

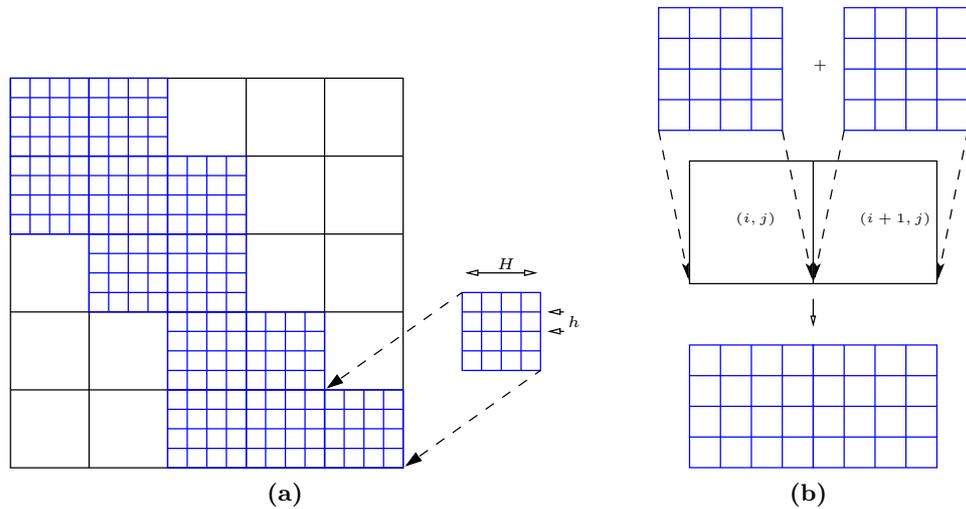


Abbildung 4.15: (a) Ein zweidimensionales Beispiel für ein patch grid, bestehend aus dem groben Gitter mit Maschenweite H und den grid patches (blau) mit Maschenweite h und der Irregularität $\iota = 2$. (b) Zwei grid patches benachbarter Grobgitterzellen gehen nahtlos ineinander über.

dung 4.15b, dass die oben genannten Eigenschaften eines patch grids dafür sorgen, dass grid patches, die mit jeweils benachbarten Grobgitterzellen assoziiert sind, nahtlos ineinander übergehen und lokal ein strukturiertes Gitter ergeben.

Bemerkung: Alle grid patches sind bezüglich ihrer logischen Struktur identisch. Das grobe Gitter jedoch kann lokal anisotrope Maschenweiten besitzen. Da Adaptivität in dieser Arbeit grundsätzlich über ein zweites Gitter mit substantiell kleinerer Maschenweite eingeführt wird, wird vorausgesetzt, dass die Variation der anisotropen Maschenweite klein im Vergleich zur Irregularität ist. Das bedeutet, dass die feinste Grobgittermaschenweite H_{min} stets wesentlich größer ist als die grbste Feingittermaschenweite h_{max} ; also

$H_{min}/h_{max} = \mathcal{O}(2^i)$. Daher soll der Fall der Anisotropie als solcher nicht mehr explizit behandelt werden.

Die Position der Freiheitsgrade innerhalb eines grid patches ist gerade so gewählt, dass die Freiheitsgrade der assoziierten Grobgitterzelle stets in dem feinem Gitter enthalten sind (vergleiche auch Abbildung 4.16). Daher soll gelten:

Definition 4.2 [POSITION DER FREIHEITSGRADE]

Sei $(0,0,0)$ die linke untere Ecke des grid patches, (dx, dy, dz) dessen Ausdehnung in x -, y - und z -Richtung und i der Grad der Irregularität. Für die Maschenweiten (h_x, h_y, h_z) des grid patches ergibt sich dann

$$\begin{pmatrix} h_x \\ h_y \\ h_z \end{pmatrix} := \begin{pmatrix} dx/2^i \\ dy/2^i \\ dz/2^i \end{pmatrix}$$

Die Freiheitsgrade des Gitters seien mit (i, j, k) indiziert, wobei $i, j, k \in \{1 \dots 2^i\}$ und $i \geq 1$. Dann ist die Position $P(i, j, k)$ des Freiheitsgrades P mit dem Index (i, j, k) definiert als

$$P(i, j, k) = \begin{pmatrix} i \cdot h_x \\ j \cdot h_y \\ k \cdot h_z \end{pmatrix}$$

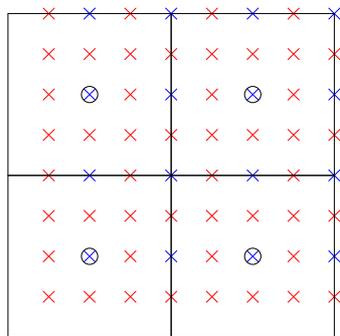


Abbildung 4.16: Die Freiheitsgrade von grid patches: Es sind vier Grobgitterzellen in zwei Dimensionen abgebildet, von denen jede mit einem grid patch versehen ist. Die schwarzen Kreise (o) stellen die Freiheitsgrade des groben Gitters dar, o und x sind die Freiheitsgrade der nächst feineren grid patches mit der Irregularität 1. o, x und x kennzeichnen die Freiheitsgrade der grid patches der Irregularität 2.

Die Wahl eines wie oben beschriebenen Gitters ist zum einen mit den auf Seite 69 genannten Forderungen an die neue Technik konform, zum anderen ergeben sich gewisse Vorteile gegenüber anderen adaptiven Gittertypen, wie sie in den Abschnitten 2.1.1 und 2.1.2 ab Seite 10 beschrieben werden. Gegenüber triangulierten Gittern sind patch

grids aus kartesischen, strukturierten Teilen aufgebaut, so dass die Position der Freiheitsgrade im Wesentlichen implizit gegeben ist. Daher können patch grids ihre Freiheitsgrade effizienter speichern und einen schnelleren Zugriff auf diese gewähren. Ferner können bestehende Finite-Differenzen-Schemata auf die einzelnen grid patches und über Grenzen benachbarter grid patches hinweg weiter verwendet werden. Die Gittererzeugung und Anpassung im zeitlichen Verlauf einer Rechnung ist mit patch grids deutlich leichter als bei Triangulierungen, da das Gitter gemäß des Adaptivitätskriteriums modifiziert werden kann, ohne anschließend Korrekturen durchführen zu müssen, um etwa hängende Knoten oder entartete Elemente zu eliminieren. Mit dem Grad der Irregularität ι steigt die Effizienz eines patch grids gegenüber einer triangulierten Gitterversion. Im Vergleich zu octrees besitzen patch grids genau zwei Maschenweiten und bilden daher große Teilgebiete, in denen das feine Gitter lokal als kartesisch und uniform betrachtet werden kann. Die Grenze zwischen feinen grid patches und Grobgitterzellen stellt stets einen Übergang mit konstanter Irregularität dar. Abschließend wird der Begriff des relevanten Gitters eingeführt.

Definition 4.3 [DAS RELEVANTE GITTER]

Das relevante Gitter $G_r^\epsilon \subset \Omega \subset \mathbb{R}^3$ sei das Teilgebiet, das gemäß eines Verfeinerungskriteriums von grid patches überdeckt wird:

$$G_r^\epsilon := \{x \in \Omega \mid \text{mit } \exists c \in G_H \text{ mit } x \in c \text{ und } \text{refine}(c) = 1\},$$

wobei refine eines der Verfeinerungskriterien refine_ϵ oder $\widehat{\text{refine}}_\epsilon$ ist. Ferner bezeichne das relevante Gitter ${}^H G_r^\epsilon$ die Menge aller Grobgitterzellen, die gemäß eines Verfeinerungskriteriums mit grid patches assoziiert sind und ${}^P G_r^\epsilon$ die Menge aller grid patches.

Solange eine Verwechslung auszuschließen ist, werden von nun an G_r^ϵ , ${}^H G_r^\epsilon$ und ${}^P G_r^\epsilon$ allgemein als relevantes Gitter bezeichnet. Mit $\text{ass}(c)$ für eine Zelle $c \in {}^H G_r^\epsilon$ wird im Folgenden das mit der Zelle c assoziierte grid patch bezeichnet.

4.2.4 Datenstruktur

Bei der Implementierung von patch grids kommt eine zweigeteilte Datenstruktur zum Einsatz. Auf der einen Seite stehen die grid patches, die in dreidimensionalen arrays die zu jeweils einem grid patch gehörenden Freiheitsgrade der Level-Set-Funktion speichern, auf der anderen Seite erfüllt der Patch-Manager die Funktion, die grid patches zu verwalten, sie logisch zu einem Gitter zu organisieren und alle nötigen numerischen Algorithmen zur Verfügung zu stellen. Zu den Aufgaben des Patch-Managers gehören das Erzeugen von neuen und das Entfernen von alten grid patches, die Initialisierung der Freiheitsgrade der Level-Set-Funktion sowie das Operieren auf den Daten, die von den grid patches bereitgestellt werden. Dazu werden Referenzen auf die grid patches unter

Verwendung des Index der assoziierten Grobgitterzelle als Suchschlüssel in einem Container abgelegt. Eine hash map mit universellem hashing als spezielle Implementierung dieses Containers erlaubt eine konstante mittlere Zugriffszeit $\mathcal{O}(1)$ auf einzelne grid patches. Ferner verfügt der Patch-Manager über alle Informationen, die das grobe Gitter genau definieren, sowie über die Level-Set-Funktion des groben Gitters selbst.

4.2.5 Der Algorithmus

In den vorhergehenden Abschnitten 4.1 und 4.2 ist das Problem der numerischen Diffusion und eine erste Idee zu dessen Lösung mittels eines sogenannten Zweiparameterproblems vorgestellt worden. Darin stehen diese beiden Parameter für die Kenngrößen zweier verschiedener Gitter in der Diskretisierung – die Maschenweiten H und h . Der Entwurf eines neuen Algorithmus für die Level-Set-Reinitialisierung berücksichtigt nun die Tatsache, dass die Größen der Strömungssimulation wie Geschwindigkeitskomponenten (U, V, W) und Druck P , aber auch die Level-Set-Funktion auf dem groben Gitter G_H mit Maschenweite H definiert sind und darauf berechnet werden, das Problem der numerischen Diffusion jedoch lediglich die Reinitialisierung der Level-Set-Funktion betrifft. Damit ergibt sich die Möglichkeit, für eine verbesserte Level-Set-Reinitialisierung eine eigenständige, von der Numerik des Navier-Stokes-Lösers entkoppelte Diskretisierung einzuführen. Damit das Fehlerkalkül der Level-Set-Numerik nun auf Basis von h anstatt H durchgeführt werden kann, liegt es nahe, die Reinitialisierung auf das feine Gitter G_h zu übertragen. Aus den Überlegungen über Verfeinerungskriterien und den Verlauf von Charakteristiken in der Hamilton-Jacobi-Iteration in Abschnitt 4.2.1 entsteht der Gedanke, dass alle wesentlichen Informationen im relevanten Gebiet vorliegen; darüber hinaus fließt während der Reinitialisierung Information, von der Nullkontur ausgehend, immer vom relevanten Gebiet in das übrige Teilgebiet $\Omega \setminus \Omega_f^\epsilon$ hinein, aber niemals in umgekehrter Richtung. Die Idee besteht daher nun darin, das relevante Gebiet unabhängig vom übrigen Teilgebiet zu betrachten und eine Reinitialisierung lediglich lokal auf dem relevanten Gitter durchzuführen. Durch diese Vorgehensweise erübrigen sich Kopplungen der Teilgebiete, wie sie bei Gebietszerlegungsmethoden erforderlich sind, oder komplizierte Interpolationen an den Interfaces. Es entsteht ein erster Algorithmus in der Art eines Zweigitterverfahrens mit zyklischer Abfolge – im Folgenden Patch-Grid-Zyklus genannt. In jedem Zeitschritt der Strömungssimulation wird die Sequenz

1. Lösen der Navier-Stokes-Gleichungen für diesen Zeitschritt
2. Transfer der Level-Set-Funktion vom groben auf das feine Gitter
3. Reinitialisierung auf dem feinen Gitter
4. Transfer vom feinen auf das grobe Gitter

durchgeführt (siehe auch Abbildung 4.17). Der Algorithmus 2 (mit den Unterroutinen 3,

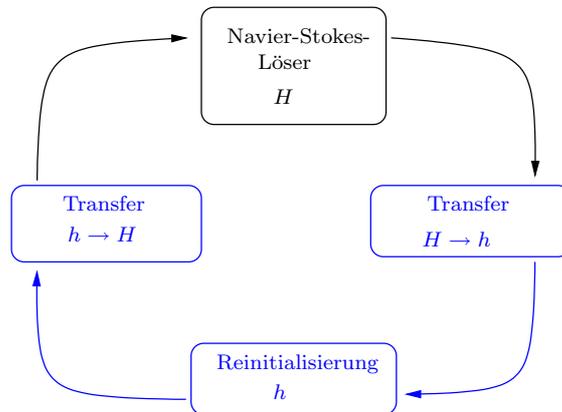


Abbildung 4.17: In jedem Zeitschritt der Strömungssimulation wird ein Patch-Grid-Zyklus durchgeführt. Der Algorithmus 2 ist durch die blau hervorgehobenen Komponenten definiert. Er ist eine eigenständige Komponente neben dem Navier-Stokes-Löser.

4, 5 und 6) beschreibt die Reinitialisierung auf patch grids instruktiv als Funktion `reinitPatchGrid`. Sie ist neben dem Navier-Stokes-Löser eine separate Komponente des gesamten Strömungslösers und somit ein modularer Bestandteil von `NaSt3DGPF`; sie benötigt als Eingabe weder Informationen über Strömungsgrößen noch verändert sie diese.

Algorithmus : `reinitPatchGrid(G_H, L, ν, ϵ)`

Input : Grobgitter G_H , gestörte Abstandsfunktion L , Irregularität ν , Bandbreite ϵ

Output : L ist Abstandsfunktion in einem ϵ -Band um die zu Beginn gegebene Nullkontur.

```

adaptGrid( $G_H, L, \nu, \epsilon, PG$ );          /* Konstruiere das patch grid  $PG$  */
transferUp( $L, PG$ );                      /* Transferiere Daten auf das patch grid. */
reinitPatches( $PG$ );                     /* Reinitialisiere auf dem patch grid. */
transferDown( $L, PG$ );                    /* Transferiere Daten auf das grobe Gitter. */
    
```

Algorithmus 2 : Ein Patch-Grid-Zyklus

4.3 Numerik

Dieser Abschnitt wendet sich der Beschreibung der wesentlichen numerischen Komponenten des Patch-Grid-Zyklus zu. Zu diesen zählen neben der eigentlichen Reinitialisierung die Transferoperatoren und die Randbedingungen (siehe auch Abbildung 4.17). Ein elementares Merkmal eines Patch-Grid-Zyklus ist der Gebrauch von zwei Gittern

Algorithmus : $\text{adaptGrid}(G_H, L, \iota, \epsilon, PG)$

Input : Grobgitter G_H , Level-Set-Funktion auf grobem Gitter L , Irregularität ι , Bandbreite ϵ , patch grid PG

Output : Ein patch grid PG mit der Eigenschaft: Innerhalb des ϵ -Bandes um die Nullkontur von L sollen Grobgitterzellen mit grid patches der Irregularität ι versehen werden; grid patches außerhalb des ϵ -Bandes sollen entfernt werden. Wende ein Verfeinerungskriterium refine_ϵ oder $\widehat{\text{refine}}_\epsilon$ an (siehe Abschnitt 4.2.1).

foreach $c \in G_H$ **do**

if $\text{refine}(c) == 1$ **then**

if c besitzt kein grid patch **then**

 └ erzeuge ein grid patch und assoziiere es mit c

else

if c besitzt ein grid patch **then**

 └ entferne das grid patch von c

Algorithmus 3 : Der Patch-Manager passt das patch grid der Nullkontur an.

Algorithmus : $\text{transferUp}(L, PG)$

Input : Level-Set-Funktion auf grobem Gitter L , patch grid PG

Output : Der Patch-Manager transferiert die Level-Set-Funktion vom groben Gitter auf das patch grid.

Algorithmus 4 : Der Patch-Manager transferiert die Level-Set-Funktion vom groben Gitter auf das patch grid.

Algorithmus : $\text{transferDown}(L, PG)$

Input : Level-Set-Funktion auf grobem Gitter L , patch grid PG

Output : Der Patch-Manager transferiert die Level-Set-Funktion vom patch grid zurück auf das grobe Gitter.

Algorithmus 5 : Der Patch-Manager transferiert die Level-Set-Funktion vom patch grid zurück auf das grobe Gitter.

Algorithmus : $\text{reinitPatches}(PG)$

Input : patch grid PG

Output : Die grid patches des patch grid PG repräsentieren eine Abstandsfunktion zu der zuvor gegebenen Nullkontur.

Algorithmus 6 : Der Patch-Manager führt die Reinitialisierung auf den grid patches durch.

unterschiedlicher Maschenweite. Das grobe Gitter G_H trägt alle Größen des Strömungslösers sowie die eigentliche Level-Set-Funktion. Auf ihm operiert der Strömungslöser; der Transport der Level-Set-Funktion gemäß Gleichung 2.17 findet ebenfalls auf G_H statt. Da nicht die Diskretisierungsgenauigkeit, sondern die Reinitialisierung auf G_H insuffizient ist, soll lediglich die Reinitialisierung auf dem feinen Gitter G_h durchgeführt werden. Dazu muss in jedem Zeitschritt des Strömungslösers, an den sich die Reinitialisierung anschließt, zunächst ein Transfer der Level-Set-Funktion von G_H nach G_h stattfinden. Es ist ebenfalls notwendig, die auf G_h reinitialisierte Level-Set-Funktion, durch einen Transfer von G_h nach G_H dem Strömungslöser wieder zur Verfügung zu stellen.

Abschnitt 4.2.3 zeigt, dass sich benachbarte grid patches nahtlos aneinanderfügen und auf diese Weise lokal wie ein einziges strukturiertes Gitter erscheinen. Aus diesem Grund können auf jedem Freiheitsgrad innerhalb des relevanten Gitters herkömmliche Finite-Differenzen-Schemata zur Berechnung der Differentialoperatoren angewendet werden.

