



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Michael Bergmann

Erweiterung eines 3D Discontinuous Galerkin Strömungslösers

*Fakultät Technik und Informatik
Studiendepartment Maschinenbau und Produktion*

*Faculty of Engineering and Computer Science
Department of Engineering and Manufacture*

Michael Bergmann
Matrikelnummer: 1984599

Erweiterung eines 3D Discontinuous Galerkin Strömungslösers

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science - Berechnung und Simulation im Maschinenbau

am Department Maschinenbau und Produktion
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

in Zusammenarbeit mit dem

Deutschen Zentrum für Luft- und Raumfahrt e.V.
Institut für Antriebstechnik
Numerische Methoden
Linder Höhe
51147 Köln

Betreuender Prüfer: Prof. Dr.-Ing. habil. Frank Ihlenburg
Zweitgutachter: Dr. Svetlana Drapkina

Eingereicht am: 14. Februar 2015

Michael Bergmann, Matr.-Nr.: 1984599

Thema der Arbeit

Erweiterung eines 3D Discontinuous Galerkin Strömungslösers

Stichworte

Discontinuous Galerkin, Strömungslöser, Euler-Gleichungen, TRACE

Kurzzusammenfassung

In dieser Masterarbeit wird ein 3D Discontinuous Galerkin Strömungslöser um Ansätze für Hexaeder-Elemente bis zur vierten Ordnung erweitert. Die Implementierungen werden an geeigneten Beispielen validiert und verifiziert. Dabei wird insbesondere der Einfluss numerischer Parameter, z.B. der Approximationsordnung auf die Genauigkeit und die Konvergenzordnung der numerischen Lösung untersucht.

Title of the paper

Extension of a 3D discontinuous Galerkin flow solver

Keywords

discontinuous Galerkin, flow solver, Euler equations, TRACE

Abstract

In this master thesis hexahedral elements up to fourth order are added to a 3D discontinuous Galerkin flow solver. The implementations will be tested through appropriate examples. Especially the influence of numerical parameters are investigated.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Hintergrund	2
1.2	Zielsetzung	3
2	Euler-Gleichungen	4
2.1	Erhaltungsprinzipien	4
2.1.1	Erhaltung der Masse	4
2.1.2	Erhaltung des Impulses	4
2.1.3	Erhaltung der Energie	5
2.2	Kompakte Schreibweise	6
2.3	Zustandsgleichung und Machzahl	7
3	Discontinuous-Galerkin-Verfahren	9
3.1	DG-Ortsdiskretisierung der Euler-Gleichungen	9
3.1.1	Numerischer Fluss	13
3.1.2	Randbedingungen	14
3.1.3	Vergleich zur Finiten-Volumen-Methode	15
3.2	Elemente und Basisfunktionen	16
3.2.1	Transformation auf das Referenzelement	16
3.2.2	Basisfunktionen	17
3.2.3	Numerische Integration	21
3.2.4	Gekrümmte Ränder	25
3.3	Zeitdiskretisierung	27
3.3.1	Explizite Verfahren	27
3.3.2	CFL-Bedingung	28
3.4	Fehlerabschätzung und Konvergenzordnung	29
4	Strömungslöser TRACE	31

5	Numerische Ergebnisse	35
5.1	Konvergenzstudie	35
5.1.1	Aufbau des Modells	35
5.1.2	Auswertung für DGNODAL	40
5.1.3	Auswertung für DGSEM	43
5.1.4	Vergleich von DGNODAL und DGSEM	45
5.2	Umströmung eines Zylinders mit gekrümmten Rändern	46
5.3	Umströmung eines Tragflügelprofils	50
6	Fazit und Ausblick	58

Abbildungsverzeichnis

3.1	Vergleich zwischen DG und FEM	11
3.2	Definition der Randwerte eines Elements	12
3.3	Eindimensionales Riemann-Problem	13
3.4	Referenzhexaeder	16
3.5	Lagrange-Basisfunktionen	19
3.6	Basisfunktionen höherer Ordnung	20
3.7	Knoten der kubischen Hexaeder-Elemente	21
3.8	Integrationspunkte und Knoten im Eindimensionalen	23
3.9	Normalenvektor der gekrümmten Oberfläche	26
4.1	Simulationsprozessablauf	32
4.2	Ablaufplan Runge-Kutta-DG-Solver	33
5.1	Initialisierung der Konvergenzstudie	37
5.2	Gauß-Puls auf dem Netz mit 54 Elementen	39
5.3	Initialisierungen auf dem Netz mit 2000 Elementen	40
5.4	Simulationsergebnisse auf dem Netz mit 2000 Elementen	41
5.5	Konvergenzraten der DGNODAL-Implementierung	42
5.6	Fehler über Freiheitsgrade und CPU-Zeit (DGNODAL)	43
5.7	Konvergenzraten der DGSEM-Implementierung	44
5.8	Vergleich von DGNODAL und DGSEM	45
5.9	Vernetzung des umströmten Zylinders	46
5.10	Mach-Kontur mit geraden Elementen	47
5.11	Vergleich der Knoten mit geraden und gekrümmten Ränder	47
5.12	Mach-Kontur mit gekrümmten Elementen	48
5.13	Entropie der Strömung um einen Zylinder	49
5.14	Druckbeiwerte am umströmten Zylinder	50
5.15	Vernetzung des NACA0012-Profiles	51
5.16	Mach-Konturen mit gekrümmten und geraden Rändern	52
5.17	Referenznetz und Referenzlösung mit dem FV-Solver	52
5.18	Vergleich der Entropie	53
5.19	Entropie-Kontur verschiedener Approximationsordnungen	54
5.20	Druckbeiwerte am NACA-Profil	55
5.21	Druckbeiwerte der Integrationspunkten am NACA-Profil	56

Tabellenverzeichnis

3.1	Überblick über DGSEM und DGNODAL	23
5.1	Verfeinerungsstufen der Konvergenzstudie	38

Symbolverzeichnis

E	totale spezifische Energie, Seite 5
H	totale spezifische Enthalpie, Seite 6
Ma	Machzahl, Seite 8
ρ	Dichte eines Fluides, Seite 4
\mathcal{H}	numerischer Fluss, Seite 11
$L^2(\Omega)$	Lebesgue-Raum mit $p = 2$, Seite 10
$Q_p^m(\Omega)$	diskreter Funktionsraum der Hexaeder-Elemente, Seite 10
\mathcal{T}_h	Zerlegung des Gebiets Ω , Seite 9
h	charakteristische Netzgröße, Seite 9
κ	Element der Zerlegung \mathcal{T}_h , Seite 9
$\partial\kappa$	Randflächen eines Elements κ , Seite 10
p	maximale Polynomordnung der Basisfunktionen, Seite 10
v_{ij}	Vandermonde-Matrix, Seite 18
m_{ij}	Massenmatrix, Seite 12
\mathbf{J}	Jacobi-Matrix, Seite 17
\mathbf{g}	Vektor der Erdbeschleunigung = $(g_x, g_y, g_z)^T$, Seite 5
\mathbf{h}	numerischer Flussvektor, Seite 12
\mathbf{n}	Normalenvektor = $(n_x, n_y, n_z)^T$, Seite 10
\mathbf{F}	konservative Flussmatrix, Seite 7
\mathbf{q}	Vektor der konservativen Variablen, Seite 7
\mathbf{r}	Residuenvektor, Seite 12
\mathbf{s}	Steifigkeitsterm (Euler-Fluss), Seite 12
\mathbf{u}	Geschwindigkeitsvektor = $(u, v, w)^T$, Seite 4
\mathbf{x}	physikalische Koordinaten = $(x, y, z)^T$, Seite 16
ξ	natürliche Koordinaten = $(\xi, \eta, \zeta)^T$, Seite 16

Abkürzungsverzeichnis

CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CGNS	CFD General Notation System
DG	Discontinuous-Galerkin
DGNODAL	Nodales Discontinuous-Galerkin-Verfahren
DGSEM	Discontinuous-Galerkin-Spectral-Element-Method
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DOF	Degrees of Freedom
ERK	Explizites Runge-Kutta-Verfahren
FDM	Finite-Differenzen-Methode
FE	Finite-Elemente
FEM	Finite-Elemente-Methode
FV	Finite-Volumen
FVM	Finite-Volumen-Methode
GMC	General Mesh Connector
LG	Legendre-Gauss
LGL	Legendre-Gauss-Lobatto
TRACE	Turbomachinery Research Aerodynamics Computational Environment

1 Einleitung

Die stetig wachsenden Ansprüche an moderne Turbomaschinen, im Hinblick auf die Einsparung von Treibstoff und die Reduktion der Emission, machen es unverzichtbar, genaue Kenntnisse über das Verhalten der Strömung zu haben. Neben der analytischen Theorie und den experimentellen Untersuchungen hat sich die numerische Strömungsmechanik (CFD) in den letzten Jahren als ein wichtiges Werkzeug zur Beantwortung dieser Fragestellung herausgebildet. Insbesondere durch steigende Rechenleistung und immer effizienteren numerischen Methoden rückt die CFD in den Fokus des Auslegungsprozesses und ersetzt bis zu einem gewissen Grad teure und aufwändige experimentelle Untersuchungen.

Der Ausgangspunkt eines numerischen Verfahrens ist die mathematische Formulierung eines physikalischen Problems in partiellen Differentialgleichungen. Da diese Gleichungen im Allgemeinen rein analytisch nicht gelöst werden können, werden numerische Methoden, wie zum Beispiel die Finite-Differenzen- (FDM), die Finite-Volumen- (FVM) oder die Finite-Elemente-Methode (FEM), eingesetzt. Die zugrunde liegenden Differentialgleichungen der Strömungsmechanik sind die Navier-Stokes-Gleichungen oder mit Vernachlässigung der Viskosität (Turbulenz) die Euler-Gleichungen. Zur Ortsdiskretisierung hat sich die FVM als Stand der Technik herausgebildet, die gewöhnlich auf ein Verfahren zweiter Ordnung $\mathcal{O}(h^2)$ führt. Zweite Konvergenzordnung bedeutet hier, dass sich bei einer Halbierung der Netzgröße h der Fehler der numerischen Lösung viertelt. Wünschenswert wäre natürlich eine möglichst hohe Konvergenzordnung um ähnliche Fehlergrößen auf vergleichsweise groben Netzen zu erreichen. In der Regel reduziert sich dabei die Rechenzeit deutlich gegenüber Verfahren niedriger Ordnung, vgl. [42]. Von Verfahren höherer Ordnung spricht man ab einer Genauigkeitsordnung von $\geq \mathcal{O}(h^3)$.

Auf strukturierten Netzen wurden in den letzten Jahren viele Verfahren höherer Ordnung entwickelt, die den Raum mithilfe der FDM diskretisieren (siehe dazu z.B. [40] und [33]). Diese Verfahren sind oft an Effizienz und Einfachheit nicht zu schlagen. Der Nachteil ist die Begrenzung auf strukturierte Netze, die für komplexe Geometrien oft nicht erstellt werden können. In den Fokus der Forschung sind deshalb Methoden auf unstrukturierten Netzen gerückt. Mithilfe von Rekonstruktionsformeln lassen sich mit der FV-Diskretisierung Verfahren höherer Ordnung auf unstrukturierten Netzen realisieren. Die Rekonstruktion der Lösung erfolgt durch die Einbeziehung mehrerer

umliegender Zellen (*stencil*) mit jeweils konstanten Lösungen. Daraus resultieren nach [37] häufig sehr aufwendige Implementierungen und hohe Rechenkosten.

Eine hoffnungsvolle Alternative zu den genannten Methoden ist das Discontinuous-Galerkin-Verfahren (DG). Diese Methode wurde als Erstes zur Lösung von Neutronentransportproblemen im Jahre 1973 von Reed und Hill angewendet [38] und erst kürzlich auf allgemeine Probleme der Strömungsmechanik ausgeweitet (siehe [4], [5], [11]). Das DG-Verfahren kann als eine Kombination aus der FVM und der FEM angesehen werden. Wie in der klassischen FEM werden die Test- und Basisfunktionen im gleichen Ansatzraum definiert, woher die Bezeichnung *Galerkin* stammt. Für die Approximationen gelten im Gegensatz zur FEM keine Stetigkeitsbedingungen, sodass Unstetigkeiten zwischen den Elementen auftreten können. Ähnlich der FVM führt dies zu einem numerischen Fluss, der für die guten Stabilitätseigenschaften des DG-Verfahrens für hyperbolische Erhaltungsgleichungen verantwortlich ist. Die Höhe der Konvergenzordnung richtet sich nun nach der Approximationsordnung innerhalb der Elemente und nicht mehr nach der Größe des *stencils*. Die Ordnung des Verfahrens kann dementsprechend beliebig gesteigert werden, ohne viele Elemente in der Approximation miteinzubeziehen. Daraus resultiert eine exzellente Eignung der DG-Diskretisierung für eine Parallelisierung des Codes. Weiterhin führt die fehlende Stetigkeitsbedingung zu nicht gekoppelten Elementen, wodurch jedes Element für sich betrachtet werden kann. Aufgrund dieser lokalen Formulierung ist das DG-Verfahren prädestiniert für eine automatische, lokale Netzverfeinerung und lokale Erhöhung der Approximationsordnung (hp-Adaption). Die angesprochenen Stärken und Möglichkeiten machen das DG-Verfahren zu einer interessanten numerischen Methode mit enormen Potential für die Strömungssimulation, siehe dazu [4], [31] und [29].

1.1 Hintergrund

Diese Masterthesis entstand in Zusammenarbeit mit dem Deutschen Zentrum für Luft- und Raumfahrt (DLR). In der Abteilung Numerische Methoden des Instituts für Antriebstechnik in Köln wird der Strömungslöser TRACE (Turbomachinery Research Aerodynamics Computational Environment) entwickelt. TRACE ist ursprünglich ein FV-Solver für die Lösung der instationären, dreidimensionalen Navier-Stokes-Gleichungen auf strukturierten und unstrukturierten Netzen. Neben weiteren Forschungsthemen wie zum Beispiel der Aeroelastik und der Turbulenzmodellierung, wird an der Implementierung des DG-Verfahrens zur Ortsdiskretisierung gearbeitet. Für die dreidimensionalen, instationären Euler-Gleichungen wurde die DG-Diskretisierung unter der Verwendung von Tetraeder-Elementen bereits realisiert und dient als Ausgangsbasis für diese Arbeit. Die Zeitdiskretisierung wird mittels einem expliziten Runge-Kutta-Verfahren vollzogen.

Da TRACE auch in der Industrie Anwendung findet, steht neben der Methode an sich auch immer die Effizienz, die Kompatibilität, die Integration in den vorhandenen Code und das Qualitätsmanagement im Fokus der Implementierungen.

1.2 Zielsetzung

Ausgangspunkt dieser Arbeit ist die in TRACE implementierte DG-Ortsdiskretisierung der Euler-Gleichungen mit Tetraeder-Elementen bis zur vierten Ordnung. In der Masterthesis soll der Code nun um nodale Ansätze für Hexaeder-Elemente bis zur vierten Ordnung erweitert werden. Neben den Implementierungen im Strömungslöser TRACE sind Routinen zur Auswertung der DG-Ergebnisdateien im Post-Processingtool POST zu schreiben. Weiterhin sind die Entwicklungen an geeigneten Testfällen zu validieren. Um die Abhängigkeit der Ordnung des Verfahrens auf die Approximationsordnung zu überprüfen, ist eine geeignete Konvergenzstudie durchzuführen. Außerdem soll die Implementierung der gekrümmten Ränder getestet werden. Abschließend ist mithilfe der DG-Diskretisierung die Strömung um ein Tragflügelprofil zu untersuchen. Parallel zu der Entwicklung der nodalen Hexaeder-Elemente innerhalb dieser Arbeit wird die DG-Spectral-Element-Method (DGSEM), welche spezielle Interpolations- und Integrationspunkte benutzt, in den Code implementiert. Diese beiden Verfahren sind durch geeignete numerische Untersuchungen miteinander zu vergleichen und auszuwerten.

2 Euler-Gleichungen

Die Euler-Gleichungen sind ein System von nichtlinearen, hyperbolischen Erhaltungsgleichungen. Sie stellen die mathematische Beschreibung der Strömung eines Fluides unter Vernachlässigung der Reibung dar und sind somit ein Sonderfall der Navier-Stokes-Gleichungen. Die Nichtberücksichtigung der viskosen Terme, die verantwortlich für die Reibung sind, ist für die meisten technischen Anwendungen nicht tragbar. Insbesondere in Randschichten wirkt sich die Viskosität sehr stark auf das Strömungsverhalten aus. Zur Entwicklung eines neuen numerischen Löser zur Strömungsberechnung ist es jedoch sinnvoll, in einem ersten Ansatz auf die schwer zu modellierenden viskosen Terme zu verzichten und diese im Anschluss hinzuzufügen.

Grundsätzlich basieren die Euler-Gleichungen auf drei Erhaltungsprinzipien: die Erhaltung der Masse, des Impulses und der Energie. Im Folgenden sollen diese kurz beschrieben werden [6].

2.1 Erhaltungsprinzipien

2.1.1 Erhaltung der Masse

Das Massenerhaltungsprinzip besagt, dass in einem abgeschlossenen Fluidsystem weder Masse erzeugt, noch vernichtet werden kann. Daraus ergibt sich die Kontinuitätsgleichung

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (2.1)$$

wobei ρ die Dichte des Fluids und $\mathbf{u} = (u, v, w)^T$ der dreidimensionale Geschwindigkeitsvektor ist.

2.1.2 Erhaltung des Impulses

Das zweite Newton'sche Axiom besagt, dass die zeitliche Änderung des Impulses eines Fluidteilchens gleich der Summe der am Massenelement angreifenden äußeren

Kräfte ist. Aus diesem Zusammenhang resultiert die Cauchy-Gleichung für die Impulserhaltung in Differentialform

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{g} + \nabla \cdot \mathbf{T}. \quad (2.2)$$

Hierbei bezeichnet $\mathbf{g} = (g_x, g_y, g_z)^T$ den Vektor der Erdbeschleunigung und \mathbf{T} den Spannungstensor. Der Spannungstensor beschreibt die auf die Oberfläche einwirkenden Kräfte und ist unter Annahme eines newton'schen Fluids

$$\mathbf{T} = - \left(p + \frac{2}{3} \mu \nabla \cdot \mathbf{u} \right) \mathbf{I} + \mu \mathbf{D}, \quad (2.3)$$

mit dem einwirkenden Druck p , der Viskosität μ , der Volumenänderung $\nabla \cdot \mathbf{u}$, der Einheitsmatrix \mathbf{I} und dem Deformationsgeschwindigkeitstensor \mathbf{D} [15]. Da bei der Betrachtung der Euler-Gleichungen die Viskosität vernachlässigt werden soll ($\mu = 0$), wird aus Gleichung 2.2

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{g} - \nabla p. \quad (2.4)$$

Häufig wird zusätzlich die Erdbeschleunigung nicht berücksichtigt, womit sich die endgültige, zu betrachtende Gleichung für die Impulserhaltung mit

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \mathbf{0} \quad (2.5)$$

ergibt.

2.1.3 Erhaltung der Energie

Der erste Hauptsatz der Thermodynamik besagt, dass jede Änderung der totalen Energie eines Systems gleich der Summe der verrichteten Arbeit und der zugeführten Wärme ist. Die totale Energie E eines Fluids ist dabei die Summe aus der spezifischen inneren Energie und der spezifischen kinetischen Energie

$$E = e + \frac{|\mathbf{u}|^2}{2}. \quad (2.6)$$

Die Erhaltungsgleichung für die Energie ist demnach

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho E \mathbf{u}) = \rho \mathbf{g} \cdot \mathbf{u} + \nabla \cdot (\mathbf{T} \cdot \mathbf{u}) + \nabla \cdot (\lambda \nabla T) + \rho \dot{s}. \quad (2.7)$$

Unter Vernachlässigung der Schwerkraft \mathbf{g} , der Wärmequellen \dot{s} und der Viskosität μ wird aus der Energieerhaltungsgleichung

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho E \mathbf{u} + p \mathbf{u}) = 0. \quad (2.8)$$

Üblicherweise wird die Gleichung 2.8 mithilfe der totalen spezifischen Enthalpie

$$H = E + \frac{p}{\rho}, \quad (2.9)$$

ausgedrückt und ergibt sich damit in der gebräuchlichen Form [15]

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho H \mathbf{u}) = 0. \quad (2.10)$$

2.2 Kompakte Schreibweise

Aus den drei Erhaltungsprinzipien resultiert das System der kompressiblen Euler-Gleichungen

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (2.11a)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \mathbf{0}, \quad (2.11b)$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho H \mathbf{u}) = 0. \quad (2.11c)$$

Durch die identische Struktur der gekoppelten Euler-Gleichungen kann man den Vektor der Erhaltungsgrößen

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \quad (2.12)$$

und die konservative Flussmatrix $\mathbf{F} = (F^x, F^y, F^z)$ mit

$$F^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho Hu \end{pmatrix}, \quad F^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \end{pmatrix}, \quad F^z = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho Hw \end{pmatrix} \quad (2.13)$$

aufstellen und die Euler-Gleichungen 2.11 in kompakter Weise formulieren:

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{0}. \quad (2.14)$$

2.3 Zustandsgleichung und Machzahl

Bisher beschreiben die fünf Erhaltungsgleichungen sechs unbekannte Größen: die drei Zustandsvariablen Dichte ρ , Druck p und innere Energie e , sowie die drei Geschwindigkeitskomponenten u, v, w . Damit daraus ein geschlossenes Gleichungssystem wird, benötigt man eine weitere Gleichung, die sogenannte Zustandsgleichung. Diese beschreibt den thermodynamischen Zusammenhang zwischen den Zustandsgrößen. Da in dieser Arbeit von einem idealen Gas ausgegangen werden soll, ergibt sich die Zustandsgleichung nach [15]

$$p = \rho RT, \quad (2.15)$$

mit der charakteristischen Gaskonstante R . Die Verknüpfung der inneren spezifischen Energie e und der Temperatur T ist für ideale Gase definiert durch

$$e = c_v T. \quad (2.16)$$

Hierbei ist c_v die spezifische Wärmekapazität bei konstantem Volumen. Mithilfe der spezifischen Wärmekapazität bei konstantem Druck c_p , welche die Enthalpie mit der Temperatur verknüpft ($H = c_p T$), lässt sich der Isentropenexponent für ein kalorisch ideales Gas definieren [36]

$$\gamma = \frac{c_p}{c_v}. \quad (2.17)$$

Er beschreibt die temperaturbedingte Gausdehnung und soll für die weiteren Betrachtungen mit dem für Luft üblichen Wert $\gamma = 1.4$ festgelegt werden.

