Rekonstruktionsverfahren auf unstrukturierten Gittern zur numerischen Simulation von Erhaltungsprinzipien

Dissertation
zur Erlangung des Doktorgrades
des Fachbereichs Mathematik
der Universität Hamburg

vorgelegt von

Daniel Hempel

aus Kassel

Hamburg
1999

Als Dissertation angenommen vom Fachbereich
Mathematik der Universität Hamburg

auf Grund der Gutachten von Prof. Dr. Thomas Sonar
und Prof. Dr. Klaus Glashoff

Hamburg, den 1. Juli 1999

Prof. Dr. Hans Daduna
Dekan des Fachbereichs Mathematik

Inhaltsverzeichnis

Einleitung **5**

1 Numerik hyperbolischer Erhaltungsgleichungen **9**
Skalare Erhaltungsgleichungen 11
Lineare Systeme in einer Raumdimension 13
Die Euler-Gleichungen der Gasdynamik 14
Finite-Volumen-Verfahren 16
Zeitschrittverfahren 17
Finite-Volumen-Gitter 19
Randintegration 20
Rekonstruktion 21
Numerische Flußfunktionen 23
Randbedingungen 33

2 Rekonstruktion der Zustandsvariablen **35**
Rekonstruktion mit Polynomen 39
Konvergente Verfahren 39
Lokale Ausgleichsprobleme 42
Konstruktion lokaler Gitter 46
Algorithmensammlung für bivariate Polynome 47
Limitierung zentraler Rekonstruktionen 53
Das Verfahren von Barth und Jespersen 54
Ein ordnungserhaltendes Limitierungsverfahren 55
Gewichtete ENO-Rekonstruktionen 63
Vergleich der Verfahren in numerischen Beispielen 65
Geschwindigkeitsvergleich der Verfahren 78

3 Optimale Rekonstruktion in Semi-Hilberträumen **81**
Das Ausgleichsproblem (Fortsetzung) 82
Optimale Rekonstruktion 84
Reellwertige Funktionen 88

Thin-Plate-Splines als Beispiel 90
Basisfunktionen mit kompaktem Träger 91
Radiale Basisfunktionen für ENO-Verfahren 92

4 Adaption und Parallelisierung **99**
Gitteradaption 99
Rot-Grün-Adaption 99
Interpolation von Daten zwischen Gittern 103
Adaptionsindikatoren für instationäre Probleme 104
Parallelisierung 111
Parallelisierung der Rekonstruktion 112
Parallelisierung der Flußschleife 113
Beschleunigungsraten 114
Ergebnisse adaptiver & paralleler Rechnungen 114

Zusammenfassung und Ausblick **121**

Symbolverzeichnis **125**

Literaturverzeichnis **127**

Einleitung

In vielen technischen und wissenschaftlichen Bereichen ermöglicht der Computer, das klassische Experiment durch die numerische Simulation zu ergänzen oder gar zu ersetzen. Im Bereich der Strömungsmechanik ist dies besonders interessant, weil dort Experimente oft sehr aufwendig und teilweise auch unmöglich sind. Zu den Anwendungen der numerischen Simulation in der Strömungsmechanik gehören beispielsweise der Flugzeug- und Automobilbau, die Chemie und die Medizin.
Die vorliegende Arbeit beschäftigt sich mit numerischen Verfahren zur Simulation von Erhaltungsprinzipien, zu denen insbesondere die Gesetze der Strömungsmechanik zählen. Im Vordergrund steht die Entwicklung neuer numerischer Methoden, die sich dicht an Anforderungen der Technik orientieren: Der wohl wichtigste Aspekt dieser Arbeit ist die Ausrichtung auf numerische und algorithmische Methoden für unstrukturierte Gitter. Diese ermöglichen es, komplexe geometrische Körper und Volumina darzustellen. Thema dieser Arbeit ist weniger die Erstellung solcher Gitter, als vielmehr die Konstruktion effizienter Verfahren zur Simulation von Strömungsprozessen mit diesen Gittern.
Im Vordergrund steht das Problem, physikalische Zustandsgrößen, wie beispielsweise den Druck oder die Dichte eines Gases, als Funktion auf unstrukturierten Gittern darzustellen und zu berechnen. Die folgende Aufgabenstellung charakterisiert das Problem:

> Gegeben ist ein Gitter aus Zellen und für jede Zelle ist der Mittelwert einer physikalischen Größe bekannt. Rekonstruiere hieraus eine Funktion, die eine möglichst gute Approximation an die (nicht bekannte) Zustandsverteilung ist!

Wenn die gesuchte Zustandsverteilung hinreichend glatt ist, wird man zur Lösung der Aufgabe auf klassische lineare Interpolationsmethoden zurückgreifen können. Bei Gasströmungen im Bereich der Schallgeschwindigkeit tritt das Phänomen von Unstetigkeiten in Form endlicher Sprünge längs von Untermannigfaltigkeiten des Raumes auf. Diese führen bei der

numerischen Simulation zu starken Oszillationen in den Lösungen und die Verfahren werden instabil. Für die Entwicklung von nichtlinearen Interpolationsverfahren ist es wichtig, ein einfaches Gleichungsmodell der auftretenden Phänomene zu haben. Dieses bieten die Euler-Gleichungen der Gasdynamik. Sie beschreiben den Transport von Masse, Energie und Impuls in einem Gas, welches keinen Reibungskräften unterliegt. Das Problem der Rekonstruktion unstetiger Zustandsverteilungen wird in dieser Arbeit anhand dieses vereinfachten Modelles untersucht.
Für den Einsatz unstrukturierter Gitter sind mehrdimensionale räumliche Gebiete zu betrachten. Für das mathematische Ziel dieser Arbeit reicht es aus, ein Verfahren für die numerische Berechnung der zweidimensionalen Euler-Gleichungen zu verwenden. Im ersten Kapitel der Arbeit werden die Grundlagen eines expliziten Finite-Volumen-Verfahrens für die Euler-Gleichungen angegeben. Die dargestellte Theorie, die anzugebenden numerischen Flußfunktionen sowie das Konstruktionsprinzip der Finite-Volumen-Verfahren sind ohne Beschränkung der Raumdimension dort beschrieben. Die Einschränkung auf zwei Raumdimensionen trifft lediglich für die verwendeten unstrukturierten Dreiecksgitter zu.
Nach den Vorbereitungen werden im zweiten Kapitel Rekonstruktionsverfahren für die Finite-Volumen-Methode hergeleitet. Insbesondere wird eine eigens für diese Arbeit entwickelte Methode vorgestellt. Diese wird kritisch verglichen mit den aktuellen Entwicklungen der sogenannten ENO-Verfahren. Im zweiten Kapitel werden ausschließlich Verfahren untersucht, bei denen die rekonstruierten Funktionen Polynome sind. Die notwendigen Anforderungen an die Gitter für die Konstruktion konvergenter Verfahren werden erläutert, und es werden Algorithmen für die wesentlichen Operationen hergeleitet, die in einem Finite-Volumen-Verfahren vorkommen: Die Integration von Polynomen auf polygonalen Zellen und die Auswertung.
Im dritten Kapitel wird untersucht, ob es möglich und sinnvoll ist, andere Funktionen als nur Polynome zur Rekonstruktion zu verwenden. Hier sind besonders diejenigen Funktionenräume interessant, die quadratische Energiefunktionale minimieren (ähnlich wie die bekannten kubischen Splines in einer Raumdimension). Dies führt zur Theorie der optimalen Rekonstruktion in Semi-Hilberträumen und zu den radialen Basisfunktionen. Der Zusammenhang wird knapp dargestellt und dann auf den Rekonstruktionsfall innerhalb des Finite-Volumen-Verfahrens übertragen. Schließlich wird in einer kritischen Diskussion die Effizienz solcher Rekonstruktionsfunktionen untersucht und mit den polynomiellen Verfahren verglichen.
Neben den Ansätzen, die Güte numerischer Lösungen durch geeignete Rekonstruktionsfunktionen zu verbessern, kann man natürlich auch das Rechengitter verändern, denn die Qualität der numerischen Lösungen hängt

maßgeblich hiervon ab. Um bewegte Unstetigkeiten hochauflösend darzustellen, ist die lokale Gitteradaption ein besonders guter Ansatz, der jedoch einige schwer lösbare Problemstellungen aufwirft. Besonders wichtig ist hier die Zuverlässigkeit der Adaptionsindikatoren: Sie müssen das Entstehen und das Bewegen von Unstetigkeiten durch pessimistische Abschätzungen vorwegnehmen und gleichzeitig das Rechengitter hinreichend grob halten, um einen Rechenzeitgewinn gegenüber global feinen Gittern zu ermöglichen. Die Aufgabenstellung ist vergleichbar mit dem Erfassen scharfer Kanten bei der Kompression von bewegten Bildern. Jedoch unterliegt die Gitteradaption einigen zusätzlichen Nebenbedingungen: Bei der Gitteradaption ist die Verwendung zukünftiger Zustände aus Kostengründen kaum möglich, und zudem soll die Gitteradaption nicht nur einen Speichergewinn, sondern auch einen erheblichen Rechenzeitgewinn erwirtschaften. Die Gitteradaption lohnt nur, wenn sie deutlich mehr Zeit und Speicher einspart, als sie selbst kostet. Im letzten Kapitel der Arbeit wird eine Implementierung zur Adaption zweidimensionaler Triangulierungen zusammen mit algorithmischen Details vorgestellt, und es wird die Effizienz dieses Verfahrens mit global feinen Gittern in Beispielen verglichen.
Schließlich wird ein Ansatz zur Parallelisierung von expliziten Finite-Volumen-Verfahren skizziert, der im Rahmen dieser Arbeit implementiert wurde. Hier ist eine einfache Lösung gelungen, bei der keine Gebietszerlegung notwendig ist.

Kapitel 1

Numerik hyperbolischer Erhaltungsgleichungen

Eine Vielzahl physikalischer und chemischer Prozesse wird durch die Erhaltung bestimmter Größen beschrieben: Energie, Impuls, Masse, Stoffmengen bei chemischen Reaktionen. Das gemeinsame mathematische Modell solcher „Nullbilanzen“ ist der folgende Gleichungstyp:

$$\partial_t u = -\nabla_x \cdot (F \circ u). \tag{1.1}$$

Hierin bedeutet $u : \Omega \times [0,T] \to S$ die gesuchte, vom Ort und der Zeit abhängige Funktion „physikalischer“ Zustände $S \subseteq \mathbf{R}^s$, und die Menge $\Omega \subset \mathbf{R}^d$ ist ein Intervall beziehungsweise ein ebenes oder räumliches Gebiet. Die Anfangsverteilung $u_0(x) := u(x,0)$ zur Zeit $t = 0$ sei bekannt. Gesucht ist die Fortentwicklung von u bis zur Zeit $T > 0$. Diese Evolution von u wird in Gleichung (1.1) durch die Flußfunktion $F : S \to \mathbf{R}^{s\times d}$ beschrieben, die für jede der s Zustandskomponenten den Fluß in die d Raumrichtungen angibt. Der Einfachheit halber betrachten wir hier nur Flußfunktionen, die ausschließlich von den Zuständen abhängen und nicht von deren Ableitungen oder vom Ort oder gar der Zeit. Neben den Anfangsbedingungen seien zusätzlich Randbedingungen an die Zustandsverteilung oder an Flüsse auf $\partial\Omega$ zu erfüllen.
Der Differentialoperator ∂_t sei die partielle Ableitung nach der Zeit und $(\nabla_x \cdot)$ die räumliche Divergenz, welche auf die d-dimensionalen Flüsse in jeder der s Zustandsgleichung anzuwenden ist: Die k-te Zeile ($k = 1, \ldots, s$) der Erhaltungsgleichung (1.1) läßt sich dann ausführlich als

$$\partial_t u_k = -\sum_{i=1}^{d} \partial_{x_i}(F_{k,i} \circ u) \tag{1.2}$$

schreiben.

Die Gleichungen (1.1) bzw. (1.2) werden **Erhaltungsgleichungen** genannt, weil sie nach Integration über ein hinreichend glatt berandetes Volumen $\omega \subseteq \Omega$, Zeitintegration über ein Intervall $[t_1, t_2] \subseteq [0, T]$ und Anwendung der partiellen Integrationsregel in die folgende Form überführt werden können:

$$\int_\omega u_k(x, t_2)\ \mathsf{d}x = \int_\omega u_k(x, t_1)\ \mathsf{d}x - \int_{t_1}^{t_2} \int_{\partial\omega} \mathsf{n}(x) \cdot F_k(u(x,t))\ \mathsf{d}\sigma\ \mathsf{d}t. \tag{1.3}$$

Hierin sei $\mathsf{n}(x)$ der äußere Normalenvektor an den Rand $\partial\omega$ und σ sei eine geeignete Parametrisierung dieses Randes. Für eine Lösung u der Gleichung (1.1) wird die Beziehung (1.3) für beliebige Zeitintervalle $[t_1, t_2]$ und gleichzeitig für beliebige Lebesgue-meßbare Kontrollvolumina ω erfüllt, sofern die partielle Integrationsregel anwendbar ist.
Die Darstellung (1.3) wird **Erhaltungsform** genannt. In Worten beschreibt sie, daß die Zu- oder Abnahme der Zustände in einem Teilgebiet innerhalb eines Zeitintervalles gleich dem Fluß über den Rand dieses Gebietes in dieser Zeit ist. Quellen und Senken sind nach (1.3) ausgeschlossen. Allerdings können sie leicht durch zusätzliche additive Terme in (1.3) modelliert werden. Wir unterscheiden im nachfolgenden Text die Erhaltungsform von der Darstellung (1.1), die wir als **Divergenzform** bezeichnen.
Die Erhaltungsform hat gegenüber der Divergenzform den Vorteil, daß sie weder räumliche Differenzierbarkeit von $F(u(\,\cdot\,, t))$ noch zeitliche Differenzierbarkeit von $u(x,\,\cdot\,)$ voraussetzt. Sie ist somit eine abgeschwächte Form der Differentialgleichung (1.1). Weil für nichtlineare Flußfunktionen F selbst glatte Anfangszustände u_0 nach kurzen Zeiten Unstetigkeiten ausbilden können, sucht man allgemeiner nach schwachen Lösungen in Lebesgue-Räumen. Für eine Einleitung in die Theorie schwacher Entropielösungen sei auf das erste Kapitel der Arbeit [Son97a] und die dort angegebene Literatur verwiesen. Für die Herleitung numerischer Methoden gehen wir in dieser Arbeit von der eindeutigen Lösbarkeit der Erhaltungsform (1.3) aus und zitieren aus [Son97a] die sogenannte Rankine-Hugoniot-Bedingung, die die Ausbreitung von endlichen unstetigen Sprüngen senkrecht zu echten Untermannigfaltigkeiten des Raumes konsistent zu (1.3) beschreibt:

Satz 1.1 *Es sei u eine stückweise stetig differenzierbare Lösung des Systems (1.3). In Ω sei γ eine stetig differenzierbare Hyperfläche, längs der u eine Unstetigkeit aufweist. Die einseitigen Limiten an γ seien u_ℓ und u_r und $\mathsf{n} = (\mathsf{n}_t, \mathsf{n}_1, \ldots, \mathsf{n}_d)$ bezeichne die Einheitsnormale an γ (in $\mathbf{R} \times \mathbf{R}^d$). Dann gilt die Sprungbedingung (Rankine-Hugoniot-Bedingung)*

$$\forall (x,t) \in \gamma: \quad \mathsf{n}_t(u_\ell - u_r) + \sum_{i=1}^{d} \mathsf{n}_i \left(F_{\cdot,i}(u_\ell) - F_{\cdot,i}(u_r)\right) = 0 \tag{1.4}$$

In dieser Arbeit sollen ausschließlich hyperbolische Erhaltungsgleichungen betrachtet werden:

Definition 1.2 *Ein System von Erhaltungsgleichungen* (1.1) *heißt* ***hyperbolisch****, wenn die Flußfunktionen $F_{\cdot,i}$ für alle $i \in \{1, \ldots, d\}$ differenzierbar in S sind und für die $s \times s$-Ableitungsmatrizen $(D_u F_{\cdot,i})$ die folgende Bedingung erfüllt wird: Für jeden normierten Vektor $\mathsf{n} \in \mathbf{R}^d$ besitzt die $s \times s$-Matrix*

$$\mathsf{n} \cdot D_u F \quad = \quad \sum_{i=1}^{d} \mathsf{n}_i (D_u F_{\cdot,i})$$

stets s linear unabhängige Eigenvektoren $r_1(u, \mathsf{n}), \ldots, r_s(u, \mathsf{n}) \in S$ und zugehörige reelle Eigenwerte $\lambda_1(u, \mathsf{n}), \ldots, \lambda_s(u, \mathsf{n}) \in \mathbf{R}$, die stetige Funktionen in u sind.

Diese Eigenschaft wird in den nachfolgenden Kapiteln einerseits bei der Herleitung analytischer Lösungen verwendet werden, andererseits werden wir sie bei der Konstruktion numerischer Verfahren einsetzen. Nach diesen einleitenden Definitionen wollen wir nun die in dieser Arbeit betrachteten Modellgleichungen diskutieren.

Skalare Erhaltungsgleichungen

Bei skalaren Erhaltungsgleichungen ist der Zustandsraum eindimensional: $s = 1$. Wir betrachten der Einfachheit halber den Fall $S = \mathbf{R}^1$. Die Erhaltungsgleichung ist bereits dann hyperbolisch, wenn die Flußfunktion stetig differenzierbar ist: Wir definieren $\nu_i(u) := D_u F_{1,i}(u) \in \mathbf{R}$. Dies ist der Eigenwert für den Normalenvektor e_i, dem i-ten Einheitsvektor. Für einen allgemeinen Normalenvektor n erhält man den Eigenwert $\lambda_1(u, \mathsf{n}) = \mathsf{n} \cdot \nu(u)$ zum Eigenvektor $1 = r_1(u, \mathsf{n}) \in S$.
Wesentliches Merkmal skalarer hyperbolischer Erhaltungsgleichungen ist, daß für stetig differenzierbares u Funktionswerte entlang gerader Bahnen mit konstanter Geschwindigkeit „verschoben“ werden: Für den Nachweis dieser Aussage suchen wir stetig differenzierbare Kurven $\phi : [0, T] \to \Omega$, entlang derer der Zustand $u(\phi(t), t)$ zeitlich konstant ist:

$$\begin{aligned} 0 \quad \stackrel{!}{=} \quad & \frac{\mathsf{d}}{\mathsf{d}t} u(\phi(\cdot), \cdot) \quad = \quad (\nabla_x u)^{\mathsf{t}}|_{(\phi(\cdot),\cdot)} \frac{\mathsf{d}}{\mathsf{d}t}\phi + (\partial_t u)|_{(\phi(\cdot),\cdot)} \\ = \quad & \sum_{i=1}^{d} (\partial_{x_i} u) \frac{\mathsf{d}}{\mathsf{d}t}\phi_i - (\partial_{x_i} F_{\cdot,i} \circ u) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{d} (\partial_{x_i} u) \frac{\mathsf{d}}{\mathsf{d}t} \phi_i - (\partial_{x_i} u) D_u F_{1,i}(u(\phi(t), t)) \\
&= \sum_{i=1}^{d} (\partial_{x_i} u) \left(\frac{\mathsf{d}}{\mathsf{d}t} \phi_i - \nu_i(u(\phi(t), t)) \right).
\end{aligned}
$$

Die Summanden verschwinden unabhängig von der räumlichen Ableitung $\partial_{x_i} u$, falls für den Weg ϕ die folgende Differentialgleichung erfüllt wird:

$$
\frac{\mathsf{d}}{\mathsf{d}t} \phi(t) = \nu(u(\phi(t), t)). \tag{1.5}
$$

Da der rechtsstehende Ausdruck nur vom Zustand $u(\phi(t), t)$ abhängt, dieser aber gerade konstant entlang des Weges ϕ ist, ist somit der Geschwindigkeitsvektor $\frac{\mathsf{d}}{\mathsf{d}t} \phi(t)$ zeitlich konstant. Dies bedeutet aber gerade, daß der Zustand $u(\phi(t), t)$ auf gradlinig gleichförmigen Bahnen verschoben wird.
Für den Fall, daß diese Bahnen untereinander überschneidungsfrei sind und vollständig innerhalb des Gebietes Ω verlaufen, kann man für bestimmte Konfigurationen mittels (1.5) explizite Lösungen konstruieren: Für den einfachen Fall der **linearen Advektionsflüsse**

$$
F^A_{1,i}(u) = u\nu \tag{1.6}
$$

mit vorgegebenen $\nu \in \mathbf{R}^d$ und $\Omega = \mathbf{R}^d$ wird jeder Zustand gradlinig gleichförmig mit der Geschwindigkeit ν verschoben. Die explizite Lösung der Differentialgleichung (1.1) lautet dann

$$
u(x, t) = u_0(x - t \cdot \nu). \tag{1.7}
$$

Diese ist, unabhängig von der Glätte von $u_0 \in L^1(\Omega \to \mathbf{R})$, eine zulässige Lösung der Erhaltungsgleichung (1.3). Sie ist zudem für Unstetigkeiten konsistent mit der Rankine-Hugoniot-Bedingung (1.4).
Für einfache numerische Experimente mit der linearen Advektionsgleichung (1.6) wurde innerhalb dieser Arbeit stets die Transportrichtung $\nu = (1, \ldots, 1)^{\mathsf{t}}$ verwendet.
Neben der linearen Advektion wurde als nichtlineares, skalares Modellproblem eine mehrdimensional verallgemeinerte Variante von Burgers' Gleichung mit der Flußfunktion

$$
F^B_{1,i}(u) = u^2 \tag{1.8}
$$

und der von u abhängenden Transportgeschwindigkeit $\nu(u) = (2u, \ldots, 2u)^{\mathsf{t}}$ gerechnet. Die charakterischen Bahnen sind bei diesem Beispiel stets parallel. Weil jedoch die Transportgeschwindigkeiten proportional von u abhängen, können sich die Bahnen überlagern. In diesen Fällen bilden sich Unstetigkeiten entlang von Untermannigfaltigkeiten (Stöße) heraus, deren Ausbreitungsverhalten wieder durch die Rankine-Hugoniot-Bedingung (1.4) beschrieben wird.

Lineare Systeme in einer Raumdimension

Im Falle $d = 1$ betrachten wir hier den Spezialfall linearer Flüsse

$$\begin{aligned} F^L(u) &:= Au \\ D_u F &= A \end{aligned}$$

mit einer Matrix $A \in \mathbf{R}^{s\times s}$, die mit den linear unbhängigen Eigenvektoren $r_1, \ldots, r_s \in S := \mathbf{R}^s$ diagonalisierbar sei und die Eigenwerte $\lambda_1 \leq \ldots \leq \lambda_s$ habe. Unter diesen Voraussetzungen ist das System hyperbolisch.
Mit der Matrix

$$R := (r_1, \ldots, r_s)$$

existiert nach Voraussetzung die Inverse, und es gilt

$$D := \mathrm{diag}(\lambda_1, \ldots, \lambda_s) = R^{-1} A\, R.$$

Ersetzt man nun in der Erhaltungsgleichung (1.3) die Variable u durch

$$\tilde{u} := R^{-1} u \in \mathbf{R}^s,$$

dann erhält man aus (1.3) für $k \in \{1, \ldots, s\}$ und nach dem Einsetzen der Diagonalisierungseigenschaft die Gleichungen

$$\int_\omega \tilde{u}_k(x, t_2)\ \mathsf{d}x = \int_\omega \tilde{u}_k(x, t_1)\ \mathsf{d}x - \int_{t_1}^{t_2} \int_{\partial\omega} \lambda_k \tilde{u}_k(x, t)\ \mathsf{d}\sigma\ \mathsf{d}t.$$

Dies sind genau s voneinander entkoppelte, skalare, lineare Advektionsaufgaben mit den Flußfunktionen $F_k(\tilde{u}_k) = \lambda_k \tilde{u}_k$. Die Lösungen hiervon hatten wir im letzten Kapitel explizit angegeben. Sie sind Translationen der Anfangsverteilung u_0:

$$\tilde{u}_k(x, t) = R_k^{-1} u_0(x - \lambda_k t).$$

R_k^{-1} sei hier die k-te Zeile von R^{-1}. Insgesamt kann damit die Lösung des linearen Systems in einer Raumdimension angegeben werden:

$$u(x, t) = R \begin{pmatrix} R_1^{-1} u_0(x - \lambda_1 t) \\ \vdots \\ R_s^{-1} u_0(x - \lambda_s t) \end{pmatrix} = \sum_{k=1}^{s} (R_k^{-1} u_0(x - \lambda_k t)) r_k. \tag{1.9}$$

Wir werden diese explizite Lösung später zur Konstruktion numerischer Flußfunktionen verwenden.

Die Euler-Gleichungen der Gasdynamik

Die Euler-Gleichungen der Gasdynamik beschreiben die Ausbreitung eines Gases unter Vernachlässigung der dämpfenden Reibungskräfte. Die Gleichungen können als gutes Modell für Überschallströmungen angesehen werden, bei denen weder Wärme zu- noch abgeführt wird. Für den Numeriker beinhalten sie einen besonders interessanten Aspekt der Strömungsmechanik, nämlich die Entstehung und Ausbreitung von niederdimensionalen Unstetigkeiten.
Der zu den Euler-Gleichungen gehörige Zustandsraum besteht aus der Massendichte ρ des Gases, seiner Impulsdichte $\rho\mathsf{v} \in \mathbf{R}^d$ und der Totalenergiedichte ρE. Die Flüsse für die i-te Raumrichtung ($i = 1, \ldots, d$) sind:

$$F_{\cdot,i}(u) = \begin{pmatrix} \rho\mathsf{v}_i \\ \rho\mathsf{v}_i \cdot \mathsf{v} + \mathsf{p} \cdot e_i \\ (\rho E + \mathsf{p})\mathsf{v}_i \end{pmatrix}, \qquad u = \begin{pmatrix} \rho \\ \rho\mathsf{v} \\ \rho E \end{pmatrix}. \tag{1.10}$$

Hierin ist e_i der i-te Einheitsvektor in $\mathbf{R}^d$ und folglich ist die mittlere Zeile als Kurzschreibweise für d verschiedene Zeilen zu lesen. Die Größe p ist der Druck, der im Falle eines idealen Gases durch

$$\mathsf{p} = (\kappa - 1)\left(\rho E - \tfrac{1}{2}\rho\mathsf{v}^\mathsf{t}\mathsf{v}\right) \tag{1.11}$$

als Funktion der übrigen Zustände beschrieben wird. Die Konstante κ nimmt für zweiatomige Gase, beispielsweise trockene Luft, den Wert 7/5 an. Zur Verkürzung nachfolgender Ausdrücke führen wir zusätzlich die folgende Konstante ein:

$$\tilde{\kappa} \;:=\; \kappa - 1.$$

Die Jacobi-Matrix der Euler-Flüsse (1.10) lautet allgemein:

$$D_u F_{\cdot,i}(u) \;=\; \begin{pmatrix} 0 & e_i^\mathsf{t} & 0 \\ \frac{1}{2}\tilde{\kappa}(\mathsf{v}^\mathsf{t}\mathsf{v})e_i - \mathsf{v}_i\mathsf{v} & \mathsf{v}e_i^\mathsf{t} + \mathsf{v}_i \cdot I - \tilde{\kappa}(e_i\mathsf{v}^\mathsf{t}) & \tilde{\kappa}e_i \\ (\tilde{\kappa}\mathsf{v}^\mathsf{t}\mathsf{v} - \kappa E)\mathsf{v}_i & (\kappa E - \frac{1}{2}\tilde{\kappa}\mathsf{v}^\mathsf{t}\mathsf{v})e_i^\mathsf{t} - \tilde{\kappa}\mathsf{v}_i\mathsf{v}^\mathsf{t} & \kappa\mathsf{v}_i \end{pmatrix}.$$

In der Mitte dieser Matrix sind jeweils d Zeilen und d Spalten angegeben, und mit I ist die $d \times d$-Identitätsmatrix bezeichnet. Entsprechend erhält man den Anteil in Richtung eines normierten Vektors $\mathsf{n} \in \mathbf{R}^d$:

$$\mathsf{n} \cdot D_u F \;=\; \begin{pmatrix} 0 & \mathsf{n}^\mathsf{t} & 0 \\ \frac{1}{2}\tilde{\kappa}(\mathsf{v}^\mathsf{t}\mathsf{v})\mathsf{n} - \mathsf{v}_\mathsf{n}\mathsf{v} & \mathsf{v}\mathsf{n}^\mathsf{t} + \mathsf{v}_\mathsf{n} \cdot I - \tilde{\kappa}(\mathsf{n}\mathsf{v}^\mathsf{t}) & \tilde{\kappa}\mathsf{n} \\ (\tilde{\kappa}\mathsf{v}^\mathsf{t}\mathsf{v} - \kappa E)\mathsf{v}_\mathsf{n} & (\kappa E - \frac{1}{2}\tilde{\kappa}\mathsf{v}^\mathsf{t}\mathsf{v})\mathsf{n}^\mathsf{t} - \tilde{\kappa}\mathsf{v}_\mathsf{n}\mathsf{v}^\mathsf{t} & \kappa\mathsf{v}_\mathsf{n} \end{pmatrix}.$$

Hierin sei

$$\mathsf{v_n} \;:=\; \mathsf{v}^{\mathsf{t}}\mathsf{n}.$$

Die Eigenwerte der Matrix $\mathsf{n} \cdot D_u F$ sind

$$\begin{aligned} \lambda_1(u,\mathsf{n}) &= \mathsf{v_n} - \mathsf{a} \\ \lambda_2(u,\mathsf{n}) = \ldots = \lambda_{d+1}(u,\mathsf{n}) &= \mathsf{v_n} \\ \lambda_{d+2}(u,\mathsf{n}) &= \mathsf{v_n} + \mathsf{a}. \end{aligned} \tag{1.12}$$

Die lokale Schallgeschwindigkeit a ist durch

$$\mathsf{a} \;:=\; \sqrt{\kappa \frac{\mathsf{p}}{\rho}}$$

definiert.
Die zu den Eigenwerten (1.12) gehörigen Eigenvektoren sind

$$\begin{aligned} r_1(u,\mathsf{n}) &= \begin{pmatrix} 1 \\ \mathsf{v} - \mathsf{an} \\ E + \frac{\mathsf{p}}{\rho} - \mathsf{av_n} \end{pmatrix} \\ r_2(u,\mathsf{n}) &= \begin{pmatrix} 1 \\ \mathsf{v} \\ \frac{1}{2}(\mathsf{v}^{\mathsf{t}}\mathsf{v}) \end{pmatrix} \\ r_{i+2}(u,\mathsf{n}) &= \begin{pmatrix} 0 \\ \tau_i \\ \mathsf{v}^{\mathsf{t}}\tau_i \end{pmatrix} \quad \forall i \in \{1,\ldots,d-1\} \\ r_{d+2}(u,\mathsf{n}) &= \begin{pmatrix} 1 \\ \mathsf{v} + \mathsf{an} \\ E + \frac{\mathsf{p}}{\rho} + \mathsf{av_n} \end{pmatrix}. \end{aligned}$$

Hierbei sei $(\mathsf{n}, \tau_1, \ldots, \tau_{d-1})$ eine Orthonormalbasis des $\mathbf{R}^d$. Wir definieren die Matrix

$$R(u,\mathsf{n}) := \left(r_1(u,\mathsf{n}), \ldots, r_{d+2}(u,\mathsf{n})\right).$$

Die Inverse dieser Matrix ist

$$(R(u,\mathsf{n}))^{-1} \;=\; \frac{1}{2\mathsf{a}^2} \begin{pmatrix} \frac{1}{2}\tilde{\kappa}\mathsf{v}^{\mathsf{t}}\mathsf{v} + \mathsf{v_n a} & -\tilde{\kappa}\mathsf{v}^{\mathsf{t}} - \mathsf{an}^{\mathsf{t}} & \tilde{\kappa} \\ 2\mathsf{a}^2 - \tilde{\kappa}\mathsf{v}^{\mathsf{t}}\mathsf{v} & 2\tilde{\kappa}\mathsf{v}^{\mathsf{t}} & -2\tilde{\kappa} \\ -2\mathsf{a}^2\mathsf{v}^{\mathsf{t}}\tau_1 & 2\mathsf{a}^2\tau_1^{\mathsf{t}} & 0 \\ \vdots & \vdots & \vdots \\ -2\mathsf{a}^2\mathsf{v}^{\mathsf{t}}\tau_{d-1} & 2\mathsf{a}^2\tau_{d-1}^{\mathsf{t}} & 0 \\ \frac{1}{2}\tilde{\kappa}\mathsf{v}^{\mathsf{t}}\mathsf{v} - \mathsf{v_n a} & -\tilde{\kappa}\mathsf{v}^{\mathsf{t}} + \mathsf{an}^{\mathsf{t}} & \tilde{\kappa} \end{pmatrix}.$$

Dies kann man leicht durch Ausmultiplizieren bestätigen. Es folgt, daß die Matrix $R(u, \mathsf{n})$ regulär ist und damit die Eigenvektoren linear unabhängig sind. Schließlich hängen die angegebenen Eigenwerte stetig von u ab, und somit ist das System der Euler-Gleichungen hyperbolisch gemäß Definition 1.2. Mit den angegebenen Matrizen läßt sich $\mathsf{n} \cdot D_u F$ diagonalisieren:

$$(R(u,\mathsf{n}))^{-1} \, (\mathsf{n} \cdot D_u F) \; R(u,\mathsf{n}) \;\; = \;\; \mathrm{diag}(\lambda_1(u,\mathsf{n}), \ldots, \lambda_{d+2}(u,\mathsf{n})).$$

Dieser Sachverhalt wird im Zusammenhang mit der Konstruktion numerischer Flußfunktionen im nächsten Kapitel verwendet werden.

Finite-Volumen-Verfahren

Bei einem Finite-Volumen-Verfahren geht man von der Erhaltungsform (1.3) aus und fordert diese Beziehung nicht für alle möglichen Kontrollvolumina, sondern nur für eine zuvor festgelegte **endliche** Menge $\mathcal{G}$ sogenannter **Zellen**. Jedes Element $\omega \in \mathcal{G}$ sei eine offene Teilmenge von Ω, für die die Anwendung der partiellen Integrationsregel zulässig ist. Die Zellen seien paarweise disjunkt zueinander, und ihre Abschlüsse überdecken zusammen das Gebiet: $\bigcup_{\omega \in \mathcal{G}} \overline{\omega} = \overline{\Omega}$.
Für die Beschreibung des Finite-Volumen-Verfahrens benötigen wir nun zu den Zellen $\omega \in \mathcal{G}$ die **Zellmittellungsfunktionale**:

$$\begin{aligned} \delta_\omega : L^1(\omega \to \mathbf{R}) \;\; &\longrightarrow \;\; \mathbf{R}, \\ \delta_\omega(v) \;\; &:= \;\; \frac{1}{|\omega|} \int\limits_\omega v \; \mathsf{d}x. \end{aligned}$$

Die Größe $|\omega|$ ist der Rauminhalt (Länge, Fläche, Volumen) der Zelle ω.
Von der Funktion u sucht man nun für die Zellen $\omega \in \mathcal{G}$ die Mittelwerte zu festen Zeiten $0 = t_0 < t_1 < \ldots < t_N = T$. Es sind also die endlich vielen Werte

$$\bar{u}_k(\omega, t_n) \;\; := \;\; \delta_\omega(u_k(\,\cdot\,, t_n))$$

für alle $k \in \{1, \ldots, s\}$, $\omega \in \mathcal{G}$ und $n \in \{0, \ldots, N\}$ gesucht.
Für $t_0 = 0$ erhält man diese Werte aus der als bekannt vorausgesetzten Anfangsverteilung u_0:

$$\bar{u}_k(\omega, 0) \;\; = \;\; \delta_\omega(u_0).$$

Für alle weiteren Zeiten folgt aus der Erhaltungsgleichung (1.3) ein Ansatz für ein Iterationsverfahren zur Bestimmung der Zellmittelwerte zur Zeit t_{n+1}

aus den Werten zur Zeit t_n:

$$\bar{u}(\omega, t_{n+1}) \;=\; \bar{u}(\omega, t_n) - \int_{t_n}^{t_{n+1}} \underbrace{\frac{1}{|\omega|} \int_{\partial\omega} \mathsf{n}(x) \cdot F(u(x,t))\ \mathsf{d}\sigma}_{=: f(\omega, u(\cdot,t))}\ \mathsf{d}t. \qquad (1.13)$$

Die Indizierung der Zustandskomponenten wurde der Übersichtlichkeit halber für die weitere Darstellung fortgelassen. Die Gleichung (1.13) ist also vektoriell mit s Zeilen zu lesen.
Entscheidend für die numerische Bestimmung des linken Wertes ist die **Approximation** der beiden rechtsstehenden Integrale. Die Diskretisierung des Zeitintegrales soll im nächsten Abschnitt beschrieben werden. Im Anschluß daran wird die numerische Berechnung des Integranden $f(\omega, u(\cdot, t))$ erklärt.

Zeitschrittverfahren

Für die numerische Berechnung des Zeitintegrales in (1.13) verwendet man typischerweise ein explizites oder implizites Runge-Kutta-Verfahren (siehe beispielsweise [Sch88, Kapitel 9.1]) Für die Beschreibung einer impliziten Methode speziell für die Euler- und Navier-Stokes-Gleichungen verweisen wir auf die Arbeit [Mei96]. Weil die Untersuchung von Zeitschrittverfahren nicht im Vordergrund dieser Arbeit steht, beschränken wir uns auf den einfacheren Fall expliziter Iterationsmethoden. Für diese Arbeit wurden drei Verfahren mit unterschiedlicher Integrationsordnung dem Artikel [SO88] entnommen. Wir geben die Iterationsverfahren an und erwähnen die für glatte Zustandsverteilungen u jeweils zu erwartende Fehlerordnung. Genauigkeitsbeweise und Stabilitätsanalysen sind in der angegebenen Quelle zu finden.

Bemerkungen zur Notation: Für die nachfolgende Angabe der Iterationsverfahren verzichten wir bei der Notation auf die Indizierung der Zellen und die der Zustandskomponenten: Unter $\bar{v}(t) \in S^{\mathcal{G}}$ verstehen wir einen Vektor aus Zellmitteln für alle Zellen und alle Zustandskomponenten zur Zeit t. In Ausdrücken, bei denen die Zuordnung der Zellmittel zu einem Zeitpunkt t nicht wesentlich ist, werden wir die funktionale Schreibweise aufgeben und stattdessen $\bar{v} \in S^{\mathcal{G}}$ schreiben. Um die Mittelwerte einer bestimmten Zelle $\omega \in \mathcal{G}$ zu bezeichnen, indizieren wir mit der Zelle ω: $\bar{v}_\omega(t) \in S^{\mathcal{G}}$. Um ferner Mittelwerte der k-ten Zustandskomponente zu indizieren, verwenden wir den Ausdruck $\bar{v}_k \in \mathbf{R}^{\mathcal{G}}$. Für den Mittelwert der k-ten Zustandskomponente der Zelle ω schreiben wir $\bar{v}_{k,\omega} \in \mathbf{R}$.
Für die Durchführung eines Runge-Kutta-Schrittes werden numerische Approximationen des Integranden $f(\omega, u(\,\cdot\,, t))$ zu einzelnen Zeitpunkten t

benötigt. Hierbei ist die Funktion $u(\,\cdot\,,t)$ zu diesen Zeitpunkten unbekannt. Dagegen werden jeweils numerisch approximierte Mittelwerte $\bar{v}_\omega \approx \bar{u}_\omega(t)$ zu diesen Zeiten zur Verfügung stehen. Daher gehen wir von einem Verfahren aus, welches zu gegebenen Zellmittelwerten $\bar{v}$ einer Funktion $v : \Omega \to S$ Approximationen an $(f(\omega, v))_{\omega\in\mathcal{G}}$ berechnet. Wir bezeichnen dieses Verfahren mit

$$\begin{array}{rccccc} \mathcal{F}: & S^{\mathcal{G}} & \longrightarrow & S^{\mathcal{G}} & & \\ & \bar{v} & \mapsto & \mathcal{F}(\bar{v}) & \approx & (f(\omega,v))_{\omega\in\mathcal{G}} \end{array}$$

Die numerische Realisierung dieser Abbildung $\mathcal{F}$ soll in den nachfolgenden Abschnitten erklärt werden. Für die Beschreibung der Zeitschrittiterationen können wir zunächst auf eine Konkretisierung verzichten und verwenden die Abbildung $\mathcal{F}$ abstrakt.
Das einfachste der verwendeten Verfahren ist die bekannte Vorwärts-Euler-Iteration, die für hinreichend glatte Funktionen u ein Verfahren erster Fehlerordnung in Abhängigkeit von der Zeitschrittweite $\Delta t_n := t_{n+1} - t_n$ ergibt:

$$\bar{v}(t_{n+1}) \;:=\; \bar{v}(t_n) - \Delta t_n \mathcal{F}(\bar{v}(t_n))$$

Als Iterationsmethode zweiter Ordnung diente die folgende zweistufige Formel mit den Zwischenwerten $\bar{w}(t_{n+1})$:

$$\begin{aligned} \bar{w}(t_{n+1}) &:= \bar{v}(t_n) - \Delta t_n \mathcal{F}(\bar{v}(t_n)) \\ \bar{v}(t_{n+1}) &:= \tfrac{1}{2}\bar{v}(t_n) + \tfrac{1}{2}\left(\bar{w}(t_{n+1}) - \Delta t_n \mathcal{F}(\bar{w}(t_{n+1}))\right) \end{aligned}$$

Weiterhin wurde das folgende Iterationsverfahren dritter Ordnung mit den temporären Daten $\bar{w}(t_{n+1})$ und $\bar{w}(t_n + \frac{1}{2}\Delta t_n)$ verwendet:

$$\begin{aligned} \bar{w}(t_{n+1}) &:= \bar{v}(t_n) - \Delta t_n \mathcal{F}(\bar{v}(t_n)) \\ \bar{w}(t_n + \tfrac{1}{2}\Delta t_n) &:= \tfrac{3}{4}\bar{v}(t_n) + \tfrac{1}{4}\left(\bar{w}(t_n) - \Delta t_n \mathcal{F}(\bar{w}(t_{n+1}))\right) \\ \bar{v}(t_{n+1}) &:= \tfrac{1}{3}\bar{v}(t_n) + \tfrac{2}{3}\left(\bar{w}(t_n + \tfrac{1}{2}\Delta t_n) - \Delta t_n \mathcal{F}(\bar{w}(t_n + \tfrac{1}{2}\Delta t_n))\right) \end{aligned}$$

Es sei noch erwähnt, daß die Zeitschrittweite Δt_n für explizite Verfahren der bekannten Courant-Friedrichs-Lewy-Bedingung, siehe [CFL28], als Stabilitätsgrenze unterliegt:

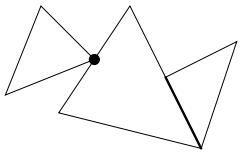
$$\Delta t_n \max\{\|\lambda(u(x,t_n),\mathsf{n})\|_\infty \,:\, x\in\Omega, \|\mathsf{n}\|_2 = 1\} \;\leq\; \mathsf{CFL}\cdot h. \quad (1.14)$$

Auf der linken Seite der Ungleichung ist das globale Betragsmaximum der Eigenwerte aller Matrizen $\mathsf{n}D_uF$ angegeben. Auf der rechten Seite bedeutet h den minimalen Inkreisradius der Zellen, und CFL ist eine vom Zeitschrittverfahren und der Realisierung der Abbildung $\mathcal{F}$ abhängige Größe. Sie wurde

im Rahmen dieser Arbeit stets in der Größenordnung 1 gewählt, und die Zeitschrittweite Δt_n wurde so groß gewählt, daß Gleichheit in (1.14) vorlag. Das Betragsmaximum der Eigenwerte von u wurde als Betragsmaximum der Eigenwerte der Zellmitteldaten $\bar{v}(t_n)$ approximiert.
Die Courant-Friedrichs-Lewy-Bedingung liefert eine Einschränkung für die Gestalt der Zellen im Gitter $\mathcal{G}$, da die Zeitschrittweite proportional zum Inkreisradius der Zellen ist und somit der Rechenaufwand für Gitter mit sehr feinen Zellen entsprechend steigt.
Nachdem nun die verwendeten Techniken für die Zeitschrittdiskretisierung erläutert wurden, werden wir uns im verbleibenden Teil dieser Arbeit mit der „räumlichen“ Diskretisierungen beschäftigen: Im nächsten Abschnitt werden Methoden zur Konstruktion von Finite-Volumen-Gittern beschrieben, und anschließend werden numerische Realisierungen der Abbildung $\mathcal{F}$ besprochen.

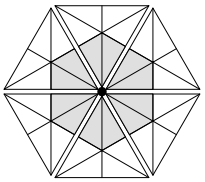
Finite-Volumen-Gitter

Bei eindimensionalen Problemen ist das Gebiet Ω ein Intervall, welches in Teilintervalle Ω_i zerlegt wird. Für die eindimensionalen Testprobleme in dieser Arbeit wurden immer äquidistante Zerlegungen des Einheitsintervalles $[0, 1]$ verwendet.
Bei Berechnungen mit zweidimensionalen Gebieten wurden konforme Triangulierungen verwendet, aus denen sogenannte Box-Gitter konstruiert wurden. Eine konforme Triangulierung ist ein endliche Menge von Dreiecken, die zusammen ein gegebenes Gebiet Ω überdecken und bei denen der Schnitt zweier Dreiecke jeweils entweder leer ist oder aus einer gemeinsamen Kante oder genau einem gemeinsamen Eckpunkt besteht. Somit sind die beiden am Rand gezeigten Konfigurationen ausgeschlossen, da hier die Schnitte der Dreiecke weder gemeinsame Kanten noch gemeinsame Eckpunkte bilden.[1] Mit Ausnahme einiger sehr einfacher Gitter wurden die in dieser Arbeit verwendeten Triangulierungen mit dem in [Fri93] beschriebenen Verfahren erzeugt.



Für ein Finite-Volumen-Verfahren könnte man nun direkt die Dreiecke als Zellen verwenden. Ein solcher Primärgitteransatz verursacht jedoch in der Praxis einige Probleme, denn zwischen Dreiecken, die sich nur in einem Eckpunkt berühren, werden innerhalb eines Zeitschrittes keine Flußbilanzen berechnet, obschon ihr geometrischer Abstand sehr gering ist. Deswegen konstruiert man mit den Dreiecken ein sekundäres Gitter: Jedem Gitterpunkt

[1] Die Überdeckung des Gebietes mittels endlich vieler Dreiecke ist natürlich nur im Falle einer polygonalen Berandung möglich. Im Falle gekrümmter Ränder kann eine Approximation durch Polygone vorgenommen werden, oder es müssen neben Dreiecken noch andere geometrische Figuren zugelassen werden.



der Triangulierung wird eine Zelle zugeordnet. Diese erhält man durch Teilung jedes Dreieckes durch die jeweils drei Seitenhalbierenden. Hierdurch entstehen aus einem Dreieck sechs neue Dreiecke. Von diesen Teildreiecken vereinigt man jeweils diejenigen zu einer gemeinsamen Zelle, die gemeinsam um einen Gitterpunkt der Ausgangstriangulierung liegen. In der nebenstehenden Abbildung ist dieser Konstruktionsprozeß illustriert.
Dreidimensionale Probleme werden in dieser Arbeit nicht behandelt. Allerdings sind nahezu alle in dieser Arbeit enthaltenen Strategien auf Probleme in drei Raumdimensionen übertragbar.