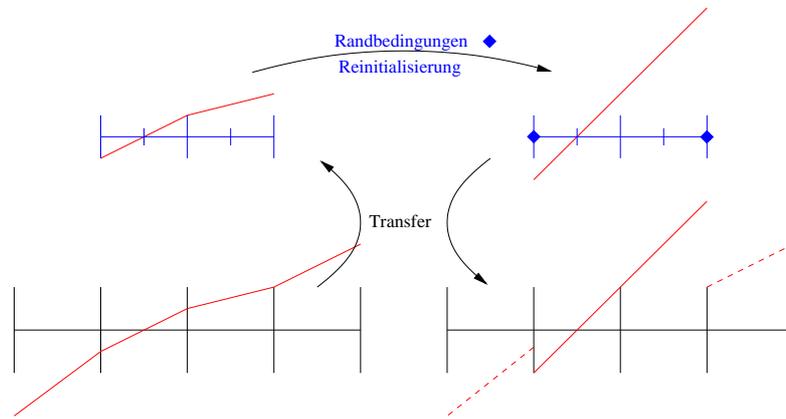


Abbildung 4.18: Ein eindimensionales Beispiel für einen Patch-Grid-Zyklus: Auf dem groben Gitter liegt eine gestörte Abstandsfunktion (rot) vor. Diese wird auf das relevante Gitter ${}^P G_r^\epsilon$ (blau) transferiert. Es werden Randbedingungen \blacklozenge am Rand von ${}^P G_r^\epsilon$ definiert und anschließend die Abstandseigenschaft der Level-Set-Funktion durch eine Reinitialisierung auf dem relevanten Gitter ${}^P G_r^\epsilon$ wiederhergestellt. Den letzten Schritt im Zyklus stellt der Transfer der Abstandsfunktion vom patch grid ${}^P G_r^\epsilon$ auf ${}^H G_r^\epsilon$ – die von grid patches überdeckten Grobgitterzellen – dar.

Der Entwurf eines Algorithmus für eine verbesserte Level-Set-Reinitialisierung in 4.2.5 sieht in einem seiner Teilschritte eine Reinitialisierung auf dem feinen relevanten Gitter unabhängig und entkoppelt vom groben Gitter vor; das bedeutet, dass der Algorithmus während der Reinitialisierung nur das relevante Gitter sieht und daher an dessen Rand eigene Randbedingungen definiert werden müssen (siehe auch Abbildung 4.18). In den folgenden Abschnitten werden die Transferoperatoren und Randbedingungen genauer beschrieben.

4.3.1 Transferoperatoren

Transferoperatoren dienen ganz allgemein dem Transport von Information zwischen den Gittern. Speziell in diesem Zweigitterverfahren geht es um den Transport zwischen zwei Gittern mit den signifikant unterschiedlichen Maschenweiten H und h . Daher finden im Folgenden die dem Mehrgitterverfahren entlehnten Begriffe Prolongation für den Transport von G_H nach ${}^{\text{P}}G_{\text{r}}^{\epsilon}$ und Restriktion für den Transport in entgegengesetzter Richtung von ${}^{\text{P}}G_{\text{r}}^{\epsilon}$ nach G_H Verwendung.

Prolongation

Der Transfer bei der Prolongation geschieht mittels polynomieller Interpolation; dabei erfolgt an der Position jedes Freiheitsgrades des relevanten Gitters ${}^{\text{P}}G_{\text{r}}^{\epsilon}$ eine Auswertung des Polynoms, welches die Level-Set-Funktion an benachbarten Punkten des groben Gitters G_H interpoliert (siehe Abbildung 4.19). Eine wesentliche Größe stellt hier

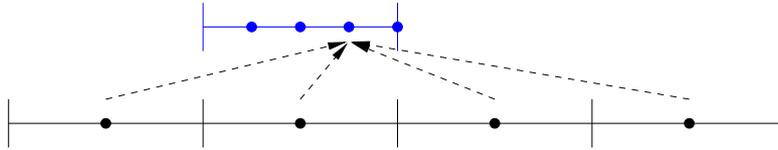


Abbildung 4.19: Ein eindimensionales Beispiel für einen Prolongationsoperator: Ein Freiheitsgrad eines grid patches wird mittels Polynominterpolation vierter Ordnung aus den vier umliegenden Freiheitsgraden des groben Gitters G_H interpoliert.

die Interpolationsordnung dar. Sie nimmt nicht nur Einfluss auf die Stabilität des Verfahrens, sondern muss der Reinitialisierung auf ${}^{\text{P}}G_{\text{r}}^{\epsilon}$ auch hinreichend gute Daten zur Verfügung stellen. Besonders wichtig ist, dass die Prolongation keine numerische Diffusion verursacht, die Nullkontur der Level-Set-Funktion also auf ${}^{\text{P}}G_{\text{r}}^{\epsilon}$ mit der auf ${}^{\text{H}}G_{\text{r}}^{\epsilon}$ identisch ist. Ein weiterer wichtiger Aspekt ist die Tatsache, dass der Gradient der prolongierten Daten bei Verwendung niedriger Interpolationsordnung starke Oszillationen aufweist, welche die Reinitialisierung in Bezug auf die Massenerhaltung erheblich verschlechtern. Im Abschnitt 5.1.1 werden Prolongationsoperatoren verschiedener Ordnung an unterschiedlichen Testfällen hinsichtlich dieser Aspekte untersucht. Von nun an bezeichne $\text{trans}_h(\phi_H)$ die von ${}^{\text{H}}G_{\text{r}}^{\epsilon}$ nach ${}^{\text{P}}G_{\text{r}}^{\epsilon}$ transferierte Funktion ϕ_H .

Restriktion

Die Restriktion hingegen ist stets trivial, da jeder Freiheitsgrad in ${}^{\text{H}}G_{\text{r}}^{\epsilon}$ einen korrespondierenden Freiheitsgrad in ${}^{\text{P}}G_{\text{r}}^{\epsilon}$ hat; das heißt genauer: Der Freiheitsgrad von Zelle $c \in {}^{\text{H}}G_{\text{r}}^{\epsilon}$ besitzt dieselben Ortskoordinaten wie der Freiheitsgrad mit dem Index

$(i, j, k) = (2^{l-1}, 2^{l-1}, 2^{l-1})$ im grid patch $\text{ass}(c)$ (siehe auch Definition 4.2 und Abbildung 4.16). Aus diesem Grund bilden Prolongation und Restriktion, nacheinander ausgeführt, die Identität. Von nun an bezeichne $\text{trans}_H(\phi_h)$ die von ${}^P G_r^\epsilon$ nach ${}^H G_r^\epsilon$ transferierte Funktion ϕ_h . Es gilt also $\text{trans}_H(\text{trans}_h(\phi_H)) = \text{Id}(\phi_H)$.

4.3.2 Randbedingungen

Es stehen im Wesentlichen drei Arten von Randbedingungen für die Reinitialisierung beziehungsweise die Hamilton-Jacobi-Iteration zur Wahl. Das sind:

- Neumann'sche Randbedingungen
- Extrapolation: lineare Fortsetzung
- Rauhe Randbedingungen

Im Fall von Neumann'schen Randbedingungen wird der Wert des äußersten Freiheitsgrades, der noch in ${}^P G_r^\epsilon$ liegt, in die Ghostzellen kopiert, so dass der Gradient über den Rand hinweg verschwindet. Die lineare Fortsetzung hingegen setzt den Gradienten über den Rand hinaus fort und lässt diesen damit unverändert. Daher erhält man glatte Daten am Rand, muss jedoch auch in Kauf nehmen, dass große Gradienten am Rand zu extremen Werten in den Ghostzellen führen. Bei rauhen Randbedingungen werden sehr große Werte (etwa 10^{15}) mit alternierendem Vorzeichen in die Ghostzellen gesetzt mit dem Ziel, die Upwind-Schemata ENO/WENO dazu zu bringen, Informationen aus dem Inneren des Gebiets zu holen.

Kapitel 5

Numerische Ergebnisse

In diesem Kapitel soll die Korrektheit und Tauglichkeit der neuen adaptiven Hamilton-Jacobi-Level-Set-Reinitialisierung als Zweigitterverfahren auf patch grids ausgiebig getestet und verifiziert werden. Dazu folgen nun ein paar Vereinbarungen und Definitionen bezüglich der angewandten Indizes, Normen und Messverfahren der anschließenden Kapitel: Von nun an bezeichnen hochgestellte Indizes, wie in ϕ^n oder ϕ^t , Iterationsschritte oder Zeiteinheiten, tiefgestellte Indizes dienen hingegen der räumlichen Bestimmung. So steht zum Beispiel ϕ_i^n für den Wert von ϕ im Punkt i nach n Iterationen. Oftmals wird ϕ^* für eine exakte Lösung oder eine hinreichend genaue Referenzlösung verwendet. Ein tiefgestelltes H oder h kann auch zur leichteren Unterscheidung verdeutlichen, dass eine diskrete Funktion auf dem Gitter mit der Maschenweite H oder h definiert ist. Mit der l_p -Norm aus Definition 5.1 kann den diskreten Funktionen des Gitters ein Messwert zugeordnet werden. Betrachtet man die diskrete Funktion ϕ des Gitters G als eine Funktion aus L_p , die auf jeder Gitterzelle $c \in G$ konstant ist, so stimmt die l_p -Norm mit der L_p -Norm überein.

Definition 5.1 [l_p -NORM]

Sei G ein Gitter mit den Gitterzellen $c \in G$, und sei ϕ eine Funktion auf G , wobei $\phi(c)$ der Wert von ϕ auf dem Freiheitsgrad von Zelle c und $V(c)$ das Volumen von c ist, dann sei

$$\|\phi\|_{l_p} := \sqrt[p]{\sum_{c \in G} |\phi(c)|^p \cdot V(c)} .$$

Mithilfe dieser Norm lässt sich nun der Begriff des relativen Fehlers definieren.

Definition 5.2 [RELATIVER FEHLER]

Der relative Fehler e der (diskreten) Funktion ϕ in Bezug auf die Funktion ϕ^* sei wie folgt definiert:

$$e := \frac{\|\phi - \phi^*\|_{l_p}}{\|\phi^*\|_{l_p}}$$

Von zentraler Bedeutung in dieser Arbeit ist der Begriff der Masse: Falls die Nullkontur der Level-Set-Funktion eine geschlossene Fläche beschreibt, steht der Begriff der Masse für das von der Nullkontur der Level-Set-Funktion eingeschlossene Volumen. Andernfalls

teilt die Nullkontur das Gebiet in zwei oder mehrere Teilgebiete auf, dann sei die Masse als das Volumen der Teilgebiete definiert, auf denen die Level-Set-Funktion ein positives (oder negatives) Vorzeichen hat. Numerisch ermittelt man die Masse, indem unter Verwendung einer numerischen Quadratur das Integral über $H_\epsilon(\phi)$ berechnet wird, wobei H_ϵ die regularisierte Heavyside'sche Gewichtsfunktion ist.

Definition 5.3 [MASSE]

Die Masse der Level-Set-Funktion ϕ auf dem Gitter G ist definiert als

$$m := \mathcal{Q}_G(H_\epsilon(\phi)),$$

wobei H_ϵ die regularisierte Heavyside'sche Gewichtsfunktion und \mathcal{Q}_G eine Quadraturformel auf dem Gitter G ist.

Definition 5.4 [REGULARISIERTE HEAVYSIDE'SCHE GEWICHTSFUNKTION]

Die regularisierte Heavyside'sche Gewichtsfunktion ist definiert als

$$H_\epsilon(\phi) := \begin{cases} 0 & , \text{ falls } \phi < -\epsilon \\ \frac{1}{2} \left[1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right) \right] & , \text{ falls } |\phi| \leq \epsilon \\ 1 & , \text{ falls } \phi > \epsilon \end{cases}$$

Weiterhin von großem Interesse ist der bisher oft erwähnte Massenverlust durch numerische Diffusion (siehe Abschnitt 4.1 ab Seite 57). Um diesen quantifizieren zu können, wird der relative Massenverlust m_Δ bezogen auf eine ursprüngliche Masse m_0 betrachtet:

Definition 5.5 [RELATIVER MASSENVERLUST]

Sei m_0 die ursprüngliche Masse, dann sei der relative Massenverlust zum Zeitpunkt t beziehungsweise Iterationsschritt n definiert als

$$m_\Delta^{t,n} := \frac{m^{t,n} - m_0}{m_0}$$

mit $m^{t,n}$ als Masse zum Zeitpunkt t beziehungsweise Iterationsschritt n .

Je nach Problemstellung kann die ursprüngliche Masse m_0 direkt analytisch oder aber hinreichend genau auf einem besonders feinen Gitter berechnet werden. Im Allgemeinen bedeutet dies, dass der relative Massenverlust m_Δ^0 schon zum Zeitpunkt 0 von 0 verschieden ist und als Diskretisierungsfehler auf dem Gitter G angesehen werden kann. In den Testfällen, in denen keine hinreichend genaue Beschreibung der Anfangsbedingungen vorliegt – zum Beispiel bei Strömungsproblemen, deren freie Oberfläche komplexe Geometrien aufweist – wird die ursprüngliche Masse m_0 mit der berechneten Masse m^0 zum Zeitpunkt 0 gleichgesetzt.

Vor der Untersuchung des gesamten Algorithmus sollen zunächst dessen Einzelkomponenten Prolongation, Reinitialisierung und Restriktion unter verschiedenen Bedingungen untersucht werden.

5.1 Diskretisierungsfehler

Dieser Abschnitt soll die Frage nach der Qualität der prolongierten Daten beantworten, denn diese sind das Ausgangsmaterial der Reinitialisierung auf dem feinen Gitter. Es ist daher von großer Bedeutung, dass der Transfer die Nullkontur unverändert lässt, da eine anfängliche Störung in der Nullkontur nicht wiederhergestellt werden kann. Darüber hinaus soll der Transfer den Gradienten der Level-Set-Funktion möglichst unverändert und störungsfrei belassen. Wählt man nun eine analytisch definierte Level-Set-Funktion ϕ^* , beispielsweise die Abstandsfunktion zu einer Sphäre und wertet diese auf dem groben Gitter G_H aus, so erhält man $\phi_H := \phi^*|_{G_H}$; anschließend werden die Daten vom Grobgitter des relevanten Gebiets ${}^H G_r^\epsilon$ auf die grid patches ${}^P G_r^\epsilon$ transferiert, so dass sich $\phi_h := \text{trans}_h(\phi_H)$ ergibt. ϕ_h wird in diesem Abschnitt Gegenstand näherer Untersuchungen:

5.1.1 Prolongation und Restriktion

Nun soll gemessen werden, wie gut ϕ_h die analytische Funktion ϕ^* in Abhängigkeit von der Interpolationsordnung der Prolongation approximiert. Sei ϕ^* die Abstandsfunktion zu einer Sphäre, dann kann der Fehler $(\phi^*|_{{}^P G_r^\epsilon} - \phi_h)$ in der ℓ_2 -Norm berechnet werden. Wie erwartet, ist der Fehler bei trilinearer Interpolation von zweiter Ordnung $\mathcal{O}(H^2)$ und bei trikubischer Interpolation von vierter Ordnung $\mathcal{O}(H^4)$ (vergleiche auch mit Ab-

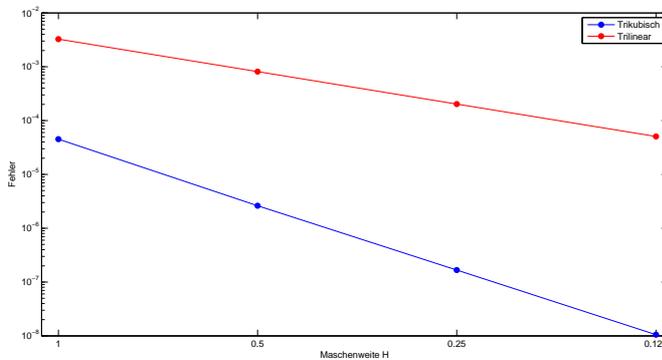


Abbildung 5.1: Die Abstandsfunktion einer Sphäre ϕ^* wird zunächst auf dem groben Gitter G_H ausgewertet und anschließend auf die grid patches ${}^P G_r^\epsilon$ transferiert. Die auf diese Weise gewonnene Funktion $\phi_h = \text{trans}_h(\phi^*|_{G_H})$ approximiert ϕ^* . Die Graphik zeigt den Fehler $(\phi^*|_{{}^P G_r^\epsilon} - \phi_h)$ in der ℓ_2 -Norm für trilineare und trikubische Interpolation in Abhängigkeit der Maschenweite H des groben Gitters. Die Raten sind $\mathcal{O}(H^2)$ (trilinear) und $\mathcal{O}(H^4)$ (trikubisch).

bildung 5.1). Abbildung 5.2a zeigt deutlich, dass die Schnittebene von ϕ_h zwischen je

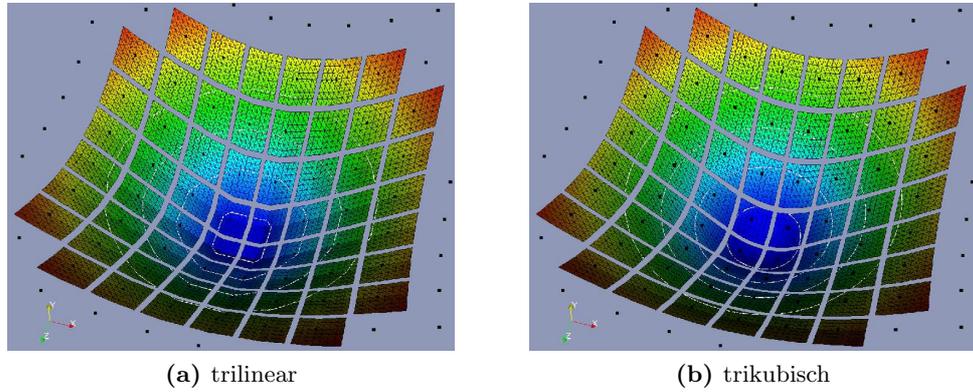


Abbildung 5.2: Die prolongierte Level-Set-Funktion $\phi_h = \text{trans}_h(\phi^*|_{G_H})$: Prolongation mittels trilinearer (a) und trikubischer (b) Interpolation. Zu sehen ist eine Schnittebene durch den Äquator der Nullkontur (Sphäre) von ϕ^* . Zur besseren Illustration ist die Schnittebene abhängig vom Wert der Funktion ϕ_h eingefärbt und deformiert. Ferner sind in weiß einige weitere Kontouren von ϕ_h (eingeschränkt auf die Schnittebene) eingezeichnet. Die schwarzen Punkte bezeichnen die Freiheitsgrade des groben Gitters G_H .

vier der Freiheitsgrade von G_H bilinearen Verlauf hat, was auf die trilineare Interpolation zurückzuführen ist. Ebenso sind die Konturen lediglich linear rekonstruiert. Es wird deutlich, dass eine quadratische Funktion (Polynom dritter Ordnung) wie die Abstandsfunktion einer Sphäre nicht hinreichend genau durch das patch grid approximiert werden kann, wenn die Daten vom groben Gitter mit nur zweiter Ordnung transferiert werden. Dabei ist es unerheblich, wie fein die Maschenweite h der grid patches ist. Mit anderen Worten führt ein erhöhter Aufwand nicht zu akkurateren Resultaten. Erst bei Anwendung von Interpolationen, deren Ordnung größer oder gleich 3 ist, kann die Abstandsfunktion der Sphäre exakt auf das feinere Gitter übertragen werden, so dass nun jede Verkleinerung der Feingittermaschenweite zu besseren Approximationen der Abstandsfunktion auf dem feinen Gitter führt. So sieht man in Abbildung 5.2b, dass die Prolongation vierter Ordnung in der Lage ist, die Abstandsfunktion auf den grid patches ${}^{\text{PG}}_f^\epsilon$ zu rekonstruieren. Es folgt ein weiteres Beispiel, in dem mit ϕ^* eine Level-Set-Funktion definiert wird, die aus einer Abstandsfunktion zu einer Ebene durch Skalierung mit einer kubischen Funktion (Polynom vierter Ordnung) entsteht. Daher ergibt sich, dass bei trikubischer Interpolation der kubische Verlauf der Konturen exakt rekonstruiert werden kann, während bei trilinearer Interpolation diese Information verloren geht.