Eine wichtige Größe für die Beschreibung der Strömung ist die Machzahl Ma , welche die Strömungsgeschwindigkeit mit der Schallgeschwindigkeit c in Verbindung setzt. Die Schallgeschwindigkeit eines idealen Gases ist definiert durch

$$c^2 = \gamma RT = \gamma \frac{p}{\rho}. \quad (2.18)$$

Damit ergibt sich für die Machzahl folgender Zusammenhang

$$Ma = \frac{|\mathbf{u}|}{\sqrt{\gamma \frac{p}{\rho}}}. \quad (2.19)$$

Unterhalb einer Machzahl von ca. $Ma < 0.3$ zeigen Strömungen ein näherungsweise inkompressibles Verhalten. Oberhalb dieser Machzahl gelten Strömungen als kompressibel und werden unterteilt in subsonische ($Ma < 1$), transsonische ($Ma \approx 1$), supersonische ($Ma > 1$) und hypersonische ($Ma > 5$) Strömungen. Die Einteilung in verschiedene Machzahl-Bereiche wird in Verbindung mit den Charakteristiken unter anderem zum Aufstellen der Randbedingungen benötigt [36].

3 Discontinuous-Galerkin-Verfahren

In dieser Arbeit wird das unstetige Galerkin Verfahren zur örtlichen Diskretisierung der in Kapitel 2 vorgestellten Euler-Gleichungen genutzt. Prinzipiell ist auch möglich, den DG-Ansatz sowohl für die Orts- als auch für die Zeitdiskretisierung zu nutzen, siehe dazu [7]. Für die zeitliche Diskretisierung soll in der Arbeit ein klassisches explizites Integrationschema verwendet werden, siehe Kapitel 3.3.

In diesem Kapitel wird nun das allgemeine Schema des DG-Verfahrens vorgestellt und für die Euler-Gleichungen angewendet. Im Weiteren wird auf die verwendeten Basisfunktionen, die geometrische Transformation, die numerische Integration sowie auf die Behandlung gekrümmter Elemente eingegangen. Abschließend soll ein kurzer Überblick über die Zeitdiskretisierung und eine mögliche Fehlerabschätzung gegeben werden.

3.1 DG-Ortsdiskretisierung der Euler-Gleichungen

Die dreidimensionalen, instationären Euler-Gleichungen sind nach Kapitel 2.2 definiert als

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{0}. \quad (3.1)$$

Ähnlich zu dem FE-Verfahren ist die Variationsformulierung der Ausgangspunkt der DG-Diskretisierung.

Dazu wird das Gebiet Ω in ein Netz mit regelmäßiger Form $\mathcal{T}_h = \{\kappa\}$ zerlegt, welches aus den Elementen $\kappa \in \mathcal{T}_h$ besteht. Es gilt

$$\Omega = \bigcup_{\kappa \in \mathcal{T}_h} \kappa. \quad (3.2)$$

Dabei bezeichnet h die stückweise konstanten Netzfunktionen, welche durch $h|_{\kappa} = \text{diam}(\kappa)$ für alle $\kappa \in \mathcal{T}_h$ definiert werden. Nun fordert man, dass die L^2 -Projektion der Differentialgleichungen (Residuum) auf einen Vektor \mathbf{v} , der die Testfunktionen beinhaltet, Null ist. Die Testfunktionen sollen dabei auf κ beliebig oft differenzierbar sein.

$L^2(\Omega)$ repräsentiert einen Lebesgue-Raum und beinhaltet die Funktionen, für die

$$\int_{\Omega} v^2 d\Omega < \infty$$

erfüllt ist [18].

Für die Projektion multipliziert man nun die Gleichung 3.1 mit \mathbf{v} und integriert über ein Element $\kappa \in \mathcal{T}_h$

$$\int_{\kappa} \mathbf{v} \cdot \frac{\partial \mathbf{q}}{\partial t} d\kappa + \int_{\kappa} \mathbf{v} \cdot (\nabla \cdot \mathbf{F}(\mathbf{q})) d\kappa = 0 \quad \forall \kappa \in \mathcal{T}_h, \quad \forall \mathbf{v}. \quad (3.3)$$

Die Anforderungen an \mathbf{q} bezüglich der Differenzierbarkeit sind noch stark. Um eine schwächere Formulierung zu erreichen, wird die Gleichung partiell integriert

$$\int_{\kappa} \mathbf{v} \cdot \frac{\partial \mathbf{q}}{\partial t} d\kappa - \int_{\kappa} \nabla \mathbf{v} : \mathbf{F}(\mathbf{q}) d\kappa + \int_{\partial\kappa} \mathbf{v} \cdot (\mathbf{F}(\mathbf{q}) \cdot \mathbf{n}) ds = 0. \quad (3.4)$$

$\partial\kappa$ beschreibt die Randflächen eines Elements κ und \mathbf{n} den nach außen gerichteten Normalenvektor auf den Randflächen $\partial\kappa$. Im Folgenden wird der Lösungsvektor \mathbf{q} durch die diskrete Lösung \mathbf{q}_h ersetzt. Als Funktionsraum soll ein Raum mit stückweise definierten Polynomen gewählt werden. Für Hexaeder-Elemente ist dieser nach [3]

$$\mathbb{Q}_p^m(\Omega) := \{\mathbf{v}_h \in [L^2(\Omega)]^m : \mathbf{v}_h|_{\kappa} \in [\mathbb{Q}_p(\kappa)]^m, \forall \kappa \in \mathcal{T}_h\}. \quad (3.5)$$

Hierbei ist m die Länge des Vektors \mathbf{v}_h . Der zugehörige Polynomraum für ein Element κ ist

$$\mathbb{Q}_p(\kappa) := \left\{ f : \kappa \rightarrow \mathbb{R} : f(\mathbf{x}) = \sum_{\alpha: \alpha_i \leq p} \mu_{\alpha} \mathbf{x}^{\alpha} \right\}, \quad (3.6)$$

$$\mu_{\alpha} \in \mathbb{R} \quad \text{und} \quad \alpha = (\alpha_1, \dots, \alpha_d).$$

Der Raum \mathbb{Q}_p umfasst die Funktionen auf einem Element κ mit einem Polynomgrad kleiner gleich p in jeder Raumrichtung und ist im Dreidimensionalen geometrisch als Hexaeder zu interpretieren.

Man kann hier einen Unterschied zu dem stetigen Galerkin-Verfahren erkennen. In diesem werden die diskreten Funktionen mit einer zusätzlichen Stetigkeitsbedingung in dem Gebiet Ω definiert. Der sich ergebene Unterschied in den Lösungen ist für 2D-Elemente in Abbildung 3.1 dargestellt.

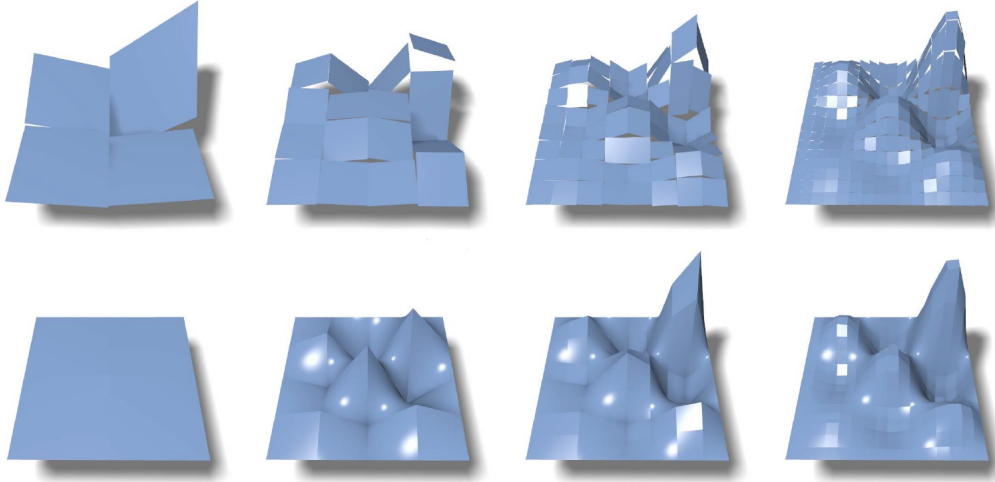


Abbildung 3.1: Lösung der Poisson-Gleichung mit einem DG-Verfahren (oben) und einem stetigen FE-Verfahren (unten) auf einem quadratischen Netz mit 2^2 , 4^2 , 8^2 und 16^2 - Elementen - lineare Ansätze [27]

Die Basisfunktionen des dreidimensionalen Raums $\mathbb{Q}_p(\kappa)$ seien $\{\phi_j\}_{j=1}^{N^3}$. Innerhalb eines Elements κ kann die diskrete Lösung $\mathbf{q}_h|_{\kappa} \in \mathbb{Q}_p(\kappa)$ nun als eine Linearkombination der Basisfunktionen ausgedrückt werden, vgl. [17]

$$\mathbf{q}_h(\mathbf{x}, t)|_{\kappa} = \mathbf{q}^{\kappa}(\mathbf{x}, t) = \sum_{j=1}^{N^3} \mathbf{q}_j^{\kappa}(t) \phi_j(\mathbf{x}). \quad (3.7)$$

Hierbei entspricht $\mathbf{q}_j^{\kappa}(t)$ den numerischen Freiheitsgraden eines Elements κ . Die Anzahl der Freiheitsgrade innerhalb eines Elements ist $N^3 := (p+1)^3$. Der diskrete Vektor der Testfunktionen \mathbf{v}_h wird gleich den Basisfunktionen $\{\phi_i\}_{i=1}^{N^3}$ gewählt. Die diskrete Formulierung ergibt sich durch

$$\int_{\kappa} \mathbf{v}_h \cdot \frac{\partial \mathbf{q}_h}{\partial t} d\kappa - \int_{\kappa} \nabla \mathbf{v}_h : \mathbf{F} d\kappa + \int_{\partial\kappa} \mathbf{v}_h \cdot \mathcal{H}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) ds = 0 \quad \forall \mathbf{v}_h \in Q_p^m(\Omega). \quad (3.8)$$

Da die Lösung \mathbf{q}_h unstetig zwischen zwei Nachbarelementen sein kann und somit keine wohldefinierten Werte auf den Grenzflächen hat, ist der Fluss $\mathbf{F}(\mathbf{q}_h) \cdot \mathbf{n}$ an den Rändern nicht eindeutig. Um ein eindeutiges Verfahren zu erhalten, wird der Fluss durch eine Flussfunktion approximiert. Diese Approximation wird üblicherweise als numerischer Fluss $\mathcal{H}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n})$ bezeichnet. Hierbei entsprechen \mathbf{q}_h^{\pm} den Randwerten von \mathbf{q}_h , bezogen auf das Innere des Elements κ bzw. κ' (siehe Abbildung 3.2), vgl. [22]. Beispiele für den numerischen Fluss werden in Kapitel 3.1.1 gegeben.

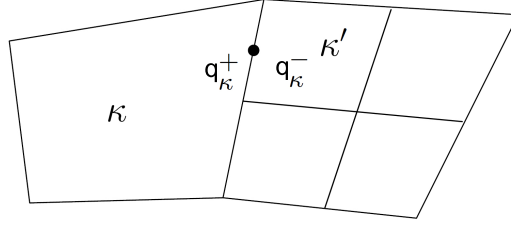


Abbildung 3.2: Definition der Randwerte eines Elements [22]

Setzt man den Ansatz 3.7 in die diskrete Formulierung ein und testet gegen die Basisfunktionen ϕ_i , wird daraus für alle $\kappa \in \mathcal{T}_h$

$$\int_\kappa \phi_i \frac{\partial \mathbf{q}_j^\kappa}{\partial t} d\kappa - \int_\kappa \nabla \phi_j \cdot \mathbf{F} d\kappa + \int_{\partial\kappa} \phi_j \mathcal{H}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) ds = \mathbf{0} \quad \text{für } i, j = 1, \dots, N^3. \quad (3.9)$$

Da der Lösungsvektor $\mathbf{q}_j^\kappa(t)$ ausschließlich von der Zeit abhängt und nur die Basisfunktionen $\phi(\mathbf{x})$ ortsabhängig sind, kann man die zeitliche Ableitung aus dem Integral ziehen.

$$\underbrace{\frac{\partial \mathbf{q}_j^\kappa}{\partial t} \int_\kappa \phi_i \phi_j d\kappa}_{\text{I}} - \underbrace{\int_\kappa \nabla \phi_i \cdot \mathbf{F} d\kappa}_{\text{II}} + \underbrace{\int_{\partial\kappa} \phi_i \mathcal{H}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) ds}_{\text{III}} = \mathbf{0}. \quad (3.10)$$

Das erste Volumenintegral (I) wird üblicherweise als Massenmatrix m_{ij} geschrieben und demnach definiert als

$$m_{ij}^\kappa = \int_\kappa \phi_i \phi_j d\kappa \quad \text{für } i, j = 1, \dots, N^3, \quad (3.11)$$

mit der inversen Massenmatrix ($\tilde{m}_{ji}^\kappa m_{ij}^\kappa = \delta_{ij}$). Die Massenmatrix ist im allgemeinen Fall eine symmetrische Matrix. Der Term (II) enthält die örtlichen Ableitungen der Basisfunktionen und wird daher analog zur FEM der Strukturmechanik als Steifigkeitsterm \mathbf{s}_i^κ bezeichnet. Zusammen mit dem numerischen Fluss \mathbf{h}_i^κ lässt sich die diskretisierte Gleichung in der kompakten Form schreiben, vgl. [8]:

$$\frac{\partial \mathbf{q}_j^\kappa(t)}{\partial t} + \mathbf{r}_j^\kappa = \mathbf{0} \quad \text{für } j = 1, \dots, N^3, \quad \forall \kappa \in \mathcal{T}_h, \quad (3.12)$$

wobei $\mathbf{r}_j^\kappa = \tilde{m}_{ji}^\kappa (\mathbf{h}_i^\kappa - \mathbf{s}_i^\kappa)$ ist.

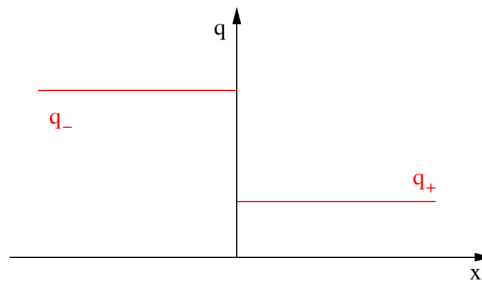


Abbildung 3.3: Das Riemann-Problem mit zwei Initialzuständen q_- und q_+ [24]

3.1.1 Numerischer Fluss

Der numerische Fluss \mathcal{H} ist bedingt durch die Approximation des physikalischen Flusses zwischen zwei Elementen. Im Folgenden soll ein kurzer Überblick über die Implementierung des numerischen Flusses gegeben werden. Da die Herleitung des numerischen Flusses nicht Hauptbestandteil der Arbeit ist, wird für weitere Informationen auf [41] verwiesen.

Das Flächenintegral (III) aus den diskretisierten Euler-Gleichungen 3.10 wird über die Ränder eines Elements gebildet. Die Ränder können entweder an anderen Elementen oder an Rändern des Berechnungsgebiets liegen. Der numerische Fluss an diesen Rändern soll mit $\mathcal{H}^\Gamma(\mathbf{q}_h^+, \mathbf{q}_h^\Gamma, \mathbf{n})$ bezeichnet werden und statt dem Zustand im Nachbarerelement \mathbf{q}_h^- , den Zustand in einer Geisterzelle \mathbf{q}_h^Γ betrachten.

Physikalisch betrachtet, gibt es nur eine korrekte Lösung für den Fluss an einem Elementübergang. Um diesen numerisch zu finden, muss man das Riemann-Problem lösen, welches durch die eindimensionale nicht-viskose Erhaltungsgleichung

$$q_t + f(q)_x = 0, \quad (3.13)$$

mit den Anfangsbedingungen $q = q_-$ für die linke Seite des Übergangs ($x < 0$) und $q = q_+$ für die rechte Seite ($x > 0$) gegeben ist (siehe Abb. 3.3), vgl. [32].

Die exakte Lösung des Problems an jedem Übergang zu finden, ist mit einem hohen numerischen Aufwand verbunden. Somit greift man üblicherweise auf eine Approximation zurück.

Die simpelste Variante für eine Approximation des Flusses ist vermutlich die Bildung des Durchschnitts der beiden Seiten:

$$\mathcal{H}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) = \frac{1}{2} \left(\mathbf{F}(\mathbf{q}_h^+) \cdot \mathbf{n} + \mathbf{F}(\mathbf{q}_h^-) \cdot \mathbf{n} \right). \quad (3.14)$$

Obwohl die Implementierung sehr einfach ist, wird diese Approximation selten verwendet, da sie im Allgemeinen zu einem instabilen Verfahren führt.

Grundsätzlich sollte die numerische Flussfunktion konsistent und konservativ sein. Es gibt einige Flussfunktionen, die diese Bedingungen erfüllen, wie zum Beispiel der Godunov Fluss, Lax-Friedrichs Fluss oder der Roe Fluss, vgl. [41]. Letzterer wird in der vorliegenden Implementierung verwendet.

Der Roe Fluss basiert auf dem linearisierten Riemann Problem

$$q_t + \mathbf{A} q_x = 0, \quad (3.15)$$

wobei \mathbf{A} die Jacobi-Matrix des Flusses $\frac{\mathbf{F} \cdot \mathbf{n}}{q}$ ist [43]. Der numerische Fluss ergibt sich dann wie folgt:

$$\mathcal{H}^{Roe}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) = \frac{1}{2} \left(\mathbf{F}(\mathbf{q}_h^+) \cdot \mathbf{n} + \mathbf{F}(\mathbf{q}_h^-) \cdot \mathbf{n} - \psi \left(\frac{\mathbf{F} \cdot \mathbf{n}}{q} \Big|_{\bar{\mathbf{q}}_{Roe}} \right) \cdot (\mathbf{q}_h^- - \mathbf{q}_h^+) \right) \quad (3.16)$$

Der Faktor ψ ist dabei eine in [21] vorgeschlagene Korrektur, durch die das Auftreten nicht-physikalischer Schocks vermieden werden kann. $\bar{\mathbf{q}}_{Roe}$ ist der sogenannte Roe average state, vgl. [21].

3.1.2 Randbedingungen

In diesem Kapitel soll ein kurzer Überblick über die implementierten Randbedingungen gegeben werden.

Reibungsfreie Wand

An reibungsfreien Wänden läuft die Strömung tangential an der Wand entlang. Dementsprechend werden die Geschwindigkeitskomponenten normal zur Wand gleich Null gesetzt.

$$\mathbf{n} \cdot \mathbf{u} = 0 \quad \text{auf } \Gamma. \quad (3.17)$$

Um dies zu gewährleisten, berechnen sich die Geschwindigkeitskomponenten der Geisterzellen nach folgendem Schema:

$$(\rho \mathbf{u})^\Gamma = (\rho \mathbf{u})^+ - 2 \left((\rho \mathbf{u})^+ \cdot \mathbf{n} \right) \mathbf{n}. \quad (3.18)$$

Der numerische Randfluss ergibt sich damit durch

$$\mathcal{H}^\Gamma = \begin{pmatrix} 0 \\ p_h \mathbf{n} \\ 0 \end{pmatrix}, \quad (3.19)$$

wobei p_h der arithmetische Durchschnitt des inneren und des Geisterzellen-Drucks ist, vgl. [8].

Riemann Randbedingungen

Die Idee der Riemann Randbedingungen ist die Vorgabe von Randwerten für subsonische Einström- und Ausströmränder durch die Extrapolation der Riemann Invarianten nach [30]

$$R_{\pm} = |\mathbf{u}| \pm \frac{2c}{\gamma - 1}, \quad (3.20)$$

von den inneren Randlelementen. An Einströmrändern ist \mathbf{q}_h^{Γ} so definiert, dass die Riemann Invarianten $R_-(\mathbf{q}_h^{\Gamma})$ und $R_-(\mathbf{q}_h^+)$ identisch sind und dass die Randbedingungen für den Totaldruck, die Totaltemperatur und den Eintrittswinkel erfüllt werden. Beim Austritt soll die Bedingung des Gegendrucks erfüllt werden und die Riemann Invarianten R_+ , die Entropie und der Austrittswinkel soll gleich den inneren Randwerten sein. Mit den Randwerten \mathbf{q}_h^{Γ} wird der Fluss durch den Roe Fluss approximiert, siehe dazu [8].