Randintegration

In einer Raumdimension kann die Integration über den Zellrand $\partial\omega$ einer Zelle $[a,b] \in \mathcal{G}$ in (1.13) aufgelöst werden:

$$\int_{\partial\omega} \mathsf{n}(x) \cdot F(u(x,t)) \; \mathsf{d}\sigma = F(u(b,t)) - F(u(a,t)).$$

In höheren Raumdimensionen dagegen wird das Randintegral durch numerische Quadraturformeln ersetzt, und der Integrand wird an vorgegebenen Punkten ausgewertet. Für die im letzten Abschnitt beschriebenen Boxgitter sind die Zellen polygonal. Hier kann die Randintegration als Summe der Integrale über die einzelnen Geradenstücke des Randes geschrieben werden. Sind $P_0, \ldots, P_{N-1}, P_N := P_0$ die im mathematisch positivem Drehsinn angeordneten N Eckpunkte des Polygons $\omega \in \mathcal{G}$, dann ergibt sich das Randintegral zu:

$$\int_{\partial\omega} \mathsf{n}(x) \cdot F(u(x,t)) \; \mathsf{d}\sigma = \sum_{j=0}^{N-1} \|P_{j+1} - P_j\|_2 \int_0^1 \mathsf{n}_j \cdot F(u((1-\theta)P_{j+1} - \theta P_j, t)) \; \mathsf{d}\theta$$

Der Normalenvektor n_j ist für ein Geradenstück konstant und kann als Normierung und 90°-Drehung des Vektors $P_{j+1} - P_j$ berechnet werden.
Die Integration über die Intervalle $[0,1]$ wird durch bekannte eindimensionale Quadraturformeln berechnet. Um mit möglichst wenigen Quadraturpunkten einen möglichst hohen Approximationsgrad zu erreichen, werden hier Gauß-Formeln, wie sie beispielsweise in [Sch88, Kapitel 8.4] zu finden sind, eingesetzt. Für die mit dem Finite-Volumen-Verfahren gerechneten Beispiele wurden die bekannten Ein- und Zweipunkte-Formeln verwendet:

$$\int_0^1 f(\theta) \; \mathsf{d}\theta \quad \approx \quad f(\tfrac{1}{2}),$$

$$\int_0^1 f(\theta)\ \mathsf{d}\theta \quad \approx \quad \tfrac{1}{2} f(\tfrac{1}{2} - \tfrac{1}{2\sqrt{3}}) + \tfrac{1}{2} f(\tfrac{1}{2} + \tfrac{1}{2\sqrt{3}}).$$

In den beiden Formeln bezeichnet f die zu integrierende Funktion. Die erste Formel ist exakt für alle Polynome bis zum Höchstgrad 1 und die zweite bis zum Höchstgrad 3.
Bei der Diskretisierung ist zu beachten, daß der Integrationsweg geschlossen ist und wegen der Multiplikation der Flußfunktion mit dem Normalenvektor somit ein Differentialausdruck vorliegt. Der Ausdruck ist numerisch sehr empfindlich gegenüber unregelmäßigen Diskretisierungen des Randes. Die Kantenstücke, über die numerisch integriert wird, sollten möglichst gleiche Länge besitzen, und die Zelle selbst sollte ein möglichst großes Verhältnis ρ_i/ρ_u zwischen Inkreis- ρ_i und Umkreisradius ρ_u haben.

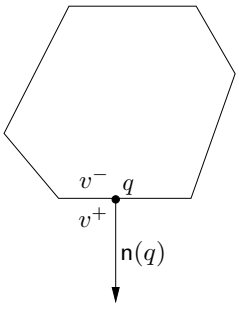
Rekonstruktion

Zur Bestimmung der Flüsse $\mathsf{n}(\mathsf{q}) \cdot F(u(\mathsf{q}, t))$ an den Quadraturpunkten q der im letzten Abschnitt beschriebenen Randintegrale wird zunächst aus den bekannten Zellmitteldaten $\bar{v}$ eine Rekonstruktionsfunktion $\mathcal{R}\bar{v}$ mit einem Rekonstruktionsalgorithmus $\mathcal{R}$ berechnet. Hierbei wird, um das Verfahren möglichst lokal zu gestalten, für jede Zelle aus den vorliegenden Zellmittelwerten und gegebenenfalls aus Zellmittelwerten benachbarter Zellen eine Rekonstruktionsfunktion mit möglichst hoher Approximationskraft bestimmt. Dabei ist zu berücksichtigen, daß die Funktion u Unstetigkeiten besitzen kann. Das Rekonstruktionsverfahren muß aus Stabilitätsgründen das Auftreten starker Oszillationen der rekonstruierten Funktionen in der Nähe von Unstetigkeiten vermeiden. Hierauf werden wir ausführlich im Kapitel 2 eingehen. Vorläufig beschreiben wir den Rekonstruktionsprozeß durch einen abstrakten Operator

$$\begin{aligned} \mathcal{R} : S^{\mathcal{G}} \quad &\longrightarrow \quad V(\Omega \to S), \\ \bar{v}(t) \quad &\mapsto \quad v(t) \approx u(\,\cdot\,, t), \end{aligned}$$

der für einen gegebenen Satz an Zellmitteldaten $\bar{v}(t)$ zur Zeit t eine Approximationsfunktion $v(t)$ berechnet. Der Funktionenraum $V(\Omega \to S)$ ist hierbei ein diskreter Funktionenraum für räumliche Zustandsverteilungen. Dieser setzt sich typischerweise aus lokalen Ansatzräumen V_ω für die einzelnen Zellen zusammen:

$$V(\Omega \to S) \quad := \quad \prod_{\omega \in \mathcal{G}} V_\omega.$$



An den Zellrändern sind die Rekonstruktionsfunktionen nicht wohldefiniert. Allerdings soll für jede Zelle die Funktion ihres Inneren auf den Rand stetig fortsetzbar sein. Für Zellränder, die im Inneren des Gebietes Ω liegen, grenzen jeweils zwei Zellen an eine Kante. Daher sind für jeden Quadraturpunkt eines inneren Zellrandes zwei Funktionswerte als einseitige Grenzwerte vorhanden. Wir bezeichnen diese beiden Grenzwerte gemäß der nebenstehenden Abbildung mit $v^- \in S$ und $v^+ \in S$. Diese beiden Zustände werden verwendet, um den physikalisch relevanten Fluß $\mathsf{n}(\mathsf{q}) \cdot F(u(\mathsf{q},t))$ in Richtung $\mathsf{n}(\mathsf{q})$ am Quadraturpunkt q zu approximieren. Diese Flußberechnung geschieht durch Einsatz einer numerischen Flußfunktion

$$\begin{aligned} H : S \times S \times \mathbf{R}^d &\longrightarrow S \\ (v^-, v^+, \mathsf{n}) &\longrightarrow H(v^-, v^+, \mathsf{n}). \end{aligned}$$

Die numerische Flußfunktion selbst soll stetig sein und sie soll den Fluß des zugehörigen eindimensionalen **Riemann-Problems** möglichst gut approximieren. Ein solches Problem ist durch die folgende (eindimensionale) Anfangsbedingung gegeben:

$$w_0(x) = \mathsf{H}(-x)v^- + \mathsf{H}(x)v^+. \tag{1.15}$$

Hierin ist H die Heaviside-Funktion ($\mathsf{H}(x) = 0$ für $x \leq 0$ sonst $\mathsf{H}(x) = 1$). Gesucht ist nun die Lösung des folgenden eindimensionalen Erhaltungsgleichungssystems mit der Flußfunktion $F_{\mathsf{n}} := \mathsf{n} \cdot F$:

$$\int_a^b w(x,t_2)\ \mathsf{d}x = \int_a^b w(x,t_1)\ \mathsf{d}x - \int_{t_1}^{t_2} F_{\mathsf{n}}(w(b,t)) - F_{\mathsf{n}}(w(a,t))\ \mathsf{d}t \tag{1.16}$$

für alle Intervalle $[a,b]$ und $[t_1,t_2]$. Ist w die Lösung des Problems unter den genannten Anfangsbedingungen, so soll die numerische Flußfunktion möglichst gut den Fluß an der Stelle 0 approximieren:

$$H(v^-, v^+, \mathsf{n}) \approx \lim_{t \searrow 0} \mathsf{n} \cdot F(w(0,t)). \tag{1.17}$$

Der Grenzwert wird für positive t gebildet, um für positive Zeitschritte Δt einen konsistenten Fluß zur Verfügung zu haben. Für stetige Übergänge $v := v^- = v^+$ soll ferner die numerische Flußfunktion H den korrekten Fluß $H(v,v,\mathsf{n}) = F_{\mathsf{n}}(v)$ liefern und sie soll Lipschitz-stetig in ihren ersten beiden Argumenten sein.
Neben den Zellrändern, die zum Inneren des Gebietes Ω gehören, gibt es noch solche, die am Rande $\partial\Omega$ des Gebietes liegen. Hier gibt es zunächst für die

Quadraturpunkte nur den Rekonstruktionswert v^-. Die Modellierung eines geeigneten Flusses $\mathsf{n} \cdot F$ ist abhängig von den jeweils gewählten Randbedingungen. Auch hier werden zum Teil numerische Flußfunktionen verwendet, und deswegen werden wir zuerst im nachfolgenden Abschnitt die im Rahmen dieser Arbeit eingesetzten numerischen Flußfunktionen angeben, um im Anschluß die Behandlung der Gebietsränder zu diskutieren.

Numerische Flußfunktionen

Wir wollen nun numerische Flußfunktionen zur Bestimmung des Flusses in (1.17) mit der Lösung aus Gleichung (1.16) unter den Anfangsbedingungen (1.15) konstruieren. Zunächst sollen wieder lineare Flußfunktionen $F_{\mathsf{n}}(u) := Au$ mit diagonalisierbarer Matrix A untersucht werden. Hierzu übernehmen wir die Notationen des Abschnittes „Lineare Systeme in einer Raumdimension“ auf Seite 13. Wir wollen die explizite Lösung des Riemann-Problems mittels (1.9) angeben. Sei hierzu

$$I^+ := \operatorname{diag}(\mathsf{H}(+\lambda_1), \ldots, \mathsf{H}(+\lambda_s)), \tag{1.18}$$

$$I^- := \operatorname{diag}(\mathsf{H}(-\lambda_1), \ldots, \mathsf{H}(-\lambda_s)). \tag{1.19}$$

Dann lautet die explizite Lösung an der Stelle $x = 0$ für $t > 0$:

$$\begin{aligned} w(0,t) &= R \begin{pmatrix} R_1^{-1} w_0(-\lambda_1 t) \\ \vdots \\ R_s^{-1} w_0(-\lambda_s t) \end{pmatrix} \\ &= R \begin{pmatrix} R_1^{-1} \left(v^- \mathsf{H}(\lambda_1) + \mathsf{H}(-\lambda_1) v^+\right) \\ \vdots \\ R_s^{-1} \left(v^- \mathsf{H}(\lambda_s) + \mathsf{H}(-\lambda_s) v^+\right) \end{pmatrix} \\ &= RI^+ R^{-1} v^- + RI^- R^{-1} v^+. \end{aligned}$$

Entspechend lautet der für positive Zeiten konsistente Fluß:

$$F_{\mathsf{n}}(w(0,t)) = Aw(0,t) = RDR^{-1} w(0,t) = RDI^+ R^{-1} v^- + RDI^- R^{-1} v^+.$$

Wir führen die Matrizen

$$A^+ := RDI^+ R^{-1},\ A^- := RDI^- R^{-1},\ |A| := A^+ - A^-. \tag{1.20}$$

ein und können so eine numerische Flußfunktion für lineare Systeme von Erhaltungsgleichungen als exakte Lösung des zugehörigen Riemann-Problems angeben:

$$H(v^-, v^+, \mathsf{n}) = A^+ v^- + A^- v^+ \tag{1.21}$$

Für die Flußfunktion (1.6) der linearen Advektion mit der Transportrichtung $\nu \in \mathbf{R}^d$ erhält man beispielsweise die numerische Flußfunktion

$$H(v^-, v^+, \mathsf{n}) = \begin{cases} (\mathsf{n} \cdot \nu) v^- & \text{für } \mathsf{n} \cdot \nu \geq 0, \\ (\mathsf{n} \cdot \nu) v^+ & \text{für } \mathsf{n} \cdot \nu \leq 0. \end{cases}$$

Für nichtlineare Flußfunktionen F_{n} ist die Jacobi-Matrix A eine Funktion der Zustände v. Somit kann eine einfache Zerlegung wie in (1.21) nicht angegeben werden. Für nichtlineare Systeme, wie den Euler-Gleichungen, ist es nur für spezielle Konfigurationen der Zustände v^- und v^+ möglich, den exakten Fluß eines Riemann-Problems anzugeben; siehe beispielsweise [Spe87, Kapitel 1.3]. Für die Konstruktion einiger bekannter numerischer Flußfunktionen geht man von einer äquivalenten, integralen Darstellung der numerischen Flußfunktion für den linearen Fall (1.21) aus:

$$\begin{aligned} H(v^-, v^+, \mathsf{n}) &= A^+ v^- + A^- v^+ \\ &= (A - A^-) v^- + A^- v^+ \\ &= A v^- + (A^- v^+ - A^- v^-) \\ &= A v^- + \int_{v^-}^{v^+} A^- \, \mathsf{d}v. \end{aligned}$$

Das Integral ist hier ein Wegintegral durch den Zustandsraum, welches allerdings von der Wahl des Verbindungsweges von v^- nach v^+ unabhängig ist, da jede Zeile A_k^- des Integranden der Gradient des Gradientenfeldes $A_k^- v$ ist. Für nichtlineare Systeme ist die Jacobi-Matrix A und somit auch die Matrix A^- vom jeweiligen Zustand v abhängig. Numerische Flußfunktionen für nichtlineare Systeme werden analog zur obigen Formel konstruiert:

$$H(v^-, v^+, \mathsf{n}) = F_{\mathsf{n}}(v^-) + \int_{v^-}^{v^+} A^-(v) \, \mathsf{d}v. \tag{1.22}$$

Hierbei ist A^- analog zu (1.19) und (1.20) definiert. Auf die Abhängigkeit von A^- vom Normalenvektor haben wir hier in der Notation verzichtet. In (1.22) ist zu beachten, daß für mehrdimensionale Zustandsräume das Wegintegral abhängig vom jeweils gewählten Weg ist, da im allgemeinen die Zeilen $A_k^-(v)$ nun keine Gradientenfelder mehr sind. Im skalaren Fall jedoch ist das Integral weiterhin wegunabhängig. Hier ist (1.22) gerade die numerische Flußfunktion von Engquist und Osher [EO81]. Für Burgers' Gleichung (1.8) erhält man beispielsweise

$$\begin{aligned} H(v^-, v^+, \mathsf{n}) &= \theta \cdot \left(\mathsf{H}(+\theta v^-) \cdot (v^-)^2 + \mathsf{H}(-\theta v^+) \cdot (v^+)^2 \right), \\ \theta &:= (1, \ldots, 1) \, \mathsf{n}. \end{aligned}$$

Osher und Solomon haben in [OS82] für das Integral (1.22) einen Weg angegeben, so daß das Integral aufgelöst werden kann und gleichzeitig die resultierende numerische Flußfunktion nach ihren ersten beiden Argumenten differenzierbar ist. Wir wollen im nachfolgenden diese Strategie erläutern und werden sie auf die Euler-Gleichungen anwenden. Die hier gewählte Darstellung orientiert sich an der Arbeit [Spe87, Kapitel 2.2]. Dort sind zwei verschiedene Methoden angegeben, von denen die eine als Osher-Variante bezeichnet wird und die andere als die physikalische Variante. Sie unterscheiden sich in der Sortierung der Eigenwerte $\lambda_1, \ldots, \lambda_s$ und wir werden beide Varianten zusammen erläutern. Hierfür benötigen wir zunächst die folgende

Definition 1.3 *Eine stetig differenzierbare Funktion $\psi^k : S \to \mathbf{R}$ heißt **Riemannsche Invariante** zum Eigenvektor r_k, wenn der Gradient $\nabla_u \psi^k(u)$ für alle $u \in S$ senkrecht auf dem Eigenvektor $r_k(u)$ steht:*

$$\nabla_u \psi^k(u) \cdot r_k(u) = 0 \quad \forall u \in S.$$

*Ist der Eigenwerte $\lambda_k(u)$ eine Riemannsche Invariante, so nennt man ihn **linear degeneriert**. Ist der Gradient des Eigenwertes $\lambda_k(u)$ dagegen niemals senkrecht auf $r_k(u)$, so nennt man ihn **echt nichtlinear**:*

$$\nabla_u \lambda_k(u) \cdot r_k(u) \neq 0 \ \forall u \in S.$$

In der Regel gibt es zu jedem Eigenvektor $s - 1$ Riemannsche Invarianten. Diese zeichnen sich dadurch aus, daß sie sich längs des Eigenvektors r_k nicht ändern. Für die Euler-Gleichungen geben wir die Riemannschen Invarianten im nachfolgenden Satz an. Dieser kann durch elementare Rechnungen bestätigt werden.

Satz 1.4 *Es gelten die Bezeichnungen des Abschnittes „Die Euler-Gleichungen der Gasdynamik“ auf Seite 14. Für $j \in \{1, \ldots, d-1\}$ definieren wir die folgenden Funktionen*

$$\begin{aligned}
\psi_i^1(u) &:= \tau_i^{\mathsf{t}} \mathsf{v}, \ i \in \{1, \ldots, d-1\}, \\
\psi_d^1(u) &:= \mathsf{v_n} + \frac{2}{\kappa - 1} \mathsf{a}, \\
\psi_{d+1}^1(u) &:= \ln\left(\frac{\mathsf{p}}{\rho^\kappa}\right), \\
\psi_i^2(u) &:= \tau_i^{\mathsf{t}} \mathsf{v}, \ i \in \{1, \ldots, d-1\}, \\
\psi_d^2(u) &:= \mathsf{v_n},
\end{aligned}$$

$$
\begin{aligned}
\psi_{d+1}^{2}(u) &:= \mathsf{p}, \\
\psi_{i}^{j+2}(u) &:= \tau_i^{\mathsf{t}}\mathsf{v},\ i \in \{1,\ldots,d-1\} \setminus \{j\}, \\
\psi_{j}^{j+2}(u) &:= \rho, \\
\psi_{d}^{j+2}(u) &:= \mathsf{v_n}, \\
\psi_{d+1}^{j+2}(u) &:= \mathsf{p}, \\
\psi_{i}^{d+2}(u) &:= \tau_i^{\mathsf{t}}\mathsf{v},\ i \in \{1,\ldots,d-1\}, \\
\psi_{d}^{d+2}(u) &:= \mathsf{v_n} - \frac{2}{\kappa-1}\mathsf{a}, \\
\psi_{d+1}^{d+2}(u) &:= \ln\left(\frac{\mathsf{p}}{\rho^\kappa}\right).
\end{aligned}
$$

Dann sind für $k \in \{1,\ldots,s\}$ und $j \in \{1,\ldots,s-1\}$ die Funktionen ψ_j^k Riemannsche Invarianten zum Eigenvektor r_k. Die zu einem Eigenwert angegebenen Riemannschen Invarianten haben linear unabhängige Gradienten. Weiterhin sind die Eigenwerte λ_1 und λ_s echt nichtlinear, und die übrigen Eigenwerte sind linear degeneriert.

Wir konstruieren nun den Weg Γ im Zustandsraum für das Integral in (1.22), der die beiden Zustände v^- und v^+ verbindet. Dieser Weg soll sich stückweise aus den Wegen Γ_k ($k \in \{1,\ldots,s\}$) zusammensetzen. Der Weg Γ_k sei über ein Intervall $[a_k, b_k]$ durch die Funktion γ_k parametrisiert und verbinde die beiden Zustände S_k und E_k:

$$
\begin{aligned}
\Gamma_k : [a_k, b_k] &\longrightarrow S \\
\gamma_k(a_k) = S_k &\qquad \gamma_k(b_k) = E_k.
\end{aligned}
$$

Wir fordern, daß die Richtung des Weges Γ_k parallel zum Eigenvektor r_k sei:

$$
\frac{\mathsf{d}\gamma_k}{\mathsf{d}\xi}(\xi) = r_k(\gamma_k(\xi)). \tag{1.23}
$$

Damit bestimmen wir nun den Wert des Integrales

$$
\begin{aligned}
\int_{\Gamma_k} A^-(u)\ \mathsf{d}u &= \int_{a_k}^{b_k} A^-(\gamma_k(\xi))\frac{\mathsf{d}\gamma_k}{\mathsf{d}\xi}(\xi)\ \mathsf{d}\xi \\
&= \int_{a_k}^{b_k} A^-(\gamma_k(\xi)) r_k(\gamma_k(\xi))\ \mathsf{d}\xi \\
&= \int_{a_k}^{b_k} \mathsf{H}(-\lambda(\gamma_k(\xi)))\,\lambda(\gamma_k(\xi))\,r_k(\gamma_k(\xi))\ \mathsf{d}\xi.
\end{aligned}
$$

Ist $\lambda_k \geq 0$ auf Γ_k, so verschwindet das obige Integral:

$$\int\limits_{\Gamma_k} A^-(u)\ \mathsf{d}u = 0.$$

Ist dagegen $\lambda_k \leq 0$ auf Γ_k, dann können wir das Integral wie folgt auflösen:

$$\begin{aligned}
\int\limits_{\Gamma_k} A^-(u)\ \mathsf{d}u &= \int\limits_{a_k}^{b_k} \lambda_k(\gamma_k(\xi))\, r_k(\gamma_k(\xi))\ \mathsf{d}\xi \\
&= \int\limits_{a_k}^{b_k} A(\gamma_k(\xi)))\, r_k(\gamma_k(\xi))\ \mathsf{d}\xi \\
&= \int\limits_{a_k}^{b_k} (D_u F_\mathsf{n})(\gamma_k(\xi))\, \frac{\mathsf{d}\gamma_k}{\mathsf{d}\xi}(\xi)\ \mathsf{d}\xi \\
&= F_\mathsf{n}(\gamma_k(b_k)) - F_\mathsf{n}(\gamma_k(a_k)) \ = \ F_\mathsf{n}(E_k) - F_\mathsf{n}(S_k). \quad (1.24)
\end{aligned}$$

Wechselt λ_k das Vorzeichen, so wird der Wert des Integrales sich als Summe derjenigen Intervalle darstellen lassen, auf denen λ_k negatives Vorzeichen hat. Auf diese Weise entstehen mehrere Summanden $F_\mathsf{n}(E_k^j) - F_\mathsf{n}(S_k^j)$ wie in (1.24).
Ist λ_k linear degeneriert, so ist der Gradient senkrecht auf r_k, und daher ist λ_k konstant auf dem Wege Γ_k. In diesem Fall kann das Integral wie oben beschrieben aufgelöst werden. Ist λ_k echt nichtlinear, so ändert $(\nabla_u \lambda_k) \cdot r_k$ das Vorzeichen nicht, und daher ist λ_k monoton auf γ_k. Dann gibt es höchstens eine Nullstelle $N_k \in \Gamma_k$ mit $\lambda_k(N_k) = 0$. In diesem Fall kann das Integral wie folgt angegeben werden:

$$\int\limits_{\Gamma_k} A^-(u)\ \mathsf{d}u = \begin{cases} F_\mathsf{n}(N_k) - F_\mathsf{n}(S_k) & \text{falls } \lambda_k(S_k) < 0 \\ F_\mathsf{n}(E_k) - F_\mathsf{n}(S_k) & \text{falls } \lambda_k(E_k) < 0 \end{cases}$$

Gibt es keine solche Nullstelle, so ändert λ_k das Vorzeichen nicht, und dann kann das Integral wie bereits beschrieben berechnet werden.
Sind die Flüsse, wie im Falle der Euler-Gleichungen, so beschaffen, daß alle Eigenwerte entweder linear degeneriert oder echt nichtlinear sind, dann kann das einfache Regelwerk angegeben werden:

$$\int\limits_{\Gamma_k} A^-(u)\ \mathsf{d}u = \begin{cases} 0 & \text{lin. deg. konstant} = \lambda_k \geq 0 \\ F_\mathsf{n}(E_k) - F_\mathsf{n}(S_k) & \text{lin. deg. konstant} = \lambda_k \leq 0 \\ 0 & \text{e. nichtlin. } \lambda_k(S_k) \geq 0,\ \lambda_k(E_k) \geq 0 \\ F_\mathsf{n}(E_k) - F_\mathsf{n}(S_k) & \text{e. nichtlin. } \lambda_k(S_k) \leq 0,\ \lambda_k(E_k) \leq 0 \\ F_\mathsf{n}(N_k) - F_\mathsf{n}(S_k) & \text{e. nichtlin. } \lambda_k(S_k) < 0,\ \lambda_k(N_k) = 0 \\ F_\mathsf{n}(E_k) - F_\mathsf{n}(N_k) & \text{e. nichtlin. } \lambda_k(N_k) = 0,\ \lambda_k(E_k) < 0 \end{cases}$$

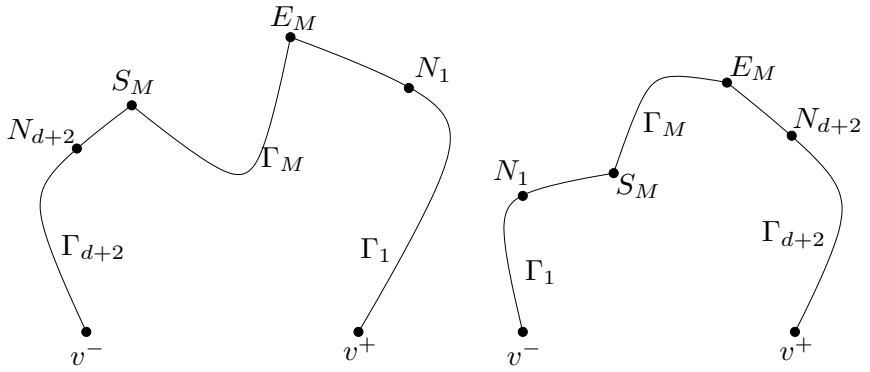


Abbildung 1.1: Links ist die Variante von Osher mit absteigend sortierten Eigenwerten und rechts ist die Variante mit aufsteigender Sortierung für die Sitution der Euler-Gleichungen dargestellt. Die Kurve Γ_M ergibt sich für den linear degenerierten Eigenwert. Gesucht ist der Ort der Punkte S_M und E_M. Die Punkte N_1 und N_{d+2} sind Punkte auf den Kurven Γ_1 und Γ_{d+2}, an denen der zugehörige Eigenwert das Vorzeichen wechselt (für jede der beiden Kurven gibt es höchstens einen solchen Punkt).

Diese Regeln demonstrieren, daß wir nur die Endpunkte S_k und E_k und gegebenenfalls die Zwischenpunkte N_k der Kurven bestimmen müssen. Die einzelnen Kurven sollen nun zu einem gemeinsamen Weg zusammengefügt werden. Hierdurch fallen von den s Kurven jeweils $s-1$ Anfangs- und Endpunkte zusammen. Da der Weg Γ bei v^- beginnt und bei v^+ endet, sind also nur $s-1$ Zwischenpunkte mit ihren s Koordinaten unbekannt. Auf jeder der s Kurven sind allerdings jeweils $s-1$ Riemannsche Invarianten konstant, weil ihre Gradienten wegen (1.23) senkrecht zu Γ_k sind. Dies liefert für die $s \cdot (s-1)$ Unbekannten $s \cdot (s-1)$ Gleichungen

$$\psi_j^k(S_k) = \psi_j^k(E_k)$$

für $j \in \{1, \ldots, s-1\}$ und $k \in \{1, \ldots, s\}$.
Nun muß man sich für eine Reihenfolge der Teilwege Γ_k entscheiden: Osher schlägt vor, diese Wege so anzuordnen, daß am Startpunkt v^- die Kurve für den größten Eigenwert beginnt und dann die Wege in absteigender Reihenfolge bis schließlich zur Kurve Γ_1, die mit dem Punkt v^+ enden soll. Dagegen schlägt Spekreijse gerade die umgekehrte Reihenfolge vor. Bei beiden Varianten ist jedoch wesentlich, daß es überhaupt eine Sortierung der Eigenwert**funktionen** gibt: Schließlich sind die Eigenwerte abhängig vom jeweiligen Zustand; die Größenverhältnisse könnten sich daher längs einer Kurve ändern. Für die Euler-Gleichungen tritt dieser Fall jedoch nicht auf, und die angegebene Reihenfolge der Eigenwerte ist stets aufsteigend:

$\lambda_1 \leq \lambda_2 = \ldots = \lambda_{d+1} \leq \lambda_{d+2}$. Wir werden daher im folgenden von der Sortierbarkeit der Eigenwerte ausgehen. Für zwei Eigenvektoren zum gleichen Eigenwert gibt es allerdings keine vorgegebene Sortierung. Hier tritt bei den Euler-Gleichungen ein weiterer Spezialfall auf: Der Eigenwert $\lambda_M := \lambda_2 = \ldots = \lambda_{d+1}$ ist linear degeneriert, und somit ist er auf den Wegen $\Gamma_2, \ldots, \Gamma_{d+1}$ konstant. Ist $\lambda_M \geq 0$, so verschwindet das Integral für alle zugehörigen Wege, und wir sehen, daß die Sortierung der Wege in diesem Fall nicht wesentlich ist. Wir untersuchen nun den Fall $\lambda_M < 0$: Da die Kurven zum gleichen Eigenwert in jeder Sortierung unterbrechungsfrei aneinander gereiht werden, entsteht eine zusammenhängende Kurve $\Gamma_M := \Gamma_2 \cup \ldots \cup \Gamma_{d+1}$ mit dem Anfangspunkt S_M und dem Endpunkt E_M. Da nach Voraussetzung $\lambda_M < 0$ ist, folgt analog zur Herleitung (1.24)

$$\int_{\Gamma_M} A^-(u)\ \mathsf{d}u = \int_{\Gamma_M} A(u)\ \mathsf{d}u = F_{\mathsf{n}}(E_M) - F_{\mathsf{n}}(S_M).$$

Es kommt also im Falle $\lambda_M < 0$ nur auf die Lage der beiden Punkte S_M und E_M an, nicht aber auf den genauen Verlauf und die Sortierung der zugehörigen Kurven $\Gamma_2, \ldots, \Gamma_{d+1}$. Für die Euler-Gleichungen reduziert sich das System auf die Anordnung dreier Wege Γ_1, Γ_M und Γ_{d+2} zu den verschiedenen Eigenwerten. Die beiden möglichen Varianten (aufsteigende oder absteigende Sortierung) sind in der Abbildung 1.1 skizziert.
Wir bestimmen nun für die Euler-Gleichungen die Zustände S_M und E_M. Sei hierzu:

$$v^- =: \begin{pmatrix} \rho^0 \\ \rho^0 \mathsf{v}^0 \\ \rho^0 E^0 \end{pmatrix},\ S_M =: \begin{pmatrix} \rho^s \\ \rho^s \mathsf{v}^s \\ \rho^s E^s \end{pmatrix}\ E_M =: \begin{pmatrix} \rho^e \\ \rho^e \mathsf{v}^e \\ \rho^e E^e \end{pmatrix}\ v^+ =: \begin{pmatrix} \rho^1 \\ \rho^1 \mathsf{v}^1 \\ \rho^1 E^1 \end{pmatrix}.$$

Gemäß Satz 1.4 ist die Normalengeschwindigkeit v_{n} und der Druck p eine Riemannsche Invariante auf dem Weg Γ_M. Es gilt daher

$$\begin{aligned} \mathsf{v}_{\mathsf{n}}^{se} &:= \mathsf{n} \cdot \mathsf{v}^s \overset{!}{=} \mathsf{n} \cdot \mathsf{v}^e, \\ \mathsf{p}^{se} &:= (\kappa - 1)(\rho^s E^s - \tfrac{1}{2}\rho^s(\mathsf{v}^s \cdot \mathsf{v}^s)) \overset{!}{=} (\kappa - 1)(\rho^e E^e - \tfrac{1}{2}\rho^e(\mathsf{v}^e \cdot \mathsf{v}^e)). \end{aligned}$$

Die Tangentialgeschwindigkeiten $\mathsf{v}^{\mathsf{t}}\tau_j$ sind Riemannsche Invarianten für alle $j \in \{1, \ldots, d\}$ auf den Kurven Γ_1 und Γ_{d+2}. Damit können bereits die Tangentalgeschwindigkeiten der Zustände S_M und E_M aus den Tangentialgeschwindigkeiten von v^- und v^+ bestimmt werden:

$$\begin{aligned} \mathsf{v}^s \cdot \tau_j &\overset{!}{=} \mathsf{v}^0 \cdot \tau_j, \\ \mathsf{v}^e \cdot \tau_j &\overset{!}{=} \mathsf{v}^1 \cdot \tau_j. \end{aligned}$$

Auch die Entropie (ψ^1_{d+1} und ψ^{d+2}_{d+1}) ist enlang der beiden Wege Γ_1 und Γ_{d+2} konstant. Da der Logarithmus strikt monoton ist, folgt

$$Z^0 := \frac{\mathsf{p}^{se}}{(\rho^s)^\kappa} \stackrel{!}{=} \frac{\mathsf{p}^0}{(\rho^0)^\kappa}, \tag{1.25}$$

$$Z^1 := \frac{\mathsf{p}^{se}}{(\rho^e)^\kappa} \stackrel{!}{=} \frac{\mathsf{p}^1}{(\rho^1)^\kappa}, \tag{1.26}$$

wobei p^0 und p^1 jeweils der Druck bei v^- und v^+ sei. Mit den beiden letzten Gleichungen kann man zusammen das Verhältnis der Dichte an den Punkten S_M und E_M bestimmen:

$$\theta_\rho := \frac{\rho^e}{\rho^s} = \frac{\rho^1}{\rho^0}\left(\frac{\mathsf{p}^0}{\mathsf{p}^1}\right)^{1/\kappa}.$$

Das Verhältnis der Schallgeschwindigkeiten an den beiden Punkten ergibt sich zu

$$\theta_\mathsf{a} := \frac{\mathsf{a}^s}{\mathsf{a}^e} = \sqrt{\theta_\rho}.$$

Die bisherigen Betrachtungen waren nicht davon abhängig, ob wir die aufsteigende Sortierung oder die absteigende Sortierung der Eigenwerte gewählt haben. Da sich die beiden entscheidenden Eigenwerte nur durch ein einziges Vorzeichen unterscheiden, führen wir ein vom jeweiligen Verfahren abhängiges Vorzeichen ein:

$$\mathsf{sig} := \begin{cases} +1 & \text{Sortierung absteigend,} \\ -1 & \text{Sortierung aufsteigend.} \end{cases} \tag{1.27}$$

Mit den Riemannschen Invarianten ψ^1_d und ψ^{d+2}_d folgt, daß der bei v^- startende Weg die Riemannsche Invariante

$$\mathsf{v_n} - \mathsf{sig}\frac{2}{\kappa-1}\mathsf{a} \tag{1.28}$$

besitzt. Der bei v^+ endende Weg hat entsprechend die Riemannsche Invariante

$$\mathsf{v_n} + \mathsf{sig}\frac{2}{\kappa-1}\mathsf{a}. \tag{1.29}$$

Damit folgt für die Zustände S_M und E_M:

$$\Psi^0 := \mathsf{n}\cdot\mathsf{v}^0 - \mathsf{sig}\frac{2}{\kappa-1}\mathsf{a}^0 \stackrel{!}{=} \mathsf{v}^{se}_\mathsf{n} - \mathsf{sig}\frac{2}{\kappa-1}\mathsf{a}^s, \tag{1.30}$$

$$\Psi^1 := \mathsf{n}\cdot\mathsf{v}^1 + \mathsf{sig}\frac{2}{\kappa-1}\mathsf{a}^1 \stackrel{!}{=} \mathsf{v}^{se}_\mathsf{n} + \mathsf{sig}\frac{2}{\kappa-1}\mathsf{a}^e. \tag{1.31}$$

Ψ^1 und Ψ^2 lassen sich mit den mittleren Ausdrücken berechnen, und man kann die beiden Gleichungen dann nach der gesuchten Normalgeschwindigkeit umstellen:

$$\mathsf{v}_\mathsf{n}^{se} = \frac{\Psi^0 + \theta_\mathsf{a}\Psi^1}{1+\theta_\mathsf{a}}.$$

Damit sind alle Geschwindigkeitskomponenten v^s und v^e von S_M und E_M berechenbar. Umgekehrt lassen sich dann mit (1.30) und (1.31) die Schallgeschwindigkeiten berechnen:

$$\begin{aligned}\mathsf{a}^s &= \mathsf{sig}\frac{\kappa-1}{2}(\mathsf{v}_\mathsf{n}^{se} - \Psi^0),\\ \mathsf{a}^e &= \frac{\mathsf{a}^s}{\theta_a}.\end{aligned}$$

Mit den Schallgeschwindigkeiten a^s und der Geschwindigkeit v^s kann die Energie E^s berechnet werden:

$$\begin{aligned}(\mathsf{a}^s)^2 &= \kappa\frac{\mathsf{p}^s}{\rho^s} = \kappa(\kappa-1)(E^s - \tfrac{1}{2}(\mathsf{v}^s\cdot\mathsf{v}^s))\\ \Rightarrow \quad E^s &= \frac{(\mathsf{a}^s)^2}{\kappa(\kappa-1)} + \tfrac{1}{2}(\mathsf{v}^s\cdot\mathsf{v}^s).\end{aligned}$$

Entsprechend folgert man

$$E^e = \frac{(\mathsf{a}^e)^2}{\kappa(\kappa-1)} + \tfrac{1}{2}(\mathsf{v}^e\cdot\mathsf{v}^e).$$

Mit den bekannten Größen Z^0 aus (1.25) lassen sich Dichte und Druck in der folgenden Reihenfolge bestimmen:

$$\begin{aligned}\rho^s &= \left(\frac{(\mathsf{a}^s)^2}{\kappa\cdot Z^0}\right)^{1/(\kappa-1)},\\ \mathsf{p}^{se} &= \frac{1}{\kappa}(\mathsf{a}^s)^2\rho^s,\\ \rho^e &= \kappa\frac{\mathsf{p}^{se}}{\mathsf{a}^e}.\end{aligned}$$

Damit sind die beiden Zustände S_M und E_M vollständig bestimmt.
Für die Auswertung der Integrale sind eventuell noch Nullstellen der entsprechenden Eigenwerte zu berechnen. Die Kurve, die am Punkt v^- beginnt, gehört zum Eigenwert $\mathsf{v_n} - \mathsf{sig}\cdot\mathsf{a}$. Eine Nullstelle liegt dann vor, wenn

$$(\mathsf{v}_\mathsf{n}^0 - \mathsf{sig}\cdot\mathsf{a}^0)\cdot(\mathsf{v}_\mathsf{n}^s - \mathsf{sig}\cdot\mathsf{a}^s) < 0.$$

Eine solche Nullstelle für diese Kurve bezeichnen wir mit N^ℓ, und entsprechend nennen wir eine Nullstelle auf der Kurve bei v^+ N^r. Der Eigenwert zur letzteren Kurve ist $\mathsf{v_n} + \mathsf{sig} \cdot \mathsf{a}$, und entsprechend existiert N^r, wenn

$$(\mathsf{v}_\mathsf{n}^1 + \mathsf{sig} \cdot \mathsf{a}^1) \cdot (\mathsf{v}_\mathsf{n}^e + \mathsf{sig} \cdot \mathsf{a}^e) < 0$$

gilt. Die Komponenten der Zustände N^ℓ und N^r bezeichnen wir wie folgt:

$$N^\ell =: \begin{pmatrix} \rho^\ell \\ \rho^\ell \mathsf{v}^\ell \\ \rho^\ell E^\ell \end{pmatrix}, \; N^r =: \begin{pmatrix} \rho^r \\ \rho^r \mathsf{v}^r \\ \rho^r E^r \end{pmatrix}.$$

Die Tangentialgeschwindigkeiten $(\tau_j \cdot \mathsf{v})$ sind auf beiden Kurven Riemannsche Invarianten und es folgt für $j \in \{1, \ldots, d-1\}$:

$$\begin{aligned} (\tau_j \cdot \mathsf{v}^\ell) &= (\tau_j \cdot \mathsf{v}^0), \\ (\tau_j \cdot \mathsf{v}^r) &= (\tau_j \cdot \mathsf{v}^1). \end{aligned}$$

Da bei N^ℓ und N^r gerade die zugehörigen Eigenwerte verschwinden, ergibt sich für die Normalengeschwindigkeit:

$$\mathsf{v}_\mathsf{n}^\ell = \mathsf{sig} \cdot \mathsf{a}^\ell, \tag{1.32}$$

$$\mathsf{v}_\mathsf{n}^r = -\mathsf{sig} \cdot \mathsf{a}^r. \tag{1.33}$$

Zusammen mit den Riemannschen Invarianten aus (1.28) und (1.29) lassen sich die Normalengeschwindigkeiten berechnen:

$$\begin{aligned} \mathsf{v}_\mathsf{n}^\ell &= \frac{1}{\kappa+1}((\kappa-1)\mathsf{v}_\mathsf{n}^0 - 2 \cdot \mathsf{sig} \cdot \mathsf{a}^0), \\ \mathsf{v}_\mathsf{n}^r &= \frac{1}{\kappa-3}((\kappa-1)\mathsf{v}_\mathsf{n}^1 + 2 \cdot \mathsf{sig} \cdot \mathsf{a}^1). \end{aligned}$$

Damit sind alle Geschwindigkeitskomponenten bekannt. Die Schallgeschwindigkeiten a^ℓ und a^r lassen sich jetzt mit (1.32) und (1.33) berechnen. Aus den Geschwindigkeiten und den Schallgeschwindigkeiten berechnet sich die Energie in der bekannten Weise:

$$\begin{aligned} E^\ell &= \frac{(\mathsf{a}^\ell)^2}{\kappa(\kappa-1)} + \tfrac{1}{2}(\mathsf{v}^\ell \cdot \mathsf{v}^\ell), \\ E^r &= \frac{(\mathsf{a}^r)^2}{\kappa(\kappa-1)} + \tfrac{1}{2}(\mathsf{v}^r \cdot \mathsf{v}^r). \end{aligned}$$

Auf beiden Kurven ist wieder die Entropie (ψ^1_{d+1} und ψ^{d+2}_{d+1}) invariant, und es folgt mit (1.25) und (1.26) die Formel für die Dichte:

$$\begin{aligned} \rho^\ell &= \left(\frac{(\mathsf{a}^\ell)^2}{\kappa \cdot Z^0}\right)^{1/(\kappa-1)}, \\ \rho^r &= \left(\frac{(\mathsf{a}^r)^2}{\kappa \cdot Z^1}\right)^{1/(\kappa-1)}. \end{aligned}$$

Damit sind alle Zustandkomponenten der eventuell existierenden Nullstellen N^ℓ und N^r berechenbar.

Randbedingungen

Liegt ein Quadraturpunkt q am Rande des Gebietes Ω, so ist nur der Innenwert v^- in der Notation von Seite 22 bekannt. Ein entsprechender Wert v^+ kann aber beispielsweise durch Randbedingungen der Form

$$u(x,t) \stackrel{!}{=} b(x,t) \quad \text{für } x \in B \subseteq \partial\Omega,\ t \in (0,T]$$

mit einer vorgegebenen Randfunktion b auf dem Randstück B ermittelt werden. Wir setzen dann, abhängig vom Zeitpunkt t:

$$v^+ := b(\mathsf{q}, t)$$

und berechnen wie im letzten Abschnitt den Wert der numerischen Flußfunktion $H(v^-, v^+, \mathsf{n})$. Mit dieser Form von Randbedingung wurden sämtliche zweidimensionalen Beispiele mit skalaren Flußfunktionen gerechnet. Bei den Euler-Gleichungen wurden auf diese Weise die Ein- und Ausströmränder modelliert, indem die jeweiligen Zustände aus den Anfangsbedingungen als konstante Außenzustände v^+ gesetzt wurden. Mit einer weiteren Randbehandlung wurde ermöglicht, die geradlinig gleichförmige Ausbreitung einer Unstetigkeit in Abhängigkeit von der Zeit und dem Ort als Außenzustand zu setzen; siehe hierzu das Beispiel für die Reflexion eines Stoßfront ab Seite 73. Für die eindimensionalen skalaren Probleme im Intervall $\Omega = [0,1]$ wurden ferner periodische Randbedingungen gewählt. Der Fluß in Richtung $\mathsf{n} = 1$ wurde an den Quadraturpunkten 0 und 1 mit der numerischen Flußfunktion $H(v(1), v(0), 1)$ berechnet.
Für die Modellierung fester Ränder bei den Euler-Gleichungen wurde eine weitere Randbedingung verwendet. Für eine feste Wand gilt hier die Bedingung, daß die Geschwindigkeit höchstens tangential zur Wand verläuft:

$\mathsf{v} \cdot \mathsf{n} = 0$. Hieraus folgert man für den Euler-Fluß:

$$\mathsf{n} \cdot F(u) = \sum_{i=1}^{s} \mathsf{n}_i \cdot \begin{pmatrix} \rho \mathsf{v}_i \\ \rho \mathsf{v}_i \cdot \mathsf{v} + \mathsf{p} \cdot e_i \\ (\rho E + \mathsf{p}) \mathsf{v}_i \end{pmatrix} = \begin{pmatrix} 0 \\ \mathsf{p} \cdot \mathsf{n} \\ 0 \end{pmatrix}.$$

Hierin wurde der Druck p mit dem Rekonstruktionswert v^- ermittelt. Sind ρ, ρv und ρE die Komponenten von v^-, dann wurde der Druck p durch die Formel (1.11) auf Seite 14 berechnet und anschließend die obige Formel zur Berechnung des Flusses $\mathsf{n} \cdot F$ verwendet.