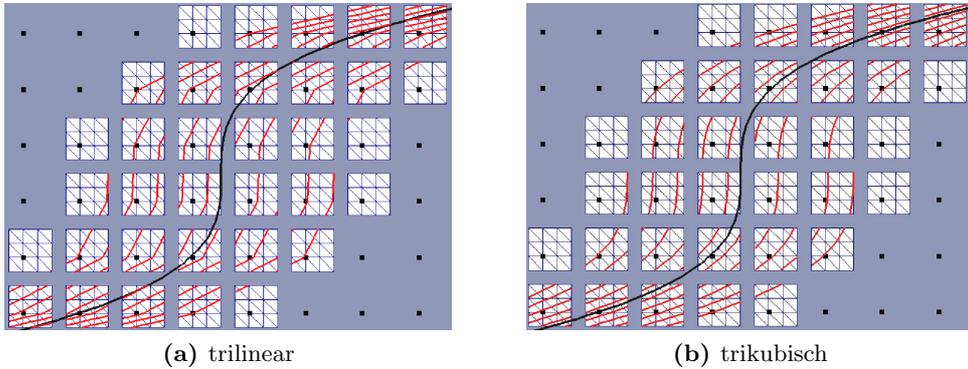


Abbildung 5.3: Die prolongierte Level-Set-Funktion $\phi_h = \text{trans}_h(\phi^*|_{G_H})$ im zweidimensionalen Schnitt: Die Prolongation von ϕ^* wird mittels trilinear (a) und trikubischer (b) Interpolation durchgeführt. Die schwarze Kurve stellt die Nullkontur von ϕ^* dar, die schwarzen Punkte bezeichnen die Freiheitsgrade des groben Gitters G_H . Auf den grid patches in rot sind mehrere Konturen von ϕ_h zu sehen.

5.1.2 Differentialoperatoren

Im vorhergehenden Abschnitt wurde deutlich, wie gut eine auf dem groben Gitter ausgewertete Funktion nach der Prolongation auf dem relevanten feinen Gitter ${}^pG_r^\epsilon$ in Abhängigkeit der Transferoperatoren approximiert werden kann. Da in der Hamilton-Jacobi-Iteration 2.16 (auf Seite 30) der Gradient der Level-Set-Funktion ausgewertet wird, stellt sich die Frage, ob dieser akkurat auf das feine Gitter transferiert wird. Eine erste Beobachtung zeigt, dass der Gradient im Wesentlichen unabhängig vom herangezogenen Schema (ENO/WENO) auf dem feinen Gitter ${}^pG_r^\epsilon$ zu Oszillationen neigt. Die Abbildung 5.4 zeigt diese Oszillationen anhand eines Beispiels: ϕ^* sei die Abstandsfunktion einer Kugel. ϕ^* wird auf dem groben Gitter G_H ausgewertet und auf das feine Gitter ${}^pG_r^\epsilon$ übertragen, was $\phi_h = \text{trans}_h(\phi^*|_{G_H})$ liefert. Da ϕ^* eine Abstandsfunktion ist, erfüllt diese die Eikonalgleichung $|\nabla\phi^*| = 1$; somit ist $1 - |\nabla\phi_h|$ der Fehler von $|\nabla\phi_h|$. Dieser Fehler oszilliert besonders stark an den Stellen, an denen $\nabla\phi_h$ zu keiner der Raumachsen parallel ist. Eine Messung von $1 - |\nabla\phi_h|$ in der l_2 -Norm in Abhängigkeit von der Grobgittermaschenweite H und der Interpolationsordnung beim Transfer gibt Aufschluss darüber, welchen Einfluss die Grobgitterwerte und die Transferoperatoren auf die Qualität der Daten auf dem feinen Gitter haben. Die Abbildung 5.5 zeigt den Fehler $1 - |\nabla\phi_h|$ in der l_2 -Norm in Abhängigkeit von der Grobgittermaschenweite H und der verwendeten Interpolationsordnung. Wie zu erwarten sind die Fehler von zweiter Ordnung $\mathcal{O}(H^2)$ bei trilinearer und vierter Ordnung $\mathcal{O}(H^4)$ bei trikubischer Interpolation.

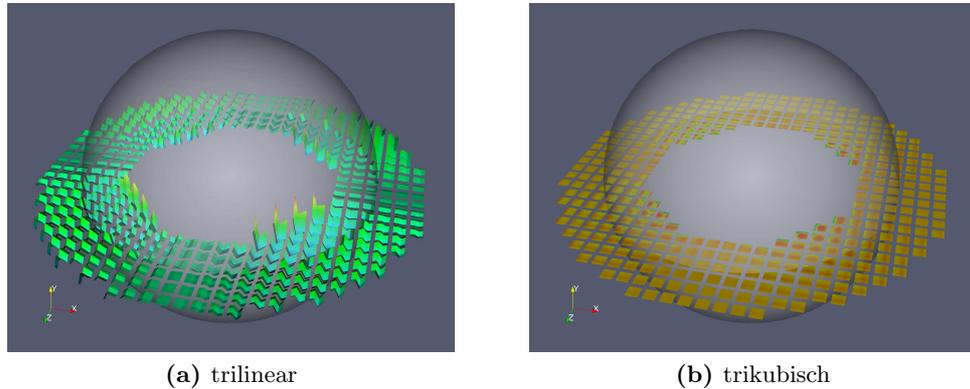


Abbildung 5.4: ϕ^* sei die Abstandsfunktion einer Sphäre, $\phi_h = \text{trans}_h(\phi^*|_{G_H})$ die vom groben aufs feine Gitter ${}^pG_r^e$ transferierte Level-Set-Funktion. Die Abbildung zeigt den Fehler von $|\nabla\phi_h|$ nämlich $1 - |\nabla\phi_h|$ auf den grid patches im zweidimensionalen Schnitt durch den Äquator der Sphäre. Die Prolongation von ϕ^* wurde mittels trilinearere (a) und trikubischer (b) Interpolation durchgeführt. Zur Illustration ist die Schnittebene abhängig von den Werten von e eingefärbt und deformiert. Auffällig in Bild (a) sind starke Oszillationen des Fehlers in diagonaler Richtung zu den Gitterachsen.

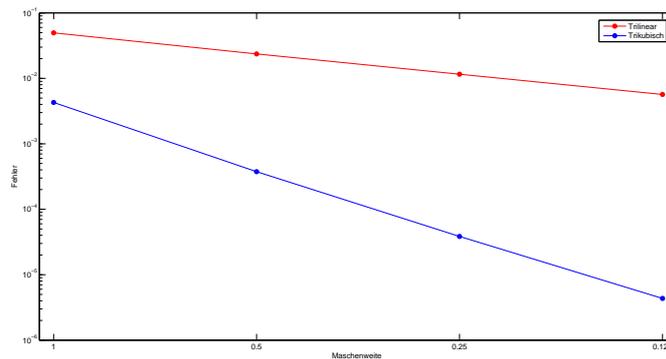


Abbildung 5.5: Der Fehler $1 - |\nabla\phi_h|$ in der l_2 -Norm in Abhängigkeit von der Grobgittermaschenweite nach trilinearem (rot) und trikubischen (blau) Transfer. Die Raten sind $\mathcal{O}(H^2)$ (trilinear) und $\mathcal{O}(H^4)$ (trikubisch).

5.2 Reinitialisierung

Das Kernstück des neuen Zweigitterverfahrens ist die Reinitialisierung. Es gibt im Wesentlichen drei Eigenschaften, auf die die Reinitialisierung hin untersucht werden soll. Das sind die numerische Diffusion, die Erhaltung der Nullkontur und die Erhaltung der geometrischen Form. Da sich die numerische Diffusion in einer Verschiebung der Nullkon-

tur entgegen der Krümmungsrichtung der Oberfläche äußert, wird zu ihrer quantitativen Beurteilung der Verlauf der Masse einer von der Nullkontur umschlossenen Teilmenge des Gebiets über die Anzahl der Hamilton-Jacobi-Iterationen gemessen. Damit sich die numerische Diffusion an jeder Stelle der Oberfläche stets mit demselben Vorzeichen auf die Masse auswirkt, dürfen nur solche Level-Set-Funktionen betrachtet werden, deren Krümmung der Nullkontur stets in das Innere des von ihr umschlossenen Gebiets zeigt. Im zweiten Unterabschnitt wird untersucht, wie gut das Zweigitterverfahren die Nullkontur während der Reinitialisierung erhält. Um diesen Effekt zu messen, wird zu einer vorgegebenen Nullkontur eine stark gestörte Abstandsfunktion reinitialisiert und der Fehler der gewonnenen Level-Set-Funktion im Vergleich zu einer analytischen Lösung beobachtet. Dazu wird auch der Testfall von Smereka und Russo aus [44] in einer modifizierten Form angewendet. Ein Test zur Formerhaltung wird an Nullkonturen durchgeführt, die starke Krümmungen und filigrane Strukturen aufweisen.

5.2.1 Numerische Diffusion

Zur Messung der numerischen Diffusion wird eine Sphäre mit unterschiedlichen Radien mit zugehöriger Abstandsfunktion als Startwert einer Level-Set-Funktion herangezogen und der Massenverlust in Abhängigkeit der Anzahl an Reinitialisierungen beziehungsweise Hamilton-Jacobi-Iterationen beobachtet. Dabei liegt das Hauptaugenmerk auf dem Vergleich zwischen dem herkömmlichen Verfahren und dem neuen Zweigitterverfahren.

Zunächst werden zwei Sphären mit den Radien von 5 und 2.5 Grobgittermaschenweiten betrachtet: $r = 5H$ und $r = 2.5H$. Die Graphik 5.6 zeigt den Verlauf des Massenverlustes des herkömmlichen Verfahrens im Vergleich zum Zweigitterverfahren mit der Irregularität $\iota = 2$ beziehungsweise mit der Feingittermaschenweite $h := H/4$. Erwartungsgemäß ist der Massenverlust im Zweigitterverfahren wesentlich geringer als beim herkömmlichen, uniformen Verfahren. Zum Vergleich zeigt die Graphik ferner den Massenverlust des uniformen Verfahrens mit der Grobgittermaschenweite $H/4$. Bei diesen Tests umfasst die Bandbreite des relevanten Gebiets ${}^{\text{p}}G_{\tau}^{\epsilon}$ in Fall $r = 5H$ doppelt so viele Grobgitterzellen wie im Fall $r = 2.5H$. Ferner sind im adaptiven Zweigitterverfahren aufgrund der CFL-Bedingung pro Reinitialisierungsschritt die vierfache Anzahl an Hamilton-Jacobi-Iterationen nötig, um die Reinitialisierung auf der gleichen Bandbreite durchzuführen wie es im uniformen Fall mit der Maschenweite H notwendig ist. Bezogen auf die Anzahl an Hamilton-Jacobi-Iterationen kann daher mit dem adaptiven Zweigitterverfahren ein noch besserer Massenerhalt erzielt werden (siehe dazu Abbildungen 5.6b und 5.6d). In der Anwendung des adaptiven Verfahrens in Zweiphasenströmungssimulationen ist man jedoch unabhängig vom Level-Set-Verfahren darauf angewiesen, die Abstandseigenschaft in einem konstant großen Bereich um die Nullkontur herum herzustellen. Das bedeutet, dass der Nachteil des adaptiven Zweigitterverfahrens darin liegt, dass pro Reinitialisierung stets die 2^{ι} -fache Anzahl an Hamilton-Jacobi-Iterationen durchgeführt werden muss. Daher bildet im Folgenden nicht mehr die Anzahl an Hamilton-Jacobi-Iterationen,

sondern stets die Anzahl an Reinitialisierungen den Bezugspunkt.

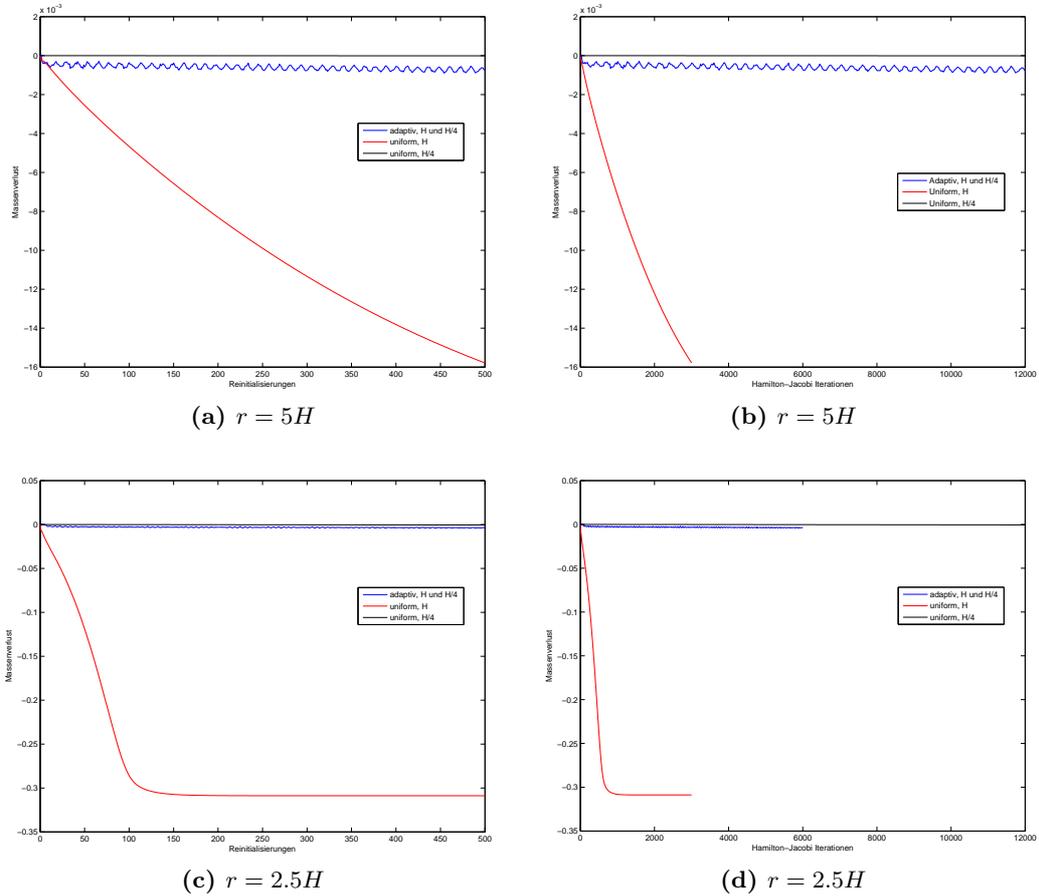


Abbildung 5.6: Der relative Massenverlust nach zahlreichen Reinitialisierungen: Eine Level-Set-Funktion, die bereits die Abstandseigenschaft einer Kugel besitzt, wird wiederholt dem Reinitialisierungsprozess unterworfen. Die Graphiken zeigen den relativen Massenverlust bezogen auf die Anfangsmasse m_0 der Kugel der auf dem groben Gitter ausgewerteten Abstandsfunktion. In (a) und (b) betragt der Radius 5 Grobgittermaschenweiten, in (c) und (d) 2.5 Grobgittermaschenweiten. Die Graphiken (a) und (c) tragen den Massenverlust gegen die Anzahl der Reinitialisierungen, (b) und (d) gegen die Anzahl der Hamilton-Jacobi-Iterationen auf. Jede Graphik zeigt den relativen Massenverlust fur das herkommliche, uniforme Verfahren mit der Gittermaschenweite H (rot), das adaptive Zweigitterverfahren mit der Grobgittermaschenweite H und der Feingittermaschenweite $H/4$ (blau) und nochmals das uniforme Verfahren, dessen Gittermaschenweite der Feingittermaschenweite des adaptiven Verfahrens entspricht – namlich $H/4$ (schwarz).

Dieses Beispiel zeigt die Abhangigkeit der numerischen Diffusion von der Krummung

der Nullkontur. Daher muss zu einer quantitativen Beurteilung des adaptiven Zweigitterverfahrens stets die Krümmung der Nullkontur berücksichtigt werden. Im folgenden Unterabschnitt wird das uniforme Verfahren gegen das adaptive Zweigitterverfahren mit unterschiedlichen Feingittermaschenweiten beziehungsweise Irregularitäten ν an Nullkonturen unterschiedlicher Krümmung getestet.

Numerische Diffusion in Abhängigkeit des Krümmungsradius

Als Nullkontur dient wieder eine Sphäre mit verschiedenen Radien mit zugehöriger Abstandsfunktion. Die Messergebnisse zeigen nun den relativen Massenverlust bezogen auf die Anfangsmasse, die auf dem groben Gitter vorliegt. In Tabelle 5.1 sind die relativen

Radius	uniform H	Adaptiv		
		$H, H/4$	$H, H/8$	$H, H/16$
1.25H	-7.92 e-2	1.59 e-2	1.46 e-2	-3.80 e-3
	-5.64 e-1	1.60 e-2	1.45 e-2	8.60 e-3
	-5.73 e-1	-2.89 e-3	8.77 e-3	1.10 e-2
	-5.76 e-1	-1.84 e-2	2.11 e-3	1.01 e-2
2.50H	-5.66 e-3	1.29 e-3	8.71 e-4	-3.60 e-4
	-17.35 e-2	-5.84 e-4	6.77 e-4	3.05 e-4
	-7.75 e-2	-1.871 e-3	-8.20 e-6	4.57 e-4
	-1.89 e-1	-4.69 e-3	-1.60 e-4	2.31 e-4
5.00H	-2.89 e-4	1.25 e-4	5.30 e-5	-5.00 e-6
	-3.08 e-4	-1.95 e-4	-9.00 e-6	1.01 e-4
	-7.39 e-4	-4.89 e-4	-2.43 e-4	1.04 e-4
	-1.23 e-3	-8.17 e-4	-2.00 e-4	1.14 e-4

Tabelle 5.1: Der relative Massenverlust in Abhängigkeit von der Krümmung der Nullkontur: Die Abstandsfunktionen zu den Sphären mit den Radien $r = 1.25H$, $r = 2.5H$ und $r = 5H$ werden zahlreichen Reinitialisierungen unterworfen. In den Spalten stehen die relativen Massenverluste bezogen auf die Anfangsmasse, die auf dem groben Gitter gemessen wird, für das herkömmliche, uniforme Verfahren mit der Maschenweite H und für das adaptive Zweigitterverfahren mit den Feingittermaschenweiten $H/4$, $H/8$ und $H/16$. Die Zeilen zeigen für jeden Radius einen Messwert jeweils nach der 1., der 10., der 50. und der 100. Reinitialisierung. Bei Werten mit gleicher Farbe entspricht der Radius immer der gleichen Anzahl an (Fein)Gittermaschenweiten: ■ $\cong 5$, ■ $\cong 10$, ■ $\cong 20$, ■ $\cong 40$, ■ $\cong 80$.