Periodische Randbedingungen

Bei einer periodischen Randbedingung werden zwei Randflächen aneinander gefügt, so als würden diese im Berechnungsgebiet direkt nebeneinander liegen. Die Randwerte \mathbf{q}_h^{Γ} sind damit identisch zu dem \mathbf{q}_h^- des verknüpften Elements. Es wird ebenfalls der Roe Fluss als numerischer Fluss verwendet.

3.1.3 Vergleich zur Finiten-Volumen-Methode

Allgemein wird das DG-Verfahren als eine Kombination von der FEM und der FVM bezeichnet. Die Analogie zur FEM wurde in den vorherigen Kapiteln bereits aufgezeigt. Der Zusammenhang zur FVM lässt sich ebenfalls leicht zeigen.

Nimmt man die Basisfunktionen in einem Element als konstant an

$$\phi \equiv 1, \quad (3.21)$$

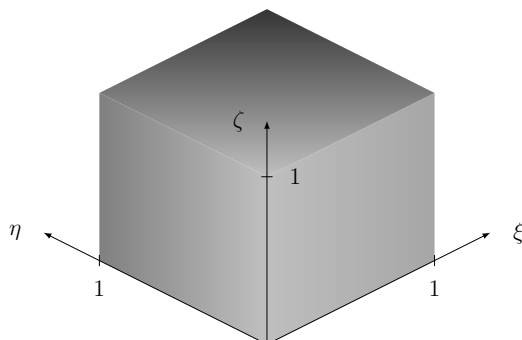
ergibt sich die Diskretisierung der Gleichung 3.10 wie folgt:

$$\frac{\partial \mathbf{q}_j}{\partial t} \int_{\kappa} 1 d\kappa - \int_{\kappa} \nabla 1 \cdot \mathbf{F} d\kappa + \int_{\partial\kappa} 1 \mathcal{H}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) ds = 0. \quad (3.22)$$

Aufgrund des Gradientens verschwindet der zweite Term und das Integral der Massenmatrix beschreibt im konstanten Fall das Volumen eines Elements V_{κ} . Damit erhält man das klassische FVM - Diskretisierungsschema:

$$\frac{\partial \mathbf{q}_j}{\partial t} V_{\kappa} + \int_{\partial\kappa} \mathcal{H}(\mathbf{q}_h^+, \mathbf{q}_h^-, \mathbf{n}) ds = 0, \quad (3.23)$$

wobei j den einzigen Freiheitsgrad in der Zellmitte darstellt.

Abbildung 3.4: Referenzhexaeder $\hat{\kappa}$

3.2 Elemente und Basisfunktionen

Prinzipiell ist es möglich, die DG-Diskretisierung für beliebige Arten von Elementen durchzuführen. Im Dreidimensionalen haben sich als Standartelemente Tetraeder, Hexaeder, Pyramiden und Prismen herausgebildet, wobei Tetraeder und Hexaeder am häufigsten Anwendung finden. In dieser Arbeit sollen die Hexaeder-Elemente im Vordergrund stehen.

Zur Vereinfachung wird das physikalische Element κ auf ein Referenzelement $\hat{\kappa}$ transformiert und alle Operationen, an denen die Elementgeometrie beteiligt ist, für das Referenzelement durchgeführt. Neben dem Aufstellen der Basisfunktionen und der numerischen Integration soll die Transformation in diesem Kapitel thematisiert werden.

3.2.1 Transformation auf das Referenzelement

Als Referenzelement wird für die Implementierung der Hexaeder-Elemente der Einheitswürfel $[0, 1]^3$ verwendet (siehe Abbildung 3.4). Die natürlichen Koordinaten der Referenzkonfiguration sollen mit $\boldsymbol{\xi} = (\xi, \eta, \zeta)^T$ bezeichnet werden.

Die physikalischen Koordinaten \boldsymbol{x} können mithilfe der Polynomdarstellung aus 3.7 innerhalb eines Elements wie folgt beschrieben werden

$$\boldsymbol{x}(\boldsymbol{\xi}) \approx \sum_{j=1}^{N^3} \boldsymbol{x}^j \psi_j(\boldsymbol{\xi}) \quad \forall \boldsymbol{x} \in \kappa. \quad (3.24)$$

Hierbei stellen \boldsymbol{x}^j die physikalischen Koordinaten der Stützpunkte (Eckpunkte) und $\psi_j(\boldsymbol{\xi})$ die Lagrange-Polynome dar. Die Anzahl der Stützpunkte richtet sich nach der Ordnung der Flächendarstellung und ist beispielsweise für ein gerades Hexaeder-Element $N^3 = 8$. Die inverse Abbildung beschreibt die natürlichen Koordinaten in Abhängigkeit der physikalischen Koordinaten $\boldsymbol{\xi} = \boldsymbol{\xi}(\boldsymbol{x})$.

Den Zusammenhang zwischen den partiellen Ableitungen kann man durch die Jacobi-Matrix $\mathbf{J}(\boldsymbol{\xi})$ ausdrücken, vgl. [28]

$$\begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}, \quad \frac{\partial}{\partial \boldsymbol{\xi}} = \mathbf{J}(\boldsymbol{\xi}) \frac{\partial}{\partial \mathbf{x}}. \quad (3.25)$$

Unter Verwendung der Polynomdarstellung 3.24 erfolgt die Berechnung der Jacobi-Matrix nach

$$\mathbf{J}(\boldsymbol{\xi}) = \sum_{j=1}^{N^3} \left(\nabla_{\boldsymbol{\xi}} \psi_j(\boldsymbol{\xi}) \otimes \mathbf{x}^j \right). \quad (3.26)$$

Die Umwandlung des Nablaoperators in natürliche Koordinaten wird über die inverse Operation

$$\nabla_{\mathbf{x}} = \mathbf{J}^{-1}(\boldsymbol{\xi}) \nabla_{\boldsymbol{\xi}} \quad \text{mit} \quad \mathbf{J}^{-1}(\boldsymbol{\xi}) = \frac{1}{|\mathbf{J}|} \mathbf{J}(\boldsymbol{\xi}) \quad (3.27)$$

realisiert. Die Determinante der Jacobi-Matrix stellt hier die Transformation des Volumenelements dV von physikalischen Koordinaten in natürliche Koordinaten dar

$$dV = |\mathbf{J}| d\xi d\eta d\zeta. \quad (3.28)$$

Durch diese Transformation ist es nun möglich, die Integrale aus der Gleichung 3.10 direkt auf dem Referenzelement $\hat{\kappa}$ auszuwerten. Daraus resultierend ist es ausschließlich notwendig, die Basisfunktionen auf dem Referenzelement in natürlichen Koordinaten $\hat{\phi}(\boldsymbol{\xi})$ zu formulieren, vgl. dazu [4].

3.2.2 Basisfunktionen

In der Literatur wird üblicherweise zwischen zwei Arten von Basisfunktionen unterschieden, den modalen und nodale Basisfunktionen. In dieser Arbeit sollen Letztere verwendet werden.

Grundsätzlich wird die skalare, diskrete Lösung q_h mithilfe von Polynomen (Basisfunktionen) interpoliert. Für N Interpolationspunkte ξ_i wird gefordert

$$q_h(\xi_i) = q_i \quad \text{und} \quad q_i = \sum_{j=1}^N a_j \hat{\phi}_j(\xi_i) \quad \text{für } i = 1, \dots, N \quad (3.29)$$

wobei q_h die approximierte Lösung, a_j unbekannte Polynomkoeffizienten und $\hat{\phi}_j$ die Basisfunktionen sind. Die kompakte Schreibweise sieht wie folgt aus:

$$v_{ij} a_j = q_i \quad \text{für } i = 1, \dots, N. \quad (3.30)$$

v_{ij} is die Vandermonde Matrix und enthält die Basisfunktionen, ausgewertet an den Interpolationspunkten

$$v_{ij} = \hat{\phi}_j(\xi_i) \quad \text{für } i, j = 1, \dots, N. \quad (3.31)$$

Mit der Invertierung der Vandermonde Matrix lassen sich die unbekannt Polynomkoeffizienten durch

$$a_j = \tilde{v}_{ji} q_i, \quad (3.32)$$

bestimmen. Die Invertierung wird trivial, wenn die Vandermonde Matrix der Einheitsmatrix entspricht. Dies ist genau dann der Fall, wenn die Interpolationspolynome an den Interpolationspunkten für $i = j$ gleich Eins und für $i \neq j$ gleich Null sind. Genau diese Eigenschaft besitzen die Lagrange-Polynome, die im Folgenden als Basisfunktionen $\hat{\phi}$ verwendet werden.

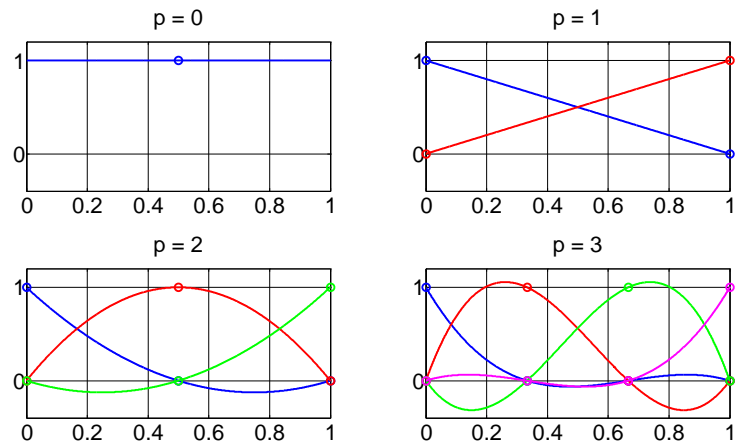
Daraus resultiert, dass die unbekannt Polynomkoeffizienten a_j gleich den Lösungen an den Interpolationspunkten q_i sind, vgl. [23]

$$a_j = \tilde{v}_{ji} q_i = \delta_{ji} q_i \quad \text{für } i, j = 1, \dots, N. \quad (3.33)$$

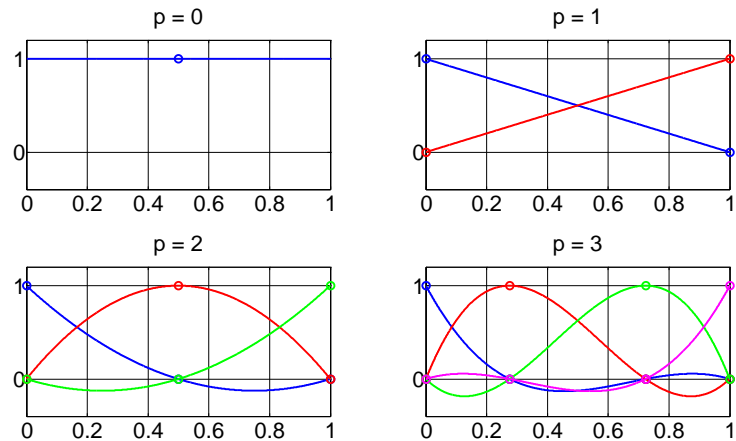
Offen bleibt lediglich die Wahl der Stützpunkte ξ_i . Grundsätzlich sind viele Arten denkbar. Häufig werden sogenannte äquidistante Knoten gewählt, wobei die Stützpunkte gleichmäßig über das Gebiet verteilt werden. Ein Vorteil ist sicherlich die einfache Implementierung der Basisfunktionen. Eine Alternative zu den gleich verteilten Stützpunkten sind die Legendre-Gauß-Lobatto (LGL)-Punkte. Diese sind definiert als die Nullstellen der Legendre-Polynome und den beiden Randpunkten, vgl. [26]. Beide Versionen von Basisfunktionen sind für die Polynomordnungen $p = 0, \dots, 3$ in Abbildung 3.5 dargestellt.

Bis zur Ordnung $p < 3$ sind die Interpolationspunkte und damit auch die Basisfunktionen identisch. Die kubischen Basisfunktionen ($p = 3$) weisen jedoch schon auf ein Problem der Verwendung von äquidistanten Knoten hin. Zur Verdeutlichung des Problems sind Basisfunktionen der Approximationsordnung $p = 8$ mit gleich verteilten und LGL-Interpolationspunkten in Abbildung 3.6 zu sehen.

Die äquidistanten Basisfunktionen oszillieren bei höherer Ordnung sehr stark (siehe Abb. 3.6a). Daraus resultieren insbesondere bei den Ableitungen sehr hohe Werte nahe den Endpunkten. Kleine Ungenauigkeiten verursachen daher einen großen Fehler in der Approximation. Im Gegensatz dazu werden die LGL-Basisfunktionen an $\xi = \xi_i$



(a) äquidistante Interpolationspunkte

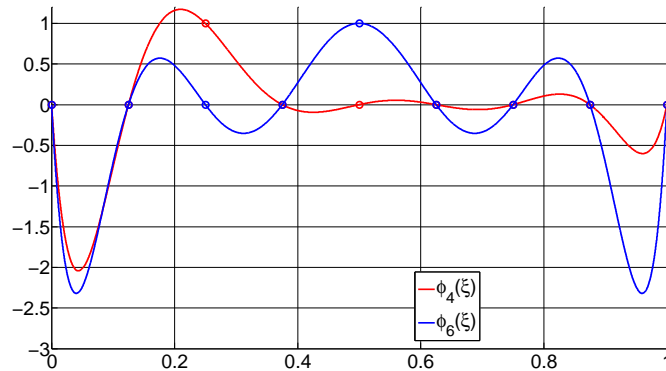


(b) LGL-Interpolationspunkte

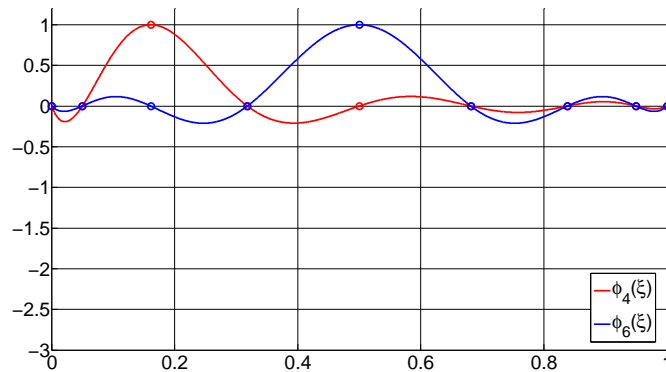
Abbildung 3.5: 1D-Lagrange-Basisfunktionen mit den Polynomordnungen p

maximal und nehmen für $|\xi - \xi_i| > 0$ rapide ab. Ungenauigkeiten haben daher einen geringeren Einfluss auf den approximierten Wert.

In [12] und [16] wurde gezeigt, dass die Interpolation mit äquidistanten Knoten bei sehr hohen Ordnungen zu einer instabilen Formulierung des Problems führt, da die Konditionszahl der Massenmatrix exponentiell mit N anwächst. Die Formulierung mit LGL-Stützpunkten hat eine bessere Konditionierung zur Folge und ist damit stabiler.



(a) äquidistante Interpolationspunkte



(b) LGL-Interpolationspunkte

Abbildung 3.6: Lagrange-Basisfunktionen mit der Polynomordnung $p = 8$

Die Übertragung der Basisfunktionen ins Dreidimensionale erfolgt über das Tensorprodukt der 1D-Basisfunktionen

$$\hat{\phi}(\boldsymbol{\xi}) = \hat{\phi}^{1D}(\xi) \otimes \hat{\phi}^{1D}(\eta) \otimes \hat{\phi}^{1D}(\zeta). \quad (3.34)$$

Dies bietet den Vorteil, dass die Basisfunktionen nur als eindimensionale Polynome implementiert werden müssen und in der DG-Implementierung auf drei Dimension erweitert werden können. Die Anzahl der Knoten wird im Dreidimensionalen kubiert (N^3). In Abbildung 3.7 sind die 64 Knoten für kubische Hexaeder-Elemente mit LGL- und äquidistanten Basisfunktionen dargestellt.

Die Elementmassenmatrix ergibt sich mit der Transformation durch die Jacobi-Matrix 3.28 und den 3D-Basisfunktionen des Referenzelements

$$m_{ij}^\kappa = \int_\kappa \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) d\kappa = \int_{\hat{\kappa}} |\mathbf{J}| \hat{\phi}_i(\boldsymbol{\xi}) \hat{\phi}_j(\boldsymbol{\xi}) d\hat{\kappa} \quad \text{für } i, j = 1, \dots, N^3. \quad (3.35)$$

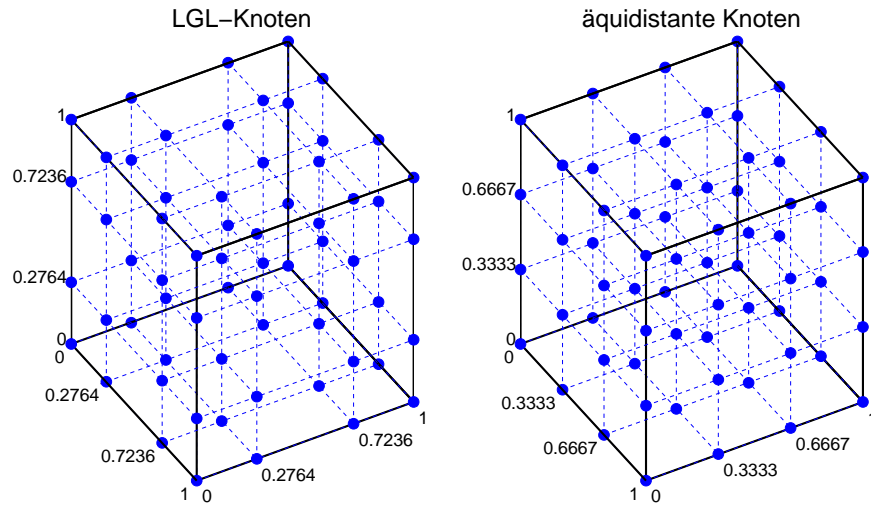


Abbildung 3.7: Knoten der kubischen Hexaeder-Elemente $\mathbb{Q}_3(\hat{\kappa})$ für LGL- und äquidistante Basisfunktionen

Die Berechnung der Integrale erfolgt mithilfe der numerischen Integration, welche im nächsten Abschnitt beschrieben werden soll.

3.2.3 Numerische Integration

Da die analytische Lösung der Integrale im allgemeinen Fall nicht möglich ist, wird eine numerische Approximation verwendet. Die Idee wird zunächst im Eindimensionalen an einem Linienintegral erläutert.

Das kontinuierliche Integral wird durch eine Summe der Form

$$I(f) = \int_a^b f(x) dx \approx Q(f) = \sum_{i=1}^M w_i f(x_i) \quad (3.36)$$

ersetzt. Die Approximation mit den Gewichten w_i und den Integrationspunkten x_i heißt Quadraturformel. Für eine Funktion f hat die Quadraturformel den Exaktheitsgrad m , wenn gilt [19]

$$I(f) = Q(f) \quad \forall f \in \mathbb{S}_m(\Omega), \quad (3.37)$$

wobei $\mathbb{S}_m(\Omega)$ den Polynomraum mit der maximalen Polynomordnung m darstellt. Der maximale Exaktheitsgrad ist für N betrachtete Integrationspunkte $m = 2M - 1$.

Dieser wird nur durch die Gauß-Quadratur erreicht. In der Gauß-Quadratur geht man von der folgenden Approximation

$$Q(f) = \sum_{i=1}^M w_i f(x_i) \approx I_w(f) = \int_a^b f(x)w(x) dx \quad (3.38)$$

aus. Hierbei ist $w(x)$ die Gewichtungsfunktion. Als Integrationspunkte werden die Nullstellen von Orthogonalpolynomen zu der Gewichtungsfunktion verwendet. Daraus resultiert der hohe Exaktheitsgrad der Quadratur. Für den Beweis siehe [19].

In der Praxis wird häufig die Gewichtungsfunktion $w(x) \equiv 1$ gewählt. Die dazugehörigen Orthogonalpolynome sind die (normierten) Legendre-Polynome. Die Legendre-Gauß-Quadratur (LG) besitzt somit den Exaktheitsgrad $2M - 1$, wobei M die Anzahl der verwendeten Integrationspunkte ist.