Kapitel 2

Rekonstruktion der Zustandsvariablen

In diesem Kapitel sollen numerische Verfahren zur Rekonstruktion der Zustandsverteilung aus ihren Zellmitteldaten diskutiert werden, wie wir es bereits abstrakt im Abschnitt „Rekonstruktion“ auf Seite 21 skizziert haben. Wir konzentrieren uns hier auf den Fall, daß von einer gesuchten skalarwertigen Funktion $u \in U(\Omega \to \mathbf{R})$ aus einem normierten Funktionenraum nur die Zellmitteldaten $\delta_\omega u$ für alle Zellen $\omega \in \mathcal{G}$ bekannt seien. Im Falle vektorwertiger Zustandsverteilungen wollen wir annehmen, daß die skalaren Komponenten einzeln rekonstruiert werden. Wir suchen für jede Zelle $\omega \in \mathcal{G}$ eine Funktion v_ω, die eine möglichst gute Approximation mit hoher Fehlerordnung an die unbekannte Funktion u sei. Von einer rekonstruierten Funktion v_ω fordern wir, daß sie aus Konsistenzgründen mindestens den lokalen Zellmittelwert interpoliert:

$$\delta_\omega v_\omega \stackrel{!}{=} \delta_\omega u. \tag{2.1}$$

Wir werden in diesem Kapitel ausschließlich den Fall von Zellmitteldaten studieren. Andere Informationen, wie zum Beispiel Funktionswerte, Ableitungswerte oder höhere Momente, wie sie bei unstetigen Galerkin-Verfahren vorliegen, werden wir hier nicht in aller Allgemeinheit betrachten. Die beschriebenen Algorithmen lassen sich jedoch auch leicht auf diese Situationen übertragen, weil wir an den meisten Stellen nur die Stetigkeit und die Linearität der Funktionale δ_ω verwenden. Wir verwenden jedoch, daß der Träger (der Wirkungsbereich) der δ_ω nur lokal ist, und wir werden ausnutzen, daß die Zellmittelwerte jederzeit eine stabile Rekonstruktion erster Fehlerordnung ermöglichen, denn ein Zellmittelwert kann als konstante Funktion interpretiert werden (Lokalität und Positivität). Diese beiden Eigenschaften teilen sich die Zellmittelungsfunktionale mit den Dirac-Funktionalen $\delta_x u = u(x)$

(Auswertung an einer Stelle x). Daher sind die hier beschriebenen Resultate mit geringfügigen Änderungen auch für Funktionswerte $u(x)$ gültig.
Für ein numerisches Verfahren wird man die lokalen Rekonstruktionsfunktionen v_ω aus einem endlichdimensionalen Ansatzraum V_ω wählen. In diesem Kapitel werden wir hierzu den Raum der Polynome $V_\omega \stackrel{!}{=} \Pi^q$ bis zu einem vorgegebenen Höchstgrad q wählen. Dies ist der natürliche Ansatzraum, wenn man sicherstellen will, daß der punktweise Fehler der Rekonstruktion in Bereichen, in denen u hinreichend oft stetig differenzierbar ist, in Abhängigkeit vom Zelldurchmesser h mit der Ordnung $\mathcal{O}(h^{q+1})$ verschwindet. Weiterhin bieten sich Polynome besonders für numerische Verfahren an, weil sie leicht auszuwerten, zu differenzieren und zu integrieren sind. Gleichzeitig sind die Polynomräume invariant unter invertierbaren, linearen Abbildungen der Urbildmenge (diese Eigenschaft ist sogar charakterisierend für endlichdimensionale Polynomräume). Wir werden in diesem ersten Abschnitt einige Algorithmen für Polynome in zwei unabhängigen Variablen vorstellen (Auswertung, Integration auf Polygonen, Translation), soweit sie für den Einsatz in dem zweidimensionalen Finite-Volumen-Verfahren erforderlich sind.
Das Rekonstruktionsproblem einer Funktion v_ω ist zunächst eine lineare Aufgabenstellung. Wir werden neben dem lokalen Zellmittel in der Nachbarschaft der Zelle ω weitere Zellmittelwerte interpolieren wollen. Je nach Dimension des Ansatzraumes V_ω ergibt sich hieraus die Aufgabenstellung: Finde $v_\omega \in V$, so daß die lokale Interpolation (2.1) erfüllt wird und für zusätzlich zur Verfügung stehende Nachbarzellen $\omega_1, \ldots, \omega_m$ das Residuum

$$\|(\delta_{\omega_i} u - \delta_{\omega_i} v_\omega)_{i \in \{1,\ldots,m\}}\| \longrightarrow \min \tag{2.2}$$

möglichst klein wird. Hierin wird man eine dem Problem angepaßte Vektornorm des $\mathbf{R}^m$ verwenden. Typischerweise ist dies ein Skalarprodukt, und dann handelt es sich um eine quadratische Minimierungsaufgabe, die leicht numerisch zu lösen ist.
Neben der Rekonstruktionsaufgabe mit linearen Nebenbedingungen besteht das Problem, daß die im Rahmen des Finite-Volumen-Verfahrens zu rekonstruierenden Funktionen Unstetigkeiten senkrecht zu echten Untermannigfaltigkeiten des Gebietes haben können. Lineare Rekonstruktionsverfahren weisen hier das sogenannte Gibbs-Phänomen auf, welches zur Instabilität des Gesamtverfahrens führt. Daher werden wir Verfahren zur Steigungslimitierung der Rekonstruktion besprechen und hier speziell einen neuen Ansatz vorstellen, der in glatten Bereichen der Funktion u die hohe Fehlerordnung eines Rekonstruktionsverfahrens nicht zerstört. Dieser Ansatz wird sowohl mit klassischen Limitierungsverfahren als auch mit den WENO-Ansätzen verglichen; siehe [JS96], [LOC94] und [Fri97].

Insgesamt werden wir die Brauchbarkeit eines Rekonstruktionsalgorithmus im Rahmen eines Finite-Volumen-Verfahrens nach folgenden Kriterien beurteilen:

1. Das Verfahren soll unter der Annahme, daß die Funktion u selbst hinreichend oft differenzierbar sei, für feiner werdende Gitter mit möglichst hoher Ordnung konvergieren.

2. Das Verfahren soll in der Nähe von Unstetigkeiten keine starken Oszillationen aufweisen.

3. Die rekonstruierten Funktionen sollen möglichst scharf in der Nähe von Unstetigkeiten sein. Damit ist gemeint, daß die Rekonstruktionsfunktion möglichst dicht in der Umgebung einer Unstetigkeit ein gutes Konvergenzverhalten aufweisen sollte, wenn die Glätte von u dies zuläßt.

4. Das Verfahren sollte möglichst schnell sein und gleichzeitig nur wenig Speicher in Anspruch nehmen. Der Speicher- und Berechnungsaufwand der Rekonstruktion sollte nicht überproportional zur Anzahl der Zellen des Gitters steigen. Rekonstruktionen können somit immer nur aus einer beschränkten Anzahl lokaler Daten berechnet werden.

Für den Rekonstruktionsprozeß gibt es inzwischen sehr unterschiedliche Ansätze, von denen es zwei wesentliche Hauptrichtungen gibt. Bei der ersten Variante wird zuerst für jede Zelle eine lokale Rekonstruktionsfunktion mit möglichst hoher Fehlerordnung berechnet. Weiterhin muß eine stabile Rekonstruktion mindestens erster Fehlerordnung zur Verfügung stehen. Aus diesen beiden Rekonstruktionen wird dann eine Konvexkombination in Abhängigkeit von der „Glätte“ der Rekonstruktion hoher Ordnung berechnet. Für das Finite-Volumen-Verfahren erhält man eine stabile Rekonstruktion erster Fehlerordnung, wenn man den lokalen Zellmittelwert trivial als konstante Rekonstruktionsfunktion v^0 fortsetzt. Eine Rekonstruktion v^q hoher Ordnung kann als zentrale Interpolante umliegender Zellmitteldaten berechnet werden. Mit einem geeigneten Parameter $\theta \in [0, 1]$ kann man dann den Einfluß der jeweiligen Rekonstruktionen durch Konvexkombination steuern:

$$v(x) := (1 - \theta)v^0(x) + \theta v^q(x)$$

Oszilliert die Rekonstruktion hoher Fehlerordnung stark, so wird man den Parameter θ möglichst klein wählen, um den negativen Einfluß gering zu halten. Ist jedoch v^q hinreichend „glatt“, so wird man den Parameter θ möglichst dicht bei 1 wählen. Durch geeignete Wahl des Parameters kann

erreicht werden, daß in glatten Bereichen die Konvergenzordnung der Rekonstruktion hoher Fehlerordnung nicht zerstört wird.
Die zweite Variante, die sogenannte WENO-Methode[1], besteht darin, mehrere Rekonstruktionen hoher Ordnung zu bestimmen und eine Konvexkombinationen zwischen diesen zu berechnen. Hierdurch wird in jedem Fall in glatten Bereichen der zu rekonstruierenden Funktion die hohe Fehlerordnung erhalten. Die verschiedenen Funktionen werden durch einseitige Auswahl der zu interpolierenden Nachbardaten berechnet. Hierdurch soll erreicht werden, daß auch in der Nähe von Unstetigkeiten möglichst eine Funktion zur Verfügung steht, die wenig oszilliert. Diese soll dann bei der Konvexkombination entsprechend ihres geringen Oszillationsverhaltens stärker gewichtet werden als andere. In ersten ENO-Verfahren, siehe beispielsweise [HEOC87], wurde genau eine der berechneten Funktionen ausgewählt. In den Arbeiten [JS96] und [LOC94] wurde dann später die beschriebene Idee der Konvexkombination verfolgt, um den Konstruktionsprozeß stetig zu gestalten. Die ersten ENO-Verfahren für unstrukturierte Dreiecksgitter wurden in den Arbeiten [Abg94], [Son97a] und [Son97b] diskutiert. In [Fri97] wird erstmals ein gewichtetes ENO-Verfahren für unstrukturierte Gitter mit quadratischen Polynomen vorgestellt.
Beide der beschriebenen Strategien besitzen einen empfindlichen und keineswegs vollständig verstandenen Parameter: Den Indikator, welcher für die Beurteilung der Glätte einer Rekonstruktion notwendig ist. Aus dieser Bewertung muß schließlich eine geeignete Konvexkombination berechnet werden. Für das WENO-Verfahren kommt erschwerend hinzu, daß eine hinreichende Anzahl verschiedener einseitiger Rekonstruktionen zur Verfügung stehen muß. Für die in dieser Arbeit verwendeten zweidimensionalen Box-Gitter wurden die Steuerungsparameter für das WENO-Verfahren aus der Arbeit [Fri97] verwendet.
Neben den beiden beschriebenen Ansätzen für Rekonstruktionsverfahren höherer Ordnung sei hier noch auf einen interessanten neuen Ansatz in [Oll97] verwiesen, bei dem eine Rekonstruktionen höherer Ordnung aus einer Minimierungsaufgabe der Form (2.2) mit einer datenabhängigen Norm des Residuums berechnet wird.
Weiterhin gibt es in [HS99] ein Verfahren, welches für jeden Quadraturpunkt am Rande einer Zelle jeweils einen einzelnen Rekonstruktionswert bestimmt. Die Autoren zeigen, daß durch eine geeignete Gewichtung mehrerer Rekonstruktionsfunktionen niedriger Ordnung ein Punktwert mit höherer Fehlerordnung rekonstruiert werden kann. Der vorgestellte Algorithmus scheint

[1] weighted essentially non oscillatory reconstruction (gewichtete, wesentlich nicht oszillierende Rekonstruktion).

jedoch vergleichsweise kostenintensiv zu sein. Dieser interessante Ansatz wurde daher in der vorliegenden Arbeit nicht verfolgt. Wir gehen hier davon aus, daß das Ergebnis des Rekonstruktionsprozesses eine Rekonstruktionsfunktion für jede Zelle sein soll.
Alle geschilderten Rekonstruktionsverfahren setzen voraus, daß eine Rekonstruktion hoher Fehlerordnung für jede Zelle berechnet wird. Daher beschäftigen wir uns zuerst mit dieser linearen Teilaufgabe.

Rekonstruktion mit Polynomen

Konvergente Verfahren

Auf einem Finite-Volumen-Gitter $\mathcal{G}$ seien die Werte $\delta_\omega(u)$ der linearen Funktionale δ_ω einer (unbekannten) Funktion $u \in U(\Omega \to \mathbf{R})$ bekannt. Wir suchen ein Verfahren, welches nach Vorgabe einer Norm auf U eine möglichst gute Approximation an u liefert.
In einem Finite-Volumen-Verfahren werden die Rekonstruktionsfunktionen an den Zellrändern punktweise ausgewertet. Eine geeignete Norm für die Beurteilung einer punktweise auszuwertenden Rekonstruktion ist die Supremumsnorm auf kompakten Teilgebieten. Allerdings ist dieses Maß nur dann sinnvoll, wenn die zu rekonstruierende Funktion mindestens gleichmäßig stetig auf diesem Teilgebiet ist. Für die Konvergenzanalyse bezüglich der Supremumsnorm werden wir sogar allgemein voraussetzen, daß die zu rekonstruierende Funktion $(q+1)$-mal gleichmäßig stetig differenzierbar auf dem betrachteten Teilgebiet sei ($q \in \mathbf{N}_0$). Um die Darstellung etwas einfacher zu gestalten, werden wir annehmen, daß die geforderte Glätte für das ganze Gebiet Ω gelte. Sei also für die Untersuchungen dieses Kapitels $u \in U := C^{q+1}(\Omega \to \mathbf{R})$. Wir studieren nun das Konvergenzverhalten von Rekonstruktionsverfahren bezüglich $\|\cdot\|_{\infty,\Omega}$.
Das gesuchte Verfahren soll für jede Zelle eine Rekonstruktionsfunktion berechnen. Die Funktionen benachbarter Zellen müssen hierbei nicht stetig angrenzen. Dies führt dazu, daß eine Rekonstruktionsfunktion nur in den offenen Zellen selbst, nicht aber auf deren Rändern wohldefiniert ist. In Ausdrücken, in denen es nicht auf diese mehrdeutigen Randwerte ankommt, werden wir trotzdem die Rekonstruktion als eine globale Funktion über Ω auffassen. Ist beispielsweise v eine solche Rekonstruktion, dann schreiben wir für den maximalen Abstand zwischen u und v auf Ω vereinfachend den Ausdruck:

$$\|v-u\|_{\infty,\Omega} := \sup_{\omega\in\mathcal{G}} \|v_\omega - u\|_{\infty,\omega} := \sup_{\omega\in\mathcal{G}} \sup_{x\in\omega} |v_\omega(x) - u(x)|.$$

Die Qualität eines Rekonstruktionsverfahrens werden wir dadurch messen, wie schnell der obige Ausdruck in Abhängigkeit der Feinheit eines Gitters verschwindet: Sei hierzu der Durchmesser einer Zelle definiert als

$$h_\omega := \operatorname{diam}(\omega) := \sup_{x,y\in\omega} \|x-y\|_2,$$

und die Maschenweite eines Gitters $h_\mathcal{G}$ sei der maximale Zelldurchmesser:

$$h_\mathcal{G} := \max_{\omega\in\mathcal{G}} h_\omega.$$

Wir betrachten nun eine Folge von Gittern $(\mathcal{G}_h)_{h\to 0}$ mit abnehmender Maschenweite h. Wie allgemein üblich indizieren wir die Gitter mit der Maschenweite selbst. Zu jedem Gitter $\mathcal{G}_h$ definieren wir die Abbildung $\mathcal{D}_h$ als Zusammenstellung der endlich vielen Datenfunktionale δ_ω für $\omega \in \mathcal{G}_h$:

$$\begin{array}{rcll} \mathcal{D}_h : U & \longrightarrow & \mathbf{R}^{\mathcal{G}_h} & (\mathbf{R}^{\mathcal{G}_h} = \operatorname{Abb}(\mathcal{G}_h \to \mathbf{R})) \\ \mathcal{D}_h(u) & := & (\delta_\omega u)_{\omega\in\mathcal{G}_h} & \end{array}$$

Für das Gitter $\mathcal{G}_h$ nennen wir das zugehörige Rekonstruktionsverfahren $\mathcal{R}_h$. Wir sagen, $\mathcal{R}_h$ sei **mindestens** von der **Fehlerordnung** $q+1$, wenn es eine von h unabhängige Konstante $C \in \mathbf{R}$ gibt, so daß für alle $u \in C^{q+1}(\Omega \to \mathbf{R})$ und alle Gitter der Folge die folgende Abschätzung gilt:

$$\|\mathcal{R}_h\mathcal{D}_h u - u\|_{\infty,\Omega} \le C\|u^{(q+1)}\|_{\infty,\Omega} h^{q+1}.$$

Hierin sei $\|u^{(q+1)}\|_{\infty,\Omega}$ das globale Maximum aller partiellen Ableitungen von u vom Grad $q+1$.
Weiß man von einem Rekonstruktionsverfahren, daß es die Fehlerordnung $q+1$ für geeignete Gitterfolgen hat, dann kann man umgekehrt die Güte eines einzelnen Gitters $\mathcal{G}_h$ mit der Maschenweite h durch folgenden Ausdruck bestimmen:

$$\operatorname{cond}(\mathcal{G}_h) := \sup_{u\in C^{q+1}, u^{(q+1)}\neq 0} \frac{\|\mathcal{R}_h\mathcal{D}_h u - u\|_{\infty,\Omega}}{\|u^{(q+1)}\|_{\infty,\Omega} h^{q+1}}.$$

Insbesondere wird man ein einzelnes Gitter so gestalten wollen, daß das Supremum möglichst klein wird. Neben diesem globalen Maß wird man für eine einzelne Zelle $\omega \in \mathcal{G}_h$ die **lokale Gitterkondition**

$$\operatorname{cond}(\omega) := \sup_{u\in C^{q+1}, u^{(q+1)}\neq 0} \frac{\|\mathcal{R}_h\mathcal{D}_h u - u\|_{\infty,\omega}}{\|u^{(q+1)}\|_{\infty,\omega} h_\omega^{q+1}}$$

betrachten. Ist dieses Maß für alle Zellen des Gitters nach oben beschränkt, so kann damit die Größe $\mathrm{cond}(\mathcal{G}_h)$ abgeschätzt werden. Aus diesem Grunde wollen wir nur die lokale Gitterkondition eines Rekonstruktionsverfahrens untersuchen. Bei der Gestaltung des Gitters werden wir für ein festes Rekonstruktionsverfahren die Gitterkondition möglichst klein halten wollen. Umgekehrt werden wir jetzt das Rekonstruktionsverfahren für eine einzelne Zelle und ihre Umgebung innerhalb eines festen Gitters so konstruieren, daß das resultierende Gesamtverfahren mit möglichst hoher Fehlerordnung und möglichst kleiner lokaler Gitterkondition konvergiert. Wir können uns also mit dem Fall begnügen, daß wir ein Rekonstruktionsverfahren für eine fest gewählte Zelle ω beschreiben und Kriterien angeben, für die die lokale Gitterkondition für möglichst große lokale Fehlerordnungen $q+1 \in \mathbf{N}$ beschränkt bleibt, wenn wir den Zelldurchmesser h_ω und entsprechend die Umgebung von ω variieren.
Für die ausgewählte Zelle ω wählen wir jetzt einen endlichdimensionalen, lokalen Ansatzraum V, wobei wir die Abhängigkeit des Ansatzraumes von der Zelle ω nicht explizit in der Notation erwähnen. Da wir in der Umgebung von ω Zellmittelwerte interpolieren wollen, müssen die Funktionen des lokalen Ansatzraumes noch in dieser Umgebung wohldefiniert sein. Wir setzen vereinfachend voraus, daß die Funktionen aus V den vollständigen Raum $\mathbf{R}^d$ als Urbildbereich haben: $V \subseteq \mathrm{Abb}(\mathbf{R}^d \to \mathbf{R})$.
Wählt man als lokalen Ansatzraum V den Raum der Polynome $\Pi^q := \Pi^q(\mathbf{R}^d \to \mathbf{R})$ bis zum Höchstgrad $q \in \mathbf{N}_0$, so ist der nachfolgende Satz entscheidend für die Konstruktion von Rekonstruktionsverfahren mit der entsprechenden Fehlerordnung $(q+1)$:

Satz 2.1 *Sei $q \in \mathbf{N}_0$ fest gewählt. Dann gibt es eine Konstante $C_q \in \mathbf{R}$, so daß für alle $h > 0$ die folgende Aussage erfüllt wird: Für die h-Umgebung $K_h := \{x \in \mathbf{R}^d : \|x\|_2 \le h\}$ und eine stetige Projektion P_h von $C^{q+1}(K_h \to \mathbf{R})$ auf den Raum der Polynome Π^q bis zum Höchstgrad $q \in \mathbf{N}_0$*

$$\begin{aligned} P_h : C^{q+1}(K_h \to \mathbf{R}) &\longrightarrow \Pi^q, \\ \forall \wp \in \Pi^q : P_h \wp &= \wp \end{aligned}$$

gilt für alle $u \in C^{q+1}(K_h \to \mathbf{R})$ die Ungleichung:

$$\|P_h u - u\|_{\infty,K_h} \le C_q \cdot (1 + \|P_h\|_{\infty,K_h}) \cdot \|u^{(q+1)}\|_{\infty,K_h} \cdot h^{q+1}.$$

Beweis Die Taylor-Entwicklung von u bis zum Grad q um den Ursprung sei mit $Tu \in \Pi^q$ bezeichnet. Nach dem Taylorschen Satze existiert eine von

h und u unabhängige Konstante $C_q > 0$, so daß die folgende Abschätzung gilt:

$$\|Tu - u\|_{\infty,K_h} \leq C_q \cdot \|u^{(q+1)}\|_{\infty,K_h} \cdot h^{q+1}.$$

Zusammen mit der Dreiecksungleichung und der Projektionseigenschaft von P_h erhalten wir hieraus die gewünschte Abschätzung:

$$\begin{aligned}
\|P_h u - u\|_{\infty,K_h} &\leq \|P_h u - \underbrace{Tu}_{=P_h Tu}\|_{\infty,K_h} + \|Tu - u\|_{\infty,K_h} \\
&= \|P_h(u - Tu)\|_{\infty,K_h} + \|Tu - u\|_{\infty,K_h} \\
&\leq (1 + \|P_h\|_{\infty,K_h}) \cdot \|Tu - u\|_{\infty,K_h} \\
&= C_q \cdot (1 + \|P_h\|_{\infty,K_h}) \cdot \|u^{(q+1)}\|_{\infty,K_h} \cdot h^{q+1}.
\end{aligned}$$

■

Konstruiert man für eine Zelle ein Rekonstruktionsverfahren $\mathcal{R}_h$ für Polynome $V = \Pi^q$ aus den Daten $\mathcal{D}_h u$, so ist gerade eine Abbildung $P_h := \mathcal{R}_h \mathcal{D}_h$ auf den Raum der Polynome Π^q gegeben. **Reproduziert** diese Abbildung Polynome, das heißt $\wp = P_h \wp = \mathcal{R}_h \mathcal{D}_h \wp$ für alle $\wp \in \Pi^q$, dann kann man mit dem obigen Satz die Fehlerordnung $q+1$ des Rekonstruktionsverfahrens zeigen, wenn man sichergestellt hat, daß die Norm $\|P_h\|_{\infty,K_h}$ unabhängig von h beschränkt bleibt. Es ist natürlich notwendig, daß die Zelle ω in K_h liegt: Hierfür muß man aber nur den Ursprung des $\mathbf{R}^d$ in der Nähe der Zelle wählen, oder man muß die Zelle entsprechend in den Ursprung verschieben.

Lokale Ausgleichsprobleme

Wir wollen nun für eine Zelle ω eine Rekonstruktionsfunktion aus Π^q berechnen. Hierzu nehmen wir an, daß ω in einem Kreis K_H mit dem Radius H um den Ursprung liege. Der Radius $H = \text{const} \cdot h$ sei ein festes Vielfaches des Zelldurchmessers h von ω. Weiterhin seien aus der Umgebung dieser Zelle innerhalb von K_H weitere Zellen des Gitters zu einer Menge $\mathcal{L}$ zusammengestellt. $\mathcal{L}$ enthalte die Zelle ω selbst nicht. Für die Komposition von $\mathcal{L}$ mit der Zelle ω verwenden wir das Symbol $\overline{\mathcal{L}} := \mathcal{L} \cup \{\omega\}$. Solche Zellmengen, die nur in einem beschränkten Kreis um eine Zelle liegen, nennen wir **lokale Gitter**. Die Zellmittelungsfunktionale zu den Zellen in $\overline{\mathcal{L}}$ stellen wir wieder zu einer linearen Abbildung zusammen:

$$\begin{aligned}
\mathcal{D}_{\overline{\mathcal{L}}} : C^{q+1}(K_H \to \mathbf{R}) &\longrightarrow \mathbf{R}^{\overline{\mathcal{L}}} \\
\mathcal{D}_{\overline{\mathcal{L}}}(u) &:= (\delta_\eta u)_{\eta \in \overline{\mathcal{L}}}.
\end{aligned} \tag{2.3}$$

Entsprechend sei $\mathcal{D}_{\mathcal{L}}$ definiert. Für diese Zusammenstellungen von Zellmittelungsfunktionalen kann man leicht die Operatornorm angeben, wenn wir sowohl im Bild wie im Urbildbereich die Maximumsnorm verwenden:

$$\|D_{\mathcal{L}}\|_\infty = \|D_{\overline{\mathcal{L}}}\|_\infty = 1.$$

Also sind die Datenfunktionale unabhängig von der Feinheit des Gitters beschränkt. Um im Sinne des Satzes 2.1 ein Rekonstruktionsverfahren

$$\mathcal{R}_{\overline{\mathcal{L}}} : \mathbf{R}^{\overline{\mathcal{L}}} \quad \longrightarrow \quad \Pi^q$$

der Fehlerordnung $q+1$ anzugeben, brauchen wir nur sicherstellen, daß es einerseits Polynome reproduziert und unabhängig von H eine beschränkte Norm hat, denn dann erfüllt die Komposition $\mathcal{R}_{\overline{\mathcal{L}}}\mathcal{D}_{\overline{\mathcal{L}}}$ die Voraussetzungen des Satzes.
Zur Konstruktion solcher Verfahren $\mathcal{R}_{\overline{\mathcal{L}}}$ bestimmen wir für gegebene Daten

$$d := \mathcal{D}_{\overline{\mathcal{L}}} u$$

ein interpolierendes oder approximierendes Polynom $\wp \in \Pi^q$, das möglichst $\mathcal{D}_{\overline{\mathcal{L}}}\wp = d$ erfüllt. Damit das Verfahren für jedes $\wp \in \Pi^q$ die Bedingung $\wp = \mathcal{R}_{\overline{\mathcal{L}}}\mathcal{D}_{\overline{\mathcal{L}}}\wp$ erfüllen kann, muß $\mathcal{R}_{\overline{\mathcal{L}}}$ notwendigerweise surjektiv sein. Dies erfordert, daß

$$Q := \dim \Pi^q \leq \operatorname{card} \overline{\mathcal{L}}$$

erfüllt wird. Das lokale Gitter $\overline{\mathcal{L}}$ muß also mindestens Q Zellen enthalten.
Wir wollen $\mathcal{R}_{\overline{\mathcal{L}}}d$ als Lösung eines linearen Ausgleichsproblems formulieren. Gesucht ist das Polynom $\wp \in \Pi^q$, welches das lokale Zellmittel d_ω interpoliert und mit positiven Gewichten g_η für alle $\eta \in \mathcal{L}$ die folgende Quadratsumme minimiert:

$$\sum_{\eta \in \overline{\mathcal{L}}} g_\eta^2 (d_\eta - \delta_\eta(\wp))^2 \quad \longrightarrow \quad \min, \tag{2.4}$$

$$\delta_\omega(\wp) \quad = \quad d_\omega. \tag{2.5}$$

Diese Aufgabe ist stets lösbar. Wir setzen von dem lokalen Gitter $\overline{\mathcal{L}}$ voraus, daß das System auch eindeutig lösbar ist — später werden wir ein numerisches Kriterium angeben.
Die lokale Interpolationsbedingung (2.5) kann in einem numerischen Verfahren in der folgenden Weise sichergestellt werden: Sei $\pi_0, \ldots, \pi_{Q-1}$ eine Basis von Π^q, wobei die erste Basisfunktion die konstante Funktion $1 =: \pi_0$ sei. Dann orthogonalisieren wir diese Basis bezüglich des Dualitätspaares (π_0, δ_ω):

$$\tilde{\pi}_i := \pi_i - \frac{\delta_\omega(\pi_i)}{\delta_\omega(\pi_0)}\pi_0 \qquad \forall i \in \{1, \ldots, Q-1\}.$$

Das rekonstruierende Polynom $\wp := \mathcal{R}_{\overline{\mathcal{L}}} d$ kann in der Basis $\pi_0, \tilde{\pi}_1, \ldots, \tilde{\pi}_{Q-1}$ mit den Koeffizienten $p_1, \ldots, p_{Q-1}$ angegeben werden:

$$\wp = d_\omega \pi_0 + \sum_{i=1}^{i<Q} p_i \tilde{\pi}_i.$$

Mit den Koeffizienten

$$\tilde{d}_\eta := d_\eta - d_\omega \frac{\delta_\eta(\pi_0)}{\delta_\omega(\pi_0)}$$

lautet das Ausgleichsproblem in diesen Koeffizienten:

$$\sum_{\eta \in \overline{\mathcal{L}}} g_\eta^2 \left(\tilde{d}_\eta - \sum_{i=1}^{i<Q} p_i \delta_\eta(\tilde{\pi}_i) \right)^2 \quad \longrightarrow \quad \min. \tag{2.6}$$

Bezüglich der ursprünglichen Basis $\pi_0, \ldots, \pi_{Q-1}$ besitzt die Lösung die folgende Darstellung:

$$\wp = \left(d_\omega - \sum_{i=1}^{i<Q} \frac{\delta_\omega(\pi_i)}{\delta_\omega(\pi_0)} p_i \right) \pi_0 + \sum_{i=1}^{i<Q} p_i \pi_i.$$

Nach diesen Transformationen können wir uns auf das Ausgleichsproblem aus (2.6) konzentrieren: Wir führen hierzu die Diagonalmatrix G der Gewichte und die Massenmatrix A ein, wobei wir der Einfachheit halber eine Indizierung durch die jeweiligen Zellen vornehmen:

$$G_{\eta,\eta} := g_\eta, \quad \eta \in \mathcal{L}, \tag{2.7}$$

$$A_{\eta,i} := \delta_\eta(\tilde{\pi}_i), \quad \eta \in \mathcal{L}, \quad i \in \{1, \ldots, Q-1\}. \tag{2.8}$$

Die Lösung des Ausgleichsproblems ergibt sich aus der Normalengleichung

$$A^t G^t G A p = A^t G^t \tilde{d}.$$

Um Konditionsprobleme zu vermeiden, bietet es sich an, dieses System über Householder-Transformationen der Matrix GA zu lösen. Verwendet man ein anderes Verfahren, dann sollte man unbedingt noch eine Skalierung des räumlichen Koordinatensystems mit einem lokalen Zelldurchmesser vornehmen.
Ist die Matrix $A^t G^t G A$ invertierbar, so ist das Ausgleichsproblem (2.4) eindeutig lösbar und dann ist der Rekonstruktionsalgorithmus $\mathcal{R}_{\overline{\mathcal{L}}}$ wohldefiniert. Wir werden also von dem lokalen Gitter $\overline{\mathcal{L}}$ bei der Konstruktion berücksichtigen müssen, daß die zugehörige Matrix $A^t G^t G A$ invertierbar ist. Dies ist der Fall, wenn GA maximalen Rang hat. Wir wollen zudem erreichen,

daß die Operatornorm des resultierenden Rekonstruktionsverfahrens $\mathcal{R}_{\overline{\mathcal{L}}}$ unter geeigneten Gitterfolgen $(\mathcal{G}_h)_{h\to 0}$ beschränkt bleibt. Hierzu müssen wir uns festlegen, für welche Gitterfolgen wir das Verfahren konvergent gestalten wollen. Aus praktischen Erwägungen wird man sicherstellen wollen, daß das Verfahren mindestens dann konvergiert, wenn die Zellen isotrop verkleinert werden. Damit ist gemeint, daß die Zellen eines feineren Gitters den Zellen der gröberen Gitter ähnlich sind. Dies ist sichergestellt, wenn das Verfahren unter affinen Abbildungen $\mathcal{A} : \mathbf{R}^d \to \mathbf{R}^d$, die winkeltreu sind (Kreise werden auf Kreise abgebildet) eine beschränkte Operatornorm $\|\mathcal{R}_{\overline{\mathcal{L}}}\|_{\infty,K_H}$ hat. Wir untersuchen also das Verhalten von $\mathcal{R}_{\overline{\mathcal{L}}}$ unter Rotationen, Spiegelungen, Translationen und insbesondere unter Skalierungen der Zellen in $\overline{\mathcal{L}}$ und des Kreises K_H. Wir weisen darauf hin, daß invertierbare, affine Abbildungen $\mathcal{A}$ im $\mathbf{R}^d$ lineare Abbildungen des Polynomraums induzieren und daß die Maximumsnorm eines Polynoms $\wp \in \Pi^q$ auf dem Kreis K_H identisch mit der Maximumsnorm von $\wp \circ \mathcal{A}^{-1}$ auf dem Bildkreis $\mathcal{A}K_H$ ist.
Wir betrachten nun das Ausgleichsproblem (2.4) unter der winkeltreuen Abbildung $\mathcal{A}$: Ist zu einem festen Datensatz $d \neq 0$ das Polynom $\wp$ die Lösung $\wp = \mathcal{R}_{\overline{\mathcal{L}}} d$, dann ist das Polynom $\wp \circ \mathcal{A}^{-1}$ die Lösung für die Menge der abgebildeten Zellen $\mathcal{A}\overline{\mathcal{L}}$ mit dem gleichen Datensatz d, sofern die gleichen Gewichte g_η gewählt werden. Es folgt also

$$\frac{\|\mathcal{R}_{\overline{\mathcal{L}}} d\|_{\infty,K_H}}{\|d\|_\infty} = \frac{\|\mathcal{R}_{\mathcal{A}\overline{\mathcal{L}}} d\|_{\infty,\mathcal{A}K_H}}{\|d\|_\infty},$$

und damit ist gezeigt, daß die Operatornorm von $\mathcal{R}_{\overline{\mathcal{L}}}$ invariant unter winkeltreuen Abbildungen ist, sofern die Gewichte unter diesen Abbildungen nicht verändert werden.
In den Beispielen dieser Arbeit wurden die folgenden Gewichte verwendet:

$$g_\eta := \frac{h_\omega^d}{\|b_\eta - b_\omega\|_2^d}. \tag{2.9}$$

Hierin sind die b_η die Schwerpunkte der Zellen. Wie man leicht sieht, ist das obige Verhältnis invariant unter winkeltreuen Abbildungen. Durch den Exponenten d soll sichergestellt werden, daß Fehlerterme weit entfernter Zellen schwächer gewichtet werden als nahe gelegener Zellen.[2]
Sofern also die lokalen Gitter $\overline{\mathcal{L}}$ winkeltreu für $h \to 0$ abgebildet werden und die zugehörige Matrix $A^{\mathrm{t}} G^{\mathrm{t}} G A$ invertierbar ist, konvergiert das Verfahren. In dem implementierten Programm wird für jedes lokale Gitter überprüft, ob die Kondition der Matrix unterhalb einer gewissen Schranke liegt. Die

[2]Bei dieser Wahl nimmt das Gesamtgewicht der Zellen in einer Kugelschale mit fester Dicke moderat mit dem Radius der Kugelschale ab.

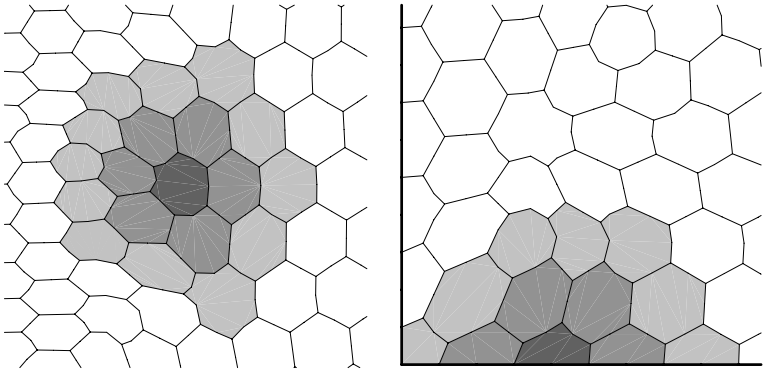


Abbildung 2.1: Beispiele für lokale Gitter, die zentral um eine Zelle angeordnet sind. Die Zelle ist dunkel hervorgehoben, die anderen Zellen des lokalen Gitters sind grau schattiert. Das linke Bild zeigt ein lokales Gitter im Inneren des Gebietes, und rechts ist eines für eine Randzelle dargestellt.

natürliche Norm für die Konditionsbestimmung ergibt sich aus der Supremumsnorm des Polynomraumes für die Kreisscheibe K_H. Diese Norm hängt jedoch von der Skalierung H ab, und daher bietet es sich an, die Koordinaten des $\mathbf{R}^d$ vor der Rekonstruktion mit $1/H$ zu skalieren und das Rekonstruktionsproblem für die Kreisscheibe K_1 zu lösen. Dann ist die Maximumsnorm des Polynomraumes unabhängig von H und man kann für die Konditionsmessung der Matrix $A^{\mathsf{t}}G^{\mathsf{t}}GA$ eine einfach berechenbare Matrixnorm nehmen. In dieser Arbeit wurde das Verhältnis des maximalen zum minimalen Betrag der Diagonalwerte der rechten oberen Dreiecksmatrix der QR-Zerlegung von GA verwendet. War dieser Wert größer als eine vorgegebene Schranke, so wurde das lokale Gitter $\overline{\mathcal{L}}$ nicht für Rekonstruktionszwecke verwendet.

Konstruktion lokaler Gitter

Wir beschreiben jetzt einen Algorithmus, mit dem für eine Gitterzelle ω ein lokales Gitter $\mathcal{L}$ konstruiert werden kann, das möglichst aus allen Raumrichtungen gleichviele Zellen verwendet. Wir nennen ein solches lokales Gitter **zentral**. Für Zellen am Rande des Gebietes läßt sich die Forderung natürlich weniger gut erfüllen als im Inneren — hier werden die zentralen lokalen Gitter einseitig im Inneren des Gebietes liegen (halbkreisförmig).
Der Algorithmus zum Einsammeln von Zellen für ein lokales Gitter $\overline{\mathcal{L}}$ einer Zelle $\omega \in \mathcal{G}$ startet zunächst mit der einelementigen Menge $\overline{\mathcal{L}} := \{\omega\}$ und fügt anschließend in einer Schleife jeweils die direkten Nachbarzellen der in $\overline{\mathcal{L}}$ enthaltenden Zellen hinzu. Auf diese Weise werden neue Zellen schichtweise hinzugefügt, wie es die Abbildung 2.1 illustriert. Dieser Algorithmus

wird zunächst solange durchgeführt, bis mindestens Q Zellen in $\overline{\mathcal{L}}$ enthalten sind. In den praktischen Beispielen zu dieser Arbeit waren die so entstandenen lokalen Gitter stets gut genug für die geforderte Polynomordnung q. Sicherheitshalber wurde das Verfahren so gestaltet, daß solange Schichten hinzugefügt werden, bis die gemessene lokale Gitterkondition unter einer vorgegebenen Schranke lag (siehe die Diskussion des letzten Abschnittes).
Für die Konstruktion einseitiger lokaler Gitter für ein gewichtetes ENO-Verfahren auf unstrukturierten Gittern sei auf die Arbeit [Fri97, Seite 200] verwiesen. Für Berechnungen mit diesem Verfahren wurden hier genau die dort beschriebenen lokalen Gitter verwendet.

Algorithmensammlung für bivariate Polynome

Für Polynome allgemeinen Grades in zwei Veränderlichen geben wir nun Algorithmen zur Auswertung und Integration auf allgemeinen polygonalen Zellen an. Weiterhin beschreiben wir ein Verfahren, mit dem der Ursprung einer lokalen Basis verschoben werden kann. Für Polynome in einer Veränderlichen können diese Algorithmen leicht hergeleitet werden — für das Horner-Schema siehe beispielsweise [Sto94, Seite 314].

Anordnung und Auswertung

Die Anordnung der reellwertigen Monome $x^i y^j$ vom Höchstgrad $q \geq i + j$ wurde so gewählt, daß Monome gleichen Grades jeweils in einzelnen Gruppen zusammenstehen:

$$1, \quad x,\ y, \quad x^2,\ xy,\ y^2, \quad x^3,\ x^2y,\ xy^2,\ y^3, \quad x^4, \ldots$$

Die Nummer des Monoms $x^i y^j$ innerhalb der bei Null beginnenden Anordnung ist

$$\#(i,\, j) := ((i + j) \cdot (i + j + 1))/2 + j.$$

Die Anzahl an Monomen und die Dimension des Polynomraumes ist folglich

$$Q := ((q + 1) \cdot (q + 2))/2.$$

Die Divisionen der letzten beiden Formeln sind bei der angegebenen Klammerung ganzzahlig.
Bei der Auswertung der Polynome treten zwei Anwendungen auf, die aus Effizienzgründen getrennt implementiert werden sollten: Erstens die gleichzeitige Auswertung aller Monome an einer Stelle für Interpolationsaufgaben aus Punktwerten und zweitens das Auswerten eines gegebenen Polynoms an einer Stelle.

Den Algorithmus zur gleichzeitigen Auswertung aller Monome bis zum Höchstgrad q an einer gegebenen Stelle $(x, y)^{\mathsf{t}}$ erhält man durch rekursive Auswertung der folgenden Ausdrücke:

$$\begin{aligned} x^0y^0 &:= 1, \\ x^{i+1}y^0 &:= x \cdot (x^i y^0), \\ x^i y^{j+1} &:= y \cdot (x^i y^j). \end{aligned}$$

Hierbei können die Multiplikationen mit 1 für die linearen Anteile bei geschickter Programmierung entfallen. Die Ergebnisse werden in einem Feld $r = (r_i)_{i \in \{0,\ldots,Q-1\}}$ der Länge Q berechnet. Es ergibt sich der folgende Algorithmus:

$r_0 = 1;$
falls $q > 0$
    $r_1 = x;\ r_2 = y;$
    $n = 3;\ m = 1;$
    für $k = 1, \ldots, q-1$
        $r_n = r_m \cdot x;$
        $n = n + 1;$
        für $j = 0, \ldots, k$
            $r_n = r_m \cdot y;$
            $n = n + 1;\ m = m + 1;$

Ist der Polynomgrad zur Übersetzungszeit des Programmes bekannt, so kann die Abfrage und die Schleife vom Compiler eliminiert werden.
Die Auswertung eines Polynoms mit den Koeffizienten a an der Stelle $(x, y)^{\mathsf{t}}$ könnte als Skalarmultiplikation von a mit dem oben berechneten Vektor r bestimmt werden. Dies ist jedoch aus Effizienzgründen nicht zu empfehlen, da hierdurch die Gesamtzahl an nötigen Operationen ungefähr verdoppelt wird. Daher wurde eine spezielle Implementierung, die auf das Horner-Schema für eindimensionale Polynome zurückgreift, programmiert: Die Hauptrekursion ergibt sich aus den Ausdrücken

$$\begin{aligned} s_q &:= a_{Q-1}, \\ s_{j-1} &:= t_{j-1} + y \cdot s_j. \end{aligned}$$

Hierbei ist a der Koeffizienten-Vektor des Polynoms in der angegebenen Numerierung. Der Wert des Polynoms findet sich am Ende der Rekursion in s_0. Die temporären Zwischengrößen t_j sind die Werte der Polynome über x, die sich aus den Koeffizienten derjenigen Monome ergeben, deren y-Exponent

gerade j ist:

$$t_j \;:=\; \sum_{i=0}^{q-j} a_{\#(i,j)} x^i$$

Für t_j wendet man das bekannte Horner-Schema an. Es ergibt sich das folgende Programm:

$p = Q - 1;$
$s = a_p;$
für $j = q, \ldots, 1$ $\{j \geq 1$ absteigend$\}$
    $s = s \cdot y;$
    $p = p - 1;\ k = p;$
    $t = a_k;$
    für $i = q, \ldots, j$ $\{i \geq j$ absteigend$\}$
        $k = k - i;$
        $t = t \cdot x + a_k;$
    $s = s + t;$

Nach dieser Schleife enthält s den Wert des Polynoms an der Stelle $(x, y)^{\mathrm{t}}$.

Integration auf polygonalen Zellen

Wir beschreiben jetzt ein Verfahren, mit dem sich Polynome vom Höchstgrad q auf einer polygonal berandeten, einfach zusammenhängenden Zelle ω integrieren lassen. Der Rand bestehe aus N Eckpunkten $P_0, \ldots, P_{N-1}$, $P_N := P_0$. Die Eckpunkte seien so numeriert, daß sie im mathematisch positiven Drehsinn angeordnet sind (das Zellinnere befindet sich links).
Der Algorithmus kann in zwei Bestandteile zerlegt werden: Im ersten Schritt werden in einem Datenfeld r der Länge Q die Integrale aller Monome für die Zelle berechnet. Anschließend erhält man den Wert eines Integrales für ein Polynom mit dem Koeffizienten-Vektor a als Skalarprodukt mit dem Vektor r. Diese Zerlegung des Algorithmus ermöglicht es dann auch, auf einfache Weise Interpolationsmatrizen für Zellmitteldaten aufzustellen. Im folgenden beschreiben wir die Berechnung des Vektors r.
Für das angestrebte Ziel gibt es mehrere Lösungsmöglichkeiten. Der hier vorgestellte Algorithmus zeichnet sich dadurch aus, daß er selbst für dreieckige Zellen nicht mehr Operationen benötigt als explizite Formeln. Auch für die einfachen Aufgaben, wie Flächenberechnung (Integral des ersten Monoms) und Schwerpunktbestimmung (Zellmittel des zweiten und dritten Monoms) kann der Algorithmus effizient verwendet werden. Die auftretenden Divisionen sind ausschließlich ganzzahlig. In dieser Hinsicht ist das Verfahren

numerisch besonders gutartig: Der Fall singulärer Randkanten stellt kein numerisches Problem dar. Die Entwicklung ist allerdings nicht symmetrisch in den Variablen x und y. Dieser Nachteil kann dadurch behoben werden, daß das Verfahren zweimal mit vertauschten Rollen von x und y aufgerufen wird und die Ergebnisse anschließend gemittelt werden: Gestaltet man das Verfahren von Anfang an symmetrisch, so wird man ohnehin die doppelte Operationszahl benötigen. In den praktischen Anwendungen des Autors wurde diese Symmetrisierung aus Kostengründen nie verwendet. Wir stellen im folgenden das schnellere, asymmetrische Verfahren vor.
In einer Schleife sollen die Werte der Integrale

$$r_{\#(i,j)} \quad = \quad \int\limits_{\omega} x^i y^j \ \mathsf{d}(x,y)$$

für alle $i+j \leq q$ bestimmt werden. Durch Anwenden des Gaußschen Integralsatzes kann man diesen Ausdruck in ein Randintegral mit Normalenvektor n überführen:

$$\begin{aligned} r_{\#(i,j)} &= \frac{1}{i+1} \int\limits_{\partial\omega} \begin{pmatrix} x^{i+1} y^j \\ 0 \end{pmatrix} \cdot \mathsf{n} \ \ \mathsf{d}\sigma \\ &= \frac{1}{i+1} \sum_{k=0}^{N-1} \frac{v_{k,y}}{\|v_k\|_2} \int\limits_{P_k}^{P_k+v_k} x^{i+1} y^j \ \ \mathsf{d}\sigma \\ &= \frac{1}{i+1} \sum_{k=0}^{N-1} v_{k,y} \int\limits_0^1 (P_{k,x} + v_{k,x}\lambda)^{i+1} (P_{k,y} + v_{k,y}\lambda)^j \ \ \mathsf{d}\lambda \quad (2.10) \end{aligned}$$

Hierin ist $v_k := P_{k+1} - P_k$. Der Normalenvektor n berechnet sich auf jeder Kante $\overline{P_k P_{k+1}}$ durch Rotation und Normierung von v_k. Die Normierung kürzt sich schließlich bei der Umparametrisierung auf $\lambda \in [0,1]$ weg.
Das Integral in (2.10) kann man auf zwei Arten vereinfachen. Entweder wendet man partielle Integration an und erhöht schrittweise einen der beiden Exponenten und verkleinert den anderen oder man multipliziert mit dem binomischen Lehrsatz die Summen aus. Der erste Weg führt in eine Sackgasse, da entweder durch Potenzen von $v_{k,x}$ oder von $v_{k,y}$ geteilt werden muß. Dies ist numerisch unerwünscht, und daher verfolgen wir den zweiten Weg:

$$\begin{aligned} & v_{k,y} \int\limits_0^1 (P_{k,x} + v_{k,x}\lambda)^{i+1} (P_{k,y} + v_{k,y}\lambda)^j \ \ \mathsf{d}\lambda \\ &= \ v_{k,y} \int\limits_0^1 \sum_{a=0}^{i+1} \begin{pmatrix} i+1 \\ a \end{pmatrix} P_{k,x}^{i+1-a} (v_{k,x}\lambda)^a \sum_{b=0}^{j} \begin{pmatrix} j \\ b \end{pmatrix} P_{k,y}^{j-b} (v_{k,y}\lambda)^b \ \ \mathsf{d}\lambda \end{aligned}$$

$$
\begin{aligned}
&= \sum_{a=0}^{i+1} \left(\begin{array}{c} i+1 \\ a \end{array} \right) P_{k,x}^{i+1-a} v_{k,x}^{a} \sum_{b=0}^{j} \left(\begin{array}{c} j \\ b \end{array} \right) P_{k,y}^{j-b} v_{k,y}^{b+1} \int_0^1 \lambda^{a+b} \; \mathsf{d}\lambda \\
&= \sum_{a=0}^{i+1} \underbrace{\left(\begin{array}{c} i+1 \\ a \end{array} \right) P_{k,x}^{i+1-a} v_{k,x}^{a}}_{=:X_{i,a}^{k}} \sum_{b=0}^{j} \underbrace{\left(\begin{array}{c} j \\ b \end{array} \right) P_{k,y}^{j-b} v_{k,y}^{b+1}}_{=:Z_{j,b}^{k}} \frac{1}{a+b+1} \\
&= \sum_{a=0}^{i+1} X_{i,a}^{k} \underbrace{\sum_{b=0}^{j} \frac{1}{a+b+1} Z_{j,b}^{k}}_{=:Y_{j,a}^{k}} \\
&= \sum_{a=0}^{i+1} X_{i,a}^{k} Y_{j,a}^{k} \qquad (2.11)
\end{aligned}
$$

Mit dieser Herleitung kann der Algorithmus in Worten angegeben werden: Zu Beginn werden alle Binomialkoeffizienten bis zum Grad $q+1$ und die ganzzahligen Quotienten $1/i$ für $i = 2, \ldots, q+2$ berechnet. Das Feld r wird mit Null initialisiert. In einer Schleife über die Kanten $\overline{P_k P_{k+1}}$ berechnet man die folgenden Schritte: Zuerst werden die Potenzen von $P_{k,y}^{j}$ und $v_{k,y}^{j+1}$ in zwei Feldern für $j = 0, \ldots, q$ abgespeichert. Anschließend berechnet man die Koeffizienten $Z_{j,b}^{k}$ für $j = 0, \ldots, q$ und $b = 0, \ldots, j$ in einem Feld der Länge Q. Hieraus und aus den vorbereiteten ganzahligen Divisionen berechnet man die Koeffizienten $Y_{j,a}^{k}$, $a = 0, \ldots, q+1$ und $j = 0, \ldots, q-a$, in einem weiteren Feld. Dann bestimmt man die Potenzen von $P_{k,x}^{a}$ und $v_{k,x}^{a}$ für $a = 0, \ldots, q$. Hieraus errechnet man in einem weiteren Feld die Werte $X_{i,a}^{k}$ mit $i = 0, \ldots, q$ und $a = 0, \ldots, i+1$. Schließlich bestimmt man nach Formel (2.11) die Summanden aller $r_{\#(i,j)}$ für die Kante $\overline{P_k P_{k+1}}$. Sind alle Kanten bearbeitet, so werden die $r_{\#(i,j)}$ noch durch $i+1$ mit den vorberechneten Quotienten geteilt. Dies ergibt die Flächenintegrale aller Monome bis zum Höchstgrad q im Feld r.