Massenverluste nach jeweils 1, 10, 50 und 100 Reinitialisierungen für Sphären mit einem Radius von 1.25, 2.5 und 5 Maschenweiten aufgeführt. Es wird deutlich, dass der Massenverlust – verursacht durch numerische Diffusion – stark von der Krümmung der Nullkontur abhängt: Der Verlust ist umso größer, je stärker die Krümmung beziehungsweise

je kleiner der Radius ist. Eine Verbesserung durch die Adaptivität ist klar erkennbar. So erleidet die Sphäre mit Radius $r = 1.25H$ beim uniformen Verfahren schon nach nur einer Reinitialisierung mehr als 7.9% Verlust und hat nach 10 Reinitialisierungen mehr als die Hälfte ihres Volumens verloren, während das adaptive Verfahren bei einer Feingittermaschenweite von $H/4$ den Verlust selbst nach 100 Reinitialisierungen auf unter 2% halten kann. Bei größeren Radien nimmt der Massenverlust zwar ab, wird aber dennoch durch Adaptivität von mehr als 18% auf weniger als 0.5% bei einem Radius von $r = 2.5H$ und von über 0.12% auf unter 0.09% bei einem Radius von $r = 5H$ nach jeweils 100 Reinitialisierungen reduziert. Wie auch schon auf den Abbildungen 5.6 zu erkennen ist, unterliegt die Masse bei den adaptiven Zweigitterverfahren Oszillationen, deren Amplitude mit der Krümmung bezogen auf die Feingittermaschenweite wächst. Da

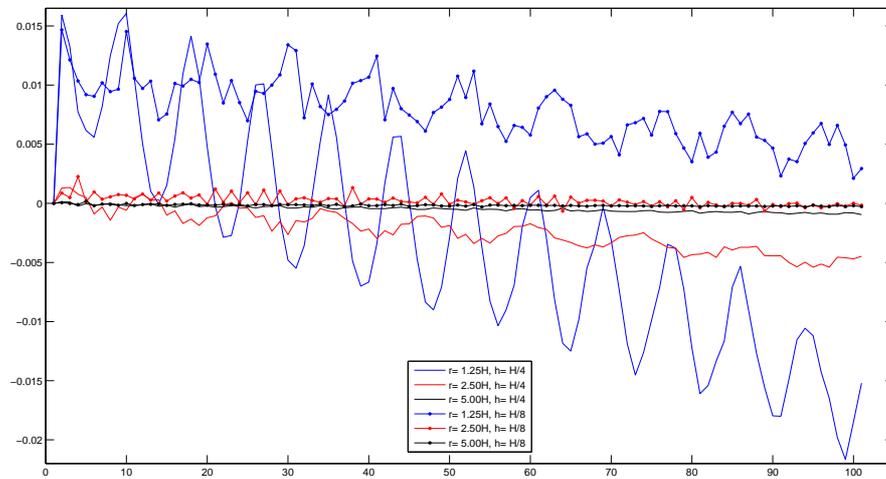


Abbildung 5.7: Oszillierende Masse bei adaptiver Reinitialisierung: Die Graphik zeigt den Verlauf der Masse von Sphären unterschiedlicher Radien in Abhängigkeit von der Anzahl der Reinitialisierungen. Die Radien sind $r = 1.25H$ (blau), $r = 2.5$ (rot) und $r = 10H$ (schwarz). Bei den einfachen Linien ist ein adaptives Verfahren mit Irregularität $\iota = 2$ und Feingittermaschenweite $h = H/4$, bei Linien mit punktförmigem Marker ein adaptives Verfahren mit Irregularität $\iota = 3$ und Feingittermaschenweite $h = H/8$ angewendet worden.

bei großen Krümmungen und einer Irregularität von $\iota = 2$ diese Oszillationen gegenüber dem eigentlichen Massenverlust sehr dominant sein können, wird die Wahl einer höheren Irregularität demnach nicht nur durch einen besseren Massenerhalt, sondern auch durch geringere Oszillationen begünstigt. Die Abbildung 5.8 visualisiert die Sphäre mit dem Radius $r = 1.25H$ nach mehreren Reinitialisierungen (1, 10, 100). In der linken Spalte der Abbildung wird das uniforme Verfahren angewendet, während die Bilder in der rechten Spalte mit dem adaptiven Zweigitterverfahren mit der Irregularität $\iota = 2$, also mit der Feingittermaschenweite $h := H/4$ gerechnet werden. Interessant ist die Tatsache, dass

sich die Nullkontur lediglich im Rahmen einer Gittermaschenweite bewegt; das heißt, die Nullkontur schmiegt sich an die in Krümmungsrichtung nächstgelegene Gitterzelle an und bewegt sich somit maximal um eine Gittermaschenweite H im uniformen Fall beziehungsweise Feingittermaschenweite h im adaptiven Fall.

Bemerkung: Der Abschnitt 4.1.2 ab Seite 66 gibt die theoretische Erklärung dafür, dass sich die Nullkontur unter Verwendung der geglätteten Signumfunktion maximal um eine Gittermaschenweite bewegt. Dass das tatsächlich so ist, zeigt die Abbildung 5.8. An dieser Stelle werden die Vorteile des adaptiven Zweigitterverfahrens deutlich, wie es im Abschnitt 4.2 motiviert worden ist:

- Da die Reinitialisierung auf einem Gitter mit signifikant kleinerer Feingittermaschenweite durchgeführt wird, ist es möglich, die Bewegung der Nullkontur auf die kleine Weite h anstatt H zu beschränken.
- Da die Krümmung der Nullkontur im adaptiven Fall relativ zur Feingittermaschenweite kleiner ist als zur Grobgittermaschenweite im uniformen Fall, fällt die numerische Diffusion im adaptiven Verfahren schwächer aus als im uniformen. Das heißt, um die maximale Verschiebung der Nullkontur bei der adaptiven Methode zu erreichen, sind mehr Reinitialisierungen nötig als bei der uniformen Methode.

Level-Set-Funktionen mit singulären Krümmungen

Die Überlegungen in Abschnitt 4.1, dass die numerische Diffusion von der Krümmung der freien Oberfläche – also von der Nullkontur der Level-Set-Funktion – abhängt, werden in den vorhergehenden Tests bestätigt. Daraus folgt, dass das adaptive Zweigitterverfahren besonders dann Vorteile gegenüber dem herkömmlichen, uniformen Verfahren bringt, wenn die freie Oberfläche starke Krümmungen beispielsweise bei Tröpfchenbildung und Topologieveränderungen aufweist. Auch starke und im Allgemeinen nichtlaminare oder gar turbulente Strömungen verursachen in der freien Oberfläche feinste Strukturen, deren Dimensionen nur noch durch die Auflösung des Gitters nach unten beschränkt sind. Das adaptive Zweigitterverfahren bietet im Gegensatz zu dem herkömmlichen uniformen Verfahren signifikante Vorteile bei freien Oberflächen mit starken Krümmungen und kann darüber hinaus geometrische Strukturen im Bereich von singulären Krümmungen besser erhalten. Dies zeigt sich in dem folgenden Test, in dem ein Würfel mit zugehöriger Abstandsfunktion als freie Oberfläche gewählt wird. Entlang jeder Ecke und Kante eines Würfels ist die Krümmung der Level-Set-Funktion singulär; diese Singularitäten setzen sich im Inneren des Würfels auf den Diagonalen und diagonal verlaufenden Ebenen fort. An diesen singulären Stellen ist die Level-Set-Funktion nicht mehr stetig differenzierbar, sondern nur noch Lipschitz-stetig; das kann dazu führen, dass die ENO/WENO-Schemata extreme Gradienten berechnen und somit die CFL-Bedingung der Hamilton-Jacobi-Iteration verletzen. Letztendlich resultieren daraus weitere Störun-

gen der geometrischen Strukturen der freien Oberfläche (siehe auch Bemerkung auf Seite 68). Erwartungsgemäß neigt die Level-Set-Reinitialisierung zum Glätten von Ecken und Kanten. Interessant ist daher die Frage nach der Fähigkeit des adaptiven Zweigitterverfahrens, diese Strukturen zu erhalten. Zu diesem Zweck dient der folgende Test: Eine Nullkontur mit der Geometrie eines Würfels und zugehöriger Abstandsfunktion wird der wiederholten Reinitialisierung unterworfen. Die Abbildung 5.9 zeigt die Nullkontur nach einer unterschiedlichen Anzahl an Iterationen sowohl für das herkömmliche als auch für das adaptive Zweigitterverfahren. Wiederum wird deutlich, dass die numerische Diffusion in Bereichen starker Krümmung eine Verschiebung der Nullkontur in Richtung der Krümmung bis um maximal eine Gittermaschenweite verursacht, wobei Ecken und Kanten zu glatteren Rundungen aufgeweicht werden. Das Maß für den Erhalt der Geometrie

Reinitialisierungen	uniform		adaptiv		
	H	$H/4$	$H, H/4$	$H, H/8$	$H, H/16$
0	0	0	0	0	0
1	1.52 e-1	5.21 e-2	1.52 e-1	9.29 e-2	8.99 e-2
2	2.44 e-1	6.05 e-2	1.50 e-1	9.10 e-2	9.27 e-2
10	3.81 e-1	7.15 e-2	1.60 e-1	8.97 e-2	9.33 e-2
20	5.49 e-1	7.18 e-2	1.60 e-1	8.73 e-2	9.25 e-2
100	8.06 e-1	7.26 e-2	1.70 e-1	8.91 e-2	9.16 e-2
200	8.88 e-1	7.29 e-2	1.73 e-1	8.94 e-2	9.08 e-2

Tabelle 5.2: Der relative Fehler der Level-Set-Funktion nach zahlreichen Reinitialisierungen in der l_2 -Norm: Eine Level-Set-Funktion ϕ^0 mit der Nullkontur in Form eines Würfels und der zugehörigen Abstandsfunktion wird iterativ reinitialisiert. Die Tabelle zeigt den relativen Fehler $\|\phi^n - \phi^0\|_{l_2} / \|\phi^0\|_{l_2}$ auf dem relevanten Gebiet in der l_2 -Norm für die uniformen Verfahren mit den Grobgittermaschenweiten H und $H/4$ und für die Zweigitterverfahren mit der Grobgittermaschenweite H und unterschiedlichen Feingittermaschenweiten $h \in \{H/4, H/8, H/16\}$.

sei der relative Fehler $\|\phi^n - \phi^0\|_{l_2} / \|\phi^0\|_{l_2}$. Dabei ist ϕ^0 die exakte Abstandsfunktion zum Würfel und ϕ^n die Level-Set-Funktion nach n -facher Reinitialisierung. Die Tabelle 5.2 zeigt die relativen Fehler in der auf dem relevanten Gebiet ausgewerteten l_2 -Norm für die herkömmlichen, uniformen Verfahren mit den Grobgittermaschenweiten H und $H/4$ und für die adaptiven Zweigitterverfahren mit den Irregularitäten $\iota \in \{2, 3, 4\}$ beziehungsweise den Feingittermaschenweiten $H/4, H/8$ und $H/16$. Aus der Tabelle 5.2 und den Graphiken 5.10a und 5.10b resultiert, dass das adaptive Verfahren die Geometrie des Würfels signifikant besser erhält als das uniforme Verfahren; sie weisen jedoch auch weitere interessante Eigenschaften auf:

- a) Für $\iota \geq 3$ hat die Irregularität ι des adaptiven Gitters keinen wesentlichen Einfluss mehr auf die Größenordnung der numerische Diffusion.

- b) Vergleicht man den Fehler des adaptiven Zweigitterverfahrens (mit $h = H/4$) mit dem uniformen Verfahren mit der Grobgittermaschenweite $H/4$, zeigt sich, dass das adaptive Verfahren mit der Irregularität $\iota = 2$ – anders als bei dem Test mit den Sphären (siehe auch Abbildung 5.6) – den Fehler nicht bis auf die Größenordnung des uniformen Verfahrens reduzieren kann.

Eine Erklärung für diese Beobachtungen liegt in der Tatsache, dass die Anfangsdaten auf dem groben Gitter vorliegen und erst durch die Transferoperatoren auf das feine Gitter (mit $h = H/2^2$) übertragen werden. Das hat zur Folge, dass die Singularitäten in der Krümmung der Level-Set-Funktion – die dort weniger glatt, also lediglich Lipschitzstetig und nicht differenzierbar ist – beim Transfer durch die Interpolation vierter Ordnung zu einem trikubischen Polynom auf dem feinen Gitter geglättet werden, während die Anfangsdaten beim uniformen Verfahren keiner Glättung unterliegen und daher die originalen Singularitäten enthalten.

- zu a) Die Freiheitsgrade des feinen Gitters sind die Auswertungen des Interpolationspolynoms zwischen den Freiheitsgraden des groben Gitters. Eine höhere Auflösung des feinen Gitters durch höhere Irregularität des adaptiven Verfahrens bringt daher keinen zusätzlichen Informationsgehalt auf dem feinen Gitter. Somit kann kein besseres Fehlerverhalten erzielt werden.
- zu b) Die Transferoperatoren verursachen durch diese Glättung eine Störung zu Beginn der Reinitialisierung, wodurch der Fehler nicht mehr das niedrige Niveau eines hochaufgelösten, uniformen Verfahrens erreichen kann.

Den Abschluss dieses Unterabschnitts bildet folgender qualitativer Test: Der Anfangswert einer Level-Set-Funktion ϕ^0 habe eine Nullkontur, die der Oberfläche zweier sich schneidender Sphären entspricht. Ferner erfülle sie die Abstandseigenschaft. Interessant ist der Schnittkreis, an dem die beiden Sphären miteinander verschmelzen. Die singuläre Krümmung weicht nach außen hin auf, und der Schnittkreis schmiegt sich den nächstgelegenen Gitterpunkten an. Die Abbildungen 5.11 zeigen, wie gut das adaptive Zweigitterverfahren die Nullkontur selbst in Bereichen extremster Krümmungen fixiert.

5.2.2 Erzeugung einer Abstandsfunktion und Erhaltung der Nullkontur

Bisher ist das adaptive Zweigitterverfahren in Bezug auf die Transferoperatoren und auf die Erhaltung der Nullkontur während der Reinitialisierung hin untersucht worden. Dazu standen stets Abstandsfunktionen als Startwert zur Verfügung. In diesem Abschnitt soll festgestellt werden, wie gut das adaptive Zweigitterverfahren

- eine gestörte Abstandsfunktion zu gegebener Nullkontur wiederherstellen kann und
- die Nullkontur beim Wiederherstellungsprozess erhält.

Zu diesem Zweck wird im Folgenden der bekannte Testfall von Peter Smereka und Giovanni Russo [44] herangezogen werden.

Modellproblem von Smereka und Russo

In diesem Abschnitt steht das Modellproblem von Peter Smereka und Giovanni Russo aus [44] im Mittelpunkt; es gibt eine Level-Set-Funktion mit der Nullkontur einer Ellipse und einer besonders stark gestörten Abstandsfunktion als Anfangswert analytisch vor. In drei Dimensionen soll der zweidimensionale Fall durch Extrusion auf der z -Achse in einen Quasi-zweidimensionalen-Fall überführt werden. Die Abbildung 5.12 illustriert das Modellproblem von Smereka und Russo. Es gilt nun, die gestörte Abstandsfunktion unter Erhaltung der Nullkontur in eine vorzeichenbehaftete Abstandsfunktion zu überführen (siehe auch Abbildung 5.12c). Beim Betrachten der Abbildungen 5.13 und 5.14 fällt auf, dass die steilen Gradienten der Level-Set-Funktion am linken Rand der Ellipse im uniformen Verfahren dazu führen, dass die Nullkontur durch die Reinitialisierung stark verformt wird, während das adaptive Verfahren die Nullkontur gut erhält. (Eine quantitative Aussage dazu folgt im nächsten Abschnitt.) Daraus resultieren zwei Schlussfolgerungen:

- Die numerische Diffusion hängt nicht nur von der Krümmung der Nullkontur, sondern auch von der Störung der Abstandseigenschaft ab.
- Der Einsatz des adaptiven Zweigitterverfahrens wird nicht nur durch das Auftreten starker Krümmungen in der freien Oberfläche von Strömungssimulationen gerechtfertigt, sondern auch durch die Vorkommnisse starker Störungen der Abstandseigenschaft der Level-Set-Funktion. Dies ist unter anderem besonders bei stark beschleunigten Fluiden und turbulenten Strömungen der Fall.