Ein interessanter Ansatz wird in der Legendre-Gauß-Lobatto Quadratur gewählt. Neben den Nullstellen der Legendre-Polynome dienen die beiden Endpunkte des Intervalls als Integrationspunkte der Quadratur. Durch das Festsetzen zweier Punkte reduziert sich der Exaktheitsgrad um zwei und ist daher $2M - 3$, wobei ebenfalls M Integrationspunkte gewählt wurden. Gegenüber dem Nachteil in der Exaktheit bietet sich mit der Verwendung von LGL-Basisfunktionen ein großer Vorteil in der Effizienz. Zur Erinnerung gelten für die nodalen LGL-Basisfunktionen

$$\phi_j(x_i) = \delta_{ij} \quad \text{für } j = 1, \dots, N \text{ und } i = 1, \dots, M, \quad (3.39)$$

wobei x_i genau die LGL-Integrationspunkte sind. Dementsprechend haben die Basisfunktionen an den Integrationspunkten den Wert Eins oder Null. Geht man von der Approximation $f_h(x) = \sum_{j=1}^N f_j \phi_j(x)$ aus, so gilt für die Quadratur des Integrals von f_h

$$I(f_h) = \int_a^b \sum_{j=1}^N f_j \phi_j(x) dx \approx \sum_{i=1}^M \sum_{j=1}^N w_i f_j \phi_j(x_i). \quad (3.40)$$

Im allgemeinen, eindimensionalen Fall müssen für die Auswertung über alle Integrationspunkte und Basisfunktionen $M \cdot N$ Schleifen durchlaufen werden. Unter Ausnutzung der LGL-Basisfunktionen und -Integration, vgl. Gleichung 3.39 und mit $M = N$, reduziert auf sich der Aufwand auf

$$I(f_h) \approx \sum_{i=1}^N w_i f_i, \quad (3.41)$$

mit nur N benötigten Schleifen. Dieses Verfahren wird als *Spectral-Element-Method* (SEM) und in Verwendung mit dem Discontinuous-Galerkin-Verfahren als *DGSEM*

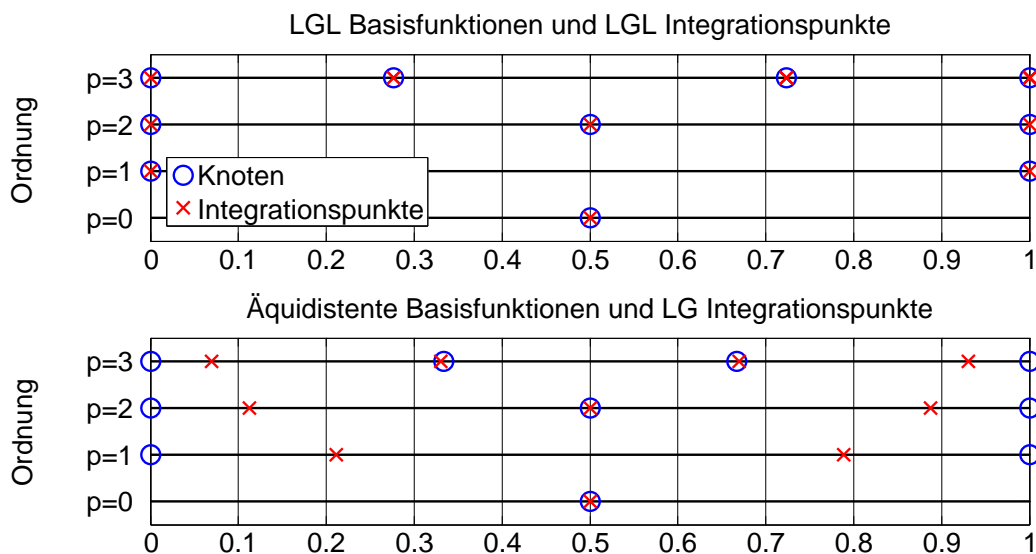


Abbildung 3.8: Vergleich der Integrationspunkte und der Knoten für DGSEM (oben) und DGNODAL (unten) im 1D Referenzsystem $[0, 1]$

	Interpolationspunkte	Integrationspunkte	Exaktheitsgrad
DGNODAL	äquidistant	LG	\mathbb{S}_{2m-1}
DGSEM	LGL	LGL	\mathbb{S}_{2m-3}

Tabelle 3.1: Überblick über DGSEM und DGNODAL

bezeichnet. Im Folgenden soll die Anzahl der Integrationspunkte gleich der Anzahl der Interpolationspunkte sein ($M = N$).

In der Tabelle 3.1 ist der Unterschied zwischen den beiden verwendeten Verfahren kurz dargestellt und in Abbildung 3.8 sind die sich ergebenden 1D-Integrationspunkte und Knoten bis zur 3. Polynomordnung abgebildet.

Da in Kapitel 3.2.1 das Referenzelement auf $[0, 1]$ festgelegt wurde, muss die vorgestellte Quadraturformel auf das Einheitsintervall transformiert werden. Diese Transformation kann mittels der affinen Abbildung

$$\xi = \frac{d-c}{b-a} \cdot x + \frac{ad-bc}{b-a}, \quad a \neq b, \quad (3.42)$$

durchgeführt werden. Für den speziellen Fall mit $[c, d] = [0, 1]$ ergibt sich nach [39]

$$\xi = \frac{1}{b-a} \cdot x - \frac{a}{b-a}. \quad (3.43)$$

Für die Quadratur der Volumenintegrale nutzt man, analog zu den Basisfunktionen, das Tensorprodukt der eindimensionalen Quadraturformeln. Approximiert werden soll das Integral im Referenzsystem

$$(I \otimes I \otimes I)(f) = \int_0^1 \int_0^1 \int_0^1 f(\xi, \eta, \zeta) d\xi d\eta d\zeta. \quad (3.44)$$

Geht man von den eindimensionalen Quadraturformeln

$$Q_\xi = \sum_{i=1}^{M_\xi} w_{\xi,i} g(\xi_i) \quad , \quad Q_\eta = \sum_{j=1}^{M_\eta} w_{\eta,j} g(\eta_j) \quad \text{und} \quad Q_\zeta = \sum_{k=1}^{M_\zeta} w_{\zeta,k} g(\zeta_k) \quad (3.45)$$

aus, folgt für die dreidimensionale Quadratur

$$(Q_\xi \otimes Q_\eta \otimes Q_\zeta)(f) = \sum_{i=1}^{M_\xi} \sum_{j=1}^{M_\eta} \sum_{k=1}^{M_\zeta} w_{\xi,i} w_{\eta,j} w_{\zeta,k} f(\xi_i, \eta_j, \zeta_k). \quad (3.46)$$

Für die Quadratur der 3D Basisfunktionen nach Gleichung 3.35 resultiert daraus

$$(Q_\xi \otimes Q_\eta \otimes Q_\zeta)(\hat{\phi}) = \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M w_{\xi,i} w_{\eta,j} w_{\zeta,k} \hat{\phi}^{1D}(\xi_i) \hat{\phi}^{1D}(\eta_j) \hat{\phi}^{1D}(\zeta_k), \quad (3.47)$$

mit $M = M_\xi = M_\eta = M_\zeta$. Es ist leicht zu erkennen, dass sich der Aufwand kubiert [20].

Anmerkung zu DGSEM

In der Implementierung ist die Anzahl der Integrationspunkte gleich der Anzahl an Interpolationspunkten auf $N = M = p + 1$ mit der Approximationsordnung p festgesetzt. Für die dreidimensionale Quadratur ergibt das $(p + 1)^3$ Schleifen über die Integrationspunkte. Bei der numerischen Integration einer Approximation nach Gleichung 3.40 resultiert in 3D

$$Q_\xi(f_h) = (Q_\xi \otimes Q_\eta \otimes Q_\zeta)(f_h) = \sum_{i=1}^{(p+1)^3} \sum_{j=1}^{(p+1)^3} w_i f_j \hat{\phi}_j(\boldsymbol{\xi}_i), \quad (3.48)$$

wobei $w_i = w_{\xi,k} w_{\eta,l} w_{\zeta,m}$ ist. Für die Auswertung der Quadratur und der Basisfunktionen werden nun insgesamt $(p + 1)^6$ Schleifen benötigt. Unter Anwendung des DGSEM-Verfahrens reduziert sich der Aufwand analog zum Eindimensionalen auf $(p + 1)^3$. Bei $p = 3$ ergibt dies eine Reduktion von 4096 benötigten Schleifen auf 64.

Für die Integration der Terme I und III aus Gleichung 3.10 ist die Effizienz des DGSEM-Verfahrens offensichtlich, aber auch für die Integration des Terms II aus

Gleichung 3.10, die den Gradienten der Basisfunktionen enthält, bietet es Geschwindigkeitsvorteile. Dies soll nun am Steifigkeitsterm (II) gezeigt werden.

$$\int_{\kappa} \sum_{j=1}^{(p+1)^3} \nabla \phi_j \cdot \mathbf{F} d\kappa = \int_0^1 \int_0^1 \int_0^1 \sum_{j=1}^{(p+1)^3} |\mathbf{J}| \mathbf{J}^{-1} \nabla_{\xi} \hat{\phi}_j \cdot \mathbf{F} d\xi d\eta d\zeta. \quad (3.49)$$

Die Quadratur erfolgt nach

$$Q_{\xi} = \sum_{i=1}^{(p+1)^3} \sum_{j=1}^{(p+1)^3} |\mathbf{J}| \mathbf{J}^{-1} \nabla_{\xi} \hat{\phi}_j(\xi_i, \eta_i, \zeta_i) \cdot \mathbf{F}, \quad (3.50)$$

wobei man über alle Integrationspunkte und Basisfunktionen insgesamt $(p+1)^6$ Schleifen braucht. Im allgemeinen Fall lässt sich bei der Verwendung von DGSEM keine Aussage über den Gradienten treffen, sodass diese Quadratur komplett gelöst werden müsste. Mithilfe des Tensorproduktes kann man den Gradienten jedoch aufschlüsseln:

$$\nabla_{\xi} \hat{\phi}(\boldsymbol{\xi}) = \begin{pmatrix} \frac{\partial \hat{\phi}(\boldsymbol{\xi})}{\partial \xi} \\ \frac{\partial \hat{\phi}(\boldsymbol{\xi})}{\partial \eta} \\ \frac{\partial \hat{\phi}(\boldsymbol{\xi})}{\partial \zeta} \end{pmatrix} = \begin{pmatrix} \frac{\partial \hat{\phi}^{1D}(\xi)}{\partial \xi} \hat{\phi}^{1D}(\eta) \hat{\phi}^{1D}(\zeta) \\ \hat{\phi}^{1D}(\xi) \frac{\partial \hat{\phi}^{1D}(\eta)}{\partial \eta} \hat{\phi}^{1D}(\zeta) \\ \hat{\phi}^{1D}(\xi) \hat{\phi}^{1D}(\eta) \frac{\partial \hat{\phi}^{1D}(\zeta)}{\partial \zeta} \end{pmatrix}. \quad (3.51)$$

Es lassen sich nun die Eigenschaften der 1D LGL-Basisfunktionen ausnutzen. Für die ξ -Komponente des Gradienten brauchen lediglich $p+1$ Basisfunktionen ausgewertet werden, da $\hat{\phi}_k^{1D}(\eta_i)$ oder $\hat{\phi}_l^{1D}(\zeta_i)$ nur an einem Integrationspunkt (ξ_i, η_i, ζ_i) nicht Null sind ($i=k$ & $i=l$). Dies folgt analog für die η und ζ Komponenten des Gradienten, womit sich der Aufwand auf $(p+1)^3 \cdot ((p+1) + (p+1) + (p+1)) = 3(p+1)^4$ reduziert.

3.2.4 Gekrümmte Ränder

Für Verfahren höherer Ordnung ist eine exakte Behandlung komplexer Oberflächengeometrie sehr wichtig, um genaue Ergebnisse zu erhalten (siehe dazu [4]). Üblicherweise erfolgt die Vernetzung des Berechnungsgebiets mit geraden Elementen, für die nur die Koordinaten der Eckpunkte gespeichert werden. In [13] wurde gezeigt, dass wenn man die Ordnung des Verfahrens erhöht ($p \geq 1$), stimmen die Interpolations- und Integrationspunkte nicht mit der gekrümmten Oberflächenkontur überein. Berechnet man mit den klassischen Elementen eine reibungsfreie Strömung um einem Zylinder, erhält man für höhere Ordnungen Oszillationen der Lösung an den geraden Randflächen, vgl. dazu Kapitel 5.2

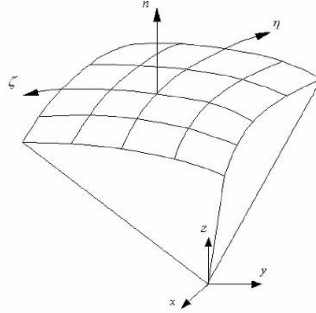


Abbildung 3.9: Normalenvektor der gekrümmten Oberfläche [35]

Dieser Effekt kann durch eine Verwendung von gekrümmten Randelementen verhindert werden. Die Transformation von geraden auf gekrümmte Elemente erfolgt über die Polynomdarstellung 3.24 unter Einbeziehung zusätzlicher Stützpunkte auf der gekrümmten Oberfläche. Auf die Berechnungsweise der Jacobi-Matrix hat dies keinen Einfluss und kann weiterhin, wie in Kapitel 3.2.1 vorgestellt, durchgeführt werden. Durch die höhere Ordnung der Parametrisierung erhöht sich jedoch der Aufwand [8]. Die Berechnung des Normalenvektors \mathbf{n} erfolgt über das Kreuzprodukt der Tangenten der gekrümmten Fläche. Die Tangenten aus Abbildung 3.9 ergeben sich wie folgt [35]:

$$\mathbf{t}^\eta = \begin{pmatrix} \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \eta} \\ \frac{\partial z}{\partial \eta} \end{pmatrix} = \mathbf{J}(\boldsymbol{\xi}) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{und} \quad \mathbf{t}^\zeta = \begin{pmatrix} \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \zeta} \end{pmatrix} = \mathbf{J}(\boldsymbol{\xi}) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (3.52)$$

Damit ist der Normalenvektor gegeben durch

$$\mathbf{n} = \mathbf{t}^\eta \times \mathbf{t}^\zeta. \quad (3.53)$$

Der Normalenvektor im Flächenintegral und die Jacobi-Matrix sind bei gekrümmten Rändern in jedem Fall von den natürlichen Koordinaten abhängig und müssen an jedem Integrationspunkt neu ausgewertet werden. Dadurch wird die Simulation von gekrümmten Elementen deutlich rechenintensiver.

3.3 Zeitdiskretisierung

Die Ortsdiskretisierung der Euler-Gleichungen durch die DG-Methode führt auf ein System gewöhnlicher Differentialgleichungen in der Zeit

$$\frac{\partial \mathbf{q}_j^\kappa(t)}{\partial t} + \mathbf{r}_j^\kappa = \mathbf{0}. \quad (3.54)$$

Hierbei entspricht $\mathbf{q}_j^\kappa(t)$ dem Lösungsvektor in jedem Freiheitsgrad $j = 1, \dots, N^3$ eines Elements $\kappa \in \mathcal{T}_h$. Für den Residuenvektor \mathbf{r}_j^κ verhält es sich analog. Die Gleichung 3.54 soll nun auch in der Zeit diskretisiert werden. Hierfür sind zahlreiche Verfahren bekannt, die man zunächst in implizite und explizite Verfahren unterscheiden kann. Bei expliziten Methoden wird die gesuchte Lösung an dem Zeitpunkt t_{n+1} durch einen bekannten Zustand berechnet. Der Vorteil eines solchen Verfahrens liegt in der einfachen Implementierung und in dem geringen numerischen Aufwand. Allerdings unterliegt ein explizites Verfahren gewisser Stabilitätsbedingungen. Die maximale Zeitschrittgröße muss die Courant-Friedrichs-Lewy (CFL) Bedingung erfüllen, damit das Verfahren stabil bleibt. Implizite Verfahren sind dagegen unbedingst stabil und lassen im Allgemeinen größere Zeitschritte als explizite Verfahren zu. Der Zustand an t_{n+1} wird hierbei mit dem noch unbekanntem Residuenvektor des Zustands an dem Zeitpunkt t_{n+1} berechnet, vgl. [31]. Dies erfordert eine iterative Lösung des Systems und ist dementsprechend deutlich aufwendiger. In der vorliegenden Implementierung wird ein explizites Verfahren verwendet.

3.3.1 Explizite Verfahren

Ausgehend von einer Taylor-Entwicklung der Gleichung 3.54 bis zur 2. Ordnung

$$\mathbf{q}_j^\kappa(t+h) = \mathbf{q}_j^\kappa(t) + h \frac{\partial \mathbf{q}_j^\kappa(t)}{\partial t} + \mathcal{O}(h^2) \quad (3.55)$$

lässt sich leicht ein explizites Verfahren erster Ordnung herleiten. Setzt man $\mathbf{q}_j^\kappa(t) = \mathbf{q}^n$, $\mathbf{q}_j^\kappa(t+h) = \mathbf{q}^{n+1}$, $h = \Delta t$ und nach Gleichung 3.54

$$\frac{\partial \mathbf{q}^n(t)}{\partial t} = \mathbf{r}(\mathbf{q}^n, t^n),$$

wird aus der Taylor-Entwicklung das explizite Euler-Verfahren 1. Ordnung

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \mathbf{r}(\mathbf{q}^n, t^n). \quad (3.56)$$

Um Verfahren höherer Konvergenzordnung zu erhalten, kann die Taylor-Entwicklung bis zu einem beliebigen Abbruchglied fortgeführt werden. Die entstehenden Einschritt-Verfahren mit mehreren Stufen pro Zeitschritt (stages), nennt man Runge-Kutta-

Verfahren (RK). In der DG-Implementierung wird ein explizites RK-Verfahren dritter Ordnung verwendet. Die Zwischenstufen ergeben sich nach [8]

$$\begin{aligned} \mathbf{q}^1 &= \mathbf{q}^n + \Delta t \mathbf{r}(\mathbf{q}^n, t^n), \\ \mathbf{q}^2 &= \frac{3}{4} \mathbf{q}^n + \frac{1}{4} \left(\mathbf{q}^1 + \mathbf{r}(\mathbf{q}^1, t^n + \Delta t) \right), \\ \mathbf{q}^{n+1} &= \frac{1}{3} \mathbf{q}^n + \frac{2}{3} \left(\mathbf{q}^2 + \mathbf{r}(\mathbf{q}^2, t^n + \frac{1}{2} \Delta t) \right). \end{aligned} \quad (3.57)$$

Bis zur Ordnung $s < 5$ haben s -stufige ERK auch die Konvergenzordnung s .

3.3.2 CFL-Bedingung

Wie am Anfang des Abschnitts angedeutet, sind explizite Zeitintegrations-Verfahren bedingt stabil. Das bedeutet, dass sie nur unter bestimmten Bedingungen ein stabiles Verhalten zeigen. Sind diese nicht erfüllt, wird das Verfahren instabil. Um im Vorfeld die CFL-Zahl zu ermitteln, die unter der Bedingung

$$|c| \frac{\Delta t}{\Delta x} \leq \text{CFL}, \quad (3.58)$$

ein stabiles Verhalten gewährleistet, muss eine von-Neumann-Stabilitätsanalyse für den linearen Fall durchgeführt werden. Hierbei ist $|c|$ die Schallgeschwindigkeit, Δt die Zeitschrittgröße und Δx eine Netzgröße. Diese Bedingung ist ebenfalls für den nichtlinearen Fall gültig, da nur dann die Rundungsfehler nicht vergrößert werden, vgl. [11].

Für die DG-Diskretisierung der Ordnung p und der Verwendung eines $(p+1)$ -stufigen expliziten Runge-Kutta-Verfahren, zeigte sich in den numerischen Tests in [11] eine CFL-Zahl von

$$\text{CFL} = \frac{1}{2p+1}. \quad (3.59)$$

Für die Ordnung $p = 0$ und $p = 1$ lässt sich die Bedingung beweisen (siehe dazu [10]). Ab einer Ordnung von $p \geq 2$ trifft diese Vorhersage nicht mehr vollständig zu, da für höhere Ordnungen Abweichungen von etwa 5% auftreten [11].

Die Stabilitätsbedingung ist in der Praxis häufig der limitierende Faktor für die Zeitschrittgröße. In vielen Anwendungen sind das Netz (Δx) und die Geschwindigkeit ($|\mathbf{u}|$) fest vorgegeben, so dass nur die Zeitschrittgröße angepasst werden kann. Dies resultiert insbesondere für höhere Ordnungen in sehr kleinen Zeitschritten und langen Rechenzeiten.

3.4 Fehlerabschätzung und Konvergenzordnung

Das DG-Verfahren zur Ortsdiskretisierung ist konvergent, wenn der Fehler ε bei einer Verkleinerung der Netzgröße h oder bei Vergrößerung der Polynomordnung p kleiner wird. Innerhalb dieser Arbeit soll nur die h -Konvergenz betrachtet werden. Für diese gilt im L^2 -Raum

$$\|\varepsilon\|_{L^2(\Omega)} = C \cdot h^k. \quad (3.60)$$

Der Diskretisierungsfehler ε ist hier die Differenz zwischen der exakten Lösung q und der approximierten Lösung q_h :

$$\varepsilon = q - q_h. \quad (3.61)$$

C bezeichnet eine Konstante, h eine Netzgröße und k die Konvergenzordnung. Im Allgemeinen ist die Fehlerabschätzung sehr kompliziert und nur für spezielle Fälle möglich. Hier sollen zwei mögliche Abschätzungen für den skalaren linearen Fall nach [9] angegeben werden.

Theorem 3.62 (Erste L^2 Fehlerabschätzung) Sei q_0 die Anfangsbedingung, κ ein Element der Zerlegung \mathcal{T}_h , $q_0 \in H^{p+1}(\kappa)$, $q_h \in Q_p^1$ und ε der Diskretisierungsfehler, dann gilt

$$\|\varepsilon(t)\|_{L^2(\kappa)} \leq C |q_0|_{H^{p+1}(\kappa)} h^{p+1/2},$$

wobei C nicht von h abhängt.

Unter der Annahme, dass die Anfangsbedingung noch regelmäßiger ist, kann die Bedingung belegt werden.

Theorem 3.63 (Zweite L^2 Fehlerabschätzung) Sei q_0 die Anfangsbedingung, κ ein Element der Zerlegung \mathcal{T}_h , $q_0 \in H^{p+2}(\kappa)$, $q_h \in Q_p^1$ und ε der Diskretisierungsfehler, dann gilt

$$\|\varepsilon(t)\|_{L^2(\kappa)} \leq C |q_0|_{H^{p+2}(\kappa)} h^{p+1},$$

wobei C nicht von h abhängt.