Bei der Implementierung des Algorithmus' sollte man unbedingt die Fälle gesondert bearbeiten, in denen Binomialkoeffizienten den Wert 1 oder Exponenten den Wert 0 annehmen. Dies erspart für kleine Polynomgrade sehr viele überflüssige Multiplikationen, und das Verfahren benötigt schließlich die gleiche Anzahl an Multiplikationen und Additionen, die man auch bei expliziter Auflösung der Formel für einen bestimmten Polynomgrad bekäme.[3]

[3] Die Anzahl der Multiplikationen des vorgestellten Verfahrens liegt bei ca. $q^3/3$ für große Polynomgrade q.

Verschieben des Ursprunges

Im Polynomraum sind affine, bijektive Selbstabbildungen der Urbildmenge lineare Abbildungen der Polynome. Durch diese Abbildungen ändern sich die Polynomgrade nicht. Daher haben die zugehörigen Abbildungsmatrizen im Polynomraum bei der angegebenen gradweisen Sortierung der Basisfunktionen dreiecksähnliche Gestalt (keine Dreiecksmatrix für Polynome über $\mathbf{R}^2$). In dem Finite-Volumen-Verfahren liegen für jede Zelle die Integrale aller Monome vor, die allerdings bezüglich einer jeweils lokalen Basis berechnet wurden. Diese lokale Basis einer Zelle ist lediglich eine Translation (keine Skalierung und keine Rotation) der globalen xy-Basis. Da bei der Rekonstruktion die Integrale der Monome für alle jeweils betroffenen Zellen in einer gemeinsamen Basis für das Aufstellen der Interpolationsmatrix benötigt werden, müssen die bekannten Integralwerte in eine gemeinsame Basis verschoben werden. Hierzu benötigt man ein Werkzeug, welches bei bekannten Werten von

$$\lambda^0_{i,j} = \int_\omega x^i y^j \ \mathsf{d}(x,y) \tag{2.12}$$

die Werte

$$\lambda^v_{i,j} = \int_\omega (x+v_x)^i (y+v_y)^j \ \mathsf{d}(x,y) \tag{2.13}$$

bezüglich der um $v \in \mathbf{R}^2$ verschobenen Monome berechnet. Dies ist eine allgemeine Formulierung für die in [Fri97] abgehandelte Aufgabe der Verschiebung der Integralwerte eines zweidimensionalen Polynoms zweiten Grades.
Mit Gewinn nicht nur für Finite-Volumen-Verfahren, sondern auch für Finite-Elemente-Verfahren hoher Ordnung läßt sich die Aufgabe weiter abstrahieren: Gegeben seien alle Werte

$$\lambda^0_{i,j} = \lambda(x^i y^j) \tag{2.14}$$

der Monome bis zum Höchstgrad q eines linearen Funktionals λ (z.B. die Zellintegration). Gesucht ist ein Verfahren, welches die Werte der um $v \in \mathbf{R}^2$ verschobenen Werte

$$\lambda^v_{i,j} = \lambda((x+v_x)^i (y+v_y)^j) \tag{2.15}$$

berechnet.[4]
Nun wendet man den binomischen Lehrsatz an, nutzt die Linearität von λ, und erhält:

$$\lambda((x+v_x)^i (y+v_y)^j) \ = \ \sum_{a=0}^{i} \left(\begin{array}{c} i \\ a \end{array} \right) v_x^{a-i} \sum_{b=0}^{j} \left(\begin{array}{c} j \\ b \end{array} \right) v_y^{b-j} \lambda(x^i y^j)$$

[4]Die Inverse dieser Abbildung erhält man durch Einsetzen von $-v$.

$$= \sum_{a=0}^{i} \underbrace{\binom{i}{a} v_x^{a-i}}_{=:X_{a,i}} \underbrace{\sum_{b=0}^{j} \underbrace{\binom{j}{b} v_y^{b-j}}_{=:Y_{b,j}} \lambda_{i,j}^{v}}_{=:Z_{i,j}}.$$

Ähnlich wie im letzten Abschnitt lautet der Algorithmus nun: Berechne zunächst die Potenzen und Binomialkoeffizienten. Berechne anschließend alle $X_{a,i}$ und $Y_{b,j}$. Hieraus bestimmt man alle $Z_{i,j}$, und schließlich berechnet man für $i = 0, \ldots, q$ und $j = 0, \ldots, q - i$ die Ergebnisse $\lambda((x + v_x)^i (y + v_y)^j)$, wobei die letzten beiden Schritte auch zusammengefaßt werden können.
Bei der Implementierung des Algorithmus sollte man auch hier bekannte Multiplikationen mit 1 aus Effizienzgründen eliminieren.

Limitierung zentraler Rekonstruktionen

Wie wir in der Einleitung zu diesem Kapitel auf den Seiten 36ff beschrieben haben, treten bei hyperbolischen Erhaltungsgleichungen Unstetigkeiten der Lösungen auf echten Untermannigfaltigkeiten von $\mathbf{R}^d$ beziehungsweise von Ω auf. Rekonstruktionen mit zentralen, lokalen Gittern weisen dort das bekannte Gibbs-Phänomen auf; in Abbildung 2.2 ist ein einfaches Beispiel hierzu illustriert. Das auftretende Oszillationsverhalten wird die Konvergenz des Verfahrens für feiner werdende Gitter zerstören. Daher ist ein **Limitierungsverfahren** zur Dämpfung dieses Phänomens notwendig, sofern man zentrale, lokale Gitter zur Berechnung von Rekonstruktionen hoher Ordnung verwendet. Wir wollen in diesem Abschnitt zwei Verfahren, die sich für Finite-Volumen-Gitter eignen, angeben und miteinander vergleichen. Der Algorithmus für diese beiden Verfahren besitzt die folgende Gestalt:

1. Für jede Zelle $\omega \in \mathcal{G}$ wird eine Rekonstruktion v_ω^q hoher Fehlerordnung auf einem zentralen lokalen Gitter berechnet. Daneben steht die konstante Funktion $v_\omega^0 = \bar{v}_\omega$, zur Verfügung.

2. Für jede Zelle $\omega \in \mathcal{G}$ wird ein geeigneter Konvexparameter $\theta_\omega \in [0, 1]$ berechnet. Dieser Parameter sollte in glatten Bereichen der Lösung möglichst dicht bei 1 liegen. Dagegen sollte er in der Nähe von Unstetigkeiten dicht bei 0 liegen.

3. Schließlich wird für jede Zelle die Konvexkombination

$$v_\omega := (1 - \theta_\omega) v_\omega^0 + \theta_\omega v_\omega^q \tag{2.16}$$

als Rekonstruktionsfunktion berechnet.

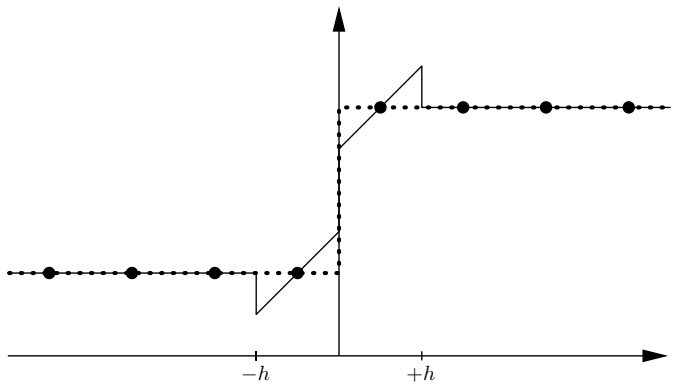


Abbildung 2.2: Zentrale Rekonstruktion mit linearen Polynomen bei Zellmittelwerten einer unstetigen Funktion. Das Gitter besteht aus den Zellen $[ih, (i+1)h]$ für $i \in \mathbf{Z}$. Die unstetige Funktion ist gestrichelt gezeichnet. Die zentrale, lineare Rekonstruktion nimmt an den Stellen $-h$ und $+h$ zu kleine und zu große Werte an, wenn man für die Rekonstruktion jeweils die linke und die rechte Nachbarzelle mit gleicher Gewichtung verwendet. Der maximale Fehler beträgt ein Viertel der Sprunghöhe der ursprünglichen Funktion. Er ist unabhängig von der Maschenweite des Gitters.

Das Verfahren von Barth und Jespersen

Zur Berechnung des Limitierungsfaktors θ_ω gibt es für unstrukturierte Finite-Volumen-Gitter das Verfahren von Barth und Jespersen [BJ89]. Die Idee dieses Verfahrens ist, den Konvexparameter so zu wählen, daß der Abstand zwischen Maximum und Minimum der Rekonstruktion innerhalb der Zelle beschränkt bleibt. Sei hierzu $\overline{\mathcal{N}}$ die Menge der direkt angrenzenden Nachbarn einer Zelle ω inklusive der Zelle ω selbst. Seien ferner $\bar{v}(\eta)$ für $\eta \in \overline{\mathcal{N}}$ die zugehörigen Zellmittelwerte. Dann werden die folgenden beiden Extremwerte bestimmt:

$$\begin{aligned} \bar{v}_{\min} &:= \min_{\eta \in \overline{\mathcal{N}}} \bar{v}(\eta), \\ \bar{v}_{\max} &:= \max_{\eta \in \overline{\mathcal{N}}} \bar{v}(\eta). \end{aligned}$$

Von der Rekonstruktion v_ω^q hoher Fehlerordnung werden Maximum und Minumum in der Zelle ω berechnet:

$$v_{\min}^q := \min_{x \in \overline{\omega}} \bar{v}(x), \tag{2.17}$$

$$v_{\max}^q := \max_{x \in \overline{\omega}} \bar{v}(x). \tag{2.18}$$

Der Konvexparameter $\theta_\omega \in [0,1]$ wird nun so bestimmt, daß die limitierte Funktion in dem Intervall $[\bar{v}_{\min}, \bar{v}_{\max}]$ verläuft:

$$\begin{aligned}
\theta_{\min} &:= \begin{cases} 1 & \text{falls } v^q_{\min} \geq \bar{v}(\omega) \\ \dfrac{\bar{v}(\omega) - \bar{v}_{\min}}{\bar{v}(\omega) - v^q_{\min}} & \text{sonst} \end{cases} \\
\theta_{\max} &:= \begin{cases} 1 & \text{falls } v^q_{\max} \leq \bar{v}(\omega) \\ \dfrac{\bar{v}_{\max} - \bar{v}(\omega)}{v^q_{\max} - \bar{v}(\omega)} & \text{sonst} \end{cases} \\
\theta_\omega &:= \min\{1, \theta_{\min}, \theta_{\max}\}.
\end{aligned}$$

In der Nähe von differenzierbaren Extremstellen der Ausgangsfunktion wird der Abstand des Extremwertes vom Datenintervall $[\bar{v}_{\min}, \bar{v}_{\max}]$ die Größenordnung $\mathcal{O}(h^2)$ in Abhängigkeit vom lokalen Zelldurchmesser h haben. Daher reduziert das beschriebene Limitierungsverfahren die Fehlerordnung auf $\mathcal{O}(h^2)$ unabhängig davon, welche Fehlerordnung das Verfahren zur Bestimmung von v^q_ω hatte. Für Probleme in mehreren Raumdimensionen wird das Limitierungsverfahren auch jenseits von Extremstellen die Fehlerordnung auf $\mathcal{O}(h)$ reduzieren, sofern die Anzahl an direkten Nachbarn sehr gering ist und hierdurch die Richtung des steilsten Aufstiegs in den Daten nicht genügend berücksichtigt wird. Im allgemeinen ist es bei dem Verfahren von Barth und Jespersen nicht lohnend, Polynome mit einem höheren Polynomgrad als $q = 1$ zu verwenden. In diesem Fall ist die Berechnung der Extremwerte (2.17) und (2.18) besonders einfach: Sie werden auf dem Rand von ω angenommen. Ist dieser polygonal, dann liegen die Extremwerte sogar in einer Ecke der Zelle.
Da das Limitierungsverfahren von Barth und Jespersen in ungünstigen Fällen keine höhere Fehlerordnung als $\mathcal{O}(h)$ zuläßt, wollen wir im nächsten Abschnitt eine ähnliches Verfahren vorstellen, welches allerdings in hinreichend glatten Bereichen der zu rekonstruierenden Funktion die hohe Fehlerordnung der zentralen Rekonstruktion v^q_ω beibehält und gleichzeitig in den praktischen Anwendungen zu dieser Arbeit sehr gute numerische Ergebnisse geliefert hat.

Ein ordnungserhaltendes Limitierungsverfahren

Zur Konstruktion eines alternativen Limitierungsverfahren, welches die Ordnung der zentralen Rekonstruktionen v^q_ω aus (2.16) in glatten Bereichen der Lösung nicht zerstört, werden wir die zentralen Rekonstruktionen benachbarter Zellen miteinander vergleichen. In glatten Bereichen der Lösung unterscheiden sich diese benachbarten Rekonstruktionen punktweise höchstens um die Fehlerordnung $\mathcal{O}(h^{q+1})$ der zentralen Rekonstruktion

v_ω^q. Dagegen liegt der Abstand dieser Werte in Bereichen von Sprungstellen in der Größenordnung des jeweiligen Sprunges. Wir wollen daher durch den Vergleich benachbarter Rekonstruktionen das Oszillationsverhalten der zentralen Rekonstruktion messen. Dieser Vergleich soll an den Gauß-Quadraturpunkten vorgenommen werden, die auch später für die Berechnung der Flüsse verwendet werden. Sei also q ein Quadraturpunkt zwischen den beiden benachbarten Zellen $\omega, \eta \in \mathcal{G}$. Durch die Wahl eines Limitierungsfaktors $\theta_\omega \in [0,1]$ wird die limitierte Funktion v_ω an dem Quadraturpunkt q einen Funktionswert innerhalb des folgenden Intervalles annehmen:

$$I_\omega \;:=\; \bar{v}_\omega + [0,1] \cdot \underbrace{(v_\omega^q(\mathsf{q}) - \bar{v}_\omega)}_{=:d_{\omega,\mathsf{q}}}.$$

Entsprechend wird der Funktionswert von v_η im Intervall

$$I_\eta \;:=\; \bar{v}_\eta + [0,1] \cdot \underbrace{(v_\eta^q(\mathsf{q}) - \bar{v}_\eta)}_{=:d_{\eta,\mathsf{q}}}$$

liegen. Wir wollen zunächst ein Verfahren angeben, welches für den Quadraturpunkt q jeweils einen neuen Funktionswert $z_{\omega,\mathsf{q}} \in I_\omega$ und einen neuen Funktionswert $z_{\eta,\mathsf{q}} \in I_\eta$ berechnet. Hierbei werden wir aus Symmetriegründen nur die Konstruktion des Funktionswertes $z_{\omega,\mathsf{q}}$ beschreiben; den Wert $z_{\eta,\mathsf{q}}$ erhält man durch Vertauschen der Rollen von η und ω. Der Funktionswert $z_{\omega,\mathsf{q}}$ ergibt in kanonischer Weise einen Limitierungsfaktor für den Quadraturpunkt q:

$$\theta_{\omega,\mathsf{q}} \;:=\; \begin{cases} 1 & \text{falls } d_{\omega,\mathsf{q}} = 0, \\ \dfrac{z_{\omega,\mathsf{q}} - \bar{v}_\omega}{d_{\omega,\mathsf{q}}} & \text{sonst.} \end{cases} \tag{2.19}$$

Wir berechnen also für jeden Quadraturpunkt der Zelle ω, der im Inneren des Gebietes Ω liegt, einen neuen Funktionswert $z_{\omega,\mathsf{q}}$ und hieraus einen Limitierungsfaktor $\theta_{\omega,\mathsf{q}}$. Aus den einzelnen Limitierungsfaktoren $\theta_{\omega,\mathsf{q}}$ soll dann später durch die Bildung eines geeigneten Minimums ein einzelner Limitierungsfaktor θ_ω für die Zelle berechnet werden. Hierauf werden wir genauer eingehen, sobald wir die Bestimmung der Funktionswerte $z_{\omega,\mathsf{q}}$ beschrieben haben.
Damit der Wert $z_{\omega,\mathsf{q}}$ in glatten Bereichen der Lösung höchstens die Fehlerordnung $\mathcal{O}(h^{q+1})$ hat, wählen wir diesen Wert im Intervall

$$I := v_\eta^q(\mathsf{q}) + [0,1] \cdot \underbrace{(v_\omega^q(\mathsf{q}) - v_\eta^q(\mathsf{q}))}_{=:d_\mathsf{q}},$$

denn die Länge $|d_{\mathsf{q}}|$ dieses Intervalles hat die Größenordnung $\mathcal{O}(h^{q+1})$ in glatten Bereichen der Lösung. An die Funktionswerte $z_{\omega,\mathsf{q}}$ und $z_{\eta,\mathsf{q}}$ haben wir die folgenden symmetrischen Anforderungen gestellt:

$$z_{\omega,\mathsf{q}} \in I \cap I_\omega, \tag{2.20}$$

$$z_{\eta,\mathsf{q}} \in I \cap I_\eta. \tag{2.21}$$

Wir bestimmen diese beiden Funktionswerte nach den folgenden Regeln:

1. Ist $I \cap I_\omega \cap I_\eta$ nichtleer, so wähle den Mittelpunkt dieses Intervalles als Funktionswert $z_{\omega,\mathsf{q}} = z_{\eta,\mathsf{q}}$.

2. Andernfalls wähle das eindeutig bestimmte Proximum aus $I \cap I_\omega$ an das Intervall $I \cap I_\eta$ als Funktionswert $z_{\omega,\mathsf{q}}$ und entsprechend das Proximum aus $I \cap I_\eta$ an das Intervall $I \cap I_\omega$ als Funktionswert $z_{\eta,\mathsf{q}}$.

Mit diesen Regeln werden die Forderungen (2.20) und (2.21) erfüllt. Die beiden neuen Werte $z_{\omega,\mathsf{q}}$ und $z_{\eta,\mathsf{q}}$ sind in hinreichend glatten Bereichen höchstens $\mathcal{O}(h^{q+1})$ von der Lösung entfernt. Für die Zelle ω werden wir nur die Limitierungsfaktoren $\theta_{\omega,\mathsf{q}}$ verwenden, die sich aus (2.19) ergeben. Sie lassen sich allerdings auch durch den nachfolgenden Algorithmus ohne Berechnung der Funktionswerte $z_{\omega,\mathsf{q}}$ allein aus den Größen d, $d_{\omega,\mathsf{q}}$ und d_η bestimmen. Sind die Sprunggrößen d_{q}, $d_{\omega,\mathsf{q}}$ und $d_{\eta,\mathsf{q}}$ die Eingabeparameter, dann lautet der Algorithmus zur Berechnung von $\theta_{\omega,\mathsf{q}}$:

falls $d_{\mathsf{q}} < 0$
    $d_{\mathsf{q}} = -d_{\mathsf{q}}$; $d_{\omega,\mathsf{q}} = -d_{\omega,\mathsf{q}}$; $d_{\eta,\mathsf{q}} = -d_{\eta,\mathsf{q}}$;
falls $d_{\omega,\mathsf{q}} \geq 0$
    Rückgabe 1;
falls $d_{\eta,\mathsf{q}} \leq 0$
    falls $-d_{\omega,\mathsf{q}} \leq d_{\mathsf{q}}$
        Rückgabe 0;
    Rückgabe $1 + d_{\mathsf{q}}/d_{\omega,\mathsf{q}}$;
falls $d_{\eta,\mathsf{q}} - d_{\omega,\mathsf{q}} \leq d_{\mathsf{q}}$
    Rückgabe 0;
$s = \min\{-d_{\omega,\mathsf{q}},\, d_{\mathsf{q}}\}$;
$t = d_{\mathsf{q}} - \min\{d_{\eta,\mathsf{q}},\, d_{\mathsf{q}}\}$;
Rückgabe $1 + (s+t)/(2d_{\omega,\mathsf{q}})$.

Man könnte nun den gemeinsamen Limitierungsfaktor θ_ω als Minimum der Werte $\theta_{\omega,\mathsf{q}}$ wählen. Diese Wahl ist allerdings viel zu pessimistisch, da die Empfindlichkeit der Parameter $\theta_{\omega,\mathsf{q}}$ hauptsächlich von dem Abstand

$|d_{\omega,\mathsf{q}}| = |v_\omega^q(\mathsf{q}) - \bar{v}_\omega|$ im Nenner von (2.19) abhängt. Dieser Abstand kann unter Umständen deutlich kleiner als der Rekonstruktionsfehler sein. In diesem Fall können auch betragsmäßig kleine Korrekturen zu sehr kleinen Werten $\theta_{\omega,\mathsf{q}}$ führen. Die Werte $\theta_{\omega,\mathsf{q}}$ sind dann sehr empfindlich gegenüber leichten Störungen der Daten. Man wird also nur an den Punkten die Limitierungsfaktoren berücksichtigen wollen, an denen auch der zugehörige Wert $|d_{\omega,\mathsf{q}}|$ in der Größenordnung der Variation der zentralen Rekonstruktion liegt. In der Praxis hat sich gezeigt, daß die Limitierung deutlich besser funktioniert, wenn man ähnlich wie bei dem Verfahren von Barth und Jespersen die Werte oberhalb des Mittelwertes $d_{\omega,\mathsf{q}} > 0$ von den Werten unterhalb des Mittelwertes $d_{\omega,\mathsf{q}} < 0$ unterscheidet. Wir bestimmen hierzu die folgenden Sprunggrößen:

$$\begin{aligned} d_\omega^- &:= \inf_{x\in\omega} v_\omega^q(x) - \bar{v}_\omega, \\ d_\omega^+ &:= \sup_{x\in\omega} v_\omega^q(x) - \bar{v}_\omega. \end{aligned}$$

Nach Vorgabe einer Konstante $M \in (0;1)$ (unabhängig von h) wählen wir nun diejenigen Quadraturpunkte aus, deren Sprung $d_{\omega,\mathsf{q}}$ entweder unterhalb von $M \cdot d_\omega^-$ oder oberhalb von $M \cdot d_\omega^+$ liegt. Wir fassen diese Quadraturpunkte zu einer Menge $\mathcal{Q}$ zusammen:

$$\begin{aligned} \mathcal{Q} := \{\mathsf{q} : \quad & \mathsf{q} \text{ ist innerer Quadraturpunkt von } \omega, \\ & d_{\omega,\mathsf{q}} < M \cdot d_\omega^- \text{ oder } d_{\omega,\mathsf{q}} > M \cdot d_\omega^+\}. \end{aligned}$$

Der Limitierungsfaktor θ_ω wird nun als Minimum der $\theta_{\omega,\mathsf{q}}$ dieser ausgewählten Quadraturpunkte berechnet:

$$\theta_\omega := \min_{\mathsf{q}\in\mathcal{Q}}\{1, \theta_{\omega,\mathsf{q}}\}. \tag{2.22}$$

In den Rechenbeispielen zu dieser Arbeit wurde die Konstante stets als $M = \cos(20°) \approx 92\%$ gewählt. Dies entspricht der Vorstellung, bei einer linearen Rekonstruktion und einer kreisförmigen Zelle nur diejenigen Quadraturpunkte zu berücksichtigen, die vom Schwerpunkt aus in einem Doppelkegel mit Öffnungswinkel $\pm 20°$ in Richtung des Gradienten liegen. Auch bei leicht verzerrten Zellen in zwei Raumdimensionen und quadratischer Rekonstruktion war dies eine Wahl, die zu guten Ergebnissen führte. Bei der numerischen Bestimmung der Zahlen d_ω^- und d_ω^+ wurden Maximum und Minimum bisher des numerischen Aufwandes wegen nur über die Menge der Quadraturpunkte, nicht über alle Punkte der Zelle $\overline{\omega}$ berechnet. Die genauere Berechnung der Extremwerte brachte in den gewählten Testbeispielen keinen Qualitätsgewinn.
Für das nun vollständig erklärte Verfahren wollen wir den nachfolgenden Satz beweisen, der absichert, daß die Fehlerordnung einer zentralen Rekonstruktion durch das Limitierungsverfahren nicht zerstört wird.

Satz 2.2 *Konvergieren für eine Folge von Finite-Volumen-Gittern die zentralen Rekonstruktionen v_ω^q aus (2.16) innerhalb der kompakten Menge $\overline{\Omega}$ gleichmäßig gegen die Funktion u mit der Konvergenzordnung $\mathcal{O}(h^{q+1})$, wobei $h \to 0$ der maximale Zelldurchmesser des Gitters sei, dann konvergieren die limitierten Rekonstruktionen gleichmäßig in $\overline{\Omega}$ gegen u mit der gleichen Konvergenzordnung $\mathcal{O}(h^{q+1})$, sofern eine von h unabhängige Konstante C existiert, so daß für alle Zellen ω der Gitterfolge und für beliebige Polynome $\wp \in \Pi^q$ die folgende Normäquivalenz gilt:*

$$\sup_{x\in\omega} |\wp(x)| \quad \leq \quad \frac{C}{|\omega|} \int\limits_{\omega} |\wp(x)| \, \mathsf{d}x. \tag{2.23}$$

Bemerkung Sobald wir den Satz unter diesen Voraussetzungen bewiesen haben, werden wir noch einmal auf den technischen Zusatz mit der Normäquivalenz eingehen, der das Degenerieren des Gitters verhindern soll. Wir werden nachträglich zeigen, daß Gitterfolgen, die durch isotrope Verfeinerungen entstehen, diese Normäquivalenz erfüllen.

Beweis Wir wählen einen festen Punkt $x \in \Omega$ und nehmen an, daß mit ω eine Zelle des jeweiligen Gitters bezeichnet wird, die x enthält. Wir verzichten also aus Gründen der besseren Lesbarkeit auf die Indizierung der Gitterfolge. Randpunkte des Gebietes oder Zellränder können bei dem Beweis außer Betracht gelassen werden, weil auf jeder Zelle stetige Funktionen rekonstruiert werden und die oberen Schranken der Fehlerabschätzung entsprechend auch für diese Punkte gelten. Für die zentrale Rekonstruktion hoher Ordnung definieren wir die folgende Fehlerfunktion

$$e_h(x) \quad := \quad v_\omega^q(x) - u(x).$$

Mit der Konvexkombination (2.16) folgt

$$\begin{aligned} v_\omega(x) - u(x) &= (1-\theta_\omega)v_\omega^0 + \theta_\omega v_\omega^q(x) - u(x) \\ &= (1-\theta_\omega)(v_\omega^0 - v_\omega^q(x)) + v_\omega^q(x) - u(x) \\ &= (1-\theta_\omega)(v_\omega^0 - v_\omega^q(x)) + e_h(x). \end{aligned} \tag{2.24}$$

Der Fehlerterm e_h ist nach Voraussetzung des Satzes gleichmäßig in $\overline{\Omega}$ durch

$$\sup_{x\in G} |e_h(x)| \quad \leq \quad \mathcal{O}(h^{q+1}). \tag{2.25}$$

beschränkt.
Für diejenigen Zellen, bei denen die Menge $\mathcal{Q}$ der berücksichtigten Quadraturpunkte leer ist, gilt $\theta_\omega = 1$, und die zentrale Rekonstruktion hoher

Ordnung wird nicht limitiert. Ist dagegen $\mathcal{Q}$ nicht leer, dann kann für den Quadraturpunkt $\mathsf{q} \in \mathcal{Q}$ die Sprunghöhe $|d_{\omega,\mathsf{q}}|$ durch

$$|d_{\omega,\mathsf{q}}| \;>\; M \cdot \min\{|d_\omega^-|,\, |d_\omega^+|\}$$

nach unten abgeschätzt werden. Unter Verwendung von (2.19) ergibt sich für den Faktor $(1-\theta_\omega)$ in (2.24)

$$\begin{aligned} 0 \;\leq\; (1-\theta_\omega) \;&\leq\; \max_{\mathsf{q}\in\mathcal{Q}} \frac{v_\omega^q(\mathsf{q}) - z_{\omega,\mathsf{q}}}{d_{\omega,\mathsf{q}}} \\ &\leq\; \frac{\max\limits_{\mathsf{q}\in\mathcal{Q}} |v_\omega^q(\mathsf{q}) - z_{\omega,\mathsf{q}}|}{M \cdot \min\{|d_\omega^-|,\, |d_\omega^+|\}}. \end{aligned}$$

Der Funktionswert $z_{\omega,\mathsf{q}}$ wird aus dem Intervall I ausgewählt. Dies ist gerade der Bereich zwischen den Funktionswerten der zentralen Rekonstruktion innerhalb der Zelle ω und der jeweils benachbarten Zelle. Da beide Funktionswerte nach Voraussetzung nicht weiter als $e_h(\mathsf{q})$ von $u(\mathsf{q})$ entfernt sind, ist auch der Abstand zwischen $v_\omega^q(\mathsf{q})$ und $z_{\omega,\mathsf{q}}$ durch $e_h(\mathsf{q})$ beschränkt. Es folgt:

$$0 \;\leq\; (1-\theta_\omega) \;\leq\; \frac{\sup\limits_{x\in\Omega} |e_h(x)|}{M \cdot \min\{|d_\omega^-|,\, |d_\omega^+|\}}. \tag{2.26}$$

Der Term $v_\omega^0 - v_\omega^q(x) = \bar{v}_\omega - v_\omega^q(x)$ wird für $x \in \omega$ durch die Summe $|d_\omega^-| + |d_\omega^+|$ beschränkt. Zusammen mit (2.26) erhalten wir aus (2.24) den maximalen Fehler in der Zelle ω

$$\sup_{x\in\omega} |v_\omega(x) - u(x)| \;\leq\; \left(\frac{1}{M} \underbrace{\frac{|d_\omega^-| + |d_\omega^+|}{\min\{|d_\omega^-|,\, |d_\omega^+|\}}}_{=:Q_\omega} + 1 \right) \sup_{x\in\omega} |e_h(x)|. \tag{2.27}$$

Jetzt müssen wir nur noch zeigen, daß der Quotient Q_ω unabhängig von ω nach oben beschränkt ist. Hierzu betrachten wir das verschobene Polynom

$$\wp := v_\omega^q - \begin{cases} \inf\limits_{x\in\omega} v_\omega^q(x) & \text{für } |d_\omega^-| \leq |d_\omega^+|, \\ \sup\limits_{x\in\omega} v_\omega^q(x) & \text{sonst.} \end{cases}$$

und erhalten mit der Normäquivalenz (2.23) die Abschätzung

$$Q_\omega = \frac{|d_\omega^-| + |d_\omega^+|}{\min\{|d_\omega^-|,\, |d_\omega^+|\}} = \frac{\sup\limits_{x\in\omega} |\wp(x)|}{\dfrac{1}{|\omega|} \displaystyle\int_\omega |\wp(x)| \; \mathsf{d}x} \leq C.$$

Setzt man dies und (2.25) in (2.27) ein, so ist der Satz 2.2 bewiesen. ∎

Im Satz 2.2 haben wir durch die Normäquivalenz (2.23) zusätzliche Restriktionen an die Gitterfolge gestellt. Daher muß jetzt untersucht werden, ob die Bedingungen für praxisrelevante Gitter noch erfüllbar sind. Für eine feste Zelle ω gibt es immer eine Konstante C_ω, so daß die Bedingung

$$\sup_{x\in\omega} |\wp(x)| \;\leq\; \frac{C_\omega}{|\omega|} \int\limits_\omega |\wp(x)| \; \mathsf{d}x. \tag{2.28}$$

für alle Polynome $\wp \in \Pi^q$ erfüllt wird, da Π^q endlichdimensional ist und Normen endlichdimensionaler Räume stets äquivalent sind. Wir betrachten das Verhalten von (2.28) unter bijektiven affinen Abbildungen $\mathcal{A}$ der Zelle ω. Man überlegt sich leicht, daß sowohl das Supremum als auch das Zellmittel sich unter $\mathcal{A}$ nicht ändern:

$$\begin{aligned} \sup_{x\in\omega} |\wp(x)| &= \sup_{x\in\mathcal{A}\omega} |(\wp\circ\mathcal{A}^{-1})(x)|, \\ \frac{1}{|\omega|}\int\limits_\omega |\wp(x)| \; \mathsf{d}x &= \frac{1}{|\mathcal{A}\omega|}\int\limits_{\mathcal{A}\omega} |(\wp\circ\mathcal{A}^{-1})(x)| \; \mathsf{d}x. \end{aligned}$$

Da bijektive affine Abbildungen des $\mathbf{R}^d$ bijektive lineare Abbildungen des Polynomraumes Π^q induzieren, kann die Konstante in (2.28) unabhängig von beliebigen affinen Abbildungen $\mathcal{A}$ der Zellen gewählt werden: $C_\omega = C_{\mathcal{A}\omega}$.
Jetzt können wir sicher sein, daß die Konvergenzaussage des Satzes 2.2 mindestens dann gilt, wenn wir Gitterfolgen betrachten, deren Zellen alle durch affine Abbildungen eines endlichen Reservoirs an Ausgangszellen entstanden sind. Damit ist die Konvergenz wenigstens für alle beliebigen Simplex-Gitter (Strecken, Dreiecke, Tetraeder) gesichert.

ENO-Eigenschaft für lineare Rekonstruktionen

In [HEOC87] wird einem Rekonstruktionsverfahren in einer Raumdimension die ENO-Eigenschaft[5] zugeschrieben, wenn die Totalvariation der rekonstruierten Funktion hinreichend schnell gegen die Totalvariation der Ausgangsfunktion konvergiert. Besitzt die Rekonstruktion v die Fehlerordnung $\mathcal{O}(h^{q+1})$ bezüglich der ursprünglichen Funktion u, dann wird von der Totalvariation gefordert, daß sie mit der gleichen Fehlerordnung konvergiert:

$$\mathsf{TV}_\Omega(v) \leq \mathsf{TV}_\Omega(u) + \mathcal{O}(h^{q+1}). \tag{2.29}$$

[5] essentially non-oscillatory = wesentlich nicht oszillierend

In Bereichen, in denen die Funktion hinreichend glatt ist, wird die zentrale Rekonstruktion diese Konvergenzordnung der Totalvariation zeigen. Weil das Limitierungsverfahren die Totalvariation der zentralen Rekonstruktion wegen $\theta_\omega \in [0,1]$ verringert, gilt auch Entsprechendes für die limitierte Rekonstruktion. Wir interessieren uns nun insbesondere für das Verhalten der Totalvariation in der Nähe von Sprungstellen, wie sie beispielsweise in Abbildung 2.2 zu sehen ist. In der Nähe solcher Sprungstellen darf man nicht erwarten, daß die Zellmittelwerte als Eingabedaten der Funktion u genauer als $\mathcal{O}(h)$ zur Verfügung stehen. Es ist daher in der Praxis zu erwarten, daß auch die Totalvariation höchstens mit der Konvergenzordnung $\mathcal{O}(h)$ gegen die Totalvariation der Ausgangsfunktion u konvergiert. Wir schwächen also die Forderung (2.29) in der Nähe von echten Sprungstellen wie folgt ab:

$$\mathsf{TV}_\Omega(v) \leq \mathsf{TV}_\Omega(u) + \mathcal{O}(h). \tag{2.30}$$

Damit ist wenigstens sichergestellt, daß die Totalvariation konvergiert. Wir wollen die Eigenschaft (2.30) für das Limitierungsverfahren des vorangehenden Abschnittes für den Spezialfall nachweisen, daß mit linearen Polynomen ($q = 1$) rekonstruiert wird. In diesem Fall liegen der maximale sowie der minimale Funktionswert der Rekonstruktion jeweils in den beiden Randpunkten einer Zelle, und damit besteht die Menge $\mathcal{Q}$ auch aus diesen beiden Randpunkten. In diesem Fall werden bei der Minimierung (2.22) beide Quadraturpunkte berücksichtigt.
Ähnlich wie in [HEOC87] betrachten wir ein festes Intervall $[a,b]$ und nehmen an, daß die Funktion u innerhalb des Intervalles hinreichend glatt ist und höchstens Unstetigkeiten am rechten oder linken Intervallrand vorliegen.[6] Aus Symmetriegründen reicht es, den Fall einer Unstetigkeit am rechten Intervallrand b zu studieren.
Für die zentrale Rekonstruktion einer Zelle sollen nur die Zellmittel der linken und der rechten Nachbarzelle berücksichtigt werden. Dann wird eine Unstetigkeit bei b höchstens die zentrale Rekonstruktion derjenigen Zelle $\omega = [\mathsf{q}, b] \subseteq [a,b]$ beeinflussen, die am rechten Rand des Intervalles $[a,b]$ liegt. Nach den Voraussetzungen können wir annehmen, daß die Totalvariation im Intervall $[a,\mathsf{q}]$ mindestens mit $\mathcal{O}(h)$ gegen die Totalvariation der Funktion u in $[a,b]$ konvergiert. Daher verbleibt zu zeigen, daß durch die Rekonstruktion in der Zelle ω die Totalvariation höchstens um einen Fehler der Ordnung $\mathcal{O}(h)$ „gestört“ wird.

[6] In der genannten Veröffentlichung werden nicht Zellmitteldaten, sondern Funktionswerte interpoliert. Es wird angenommen, daß die Unstetigkeiten weit genug auseinander liegen und entweder linksseitig oder rechtsseitig eine konvergente Rekonstruktion gebildet werden kann.

Die linke Nachbarzelle von ω sei die Zelle ℓ. Wir nehmen an, daß das Gitter so fein ist, daß die Zelle ℓ und ihr linker Nachbar vollständig in $[a, b]$ liegen. Dann konvergiert die zentrale Rekonstruktion v_ℓ^q, und wir können annehmen, daß der Funktionswert $v_\ell^q(\mathsf{q})$ höchstens um $\mathcal{O}(h)$ von den Zellmittelwerten $\bar{v}_\ell$ und $\bar{v}_\omega$ abweicht. Nun wird für die Zelle ω am Quadraturpunkt q ein neuer Funktionswert $z_{\omega,\mathsf{q}}$ durch das Limitierungsverfahren konstruiert. Dieser ist als Proximum der Menge $I \cap I_\omega$ an $I \cap I_\ell$ auch höchstens $\mathcal{O}(h)$ vom Zellmittel $\bar{v}_\omega$ entfernt (siehe die Regeln auf Seite 57). Die limitierte Rekonstruktion v_ω am Punkt q kann nicht weiter als der konstruierte Funktionswert $z_{\omega,\mathsf{q}}$ vom Zellmittel $\bar{v}_\omega$ entfernt sein. Weil v_ω das Zellmittel in der Zelle ω interpoliert, ist die Steigung von v_ω unabhängig von h beschränkt. Damit ist die in der Zelle ω vorliegende Störung der Totalvariation mit der Ordnung $\mathcal{O}(h)$ nach dem Limitieren beschränkt. Dies gilt, unabhängig von beliebigen Störungen außerhalb des Intervalles $[a, b]$, da in die obigen Betrachtungen weder die zentrale Rekonstruktion v_ω^q noch das Zellmittel der rechten Nachbarzelle von ω eingegangen sind. Wir sehen also, daß die Totalvariation der limitierten Rekonstruktion mindestens mit der in (2.30) beschriebenen Größenordnung konvergiert.

Gewichtete ENO-Rekonstruktionen

Das ordnungserhaltende Limitierungsverfahren des letzten Abschnittes soll sowohl mit dem Limitierungsverfahren von Barth und Jespersen als auch mit einer Variante des gewichteten ENO-Verfahrens aus [LOC94] in Kombination mit dem Oszillationsindikator aus [JS96] in einer und zwei Raumdimensionen numerisch verglichen werden. Für die Vergleichsrechnungen in zwei Raumdimensionen stand das Verfahren von O. Friedrich [Fri97] zur Verfügung. Für die eindimensionalen Vergleiche wurde das Verfahren analog zu diesem zweidimensionalen Verfahren implementiert. Der Vollständigkeit halber erläutern wir in diesem Abschnitt die gewählten Parameter des eindimensionalen Verfahrens.

Hier bestand das Gitter stets aus einer äquidistanten Zerlegung des Intervalles $[0, 1]$. Da die Beispiele periodisch waren, war keine gesonderte Randbedingung zu implementieren: Jede Zelle hatte jeweils genau einen linken und einen rechten Nachbarn. Die Beispiele wurden mit linearer und quadratischer Rekonstruktion gerechnet. In beiden Fällen wurden jeweils drei Rekonstruktionen pro Zelle berechnet, die stets das lokale Zellmittel $\bar{v}_\omega$ interpolierten. Für die linearen Rekonstruktionen verbleibt daher ein weiterer Freiheitsgrad, und für die quadratischen Rekonstruktionen entsprechend zwei weitere Freiheitsgrade. Es wurde jeweils eine **linksseitige**, eine **zentrale** und eine **rechtsseitige** Rekonstruktion berechnet, die wir mit v_ω^ℓ, v_ω^z

und v_ω^r bezeichnen. Diese Rekonstruktionen ergeben sich in der folgenden kanonischen Weise: Die linksseitige Rekonstruktion v_ω^ℓ interpolierte jeweils das Zellmittel des linken Nachbarn, die rechtsseitige Rekonstruktion v_ω^r das Zellmittel der rechten Nachbarzelle. Bei quadratischer Rekonstruktion interpolierte die linksseitige Rekonstruktion zusätzlich das Zellmittel des nächsten linken Nachbarn (linker Nachbar des linken Nachbarn) und die rechtsseitige Rekonstruktion interpolierte das Zellmittel des nächsten rechten Nachbarn. Bei linearer Rekonstruktion wurde die zentrale Rekonstruktion als arithmetisches Mittel der linksseitigen und der rechtsseitigen Rekonstruktion berechnet. Bei quadratischer Rekonstruktion interpolierte die zentrale Rekonstruktion gerade die Zellmittel des linken und rechten Nachbarn.
Jeder der drei Rekonstruktionen wurde ein Gewicht g zugeordnet, welches abhängig vom Oszillationsverhalten in der Zelle ω gesteuert wird:

$$g(v) := \left(\frac{1}{\epsilon + \mathsf{TV}_\omega(v)} \right)^p .$$

Hierin ist $\epsilon \approx 10^{-20}$ abhängig von der Maschinengenauigkeit und soll die Division durch Null verhindern. Für den Exponenten wurde in den Anwendungen der Wert $p = 4$ gewählt. Er hängt von der Wahl des Oszillationsindikators ab. In den eindimensionalen Anwendungen wurde die Totalvariation $\mathsf{TV}_\omega(v)$ der jeweiligen Rekonstruktion v in der Zelle ω verwendet. Diese ist jedoch für Polynome höheren Grades oder auch für quadratische Polynome in zwei Raumdimensionen numerisch schwer zu bestimmen. Statt der Totalvariation wurde daher, wie in der Arbeit [Fri97] beschrieben, der folgende Ausdruck verwendet:

$$\sqrt{\int_\omega \|\nabla v\|_2^2 \, \mathsf{d}x}.$$

Aus den drei zu einer Zelle gehörigen Rekonstruktionen wird nun eine Konvexkombination v_ω gebildet. Hierbei müssen die unterschiedlichen Rekonstruktionen zusätzlich verschieden gewichtet werden, damit das resultierende Verfahren nicht instabil wird. Für lineare Rekonstruktion wurde durch numerische Experimente ermittelt, daß die zentrale Rekonstruktion mindestens doppelt gewichtet werden muß:

$$v_\omega := \frac{g(v_\omega^\ell)v_\omega^\ell + 2g(v_\omega^z)v_\omega^z + g(v_\omega^r)v_\omega^r}{g(v_\omega^\ell) + 2g(v_\omega^z) + g(v_\omega^r)}.$$

Im Falle quadratischer Polynome wurde die zentralen Rekonstruktionen sogar sechsfach gewichtet, um das resultierende Verfahren zu stabilisieren:

$$v_\omega := \frac{g(v_\omega^\ell)v_\omega^\ell + 6g(v_\omega^z)v_\omega^z + g(v_\omega^r)v_\omega^r}{g(v_\omega^\ell) + 6g(v_\omega^z) + g(v_\omega^r)}. \tag{2.31}$$

Vergleich der Verfahren in numerischen Beispielen

In den nachfolgenden Abschnitten wollen wir die verschiedenen Rekonstruktionsverfahren in numerischen Tests miteinander vergleichen. Um den Einfluß anderer Komponenten zu vermeiden, wurden alle Rechnungen mit dem dreistufigen Runge-Kutta-Verfahren berechnet. Bei den Rechnungen in zwei Raumdimensionen wurde stets die Gaußsche Quatraturformel mit zwei Punkten für die Randintegration verwendet.