Numerische Diffusion in Abhängigkeit der Störung der Abstandseigenschaft

Nun soll untersucht werden, inwieweit die durch die anfängliche Störung der Abstandseigenschaft verursachte numerische Diffusion bei der Reinitialisierung durch den Einsatz des adaptiven Zweigitterverfahrens kompensiert werden kann. Wie aus den vorhergehenden Unterabschnitten hervorgeht, verschiebt sich die Nullkontur – bei gleichbleibender Krümmung – in den Bereichen stärker, in denen die Level-Set-Funktion große Gradienten besitzt. Smerekas und Russos Modellproblem wird nun derart modifiziert, dass an Stelle einer Ellipse als Nullkontur diesmal eine Sphäre verwendet wird; das hat zwei Vorteile: Zum einen ist die Krümmung auf der gesamten Oberfläche konstant, so dass es unerheblich sein wird, wo die Störungen ansetzen, zum anderen kann einfacher eine exakte Abstandsfunktion zur Fehlermessung berechnet werden. Das Problem soll nun voll in drei Dimensionen betrachtet werden. Konkret wird die Abstandsfunktion einer Sphäre

mit Radius 1 betrachtet, was 10 Gittermaschenweiten entspricht. Gemäß den Untersuchungen aus Abschnitt 5.2.1 folgt, dass bei einem Krümmungsradius von $r = 10H$ die numerische Diffusion aufgrund der Krümmung moderat ist (vergleiche auch mit Tabelle 5.1), so dass der Einfluss der Störungen besser beobachtet werden kann. Eine künstliche Störung dieser Abstandsfunktion kann erreicht werden, indem sie mit einer Störfunktion f skaliert wird.

$$\begin{aligned} f : \mathbb{R}^3 &\rightarrow \mathbb{R} \\ (x, y, z) &\mapsto |\Delta x|^q + |\Delta y|^q + |\Delta z|^q, \end{aligned} \quad (5.1)$$

wobei

$$(\Delta x, \Delta y, \Delta z) = (x - x_0, y - y_0, z - z_0)$$

und (x_0, y_0, z_0) ein Punkt auf der Nullkontur ist. Der Exponent q wird aus der Menge $\{1.0, 2.0, 3.0, 4.0\}$ gewählt. Nun kann zu jedem Exponent q ein Minimum und ein Maximum für $|\nabla\phi|$ auf der Nullkontur angegeben werden. Das Minimum ist stets 0, die Maxima unter den oben genannten Werten für q sind $\{6, 12, 24, 48\}$. Die Abbildung 5.15 zeigt ein zweidimensionales Schnittbild dieses Testfalls nach einer durchgeführten Reinitialisierung für zwei verschiedene Exponenten q ; (x_0, y_0, z_0) liegt dabei oben rechts auf der Nullkontur, das Maximum der Störung (auf der Nullkontur) befindet sich demnach unten links auf der Nullkontur. Die Tabelle 5.3 zeigt die relativen Massenverluste des uni-

$\max_{x \in N} \nabla\phi(x) $ $N := \{z \phi(z) = 0\}$	uniform H	adaptiv $H, H/4$	uniform $H/4$
24.00	-1.41 e-2	5.09 e-3	-5.62 e-3
33.94	-2.09 e-2	-9.00 e-4	-7.55 e-3
48.00	-2.69 e-2	-5.80 e-3	-9.36 e-3

Tabelle 5.3: Der relative Massenverlust in Abhängigkeit vom Maximum der Störung auf der Nullkontur. Es werden das uniforme Verfahren mit der Maschenweite H und das adaptive Zweigitterverfahren mit der Grobgittermaschenweite H und der Feingittermaschenweite $h = H/4$ im Vergleich zum hochaufgelösten uniformen Verfahren mit der Maschenweite $H/4$ nach jeweils einer Reinitialisierung betrachtet.

formen Verfahrens mit Maschenweite H und des adaptiven Zweigitterverfahren mit der Grobgittermaschenweite H und der Feingittermaschenweite $h = H/4$ im Vergleich zum hochaufgelösten uniformen Verfahren mit der Maschenweite $H/4$ nach jeweils einer Reinitialisierung für unterschiedlich große Störungen. Der Massenverlust durch numerische Diffusion hängt monoton vom Grad der Störung ab. Erwartungsgemäß erreicht das adaptive Zweigitterverfahren Massenverluste, die in der Größenordnung derer des hoch aufgelösten Verfahrens liegen. Den relativen l_2 -Fehler der reinitialisierten Level-Set-Funktion zur exakten Abstandsfunktion in der l_2 -Norm zeigt uns Tabelle 5.4. Die Abbildung 5.16

$\max_{x \in N} \nabla \phi(x) $	uniform	adaptiv
$N := \{z \phi(z) = 0\}$	H	$H, H/4$
6.00	2.72 e-1	9.06 e-3
12.00	3.48 e-1	4.32 e-3
24.00	5.42 e-1	6.30 e-3
48.00	9.12 e-1	5.40 e-2

Tabelle 5.4: Der relative Fehler $\|\phi^1 - d\|_{l_2} / \|d\|_{l_2}$ von der Level-Set-Funktion nach der ersten Reinitialisierung zur exakten Abstandsfunktion d für unterschiedlich große Störungen: Es werden jeweils die Werte für das uniforme Verfahren mit der Maschenweite H und das adaptive Zweigitterverfahren mit der Grobgittermaschenweite H und der Feingittermaschenweite $h = H/4$ im Vergleich aufgeführt.

zeigt die relativen Fehler der Level-Set-Funktion nach der ersten Reinitialisierung zur exakten Abstandsfunktion in der l_2 -Norm. Zu sehen ist, dass der Fehler asymptotisch linear von der maximalen Störung $\max_{x \in N} |\nabla \phi(x)|$ auf der Nullkontur N abhängt und substantiell durch das adaptive Zweigitterverfahren verbessert werden kann.

5.3 Dynamische Systeme mit Konvektion

Nun soll das adaptive Zweigitterverfahren erstmals in einem Strömungsfeld zur Anwendung kommen. Zu diesem Zweck wird die Abstandsfunktion einer Sphäre in einem Rotationsfeld (u, v, w) mit

$$\begin{aligned}
 u(x, y, z) &= -y \frac{2\pi}{100} \\
 v(x, u, z) &= x \frac{2\pi}{100} \\
 z(x, y, z) &= 0
 \end{aligned}
 \tag{5.2}$$

im Kreis transportiert. Da in jedem Zeitschritt des Strömungslösers ein Transport der Level-Set-Funktion gemäß des Geschwindigkeitsfeldes stattfindet, der im Allgemeinen eine Reinitialisierung erfordert, muss auch in jedem Zeitschritt mit einer maximalen Verschiebung der Nullkontur um eine Gittermaschenweite gerechnet werden. Die Abbildung 5.18 zeigt die freie Oberfläche nach mehreren Zeitschritten. Es wird deutlich, dass durch den wiederholten Transport und Transfer der Level-Set-Funktion in jedem Zeitschritt des Strömungslösers eine Störung der Nullkontur verursacht wird. Dies macht sich besonders auch im adaptiven Fall bemerkbar, da hier nach wie vor der Transport auf dem groben Gitter mit den Geschwindigkeitswerten, die ebenfalls auf dem groben Gitter vorliegen, durchgeführt wird. Das führt dazu, dass die Massenverluste bei dem adaptiven Verfahren größer sind als bei einem vergleichbaren uniformen Verfahren mit

Zeitschritte	uniform	adaptiv		uniform
	H	$H, H/4$	$H, H/8$	$H/4$
25	-2.29 e-1	-8.33 e-2	-7.96 e-2	-2.28 e-4
50	-4.41 e-1	-9.14 e-2	-7.63 e-2	-3.64 e-4
75	-7.39 e-1	-7.44 e-2	-4.68 e-2	-4.28 e-4
100	-1.00 e-0	-4.89 e-2	-1.02 e-2	-5.14 e-4

Tabelle 5.5: Der relative Massenverlust zu verschiedenen Zeitschritten für das uniforme Verfahren mit der Grobgittermaschenweite H , die adaptiven Zweigitterverfahren mit den Feingittermaschenweiten $h = H/4$ beziehungsweise $h = H/8$ und das uniforme Verfahren mit der Maschenweite $H/4$.

Aufwand	uniform	adaptiv		uniform
	H	$H, H/4$	$H, H/8$	$H/4$
Speicher	2.7 e+6	15 e+6	21 e+6	58 e+6
Rechenzeit	10	106	795	1235

Tabelle 5.6: Der Aufwand für Speicher (in bytes) und Rechenzeit (in Sekunden) von 100 Zeitschritten für das uniforme Verfahren mit der Grobgittermaschenweite H , die adaptiven Zweigitterverfahren mit den Feingittermaschenweiten $h = H/4$ beziehungsweise $h = H/8$ und das uniforme Verfahren mit mit der Maschenweite $H/4$.

globaler feiner Maschenweite. Dennoch erzielt das adaptive Zweigitterverfahren erhebliche Verbesserungen des Massenverlustes (siehe Tabelle 5.5) bei vergleichsweise geringem Mehraufwand in Speicher und Rechenzeit (siehe auch Tabelle 5.6).

5.4 Test in NaSt3DGPF

Es folgen zwei Testrechnungen mit realen Zweiphasenströmungen. Darin zeigt sich das Zusammenspiel aller in NaSt3DGPF beteiligten Komponenten wie Strömungsberechnung, Transport und Transfer der Level-Set-Funktion, die Reinitialisierung dieser auf dem feinen Gitter und der anschließende Transfer zurück auf das grobe Gitter.

Im ersten Test kollabiert eine Wassersäule umgeben von Luft unter ihrer eigenen Schwerkraft und zerfließt auf dem Boden. Die Wassersäule hat einen quadratischen Querschnitt von $5\text{cm} \times 5\text{cm}$ und ist 8cm hoch. Die Auflösung des groben Gitters beträgt $H = 1\text{cm}$. Die Abbildung 5.19 zeigt die Simulation einer kollabierenden Wassersäule zu verschiedenen Zeitpunkten t sowohl für das uniforme wie auch das adaptive Zweigitterverfahren der Level-Set-Reinitialisierung. Deutlich erkennbar ist die bedeutend bessere Massenerhaltung beim adaptiven Verfahren. Tabelle 5.7 belegt diese Beobachtung. Dieser Testfall erläutert exemplarisch das Problem der numerischen Diffusion bei dünnen Fluid-

Zeit t	uniform	adaptiv	uniform
	H	$H, H/4$	$H/4$
0.01s	2.04 e-3	8.58 e-4	-1.14 e-5
0.08s	-1.69 e-2	-1.40 e-3	-1.26 e-3
0.13s	-5.39 e-2	-1.18 e-3	-2.74 e-3

Tabelle 5.7: Der relative Massenverlust zu den in den Abbildungen vorkommenden Zeitpunkten t . Es werden die Messwerte für das uniforme Verfahren mit der Grobgittermaschenweite $H = 1cm$, das adaptive Zweigitterverfahren mit der Feingittermaschenweite $h = H/4$ und das uniforme Verfahren mit der Maschenweite $H/4$ aufgeführt.

filmen: Die künstliche Verschiebung der freien Oberfläche am Rand des Films verhindert seine flächenartige Ausbreitung beziehungsweise den weiteren Benetzungsvorgang seiner Kontaktfläche. Die Auflistung der Kosten in Tabelle 5.8 beweist die Effizienz und

Aufwand	uniform	adaptiv	uniform
	H	$H, H/4$	$H/4$
Speicher	9 e+6	34 e+6	625 e+6
Rechenzeit	543	1841	166445

Tabelle 5.8: Der Aufwand für Speicher (in bytes) und Rechenzeit (in Sekunden) von 1s Simulationszeit. Es werden die Messwerte für das uniforme Verfahren mit der Grobgittermaschenweite $H = 1cm$, das adaptive Zweigitterverfahren mit der Feingittermaschenweite $h = H/4$ aufgeführt und das uniforme Verfahren mit der Maschenweite $H/4$.

Wirtschaftlichkeit des adaptiven Zweigitterverfahrens.

Im zweiten Test wirkt die Oberflächenspannung auf einen Fluidwürfel. Dessen Oberfläche beginnt daraufhin – wie man Abbildung 5.20 entnehmen kann – zu schwingen. Dieser Würfel besteht aus Wasser, ist von Luft umgeben und besitzt eine Kantenlänge von $5cm$. Die Maschenweite beträgt $H = 0.5cm$. Die Abbildung 5.20 zeigt einige interessante Zeitpunkte dieses Vorgangs. Im Fall der Reinitialisierung auf dem groben, uniformen Gitter verliert der Fluidwürfel deutlich sichtbar an Masse. Ferner kann man sehen, dass die numerische Diffusion beim adaptiven Fall erheblich weniger zu Glättungen der freien Oberfläche führt. Die Tabelle 5.9 belegt die Beobachtung: Nachdem die freie Oberfläche eine Schwingungsperiode durchlaufen hat, hat der Fluidwürfel mehr als 17% seiner Masse verloren, wenn die Level-Set-Reinitialisierung auf dem groben uniformen Gitter mit Maschenweite H durchgeführt wird. Das adaptive Zweigitterverfahren mit der Feingittermaschenweite $h = H/4$ hingegen reduziert den Massenverlust auf weniger als 5%.

Zeit t	uniform H	adaptiv $H, H/4$
1.20s	-1.74 e-1	-4.75 e-2

Tabelle 5.9: Der relative Massenverlust nach $t = 1.2s$. Es werden die Messwerte für das uniforme Verfahren mit der Grobgittermaschenweite $H = 0.5cm$ und das adaptive Zweigitterverfahren mit der Feingittermaschenweite $h = H/4$ aufgeführt.

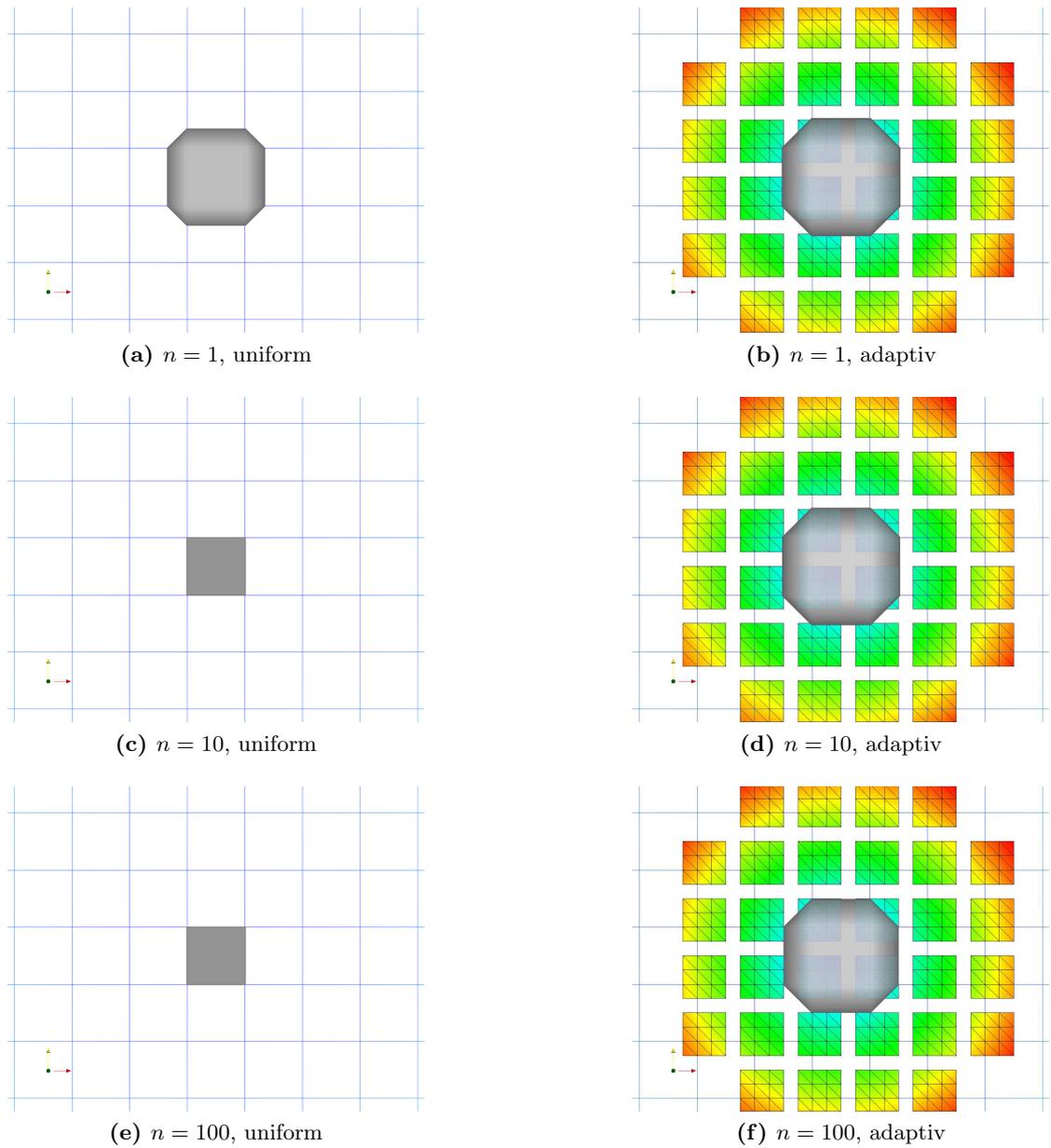


Abbildung 5.8: Die Sphäre vom Radius $r = 1.25H$ mit zugehöriger Abstandsfunktion nach 1, 10 und 100 Reinitialisierungen. Dargestellt ist die Nullkontur auf dem groben Gitter und ein Schnitt durch das grobe Gitter (blaue Linien). Bei (a), (c), (e) wird auf einem uniformen Gitter und bei (b), (d), (f) mit dem adaptiven Zweigitterverfahren gerechnet; im letzteren Fall ist noch ein Schnitt durch die grid patches dargestellt.