Beweis.

Siehe [9] und für allgemeinere Formulierungen siehe [25] and [34].

Anmerkung: H^{p+2} ist hierbei der Sobolev-Raum W_2^{p+2} .

Die Theoreme besagen, dass die DG-Diskretisierung bei einer ausreichend glatten Anfangsbedingung in einem Verfahren $(p+1)$ ter Ordnung im L^2 -Raum resultiert.

Die Erhöhung der Konvergenzordnung durch eine Erhöhung der Ordnung des Approximationspolynoms ist ein großer Vorteil gegenüber der FVM-Diskretisierung. Um mit der FVM stabile Verfahren höherer Ordnungen zu realisieren, müssen mehrere Zellen in jeder Raumrichtung in die Approximation einbezogen werden (stencil). Neben dem Anstieg des numerischen Aufwands werden dadurch mehr Informationen über das Netz benötigt [14]. Im Gegensatz dazu werden bei der DG-Diskretisierung neben dem betrachteten Element ausschließlich die unmittelbaren Nachbarzellen mit ausgewertet (numerischer Fluss).

Obwohl die vorgestellte Fehlerabschätzung lediglich für den linearen Fall durchgeführt wurde, gibt sie doch einen starken Hinweis auf die Größenordnung im nichtlinearen Fall, falls die exakte Lösung glatt genug ist [9]. In numerischen Untersuchungen konnte die angegebene Konvergenzordnung des DG-Verfahrens für die mehrdimensionalen Euler-Gleichungen bereits nachgewiesen werden [8].

Für die numerische Auswertung der Konvergenzgeschwindigkeit k ist ein Modellproblem nötig, von der man die exakte Lösung q kennt. Durch den Vergleich zweier unterschiedlicher Verfeinerungsstufen h_i und h_{i+1} , erhält man

$$\frac{\|q - q_h^i\|_{L^2(\Omega)}}{\|q - q_h^{i+1}\|_{L^2(\Omega)}} = \frac{C^i h_i^k}{C^{i+1} h_{i+1}^k}. \quad (3.64)$$

Unter Anwendung des Logarithmus und der Annahme das C nicht von h abhängt wird daraus

$$\ln \frac{\|q - q_h^i\|_{L^2(\Omega)}}{\|q - q_h^{i+1}\|_{L^2(\Omega)}} = k \cdot \ln \frac{h_i}{h_{i+1}}. \quad (3.65)$$

Nach k aufgelöst, ergibt sich

$$k = \frac{\ln \left(\|q - q_h^i\|_{L^2(\Omega)} \right) - \ln \left(\|q - q_h^{i+1}\|_{L^2(\Omega)} \right)}{\ln (h_i) - \ln (h_{i+1})}. \quad (3.66)$$

4 Strömungslöser TRACE

In dieser Arbeit soll der dreidimensionale DG-Löser in TRACE auf Hexaeder-Elemente erweitert werden. Ausgangslage der Implementierung ist der DG-Code für Tetraeder-Elemente bis zur Ordnung $p = 3$.

Entwickelt wurde TRACE am Institut für Antriebstechnik des Deutschen Zentrums für Luft- und Raumfahrt in Köln und ist ursprünglich ein FVM-Solver für die Navier-Stokes-Gleichungen auf strukturierten und unstrukturierten Netzen. Der grundlegende Prozessablauf ist in der Abbildung 4.1 rudimentär dargestellt.

Die DG-Diskretisierung ist in den Flow-Solver TRACE integriert und benutzt trotz unterschiedlicher Strukturen viele gemeinsame Funktionen (Einleseroutinen, Kommunikation, Ausgaben etc.). Der allgemeine Prozessablauf ist dementsprechend ähnlich. TRACE ist wie auch POST und PREP größtenteils in der Programmiersprache C geschrieben. Zur Netzgenerierung wurde in der Arbeit hauptsächlich Pointwise als kommerzielles Programm verwendet. Die exportierte Output-Datei ist ein strukturiertes Netz im CGNS-Format. CGNS steht für CFD General Notation System und ist die in Trace unterstützte Elementdatenbank. Die CGNS-Notation legt dabei die Elementtypen und die Nummerierung der Flächen, Eckpunkte und Knoten eines Elementes fest und muss daher in den Implementierungen stets beachtet werden. Im Pre-Processing folgt die Einrichtung des Modells. In GMC (General Mesh Connector) können Randbedingungen, Initialisierung, Zeitschrittgröße und weitere Parameter vorgegeben werden. Außerdem kann das strukturierte Netz in ein unstrukturiertes Netz umgewandelt werden. Die Erzeugung von gekrümmten Elementen (höherer Ordnung) erfolgt beispielsweise in PREP, dem Pre-Processing Tool von TRACE, vgl. dazu [13]. Über das Control-File können spezielle, neu implementierte Module angewählt werden, wie zum Beispiel die DG-Ortsdiskretisierung oder das explizite Runge-Kutta-Verfahren. Für das DG-Verfahren wird die Approximationsordnung und die Art der Basisfunktionen ebenfalls im Control-File festgelegt. Die Aufbereitung der Strömungsdaten nach der Simulation erfolgt über das Post-Processing Tool POST. Mit POST wird zum Beispiel die Verfeinerung des Netzes zur Darstellung von Approximationen höherer Ordnungen, die Berechnung des L^2 -Fehlers und die Auswertung des Druckbeiwertes durchgeführt.

Der grundsätzliche Ablauf des Flow-Solvers mit Verwendung der DG-Ortsdiskretisierung und des 3-stufigen expliziten Runge-Kutta-Verfahrens ist in Abbildung 4.2 dargestellt. Hierbei handelt es sich nicht um den vollständigen Programmablauf. Es sollten lediglich die essentiellen Programmteile aufgelistet werden. Als Ausgangspunkt der

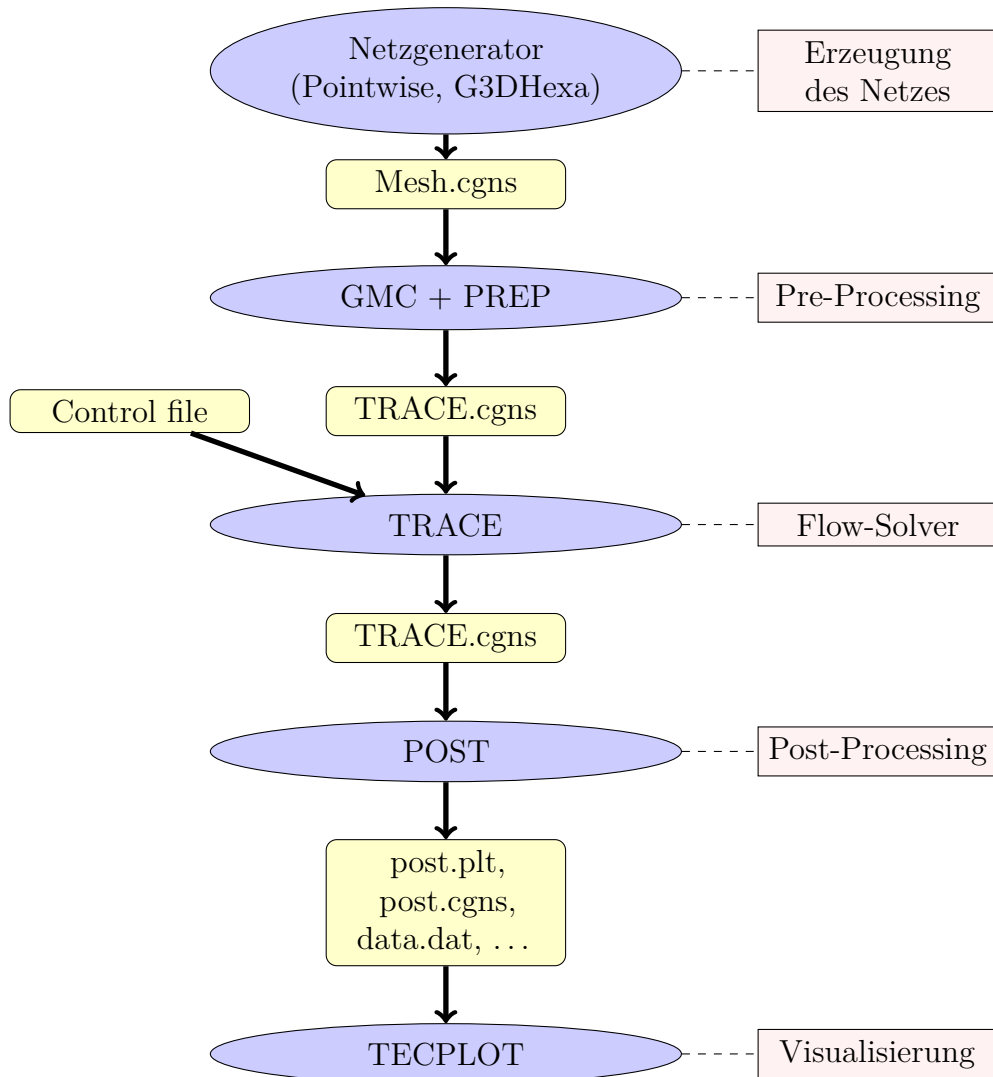


Abbildung 4.1: Allgemeiner Simulationsprozessablauf [1]

Simulation dient die CGNS- und die Control-Datei, die in der Initialisierung des Programms eingelesen werden. Auf der Grundlage werden allgemeine Solvareinstellung definiert und Parameter, wie beispielsweise die RK-Koeffizienten, bestimmt. Weiterhin wird in der Initialisierung der Speicher für die DG-Strukturen allokiert, Netzauswertungen vorgenommen (z.B. Verdrehung der Elemente zueinander) und die Randelemente höherer Ordnung parametrisiert. Im Anschluss folgt die Berechnung der Massenmatrix und der Anfangsbedingung für jedes Element und daraufhin die Anpassung der Randwerte in den Geisterzellen. Da TRACE mehrere Blöcke parallel auf verschiedenen Prozessoren rechnen kann, muss dementsprechend mit jeder neuen Lösung eine Kommunikation zwischen den Blöcken stattfinden. Nach Initialisierung

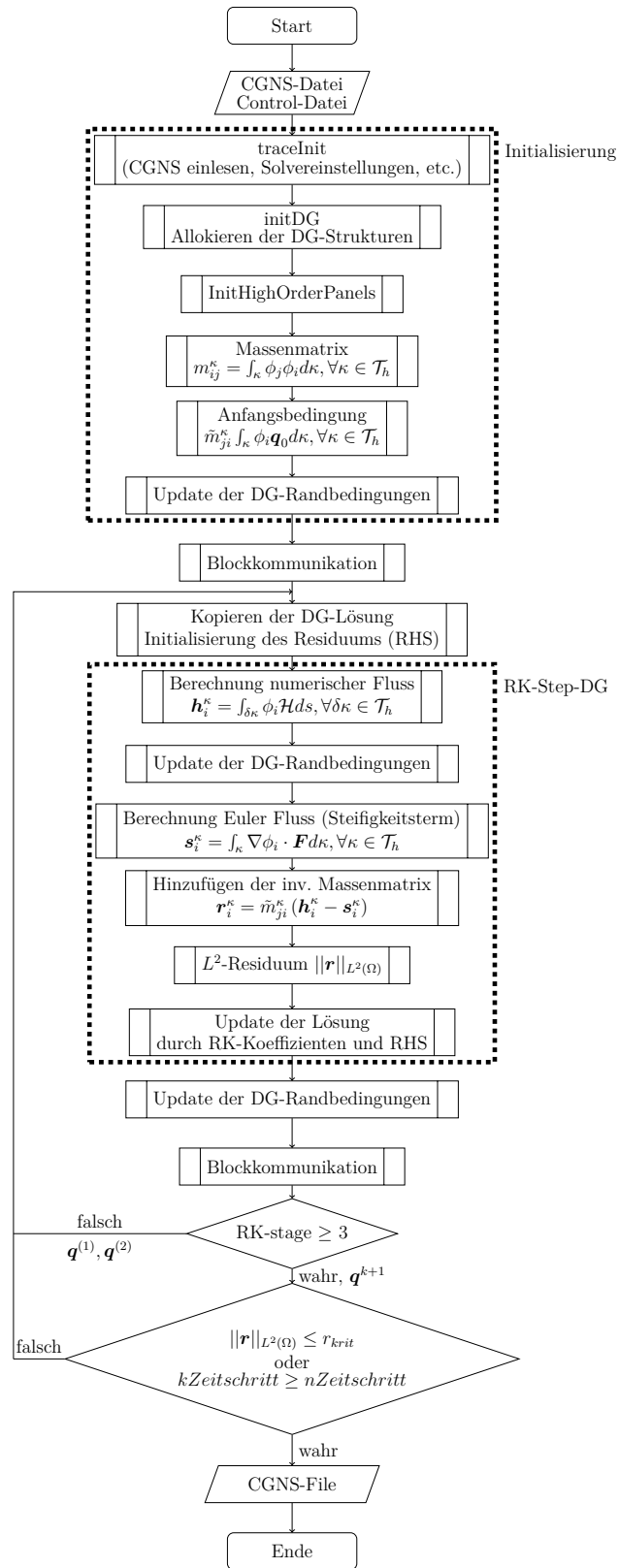


Abbildung 4.2: Ablaufplan Runge-Kutta-DG-Solver

des Residuums folgt die Berechnung der RHS, mit den numerischen Flüssen für alle Flächen, dem Steifigkeitsterm und der Multiplikation der inversen Massenmatrix. Daraufhin wird die L^2 -Norm von jeder konservativen Variablen im Residuenvektor ausgewertet und aufsummiert. In der Folge wird die Lösung analog des RK-Schemas upgedatet (abhängig von der aktuellen RK-stage). Nach dem Update der Randbedingung und einer Blockkommunikation folgt eine Überprüfung der Abbruchkriterien. Ein Abbruch kann durch Erreichen der gewünschten Größe des Residuums oder der maximalen Anzahl an Zeitschritten erfolgen. Hierbei sei angemerkt, dass ein Abbruch nur nach Beendigung der drei Runge-Kutta-stages geschehen kann. Der Output der Ergebnisse erfolgt in einer CGNS-Datei, die dann mit POST oder direkt in Tecplot ausgewertet werden kann.

5 Numerische Ergebnisse

In diesem Kapitel werden einige numerische Beispiele zur Verifikation des DG-Verfahrens für die instationären Euler-Gleichungen auf unstrukturierten Netzen vorgestellt. In den Simulationen werden die innerhalb dieser Arbeit implementierten Hexaeder-Elemente der DGNODAL-Formulierung verwendet. Für die Zeitdiskretisierung soll ein 3-stufiges explizites Runge-Kutta-Verfahren eingesetzt werden.

Einleitend erfolgt eine Konvergenzstudie des Verfahrens. In dieser wird das Konvergenzverhalten verschiedener Approximationsordnungen bezüglich der stetig feiner werdenden Netzgröße h untersucht. Die Studie soll sowohl für das DGNODAL- als auch für das DGSEM-Verfahren durchgeführt und die Ergebnisse miteinander verglichen werden.

Im Anschluss wird die Strömung um einen Zylinder simuliert. Bei diesem Beispiel steht die Verwendung von gekrümmten Rändern (Randelemente höherer Ordnung) im Fokus. Dazu soll neben dem Strömungsfeld auch der Druckbeiwert am Profil zum Vergleich der geraden und gekrümmten Ränder herangezogen werden.

Abschließend wird die Strömung um ein NACA-Profil untersucht. Mithilfe von gekrümmten Randelementen soll gezeigt werden, dass sich die Lösung mit der DG-Diskretisierung bei deutlich gröberer Vernetzung an eine Referenzlösung des FV-Solvers TRACE auf einem sehr feinen Netz annähert.

5.1 Konvergenzstudie

5.1.1 Aufbau des Modells

In dem Kapitel 3.4 wurde formal eine Konvergenzordnung von $\mathcal{O}(h^{p+1})$ im L^2 -Raum des DG-Verfahrens auf allgemeinen Netzen vorgestellt, wobei p die Approximationsordnung ist. Diese gilt es nun in praktischen Untersuchungen mit den implementierten Hexaeder-Ansätzen zu beweisen. Hierzu wird ein Testfall mit verschiedenen Approximationsordnungen $p = 0, \dots, 3$ auf systematisch feiner werdenden Netzen $h \rightarrow 0$ simuliert. Die Ergebnisse werden durch die Berechnung der L^2 -Norm des Fehlers zur exakten Lösung miteinander verglichen.

Da man für die Bildung des Fehlers die exakte Lösung des Problems kennen muss, eignen sich nur einfache Beispiele für eine Konvergenzstudie. In dieser Arbeit soll eine

dreidimensionale Schwankung der Dichte in der Form einer Gauß-Funktion berücksichtigt werden. Diese Schwankung wird mit der Strömung durch das Berechnungsgebiet geschickt und an einem Punkt mit der dort initialisierten Gauß-Funktionen verglichen. Die Anfangsbedingungen der Gaußverteilung werden wie folgt berechnet

$$\mathbf{q}_0(\rho_0, \mathbf{u}_0, p_0, \mathbf{x}_0, \mathbf{x}) = \begin{pmatrix} \rho_0 + a \exp\left(-\ln(2) \left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2} + \frac{(z-z_0)^2}{\sigma_z^2}\right)\right) \\ \rho_0 \mathbf{u}_0 \\ \frac{p_0}{\gamma - 1} + \frac{\rho_0 |\mathbf{u}_0|^2}{2} \end{pmatrix}, \quad (5.1)$$

wobei \mathbf{x}_0 die Koordinaten der Spitze, σ die Breite und a die Höhe des Pulses darstellen. Die Konstanten sind definiert nach

$$\begin{aligned} \rho_0 &= 1 \text{ kg/m}^3; \quad p_0 = 86118 Pa; \quad a = 0.01; \\ \mathbf{u}_0 &= \begin{pmatrix} 100 \\ 0 \\ 0 \end{pmatrix} \text{ m/s}; \quad \mathbf{x}_0 = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} \text{ m}; \quad \sigma = \begin{pmatrix} 0.1 \\ 0.1 \\ 0.1 \end{pmatrix} \text{ m}. \end{aligned}$$

Die Initialisierung des Strömungsfeldes ist in Abbildung 5.1 dargestellt.

Die exakte Lösung des Anfangswertproblems ist trivial und kann durch ein Verschieben des Gauß-Pulses bestimmt werden. Die Auswertungsposition soll bei $\mathbf{x} = (1.5, 0, 0)^T$ liegen und wird mit der vorgegebenen Geschwindigkeit \mathbf{u}_0 nach 0.01s erreicht.

Die Konvergenzstudie erfolgt auf einem kartesischen Netz mit gleichmäßigen, quaderförmigen Elementen und betrachtet insgesamt sechs Verfeinerungsstufen. Die Verfeinerungsstufen sind in der Übersicht 5.1 dargestellt. Als Netzgröße h wird die Kantenlänge der quaderförmigen Elemente verwendet. Die Ränder in y - und z -Richtung wurden als reibungsfreie Wände modelliert und für den Einlass und den Auslass in x -Richtung wurden periodische Randbedingungen gewählt.