Verschieben einer Sinus-Kurve

Als erstes eindimensionales Testbeispiel wurde eine Sinus-Kurve im Intervall $[0, 1]$ mit der linearen Advektion $F(u) = u$ transportiert. Die Zustandsverteilung zur Zeit $t = 0$ war:

$$u_0(x) := \tfrac{1}{2}(\sin(2\pi x) + 1). \tag{2.32}$$

Die numerischen Ergebnisse für die Zeit $T = 10$ wurden mit äquidistanten Gittern aus 21 Zellen beziehungsweise 42 Zellen berechnet. Die Gitter wurden jeweils am linken und rechten Rand periodisch fortgesetzt. In der Abbildung 2.3 sind die Ergebnisse mit dem gewichteten ENO-Verfahren, dem ordnungserhaltendem Limiter und dem Limiter von Barth und Jespersen für die beiden Gitter zusammengetragen. Die Lösungen mit dem Limiter von Barth und Jespersen zeigen deutliche Verluste der Rekonstruktionsordnung in der Nähe der beiden Extrema. Das ordnungserhaltende Limitierungsverfahren ist in der mittleren Abbildung gezeigt. Hier ist zu bemerken, daß bei linearer Rekonstruktion die Lage der Kurve falsch ist. Allerdings liegt die Kurve bis auf die Darstellungsgenauigkeit auf der Lösung ohne Anwendung des Limiters. Dies bedeutet, daß der Limiter in diesem Beispiel genau das erwünschte Verhalten aufweist: Die zentrale Rekonstruktion wird nahezu nicht limitiert. Das gewichtete ENO-Verfahren weist bei linearer Rekonstruktion sehr dissipative Ergebnisse auf. Um so besser dagegen sind die Ergebnisse im Falle quadratischer Polynome. Hier ist allerdings zu bemerken, daß die zentrale Rekonstruktion erheblich übergewichtet wird und zusätzlich die Zeischrittweite reduziert werden mußte. Andernfalls führte insbesondere der Einfluß der linksseitigen Rekonstruktion zu Lösungen, deren Maximum oberhalb und deren Minimum unterhalb der exakten Lösung lag. Um dies zu vermeiden, wurde die zentrale Rekonstruktion sechsfach gewertet (siehe Gleichung (2.31)).

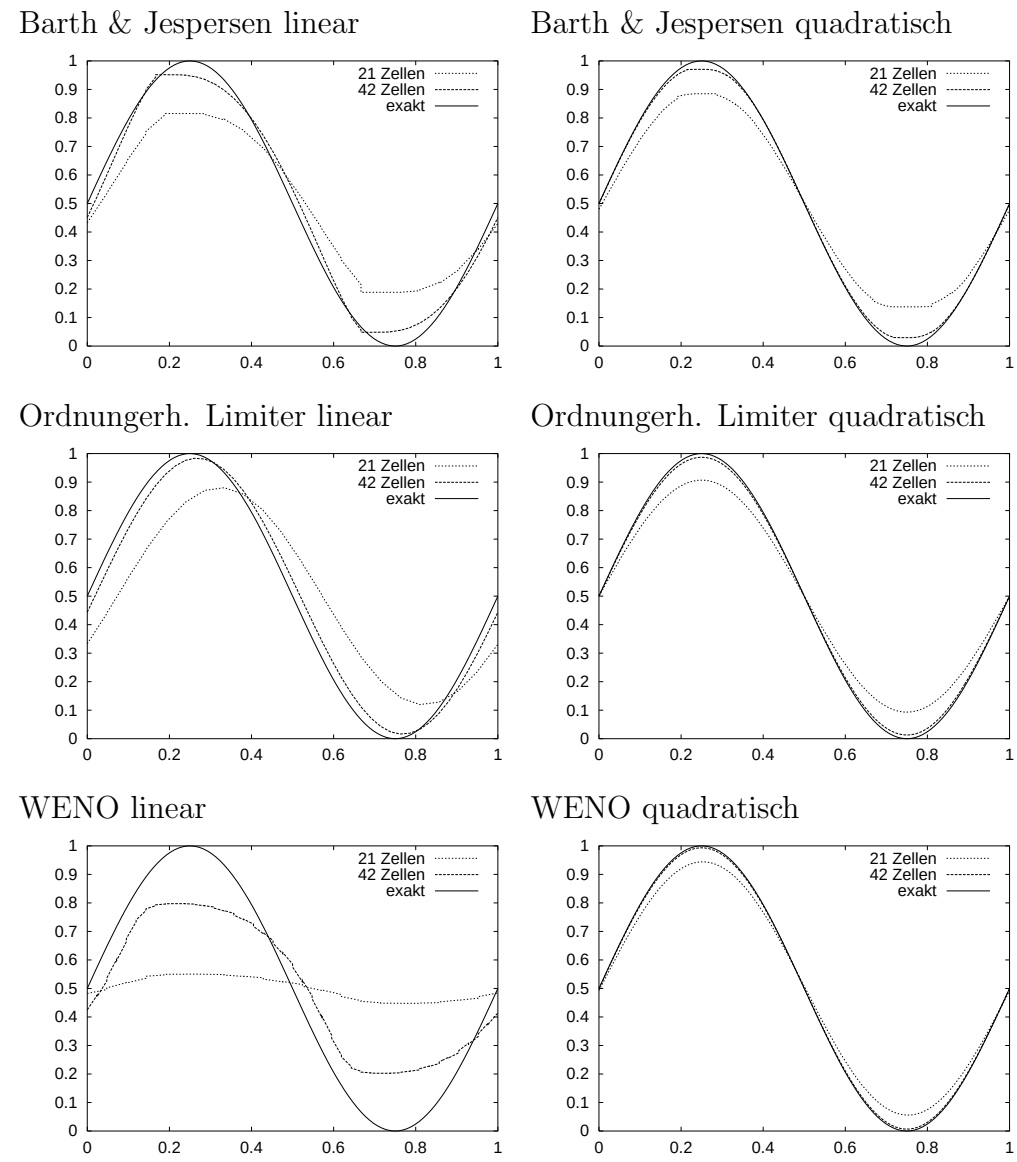
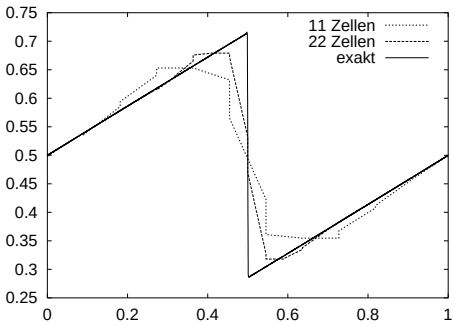
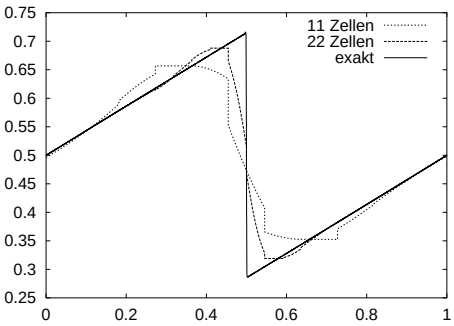


Abbildung 2.3: Lineare Advektion einer Sinus-Kurve zur Zeit $T = 10$. Auf der linken Seite sind die Verfahren unter Verwendung linearer Polynome, auf der rechten Seite mit quadratischen Polynomen jeweils auf einem Gitter mit 21 Zellen und einem mit 42 Zellen zusammen mit der exakten Lösung dargestellt.

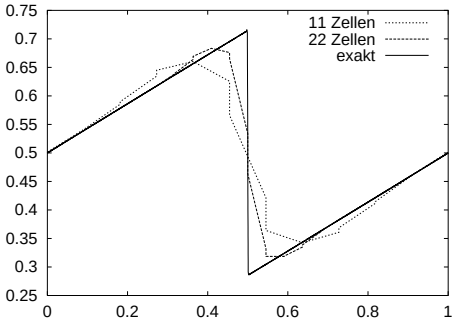
Barth & Jespersen linear



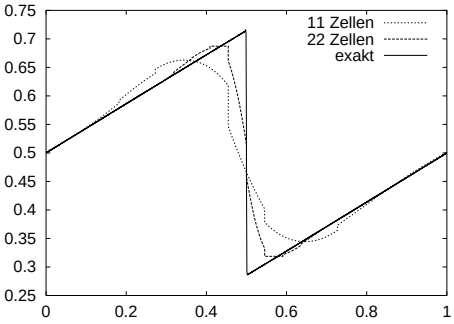
Barth & Jespersen quadratisch



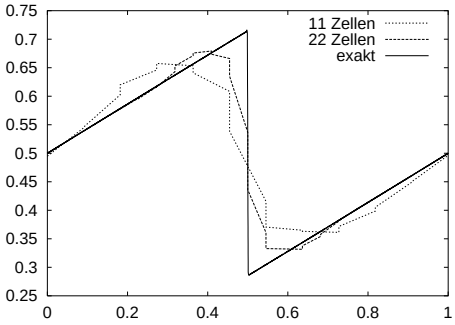
Ordnungerh. Limiter linear



Ordnungerh. Limiter quadratisch



WENO linear



WENO quadratisch

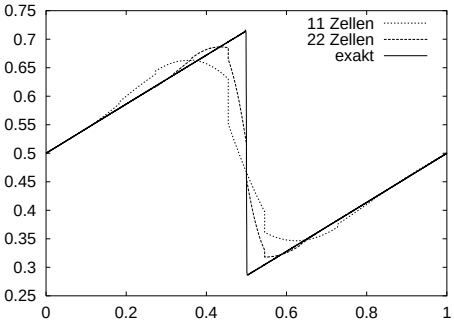


Abbildung 2.4: Entwicklung einer Unstetigkeit bei Burgers' Gleichung $F(u) = u^2$ zur Zeit $T = 1$ bei der Anfangsbedingung aus Gleichung (2.32). Auf der linken Seite sind die Verfahren unter Verwendung linearer Polynome, auf der rechten Seite mit quadratischen Polynomen jeweils auf einem Gitter mit 11 Zellen und einem mit 22 Zellen zusammen mit der exakten Lösung dargestellt.

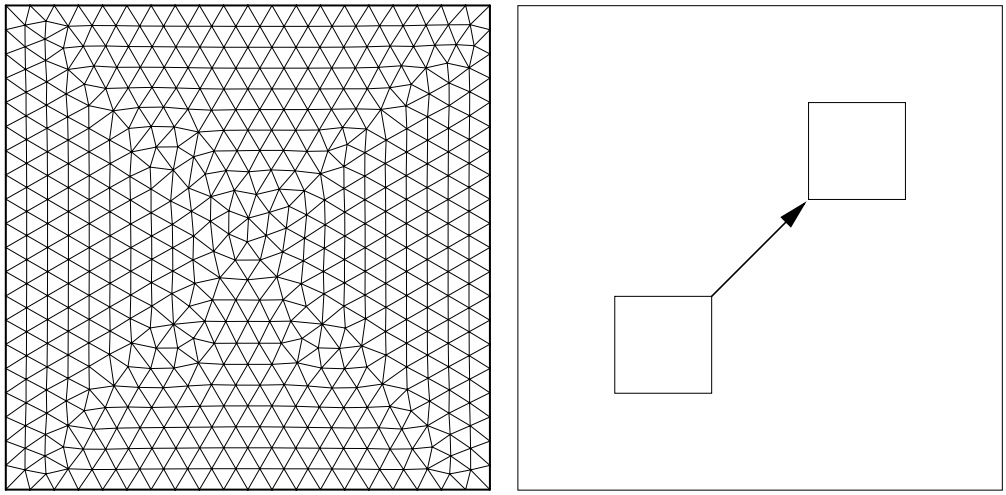


Abbildung 2.5: Verschieben des Quadrates aus Gleichung (2.33) bis zur Zeit $T = 0.4$ mit der Flußfunktion $F(u) = (u, u)^{\mathsf{t}}$. Innerhalb des kleinen Quadrates lag zu Anfang der Funktionswert 1 und außerhalb der Wert 0 vor. Die Ausgangstriangulierung mit der Maschenweite $h = 1/20$ am Rand ist links oben zu sehen. Durch Viertelung der Dreiecke mit dem Adaptionsverfahren des letzten Kapitels wurde ein Gitter mit halber Maschenweite erzeugt.

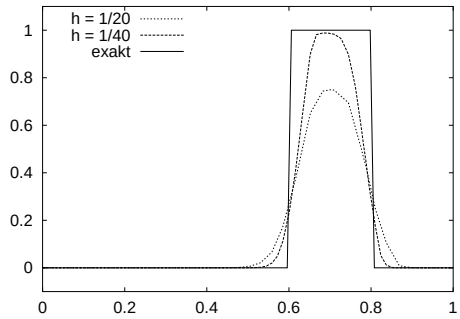
Entwicklung einer Unstetigkeit — Burgers' Gleichung

Als weiteres Beispiel wurde die Entwicklung einer Unstetigkeit bei Burgers' Gleichung $F(u) = u^2$ untersucht. Als Anfangsverteilung wurde die Funktion aus Gleichung (2.32) verwendet. Die Ergebnisse sind in Abbildung 2.4 für die Zeit $T = 1$ dargestellt. Keines der beschriebenen Verfahren zeigt wesentliche Oszillationen in der Nähe der Unstetigkeit. Die Ergebnisse mit quadratischer Rekonstruktion lösen den Sprung etwas schärfer auf als die mit linearer Rekonstruktion.

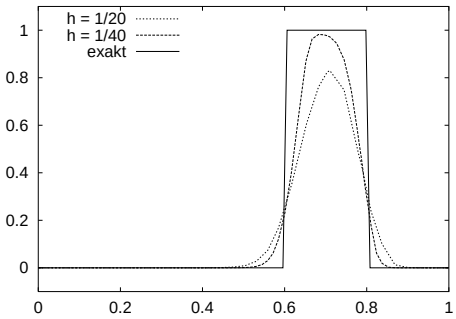
Verschieben eines Quadrates

In zwei Raumdimensionen wurde mit der Flußfunktion $F(u) = (u, u)^{\mathsf{t}}$ eine unstetige Anfangsverteilung verschoben, wie es in der Abbildung 2.5 auf der rechten Seite gezeigt wird. Das Gesamtgebiet war das Einheitsquadrat $[0, 1] \times [0, 1]$. Die Untersuchung wurde auf zwei Gittern durchgeführt, die sich aus unterschiedlich feinen Triangulierungen ergaben. Die gröbere hiervon ist auf der linken Seite von Abbildung 2.5 zu sehen. Die feinere Triangulierung wurde durch Vierteilung aller Dreiecke mit dem Adaptionsverfahren des letz-

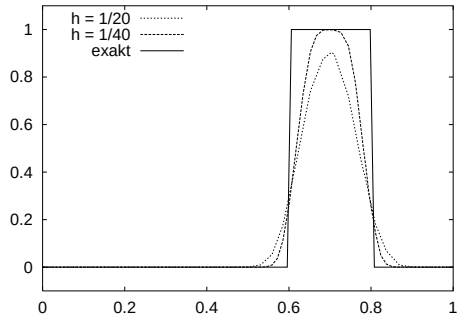
Barth & Jespersen linear



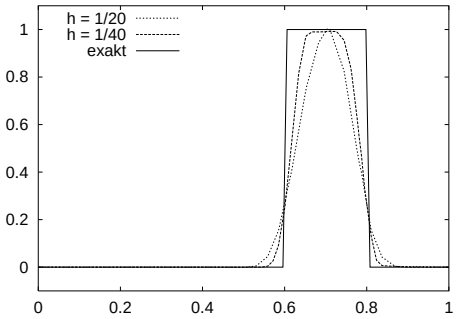
Ordnungerh. Limiter linear



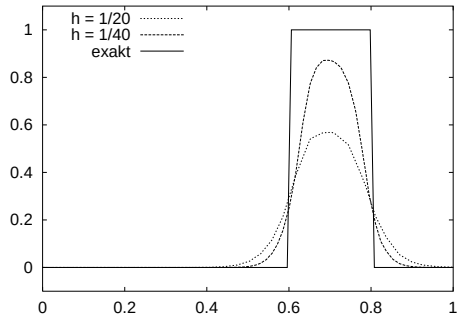
Ordnungerh. Limiter quadratisch



Ordnungerh. Limiter kubisch



WENO linear



WENO quadratisch

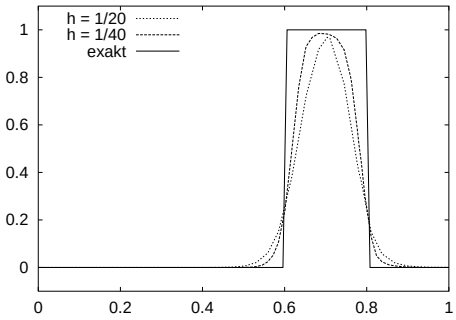


Abbildung 2.6: Querschnitte durch die Lösungen längs der Verbindungslinie von $(0,0)^t$ nach $(1,1)^t$ für das Problem aus Abbildung 2.5. Es sind jeweils die Lösungen für die beiden unterschiedlich feinen Gitter im Vergleich zur exakten Querschnittslösung dargestellt.

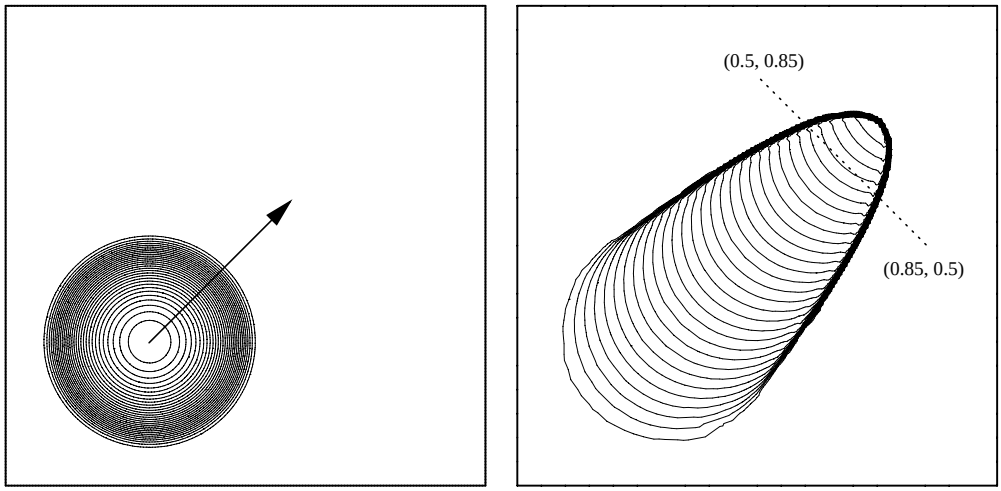


Abbildung 2.7: Links die Anfangsverteilung (2.34) (30 Isolinien), rechts die als „Referenzlösung“ bezeichneten Ergebnisse für Burgers' Gleichung auf einem feinem Gitter mit $h = 1/160$ (30 Isolinien). Zusätzlich ist die Schnittlinie angegeben, die für die Ergebnisse aus Abbildung 2.8 verwendet wurde.

ten Kapitels erzeugt. Der gewählte Anfangszustand ist:

$$u_0(x,y) = \begin{cases} 1 & 0.2 \leq x \leq 0.4 \text{ und } 0.2 \leq y \leq 0.4 \\ 0 & \text{sonst.} \end{cases} \tag{2.33}$$

In der Abbildung 2.6 sind Querschnitte längs der Diagonalen $y = x$ für die unterschiedlichen Verfahren zur Zeit $T = 0.4$ dargestellt. Dabei wurde das Limitierungsverfahren von Barth und Jespersen nicht für quadratische Polynome implementiert. Dagegen wurde das ordnungserhaltende Limitierungsverfahren zusätzlich mit kubischen Polynomen implementiert und getestet. Allerdings ist das Verfahren für diesen Rekonstruktionsgrad schon relativ empfindlich: Die lokalen Gitter sind in der Nähe des Randes stark einseitig und bisher wurden keine Randbedingungen innerhalb des Rekonstruktionsschrittes berücksichtigt. Ebenfalls getestet wurde das Verfahren für Polynome vierten Grades: Es erwies sich jedoch als so instabil, daß bereits nach kurzer Zeit singuläre Werte vorlagen.
Bis auf das ENO-Verfahren mit linearer Rekonstruktion, welches wieder sehr dissipativ ist, sind alle Ergebnisse relativ ähnlich. Die Verfahren mit quadratischer Rekonstruktion sind geringfügig schärfer als die mit linearer Rekonstruktion. Das Ergebnis mit kubischer Rekonstruktion ist in diesem Beispiel noch etwas besser als die quadratischen Verfahren. Allerdings war der Rechenaufwand 3.5 mal höher als mit quadratischer Rekonstruktion und betrug das 2.2-fache des ENO-Verfahrens mit quadratischer Rekonstruktion.

Entwicklung einer Unstetigkeit — Burgers' Gleichung in 2d

Für die beiden Gitter des letzten Abschnittes wurde die Entwicklung einer Unstetigkeit aus der nachfolgenden, glatten Anfangsverteilung berechnet:

$$\begin{aligned} u_0(x,y) &:= \begin{cases} \exp\left(\dfrac{d(x,y)}{d(x,y)-1/16}\right) & \text{falls } d(x,y) < 1/16 \\ 0 & \text{sonst.} \end{cases} \\ d(x,y) &:= (x-0.3)^2 + (y-0.3)^2 \end{aligned} \tag{2.34}$$

Diese Funktion ist auf der linken Seite der Abbildung 2.7 dargestellt. Auf der rechten Seite dieses Bildes ist die Entwicklung für die Flußfunktion $F(u) = (1,1) \cdot u^2$ zur Zeit $T = 0.4$ in Höhenlinien zu sehen. Die dort abgebildete Lösung wurde auf einem feineren Gitter ($h = 1/160$) berechnet; sie diente als „Referenzlösung" für die Ergebnisse aus Abbildung 2.8. Dort sind die Querschnitte der Lösungen längs der in der Abbildung 2.7 gezeigten Strecke dargestellt.

Stoßrohrproblem von Lax in 2d

Eine der Standardkonfigurationen für die Verifikation eines Verfahrens zur numerischen Berechnung der Euler-Gleichungen ist das Stoßrohrproblem von Lax, welches für eine Raumdimension definiert ist und durch die folgenden Anfangswerte festgelegt ist (vergleiche [Bot95]):

$$(\rho, \mathsf{v}_x, \mathsf{p}) := \begin{cases} (0.445, & 0.698, & 3.528) & \text{für } x < 0.5, \\ (0.5, & 0, & 0.571) & \text{für } x > 0.5. \end{cases}$$

Mit dem zweidimensionalen Euler-Verfahren wurde mit diesen Anfangsbedingungen und der Zusatzforderung $\mathsf{v}_y = 0$ die Lösung zur Zeit $T = 0.1445$ für ein Gebiet mit den Abmessungen $[0,1] \times [-0.05, 0.05]$ berechnet. Die verwendete Triangulierung und das zugehörige Gitter der Zellen ist in Abbildung 2.9 dargestellt. Es entspricht einer horizontalen Auflösung von 100 Zellen. An dem unteren, oberen und rechten Rand wurde die Randbedingung für feste Wände verwendet. Am linken Rand wurden jeweils Riemann-Probleme mit dem linken Zustand der Anfangsbedingung als Außenwert gelöst. Querschnitte der Dichte und der Geschwindigkeit entlang der x-Achse sind in Abbildung 2.11 für die verschiedenen Rekonstruktionsverfahren zusammen mit der exakten Lösung dargestellt. Die Ergebnisse mit dem neuen Limitierungsverfahren und kubischen Polynomen wurden hier allerdings nicht mehr berücksichtigt, da die Querschnitte in mittlerer Höhe denen mit quadratischer Rekonstruktion sehr ähnlich sind und dieses Ergebnis einen falschen

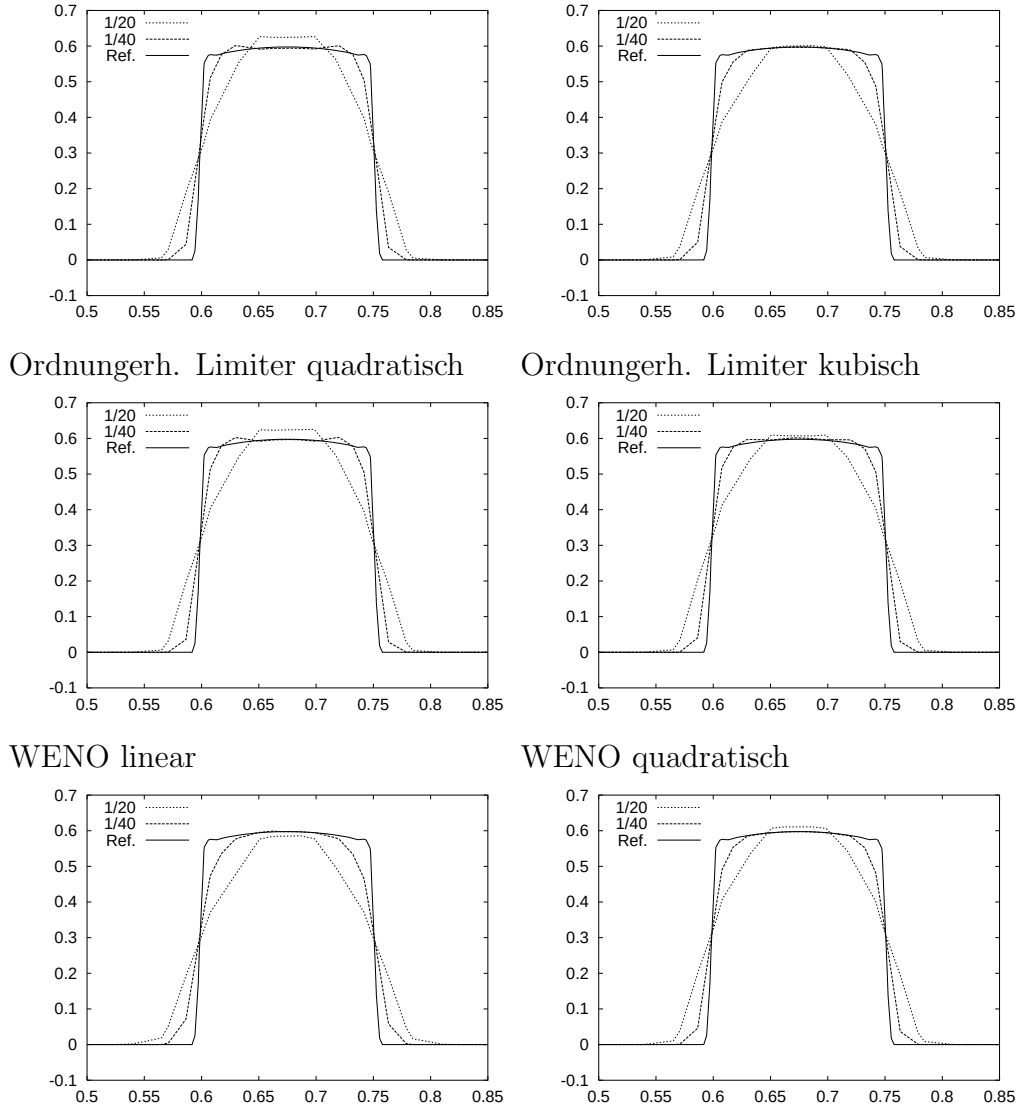


Abbildung 2.8: Querschnitte durch die Lösungen längs der in Abbildung 2.7 gezeigten Schnittlinie. Es sind jeweils die Lösungen für die beiden unterschiedlich feinen Gitter im Vergleich zu einer Referenzlösung, die auf einem deutlich feinerem Gitter berechnet wurde, dargestellt.

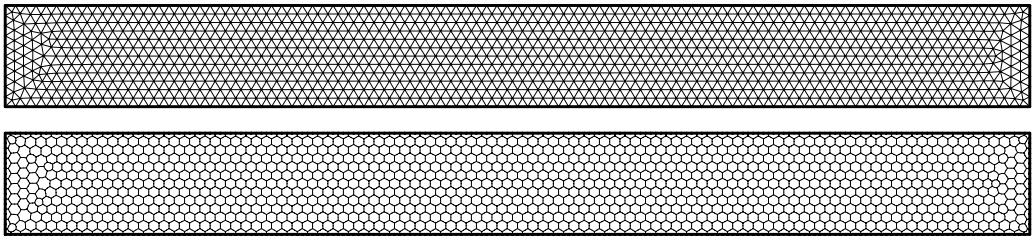


Abbildung 2.9: Triangulierung und resultierendes Gitter aus Kontrollvolumina für die Berechnung des Stoßrohrproblems von Lax. Das Gitter ist identisch mit dem in [Fri97] gezeigten und enthält circa 100 Zellen in horizontaler Richtung.

Eindruck wiederspiegeln würde. Wie in der Abbildung 2.10 zu sehen ist, hat das Verfahren mit kubischen Polynomen erhebliche Probleme in Wandnähe. Hier sind die verwendeten lokalen Gitter einseitig ausgerichtet, und die resultierende Rekonstruktion oszilliert in der Nähe von Unstetigkeiten so stark, daß wegen illegaler Werte (negative Dichte) nachträglich die Rekonstruktion auf das konstante Polynom reduziert werden muß. Dieses ließe sich unter Umständen dadurch vermeiden, daß man in das Rekonstruktions- und Limitierungsverfahren Kenntnisse über Wandwerte einbringt. Dieses wurde bisher in dem beschriebenen Verfahren nicht getan.
Wie Abbildung 2.11 auf Seite 75 zeigt, sind die Ergebnisse der linearen und der quadratischen Rekonstruktionsverfahren miteinander qualitativ vergleichbar. Dies liegt wohl unter anderem daran, daß die exakte Lösung in ihren glatten Bereichen linear ist. Das gewichtete ENO-Verfahren mit linearer Rekonstruktion ist besonders dissipativ. Um so größer ist hier der Gewinn beim Wechsel von linearen zu quadratischen Polynomen. In der Dichteverteilung nimmt die Schärfe beider Verfahren zu. Die Unstetigkeiten werden etwas besser aufgelöst, und die „Überschießer“ nehmen ab. Im Geschwindigkeitsprofil scheint sich dies jedoch zu verkehren; beide Verfahren oszillieren etwas stärker mit quadratischer Rekonstruktion.

Reflexion einer Stoßfront mit zehnfacher Schallgeschwindigkeit

Als weiteres Beispiel wurde das aus [WC84] bekannte Beispiel der Reflexion eines Stoßes, der sich relativ zum ruhenden Medium davor mit zehnfacher Schallgeschwindigkeit (Mach=10) fortbewegt, an einer keilförmig im Winkel von 30° aufgestellten flachen Wand gerechnet. Hierfür sind beispielsweise in [Dyk82] fotografische Aufnahmen aus physikalischen Experimenten zu finden. Das Rechengebiet wurde als Ausschnitt des Rechtecks $[0, 3] \times [0, 0.8]$ gewählt, wobei die Höhe und die Länge der verschiedenen Gitter etwas schwankte, um

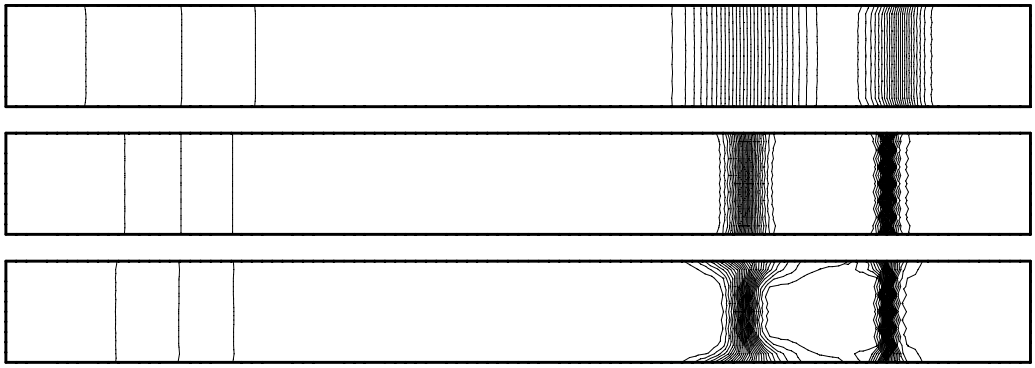


Abbildung 2.10: 30 Höhenlinien der Dichteverteilung für das Stoßrohrproblem von Lax: konstante Rekonstruktion (oben), ordnungserhaltende Limitierung mit quadratischen Polynomen (Mitte), mit kubischer Rekonstruktion (unten). Die kubische Rekonstruktion oszilliert am Rand so stark, daß dort nachträglich die Rekonstruktion wegen illegaler Werte auf die konstante Rekonstruktion reduziert wird.

möglichst gleichseitige Dreiecke für die Triangulierung verwenden zu können. Ein Beispiel für eine der verwendeten Triangulierungen ist auf Seite 108 oben zu sehen. Dort wurde der Bereich durch eine Triangulierung aus gleichseitigen Dreiecken mit der Kantenlänge $h = 1/15$ überdeckt. Für die Testrechnungen in diesem Abschnitt wurde ein entsprechendes Gitter mit der Auflösung $h = 1/100$ hergestellt (ohne Abbildung). Berechnet wurde die Lösung bis zur Zeit $T = 0.2$. Der Stoß wurde zu Beginn ($t = 0$) am unteren Rand bei $x = 1/6$ im Winkel von 60° relativ zur x-Achse positioniert. Die Anfangsdaten wurden entsprechend der Rankine-Hugoniot-Bedingung aus Satz 1.1 wie folgt gewählt:

$$
\begin{aligned}
(\rho, \mathsf{v}, \mathsf{p}) &:= \begin{cases} (8, & 8.25 \cdot \mathsf{n}, & 116.5) & \text{für } \sqrt{0.75}/6 > \mathsf{n} \cdot (x, y)^{\mathsf{t}}, \\ (1.4, & 0, & 1) & \text{sonst}, \end{cases} \\
\mathsf{n} &:= (\sqrt{0.75}, \ -0.5)^{\mathsf{t}}.
\end{aligned}
$$

Am linken, oberen und rechten Rand wurde als Außenwert für die numerischen Flußfunktionen jeweils der gradlinig gleichförmig fortbewegte Anfangszustand verwendet. Am unteren Rand wurde links von $x = 1/6$ ebenfalls dieser Außenzustand verwendet. Die Punkte rechts von 1/6 am unteren Rand wurden als feste Wände behandelt.
Neben den Testrechnungen auf einem Gitter der Maschenweite 1/100 wurde zusätzlich eine Referenzlösung mit dem im letzten Kapitel beschriebenen Adaptionsverfahren erstellt. Die kleinsten Dreiecke der zugehörigen Triangulierung hatten im Bereich der Unstetigkeiten und der Kelvin-Helmholtz-

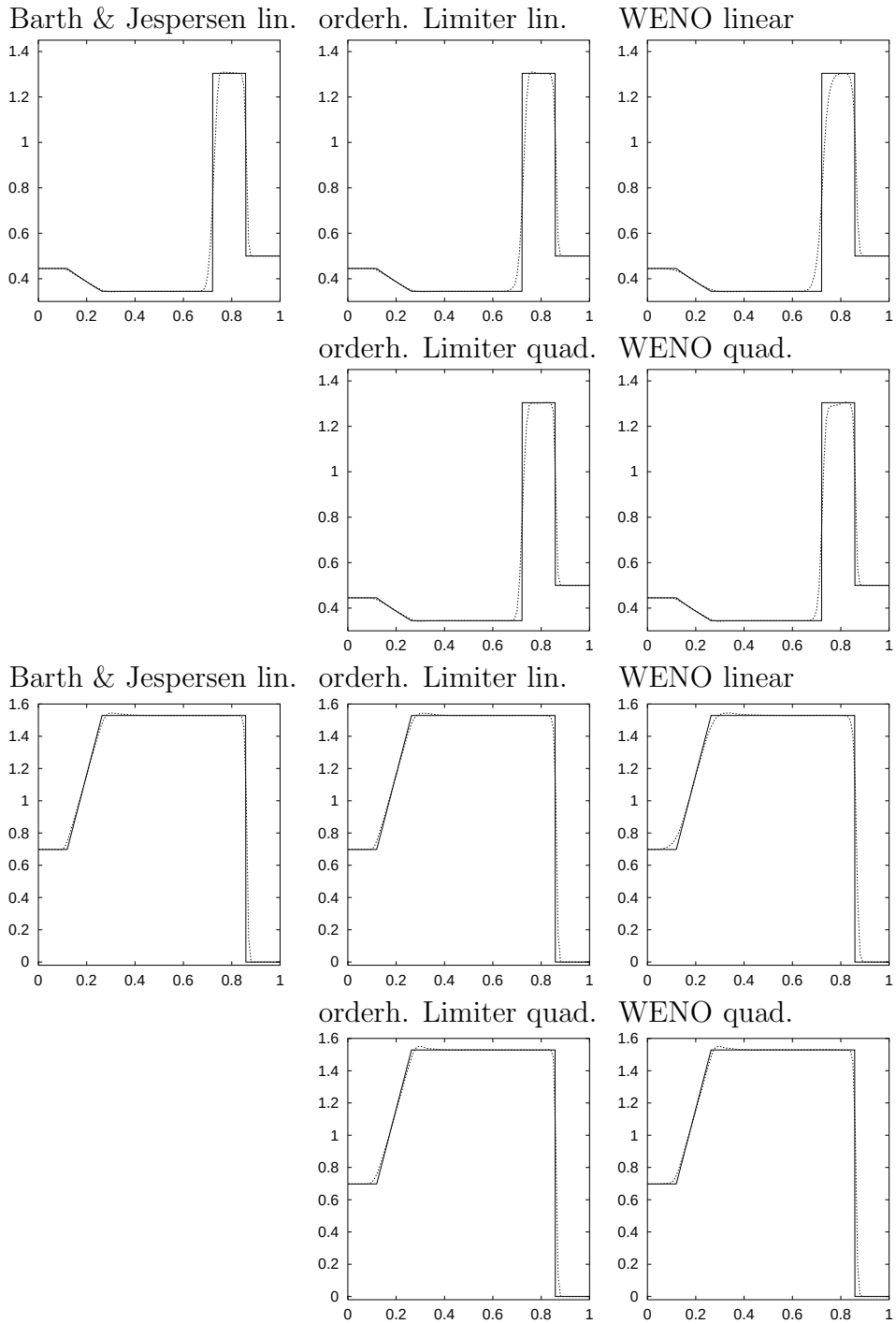


Abbildung 2.11: Schnitte entlang der x-Achse durch die Dichteverteilung (oben) und die Geschwindigkeit in x-Richtung (unten) für das Stoßrohrproblem von Lax.

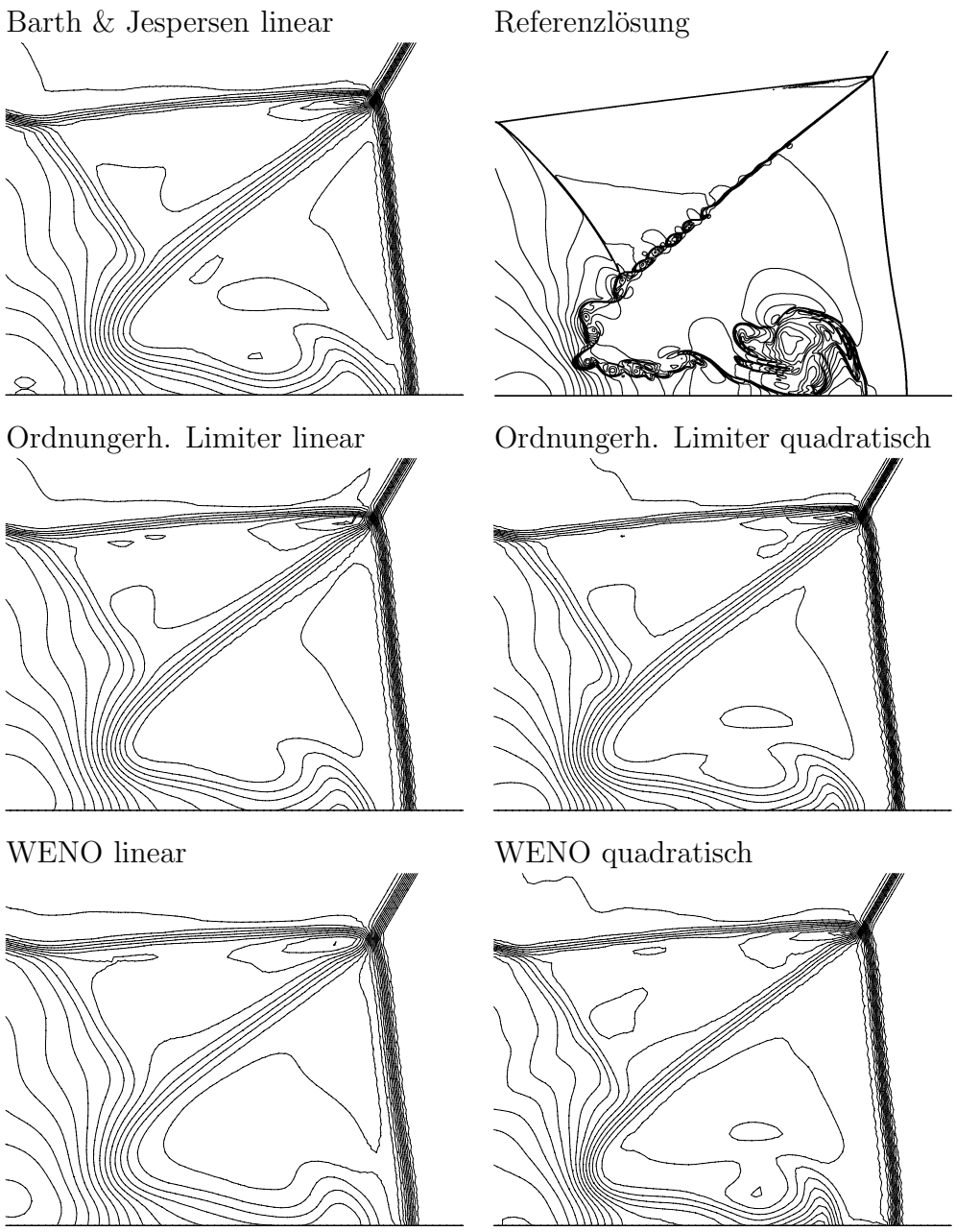


Abbildung 2.12: Ausschnitt der Dichteverteilung der Mach=10 Stoßreflexion an einer festen Wand zur Zeit $T = 0.2$ für ein regelmäßiges Gitter mit typischer Kantenlänge $h = 1/100$. Die Referenzlösung ist auf einem adaptiv mitgeführten Gitter berechnet worden, deren feinste Auflösung bei $h = 1/1920$ lag. Dargestellt sind jeweils 36 Isolinien von 1.39 bis 22.39. Die untere Kante entspricht dem in Abbildung 2.13 gezeigten Intervall.

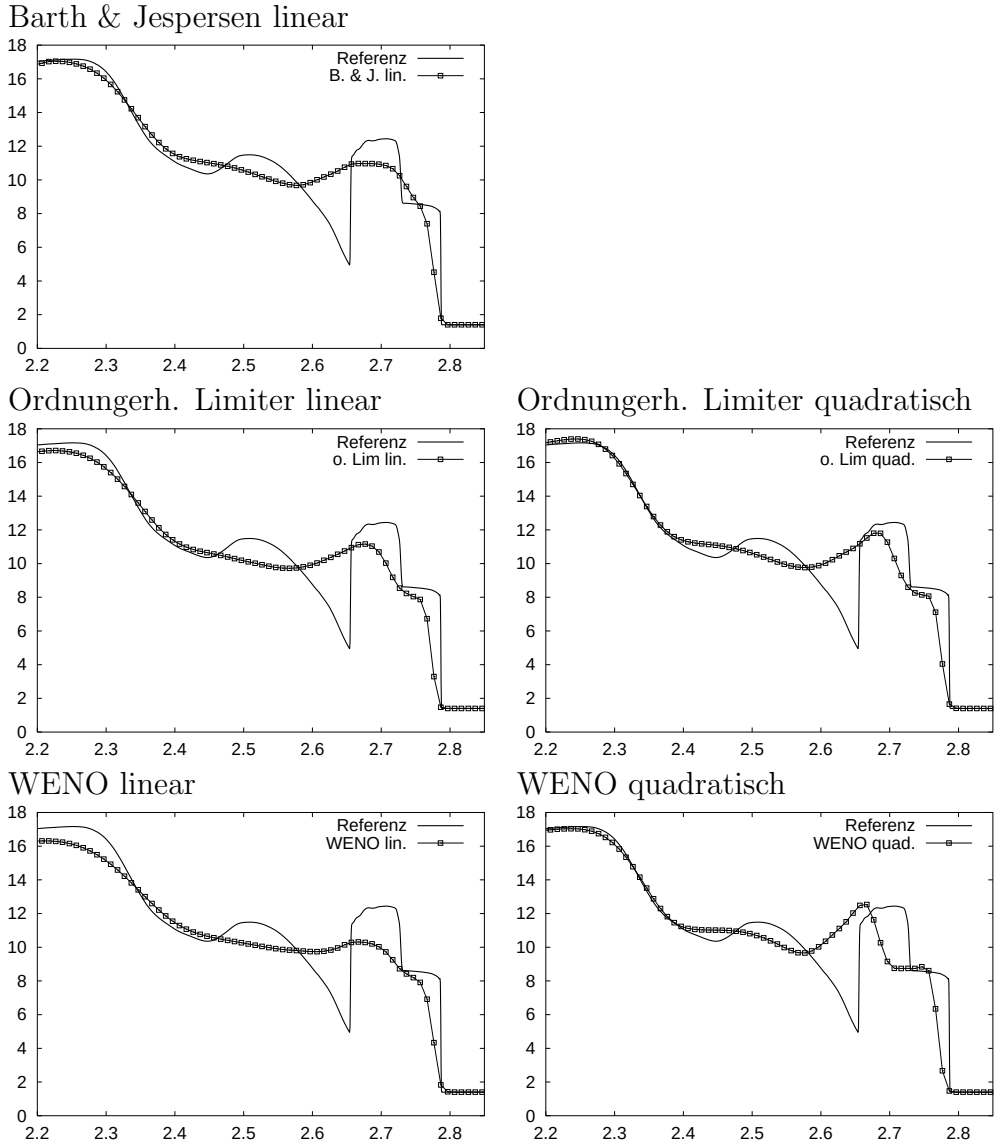


Abbildung 2.13: Dichteverteilung an der unteren Wand für die Mach=10 Stoßreflexion aus Abbildung 2.12. Dargestellt ist das Intervall $[2.2, 2.85]$ relativ zum gesamten Bereich $[0, 3]$ mit der Anfangsposition des Stoßes bei $1/6$. Die Verfahren sind zusammen mit der Referenzlösung gezeigt.