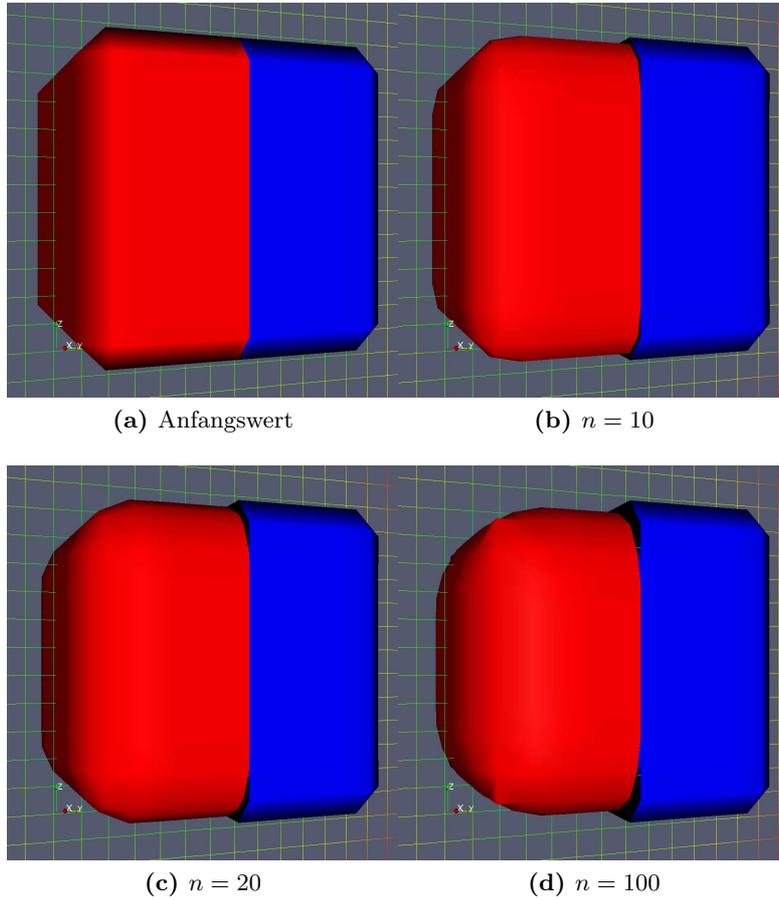
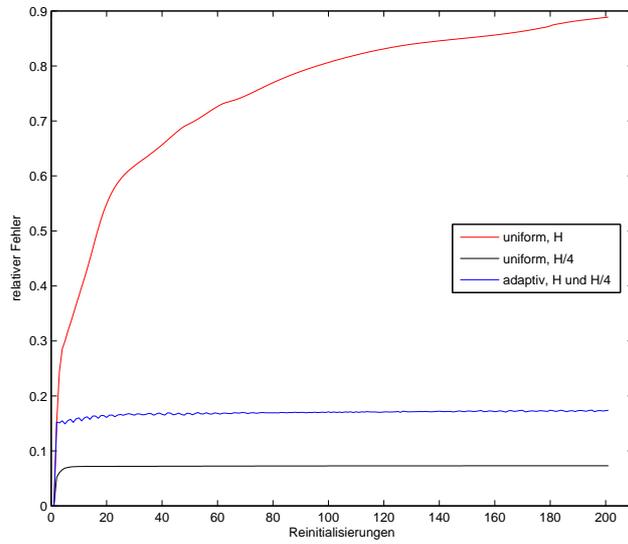
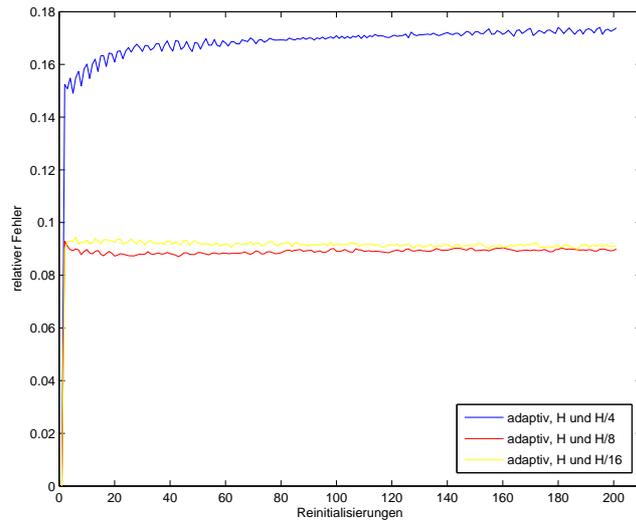


Abbildung 5.9: Wiederholte Reinitialisierung: Die Nullkontur mit der Geometrie eines Würfels und der zugehörigen Abstandsfunktion werden unterschiedlichen Anzahlen n an Reinitialisierungsschritten unterworfen. In den Graphiken ist zwecks besseren Vergleichs jeweils die linke Hälfte der Nullkontur des uniformen Verfahrens (rot) und die rechte Hälfte des adaptiven Verfahrens mit der Irregularität $\iota = 2$ mit Feingittermaschenweite $h = H/4$ dargestellt. Ein Schnitt durch das grobe Gitter dient als optischer Maßstab für Ausmaß der numerischen Diffusion.

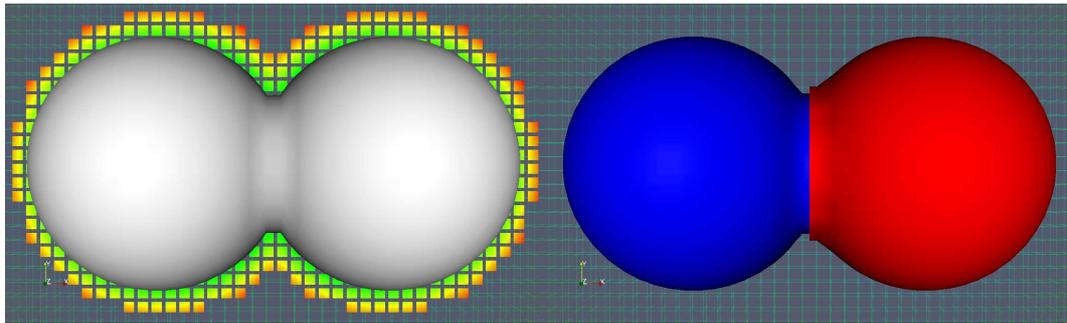


(a)

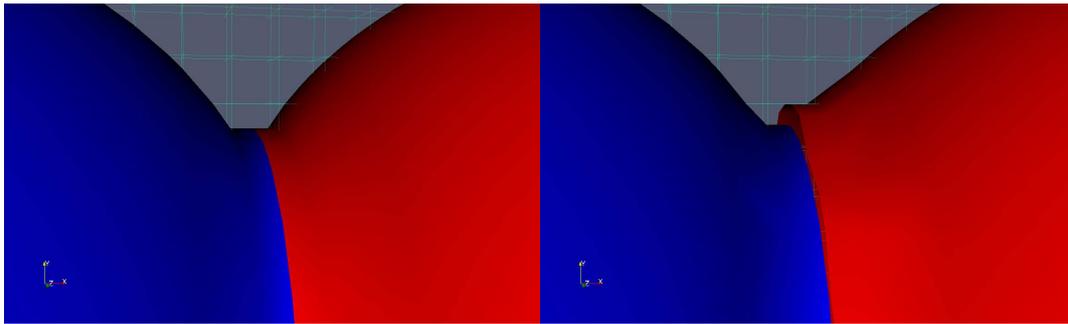


(b)

Abbildung 5.10: Der Fehler der Level-Set-Funktion: Die Graphiken (a) und (b) zeigen den Verlauf des relativen Fehlers $e^n = \|\phi^n - \phi^0\|_{L^2} / \|\phi^0\|_{L^2}$, wobei ϕ^0 eine Abstandsfunktion mit dem Würfel als Nullkontur und ϕ^n diese Funktion nach n Reinitialisierungen ist. Die Graphik (a) vergleicht das herkömmliche, uniforme Verfahren (rot) mit dem adaptiven (blau) und dem uniformen Verfahren mit feinerer Auflösung (schwarz). Die Graphik (b) zeigt die adaptiven Verfahren mit unterschiedlicher Irregularität beziehungsweise Feingittermaschenweite im Vergleich.



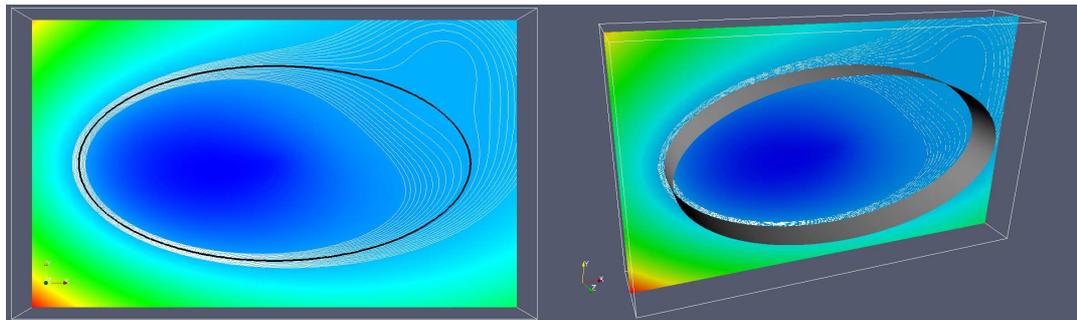
(a) Gesamtansicht, Anfangswert

(b) Gesamtansicht, $n = 40$ 

(c) Anfangswert

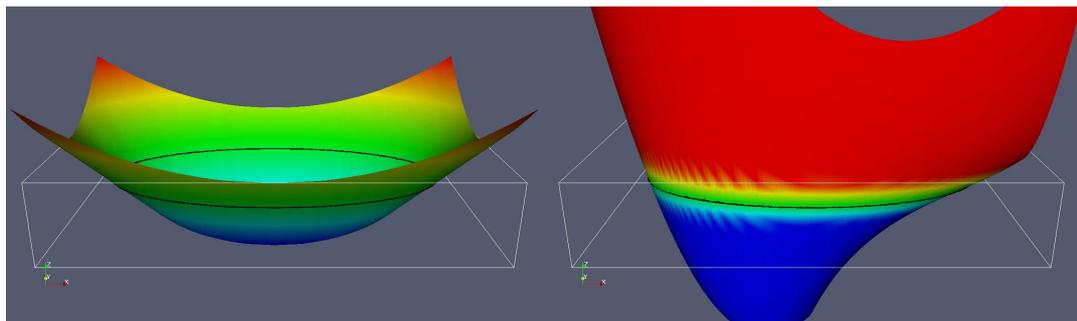
(d) $n = 40$

Abbildung 5.11: Wiederholte Reinitialisierung: Die Nullkontur ist die Oberfläche zweier sich schneidender Sphären mit der zugehörigen Abstandsfunktion. Die Graphik (a) zeigt diese Nullkontur vor dem groben Gitter und einen Schnitt durch die grid patches des adaptiven Verfahrens. Diese Level-Set-Funktion wird unterschiedlichen Anzahlen n an Reinitialisierungsschritten unterworfen. In den Graphiken ist zwecks besseren Vergleichs jeweils die rechte Hälfte der Nullkontur des uniformen Verfahrens (rot) und die linke Hälfte des adaptiven Verfahrens mit der Irregularität $\iota = 2$ mit Feingittermaschenweite $h = H/4$ dargestellt. Ein Schnitt durch das grobe Gitter dient als optischer Maßstab für das Ausmaß der numerischen Diffusion in Bezug auf die Maschenweite.



(a) Konturen der gestörten Abstandsfunktion, (Nullkontur in schwarz)

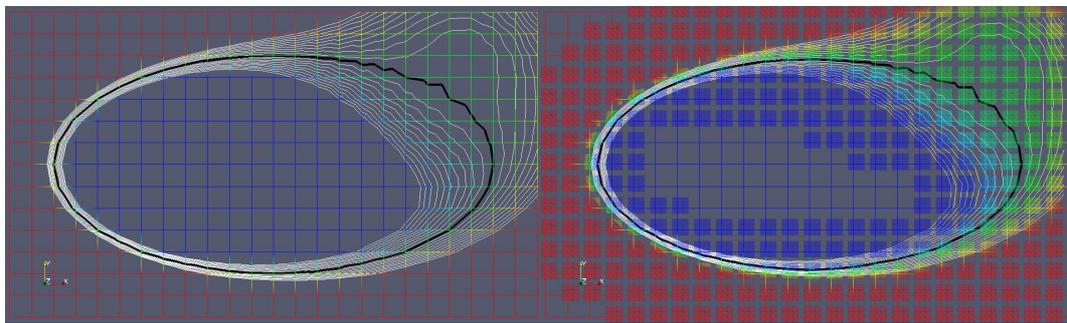
(b) Quasi-2D, konstant in z -Richtung



(c) Die Abstandsfunktion der Ellipse, Schnitt in zwei Dimensionen

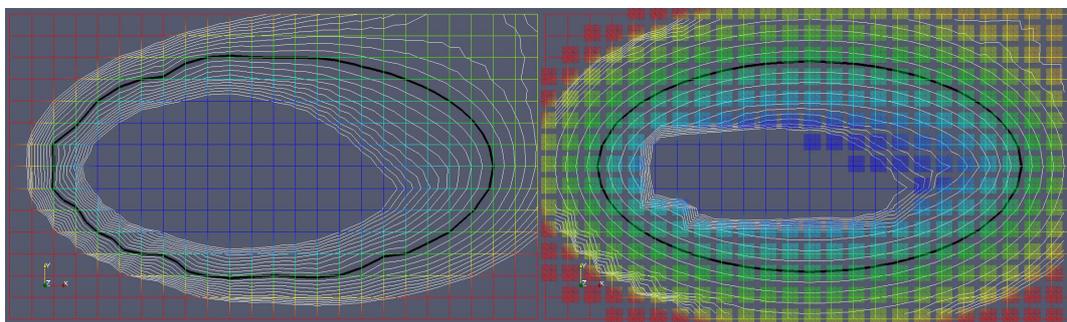
(d) Die gestörte Abstandsfunktion der Ellipse, Schnitt in zwei Dimensionen, mit Faktor $1/4$ skaliert

Abbildung 5.12: Das Problem von Smereka und Russo: Eine stark gestörte Abstandsfunktion hat eine Ellipse als Nullkontur. Die Graphiken zeigen einen Schnitt in zwei Dimensionen von der Nullkontur (schwarz) beziehungsweise die Nullkontur in drei Dimensionen in Abbildung (b). In (a) und (b) ist eine nach der Level-Set-Funktion eingefärbte Schnittebene mit weiteren Konturen mit Werten aus dem Intervall $[-2, +2]$ zu sehen. In (c) und (d) ist diese Schnittebene entsprechend dem Level-Set-Wert deformiert; wegen der extremen Werte der gestörten Abstandsfunktion in (d) ist die Deformation mit dem Faktor $1/4$ skaliert worden.



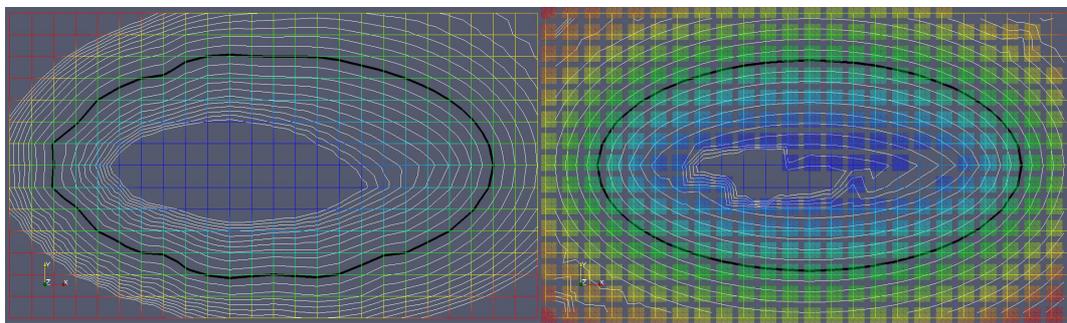
(a) Anfangswert

(b) Anfangswert



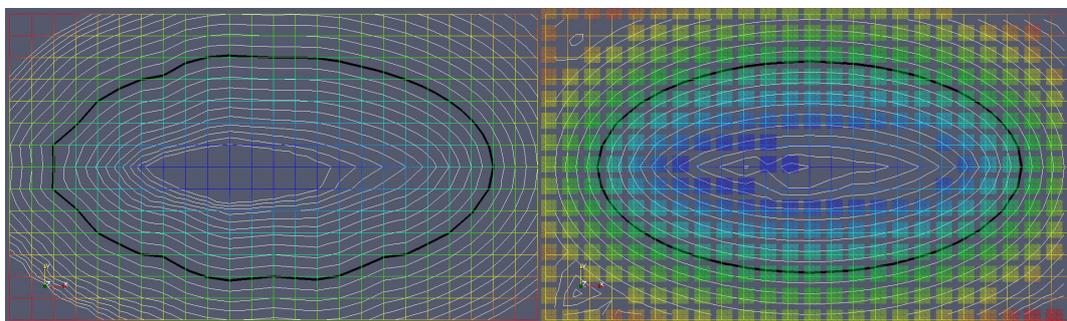
(c) $n = 1$

(d) $n = 1$



(e) $n = 2$

(f) $n = 2$



(g) $n = 3$

(h) $n = 3$

Abbildung 5.13: Fortsetzung auf der nächsten Seite

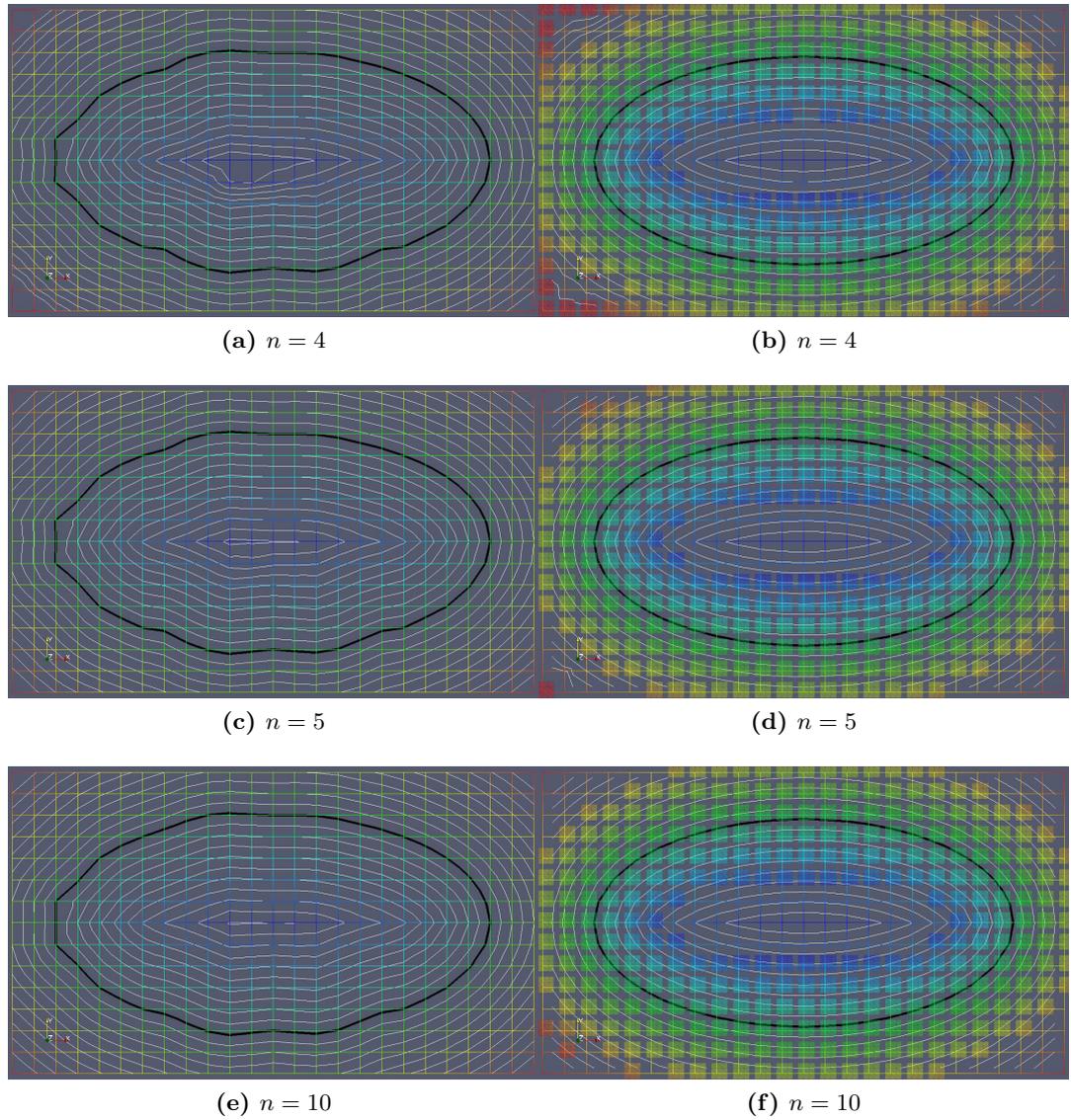


Abbildung 5.14: Wiederherstellung der Abstandseigenschaft: Das herkömmliche, uniforme Verfahren (linke Spalte) mit der Grobgittermaschenweite H im Vergleich zum adaptiven Zweigitterverfahren (rechte Spalte) mit der Grobgittermaschenweite H und der Feingittermaschenweite $h = H/4$. Zu sehen sind der Anfangswert und die Ergebnisse nach unterschiedlich vielen Reinitialisierungen $n \in \{1, 2, 3, 4, 5, 10\}$. Jedes einzelne Bild zeigt in schwarz die Nullkontur und in weiß weitere Konturen aus dem Intervall $[-2, +2]$. Ferner ist ein Schnitt durch das grobe Gitter und im adaptiven Fall ein Schnitt durch die grid patches gezeigt.