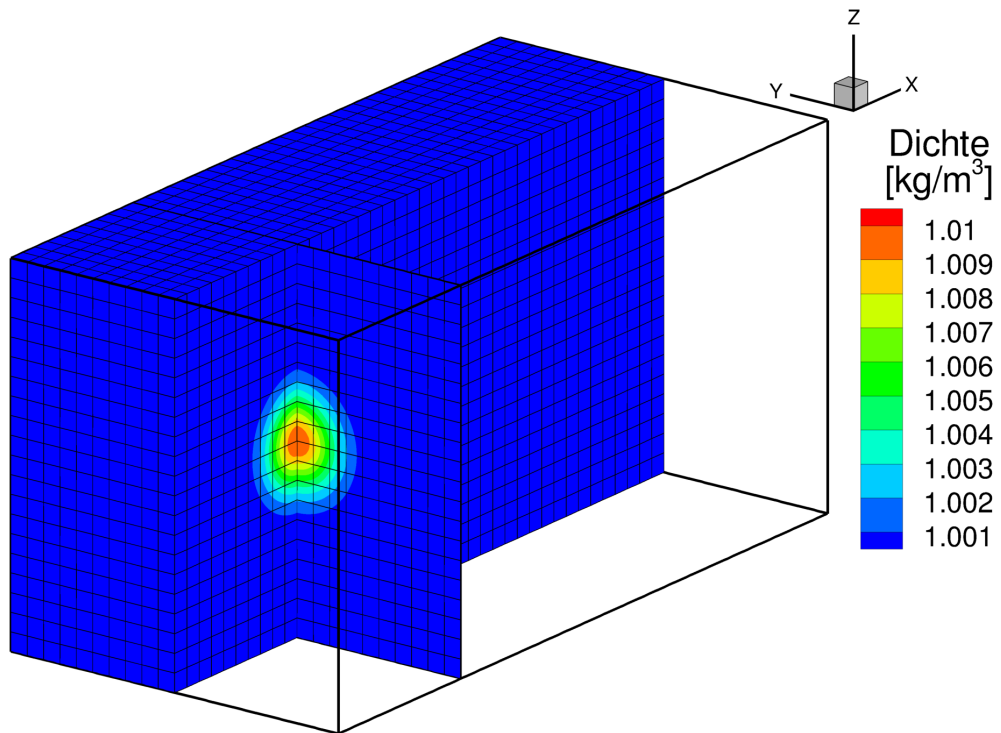
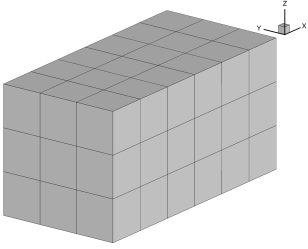
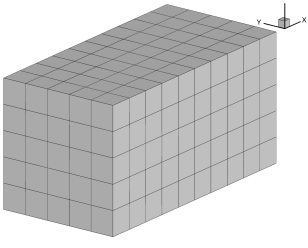


Abbildung 5.1: Initialisierung des Strömungsfeldes der Konvergenzstudie

Elementanzahl	h in [m]	Darstellung
54	$1/3$	
250	0.2	

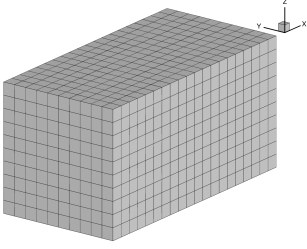
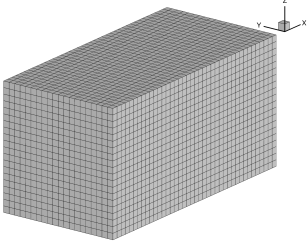
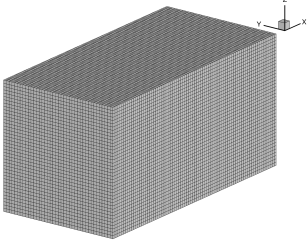
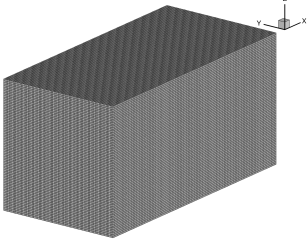
2000	0.1	
16000	0.05	
128000	0.025	
1024000	0.0125	

Tabelle 5.1: Verfeinerungsstufen der Konvergenzstudie

Es konnte jedoch nicht jedes Netz mit jeder Approximationsordnung gerechnet werden. Beispielsweise war die Größe des Gauß-Pulses für das Netz 54 bei linearer Approximation nicht ausreichend, sodass alle Knoten mit Eins initialisiert wurden (siehe Abb. 5.2). Bei $p = 0$ trat dieses Problem nicht auf, da die Knoten in der Elementmitte liegen. Außerdem war die lange Simulationszeit für hohe Ordnungen auf feinen Netzen ein limitierender Faktor.

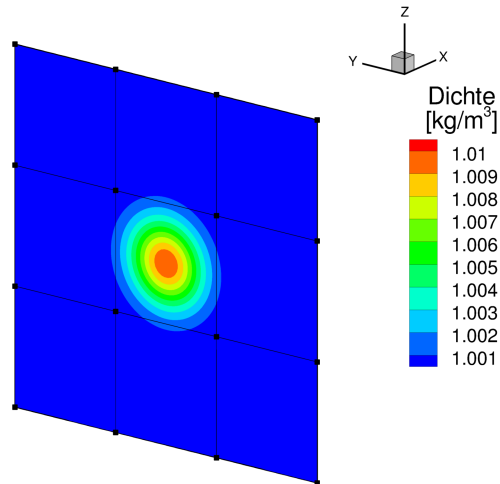


Abbildung 5.2: Gauß-Puls auf dem Netz mit 54 Elementen bei linearer Approximation ($p = 1$)

Als Beispiel sind die Initialisierungen der verschiedenen Approximationsordnungen auf dem Netz mit 2000 Elementen in Abbildung 5.3 zu sehen.

Man erkennt sofort, dass je höher die Approximationsordnung ist, desto glatter wird der Gauß-Puls dargestellt. Dies liegt an der steigenden Anzahl von Interpolationspunkten. Die ungewöhnlich glatte Form von $p = 0$, siehe Abb. 5.3a, ist durch eine interne Interpolation der elementzentrierten Lösungen des Visualisierungstools Tecplot bedingt. Infolgedessen werden die Unstetigkeiten der Elementübergänge verschmiert. Bei höherer Ordnung wirkt sich diese Interpolation nicht so stark aus, da schon vordefinierte Werte auf den Rändern (Randknoten) der Elemente existieren. Im Folgenden wurden die initialisierten Dichtefluktuationen durch das Berechnungsgebiet geschickt und nach $1m$ ausgewertet.

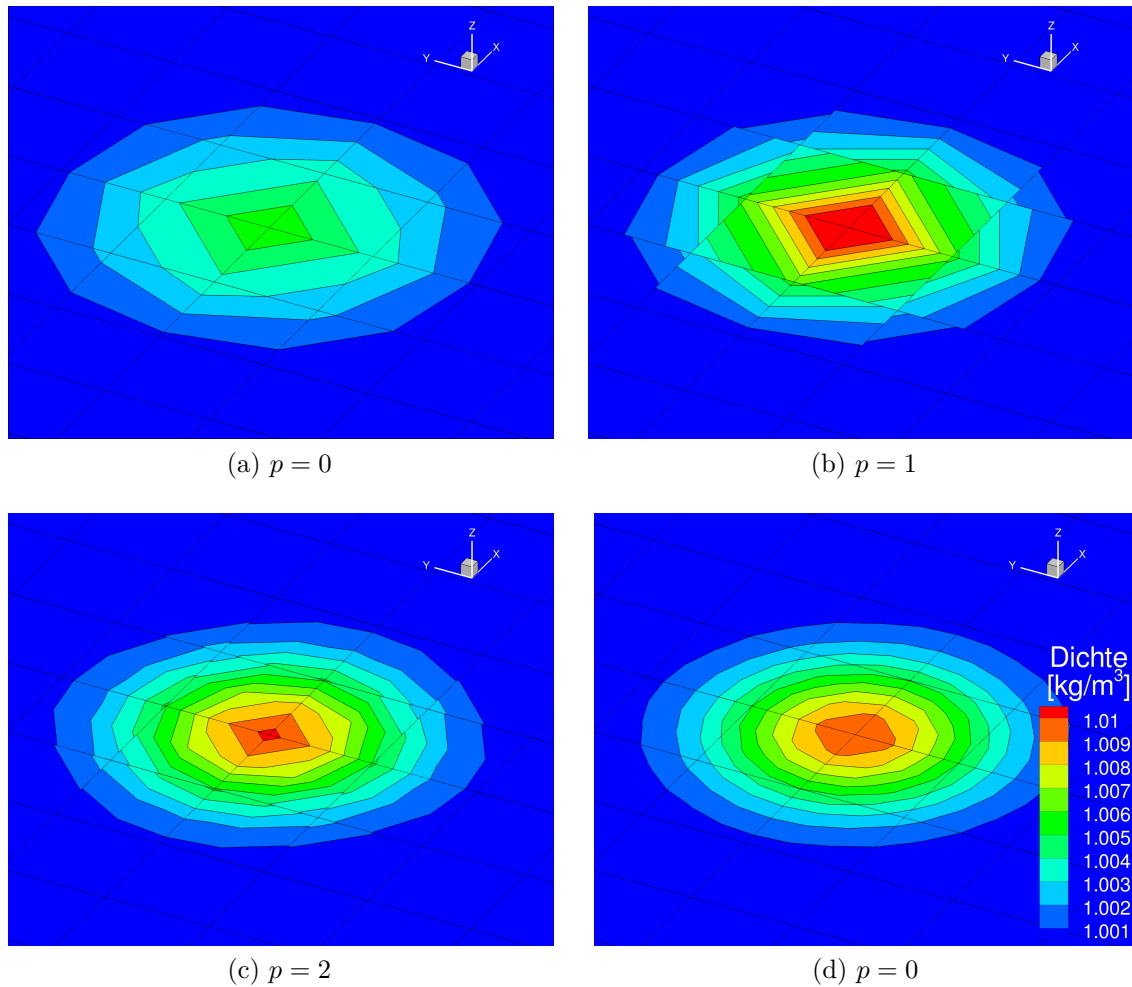


Abbildung 5.3: Initialisierung auf dem Netz mit 2000 Elementen bei Verwendung verschiedener Approximationsordnungen, gezoomter Schnitt in der xy -Ebene bei $z = 0.5m$

5.1.2 Auswertung für DGNODAL

Die Ergebnisse nach der Simulation auf dem 2000er Netz sind in Abbildung 5.4 dargestellt. Man sieht deutlich die numerische Dissipation (Diffusion) bei der konstanten und der linearen Approximation (Abb. 5.4a und 5.4b). Diese Dissipation ist durch die Approximation des numerischen Flusses bedingt und sorgt für eine Art Verschmierung der Lösung. Für $p = 0$ hat sich der Dichtepuls so ausgedehnt, dass er durch die periodischen Randbedingungen wieder ins Berechnungsgebiet eintritt. Allgemein gilt, je höher die verwendete Polynomordnung ist, desto hochwertiger wird die Flussnäherung.

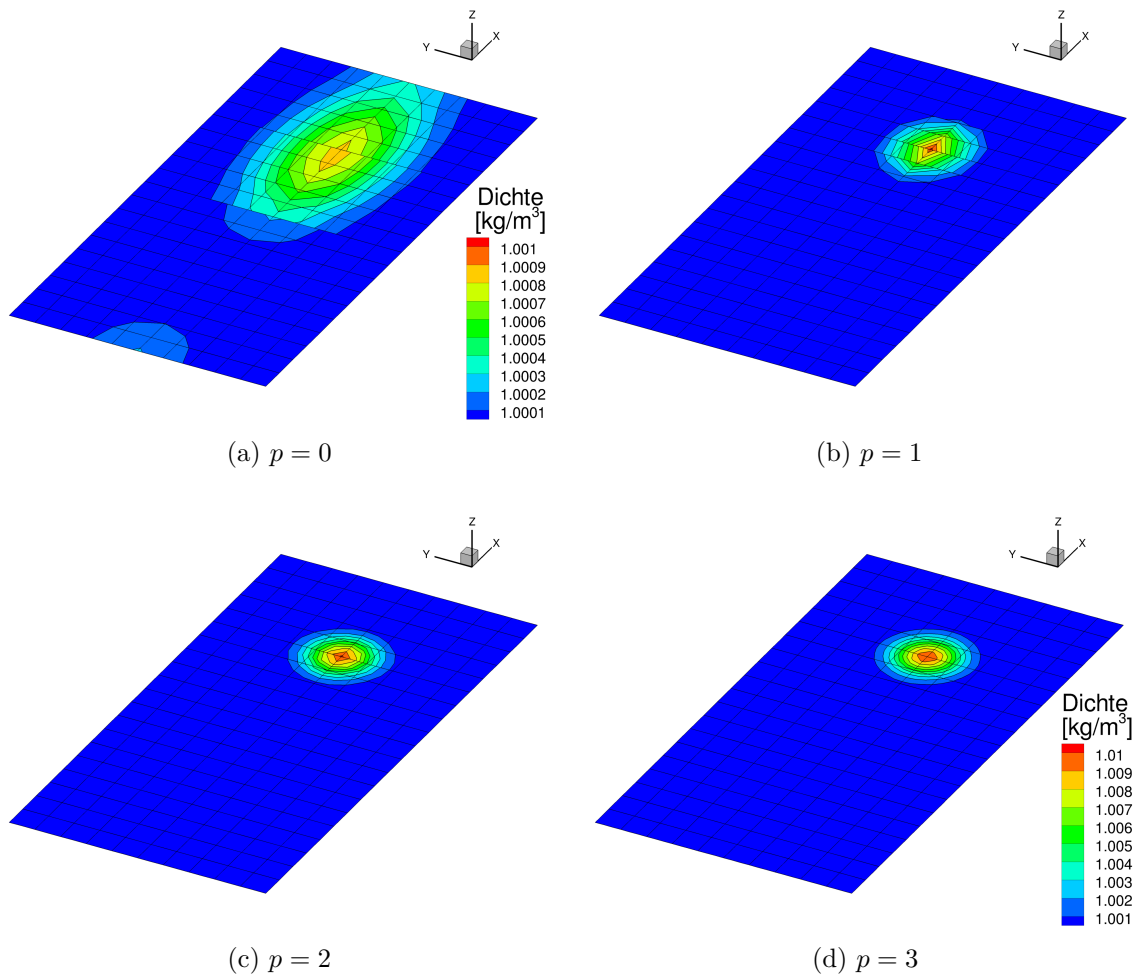


Abbildung 5.4: Simulationsergebnisse auf dem Netz mit 2000 Elementen bei Verwendung verschiedener Approximationsordnungen, Schnitt in der xy -Ebene bei $z = 0.5m$

Für die Visualisierung des konstanten Falls (Abb. 5.4a) musste die Skala angepasst werden, da das Maximum nach der Simulation weniger als 10% des Ausgangsmaximums beträgt. Für die anderen Darstellungen gilt die bisher verwendete Skala bis zu einer Dichte von 1.01 kg/m^3 . Bei der Lösung mit $p = 3$ erkennt man schon kaum einen Unterschied zum initialisierten Gauß-Puls, vgl. Abbildung 5.3d und 5.4d.

Für die Auswertung wurde die Differenz mit der exakten Lösung in jedem Knoten und daraufhin die L^2 -Norm gebildet. Die sich ergebenden Fehler sind über die Netzgröße h im LogLog-Diagramm 5.5 aufgetragen.

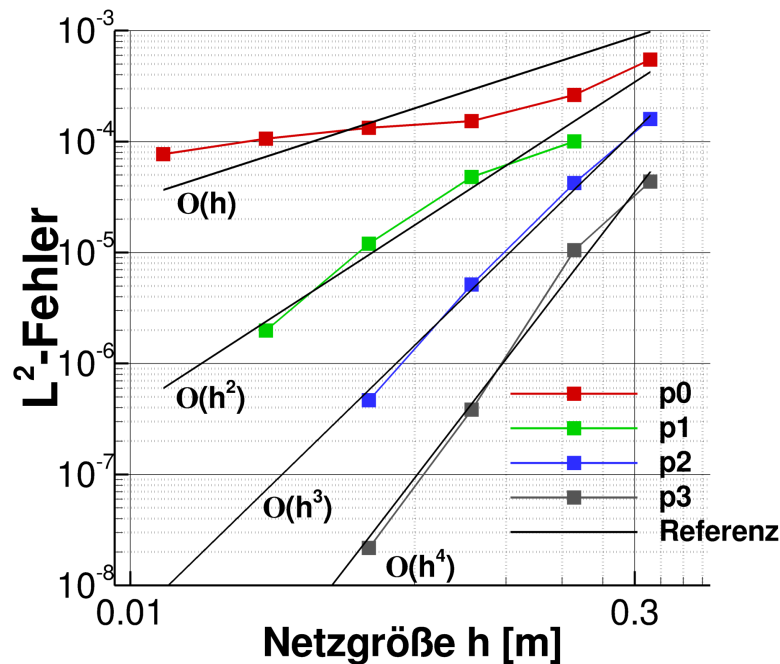


Abbildung 5.5: Konvergenzraten der DGNODAL-Implementierung

Die Steigung der Kurven stellt die Konvergenzordnung des Verfahrens da. Hierzu wurden jeweils die vorausgesagte Konvergenz als schwarze Referenzkurve geplottet. Im Mittel wurden alle erwarteten Steigungen getroffen. Weitere Verfeinerungsstufen konnten nicht mehr gerechnet werden, da die lange Rechenzeit den Nutzen (Aussagekraft) übersteigt. Herauszustellen ist, dass die Approximationsordnungen $p \geq 1$ die Konvergenzgeschwindigkeiten bei feineren Netzen sogar zu übertreffen scheinen. Dies kann aus der Verwendung regelmäßiger, kartesischer Netze resultieren, da die Fehlerabschätzung für allgemeine Zerlegungen durchgeführt wurde. Außerdem fällt auf, dass die Kurve der konstanten Approximation teilweise die erwartete Konvergenz nicht trifft. Die Steigung scheint sich jedoch zum Ende hin anzupassen und es wird erwartet, dass sich für feinere Netze die Konvergenzrate wieder einstellt. Für den praktischen Nutzen ist ebenfalls der Vergleich der Freiheitsgrade und der benötigten CPU Zeit sehr interessant (siehe Abbildung 5.6).

Bei der gleichen Anzahl an Freiheitsgraden reduziert sich der Fehler mit steigender Approximationsordnung deutlich, siehe Abb. 5.6a. Dies lässt den Rückschluss zu, dass die Simulation auf einem groben Netz mit hoher Approximationsordnung genauere Ergebnisse liefert als auf einem feinen Netz unter Verwendung einer niedrigen Ordnung bei gleicher Anzahl an Freiheitsgraden. Dieses Pro-Argument für die Verwendung von Verfahren höherer Ordnungen bestätigt sich auch in der CPU-Zeit, vgl. Abb. 5.6b. Zur

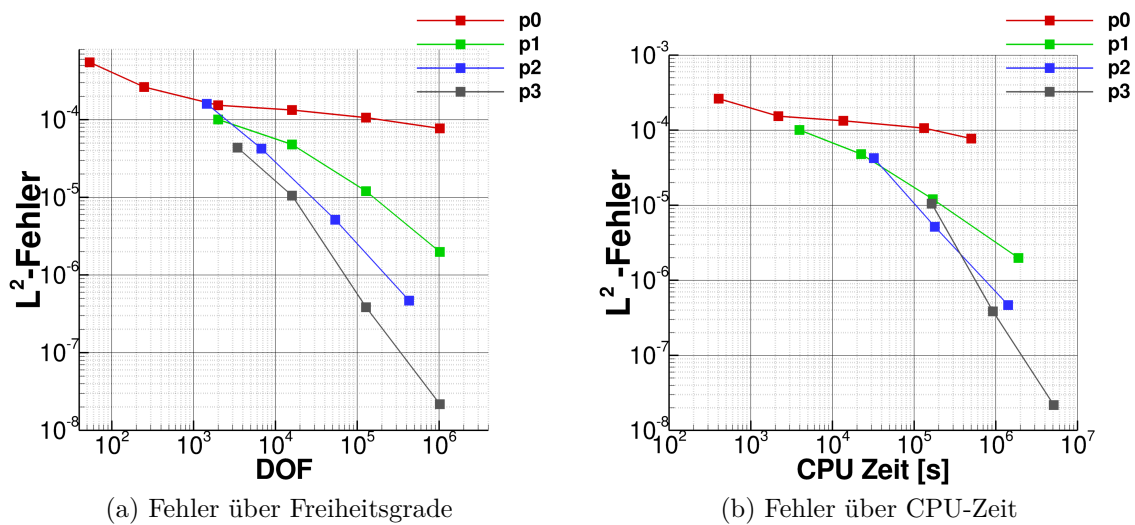


Abbildung 5.6: L^2 -Fehler über die Freiheitsgrade des Modells und die CPU-Zeit (DGNODAL)

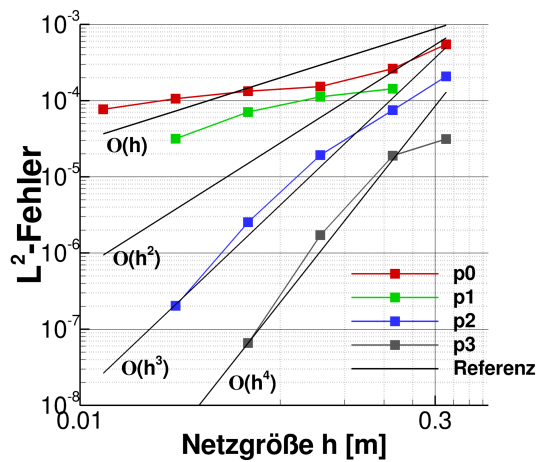
Erreichung einer gewissen Fehlerordnung ist $p = 3$ deutlich schneller als die niedrigen Ordnungen. Gleichfalls muss man anerkennen, dass diese Fehlerordnung sehr klein ist. Es stellt sich in der praktischen Anwendung die Frage, in wie weit ein Fehler dieser Größe mit den physikalischen Annahmen (z.B. Turbulenzmodellierung) für die Simulation realer Strömungen überhaupt eine Rolle spielt. Nichtsdestotrotz zeigen sich hier die Potentiale des DG-Verfahrens höherer Ordnung, die in Verbindung mit hochauflösenden Verfahren, wie beispielsweise der Direkten Numerischen Simulation (DNS), im Bereich der Forschung schon genutzt werden.

5.1.3 Auswertung für DGSEM

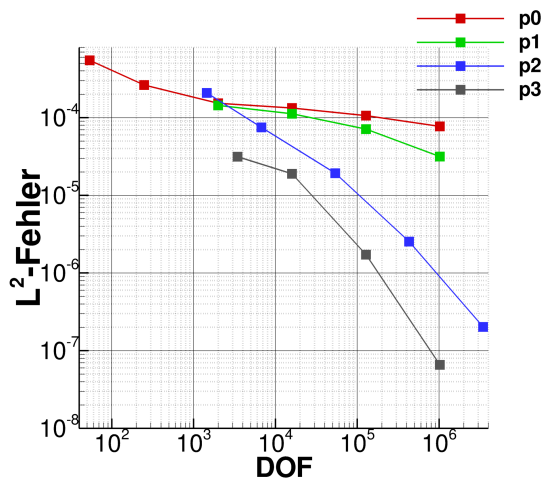
Die Ergebnisse der Konvergenzstudie sind zusammengefasst in Abbildung 5.7 dargestellt. Im Großen und Ganzen treffen die Beobachtungen der DGNODAL - Implementierung ebenfalls auf das DGSEM-Verfahren zu.