Rekonstruktionverfahren	relative CPU-Zeit
triv. Rekonstruktion (konstant)	1 (Eichwert)
Barth und Jespersen (linear)	1.29
o. Limiter (linear)	2.04
o. Limiter (quadratisch)	2.31
WENO (linear)	1.53
WENO (quadratisch)	3.80

Tabelle 2.1: Dauer eines dreistufigen Runge-Kutta-Zeitschrittes in Abhängigkeit vom Rekonstruktionsverfahren.

Instabilität eine Kantenlänge von 1/1920. Die Ergebnisse dieser Rechnung sind in der Abbildung 4.6 auf Seite 117 zu sehen. Es wurde das ordnungserhaltende Limitierungsverfahren zusammen mit quadratischen Polynomen verwendet.
In der Abbildung 2.12 sind die Lösungen für die unterschiedlichen Rekonstruktionsverfahren auf dem Gitter mit der Maschenweite 1/100 im Vergleich zu der Referenzlösung in einem Ausschnitt dargestellt. Für den Bereich $[2.2, 2.85]$ an der unteren Wand wurde die Dichte ermittelt[7] und in den Funktionsgraphen auf Seite 77 abgebildet. Hier ist zu erkennen, daß die Verfahren mit quadratischer Rekonstruktion deutlich bessere Ergebnisse liefern als diejenigen mit linearer Rekonstruktion. Die Qualitätssteigerung ist auch hier besonders groß, wenn man das lineare und das quadratische WENO-Verfahren vergleicht. Während das lineare WENO-Verfahren eine besonders dissipative Lösungen zeigt, ist dagegen die Lösung des quadratischen WENO-Verfahrens sogar noch etwas strukturreicher als die mit dem ordnungserhaltenden Limiter und quadratischen Polynomen.

Geschwindigkeitsvergleich der Verfahren

Zum Abschluß dieses Kapitels wollen wir noch die Verfahren bezüglich ihres Rechenzeitbedarfs vergleichen. Dieser Vergleich ist allerdings unter Vorbehalt zu betrachten, da die Verfahren doch sehr unterschiedlich sind und unter Umständen noch Verbesserungen in Einzelfällen möglich sind. Beispielsweise wurde im WENO-Verfahren in der verwendeten Implementierung bisher auf eine gestaffelte Berechnung der linearen Gleichungssysteme verzichtet, wie sie in [AS97] mittels Mühlbach-Reihen [Müh78] vorgeschlagen wird. Eine andere Möglichkeit der Beschleunigung des WENO-Verfahrens besteht darin, die für ein lokales Gitter berechnete Rekonstruktion simultan für al-

[7] Die Wandwerte wurden durch lineare Interpolation der Zellmitteldaten bestimmt.

le Zellen zu verwenden, deren Zellmittel interpoliert werden. Hierzu muß man entweder die rekonstruierten Polynome in einer globalen Basis darstellen oder den Algorithmus zur Verschiebung des Ursprunges eines Polynoms von Seite 52 verwenden. Durch diesen Trick der Mehrfachverwertung von berechneten Rekonstruktionen könnte unter Umständen der Rechenzeitbedarf des WENO-Verfahrens mit quadratischen Polynomen deutlich verringert werden. Der Vergleich der Verfahren kann daher nur eine Orientierung über den bisherigen Stand der jeweils verwendeten Implementierung bieten, ist aber in keinem Fall ein geeignetes Ausschlußkriterium für einen der Rekonstruktionsansätze.
In den Vergleichstests wurde für das Beispiel aus dem vorangehenden Abschnitt auf einem feinen Gitter mit dem dreistufigen Runge-Kutta-Verfahren und zwei Gauß-Quadraturpunkten pro Kante zunächst die Dauer eines Zeitschrittes für die triviale, stückweise konstante Rekonstruktion ermittelt. Die sich hieraus ergebende durchschnittliche CPU-Zeit für einen vollständigen Zeitschritt wurde mit 1 bewertet. Dies ist im wesentlichen die Dauer für das Berechnen der numerischen Flüsse (Osher/Solomon). Für die Rekonstruktionsverfahren ergeben sich dann die in der Tabelle 2.1 angebenen relativen Zeiten. Die Ergebnisse sind bei serieller (nicht paralleler) Bearbeitung ermittelt worden. Bei der im letzten Kapitel beschriebenen parallelen Implementierung verbessern sich die relativen Zeiten für die aufwendigeren Verfahren mit quadratischer Rekonstruktion.

Kapitel 3

Optimale Rekonstruktion in Semi-Hilberträumen

In der Diskussion um lokale Ausgleichsprobleme ab Seite 42 haben wir die Minimierungsaufgabe (2.4) unter der Nebenbedingung (2.5) betrachtet. Die Wahl geeigneter Gewichte g_η für die einzelnen Zellen η des lokalen Gitters $\overline{\mathcal{L}}$ wurde nur oberflächlich behandelt. In der Gleichung (2.9) wurden lediglich die Gewichte angegeben, die in dem implementierten Finite-Volumen-Verfahren verwendet wurden. Wir wollen in diesem Abschnitt die Wahl der Minimierungsgewichte detaillierter beleuchten, indem wir Lösungen von Ausgleichsproblemen als optimale Rekonstruktionen im Sinne von [GW59] und [MR84] herausarbeiten.
Von einem gegebenen Rekonstruktionsverfahren wird man im Zusammenhang der Erhaltungsgleichungen, wie sie im ersten Kapitel eingeführt wurden, erwarten dürfen, daß sie unter Ähnlichkeitsabbildungen invariant sind. Damit ist gemeint, daß unter Ähnlichkeitstransformationen $\mathcal{A}$ des $\mathbf{R}^d$ sich die Lösung $u(x,t)$ entsprechend mittransformiert: $u(\mathcal{A}^{-1}x',t)$. Diese Eigenschaft erfordert unter anderem die Rotationsinvarianz des Rekonstruktionsverfahrens, und daher wird man insbesondere für die Optimierungsaufgabe des Rekonstruktionsproblems eine rotationsinvariante Bilinearform verwenden. Unter dieser Voraussetzung stellen sich die radialen Basisfunktionen als explizite Lösungen der optimalen Rekonstruktion heraus.
Der Zusammenhang der Theorie optimaler Rekonstruktionsverfahren mit den ENO-Verfahren wird erstmalig in [Son97a] in Theorie, aber auch in numerischen Anwendungen untersucht und in [IS96] weiterverfolgt. Für ein numerisches Verfahren zu Lösung der Euler-Gleichungen auf kartesischen Gittern wird in [Gut98] der sogenannte Thin-Plate-Spline verwendet. Für diese radiale Funktion wird in [Son96] ein ENO-Verfahren für unstrukturierte Gitter vorgestellt.

Bevor wir zur Theorie der optimalen Rekonstruktion in Semi-Hilberträumen gelangen, wollen wir im nächsten Abschnitt das eingangs angesprochene Ausgleichsproblem in geeigneter Weise umformulieren.

Das Ausgleichsproblem (Fortsetzung)

Die Minimierungsaufgabe (2.4) unter der Nebenbedingung (2.5) wollen wir in eine Lagrange-Darstellung überführen. Wir wollen allerdings hier nicht die Basis im Polynomraum wechseln, um das Zellmittel in der Zelle ω leichter interpolieren zu können und führen daher die folgenden neuen Matrizen ein:

$$\begin{aligned} G'_{\eta,\ell} &:= \begin{cases} 0 & \text{für } \omega = \eta = \ell \text{ oder } \eta \neq \ell, \\ 1/g_\eta^2 & \text{sonst.} \end{cases} \\ B_{\eta,i} &:= \delta_\eta(\pi_i) \quad \eta \in \overline{\mathcal{L}}, \quad i \in \{0, \ldots, Q-1\}. \end{aligned}$$

Die Matrix G' ist eine Diagonalmatrix mit einem 0-Eintrag in der Zeile für die Zelle ω. Streicht man diese Zeile und diese Spalte, dann erhält man gerade die Inverse von $G^{\mathsf{t}}G$ aus Gleichung (2.7). Die Matrix B ist ähnlich wie die Massenmatrix A aus (2.8) durch Anwendung der verschiedenen linearen Funktionale auf die Basisfunktionen entstanden. Allerdings verwenden wir jetzt die vollständige Basis des Polynomraumes Π^q und alle linearen Funktionale einschließlich δ_ω. Die Matrix A entsteht aus der Matrix B, indem man einen Schritt des Gaußschen-Eliminationsverfahrens mit dem Pivotelement $B_{\omega,0}$ durchführt und anschließend die Zeile ω und die Spalte 0 streicht. Mit den Bezeichnungen gilt nun der folgende Satz:

Satz 3.1 *Das Ausgleichsproblem (2.4) unter der Nebenbedingung (2.5) besitzt die äquivalente Lagrange-Darstellung als lineares Gleichungssystem*

$$\begin{pmatrix} G' & B \\ B^{\mathsf{t}} & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix}, \quad \wp = \sum_{i=0}^{Q-1} \beta_i \pi_i. \tag{3.1}$$

Beweis Die ω-Zeile der Matrix G' ist eine Nullzeile. Daher entspricht diese Zeile des Gleichungssystems (3.1) der Nebenbedingung (2.5). Nachdem man in der Matrix G' die Zeile und Spalte ω gestrichen hat, erhält man die Inverse von $G^{\mathsf{t}}G$. Wir streichen ferner die Zeile ω in der Matrix B, den Vektoren α und d und nennen das Resultat $\bar{B}$, $\bar{\alpha}$ beziehungsweise $\bar{d}$. Dann vergewissern wir uns, daß die Darstellung (3.1) gleichwertig zu der folgenden ist:

$$\begin{aligned} \begin{pmatrix} (G^{\mathsf{t}}G)^{-1} & \bar{B} \\ \bar{B}^{\mathsf{t}} & 0 \end{pmatrix} \begin{pmatrix} \bar{\alpha} \\ \beta \end{pmatrix} &= \begin{pmatrix} \bar{d} \\ 0 \end{pmatrix}, \\ \alpha_\omega &= 0. \end{aligned}$$

Nach Auflösung der oberen Zeilen nach $\bar{\alpha}$ sieht man, daß der Vektor $\bar{\alpha}$ die gewichteten Residuen des Ausgleichsproblems enthält:

$$\bar{\alpha} = G^{\mathsf{t}} G(\bar{d} - \bar{B}\beta). \tag{3.2}$$

Diese multiplizieren wir von links mit der Matrix $\bar{B}^{\mathsf{t}}$ und erhalten nach Einsetzen von $\bar{B}^{\mathsf{t}}\bar{\alpha} = 0$ die Gleichung

$$\bar{B}^{\mathsf{t}} G^{\mathsf{t}} G \bar{B}\beta = \bar{B}^{\mathsf{t}} G^{\mathsf{t}} G d$$

Dies ist die zu (2.4) gehörende Normalengleichung, und damit haben wir gezeigt, daß eine Lösung von (3.1) auch eine Lösung von (2.4) mit der Nebenbedingung (2.5) ist. Die Umkehrrichtung erhält man aus den Normalengleichungen, indem man α wie in (3.2) definiert. ∎

Aus der Lösung des Gleichungssystems (3.1) erhalten wir in der beschriebenen Weise das Polynom $\wp$. Dieses interpoliert normalerweise nur das Zellmittel der Zelle ω, nicht aber die Zellmittel $(d_\eta)_{\eta\in\mathcal{L}}$. Allerdings können wir die Lösung $(\alpha, \beta)^{\mathsf{t}}$ sehr schnell in geeigneter Weise zu einer Interpolanten machen. Hierzu benötigen wir zusätzlich die charakteristischen Funktionen χ_η der Zellen $\eta \in \overline{\mathcal{L}}$. Diese Funktionen skalieren wir mit den Gewichten g_η, wobei wir aus Gründen der Eleganz annehmen, daß das Gewicht $g_\omega := \infty$ ist und die Division $(1/g_\omega^2) := 0$ verschwindet. Dann betrachten wir die skalierten charakteristischen Funktionen

$$\Phi_\eta := \frac{1}{g_\eta^2}\chi_\eta, \quad \text{für } \eta \in \overline{\mathcal{L}}. \tag{3.3}$$

Hiermit können wir eine unstetige Interpolante der Daten d angeben, die in jeder Zelle aus einem Polynom besteht:

$$v := \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta \Phi_\eta + \sum_{i=0}^{Q-1} \beta_i \pi_i.$$

Daß dies eine Interpolante ist, sieht man mittels (3.2). Innerhalb der Zelle ω sind die Funktionen $\wp$ und v identisch.

Interpretation *Wir haben eine globale, stückweise polynomielle Interpolante angegeben, indem wir den Polynomraum mit charakteristischen Funktionen angereichert haben. Mit der Gewichtsmatrix G' können wir steuern, welchen Anteil bei der Interpolation eines Zellmittels die zugehörige charakteristische Funktion haben soll. Ein Nulleintrag bedeutet, daß bereits die*

Polynome die Interpolationsaufgabe erfüllen sollen. Dagegen bedeutet ein hoher Wert, daß der Zellmittelwert mit einem großen Anteil der zugehörigen charakteristischen Funktion interpoliert wird.
Die Gewichtsmatrix G', welche die zu minimierende quadratische Form für die diskreten Daten festlegt, wird man ganz allgemein als Inverse einer Gramschen Matrix zu einer quadratischen Form konstruieren. Solche quadratischen Formen korrespondieren in reellen Vektorräumen zu symmetrischen Bilinearformen. Da die quadratischen Formen nichtnegativ sein werden, wird die zugehörige symmetrische Bilinearform zusätzlich positiv semidefinit sein und damit ein **Semiskalarprodukt**. Die Wahl der zusätzlichen Basisfunktionen Φ_η, mit denen die Interpolation erreicht werden kann, hängt einerseits von den Funktionalen δ_η ab und andererseits von dem verwendeten Semiskalarprodukt.
Der Zusammenhang wird durch die Theorie der optimalen Rekonstruktion in Semi-Hilberträumen verständlich, wie sie in [GW59] und [MR84] untersucht wurde. Wir fassen diese Ergebnisse in angepaßter Form im nächsten Abschnitt zusammen.

Optimale Rekonstruktion

Die linearen Funktionale δ_η für $\eta \in \overline{\mathcal{L}}$ stellen wir wie in Formel (2.3) auf Seite 42 zu einer linearen Abbildung $\mathcal{D}$ zusammen. Wir verzichten hier allerdings auf die Indizierung durch das lokale Gitter $\overline{\mathcal{L}}$. Weiterhin schränken wir den Operator $\mathcal{D}$ nicht auf die q-mal stetig differenzierbaren Funktionen ein, sondern verstehen ihn als stetige, lineare Abbildung eines Teilraumes V von L^1. Dann sind durch die Interpolationsbedingung

$$\mathcal{D}(u) = d \tag{3.4}$$

endlich viele lineare Nebenbedingungen gegeben. Wir gehen im folgenden davon aus, daß $\mathcal{D}$ eine surjektive Abbildung auf die Menge $\mathbf{R}^{\overline{\mathcal{L}}}$ ist und daher die Kodimension des affinen Lösungsraumes von (3.4) den endlichen Wert $M := \mathrm{card}(\overline{\mathcal{L}})$ hat. Der Raum V sei mit einer nichtnegativen, quadratischen Form ausgestattet, die wir als Semiskalarprodukt $(\,\cdot\,,\,\cdot\,)$ schreiben wollen. Das Radikal dieser Bilinearform bezeichnen wir mit

$$N := \{v \in V \,:\, (v\,, v) = 0\}. \tag{3.5}$$

Eine Funktion in V bezeichnen wir als **optimale Rekonstruktion** der Daten d, wenn sie die Lösung des folgenden Minimierungsproblems ist:

> **Minimierungsaufgabe**: Finde ein Element v mit $\mathcal{D}v = d$, so daß $(v, v) \geq 0$ minimal wird.

Die Lösung ist höchstens dann eindeutig, wenn der Schnitt $\mathrm{Kern}(\mathcal{D}) \cap N = \{0\}$ trivial ist. Dies folgt unter Zuhilfenahme der folgenden Aussage:

Lemma 3.2 *Für das Semiskalarprodukt* $(\,\cdot\,,\,\cdot\,) : V \times V \to \mathbf{R}$ *gilt für alle* $v_N \in N$ *und alle* $v \in V$ *die Gleichung* $(v\,, v_N) = 0$.

Beweis Sei $v_N \in N$ und $v \in V$. Dann gilt für alle $\lambda \in \mathbf{R}$:

$$0 \;\leq\; (v + \frac{\lambda}{2} v_N\,, v + \frac{\lambda}{2} v_N) \;=\; (v\,, v) + \lambda \cdot (v\,, v_N).$$

Der Grenzübergang $\lambda \longrightarrow \pm\infty$ beweist $(v\,, v_N) = 0$. ∎

Wir werden die Lösungen der Minimierungsaufgabe zuerst in der folgenden Weise charakterisieren:

Lemma 3.3 *Die Minimierungsaufgabe der optimalen Rekonstruktion wird genau dann von* v *mit* $\mathcal{D}v = d$ *gelöst, wenn*

$$(v\,, w) \;=\; 0 \qquad\qquad \forall w \in \mathrm{Kern}(\mathcal{D})$$

gilt.

Beweisskizze

$$0 \;=\; \frac{\mathsf{d}}{\mathsf{d}\epsilon}(v + \epsilon w\,, v + \epsilon w)\bigg|_{\epsilon=0} \;=\; 2 \cdot (v\,, w) \;=\; (v\,, w).$$

∎

Wir wissen nun, daß aus der Eindeutigkeit der Lösung die Beziehung $\mathrm{Kern}(\mathcal{D}) \cap N = \{0\}$ folgt. Wir wollen noch die Umkehrung untersuchen: Angenommen v und v' sind zwei Lösungen. Dann folgt $v - v' \in \mathrm{Kern}(\mathcal{D})$ und wegen

$$(v - v'\,, v - v') \;=\; \underbrace{(v\,, v - v')}_{\text{Lemma 3.3}} - \underbrace{(v'\,, v - v')}_{\text{Lemma 3.3}} \;=\; 0$$

folgt auch noch $v - v' \in N$ und damit $\mathrm{Kern}(\mathcal{D}) \cap N \neq \{0\}$. Damit haben wir die Eindeutigkeitsaussage des nachfolgenden Lemmas bewiesen.

Lemma 3.4 *Das Minimierungsproblem der optimalen Rekonstruktion ist eindeutig lösbar, sofern* $\mathrm{Kern}(\mathcal{D}) \cap N = \{0\}$ *ist und der Raum* V/N *mit dem durch* $(\,\cdot\,,\,\cdot\,)$ *induzierten Skalarprodukt ein Hilbertraum ist.*

Will man auf die Verwendung von Restklassenräumen verzichten, dann kann man auch alternativ die folgende Formulierung von Lemma 3.4 verwenden:

Korollar 3.5 *Das Minimierungsproblem der optimalen Rekonstruktion ist eindeutig lösbar, wenn* $\mathrm{Kern}(\mathcal{D}) \cap N = \{0\}$ *ist und bezüglich einer direkten Summenzerlegung* $H \oplus N = V$ *der Raum* H *ein Hilbertraum mit dem eingeschränkten Skalarprodukt* $(\,\cdot\,,\,\cdot\,)|_{H\times H}$ *ist.*

Wir führen den Beweis für das Korollar 3.5.

Beweis Die Eindeutigkeitsaussage ist bereits bewiesen. Sei jetzt eine direkte Summenzerlegung $H \oplus N = V$ mit dem Hilbertraum H gegeben. Dieser Hilbertraum kann unter der Voraussetzung $\mathrm{Kern}(\mathcal{D}) \cap N = \{0\}$ so gewählt werden, daß er $\mathrm{Kern}(\mathcal{D})$ als Teilmenge enthält. Sei v_0 ein Element, welches $\mathcal{D}v_0 = d$ erfüllt ($\mathcal{D}$ ist surjektiv). Dann ist der Lösungsraum $v_0 + \mathrm{Kern}(\mathcal{D})$ ein vollständiger Unterraum des verschobenen Hilbertraumes $v_0 + H$, und damit existiert ein Element in $v_0 + \mathrm{Kern}(\mathcal{D})$, welches das strikt konvexe Funktional $(\,\cdot\,,\,\cdot\,)$ minimiert (siehe auch [Heu92, Satz 22.1]). ∎

Aus $\mathrm{Kern}(\mathcal{D}) \cap N = \{0\}$ und der endlichen Kodimension von $\mathrm{Kern}(\mathcal{D})$ folgt, daß das Radikal endlichdimensional ist. Sei im folgenden $p_1, \ldots, p_{\dim N}$ eine Basis von N. Ist $V = H \oplus N$ eine direkte Summenzerlegung mit dem Hilbertraum H, dann existieren Riesz-Darstellungen r_η der auf H eingeschränkten Funktionale $\delta_\eta|_H$, $\eta \in \overline{\mathcal{L}}$. Unter diesen Voraussetzungen und Bezeichnungen ermöglicht der nachfolgende Satz, die Lösung des Minimierungsproblems durch Lösung eines endlichen Gleichungssystems zu berechnen:

Satz 3.6 *Sei* $\mathrm{Kern}(\mathcal{D}) \cap N = \{0\}$ *und* $p_1, \ldots, p_{\dim N}$ *eine Basis von* N. *Sei ferner* $V = H \oplus N$ *mit einem Hilbertraum* H. r_η, $\eta \in \overline{\mathcal{L}}$, *seien die Riesz-Darstellungen von* $\delta_\eta|_H$ *in* H. *Dann läßt sich die Lösung* v *des Minimierungsproblems in der Form*

$$v = \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta r_\eta + \sum_{j=1}^{\dim N} \beta_j p_j \tag{3.6}$$

darstellen, und die Koeffizienten α *und* β *ergeben sich als eindeutige Lösung des linearen Gleichungssystems*

$$\mathcal{D}v = d, \tag{3.7}$$

$$\sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta \delta_\eta|_N = 0. \tag{3.8}$$

Beweis Sei zuerst v wie in (3.6) die Lösung des linearen Gleichungssystems (3.7) und (3.8). Für $w \in \mathrm{Kern}(\mathcal{D})$ gibt es eine Zerlegung $w = w_H + w_N$ mit $w_H \in H$ und $w_N \in N$. Damit folgt:

$$\begin{aligned}(v\,,w) &= \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta (r_\eta\,,w) + 0 \;=\; \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta (r_\eta\,,w_H) \;=\; \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta \delta_\eta(w_H) \\ &= \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta \underbrace{\delta_\eta(w)}_{w\in\mathrm{Kern}(\mathcal{D})} - \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta \delta_\eta(w_N) \;=\; \underbrace{-\sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta \delta_\eta(w_N)}_{(3.8)} \;=\; 0\end{aligned}$$

Nach Lemma 3.3 ist v wegen (3.7) die Lösung des Minimierungsproblems. Da nach den Voraussetzungen die Lösung des Minimierungsproblems existiert und eindeutig ist, reicht für den Beweis der Rückrichtung, nachzuweisen, daß es eine Lösung des Gleichungssystems (3.7), (3.8) mit der Darstellung (3.6) gibt. Insgesamt gibt es $\mathrm{card}(\overline{\mathcal{L}}) + \dim(N)$ Unbekannte α, β. Durch (3.7) und (3.8) liegen genauso viele lineare Gleichungen vor. Damit genügt es zu zeigen, daß das homogene System mit $d = 0$ nur die triviale Lösung $\alpha = 0$, $\beta = 0$ hat. Gäbe es hier mehrere Lösungen, so würden all diese Lösungen nach den ersten Überlegungen dieses Beweises unterschiedliche Lösungen des Minimierungsproblems definieren, denn die Vektoren r_η und p_j sind linear unabhängig. Da nach Korollar 3.5 auch für $d = 0$ nur eine Lösung existiert, erhalten wir einen Widerspruch. Damit ist das Gleichungssystem (3.7) und (3.8) für Funktionen der Form (3.6) eindeutig lösbar. ∎

Die wesentliche Schlußfolgerung des letzten Satzes ist, daß man sich, sofern man die Riesz-Darstellungen der linearen Funktionale δ_η sowie eine Basis des Radikals N kennt, bei der Suche der Lösung

$$v \;=\; \sum_{\eta\in\overline{\mathcal{L}}} \alpha_\eta r_\eta + \sum_{j=1}^{\dim N} \beta_j p_j \tag{3.9}$$

des Minimierungsproblems von Seite 84 auf die Lösung des linearen Gleichungssystems

$$\begin{pmatrix} C & B \\ B^{\mathsf{t}} & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \;=\; \begin{pmatrix} d \\ 0 \end{pmatrix} \tag{3.10}$$

mit

$$\begin{aligned} C_{\eta,\ell} &:= \delta_\eta(r_\ell) \quad \eta,\ell \in \overline{\mathcal{L}}, \\ B_{\eta,j} &:= \delta_\eta(p_j) \quad \eta \in \overline{\mathcal{L}},\ j \in \{1,\ldots,\dim N\} \end{aligned} \tag{3.11}$$

beschränken kann.

Reellwertige Funktionen

Sind die Riesz-Darstellungen r_η der linearen Funktionale δ_η und eine Basis des Radikals N bekannt, so kann die optimale Rekonstruktion, die Lösung der Minimierungsaufgabe von Seite 84, durch Lösen des linearen Gleichungssystems (3.10) bestimmt werden. Wir wollen jetzt den Fall reellwertiger Funktionenräume studieren. Als einfache Folgerung des Rieszschen Darstellungssatzes erhalten wir:

Satz & Definition 3.7 *Sei $H \subseteq \mathbf{R}^X$ ein Hilbertraum reellwertiger Funktionen über einer Menge X. Sei δ_x die Linearform, die das Auswerten einer Funktion an der Stelle $x \in X$ bedeute. Ist dieses Funktional für alle $x \in X$ stetig, dann existiert eine Funktion $\Phi : X \times X \to \mathbf{R}$ mit $\forall x \in X : (y \mapsto \Phi(x, y)) \in H^\star$ und der Eigenschaft: Zu jeder Linearform $\lambda \in H^\star$ kann die Riesz-Darstellung $r \in H$ durch*

$$r(\cdot) \;:=\; \lambda(\underbrace{y \mapsto \Phi(\cdot\,,\, y)}_{\in H})$$

*berechnet werden. Die Funktion Φ heißt **reproduzierender Kern** von $(\cdot\,,\cdot)$ in H. Der reproduzierende Kern ist symmetrisch in beiden Argumenten.*

Beweis Wir betrachten die Riesz-Isometrie $G : H \to H^\star$ mit $\forall h \in H : G(h) := (h\,,\,\cdot\,)$. G besitzt nach dem Rieszschen Darstellungssatz eine stetige Inverse $G^{-1} : H^\star \to H$.
Nach Voraussetzung sind für beliebige $x,\, y \in X$ die Funktionale $\delta_x,\, \delta_y \in H^\star$. Wähle nun

$$\Phi(x,\, y) \;:=\; (\delta_x\,,\delta_y)_{H^\star} \;=\; (G^{-1}\delta_x)(y)$$

wobei $(\cdot\,,\cdot)_{H^\star}$ das Skalarprodukt des Duals $H^\star$ sei. Damit folgt die Symmetrie bereits aus der Symmetrie des Skalarproduktes. G^{-1} ist selbstadjungiert, und damit folgt

$$\lambda(y \mapsto \Phi(x,\, y)) \;=\; \lambda(G^{-1}\delta_x) \;=\; \delta_x(G^{-1}\lambda) \;=\; (G^{-1}\lambda)(x).$$

Der Ausdruck auf der rechten Seite ist gerade die Auswertung der Riesz-Darstellung von λ an der Stelle x. ∎

Besonders interessant wird die Theorie der optimalen Rekonstruktion, wenn das Semiskalarprodukt $(\cdot\,,\cdot)$ zusätzliche Invarianzeigenschaften besitzt. Diese übertragen sich dann auf den reproduzierenden Kern:

Lemma 3.8 *Es gelten die Bezeichnungen des Satzes 3.7. Ist $T : H \to H$ eine stetige und stetig invertierbare Transformation mit der folgenden Eigenschaft: Für alle $v, w \in H$ gilt $(v\,, w) = (Tv\,, Tw)$. Dann gilt im Dualraum $H^\star$ entsprechend: Für alle $\lambda, \mu \in H^\star$ gilt $(\lambda\,, \mu)_{H^\star} = (\lambda \circ T^{-1}\,, \mu \circ T^{-1})_{H^\star}$.*

Beweis Man setze die Definition des dualen Skalarproduktes ein (Produkt der Riesz-Darstellungen). ∎

Die Invarianten T des Funktionenraumes sind typischerweise geeignete Transformationen der Urbildmenge X. Ist beispielsweise X ein euklidischer Raum, so kann man die Invarianz des Semiskalarproduktes $(\,\cdot\,,\,\cdot\,)$ unter Translationen und Rotationen untersuchen. Mit Lemma 3.8 ist der nachfolgende Satz ohne Schwierigkeiten zu beweisen:

Satz & Definition 3.9 *Wir verwenden die Bezeichnungen und Voraussetzungen von Satz 3.7. X sei ein euklidischer Raum. Ist die Bilinearform $(\cdot\,, \cdot)$ wohldefiniert und invariant unter Translationen und orthogonalen Abbildungen des Raumes X, dann ist auch der reproduzierende Kern invariant unter diesen Transformationen, und es existiert eine Darstellung in der Form einer* ***radialen Basisfunktion****:*

$$\phi(\|x - y\|_X) \;\; := \;\; \Phi(x,\, y).$$

Hat man eine radiale Basisfunktion $\phi : \mathbf{R}_{\geq 0} \to \mathbf{R}$ zu einem Semiskalarprodukt gegeben, dann kann man die Lösung zusammen mit einer Basis $p_1, \ldots, p_{\dim N}$ des Radikals durch Lösung des endlichen Gleichungssystems (3.10) mit den Riesz-Darstellungen

$$r_\eta(x) = \delta_\eta(y \mapsto \phi(\|x - y\|_X)) \tag{3.12}$$

berechnen. Typischerweise besteht das Radikal aus einem Polynomraum Π^q. In diesem Fall wird das Rekonstruktionsverfahren polynomreproduzierend sein, und es läßt sich die Konvergenzaussage des Projektionssatzes 2.1 in geeigneten Fällen übertragen.
Alternativ zur Minimierung vorgegebener quadratischer Formen kann man natürlich auch von einer Funktion ϕ ausgehen und entsprechend Gleichung (3.12) verwenden, um Funktionen r_η zu konstruieren. Zusätzlich wird man eventuell noch das Radikal $N := \Pi^q$ als Polynomraum mit der Basis $p_1 := \pi_0, \ldots, p_Q := \pi_{Q-1}$ hinzunehmen. Hiermit wird man das Gleichungssystem (3.10) aufstellen, um Interpolanten der Form (3.9) zu konstruieren. Bei dieser Vorgehensweise wird man überprüfen müssen, unter welchen Voraussetzungen das Gleichungssystem lösbar ist.

Thin-Plate-Splines als Beispiel

Ein wichtiges Beispiel für einen Funktionenraum mit Semiskalarprodukt ist der Beppo-Levi-Raum der Ordnung m:

$$\mathrm{BL}^m(\mathbf{R}^d \to \mathbf{R}) := \{v \in \mathcal{D}'(\mathbf{R}^d \to \mathbf{R}) \,:\, \partial^\alpha v \in L^2(\mathbf{R}^d \to \mathbf{R}) \text{ mit } |\alpha| = m\}$$

Hierbei ist mit $\mathcal{D}'$ der Raum der Distributionen bezeichnet und entsprechend ist die Ableitung zum Multiindex α in diesem Sinne zu verstehen. In diesem Funktionenraum betrachtet man die Summe der L^2-Skalarprodukte der Ableitungen von der Ordnung m:

$$(v\,,w)_{\mathrm{BL}^m} := \sum_{|\alpha|=m} \frac{m!}{\alpha_1! \cdots \alpha_d!} \int\limits_{\mathbf{R}^d} \partial^\alpha v \partial^\alpha w \; \mathsf{d}x.$$

Das Radikal dieses Semiskalarproduktes sind gerade die Polynome vom Grad $m-1$. Wir zitieren aus [Son96] den folgenden wichtigen Satz von [Mei79]:

Satz 3.10 *Im Falle $m > d/2$ besitzt der Raum BL^m einen reproduzierenden Kern, weil die Punktauswertung δ_x stetig ist (siehe Satz 3.7).*

Für gerade Raumdimension d erhält man als zugehörige radiale Basisfunktion

$$\phi(r) \doteq r^{2m-d} \log(r),$$

und für ungerade Raumdimension ergibt sich als radiale Basisfunktion

$$\phi(r) \doteq r^{2m-d}.$$

Bei der Angabe von ϕ haben wir konstante Faktoren fortgelassen, da sie keinen Einfluß auf die Lösung haben. Die Auswertung wird bei numerischen Verfahren üblicherweise in der dargestellten Variante durchgeführt. Für eine vollständige Angabe des reproduzierenden Kerns sei auf [Son96] verwiesen. Interpolanten, die sich aus den genannten radialen Basisfunktionen zusammen mit dem Radikal Π^{m-1} ergeben, nennt man Thin-Plate-Splines. Sie stellen die natürliche Verallgemeinerung der eindimensionalen Splines dar. In [SW93] und [Pow94] findet man als Konvergenzresultat für die Thin-Plate-Splines die lokale Fehlerordnung $\mathcal{O}(h^{m-d/2})$ für die Interpolation von Punktwerten. Numerische Experimente zeigen, daß mindestens in der Nähe von Rändern diese Abschätzung scharf ist; siehe beispielsweise [Gut98].

Raumdimension	C^m			
$d = 1$	$m = 0$	$\Psi_{1,0}(r)$	$=$	$(1-r)_+$
$d = 1$	$m = 2$	$\Psi_{2,1}(r)$	$\doteq$	$(1-r)_+^3(3r+1)$
$d = 1$	$m = 4$	$\Psi_{3,2}(r)$	$\doteq$	$(1-r)_+^5(8r^2+5r+1)$
$d \leq 3$	$m = 0$	$\Psi_{2,0}(r)$	$=$	$(1-r)_+^2$
$d \leq 3$	$m = 2$	$\Psi_{3,1}(r)$	$\doteq$	$(1-r)_+^4(4r+1)$
$d \leq 3$	$m = 4$	$\Psi_{4,2}(r)$	$\doteq$	$(1-r)_+^6(35r^2+18r+3)$

Tabelle 3.1: Positiv definite radiale Basisfunktionen mit kompaktem Träger aus [Wen96]. Konstante Faktoren wurden im Falle von $\doteq$ weggelassen.

Basisfunktionen mit kompaktem Träger

Bisher haben wir für ein gegebenes Semiskalarprodukt eine radiale Basisfunktion angegeben. Man kann natürlich auch umgekehrt radiale Funktionen mit besonderen Eigenschaften konstruieren. Insbesondere interessiert man sich im Zusammenhang mit Finite-Element-Methoden für Basisfunktionen mit kompaktem Träger. Wendland gelang es in [Wen95] und [Wen96], radiale Basisfunktionen anzugeben, für die die Matrix C aus (3.11) stets positiv definit ist, sofern man anstatt der Zellmittelungsfunktionale δ_η Punktfunktionale δ_{x_i} zu paarweise verschiedenen Punkten x_i verwendet. Man nennt radiale Funktionen mit dieser Eigenschaft **positiv definit**. Positiv definite Basisfunktionen haben den Vorteil, daß das Interpolationsproblem stets lösbar ist und $N = \{0\}$ trivial gewählt werden kann.
Für radiale Basisfunktionen mit kompaktem Träger kann gezeigt werden, daß sie nicht für beliebige Raumdimensionen d gleichzeitig positiv definit sein können (siehe beispielsweise [Wen96, Satz 3.9]). Daher werden die radialen Basisfunktionen mit kompaktem Träger in Abhängigkeit von der Raumdimension angegeben. Eine radiale Basisfunktion, die im $\mathbf{R}^d$ positiv definit ist, ist trivialerweise in allen niederdimensionalen Räumen positiv definit. Wir wollen noch erwähnen, daß sich die Eigenschaft der positiven Definitheit auf den Fall von Zellmittelungsfunktionalen derselben Raumdimension überträgt (siehe beispielsweise [Son97a]).
Die von Wendland konstruierten Funktionen bestehen jeweils aus einem polynomiellen Anteil für Radien $r \in [0, 1]$ und verschwinden für Radien $r \geq 1$. Es existiert eine Iterationvorschrift, mit der man für eine Differenzierbarkeitsordnung C^m und eine gegebene Raumdimension $d > 0$ die Koeffizienten des polynomiellen Anteils bestimmen kann. Es kann gezeigt werden, daß die so konstruierten Polynome diejenigen mit dem kleinsten Grad sind, die positiv definite radiale Basisfunktionen in $\mathbf{R}^d$ mit der geforderten Differenzierbarkeitsordnung definieren (siehe [Wen96]). Weiterhin sind die konstruierten

Basisfunktionen immer bis zu einer ungeraden Raumdimension positiv definit. Daher sind sie nur zu diesen Raumdimensionen angegeben. Die für die Finite-Volumen-Verfahren interessanten Fälle sind $d = 1$ und $d \leq 3$. In der Tabelle 3.1 geben wir diese Fälle bis zur Differenzierbarkeitsordnung 4 an.

Radiale Basisfunktionen für ENO-Verfahren

Konstruktion und Effizienz

Im Rahmen dieser Arbeit sollten die Einsatzmöglichkeiten von radialen Basisfunktionen innerhalb eines zweidimensionalen Finite-Volumen-Verfahrens getestet werden. Besondere Berücksichtigung sollten hier die Basisfunktionen mit kompaktem Träger finden, wie sie im vorangehenden Abschnitt besprochen wurden. Für das zweidimensionale Finite-Volumen-Verfahren wird man wieder Rekonstruktionen auf lokalen Gittern $\overline{\mathcal{L}}$ um eine Zelle $\omega \in \mathcal{G}$ berechnen. Verwenden wir hierzu eine radiale Basisfunktion ϕ, so müssen wir die Matrix C aus (3.11) bestimmen. Für die Berechnung eines Matrixeintrages $C_{\eta,\ell}$, $\eta, \ell \in \overline{\mathcal{L}}$, ist das folgende Doppelintegral zu lösen:

$$C_{\eta,\ell} = \delta_\eta(x \mapsto \delta_\ell(y \mapsto \phi(\|x - y\|_2))). \tag{3.13}$$

Dieses Integral wird für allgemeine, polygonale Zellen nur numerisch zu approximieren sein, da im Normalfall die Integranden als radiale Funktionen keine Polynome mehr sind. Man wird also für die Zellen entsprechende numerische Quadraturformeln

$$\frac{1}{|\eta|} \int_\eta f \ \mathsf{d}x \approx \sum_{k=1}^{M_\eta} g_{\eta,k} f(\mathsf{q}_{\eta,k}) \tag{3.14}$$

mit Gewichten $g_{\eta,k}$ und Quadraturpunkten $\mathsf{q}_{\eta,k} \in \eta$ konstruieren. Die Quadraturpunkte wird man aus bekannten Formeln für Dreiecke ableiten wollen und die Zellen entsprechend triangulieren. Durch Einsetzen von (3.14) in (3.13) erhält man dann in kanonischer Weise eine Doppelsumme:

$$C_{\eta,\ell} \approx \sum_{j=1}^{M_\eta} g_{\eta,j} \sum_{k=1}^{M_\ell} g_{\ell,k} \phi(\|\mathsf{q}_{\eta,j} - \mathsf{q}_{\ell,k}\|_2) \tag{3.15}$$

Diese Doppelsumme wird selbst für sehr einfache Zellen (Dreiecke) numerisch so kostenintensiv sein, daß es sich nicht lohnt, die Koeffizienten $C_{\eta,\ell}$ während des Rekonstruktionsschrittes auszuwerten. Um zu einem bezahlbaren Verfahren zu gelangen, wird man für alle Paare von Zellen (η, ℓ), die

in einem gemeinsamen lokalen Gitter vorkommen, die zugehörigen Matrixkomponenten einmal zum Programmstart berechnen und in einer Tabelle speichern. Bedenkt man, daß typischerweise für ein Verfahren dritter Ordnung die direkten Nachbarn und deren Nachbarn in einem ENO-Verfahren berücksichtigt werden müssen, so gelangt man hier zu 19 reellen Zahlen pro Zelle, wenn man regelmäßige Dreiecksgitter mit sechs direkten Nachbarzellen voraussetzt. Dieser Aufwand ist für ein Verfahren dritter Ordnung zur Lösung der Euler-Gleichungen tragbar — auch bei einem polynomiellen Verfahren mit quadratischen Polynomen sind für vier Zustandsvariablen jeweils sechs Koeffizienten zu speichern. Daher entspricht der Speicheraufwand für die Koeffizienten $C_{\omega,\eta}$ der Speicherung einer Zustandsverteilung mit quadratischen Polynomen.
Um eine stetige Rekonstruktionsmethode zu erhalten, wird man ein gewichtetes ENO-Verfahren konstruieren. Man wird also mit unterschiedlichen lokalen Gittern Rekonstruktionen berechnen und von diesen eine Konvexkombination bilden. Dies bedeutet, daß man mindestens für die in der Vereinigung der lokalen Gitter enthaltenen Zellen die Koeffizienten speichern muß. Wir hatten oben angenommen, daß für die Konstruktion eines Verfahrens dritter Fehlerordnung bei einem unstrukturierten Gitter typischerweise 19 Zellen in der Vereinigung sind. Für die Darstellung einer einzigen Zustandskomponente wird hier circa der dreifache Speicheraufwand einer quadratischen Rekonstruktion erforderlich werden.
Ein weiteres Problem bei der Anwendung von radialen Basisfunktionen ist die Auswertung zwecks Randintegration der numerischen Flüsse. Für die Auswertung an einem einzigen Quadraturpunkt q sind alle Basisfunktionen r_η an diesem Punkt auszuwerten:

$$r_\eta(\mathsf{q}) = \delta_\eta(y \mapsto \phi(\|\mathsf{q} - y\|_2)).$$

Dies läßt sich wieder durch die numerische Integralformel (3.14) bewerkstelligen. Allerdings bietet es sich nicht unbedingt an, dies für jeden Zeitschritt zu wiederholen, da allein die Auswertung einer einzigen Funktion r_η im Normalfall teurer ist als die eines quadratischen Polynoms.[1] Aus Effizienzgründen wird man die Funktionswerte $r_\eta(\mathsf{q})$ zum Programmstart abspeichern müssen. Setzen wir wieder eine Rekonstruktion aus 19 Basisfunktionen zusammen, dann wird man für jeden Quadraturpunkt der Zelle 19 reelle Zahlen speichern müssen. Bedenkt man, daß bei typischerweise sechs Nachbarzellen jede Zelle von zwölf Kantenstücken berandet wird und auf jeder Kante zwei Quadraturpunkte für ein Verfahren dritter Fehlerordnung benötigt werden,

[1] Mindestens eine Quadratwurzel fällt für die Norm an, evtl. ein Logarithmus bei Thin-Plate-Splines oder eine verzweigte Anweisung bei den Funktionen mit kompaktem Träger.

dann sind für die Auswertungen in einer Zelle 456 reelle Zahlen zu speichern. Reduziert man die Zellen auf Dreiecke, dann muß man immerhin noch 114 reelle Zahlen speichern. Diesen Kostenaufwand haben wir für ein typisches gewichtetes ENO-Verfahren dritter Fehlerordnung abgeschätzt. Insgesamt gelangt man zu einem numerischen Verfahren, welches selbst bei gutmütiger Rechnung und sehr vielen Optimierungsversuchen einen erheblich höheren Speicherbedarf als ein Verfahren mit quadratischen Polynomen hat. Selbst wenn man den zusätzlichen Zeitaufwand für das Vorausberechnen der Koeffizienten vernachlässigt, wird man bereits bei der Auswertung der Funktionen durch einfache Skalarprodukte mit den Koeffizienten mehr als doppelt so viele Multiplikationen durchführen müssen, als für die Anwendung des einfachen Hornerschemas notwendig sind.
Aus den Betrachtungen wird man ohne Zweifel den Schluß ziehen dürfen, daß, zumindests für die beschriebenen Gitter aus Kontrollvolumina mit einer so großen Anzahl an Teilkanten, ein Verfahren, welches radiale Basisfunktionen verwendet, mehr als 16 mal so viel Speicher benötigt wie ein Verfahren mit quadratischen Polynomen. Zudem wird, abgesehen vom 16-fachen Speicherdurchsatz, das Verfahren auch noch mindestens doppelt so viel multiplizieren müssen.
Man wird also notwendigerweise das beschriebene Konzept für den praktischen Einsatz von radialen Basisfunktionen erheblich vereinfachen müssen. Eine Möglichkeit besteht darin, ausschließlich strukturierte, kartesische Gitter zu verwenden; wir verweisen auf die Arbeit [Gut98]. Weiterhin kann man etwas Speicheraufwand sparen, wenn man auf dreieckigen Zellen rechnet. Dies bereitet jedoch in der Praxis einige Schwierigkeiten, da hier immer nur drei Raumrichtungen in den numerischen Flußfunktionen berücksichtigt werden. Für den Transport von Unstetigkeiten scheinen diese Gitter daher in der Praxis nicht besonders gut geeignet zu sein. In der Arbeit [HS99] wird ein solches Verfahren vorgestellt, in dem allerdings pro Kante eine Rekonstruktion in sogenannten charakteristischen Variablen berechnet wird. Die Übertragung dieser Idee für radiale Basisfunktionen bedeutet allerdings wieder erheblich mehr Speicherbedarf.
Eine weitere Möglichkeit der Vereinfachung besteht darin, bereits bei der Konstruktion eines Verfahrens nicht von den Riesz-Darstellungen der Punktfunktionale δ_x auszugehen und diese mit den Zellmittelungsfunktionalen zu falten, sondern direkt Funktionen r_ω für die Zellen $\omega \in \mathcal{G}$ anzugeben, die einfach zu integrieren und gleichzeitig einfach auszuwerten sind. Da wir keine Stetigkeitsbedingung an den Zellgrenzen zu erfüllen haben, müssen wir bei der Konstruktion auch keine stetigen Funktionen r_ω konstruieren. Als kostengünstige Variante bieten sich hier die charakteristischen Funktionen der Zellen an: $r_\omega = \chi_\omega$. Diesen Fall haben wir bereits in Gleichung (3.3) auf

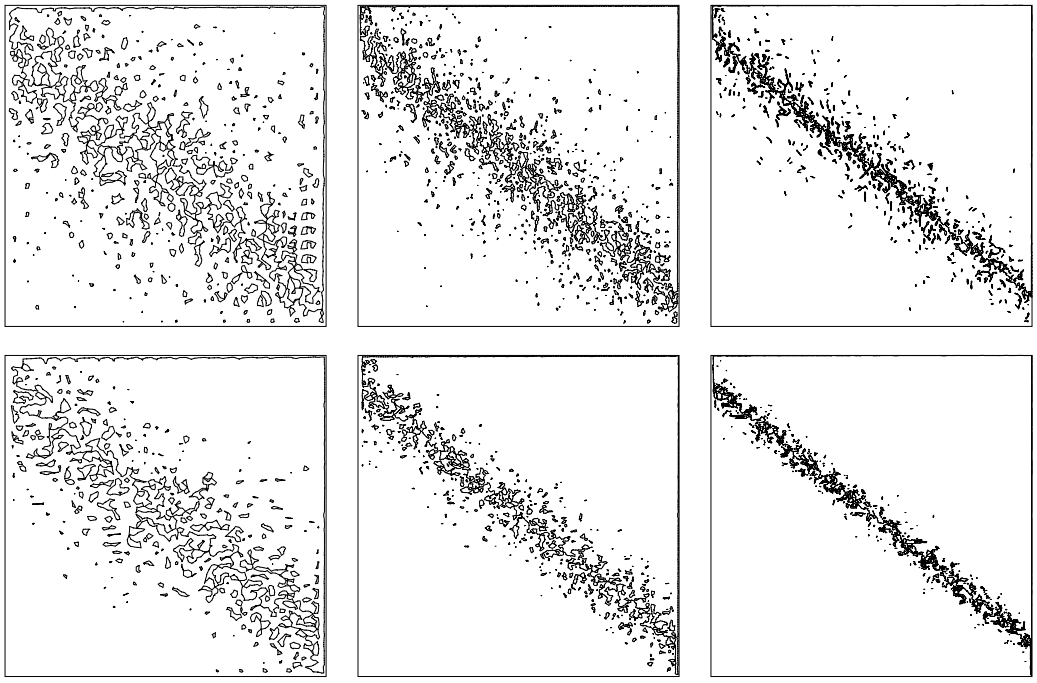


Abbildung 3.1: Rekonstruktion der Ableitung in x-Richtung der Funktion aus Gleichung (3.16) mittels $\Psi_{3,1}$ in den Gitterpunkten von drei unterschiedlich feinen Triangulierungen (von links nach rechts). In den oberen Abbildungen wurden nur die direkten Nachbarn eines Punktes hinzugenommen. Unten dagegen wurden auch noch die Nachbarn der Nachbarn berücksichtigt. Dargestellt ist die Höhenlinie für den Wert 11.045 der linear auf den Dreiecken interpolierten Daten.