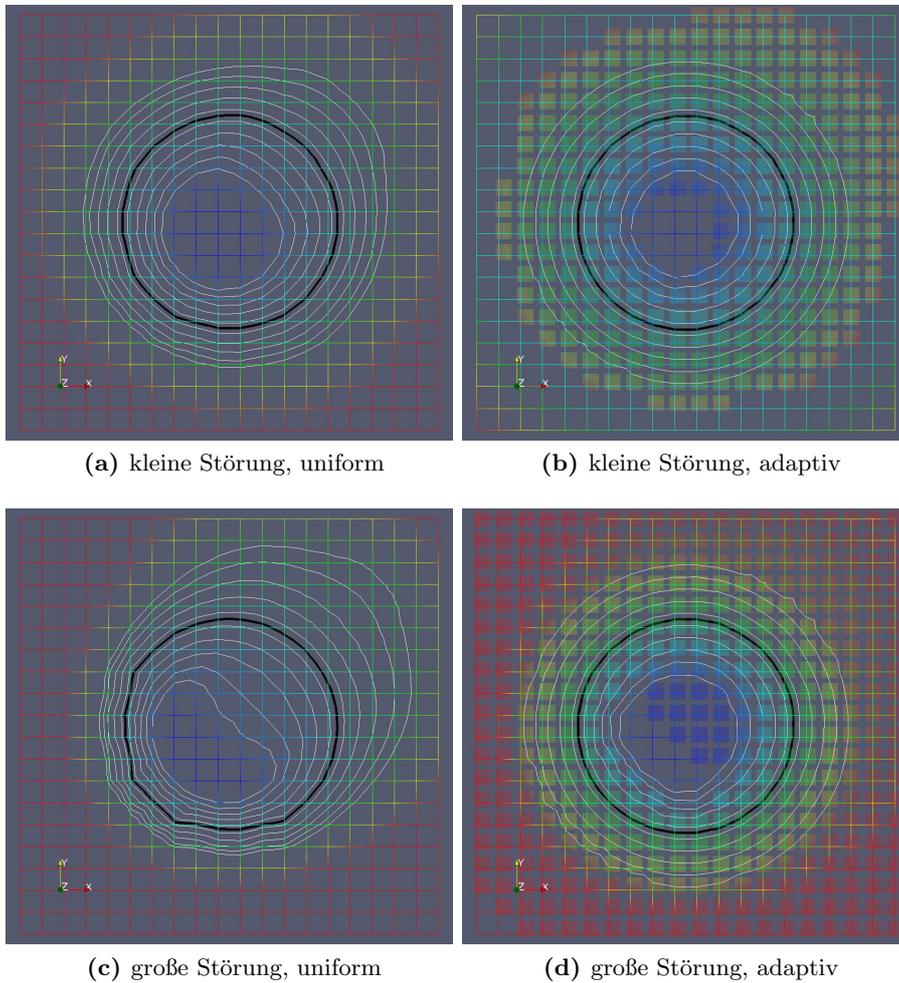


Abbildung 5.15: Numerische Diffusion nach einer Reinitialisierung bei unterschiedlich stark gestörten Abstandsfunktionen im zweidimensionalen Schnitt: Die Abstandsfunktion einer Sphäre ist in Graphik (a) und (b) schwach, in (c) und (d) stark gestört. Der Radius beträgt bei (a) bis (d) stets 5 Grobgittermaschenweiten. Jedes einzelne Bild zeigt in schwarz die Nullkontur und in weiß weitere Konturen aus dem Intervall $[-0.5, +0.5]$. Ferner ist ein Schnitt durch das grobe Gitter und im adaptiven Fall ein Schnitt durch die grid patches gezeigt.

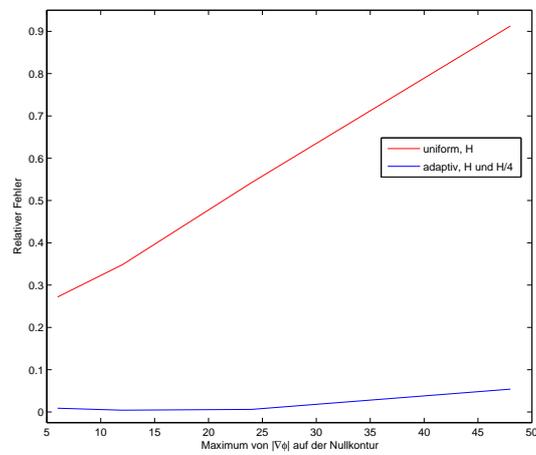


Abbildung 5.16: Der relative Fehler $\|\phi^1 - d\|_{L_2} / \|d\|_{L_2}$ von der Level-Set-Funktion nach der ersten Reinitialisierung zur exakten Abstandsfunktion d in Abhängigkeit vom Grad der Störung: Es werden jeweils die Werte für das uniforme Verfahren mit der Maschenweite H und das adaptive Zweigitterverfahren mit der Grobgittermaschenweite H und der Feingittermaschenweite $h = H/4$ im Vergleich aufgeführt.

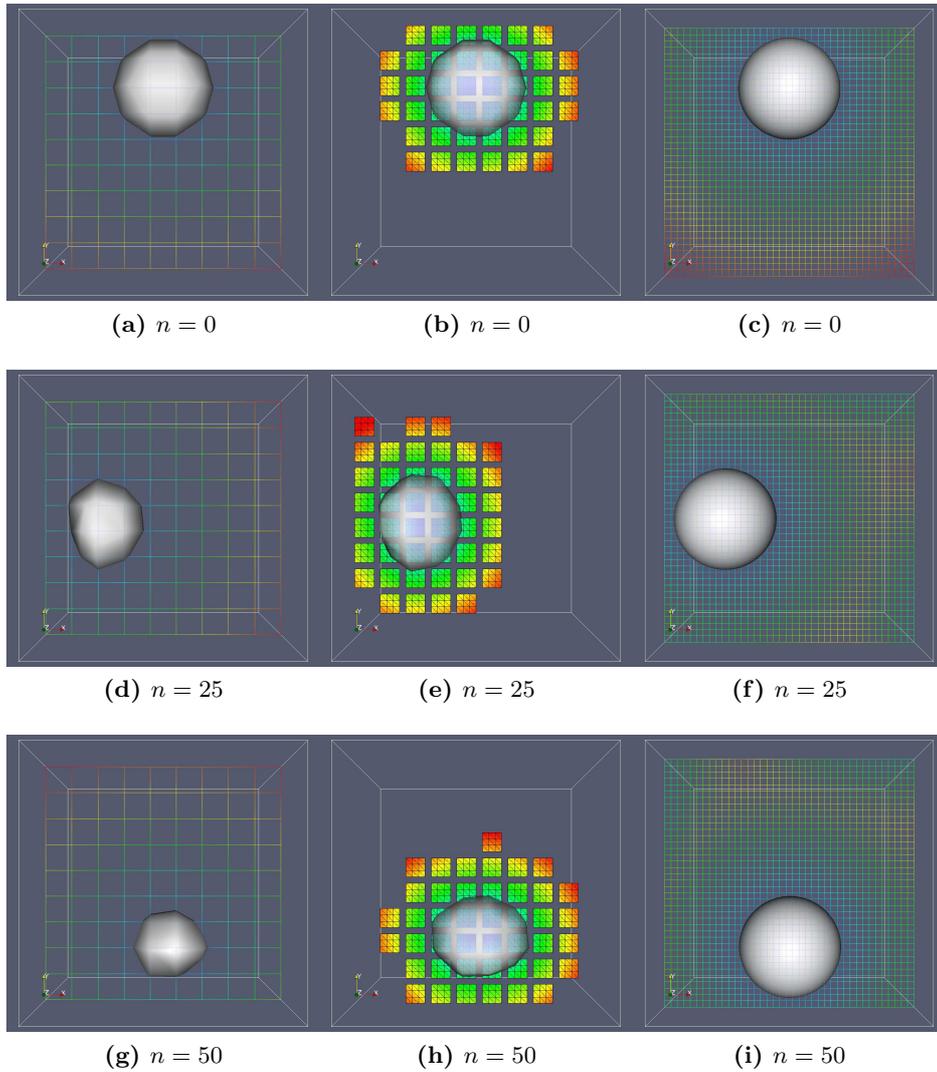


Abbildung 5.17: Fortsetzung auf der nächsten Seite.

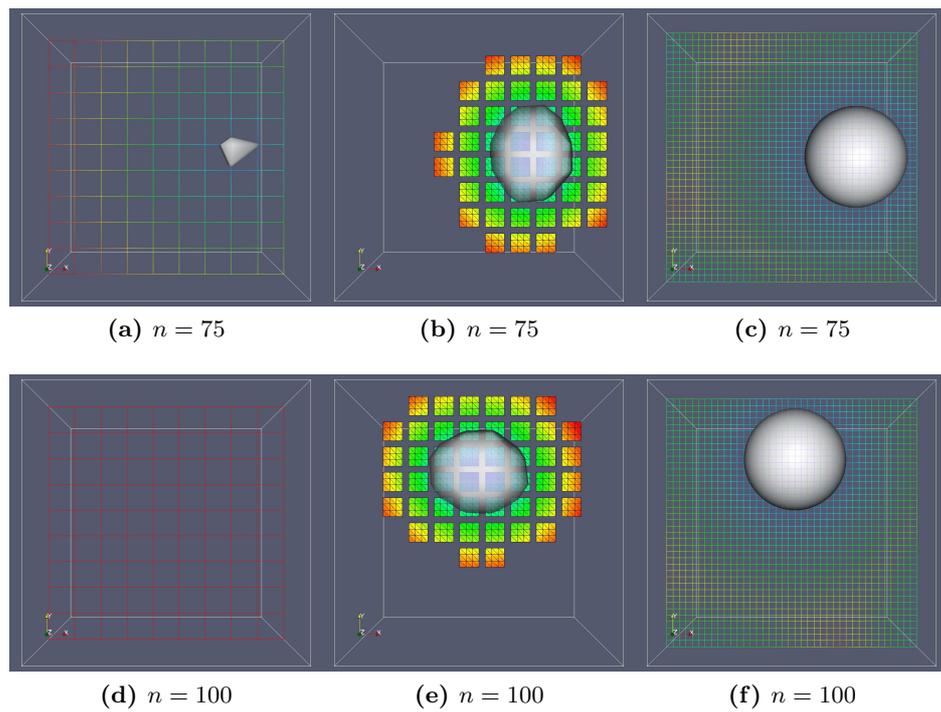


Abbildung 5.18: Eine Sphäre als freie Oberfläche im Rotationsfeld nach n Zeitschritten: In der linken Spalte ist auf einem uniformen Gitter der Maschenweite H gerechnet worden. Die mittlere Spalte zeigt die Verwendung des adaptiven Zweigitterverfahrens mit der Feingittermaschenweite $h = H/4$. Die rechte Spalte zeigt die Berechnungen auf einem uniformen Gitter der Maschenweite $H/4$.

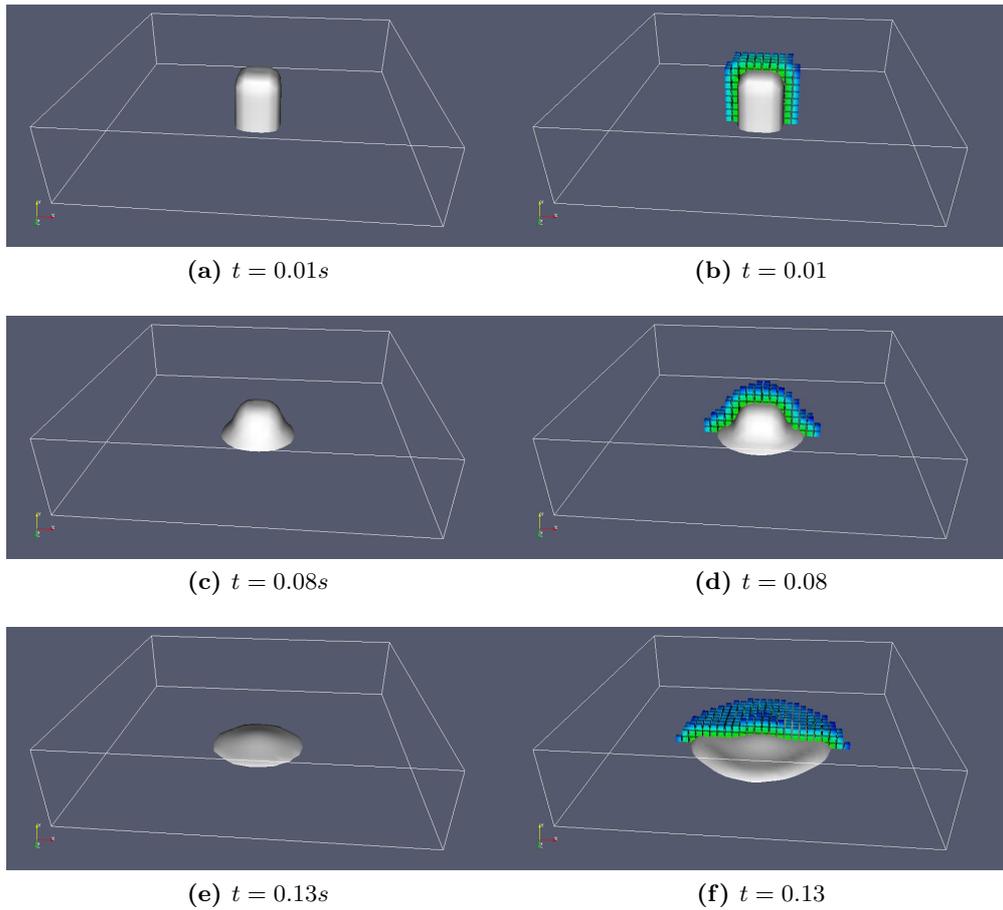


Abbildung 5.19: Eine kollabierende Wassersäule zu verschiedenen Zeitpunkten t auf dem uniformen Gitter mit Maschenweite $H = 1cm$ (linke Spalte) und auf dem adaptiven Gitter (rechte Spalte) mit der Feingittermaschenweite $h = H/4$. Ferner ist im adaptiven Fall ein Teil der grid patches mit abgebildet.

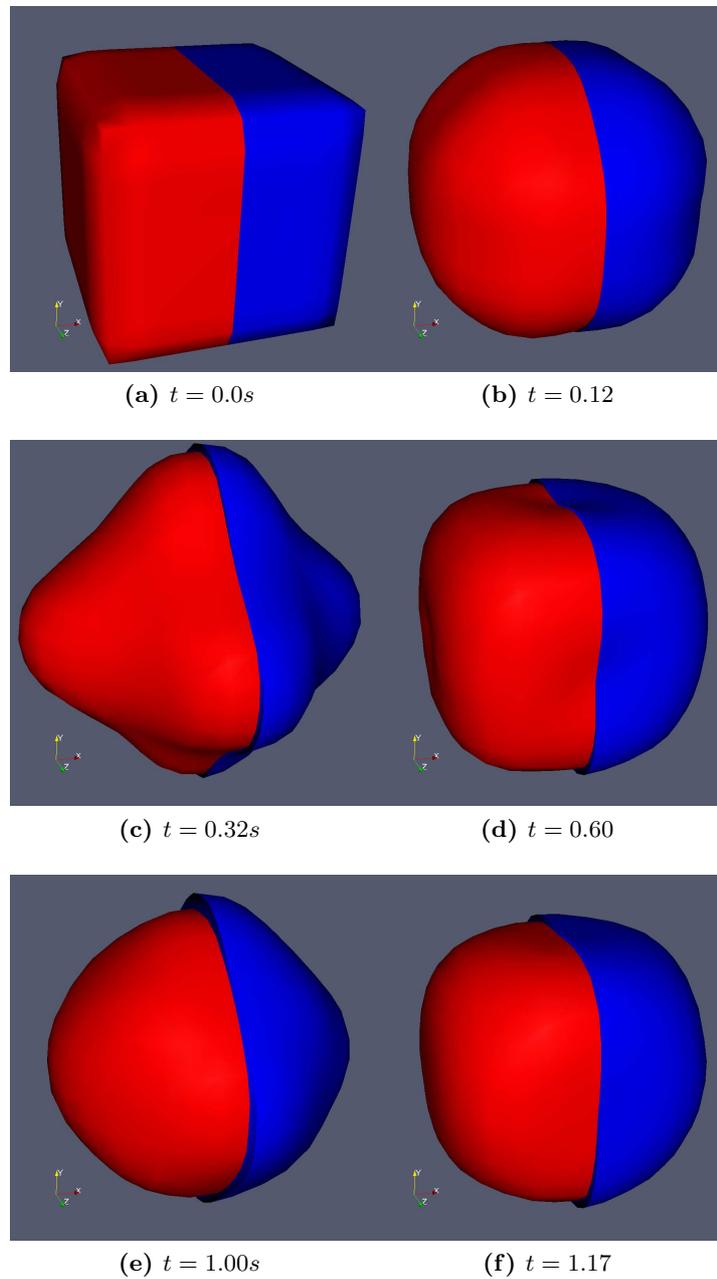


Abbildung 5.20: Der Fluidwürfel zu verschiedenen Zeitpunkten t auf dem uniformen Gitter mit Maschenweite $H = 0.5cm$ (roter Teil) und auf dem adaptiven Gitter (blauer Teil) mit der Feingittermaschenweite $h = H/4$. Es wurde die Schnittbildtechnik aus Abbildung 5.9 angewendet.