Auffallend ist hier, dass die Konvergenzordnungen auf den gröberen Netzen nicht getroffen werden, siehe Abb. 5.7a. Dies könnte daran liegen, dass auf groben Netzen der geringere Exaktheitsgrad der Integration einen großen Einfluss auf den gesamten Fehler hat. Insbesondere bei der linearen Approximation scheint sich die Verschiebung der Integrationspunkte auf den Rand des Elements stark auszuwirken. Bei feineren Netzen wird die Konvergenz jedoch gut getroffen und wie beim DGNODAL-Verfahren teilweise übertroffen. Weiterhin lässt sich feststellen, dass der Verlauf der $p = 0$ Kurve gleich dem des DGNODAL-Verfahrens ist, vgl. Abb. 5.5 und Abb. 5.7a. Dies ist nachvollziehbar, da die Knoten und die Integrationspunkte im konstanten Fall für

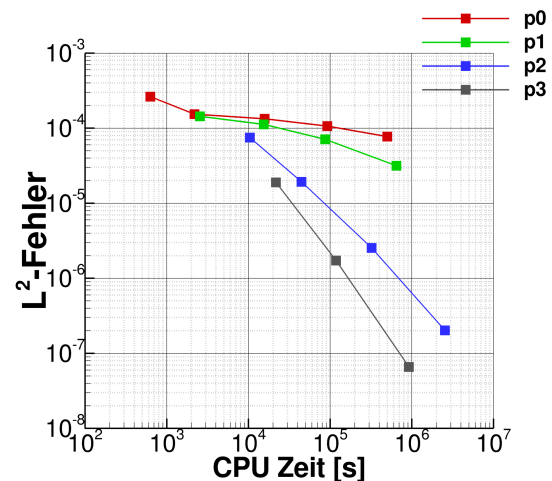
beide Verfahren identisch sind. Herauszustellen ist außerdem, dass sich eine hohe Approximationsordnung bei Betrachtung der benötigten CPU Zeit viel früher lohnt als beim DGNODAL-Verfahren, siehe dazu Abb. 5.6b und 5.7c. Das heißt, dass $p = 3$ schon für die Erreichung eines vergleichsweise großen Fehlers schneller als die niedrigeren Ordnungen ist. Daraus kann man folgern, dass das DGSEM-Verfahren besser für die Verwendung hoher Approximationsordnungen geeignet ist und erst dann seine Stärken zeigt.



(a) Fehler über Netzgröße



(b) Fehler über Freiheitsgrade



(c) Fehler über CPU Zeit

Abbildung 5.7: Konvergenzraten der DGSEM-Implementierung

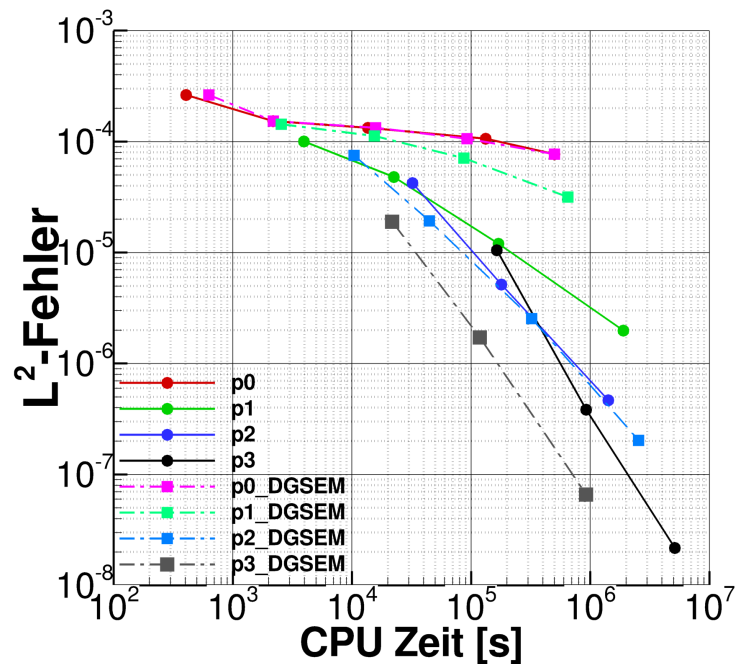


Abbildung 5.8: Vergleich der CPU Zeiten von DGNODAL (—) und DGSEM (----)

5.1.4 Vergleich von DGNODAL und DGSEM

Interessant ist nun der Vergleich von DGSEM und DGNODAL. In Kapitel 3.2.3 wurden die Vor- und Nachteile des DGSEM-Verfahrens in der Theorie gezeigt. Diese gilt es in den numerischen Untersuchungen zu belegen. Hierzu wird die CPU Zeit betrachtet und die Ergebnisse beider Implementierungen in einem Diagramm dargestellt, siehe Abb. 5.8.

De facto zeigen sich die Vorteile des DGSEM-Verfahrens erst für höhere Approximationsordnungen. Bei einer konstanten Approximation verhalten sich die beiden Verfahren annähernd gleich. Die geringen Differenzen in den Simulationszeiten sind durch die unterschiedlichen Implementierungsweisen bedingt. Bei einer Polynomordnung von $p = 1$ hat das DGNODAL-Verfahren noch Vorteile, da sich der Fehler der LGL-Integration zu sehr bemerkbar macht. Es lässt sich jedoch feststellen, dass schon bei dieser niedrigen Ordnung DGSEM Geschwindigkeitsvorteile bietet. Ab einer Ordnung von $p \geq 3$ wird der Nutzen des DGSEM-Verfahrens sichtbar. Hier hat die ungenauere Integration nur noch einen geringen Einfluss auf den Gesamtfehler und die Vorteile in der Geschwindigkeit kommen zur Geltung. Bei der kubischen Approximation ist das DGSEM-Verfahren in jedem untersuchten Fall schneller. Mit dem Ansatz die DG-Ortsdiskretisierung mit möglichst hohen Ordnungen nutzen zu wollen, ist DGSEM deutlich besser geeignet als DGNODAL.

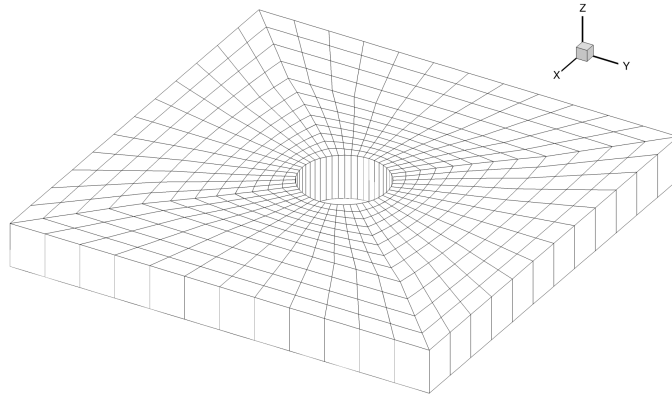


Abbildung 5.9: Vernetzung des umströmten Zylinders mit ca. 600 Hexaeder- Elementen

5.2 Umströmung eines Zylinders mit gekrümmten Rändern

Als ein nächstes Beispiel soll die Umströmung eines Zylinders ($Ma = 0.3$, $\alpha = 0^\circ$) untersucht werden. Das verwendete unstrukturierte Netz ist in Abbildung 5.9 dargestellt. In z -Richtung wurde die Vernetzung nur mit einem Element realisiert, da bei einem gleichmäßigen Profil keine Strömung in diese Richtung erwartet wird. Beim Einlass und Auslass werden Riemann-Randbedingungen verwendet. Die Zylinderoberfläche wird als reibungsfreie Wand und die seitlichen Begrenzungsflächen als periodisch modelliert.

Nach [31] stellt sich für geringe Machzahlen eine stationäre Strömung ein, die längs der mittleren y -Achse symmetrisch ist.

Die Mach-Kontur der Approximationsordnung $p = 2$ mit der Verwendung von geraden Randelementen ist in der Abbildung 5.10 zu sehen. Insbesondere in der vergrößerten Ansicht sind die Oszillationen an den Randflächen deutlich zu erkennen, siehe Abb. 5.10b. Wie in Kapitel 3.2.4 aufgeführt, entstehen diese Oszillationen durch die inkorrekte Darstellung der Randflächen und sind physikalisch nicht zu begründen. Durch die Erzeugung numerische Viskosität an diesen Stellen, entsteht ein Nachlauf hinter dem Profil (Abb. 5.10a). Darin kann man sogar scheinbar instationäre Effekte wie eine Wirbelbildung feststellen, die für die Euler-Gleichungen nicht auftreten dürften.

Im Folgenden sollen gekrümmte Ränder für die Simulation verwendet werden. Die Erstellung dieser Randelemente erfolgt über das Pre-Processingtool PREP, vgl. [13]. In Abbildung 5.11 ist der Unterschied zwischen den entstehenden Knoten für die Polynomordnung $p = 2$ dargestellt. Man sieht hier, wie sich die Knoten nun an die Krümmung der Oberfläche anpassen.

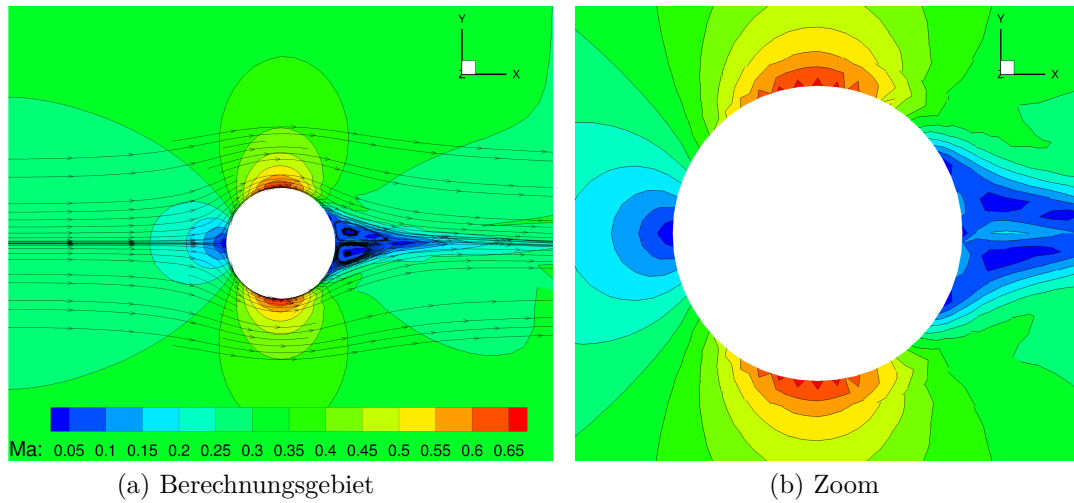


Abbildung 5.10: Mach-Kontur und Strömungslinien der Strömung um einen Zylinder unter der Verwendung von geraden Randelementen mit quadratischen Ansätzen

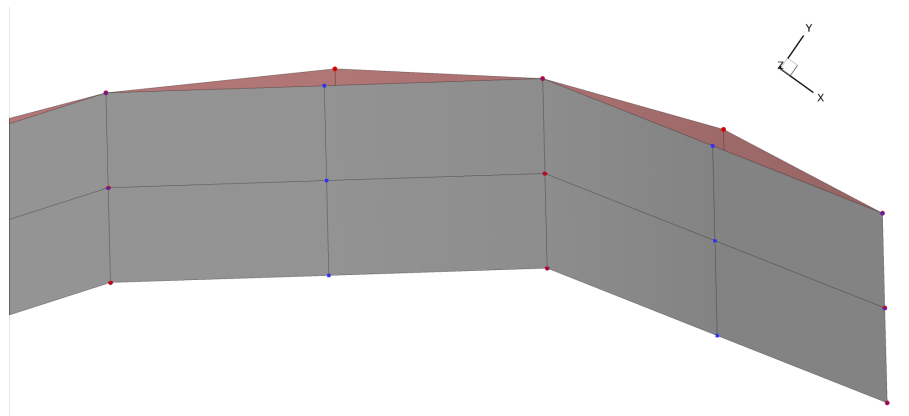


Abbildung 5.11: Vergleich der entstehenden Knoten für die Ordnung $p = 2$ mit geraden Randelementen (blau) und gekrümmten Elementen (rot)

Obwohl die Unterschiede in den Positionen der Knoten nicht außerordentlich groß sind, stellt sich eine deutliche Abweichung im Strömungsverlauf ein. Dieser ist für die Approximationsordnung $p = 2$ in Abbildung 5.12 zu sehen.

Durch die Verwendung von Randelementen mit quadratischer Flächendarstellung ergibt sich ein gleichmäßigerer Verlauf der Strömungskontur. Auch die Oszillationen an den Randflächen sind nicht mehr vorhanden. Die Symmetrie längs der mittleren y -Achse kann nur näherungsweise beobachtet werden, da sich hinter dem Profil Spitzen in den Konturlinien bilden. Diese resultieren aus der Approximation der Oberflächenkontur. Zwar ist diese durch die quadratische Darstellung angenähert

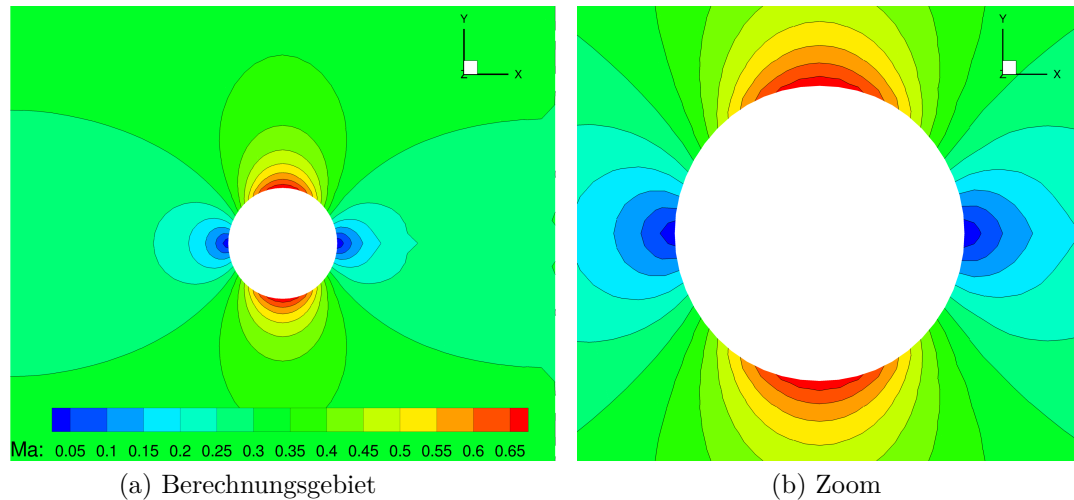


Abbildung 5.12: Mach-Kontur der Strömung um einen Zylinder unter der Verwendung von gekrümmten Randelementen (quadratische Flächendarstellung) mit quadratischen Ansätzen (links) und eine vergrößerte Ansicht (rechts)

worden, entspricht aber noch nicht der genauen Kontur. Dadurch bildet sich weiterhin ein leichter Nachlauf, der in den Konturspitzen sichtbar wird. Zur Behebung dieses Effekts müsste die Oberfläche noch genauer approximiert werden.

Zusätzlich zu den genannten Effekten kann man eine Entropieproduktion an den ungenau approximierten Rändern feststellen. Üblicherweise ist die Erzeugung von Entropie ein Effekt, der bei reibungsbehafteten Wänden zu sehen ist und daher bei einer nicht-viskosen Strömung nicht auftreten sollte. Um die Güte der numerischen Approximation zu verifizieren, kann dementsprechend die Entropie herangezogen werden. Für die DG-Diskretisierung mit geraden und gekrümmten Elementen wurde dies in Abbildung 5.13 realisiert.

Die ausgeprägte Entropieproduktion bei geraden Elementen, Abb. 5.13a, ist offensichtlich und spricht für den Fehler der Approximation. Doch auch bei gekrümmten Elementen, Abb. 5.13b, kann ein leichter Anstieg festgestellt werden. Dies bestätigt die vorherigen Beobachtungen des schwach erkennbaren Nachlaufs, vgl. Abb. 5.12a.

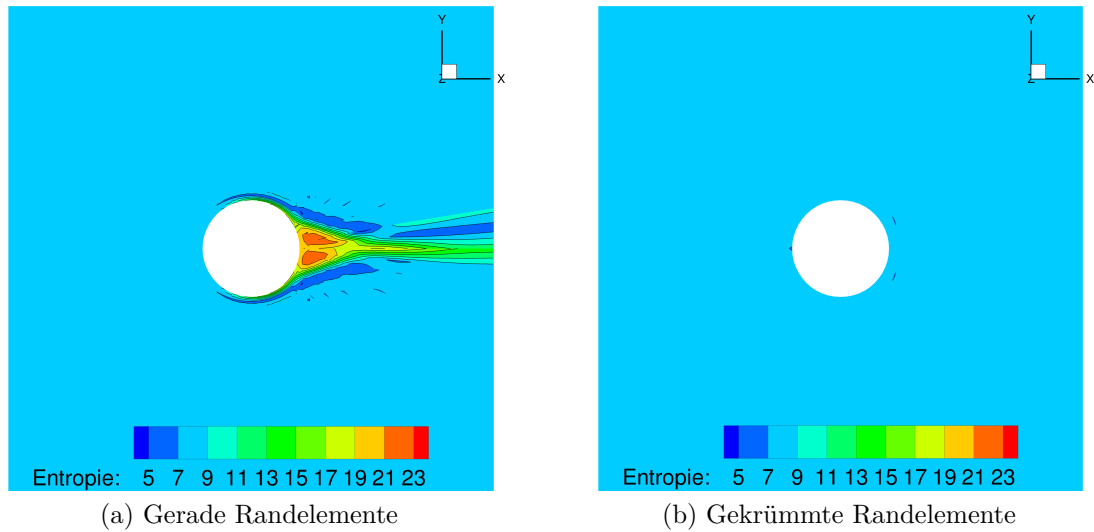


Abbildung 5.13: Vergleich der Entropie bei der reibungsfreien Umströmung eines Zylinders

Um die Ergebnisse der geraden und gekrümmten Ränder zu vergleichen, soll außerdem der Druckbeiwert c_p am Profil ausgewertet werden. Der Druckbeiwert ist eine dimensionslose Größe aus der Aerodynamik und stellt, bezogen auf einem Punkt, den Zusammenhang zwischen dem statischen und dem dynamischen Druck dar. Für eine nahezu inkompressible Strömung um einen starren Körper, ist der Druckbeiwert definiert nach [31]

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho c_\infty^2}, \quad (5.2)$$

wobei p der statische Druck am Auswertungspunkt, p_∞ der statische Druck in der Zuströmung und c_∞ die Anströmungsgeschwindigkeit ist. In einer inkompressiblen Strömung wird der Druckbeiwert im Staupunkt maximal Eins. Für die Umströmung eines Zylinders unter Vernachlässigung der Viskosität, existieren zwei Staupunkte, vor und hinter dem Profil. Die Auswertung des c_p -Wertes erfolgt in den Flächenmittelpunkten der Randelemente des Zylinders. In Abbildung 5.14 sind die Druckbeiwerte über die x -Koordinaten für gerade und gekrümmte Ränder dargestellt.

Der erwartete Druckbeiwert im Staupunkt vor dem Profil wird für alle Ordnungen gut getroffen. Im Folgenden ergeben sich jedoch Unterschiede. Die Verläufe der Druckkoeffizienten mit gekrümmten Elementen sind nahezu identisch und stimmen am Staupunkt hinter dem Zylinder mit dem erwarteten Wert $c_p = 1$ überein. Dies lässt auf eine annähernd auskonvergierte Lösung schließen, da die Erhöhung der Approximationsordnung kaum Unterschiede bringt. Die lineare Lösung $p = 1$ mit geraden Elementen kommt diesen Verlauf am nächsten, weicht jedoch insbesondere am

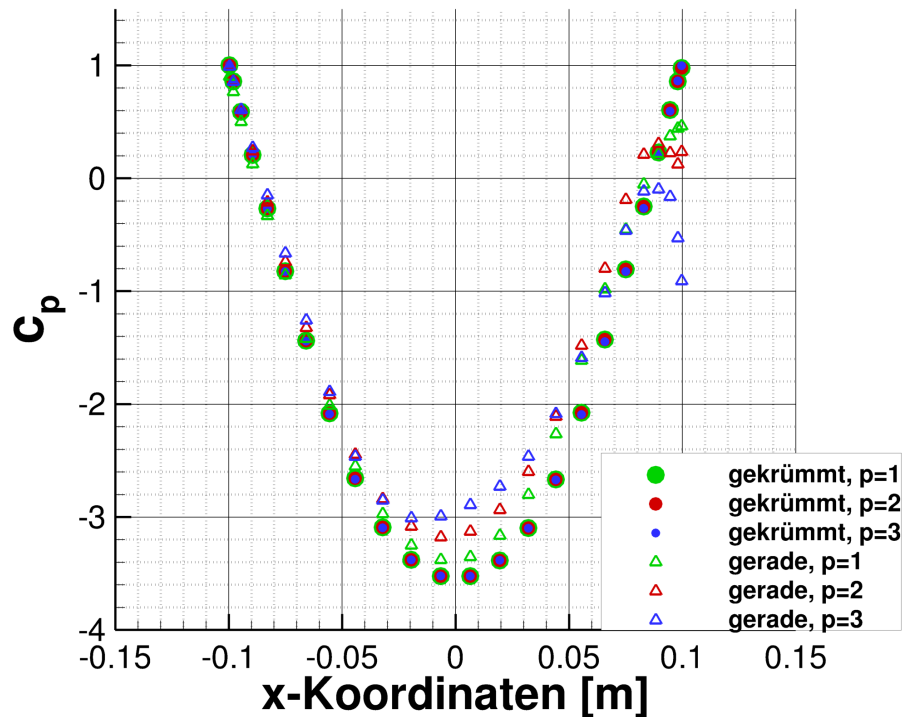


Abbildung 5.14: Vergleich der Druckbeiwerte am umströmten Zylinder von geraden und gekrümmten Elementen für verschiedene Approximationsordnungen

hinteren Staupunkt vom erwarteten Ergebnis ab. Erhöht man die Ansatzordnung auf geraden Elementen wird die Abweichung deutlich größer und ergibt unphysikalische Verläufe. Diese Untersuchungen zeigen die dringende Notwendigkeit gekrümmter Randelemente bei Verwendung der DG-Diskretisierung. Ohne Randelemente höherer Ordnung wird die Oberflächenkontur zu ungenau approximiert. Nach [4] führt dies in der Folge zu Oszillationen an den Randflächen und unphysikalischen Strömungsverläufen.