Seite 83 diskutiert.

Rekonstruktionsqualität

Trotz des sehr viel höheren Speicherbedarfs kann es natürlich sinnvoll sein, radiale Basisfunktionen einzusetzen, wenn hierdurch die Qualität der Lösungen gesteigert werden kann. Dies wollen wir zunächst für den Fall der radialen Basisfunktionen mit kompaktem Träger diskutieren.
In der Arbeit [AHS98] wurde analysiert, ob man für den Fall der Funktionen mit kompaktem Träger den Radius des Trägers mit der Maschenweite h des Gitters mitskalieren darf — ob man also als Basisfunktion die Funktion $\phi(r/h)$ mit einer von h unabhängig gewählten Funktion ϕ verwenden soll. Dies sollte man erwarten, wenn die Lösung nicht von der Skalierung des Gebietes abhängen soll. Am Beispiel der Rekonstruktion der konstanten

Funktion $u(x) = 1$ sieht man bereits, daß für $h \to 0$ der Rekonstruktionsfehler konstant bleibt, sofern der Raum der Rekonstruktionsfunktionen nicht bereits die konstanten Funktionen enthält und man immer nur eine feste Anzahl von Zellen in den Rekonstruktionsprozeß einbezieht. Daher ist für die Konstruktion eines skalierungsunabhängigen Verfahrens unbedingt erforderlich, daß man den Funktionenraum mit den konstanten Polynomen anreichert. Will man die Konvergenzordnung des Verfahrens über $\mathcal{O}(h)$ hinaus steigern, dann wird man entsprechend den Funktionenraum um die linearen Polynome erweitern müssen. Diese Überlegung wird letztlich dazu führen, daß man bei einem skalierungsinvarianten Verfahren der Ordnung $\mathcal{O}(h^{q+1})$ alle Polynome bis zum Grad q im Ansatzraum berücksichtigen muß.
Aus dem beschriebenen Grund wurde der Fall eines skalierungsabhängigen Verfahrens studiert. Es wurde untersucht, ob ein Finite-Volumen-Verfahren mit fest gewählten und nicht mitskalierten radialen Basisfunktionen mit kompaktem Träger funktionieren kann. Hierzu wurde die zweimal stetig differenzierbare radiale Basisfunktion $\Psi_{3,1}$ aus Tabelle 3.1 ohne weitere Skalierung verwendet, um die quadratische Funktion

$$u(x, y) = 3.145x^2 + 2.14y^2 + 4x + 5.25y + 7.8xy + 673 \tag{3.16}$$

zu rekonstruieren. Der Einfachheit halber wurden nicht die Zellmittelungswerte, sondern Funktionswerte auf einer relativ gleichmäßigen Triangulierung eines Quadrates rekonstruiert. Für jeden Gitterpunkt wurde eine lokale Rekonstruktion mit den Funktionswerten des Punktes selbst und seinen direkten Nachbarpunkten (über Kantenverbindungen) bestimmt. Weiter wurde eine Rekonstruktion bestimmt, in der zusätzlich die Nachbarn der Nachbarpunkte berücksichtigt wurden. Von diesen Rekonstruktionen mit der radialen Basisfunktion $\Psi_{3,1}$ wurde eine punktweise Ableitung in Richtung der x-Achse bestimmt. Eine ähnliche Ableitung würde in einem Verfahren zur Berechnung der linearen Advektionsgleichung in jedem Zeitschritt zur Berechnung der mittleren Flußdivergenz durchzuführen sein. Um das Verhalten in Abhängigkeit von der Maschenweite zu studieren, wurde mit dem Adaptionsverfahren des letzten Kapitels dieser Arbeit die Maschenweite sukzessive halbiert. Die Ergebnisse dieser Ableitungsberechnung sind in Abbildung 3.1 zu sehen. Hier ist eine Äquipotentiallinie für den Wert 11.045 mittels linearer Interpolation der Eckpunktwerte der Dreiecke dargestellt. Ein gewisses konvergentes Verhalten des Verfahrens für $h \to 0$ gegen den Verlauf der Höhenlinie

$$y \approx -0.8x + 0.9$$

ist zwar festzustellen, jedoch reicht dieses Verhalten nicht aus, um eine lineare Advektiongleichung numerisch zu lösen.

Die bisherigen Betrachtungen legen den Schluß nahe, daß man unbedingt den Funktionenraum mindestens mit Polynomen bis zum Grad 1 anreichern muß, um zu einem Rekonstruktionsverfahren der Fehlerordnung $\mathcal{O}(h^2)$ zu gelangen. In der Arbeit [Son96] wird für den Thin-Plate-Spline eine deutliche Verbesserung der Rekonstruktionsqualität gegenüber einseitigen linearen Polynomen gezeigt. In der Arbeit [Gut98] werden dagegen Thin-Plate-Splines mit Polynomen höheren Grades für den Fall von kartesischen Gittern verglichen. Dieser Vergleich zeigt deutlich, daß Polynome gegenüber den Thin-Plate-Splines keine Qualitätseinbußen zeigen (siehe beispielsweise den Vergleich der Totalvariationen). Es ist daher fraglich, ob im Rahmen einer Finite-Volumen-Methode die kostenintensiven radialen Basisfunktionen eine Alternative zu der üblichen Technik der Anhebung des Polynomgrades darstellen können.

Kapitel 4

Adaption und Parallelisierung

Gitteradaption

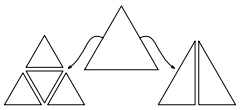
In diesem Abschnitt werden die Neuerungen im Bereich der dynamischen Gitteradaption beschrieben, die ich im Rahmen dieser Arbeit entwickelt und programmiert habe. Der Schwerpunkt dieser Beschreibungen liegt in den Verbesserungen gegenüber dem in [Hem96], [HHS93], [HHS94] und [FHMS96] beschriebenen Adaptionsverfahren, welches ich von 1992 bis 1996 bei der Deutschen Forschungs- und Versuchsanstalt für Luft- und Raumfahrt erarbeitet hatte.
Die Neuerungen bestehen in einem verbesserten Vergröberungsalgorithmus, einer Strategie für die Interpolation von Zellmitteldaten zwischen dynamisch veränderten Gittern mit beliebig hoher Fehlerordnung und einer Technik, mit der sichergestellt werden kann, daß sich die detektierten Phänomene nicht aus den verfeinerten Bereichen herausbewegen: Oft genug passiert es nämlich in der Praxis, daß Adaptionsindikatoren instationäre Phänomene „zu scharf“ auflösen, unstetiges Verhalten in der Nähe dieser Phänomene aufweisen oder der Bewegung der Phänomene nicht Rechnung tragen.

Rot-Grün-Adaption

Das Finite-Volumen-Verfahren wurde so programmiert, daß es nach einer vorgegebenen Anzahl von gerechneten Zeitschritten eine Adaption des Gitters anfordert. Bei den stark instationären Beispielen für die Euler-Gleichungen wurde typischerweise alle fünf bis zehn Zeitschritte adaptiert. Hierbei wird zuerst ein lokaler Adaptionsindikator verwendet: Dieser liefert für jedes Dreieck des Primärgitters eine positive reelle Zahl, die als Maß für die Größe des lokalen Fehlers interpretiert wird. Überschreitet dieses Maß einen gewissen Schwellwert, so wird das Dreieck für eine lokale Verfeinerung vorge-

merkt. Liegt das Maß sehr dicht bei Null und unterschreitet es einen zweiten Schwellwert, so wird das Dreieck umgekehrt für eine mögliche Vergröberung markiert. Die verwendeten Adaptionsindikatoren werden im Abschnitt über „Adaptionsindikatoren für instationäre Probleme“ ab Seite 104 erklärt.

Nachdem festgelegt ist, welche Dreiecke verfeinert, welche nicht verfeinert werden müssen und welche möglichst zu vergröbern sind (3 mögliche Zustände), wird der geometrische Teil der Gitteradaption durchgeführt. Dabei werden Dreiecke auf zwei verschiedene Arten verfeinert: Bei der „Rot-Verfeinerung“ wird jede Kante eines Dreieckes halbiert und das Dreieck in vier ähnliche „rote“ Teildreiecke geteilt. Bei der „Grün-Verfeinerung“ wird nur eine Kante des Dreiecks halbiert und das Dreieck in zwei „grüne“ Dreiecke geteilt. In der nebenstehenden Abbildung sind diese beiden Verfeinerungen illustriert. Die Grün-Verfeinerung wird nur verwendet, um hängende Knoten — Eckpunkte auf Kanten anderer Dreiecke — zu eliminieren. Damit die Dreiecke nicht degenerieren, werden grün verfeinerte Dreiecke vor einer weiteren Verfeinerung zu ihrem gemeinsamen Ausgangsdreieck vereinigt und erst dann kann eine Rot-Verfeinerung vorgenommen werden. Dies geschieht immer dann, wenn ein grün verfeinertes Dreieck einen hängenden Knoten besitzt oder es für eine weitere Verfeinerung markiert wurde. Alle anderen Dreiecke werden verfeinert, wenn sie mehr als einen hängenden Knoten besitzen oder wenn sie für eine Verfeinerung markiert wurden. Dieser Prozeß wird fortgeführt, bis alle Dreiecke, die Resultat einer Grün-Verfeinerung sind, keinen hängenden Knoten mehr haben, alle anderen höchstens einen. Die verbleibenden hängenden Knoten können danach mit einfachen Grün-Verfeinerungen eliminiert werden. Die primäre Verfeinerungsstrategie besteht also aus einer Viertelung der Dreiecke. Für ein anderes Verfeinerungsverfahren, welches ausschließlich Halbierungen von Dreiecken durchführt, sei auf die Arbeiten [Bän91] und [Hem92] verwiesen.



Das Verfeinerungsverfahren wurde in zwei Teile zerlegt: Im ersten Schritt werden alle Dreiecke für Verfeinerungen vorgemerkt, die in Folge der Verfeinerung ihrer Nachbarn verfeinert werden müssen. Dieser Programmteil ist rekursiv. Danach kann der Gesamtspeicherbedarf für die neuen Dreiecke und Punkte berechnet und reserviert werden. In einer linearen, nicht rekursiven Schleife können schließlich die markierten Dreiecke verfeinert werden.

Etwas schwieriger ist die Programmierung des ersten Schrittes, in dem die Markierungen gesetzt werden. Hierzu überprüft eine Funktion für ein Dreieck, ob es in Folge der Verfeinerung seiner benachbarten Dreiecke auch verfeinert werden muß. In diesem Fall wird das Dreieck ebenfalls markiert und diese Entscheidung kann wiederum die Situationen bei benachbarten Dreiecken beeinflussen. Es kommt vor, daß Nachbarn, deren Test vorher keine Notwendigkeit für eine Verfeinerung ergeben hat, nun mitverfeinert werden

müssen. Daher wird rekursiv für die Nachbardreiecke diese Funktion (erneut) aufgerufen, sofern sie nicht bereits für eine Verfeinerung markiert sind. Danach steht fest, welche Dreiecke verfeinert werden und welche nicht, und die Vergröberung wird vorbereitet, bevor das Gitter durch Adaptionsvorgänge verändert wird.
Der Vergröberungsalgorithmus soll vorhergegangene Verfeinerungen durch lokale Änderungen des Gitters rückgängig machen. Genauer: Das Vergröberungsverfahren soll nur Triangulierungen erzeugen, die man auch durch eine Folge von Verfeinerungen des Ausgangsgitters erhalten kann. Hierzu muß man herausfinden, aus welchen Dreiecken verfeinerte Dreiecke ursprünglich entstanden sind. Dies erreicht man im Falle der vorliegenden Verfeinerungsstrategie, indem man zu jedem Punkt des Gitters seinen Entstehungszeitpunkt in Form einer ganzen Zahl speichert: Die Punkte der Ausgangstriangulierung bekommen als *Geburtsjahr* die Zahl 0 zugewiesen. Ein nachträglich eingefügter Punkt erhält als Geburtsjahr den Wert $1 + \max\{G_1, G_2\}$, wobei G_1 und G_2 die Geburtsjahre der beiden Endpunkte der Kante sind, auf welcher der neue Punkt als Mittelpunkt eingefügt wird. Diese Information ist ausreichend, um den gesamten *Stammbaum* eines Dreieckes zu rekonstruieren:

- Ein Dreieck, dessen Punkte alle das Geburtsjahr 0 haben, stammt aus der Ausgangstriangulierung.
- Ein Dreieck, dessen Punkte gleiches positives Alter (> 0) haben, ist das innere Dreieck einer roten Verfeinerung.
- Ein Dreieck, welches genau zwei jüngste Eckpunkte besitzt, ist ein äußeres Dreieck einer Rot-Verfeinerung. Der ältere Eckpunkt ist ein Eckpunkt des *Vorfahren*.
- Ein Dreieck mit genau einem jüngsten Punkt ist ein grün verfeinertes Dreieck, und der jüngste Punkt ist der eingefügte *grüne Knoten*.

Grundlage des in [Hem96] vorgestellten Vergröberungsverfahrens ist die Suche nach einfachen Gebieten, die gemeinsam vergröbert werden können. Das Verfahren hat den Nachteil, daß es zu Konstellationen kommt, in denen man das Vergröberungsverfahren mehrfach aufrufen muß, um eine einfache Verfeinerung rückgängig zu machen.
Das neu implementierte Verfahren verzichtet auf eine solche Suche. Stattdessen wurde das Verfahren so gestaltet, daß es, wenn nur hinreichend viele Dreiecke für Vergröberungen markiert wurden, genau den gleichen *Durchsatz* wie das Verfeinerungsverfahren besitzt: Was in einem Verfeinerungslevel eingefügt wurde, kann nun in einem Schritt rückgängig gemacht werden.

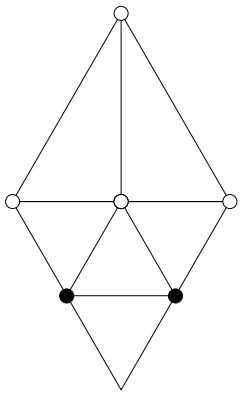
Wir erklären die neue Strategie: Jeder Punkt befindet sich in einem der folgenden beiden Zustände: Entweder ist er **gesperrt** oder **frei**. Gesperrte Punkte dürfen nicht aus der Triangulierung entfernt werden. Dagegen sind freie Punkte die Eckpunkte von Dreiecken, die vergröbert werden können. Der Algorithmus besteht aus zwei Phasen: Im ersten Schritt wird für jeden Punkt geklärt, ob er frei oder gesperrt ist. In der zweiten Phase werden die freien Punkte gelöscht, und die angrenzenden Dreiecke werden in ihre Vorfahren zurückverwandelt. Dieser zweite Schritt wird erst nach der vollständig abgeschlossenen Verfeinerung durchgeführt. Der erste Schritt wird dagegen zwischen der Vervollständigung der Information über die zu verfeinernden Dreiecke und der tatsächlichen Verfeinerung der Dreiecke ausgeführt. Dieser erste Schritt ist teilweise rekursiv:

1. Zuerst werden alle Punkte auf den freien Zustand initialisiert.

2. Anschließend werden die Punkte gesperrt, die das Geburtsjahr 0 haben, denn diese stammen aus der Ausgangstriangulierung und können nicht entfernt werden.[1]

3. Danach werden die Eckpunkte der Dreiecke gesperrt, die noch verfeinert werden sollen, ferner die Ecken der nicht grün verfeinerten Dreiecke, die nicht für eine Vergröberung markiert sind.

4. Anschließend wird bei allen Kanten, die nicht bei einer grünen Verfeinerung zwischen zwei grünen Dreiecken eingefügt wurden, überprüft, ob ihre Eckpunkte unterschiedliches Alter haben. Trifft dies zu, so wird der jeweils ältere Punkt gesperrt.



Schließlich beginnt der rekursive Part:

1. Besitzt ein inneres Dreieck einer roten Verfeinerung zwei gesperrte Punkte, dann wird auch der dritte Punkt gesperrt.

2. Alle Eckpunkte eines grünen Dreicks werden gesperrt, wenn der eingefügte grüne Knoten gesperrt ist.

Das Sperren eines Punktes führt dazu, daß alle an diesen Punkt angrenzenden Dreiecke überprüft werden müssen. Dies führt zu der erwähnten Rekursion. Sind beispielsweise in der nebenstehenden Abbildung die dunklen Punkte gesperrt, so sperrt die Rekursion auch alle hellen Punkte.

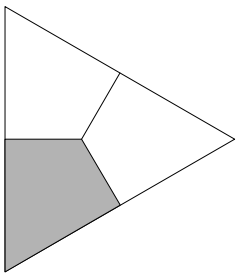
[1] Es bietet sich an, durch ein negatives Alter einen Punkt als frei zu markieren. Punkte aus der Ausgangstriangulierung sind dann stets gesperrt.

Sind die Vorbereitungen für die Verfeinerung und Vergröberungen des Gitters abgeschlossen, so können in einer einfachen Schleife Dreiecke vergröbert und verfeinert werden. Das Verfahren wurde so geschrieben, daß erst die Verfeinerungen und im Anschluß die Vergröberungen durchgeführt werden, denn es kann vorkommen, daß um einen Punkt herum sowohl Dreiecke verfeinert als auch vergröbert werden. Um überflüssigen Informationsverlust bei der Interpolation von Zellmitteldaten für Boxgitter zu vermeiden, wurde die verlustärmere Verfeinerung vorangestellt.
Das in groben Zügen nun beschriebene Adaptionsverfahren wurde in der Programmiersprache C++ als Bibliotheksklasse implementiert und kann als *blackbox* für die Veränderung von Triangulierungen in unterschiedlichsten Anwendungsbereichen (FEM/FVM/CAD) verwendet werden. Jedoch treten in jeder Anwendung unterschiedliche Interaktionsprobleme mit spezifischen Daten auf. Beispielsweise müssen in dem beschriebenen Finite-Volumen-Verfahren Zellmitteldaten von einem Gitter zum nächsten interpoliert werden. In Finite-Elemente-Anwendungen besteht das Problem, auch höhere Momente zu interpolieren. Weiterhin kann es unter Umständen erwünscht sein, neu eingefügte Randpunkte auf eine nichtlineare Randkurve zu projizieren. Für solche Ereignisse, die während der Adaption des Gitters eintreten, ruft das Verfahren virtuelle Funktionen auf, die der *Außenwelt* mitteilen, welche Veränderungen im Gitter vorgenommen werden. Beispielsweise wird beim Erzeugen eines neuen Punktes eine Funktion aufgerufen, die diesen auf den Mittelpunkt der entsprechenden Kante setzt. Überschreibt man die virtuelle Funktion, so wird entsprechend diese neue Funktion verwendet. Beispielsweise kann man hierdurch die Projektion von eingefügten Randpunkten auf kurvige Ränder erreichen.

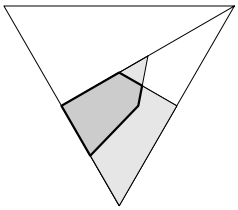
Interpolation von Daten zwischen Gittern

Das Finite-Volumen-Verfahren benötigt für die Anwendung der Adaption ein Interpolationsverfahren, um die Zellmitteldaten der Zustandsvariablen von einem Gitter vor der Adaption auf ein adaptiertes Gitter zu übertragen. Hierzu wird auf dem alten Gitter eine Rekonstruktionsfunktion mit dem gewichteten ENO-Verfahren, dem Rekonstruktionsverfahren mit Limitierung oder einem anderen Verfahren hoher Ordnung berechnet. Diese Rekonstruktionsfunktion wird dann auf den neuen Gitterzellen integriert, und die Zellmittel nach Division durch die Fläche ermittelt. Mindestens für Polynome ist die Integration der Rekonstruktionsfunktionen durch die sehr allgemeinen Werkzeuge des zweiten Kapitels ab Seite 47 vollständig abgehandelt. Jedoch besteht neben der Integration das Problem, daß man unmöglich jede Zelle des alten Gitters mit jeder Zelle des neuen Gitters schneiden kann. Ein sol-

cher Algorithmus hat quadratischen Aufwand und ist deshalb unbezahlbar. Deswegen muß die Suche nach sich schneidenden Zellen im alten und neuen Gitter auf eine möglichst lokale Suche beschränkt werden. Hierfür wird während der Adaption für jedes Dreieck eine Liste von Dreiecksnummern angelegt, die alle Nummern der alten Dreiecke enthält, die zusammen das neue Dreieck überdecken.



Nach der Adaption des Gitters wird die zuvor berechnete Rekonstruktionsfunktion auf den Zellen des neuen Gitters integriert. Dies kann als Schleife über die Dreiecke realisiert werden. Für jedes Dreieck sind die Anteile der zu den drei Eckpunkten zugeordneten Zellen zu berücksichtigen. Diese Anteile sind, wie in der nebenstehenden Abbildung gezeigt, jeweils Vierecke. Jedes dieser Vierecke des neuen Gitters wird nun mit den Vierecken, die sich aus den überlappenden alten Dreiecken ergeben, geschnitten. Konstruiert man die Boxen als Schnitte von Seitenhalbierenden der Dreiecke, so sind diese Vierecke immer konvex, und folglich sind die Schnittgebilde ebenfalls konvex und bestehen aus höchstens acht Ecken. In einer Routine für die Schnittberechnung zweier konvexer Vierecke kann man diese Eigenschaft effizient nutzen. Allerdings muß man berücksichtigen, daß es häufig vorkommt, daß Kanten dieser Vierecke exakt aufeinander liegen. Dieses Phänomen ist in der Randskizze für den Fall einer grünen Verfeinerung dargestellt. Daher können Schnittpunktberechnungen numerisch besonders kritisch werden. Eine numerisch einwandfreie Schnittpunktbestimmung und die Entscheidung, welche Eckpunkte von einem der beiden jeweils betrachteten Vierecke sukzessive abzuschneiden sind, ist unbedingt erforderlich.



Adaptionsindikatoren für instationäre Probleme

Für die Bestimmung der Dreiecke, die zu verfeinern oder zu vergröbern sind, wurden im wesentlichen zwei verschiedene Adaptionsindikatoren verwendet. Der erste hiervon ist ein Residuenfehlerschätzer, wie er beispielsweise in [Son97c] beschrieben ist. Ergänzungen hierzu findet man in [SS94], wo eine alternative, jedoch numerisch kostenintensivere Norm für das Residuum verwendet wird.
Neben diesen bekannten und bereits dokumentierten Residuenindikatoren wurde zusätzlich noch ein weiterer Indikator entwickelt, der sich besonders durch die folgenden Merkmale auszeichnet:

- einfache Programmierung,
- geringer Rechenaufwand,

- gute Ergebnisse (gerade in der Nähe interessanter Details wie Kontaktunstetigkeiten).

Wir beschreiben diesen Adaptionsindikator für gegebene Zellmittelwerte $\bar{v}_k(\omega)$ für die Zellen $\omega \in \mathcal{G}$ und die Zustandskomponenten $k = 1, \ldots, S$. Zunächst wird für jede der Zustandskomponenten die Variation der Zellmitteldaten bestimmt:

$$d_k(\Omega) \quad := \quad \max_{\omega \in \mathcal{G}} \bar{v}_k(\omega) - \min_{\omega \in \mathcal{G}} \bar{v}_k(\omega).$$

Den drei Eckpunkten eines Dreiecks T sind jeweils drei Zellen ω_1, ω_2 und ω_3 zugeordnet. Mit den Zellmittelwerten dieser Zellen kann man die Variation der Daten innerhalb des Dreiecks abschätzen:

$$d_k(T) \quad := \quad \max_{i=1,2,3} \bar{v}_k(\omega_i) - \min_{i=1,2,3} \bar{v}_k(\omega_i).$$

Jedem Dreieck kann nun eine reelle Zahl $r(T)$ zugeordnet werden, die das Verhältnismaximum an lokaler zu globaler Variation angibt:

$$r(T) \quad := \quad \max_{k=1,\ldots,s} \frac{d_k(T)}{d_k(\Omega) + \epsilon} \tag{4.1}$$

Die Konstante ϵ dient nur zur Regularisierung des Nenners für den Fall $d_k(\Omega) = 0$; ihr Wert betrug in den Anwendungen $2 \cdot 10^{-16}$.
Die lokale Größe $r(T)$ kann direkt als Indikator für Verfeinerungen und Vergröberungen verwendet werden: Übersteigt $r(T)$ eine gewisse vorgegebene Schranke $R_{\max}$, so wird das Dreieck für eine Verfeinerung vorgemerkt. Liegt dagegen der Wert von $r(T)$ unterhalb einer gewissen Schranke $R_{\min}$, so wird das Dreieck für eine Vergröberung markiert. In den Rechenbeispielen wurden die Konstanten $R_{\max}$ und $R_{\min}$ stets gleich groß und bei circa $0.02 = 2\%$ unabhängig von der Maschenweite gewählt. werden. Weil das beschriebene Vergröberungsverfahren nur dann Dreiecke vergröbert, wenn auch in der Nachbarschaft hinreichend viele Dreiecke für Vergröberungen markiert sind, ergibt sich eine gewisse (gewünschte) Trägheit des Vergröberungsverfahrens, die jedoch nicht zusätzlich durch die Wahl einer kleineren Vergröberungsschranke $R_{\min}$ verstärkt werden muß.
Sowohl der verwendete Residuenfehlerschätzer als auch der eben beschriebene Adaptionsindikator tendieren dazu, in der Nähe von Unstetigkeiten das Gitter unbeschränkt zu verfeinern. Dies kann wegen der Zeitschrittbeschränkung dazu führen, daß das Gesamtverfahren nicht in akzeptabler Rechenzeit die gewünschte Lösung berechnet. Daher wurde eine untere Schranke für die Dreiecksgröße vorgegeben: Dreiecke, deren Flächeninhalt bei einer Viertelung

einen kleineren Flächeninhalt als die angegebene Schranke haben, werden nicht für Verfeinerungen markiert. Mit dieser Regel gelingt es, die insgesamt benötigte Rechenzeit im voraus abzuschätzen.
Für die Berechnung instationärer Phänomene wird man typischerweise in regelmäßigen Abständen das Gitter den Strömungsbedingungen anpassen wollen. Hierfür gibt es einerseits die Möglichkeit, einige Zeitschritte auf einem nicht adaptierten Gitter zu rechnen und in regelmäßigen Abständen den Adaptionsindikator zu verwenden, um die später zu adaptierenden Bereiche zu bestimmen. Anschließend kann das Gitter adaptiert werden und dann wiederholt man die Zeitschritte auf diesem Gitter. Diese Vorgehensweise wurde bisher nicht realisiert, da sie zu kostenintensiv erscheint. Dagegen wurde aus Effizienzgründen das Gitter den augenblicklichen Strömungsphänomenen angepaßt und dann für einige Zeitschritte verwendet. Damit sich die Phänomene zwischen zwei Gitteradaptionen nicht aus den verfeinerten Gebieten herausbewegen, werden die feinen Gebiete im voraus größer gewählt als durch den Adaptionsindikator ursprünglich angezeigt. Zu diesen Zweck der „Vorausadaption“ wurde ein einfach zu programmierender und effizienter Algorithmus entwickelt.
Ausgangspunkt für den Algorithmus ist der für jedes Dreieck T bekannte Adaptionsindikator $r(T)$. Dieser lokale Indikator wird nun über mehrere Gitterzellen “verschmiert“, indem zu jedem Gitterpunkt P das Maximum der Indikatorwerte $r(T)$ aller angrenzenden Dreiecke T berechnet wird:

$$r(P) \;:=\; \max_{\overline{T}\cap P\neq\emptyset} r(T). \tag{4.2}$$

Mit diesem Feld von Indikatorwerten für die Gitterpunkte kann nun wieder ein Feld von Indikatorwerten für die Dreiecke bestimmt werden, indem man das Maximum der Indikatorwerte der drei Eckpunkte T_1, T_2 und T_3 eines Dreiecks verwendet:

$$\tilde{r}(T) \;:=\; \max_{i=1,2,3} r(T_i) \tag{4.3}$$

Nach der Abbildungsfolge

$$r(T) \to r(P) \to \tilde{r}(T) \tag{4.4}$$

liegen wieder Indikatorwerte für Dreiecke vor, und somit kann die Abbildungsfolge iterativ mit ihren Ausgabedaten wiederholt werden. Hierdurch werden jeweils die lokal größten Indikatorwerte um eine lokale Maschenweite verteilt. Dies ist in Abbildung 4.1 illustriert.
Bei den adaptiven Rechnungen zu dieser Arbeit wurden die Gitter alle fünf Zeitschritte adaptiert. Es hat sich gezeigt, daß es ausreicht, die

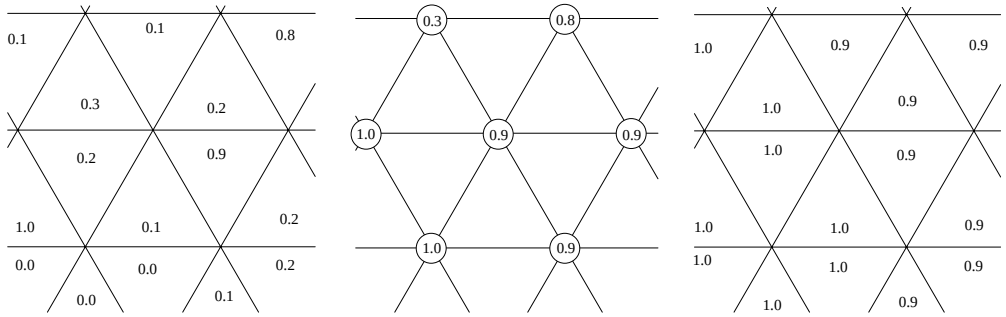


Abbildung 4.1: Beispielhaftes Verschmieren des Adaptionsindikators $r(T)$ (linkes Bild) durch Bilden des Maximums für jeden Gitterknoten nach Formel (4.2) (mittleres Bild) und anschließende Maximumsbildung nach Formel (4.3) (rechte Abbildung). Durch diese Operation gelingt es, die feinen Bereiche so zu erweitern, daß die entscheidenden Strömungsphänomene die fein aufgelösten Gitterbereiche zwischen zwei Adaptionsschritten nicht verlassen.

Glättungsabbildung $r(T) \to r(P) \to \tilde{r}(T)$ zwei- bis dreimal zu wiederholen, um sicherzustellen, daß sich keine Phänomene aus den feinen Gitterbereichen herausbewegen. Nachdem der Indikator mehrfachen Glättungsschritten unterworfen wurde, wird er in der eingangs beschriebenen Art verwendet, um zu bestimmen, welche Dreiecke zu verfeinern sind und welche gegebenenfalls vergröbert werden können. Die Vorausadaption vergrößert nur die Gebiete mit feinen Dreiecken. Die Wirkungsweise dieses einfachen Glättungsmechanismus' ist in den Abbildungen 4.2 und 4.3 zu erkennen: Um die Verdichtungsstöße und Kontaktunstetigkeiten ist jeweils ein breiteres Band zu erkennen, in dem die Dreiecke verfeinert wurden. In den Darstellungen 4.2, 4.3 und 4.4 wurde zusätzlich die Rechenzeit und der Speicherbedarf gegenüber global feinen Rechnungen mit qualitativ gleichwertigen Lösungen verglichen. In den Abbildungen 4.2 und 4.3 wurde der Variationsindikator aus Gleichung (4.1) eingesetzt. Dagegen wurde in Abbildung 4.4 der Residuenfehlerschätzer aus [Son97c] verwendet. Da hier wesentlich weniger Dreiecke für Vergröberungen markiert wurden, ist der Zeitgewinn gegenüber der global feinen Rechnung deutlich geringer. Allerdings wird dieses ungünstige Verhalten durch künstliche Oszillationen des Verfahrens in glatten Bereichen der Lösung verursacht. Die eigentlich erwünschte Sensibilität der Residuenfehlerschätzer stellt sich in diesem Fall als ungünstig heraus.

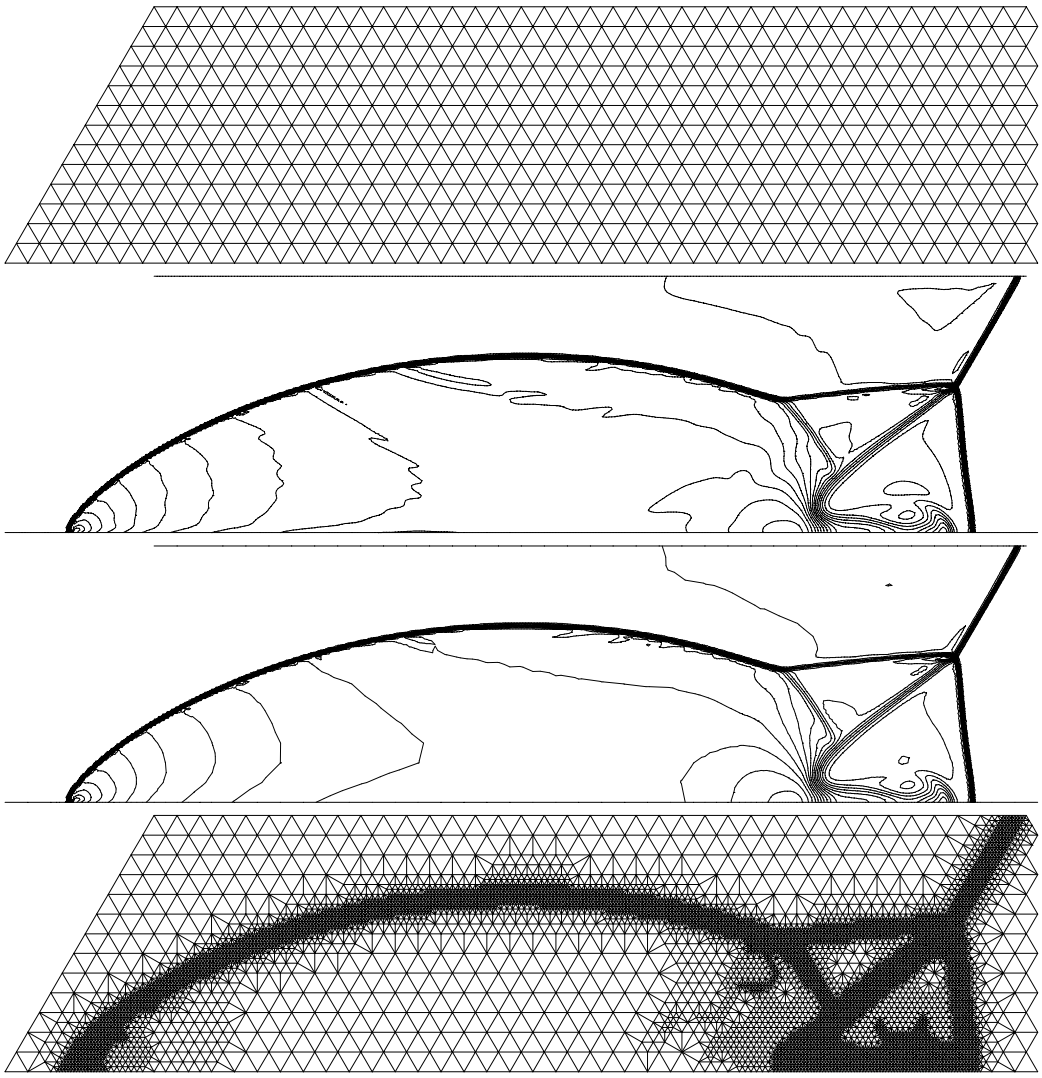


Abbildung 4.2: Vergleich der Ergebnisse mit global feinem Gitter und lokal adaptiertem Gitter. Für die global feine Rechnung wurde die oben dargestellte Triangulierung (Kantenlängen $h = 1/15$) dreimal rot verfeinert, und entsprechend wurde eine adaptive Rechnung mit einer maximalen Verfeinerungstiefe 3 durchgeführt. Das zweite Bild zeigt die Dichteverteilung als Ergebnis der global feinen Rechnung, und das dritte Bild demonstriert die Dichteverteilung zur gleichen Zeit bei adaptiver Rechnung. Im unteren Bild ist das zugehörige Gitter dargestellt. Die Berechnung mit global feinem Gitter hat insgesamt die 7.0 fache CPU-Zeit gegenüber der adaptiven Lösung benötigt. Der Speicherbedarf mit global feinem Gitter war um 4.5 mal höher als der maximale Speicherbedarf des adaptiven Verfahrens.

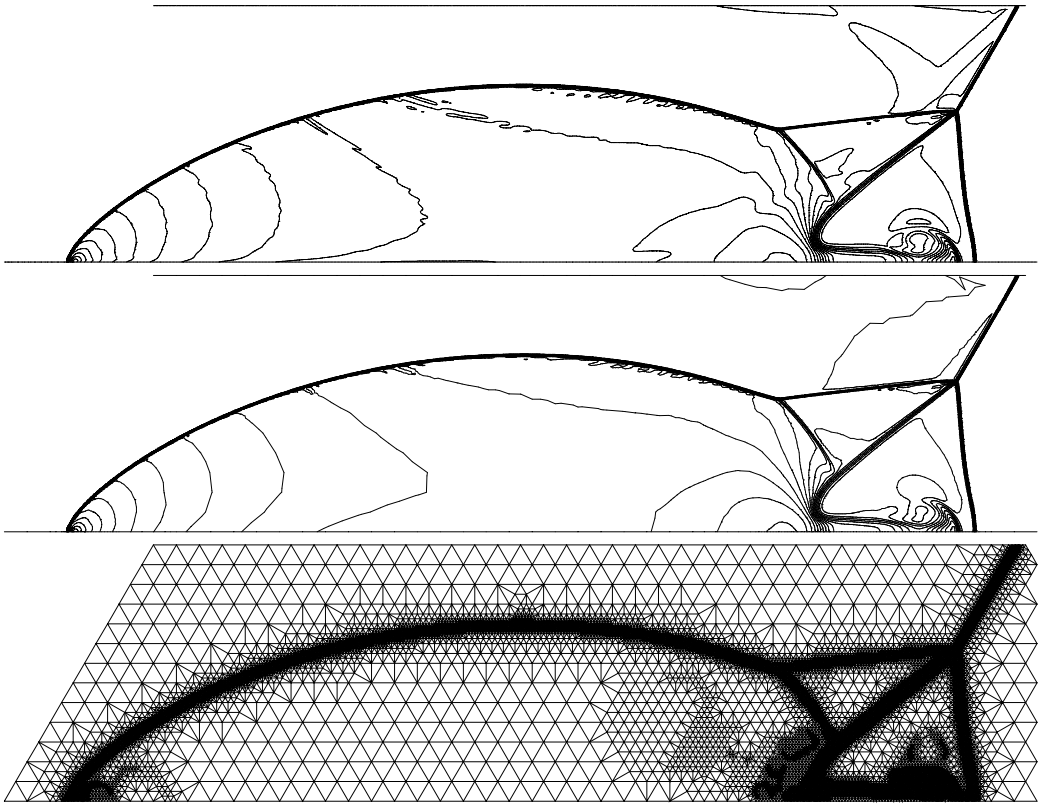


Abbildung 4.3: Gegenüber den Bildern auf Seite 108 wurde das global feine Gitter ein weiteres Mal rot verfeinert. Somit konnte das Vergröberungsverfahren bis zu vier Verfeinerungen rückgängig machen. Die obere Abbildung stellt die Dichteverteilung mit global feinem Gitter dar. Die mittlere Abbildung zeigt die Dichteverteilung als Ergebnis der adaptiven Berechnung. Die untere Abbildung zeigt das zur mittleren Abbildung gehörige Gitter. Die Lösung mit global feinem Gitter hat die 12.8 fache CPU-Zeit der adaptiven Rechnung gefordert. Der Speicheraufwand für die global feine Auflösung ist circa 8.2 mal höher als der maximale Speicherbedarf des adaptiven Verfahrens.

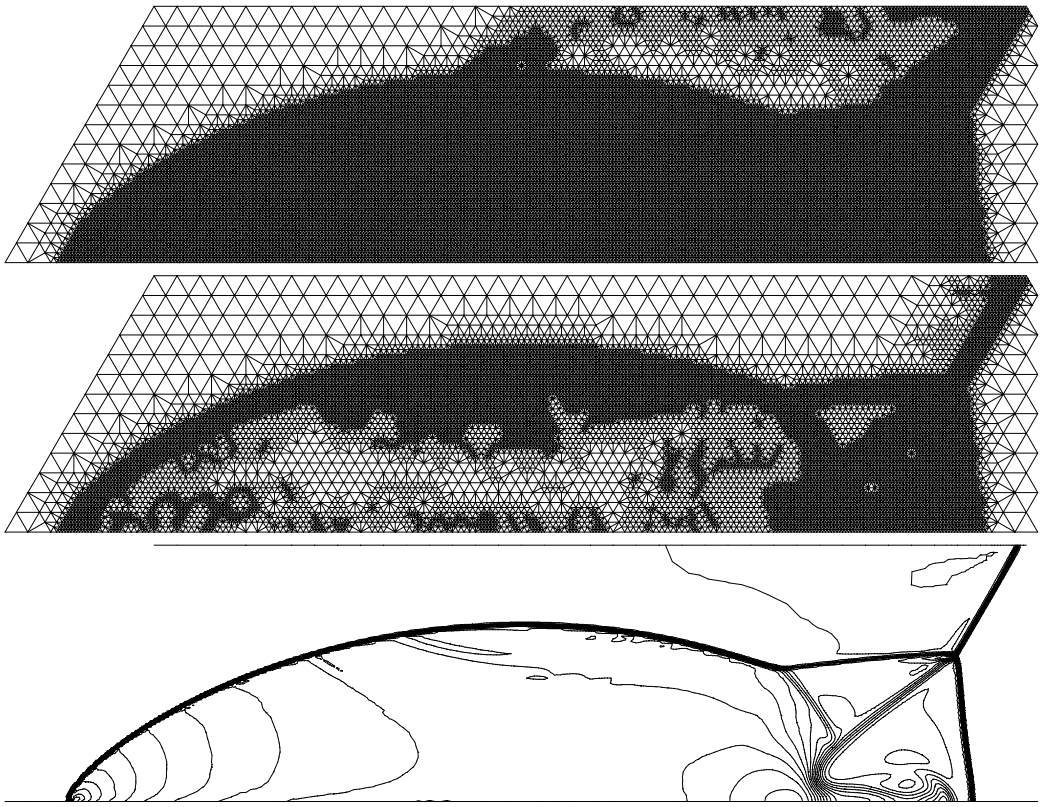


Abbildung 4.4: Auf dem gleichen Ausgangsgitter wie in Abbildung 4.2 auf Seite 108 wurde der Residuenfehlerschätzer aus [Son97c] verwendet. Die beiden oberen Abbildungen zeigen die Gitter für verschiedene Verfeinerungs- und Vergröberungsschranken. Dieser Adaptionsindikator reagiert empfindlich auf die unerwünschten Oszillationen des Verfahrens, die in der Bildmitte zu erkennen sind. Daher werden weniger Dreiecke vergröbert als bei dem Variationsindikator. Der Rechenzeitgewinn gegenüber dem global feinen Gitter beträgt im Falle der feineren Schranken (oberes Gitter) nur einen Faktor 2.6 und im Falle der gröberen Schranken (mittleres Gitter) 4.6. Im unteren Bild ist exemplarisch die mit dem Gitter aus der mittleren Abbildung berechnete Dichteverteilung dargestellt. Bei diesem Indikator war ein weiteres Anheben der Adaptionsschranken nicht möglich, ohne die Lösungen deutlich zu verschlechtern.

Parallelisierung

Für die Parallelisierung des Finite-Volumen-Verfahrens wurden sogenannte Thread-Funktionen (thread = engl. Faden) implementiert. Diese bieten einen sehr einfachen und gleichzeitig effizienten Zugang zur Parallelisierung numerischer Programme. Diese Art der Parallelisierung ist nur für Parallelrechner geeignet, die über eine Shared-Memory-Architektur verfügen, bei denen also alle Prozessoren den gleichen Hauptspeicher adressieren können. Eine Vielzahl der heute hergestellten Computer mit moderater Prozessorzahl entspricht dieser Bauart. Im Rahmen dieser Arbeit am Institut für Angewandte Mathematik in Hamburg standen folgende Shared-Memory-Maschinen zur Verfügung: Intel Dual-Pentium Computer, Zwei-Prozessor ULTRA Sparcs sowie zwei vclass-Parallelcomputer der Firma Hewlett Packard mit jeweils sechzehn Prozessoren.
Der Aufwand für die Änderung des zunächst seriellen Programmes in ein Programm zum Aufruf von Thread-Funktionen war relativ gering, verglichen mit dem alternativen und allgemeineren Konzept des Message-Passings. Der Hauptvorteil der Thread-Programmierung liegt darin, daß die parallele Bearbeitung keine oder nur geringe Neuverteilung der Daten verursacht: Der geometrische Aufwand einer Gebietseinteilung und das bisher nicht befriedigend gelöste Ausgleichsproblem (load-balancing) für adaptive Strömungslöser entfallen.
Innerhalb eines C- oder C++-Programmes kann eine Funktion als Thread-Funktion aufgerufen werden. Dies führt dazu, daß für den laufenden Prozeß ein weiterer Programmzähler erzeugt wird, der innerhalb des Betriebsystems konkurrierend zu dem ursprünglichen Programmzähler voranschreitet. Weiterhin verzweigt sich beim Erzeugen eines Threads der gemeinsame Stamm des Stacks, und der neue Thread bekommt einen eigenen Registersatz. Die Threads teilen sich den gemeinsamen Speicherbereich. Daher ist bei der Programmierung von Thread-Funktionen dafür zu sorgen, daß Speicherbereiche, die von einem der Threads beschrieben werden, von keinem anderen Thread gleichzeitig beschrieben oder auch nur gelesen werden.
Die Programmierung von Thread-Funktionen unter C und C++ wird durch einen POSIX-Standard unterstützt, für den geeignete Bibliotheken zur Verfügung stehen. Programme können so auf relativ portable Weise parallelisiert werden. Kern der Bibliothek ist im wesentlichen die Funktion `create_pthread`, der man einen Zeiger auf eine Funktion übergibt. Diese Funktion wird dann als separater Thread gestartet, und der Ausgangs-Thread kann weitere Operationen ausführen, ohne auf das Ende des gestarteten Threads warten zu müssen: Er könnte beispielsweise weitere Threads starten oder selbst einen Teil der Aufgabenstellung lösen. Für die Synchroni-

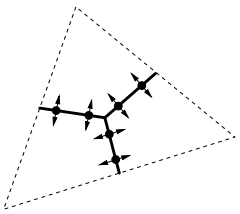
sierung gibt es mit der Funktion `pthread_join` die Möglichkeit, auf das Ende einer Thread-Funktion zu warten. Solche Wartezeiten wird man natürlich minimieren wollen. Dies ist die einzige Form der Synchronisierung, die in dem Finite-Volumen-Verfahren verwendet wurde.