Kapitel 6

Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war die Entwicklung einer robusteren Level-Set-Methode für das Programm NaSt3DGPF. Dabei sollte der Aspekt der Adaptivität eine wesentliche Rolle spielen. Da die Diskretisierung in NaSt3DGPF auf strukturierten kartesischen Gittern mit finiten Differenzen beruht, schien die Wahl von kartesischen Gitterkomponenten für das neu zu entwickelnde Verfahren angemessen. Denn dies erleichtert die Wiederverwendbarkeit der bereits in NaSt3DGPF implementierten Differenzenschemata. Mit der Entscheidung für kartesische Gitterkomponenten kamen adaptive Triangulierungen als Gitter nicht weiter in Betracht. Auf der Suche nach geeigneten Methoden, mit hängenden Knoten in kartesischen Gittern umzugehen, fanden sich unter anderem Interpolationsverfahren [39] und Mortar-Methoden [58]. Die genauere Untersuchung der Hamilton-Jacobi-Gleichung, auf der die Level-Set-Reinitialisierung beruht, zeigt aber, dass die Charakteristiken von der Nullniveaumenge der Level-Set-Funktion aus in Richtung der äußeren Normalen verlaufen. Da Adaptivität lediglich in einer Umgebung der Nullkontur benötigt wird und die Erfüllung der Abstandseigenschaft der Level-Set-Funktion außerhalb dieser Umgebung unerheblich ist, konnte auf komplizierte Verfahren zur konsistenten Diskretisierung an Fein-grob-Übergängen der Gitter verzichtet werden. Das heißt, solange die freie Oberfläche komplett in einem fein diskretisierten Gebiet liegt, spielt der Grad der Irregularität an den Fein-grob-Übergängen keine Rolle. Aufgrund dieser Idee war eine Implementierung mit grid patches möglich, die sich zudem auch noch leicht in NaSt3DGPF integrieren ließ, ohne allzu große Änderungen an NaSt3DGPF selbst durchführen zu müssen. Die Testergebnisse zeigen, dass sich die künstliche Verschiebung der Nullkontur bei der Reinitialisierung auf dem feinen Gitter tatsächlich auf maximal eine Feingittermaschenweite h beschränkt. Der Vorteil eines adaptiven Verfahrens gegenüber dem uniformen ist um so größer, je stärker die Effekte der numerischen Diffusion sind. An dieser Stelle belegen die Ergebnisse dieser Arbeit die Beobachtung von T.Y. Hou [15], dass die künstliche Bewegung der Nullkontur mit zunehmender Krümmung schneller wird. Für den Einsatz des adaptiven Verfahrens lässt sich daher die obige Aussage umformulieren: Der Vorteil eines adaptiven Verfahrens gegenüber dem uniformen ist um so größer, je weniger glatt die freie Oberfläche der Zweiphasenströmung ist.

Die Resultate dieser Arbeit zeigen jedoch auch, dass die Reinitialisierung auf dem adaptiven Gitter nicht isoliert betrachtet werden darf, sondern immer im Zusammenhang

mit dem Strömungslöser gesehen werden muss. Denn die Daten, die der adaptiven Reinitialisierung zur Verfügung stehen, stammen stets aus einer Interpolation der Grobgitterwerte. Dabei zeigt sich, dass die Interpolationsordnung des Transferoperators erheblichen Einfluss auf die Qualität der Feingitterdaten hat, welche die Anfangswerte der adaptiven Hamilton-Jacobi-Reinitialisierung sind. So sieht man an einem Beispiel mit singulären Krümmungen der freien Oberfläche, dass die Interpolation Störungen auf dem feinen Gitter verursacht, die schon vor der Reinitialisierung numerische Diffusion bewirkt. In dieser Tatsache liegt der Nachteil des adaptiven Verfahrens gegenüber dem uniformen Verfahren mit globalem feinem Gitter. Letzteres operiert ausschließlich auf feineren Gittern und leidet nicht unter Fehlern der Transferoperatoren. Kommt bei Strömungsproblemen noch der Transport der Level-Set-Funktion hinzu, muss neben dem Fehler des Transferoperators noch der Transport berücksichtigt werden. Dieser wird stets auf dem groben Gitter durchgeführt und verursacht Störungen der Nullkontur im Rahmen der Grobgittergenauigkeit, während bei einem uniformen Verfahren mit feiner Maschenweite neben der Reinitialisierung auch der Transport auf dem feinen Gitter durchgeführt wird.

Trotz dieser Nachteile konnte in den Resultaten dieser Arbeit gezeigt werden, dass eine erheblich verbesserte Massenerhaltung durch den Einsatz einer adaptiven Level-Set-Reinitialisierung erzielt werden kann. Die erhöhten Kosten für Speicher und Rechenzeit lassen sich hingegen nur schwer quantifizieren, da sie stark vom jeweiligen Problem abhängen. Weil sich die Adaptivität lokal auf eine Umgebung der freien Oberfläche beschränkt, werden grid patches lediglich auf einer zweidimensionalen Untermenge des Gebiets eingesetzt, wohingegen eine uniforme Verfeinerung mit der dritten Potenz der Auflösung in die Kosten eingeht. Die Anzahl der Hamilton-Jacobi-Iterationsschritte, die für eine Reinitialisierung notwendig sind, ist bei adaptiven Gittern aufgrund der CFL-Bedingung wegen der kleinsten vorkommenden Gittermaschenweite jedoch die gleiche wie bei uniform verfeinerten Gittern.

Zukünftige Entwicklungen einer adaptiven Level-Set-Reinitialisierung könnten beispielsweise an den oben genannten Schwachstellen ansetzen:

Der Transfer der Level-Set-Funktion vom groben auf das feine Gitter kann selbst numerische Diffusion erzeugen, so dass das Potential einer Reinitialisierung auf dem feinen Gitter nicht voll ausgeschöpft werden kann. An dieser Stelle könnten Interpolationsoperatoren noch höherer Ordnung oder Interpolationen, die nicht auf Polynomen basieren, eventuell Verbesserungen schaffen. Eine andere Möglichkeit bestünde darin, die durch den Transfer erzeugten Störungen nachträglich zu korrigieren. Hier wäre der Einsatz einer Art Hybride-Partikel-Level-Set-Methode [22] denkbar. Ferner gilt es den Fehler durch den Transport der Level-Set-Funktion zu minimieren. Eine erste Idee ist, den Transport auf dem adaptiven feinen Gitter durchzuführen. Dabei muss das Geschwindigkeitsfeld auf das feine Gitter transferiert werden. Es muss daher gewährleistet werden, dass sich dieser Transfer auf das feine Gitter zusammen mit dem Transport auf dem feinen Gitter weniger störend auf die freie Oberfläche auswirkt als der Transport auf dem groben Gitter und der anschließende Transfer auf das feine Gitter. Ein weiterer Vorteil eines Transports

auf dem feinen Gitter ist die Tatsache, dass auf den Transfer der Level-Set-Funktion vom groben auf das feine Gitter verzichtet werden kann.

Bisher dienen die adaptiven Gitter der Reinitialisierung der Level-Set-Funktion und erzielen somit enorme Verbesserungen der Massenerhaltung. Dabei wird die Level-Set-Funktion nach der Reinitialisierung auf das grobe Gitter zurücktransferiert und dem Strömungslöser wieder zugeführt. Eine Möglichkeit, die Daten auf dem feinen Gitter weiterhin nutzbar zu machen, besteht darin, das Deltafunktional und die Heavyside'sche Gewichtsfunktion (siehe Croce [18]), die der Aktivierung der Oberflächenspannungskräfte und der Regularisierung der Fluidparameter in der Umgebung der freien Oberfläche dient, ebenfalls auf dem feinen Gitter auszuwerten, und somit noch akkuratere und stabilere Zweiphasenströmungssimulationen mit Oberflächenspannungskräften berechnen zu können.

Literaturverzeichnis

- [1] ADELSBERGER, J.: *Modellierung und Simulation von globalen Atmosphärenströmungen mit Finite-Volumen-Methoden auf adaptiven Polyedergittern*. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2008.
- [2] AMBROSE, DAVID M.: *Regularization of the Kelvin-Helmholtz instability by surface tension*. PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A-MATHEMATICAL PHYSICAL AND ENGINEERING SCIENCES, 365(1858):2253–2266, SEP 15 2007.
- [3] BELGACEM, F. BEN: *The mortar finite element method with lagrange multipliers*. Numer. Math., 84:172–197, 1999.
- [4] BELGACEM, F. BEN und Y. MADAY: *The mortar element method for three dimensional finite elements*. M^2AN , 31:289–302, 1997.
- [5] BERGER, MARSHA J.: *On Conservation at Grid Interfaces*. SIAM Journal on Numerical Analysis, 24(5):967–984, 1987.
- [6] BERNDT, MARKUS, KONSTANTIN LIPNIKOV, MIKHAIL SHASHKOV, MARY F. WHEELER und IVAN YOTOV: *A mortar mimetic finite difference method on non-matching grids*. Numer. Math., 102(2):203–230, 2005.
- [7] BRAESS, D. und W. DAHMEN: *Stability estimates of the mortar finite element method for 3-dimensional problems*. East-West J. Numer. Math., 6(4):249–264, 1998.
- [8] BRAESS, DIETRICH, WOLFGANG DAHMEN und CHRISTIAN WIENERS: *A Multigrid Algorithm for the Mortar Finite Element Method*. SIAM Journal on Numerical Analysis, 37(1):48–69, 1999.
- [9] BRANDT, A.: *Multi-level adaptive solutions to boundary-value problems*. Math. Comput., 31:333–390, 1977.
- [10] BRANDT, A.: *Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software*. Academic Press, New York, 1977.
- [11] BRANDT, A.: *Guide to multigrid development*. Multigrid Methods. Springer-Verlag, Berlin, 1982.

- [12] C. BERNARDI, Y. MADAI und A.T. PATERA: *Domain Decomposition by the mortar element Method*. Asymptotic and numerical methods for partial differential equations with critical parameters. Reidel, Dordrecht, 1993.
- [13] C. BERNARDI, Y. MADAI und A.T. PATERA: *A new nonconforming approach to domain decomposition: the mortar element method*. Nonlinear partial differential equations and their applications. Paris, 1994.
- [14] CHAN, TONY F. und TAREK P. MATHEW: *Domain decomposition algorithms*. Acta Numerica 1994, Seiten 61–143, 1994.
- [15] CHANG, Y. C., T. Y. HOU, B. MERRIMAN und S. OSHER: *A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows*. Journal of Computational Physics, 124(2):449–464, 1996.
- [16] CHEN, S. und D.B. JOHNSON und P.E. RAAD: *The surface marker and micro cell method*. International Journal for numerical methods in fluids, 25(7):749–778, 1997.
- [17] CHESHIRE, GEOFFREY SAMUEL: *Composite grid construction and applications*. Doktorarbeit, Dept. Appl. Math., California Institute of Technology, Pasadena, CA, 1986.
- [18] CROCE, R.: *Ein paralleler, dreidimensionaler Navier-Stokes-Löser für inkompressible Zweiphasenströmungen mit Oberflächenspannung, Hindernissen und dynamischen Kontaktflächen*. Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, Bonn, Germany, 2002.
- [19] E. H. ATTA, J. VADYAK: *A grid overlapping scheme for flowfield computations about multicomponent configurations*. J., 21(9):1271–1277, 1983.
- [20] ELAINE S. ORAN, JAY P. BORIS: *Numerical Simulation of Reactive Flow*. Elsevier Science Publishing Co., Inc., 1987.
- [21] ENGEL, M.: *Numerische Simulation von Strömungen in zeitabhängigen Gebieten und Anwendung auf Fluid-Struktur-Wechselwirkungen*. Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, Bonn, Germany, 2002.
- [22] ENRIGHT, DOUGLAS, RONALD FEDKIW, JOEL FERZIGER und IAN MITCHELL: *A hybrid particle level set method for improved interface capturing*. J. Comput. Phys., 183(1):83–116, 2002.
- [23] FADLUN, E. A., R. VERZICCO, P. ORLANDI und J. MOHD-YUSOF: *Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations*. J. Comput. Phys., 161(1):35–60, 2000.

-
- [24] G. DIMONTE, ET AL.: *A comparative study of the turbulent Rayleigh-Taylor (RT) instability using high-resolution 3D numerical simulations: The Alpha-Group collaboration*. Phys. Fluids, 16(5):1668 – 1692, 2004.
- [25] GIBOU, FREDERIC, RONALD P. FEDKIW, LI-TIEN CHENG und MYUNGJOO KANG: *A second-order-accurate symmetric discretization of the Poisson equation on irregular domains*. J. Comput. Phys., 176(1):205–227, 2002.
- [26] GRIEBEL, M. und P. OSWALD: *On the abstract theory of additive and multiplicative Schwarz algorithms*. Numer. Math., 70(2):163–180, 1995.
- [27] HENSHAW, W. D.: *I. The numerical solution of hyperbolic systems of conservation laws; II. Composite overlapping grid techniques*. Doktorarbeit, Dept. Appl. Math., California Institute of Technology, Pasadena, CA, 1985.
- [28] HENSHAW, W. D. und G. CHESHIRE: *Multigrid on Composite Meshes*. SIAM Journal on Scientific and Statistical Computing, 8(6):914–923, 1987.
- [29] HIRT, C.W. und B.D. NICHOLS: *Volume of fluid (VOF) method for the dynamics of free boundaries*. Journal of Computational Physics, 39(1):201–225, 1979.
- [30] J. A. BENEK, J. L. STEGER, F. C. DOUGHERTY: *A flexible grid embedding technique with application to the Euler equations*. Collection of Technical Papers, 83-1944:373–382, 1983.
- [31] JOE F. THOMPSON, BHARAT K. SONI, NIGEL P. WEATHERILL (Herausgeber): *Handbook of Grid Generation*. CRC Press, 1999.
- [32] K. STÜBEN, U. TROTTENBERG: *Multigrid methods: fundamental algorithms, model problem analysis and applications*. Multigrid Methods. Springer-Verlag, Berlin, 1982.
- [33] KHOKHLOV, ALEXEI M.: *Fully Threaded Tree for Adaptive Refinement Fluid Dynamics Simulations*, 1997.
- [34] KLITZ, M.: *Homogenised Fluid Flow Equations in Porous Media with Application to Permeability Computations in Textiles*. Diplomarbeit, Institut für Numerische Simulation, Universität Bonn, 2006.
- [35] KREISS, BARBRO: *Construction of a Curvilinear Grid*. SIAM Journal on Scientific and Statistical Computing, 4(2):270–279, 1983.
- [36] OSHER, STANLEY und JAMES A SETHIAN: *Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*. Journal of Computational Physics, 79:12–49, 1988.

- [37] PENG, DANPING, BARRY MERRIMAN, STANLEY OSHER, HONGKAI ZHAO und MYUNGJOO KANG: *A PDE-based fast local level set method*. J. Comput. Phys., 155(2):410–438, November 1999.
- [38] PESKIN, CHARLES S.: *Flow patterns around heart valves: A numerical method*. J. Comput. Phys., 10(2):252–271, 1972.
- [39] POPINET, STÉPHANE: *Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries*. Journal of Computational Physics, 190(2):572–600, 2003.
- [40] QUARTERONI, ALFIO und ALBERTO VALLI: *Domain decomposition Methods for Partial Differential Equations*. 1999.
- [41] R. COURANT, K. FRIEDRICHS und H. LEWY: *Über die partiellen Differenzgleichungen der mathematischen Physik*. Mathematische Annalen, 100(1):32–74, 1928.
- [42] R. VERZICCO, J. MOHD-YUSOF, P. ORLANDI und D. HAWORTH: *Large Eddy simulation in complex geometry configurations using boundary body forces*. AIAA, 38:427–433, 2000.
- [43] REYNA, LUIS GUILLERMO M.: *I. Stability of Tchebyshev collocation. II. Interpolation for surfaces with 1-D discontinuities. III. On composite meshes*. Doktorarbeit, Dept. Appl. Math., California Institute of Technology, Pasadena, CA, 1982.
- [44] RUSSO, GIOVANNI und PETER SMEREKA: *A remark on computing distance functions*. J. Comput. Phys., 163(1):51–67, 2000.
- [45] S. MAJUMDAR, G. IACCARINO und P. DURBIN: *RANS solvers with adaptive structured boundary non-conforming grids*. Annual Research Briefs, Seiten 353–366, 2001.
- [46] SETHIAN, J.: *Curvature and the Evolution of Fronts*. Comm. in Math. Phys., 101:487–499, 1985.
- [47] SETHIAN, J.: *Curvature flow and entropy conditions applied to grid generation*. Journal of Computational Physics, 115:440–454, 1994.
- [48] SETHIAN, J.A.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Berkely, California, 1999.
- [49] SMITH, BARRY F., PETTER E. BJØRSTAD und WILLIAM D. GROPP: *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, New York, NY, USA, 1996.
- [50] STARIUS, G.: *Composite mesh difference methods for elliptic boundary value problems*. Numerische Mathematik, 28(2):243–258, 1977.

- [51] STARIUS, G.: *On composite mesh difference methods for hyperbolic difference equations*. Numerische Mathematik, 35(3):241–255, 1980.
- [52] SUSSMAN, MARK, PETER SMEREKA und STANLEY OSHER: *A level set approach for computing solutions to incompressible two-phase flow*. J. Comput. Phys., 114(1):146–159, 1994.
- [53] SUSSMAN, MARK, PETER SMEREKA und STANLEY OSHER: *A level set approach for computing solutions to incompressible two-phase flow*. J. Comput. Phys., 114(1):146–159, 1994.
- [54] TOSELLI, ANDREA und OLOF WIDLUND: *Domain Decomposition Methods – Algorithms and Theory*, Band 34 der Reihe *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg, 2005.
- [55] TSENG, YU-HENG und JOEL H. FERZIGER: *A ghost-cell immersed boundary method for flow in complex geometry*. J. Comput. Phys., 192(2):593–623, 2003.
- [56] U. TROTTENBERG, C. OOSTERLEE, A. SCHÜLLER: *Multigrid*. Academic Press, 2001.
- [57] WOHLMUTH, B.I.: *Hierarchical A Posteriori Error Estimators for Mortar Finite Element Methods with Lagrange Multipliers*. SIAM Journal on Numerical Analysis, 36(5):1636–1658, 1999.
- [58] WOHLMUTH, B.I.: *Discretization Techniques and Iterative Solvers Based on Domain Decomposition*. Lectures Notes in Computational Science and Engineering, 17, 2001.
- [59] XU, JINCHAO: *Iterative methods by space decomposition and subspace correction*. SIAM Rev., 34(4):581–613, 1992.
- [60] XU, JINCHAO und JUN ZOU: *Some Nonoverlapping Domain Decomposition Methods*. SIAM Rev., 40(4):857–914, 1998.