Parallelisierung der Rekonstruktion

Ein Finite-Volumen-Verfahren besitzt zwei kostenintensive Aufgabenstellungen: Zum einen die Rekonstruktion von Zellfunktionen aus Zellmitteldaten und zum anderen die sogenannte Flußschleife, in der mit numerischen Flußfunktionen Integrale von Flüssen über Zellgrenzen berechnet werden. Die Gitteradaption mit Adaptionsindikator, Rekonstruktion und Interpolation nimmt ungefähr die Rechenzeit eines Zeitschrittes in Anspruch. Sie wurde bisher nicht parallelisiert.
In diesem und im nächsten Abschnitt soll die Parallelisierung der beiden Hauptschleifen des Verfahrens beschrieben werden, wie sie in einer gemeinsamen Arbeit mit Oliver Friedrich realisiert wurde (siehe auch [Fri99]).
Die Rekonstruktion einzelner Zellfunktionen aus global gegebenen Zellmitteldaten zerfällt in kanonischer Weise in einzelne Threads: Jeder Thread übernimmt einen gewissen Satz (disjunkte Einteilung) an Zellen und berechnet für diese Zellen Rekonstruktionsfunktionen. Hierbei muß sowohl auf das Gitter als auch auf die Zellmitteldaten nur lesend zugegriffen werden. Die resultierenden Rekonstruktionsfunktionen können problemlos in eine globale Datenstruktur (Array) geschrieben werden, solange die von den Threads übernommenen Zellmengen paarweise disjunkt zueinander sind. Eine solche disjunkte Zuordnung erhält man durch Zerschneiden der Menge aller Zellen in möglichst gleichgroße Teilfelder. Jeder Thread übernimmt hiervon eines und berechnet für diese Zellen Rekonstruktionsfunktionen. Der ursprüngliche Thread (Ausgangsprozeß) kann natürlich auch einen Teil der Arbeit übernehmen, nachdem er die anderen Threads gestartet hat. Hat er seine Aufgabe erledigt, so wird er auf das Ende der anderen Threads warten. Für das WENO-Verfahren ist dies die einzige Synchronisierung, die durchzuführen ist. Bei dem zentralen Rekonstruktionsverfahren mit Limitierung sind erst alle zentralen Rekonstruktionen zu berechnen. Anschließend werden die einzelnen Limitierungsfaktoren bestimmt, und schließlich können diese verwendet werden, um die zentralen Rekonstruktionen zu limitieren. Zwischen diesen drei Einzelschritten ist jeweils eine Synchronisierung der Threads erforderlich. Dies verringert den Geschwindigkeitsvorteil gegenüber dem WENO-Verfahren geringfügig.

Parallelisierung der Flußschleife

In der Flußschleife des Finite-Volumen-Verfahrens werden für die einzelnen Facetten (Geradenstücke in zwei Raumdimensionen) der Boxen Integrale der Flüsse in Richtung der Randnormalen berechnet. Diese Flüsse werden abhängig von der Transportrichtung den Zellmittelwerten der angrenzenden Zellen hinzugefügt oder von ihnen subtrahiert. Für Zellberandungen, die im Inneren des Gebietes liegen, sind hier jeweils zwei Zellen betroffen. Daher wird man die Berechnung der Flußintegrale als eine Schleife über alle Zellränder im Inneren des Gebietes und eine weitere Schleife über alle Zellränder am Rande des Gebietes schreiben. Auf die Parallelisierung der letzteren Schleife wurde bisher verzichtet, weil die Anzahl der Außenränder eine Größenordnung geringer ist als die der inneren Zellränder.



Hat man als Zellen die sogenanten Boxen einer Triangulierung, so kann man die Schleife über alle inneren Zellränder als Schleife über alle Dreiecke schreiben und dann für ein Dreieck die drei zugehörigen Boxgrenzen der zu den Eckpunkten gehörigen Boxen bearbeiten. In der nebenstehenden Abbildung sind diese drei Kanten pro Dreieck zusammen mit jeweils zwei Quadraturpunkten beispielhaft hervorgehoben.

Die Flüsse für die den drei Eckpunkten eines Dreiecks zugeordneten Boxen werden zu den bestehenden Werten addiert. Hierbei ist zu berücksichtigen, daß nicht unterschiedliche Threads gleichzeitig Flußanteile für die gleiche Box berechnen. Diesen Konflikt kann man lösen, indem man die Menge aller Dreiecke in Teilmengen zerlegt, für die jeweils gilt: Jedes Dreieck befindet sich in genau einer dieser Teilmengen, und kein Dreieck innerhalb einer Teilmenge hat einen gemeinsamen Eckpunkt mit einem anderen Dreieck derselben Teilmenge.

Eine solche Zerlegung ist nach jeder Gitteradaption neu zu berechnen. Der Aufwand hierfür beträgt bei N Dreiecken $3N$ Vergleiche, und der Speicheraufwand kann mit N zusätzlichen Indizes abgeschätzt werden. Typischerweise entstehen bei der Einteilung ungefähr so viele Teilmengen, wie ein Punkt maximal an benachbarten Dreiecken besitzt. Für typische zweidimensionale Triangulierungen sind dies sechs bis sieben Teilmengen.

Die Teilmengen werden nun in der Flußschleife nacheinander bearbeitet. Parallelisiert wird die Schleife über die Dreiecke innerhalb einer Teilmenge. Das Feld der Dreiecke einer Teilmenge wird wieder in möglichst gleich große Teilfelder zerlegt und diese werden von den einzelnen Threads getrennt bearbeitet.

Prozessoren	Beschleunigung
1	1.0
2	1.9
3	2.8
4	3.8
5	4.7
6	5.4
7	6.3
8	7.1
9	7.7
10	8.3
11	9.0
12	9.7

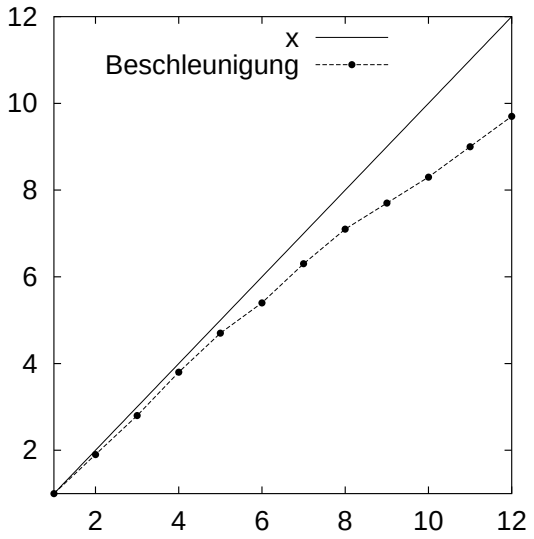


Tabelle 4.1: Beschleunigung durch Parallelisierung relativ zum seriellen Programm.

Beschleunigungsraten

Zum Schluß geben wir noch die Beschleunigungsraten an, die mit der beschriebenen Parallelisierungstechnik mittels Thread-Programmierung erzielt wurden. Die Rechenzeit wurde dabei mit einem äquivalenten Verfahren verglichen, welches keine zusätzlichen Operationen zur Parallelisierung durchführt. Die Zeiten wurden ohne Adaption des Gitters über mehrere Zeitschritte gemittelt. Es wurde mit quadratischen Polynomen zusammen mit dem ordnungserhaltenden Limiter gerechnet; die Beschleunigungsraten für das gewichtete ENO-Verfahren sind geringfügig höher. In Tafel 4.1 sind die Beschleunigungsraten angegeben, die für das Beispiel der Stoßreflexion (siehe Seite 73) auf einem Gitter der Maschenweite $h = 1/100$ erzielt wurden. Bei Rechnungen mit feineren Gittern steigt die Effizienz des parallelen Verfahrens. Die Ergebnisse wurden auf einem vclass-Parallelcomputer der Firma Hewlett Packard ermittelt.

Ergebnisse adaptiver & paralleler Rechnungen

Zusammen mit Gitteradaption und Parallelisierung war es möglich, einige Beispiele mit besonders hoher Auflösung zu berechnen. In den folgenden Beispielen wurde das Rechengitter stets alle fünf Zeitschritte mit dem Variationsindikator (4.1) adaptiert. Durch dreimaliges „Verschmieren“ der Indikatorwerte mit der Abbildungsfolge (4.4) wurden die fein aufgelösten Bereiche

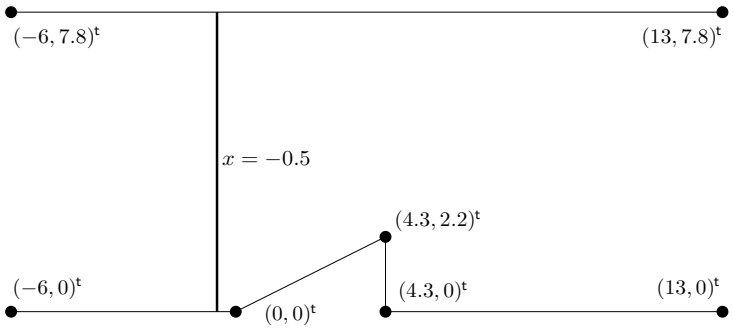


Abbildung 4.5: Geometrie für das Beispiel der Interaktion einer Stoßfront mit einem Keil. Die Anfangsposition des Stoßes ist zusätzlich eingezeichnet.

hinreichend vergrößert. Das ordnungserhaltende Limitierungsverfahren von Seite 55 wurde mit quadratischen Polynomen als Rekonstruktionsverfahren verwendet.
In der Abbildung 4.6 ist ein sehr fein aufgelöstes Ergebnis für das Beispiel der Reflexion einer Stoßfront von Seite 73 zu sehen. Bei dieser Auflösung ist besonders gut die Instabilität längs der Kontaktunstetigkeit zu beobachten. Hier und an Stößen wird die maximale Feinheit des Gitters erreicht: Die kürzesten Kantenlängen der Dreiecke sind 1/1920, wobei die Breite des Gebietes 3 beträgt.
In Abbildung 4.7 auf Seite 118f. ist die Interaktion einer Stoßfront mit einem Keil zu sehen. Dieses Beispiel wurde den Fotografien aus [Dyk82, Abbildung 241] nachempfunden, wobei die genauen Anfangsparameter nicht bekannt waren und daher kleinere Abweichungen in den numerischen Ergebnissen zu sehen sind. Der Fall wurde als symmetrisch angenommen und in der numerischen Rechnung wurde daher nur die obere Hälfte des Gebietes berücksichtigt. Die geometrischen Daten sind in der Abbildung 4.5 angegeben. Die Anfangsposition der Stoßfront war bei $x = -0.5$. Auf der rechten Seite wurde der folgende Ruhezustand gesetzt:

$$(\rho, \mathsf{v}_x, \mathsf{v}_y, \mathsf{p}) = (1.4, 0, 0, 1).$$

Links der Stoßfront wurde der Zustand

$$(\rho, \mathsf{v}_x, \mathsf{v}_y, \mathsf{p}) = (2.36551724, 0.57142857, 0, 2.12).$$

verwendet. Dies ergibt eine Stoßgeschwindigkeit von 1.4-facher Schallgeschwindigkeit relativ zum ruhenden Medium. Am oberen und unteren Rand wurde jeweils die Randbedingung für feste Wände gewählt. Die Ergebnisse zur Zeit $T = 8.5$ für eine parallele Rechnung zusammen mit Gitteradaption

sind sind in der Abbildung 4.7 dargestellt. Die kürzesten Kantenlängen der Dreiecke betragen 1/160, wobei die Breite des Gebietes 19 beträgt. Besonders gut sieht man die Struktur des Wirbels in der oberen Abbildung von Seite 119. Hier wurde in Graustufen die zweite Ableitung $\Delta\rho$ dargestellt, um einen qualitativen Vergleich mit den Schattenaufnahmen in [Dyk82, Abbildungen 241 und 81] zu ermöglichen.

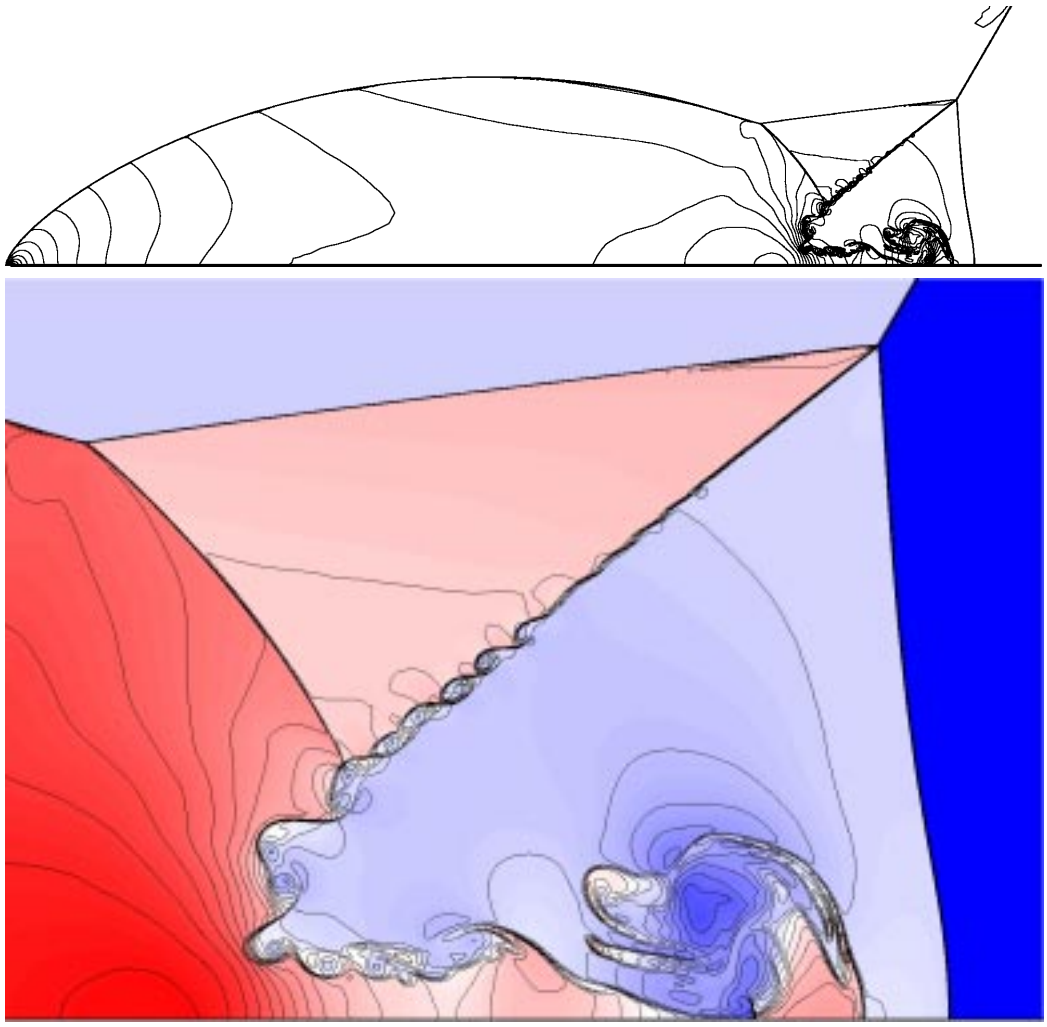


Abbildung 4.6: Reflexion eines Stoßes im Winkel von 30°; siehe die Erläuterungen ab Seite 73. In beiden Bildern sind 36 Isolinien von 1.39 bis 22.39 der Dichteverteilung zur Zeit $T = 0.2$ dargestellt. Unten wurden hohe Werte der Dichte rot unterlegt und niedrige Werte blau.

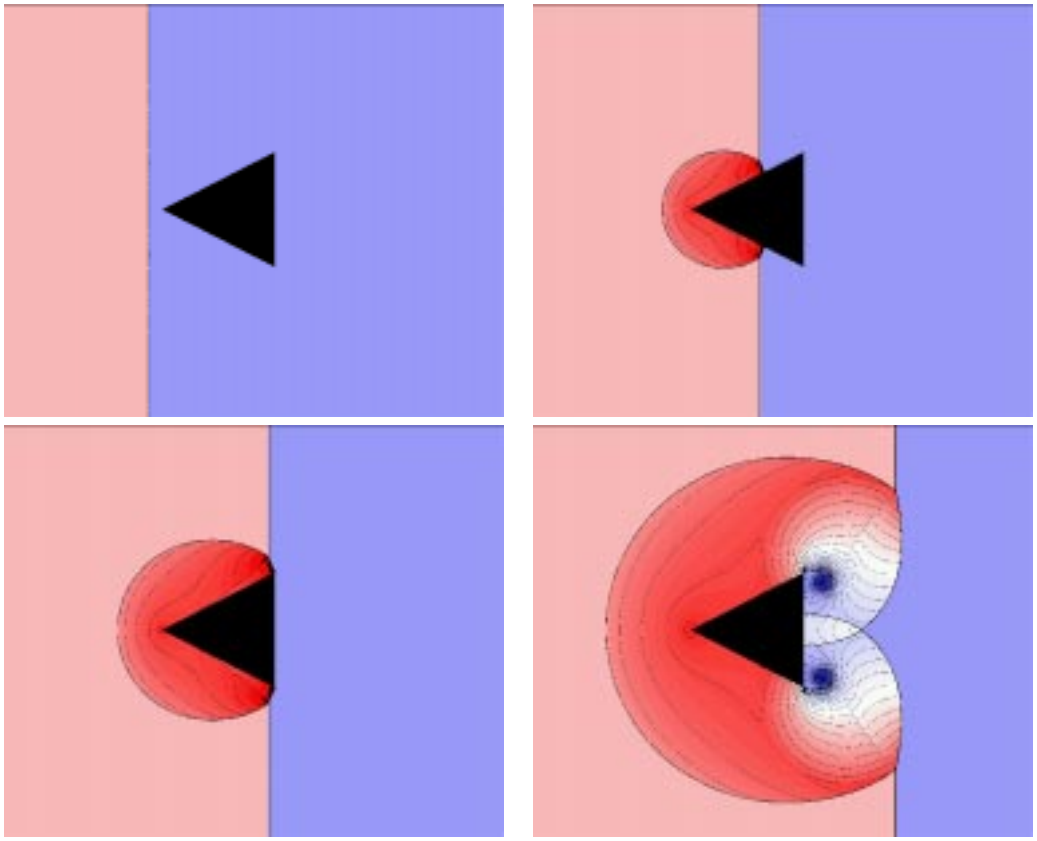
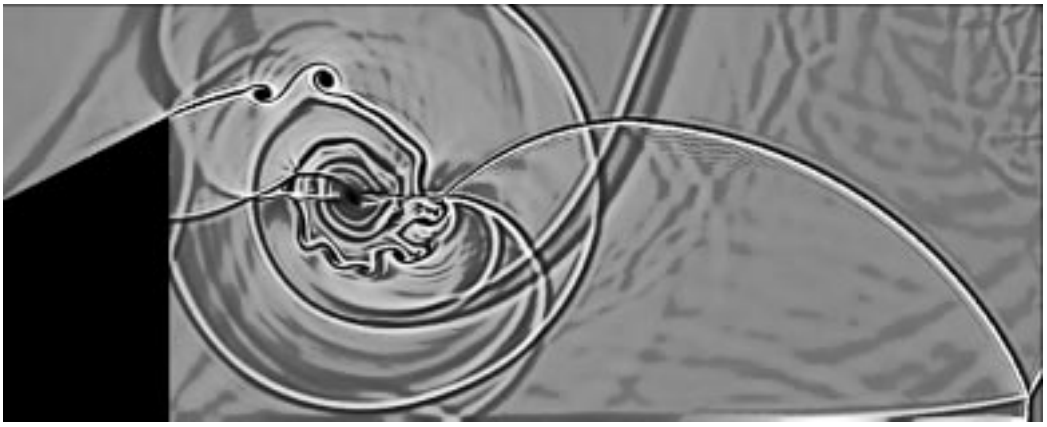
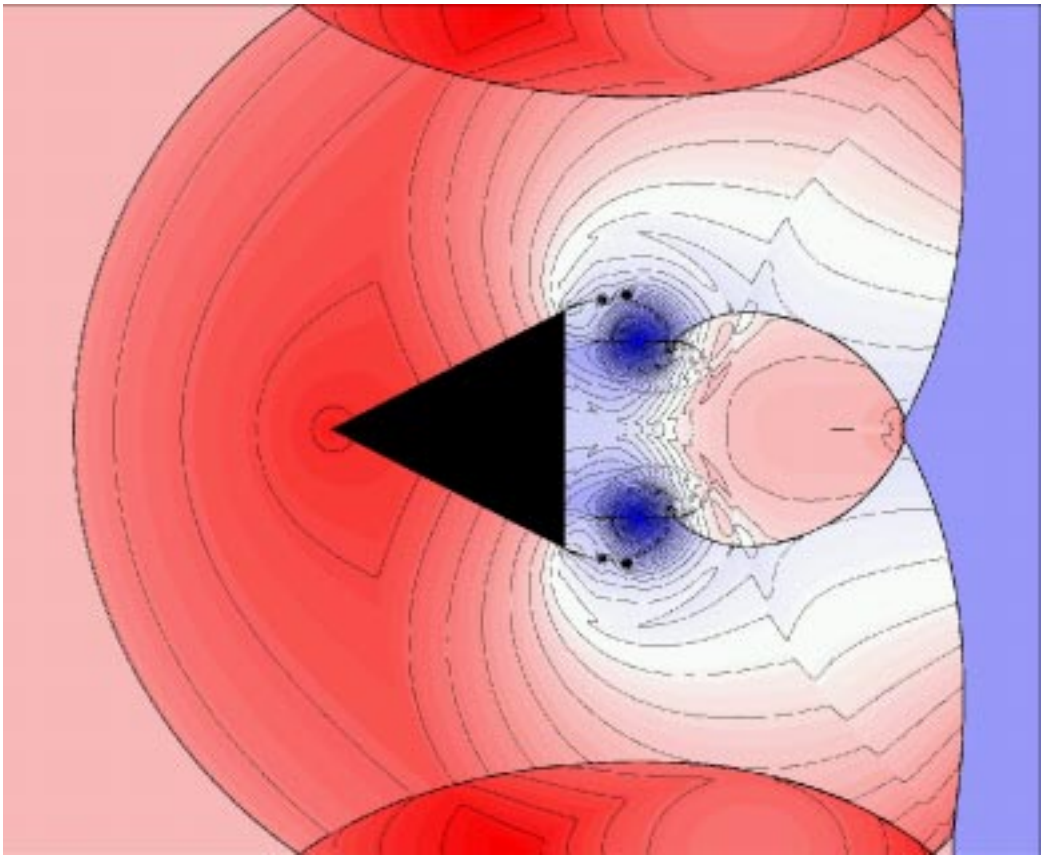


Abbildung 4.7: Interaktion einer Stoßfront mit einem Keil. Hier und auf der nachfolgenden Seite ist die Entwicklung der Dichteverteilung bis zur Zeit $T = 8.5$ dargestellt (36 Höhenlinien von 0.241 bis 2.939). Hohe Werte sind rot, niedrige blau unterlegt. Um einen Vergleich mit fotografischen Aufnahmen aus [Dyk82, Abbildung 241] zu ermöglichen, ist in der oberen Abbildung der nächsten Seite $\Delta\rho$ in Grauwerten dargestellt.





Zusammenfassung und Ausblick

In dieser Arbeit werden die expliziten Finite-Volumen-Methoden konsequent für den Fall unstrukturierter Gitter behandelt. Die verwendeten Standardtechniken sind im ersten Kapitel der Arbeit dargestellt.
Im zweiten Abschnitt werden polynomielle Rekonstruktionsverfahren diskutiert. Erst wird die Aufgabenstellung behandelt, eine Rekonstruktion möglichst hoher Fehlerordnung aus gegebenen Daten zu berechnen. Um dies in einem numerischen Verfahren durchführen zu können, benötigt man elementare Formeln zur Auswertung und Integration von Polynomen. Für bivariate Polynome allgemeinen Grades und für allgemeine polygonale Zellen werden Algorithmen hierfür beschrieben.
Die mit linearen Rekonstruktionsverfahren berechneten Funktionen oszillieren sehr stark in Bereichen von Unstetigkeiten der Lösungen. Um diese Oszillationen zu dämpfen, müssen die berechneten Rekonstruktionen limitiert werden. In der Arbeit werden für den Fall unstrukturierter Gitter das Verfahren von Barth und Jespersen [BJ89], das gewichtete ENO-Verfahren aus [Fri97] und ein neues Limitierungsverfahren verglichen. Für das neue Limitierungsverfahren wird gezeigt, daß es in glatten Bereichen der Lösung die Rekonstruktionsordnung erhält. Für einen einfachen Spezialfall kann zudem die ENO-Eigenschaft für diesen Ansatz nachgewiesen werden. Das Verfahren liefert in den Anwendungen mit quadratischen Polynomen gute Ergebnisse und ist zudem relativ schnell. Allerdings ist es bisher nicht gelungen, die Strategie auf kubische Polynome mit Gewinn zu übertragen. Hier liegt möglicherweise ein prinzipielles Problem, denn in dem Verfahren werden stets einzelne Punktwerte der Rekonstruktionen verglichen und hieraus das Verhalten der Rekonstruktion innerhalb der gesamten Zelle gesteuert. Zwar wird in dem Limitierungsansatz mit einem Auswahlkriterium vermieden, daß kleine Schwankungen große Änderung bewirken können, jedoch beruht das Kriterium auf der Äquivalenz von L^1- und Maximumsnorm in endlichen Polynomräumen. Diese Relation verschlechtert sich mit zunehmendem Polynomgrad und darin ist auch die Grenze der beschriebenen Strategie zu sehen.
Die Geschwindigkeitsvergleiche ergeben bislang, daß die WENO-Methoden

numerisch teurer sind als das beschriebene Limitierungsverfahren. Dennoch beinhalten die ENO-Verfahren meiner Ansicht nach das größere Entwicklungspotential. Im letzten Abschnitt des zweiten Kapitels werden deswegen zwei mögliche Beschleunigungsansätze beschrieben. Einen weiteren Ansatz verfolgt [Oll97], indem er die bisherige Trennung zwischen linearer Rekonstruktion, Oszillationsmessung und Konvexkombination der Rekonstruktionen aufhebt und direkt ein Ausgleichsproblem mit einer datenabhängigen Norm vorschlägt. Zwar konnte bisher nicht gesichert werden, daß das entstehende System wohlkonditioniert bleibt, jedoch beeinhaltet die Strategie die zukunftweisende Idee, das ENO-Verfahren als geeignete Pivotisierungsstrategie bei der Lösung von Ausgleichsproblemen zu realisieren. Möglicherweise wird es mit ähnlichen Ansätzen gelingen, ENO-Verfahren für allgemeine Hermite-Daten zu konstruieren, wie sie in unstetigen Galerkin-Methoden anfallen.

Im dritten Kapitel der Arbeit wird die Theorie der optimalen Rekonstruktion aus [GW59] und [MR84] übertragen auf den Fall des Finite-Volumen-Verfahrens. Diese Theorie führt in geeigneten Räumen zur Theorie der radialen Basisfunktionen. Es wird untersucht, ob mittels Interpolation mit diesen Funktionen eine Verbesserung der polynomiellen Rekonstruktionsverfahren möglich ist. Für die Berechnung der Interpolanten kann ein numerisch effizientes Schema angegeben werden. Allerdings stellt die Auswertung der Interpolanten, deren Basisfunktionen sich aus Faltungsintegralen ergeben, ein so kostspieliges Unterfangen dar, daß im Bereich der expliziten Finite-Volumen-Methoden die polynomiellen Rekonstruktionsverfahren konkurenzlos bleiben. Diese Einschätzung wird sich unter Umständen ändern, wenn man zu Kollokationsmethoden wechselt (vergleiche hierzu die Arbeiten [AHS98] und [Ahr99]).

Im letzten Kapitel dieser Arbeit werden die Algorithmen zur Gitteradaption, wie ich sie bereits bei der Deutschen Forschungs- und Versuchsanstalt für Luft- und Raumfahrt begonnen habe, weiterentwickelt. Die Veränderungen bestehen in einem besseren und schnelleren lokalen Vergröberungsalgorithmus, einer Interpolationstechnik für beliebige Polynomgrade und einem neuen Adaptionsindikator, der sehr gute Ergebnisse in der Praxis liefert. Bisher bestand bei der adaptiven Berechnung instationärer Phänomene das Problem, daß sich fein aufgelöste Phänomene bis zur nächsten Gitteradaption in grobe Gitterbereiche hineinbewegen konnten. Dieses Problem wird durch einen einfachen Algorithmus gelöst.

Insgesamt wird das lokale Adaptionsverfahren so weit entwickelt, daß im Vergleich zu global feinen Gittern erhebliche Beschleunigungen und gleichzeitig auch beachtliche Datenkompressionen erreichen lassen, ohne daß die Lösungen hierdurch wesentlich in ihrer Qualität beeinträchtigt werden.

Schließlich wird im letzten Abschnitt der Arbeit eine einfache Strategie zur Parallelisierung des Finite-Volumen-Verfahrens beschrieben.

Symbolverzeichnis

a	Schallgeschwindigkeit (Seite 15)
$\mathcal{A}$	Affine Abbildung
$\mathrm{Abb}(X \to Y)$ $= Y^X$	Menge aller Abbildungen von X nach Y
A, B, C	Matrizen
card	Anzahl an Elementen in einer endlichen Menge, Kardinalität
cond	Kondition einer Matrix oder eines Gitters (Seite 40)
d	Raumdimension des Gebietes Ω
∂_t	partielle Ableitung nach der Variablen t
$\nabla_x \cdot$	Divergenz eines Vektorfeldes
δ_ω	Zellmittelung zur Zelle ω (Seite 16)
δ_x	Auswertung an der Stelle x (Seite 35)
diag	Diagonalmatrix zu einem gegebenen Vektor
diam	Durchmesser einer Menge (Seite 40)
E	Totalenergie eines Fluids (Seite 14)
$\mathcal{G}$	Finite-Volumen-Gitter (Seite 16)
η	Zelle in der näheren Umgebung einer Zelle ω
H	Hilbertraum
$H(u, v, \mathsf{n})$	numerische Flußfunktion (Seite 22)
H	Heaviside-Funktion (Seite 22)
κ	Gaskonstante (Seite 14)
λ	lineares, stetiges Funktional
λ_i	Eigenwerte der Flußfunktion (Definition 1.2, Seite 11)
$\mathcal{L}$	Lokales Gitter um eine Zelle ohne die Zelle selbst (Seite 42)
$\overline{\mathcal{L}}$	Lokales Gitter um eine Zelle inklusive der Zelle (Seite 42)
n	Normalenvektor am Rande einer Zelle
$\mathcal{N}$	Angrenzende Nachbarzellen einer Zelle (Seite 54)
$\overline{\mathcal{N}}$	$\mathcal{N}$ inklusive der Zelle selbst (Seite 54)
$\mathbf{N}$	Menge der natürlichen Zahlen $1, 2, \ldots$
$\mathbf{N}_0$	$0, 1, 2, \ldots$
N	Radikal eines Semiskalarproduktes (Seite 84)

ω	Zelle eines Finite-Volumen-Gitters
Ω	Räumliches Gebiet
p	Druck (Seite 14)
P_i	Eckpunkte eines Polygons
P_i	Gitterpunkt einer Triangulierung
$\wp$	Polynom
π_0	konstante Funktion 1
$\pi_0, \ldots, \pi_{Q-1}$	Basis des Polynomraumes
Π^q	Raum der Polynome, deren Grad höchstens q ist
q	Quadraturpunkt am Rande einer Zelle
q	Höchstgrad der Polynome
Q	Dimension des Polynomraumes
r_i	Eigenvektoren der Flußfunktion (Definition 1.2, Seite 11)
r_η	Rieszsche Darstellungen von δ_η
$r(T)$	Adaptionsindikator für ein Dreieck (Seite 105)
$r(P)$	Adaptionsindikator für einen Punkt (Seite 106)
$\mathbf{R}$	Menge der reellen Zahlen
$\mathsf{TV}_\omega(v)$	Totalvariation einer skalaren Funktion v in einem Intervall ω.
ρ	Dichte eines Fluids (Seite 14)
s	Dimension des Zustandsraumes S
S	Zustandsraum (Seite 9)
sig	Sortierung der Eigenwerte einer Flußfunktion (Seite 30)
A^t	Transponierte der Matrix A
$\tau_1, \ldots, \tau_{d-1}$	Orthonormalsystem senkrecht zum Normalenvektor n
T	Zeit, bis zu der gerechnet werden soll (Seite 9)
T	Dreieck
u_0	Anfangsverteilung der Erhaltungsgleichung (Seite 9)
u	„exakte Lösung“ eines Problems
u, v, w	Zustandsverteilungen
U, V	Funktionenräume
v	Geschwindigkeit (Seite 14)
v^+, v^-	einseite Zustandswerte in einem Quadraturpunkt (Seite 22)
x	Punkt in $\mathbf{R}^d$
x, y	Koordinaten in $\mathbf{R}^2$
ξ_ω	Charakteristische Funktion einer Zelle ω
$\mathbf{Z}$	Menge der ganzen Zahlen
■	Beweisende

Literaturverzeichnis

[Abg94] R. Abgrall, *On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation.* J. Comp. Phys. 114, 45 (1994)

[AS97] R. Abgrall, Th. Sonar, *On the use of Mühlbach expansions in the recovery step of ENO methods.* Numer. Math., 76, 1–25 (1997)

[AHS98] A. Ahrend, D. Hempel, Th. Sonar, *Radial Basis Function Implementation of Meshless Collocation Methods.* eingereicht ZAMM, (1998)

[Ahr99] A. Ahrend, *Dissertation in Vorbereitung.* Universität Hamburg, (1999)

[Bän91] E. Bänsch, *Local Mesh Refinement in 2 and 3 Dimensions.* IMPACT Comput. Sci. Eng., 3, 181-191 (1991)

[BJ89] T.J. Barth, D.C. Jespersen, *The Design and Application of Upwind Schemes on Unstructured Meshes.* AIAA-89-0366, (1989)

[Bot95] N. Botta, *Numerical investigations of two-dimensional Euler flows: cylinder at transonic speed.* Dissertation ETH Zürich No. 10852, (1995)

[CFL28] R. Courant, K.O. Friedrichs, H. Lewy, *Über die partiellen Differenzengleichungen der mathematischen Physik.* Math. Ann. 100, 32–74, 1928

[Dyk82] M. Van Dyke, *An Album of Fluid Motion.* The Parabolic Press, Stanford (1982)

[EO81] B. Engquist, S. Osher, *One-Sided Difference Approximations for Nonlinear Conservation Laws.* Math. Comp., 36, 321–351 (1981)

[Fri93] O. Friedrich, *A new method for generating inner points of triangulations in two dimensions.* Comp. Meth. App. Mech. Eng. 104, 77–86 (1993)

[Fri97] O. Friedrich, *Weighted Essentially Non-oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids.* J. Comp. Phys. 144, 194–212, (1998)

[Fri99] O. Friedrich, *Dissertation in Vorbereitung.* Universität Hamburg, (1999)

[FHMS96] O. Friedrich, D. Hempel, A. Meister, Th. Sonar, *Adaptive Computation of Flow Fields with the DLR-τ-Code.* AGARD Conference Procedings CP-578, Seite 37.1–37.11 (1996)

[GW59] M. Golomb, H.F. Weinberger, *Optimal Approximation and Error Bounds.* The University of Wisconsin Press, Madison, in: On Numerical Approximation, Editor: R.E. Langer, (1959)

[Gut98] T. Gutzmer, *AMCIT – A New Method for Mesh Adaption when Solving Time-Dependent Conservation Laws.* Dissertation ETH No. 12452, Zürich (1998)

[HHS93] V. Hannemann, D. Hempel, T. Sonar, *Adaptive Computation of Compressible Flow Fields with the DLR τ-Code.* International Workshop on Numerical Methods for the Navier-Stokes Equations, Heidelberg, Oktober (1993)

[HHS94] V. Hannemann, D. Hempel, T. Sonar, *Dynamic Adaptivity and Residual Control in Unsteady Compressible Flow Computation.* Math. Comput. Modelling Vol. 20, No 10/11, 201–213 (1994)

[HEOC87] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, *Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III.* J. Comp. Phys. 71, 231–303 (1987)

[Hem92] D. Hempel, *Local Mesh Adaption in Two Space Dimensions.* IMPACT Comput. Sci. Eng., 5, 309–317 (1992)

[Hem96] D. Hempel, *Isotropic refinement and recoarsening in two dimensions* Numerical Algorithms 13, 33 - 43 (1996)

[Heu92] H. Heuser, *Funktionalanalysis.* B.G. Teubner Stuttgart, 3. Auflage, (1992)

[HS99] C. Hu, C.-W. Shu, *Weighted Essentially Non-Oscillatory Schemes on Triangular Meshes.* J. Comp. Phys. , 150, 97-127 (1999)

[IS96] A. Iske, Th. Sonar, *On the Structure of Function Spaces in Optimal Recovery of Point Data for ENO-Schemes by Radial Basis Functions.* Num. Math. 74, 177–201 (1996)

[JS96] G.-S. Jiang., C.-W. Shu, *Efficient Implementation of Weighted ENO Schemes.* J. Comp. Phys. 126, 202–228, (1996)

[LOC94] X-D. Liu, S. Osher, T. Chan, *Weighted Essentially Non-Oscillatory Schemes.* J. Comp. Phys. 115, 200–212, (1994)

[Mei79] J. Meinguet, *An intrinsic approach to multivariate spline interpolation at arbitrary points.* Polynomial and Spline Approximation, Dodrecht:Reidel, 163–190 (1979)

[Mei96] A. Meister, *Zur zeitgenauen numerischen Simulation reibungsbehafteter, kompressibler Strömungsfelder mit einer impliziten Finite-Volumen-Methode vom Box-Typ.* Dissertation, Darmstadt (1996)

[MR84] C.A. Micchelli, T.J. Rivlin, *Lectures on Optimal Recovery.* Springer Lecture Notes in Mathematics, 1129, 12–93 (1984)

[Müh78] G. Mühlbach, *The General Recurrence Relation for Divided Differences and the General Newton-Interpolation-Algorithm With Applications to Trigonometric Interpolation.* Numer. Math., 32, 393–408 (1979)

[Oll97] C. F. Ollivier-Gooch, *Quasi-ENO Schemes for Unstructured Meshes Based on Unlimited Data-Dependent Least-Squares Reconstruction.* J. Comp. Phys. 133, 6–17, (1997)

[OS82] S. Osher, F. Solomon, *Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws.* Math. Comp., Vol. 36, No 158, 339–374 (1982)

[Pow94] M.J.D. Powell, *The Uniform Convergence of Thin Plate Spline interpolation in two dimensions.* Num. Math. 68, 107–128 (1994)

[Sch88] H.R. Schwarz, *Numerische Mathematik* B. G. Teubner Stuttgart (1988)

[SO88] C.-W. Shu, S. Osher, *Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes.* J. Comp. Phys., 77, 439–471 (1988)

[SS94] Th. Sonar, E. Süli, *A Dual Graph Norm Refinement Indicator for Finite Volume Approximations of the Euler Equations.* Num. Math., 78, 619–658 (1998)

[Son96] Th. Sonar, *Optimal Recovery Using Thin Plate Splines in Finite Volume Methods for the Numerical Solution of Hyperbolic Conservation Laws.* IMA J. Num. Anal., 16, 549–581 (1996)

[Son97a] Th. Sonar, *Mehrdimensionale ENO-Verfahren.* B.G. Teubner, Stuttgart (1997)

[Son97b] Th. Sonar, *On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection.* Comp. Meth. Appl. Mech. Eng., 140, 157–181 (1997)

[Son97c] Th. Sonar, G. Warnecke, *On Finite Difference Error Indication for Adaptive Approximations of Conservation Laws.* Hamburger Beiträge, Reihe A, Preprint 122 (1997)

[Spe87] S.P. Spekreijse, *Multigrid Solution of the Steady Euler Equations.* Dissertation, Centrum voor Wiskunde en Informatica, Amsterdam (1987)

[Sto94] J. Stoer, *Numerische Mathematik 1.* Springer Verlag , 7. Auflage, Berlin Heidelberg New York (1994)

[SW93] R. Schaback, Z. Wu, *Local Error Estimates for Radial Basis Function Interpolation of Scattered Data.* IMA J. Num. Anal., 13, 13–27 (1993)

[Wen95] H. Wendland, *Piecewise Polynomial, Positive Definite and Compactly Supported Radial Basis Functions of Minimal Degree.* Adv. Comp. Math., 4, 389–396 (1995)

[Wen96] H. Wendland, *Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger.* Dissertationsschrift, Universität Göttingen, (1996)

[WC84] P. Woodward, P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks.* J. Comp. Phys., 54, 115–173 (1984)

Danksagung

Mein erster Gedanke geht an Professor Dr. Thomas Sonar, der nahezu meine gesamte wissenschaftliche Laufbahn nicht nur begleitet, sondern auch mitgestaltet hat. Für sein Engagement, die vielen Anregungen und Möglichkeiten, die er mir gegeben hat, und die schöne Zeit in Göttingen, Dresden, Südfrankreich und Hamburg möchte ich ihm herzlich danken.
Herrn Professor Dr. Klaus Glashoff danke ich für die freundliche Übernahme des Korreferates.
Oliver Friedrich danke ich nicht nur für seine wunderbaren kleinen und großen Programme, die er mir zur Verfügung gestellt hat, sondern auch für die kreativen Jahre der Zusammenarbeit, in denen wir an Dem Verfahren parallel und adaptiv gebastelt haben.
Arne Ahrend hatte über ein Jahr lang das Los, mit mir ein Büro und ein Projekt zu teilen. Teilweise bis spät in die Nacht und in unterschiedlichen Pizzerien haben wir über die Berechnung von Divergenzen nachgedacht. Für die amüsante Zeit und die Korrekturen zu dieser Arbeit möchte ich mich bedanken.
Friederike Schröder-Pander möchte ich für das unerfreuliche, aber sehr hilfreiche Korrekturlesen danken.
Meiner Freundin Birgit Pollak, meiner Schwester Julia Wiltinger und meinen Eltern Regina und Wolf Hempel danke ich für ihre zahlreichen Hilfestellungen und die große Geduld, die sie mit mir gehabt haben — jetzt wird alles anders!

Die vorliegende Arbeit entstand im Rahmen des DFG-Projektes So-363/1-1. Der Deutschen Forschungsgemeinschaft DFG danke ich für die finanzielle Unterstützung.

Daniel Hempel

Zusammenfassung

In der Arbeit werden explizite Finite-Volumen-Methoden für unstrukturierte Gitter behandelt. Hiermit werden hyperbolische Erhaltungsgleichungen, insbesondere die Euler-Gleichungen der Gasdynamik, diskretisiert. Im Vordergrund steht die Steigerung der Genauigkeit von Finite-Volumen-Verfahren, für die zwei sich ergänzende Ansätze beschrieben werden: Einerseits die Steigerung der Fehlerordnung in hinreichend glatten Bereichen der Lösung im Rekonstruktionsschritt des Finite-Volumen-Verfahrens und andererseits die lokale Gitteradaption zur verbesserten Auflösung von Unstetigkeiten.
Zur Steigerung der Fehlerordnung werden polynomielle Rekonstruktionsverfahren untersucht. Hier werden lineare Algorithmen angewendet, die allerdings in Bereichen von Unstetigkeiten zu stark oszillierenden Rekonstruktionen führen. Zur Dämpfung der Oszillationen wird ein neues Limitierungsverfahren vorgestellt, mit dem in hinreichend glatten Bereichen der Lösungen die hohe Fehlerordnung einer zuvor berechneten Rekonstruktion erhalten bleibt. Einer Analyse dieses Verfahrens schließt sich ein Vergleich in numerischen Beispielen mit anderen Limitierungsansätzen und den gewichteten ENO-Verfahren an.
Weiterhin wird die Theorie der optimalen Rekonstruktion und der radialen Basisfunktionen auf das Rekonstruktionsproblem im Finite-Volumen-Verfahren übertragen. Die Untersuchungen zeigen, daß für den Fall unstrukturierter Gitter diese Ansätze ohne einen qualitativen Gewinn deutlich kostenintensiver als polynomielle Verfahren sind.
In der Arbeit wird die lokale Gitteradaption für zweidimensionale Triangulierungen beschrieben. Es werden neue Algorithmen zur Indikation von Unstetigkeiten, zur Interpolation zwischen einzelnen Gittern sowie zur lokalen Verfeinerung und Vergröberung von Triangulierungen vorgestellt. Anhand von Anwendungsbeispielen wird demonstriert, daß die lokale Gitteradaption eine erhebliche Beschleunigung und Speicherersparnis gegenüber global feinen Gittern ermöglicht. Eine zusätzliche Beschleunigung des Finite-Volumen-Verfahrens kann durch Parallelisierung erreicht werden. Hierfür wird eine einfache Methode dargestellt.

Lebenslauf

Daniel Rudolf Hempel
geboren am 13. April 1969 in Kassel

Schule

Aug. 75 – Juli 79	Königstor-Grundschule Kassel
Aug. 79 – Juli 85	Albert-Schweitzer-Gymnasium Kassel
Aug. 85 – Juli 88	Herder-Gymnasium Kassel
Juli 88	Abitur

Zivildienst

Aug. 88 – März 90	Rettungshelfer in Kassel

Universität

Apr. 90 – Okt. 90	Mathematik- und Physikvorlesungen an der Universität Kassel
Okt. 90 – Juli 96	Diplomstudiengang Mathematik mit Nebenfach Informatik an der Georg-August-Universität Göttingen
April 92	Vordiplom in Mathematik
Juli 96	Diplom in Mathematik
Aug. 96 – Juli 99	DFG-Stipendium zur Promotion in angewandter Mathematik an der Universität Hamburg

Berufliche Tätigkeiten

Apr. 92 – Juli 96	Wissenschaftlicher Mitarbeiter bei der Deutschen Forschungs- und Versuchsanstalt für Luft- und Raumfahrt in Göttingen
Okt. 96 – Apr. 97	Lehrtätigkeit an der TU-Hamburg-Harburg