5.3 Umströmung eines Tragflügelprofils

In diesem Abschnitt soll die Strömung um ein anwendungsnahe Tragflächenprofil mit der implementierten DG-Diskretisierung untersucht werden. Die Ergebnisse werden mit der Referenzlösung des FV-Solvers TRACE auf einem sehr feinen Netz verglichen.

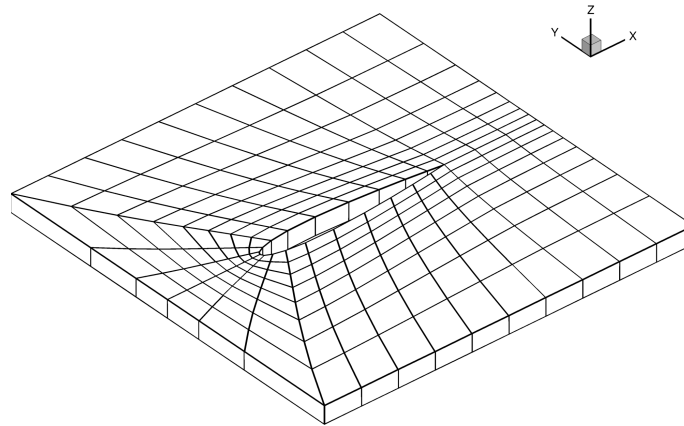


Abbildung 5.15: Vernetzung des NACA0012-Profiles mit ca. 200 Hexaeder- Elementen

Betrachtet wird die subsonische, nicht viskose Strömung um das NACA0012-Profil bei einem Anströmwinkel $\alpha = 1.49^\circ$ und der Machzahl $Ma = 0.3$. Wie in der Konvergenzstudie gezeigt, liegen die Vorteile der Verfahren höherer Ordnungen in der Möglichkeit, auf sehr groben Netzen ausreichend genaue Ergebnisse zu simulieren. Aus diesem Grund wurde das in Abbildung 5.15 dargestellte, grobe Netz mit einem Element in z -Richtung für die DG-Diskretisierung verwendet.

Wie im letzten Beispiel gezeigt, muss die gekrümmte Oberfläche des Profils mit Randelementen höherer Ordnung vernetzt werden. Der Unterschied in den Mach-Verteilung ist für die Ansatzordnung $p = 2$ in Abbildung 5.16 zu sehen. Man erkennt auch bei diesem Profil eine deutlich gleichmäßigere Form der Strömung bei der Verwendung von gekrümmten Rändern.

Als Referzergebnis soll eine Simulation des FV-Solvers TRACE auf einem sehr feinen Netz dienen. Das Netz und die Mach-Konturen sind in Abbildung 5.17 dargestellt. Im Vergleich sehen die Mach-Konturen der Referenzlösung ähnlich denen der DG-Diskretisierung mit gekrümmten Elementen aus, vgl. dazu Abb. 5.16b und 5.17b. In beiden Fällen stellt sich eine glatte Form ein und die Auswirkungen des Anstellwinkels sind gut zu erkennen. Lediglich hinter dem Profil weichen die Ergebnisse leicht voneinander ab. Die Probleme von numerischen Verfahren an den spitzen Hinterkanten bei reibungsfreien Strömungen sind jedoch hinreichend bekannt und sollten damit nicht weiter betrachtet werden. Für nähere Informationen wird auf [2] verwiesen. Die Verwendung von geraden Randelementen, zu sehen in Abbildung 5.16a, führt dagegen zu sehr großen Abweichungen von der Referenzlösung.

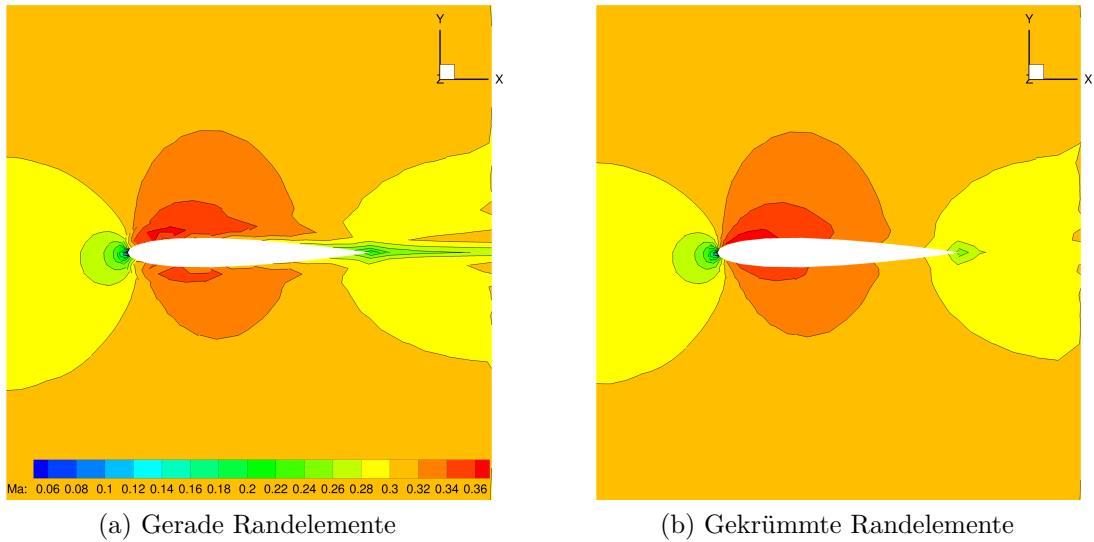


Abbildung 5.16: Vergleich der Mach-Konturen mit geraden und gekrümmten Randelementen für die Ansatzordnung $p = 2$

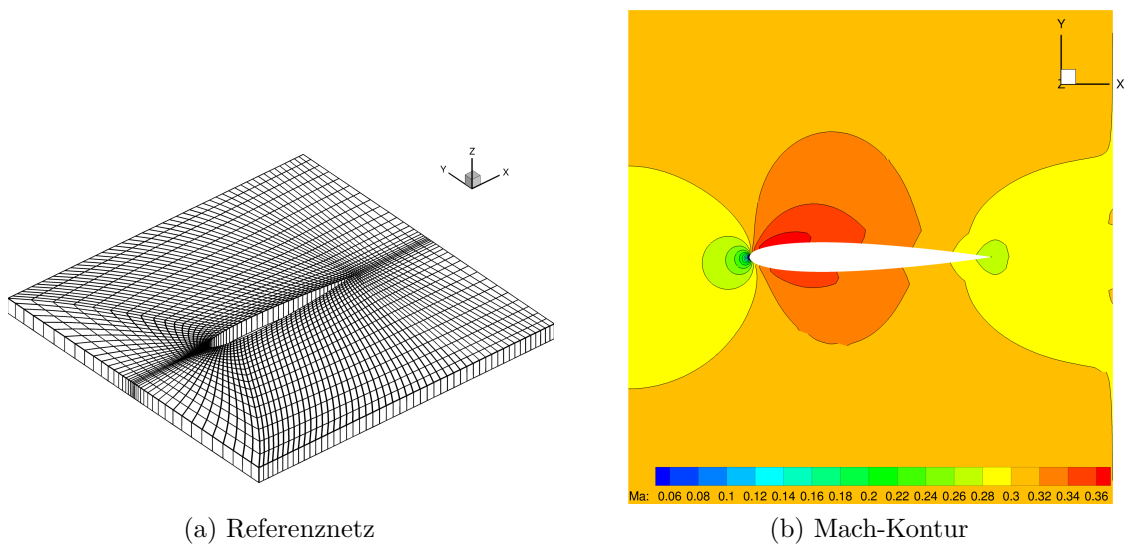


Abbildung 5.17: Referenznetz mit ca. 10000 Elementen und die Mach-Kontur auf diesem Netz mit der FV-Diskretisierung

Ein differenzierteren Blick auf die Unterschiede lässt die Betrachtung der Entropie zu. Hierfür werden die Entropie-Konturen der Referenzlösung auf dem feinen Netz, der FV-Diskretisierung auf dem groben Netz und der DG-Diskretisierung quadratischer Ansatzordnung mit geraden und gekrümmten Randelementen visualisiert. Herauszustellen ist, dass die FV-Diskretisierung auf dem groben Netz, Abb. 5.18a, für die größte Entropieproduktion sorgt und damit die ungenaueste der vorgestellten Varianten ist. Bei der Verwendung von geraden Randelementen ist die Entropiedifferenz der DG-Diskretisierung ebenfalls noch ausgeprägt (Abb. 5.18c). Mit gekrümmten Rändern sind dagegen nur noch geringe Abweichung von der Referenzlösung zu

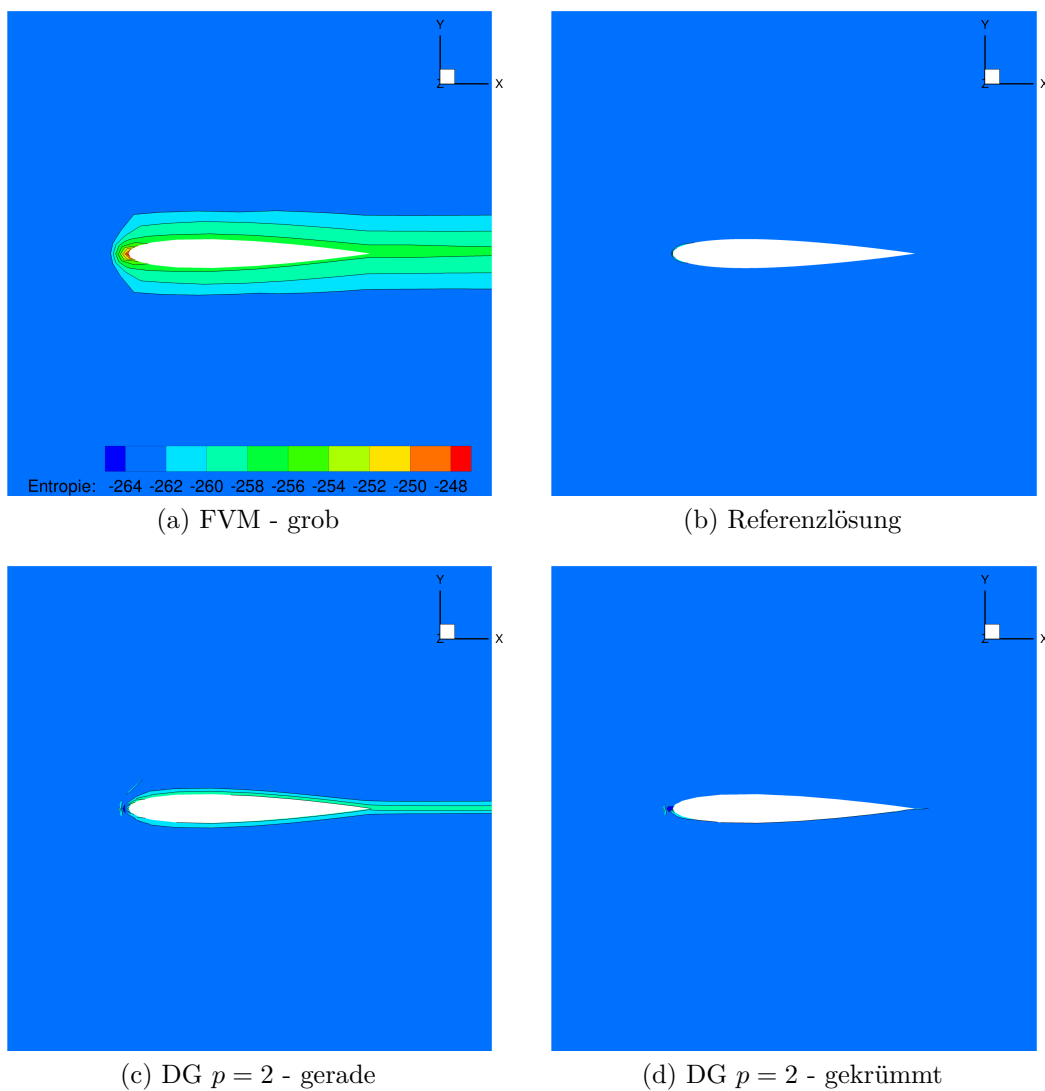


Abbildung 5.18: Vergleich der Entropie-Konturen

erkennen, vgl. Abb. 5.18d und 5.18b. Die Simulation vergleichbarer Ergebnisse auf deutlich größeren Netzen ist ein großer Vorteil der DG-Diskretisierung. Im Folgenden wird die Veränderung der Entropie-Verteilung bei Erhöhung der Polynomordnung untersucht, siehe Abbildung 5.19.

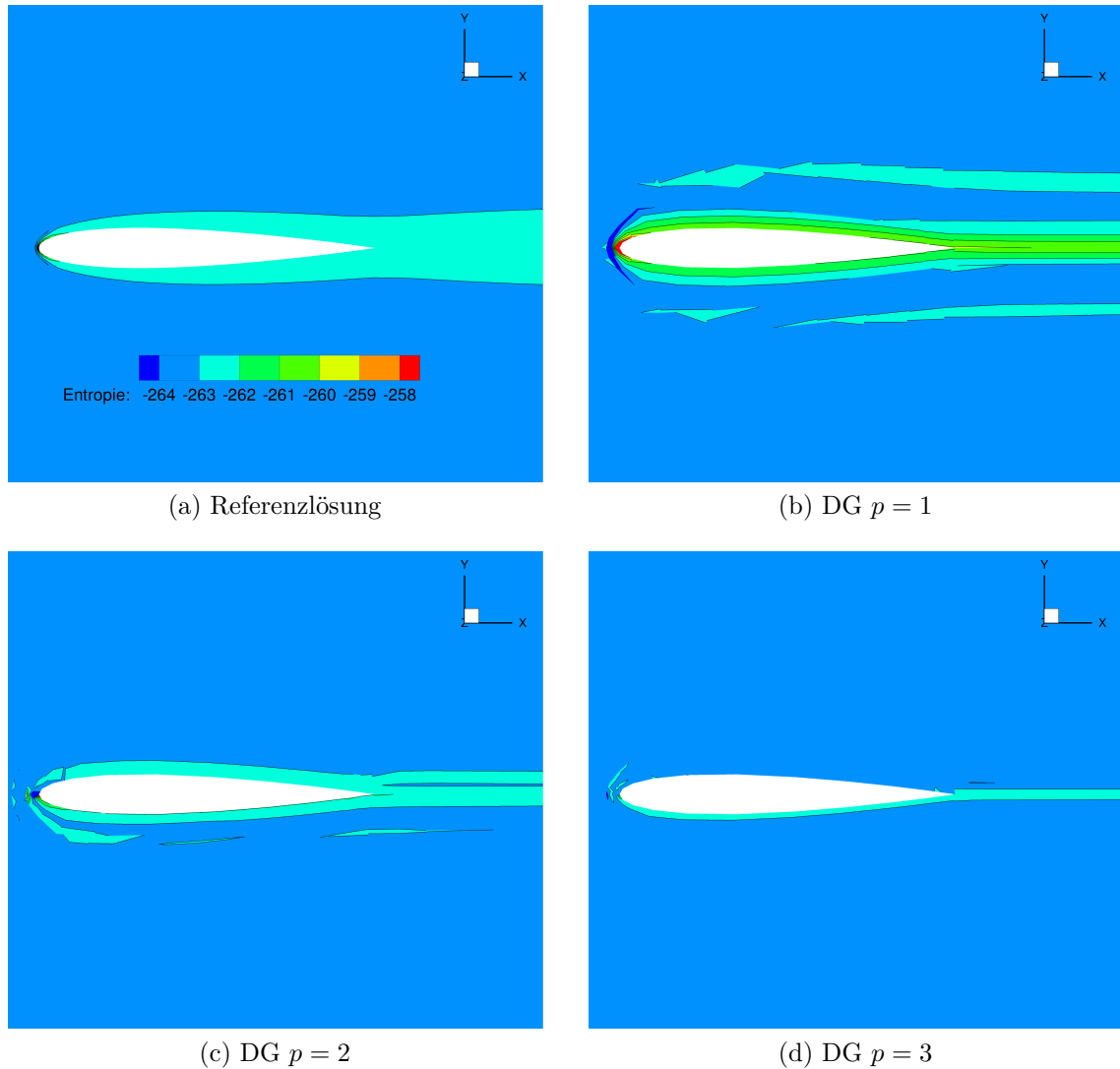


Abbildung 5.19: Vergleich der Entropie-Konturen für verschiedene Approximationsordnungen p mit gekrümmten Randelementen, Skala angepasst an die Referenzlösung

Die Skala wurde in diesen Visualisierungen auf die Referenzlösung angepasst. In diesem Rahmen weist das Ergebnis der DG-Diskretisierung erster Ansatzordnung (Abb. 5.19b) noch große Abweichungen von der Referenzlösung auf. Insbesondere vor dem Profil ist ein ausgeprägtes Maximum der Entropie zu erkennen. Erhöht man

die Diskretisierungsordnung, wird der Entropieverlauf gleichmäßiger. Die Lösung mit quadratischen Ansätzen ist der Referenzlösung am ähnlichsten, vgl. Abb. 5.19c und Abb. 5.19a. Die Verwendung kubischer Ansätze scheint sogar die Referenzlösung zu übertreffen und für eine geringere Produktion über das gesamte Profil zu sorgen. Lediglich direkt vor dem Profil existieren große Entropieunterschiede in allen DG-Diskretisierungsordnungen. Auffallend ist außerdem, dass bei allen DG-Ergebnissen Entropiesprünge im Berechnungsgebiet zu sehen, die nicht direkt am Profil liegen. Gründe dafür könnten die Unstetigkeiten an den Elementübergängen sein.

Für einen weiteren Vergleich der Approximationsordnungen sollen außerdem die Druckbeiwerte am Profil ausgewertet werden (siehe Abb. 5.20). Hierzu wird die Vorderseite des Profils betrachtet, da die c_p -Verteilung dort am aussagekräftigsten ist. Auf die Visualisierung der Ergebnisse von geraden Elementen wurde aufgrund der Übersichtlichkeit verzichtet.

Neben der Referenzlösung ist außerdem die c_p -Verteilung des groben Netzes mit FV-Diskretisierung visualisiert worden. Wie auch schon in Verbindung mit Abbildung 5.18 vermutet, zeigen sich deutliche Abweichungen und die DG-Ergebnisse sind im Allgemeinen näher an der Referenzlösung. Das Ergebnis mit linearen Ansätzen ist dabei noch am weitesten entfernt. Hier liegen die Druckbeiwerte an den Auswertungs-

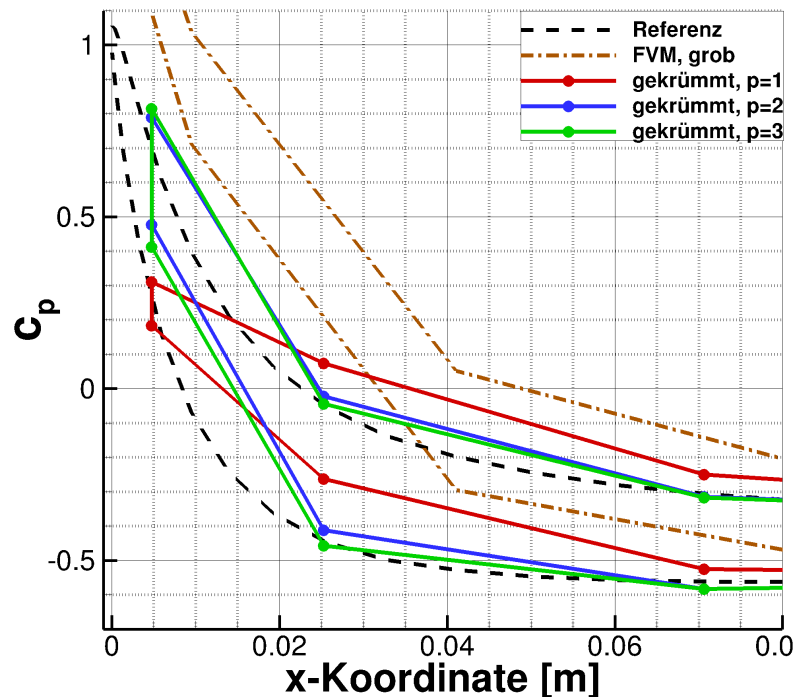


Abbildung 5.20: Vergleich der Druckbeiwerte am NACA-Profil verschiedener Approximationsordnungen ausgewertet in den Flächenmittelpunkten

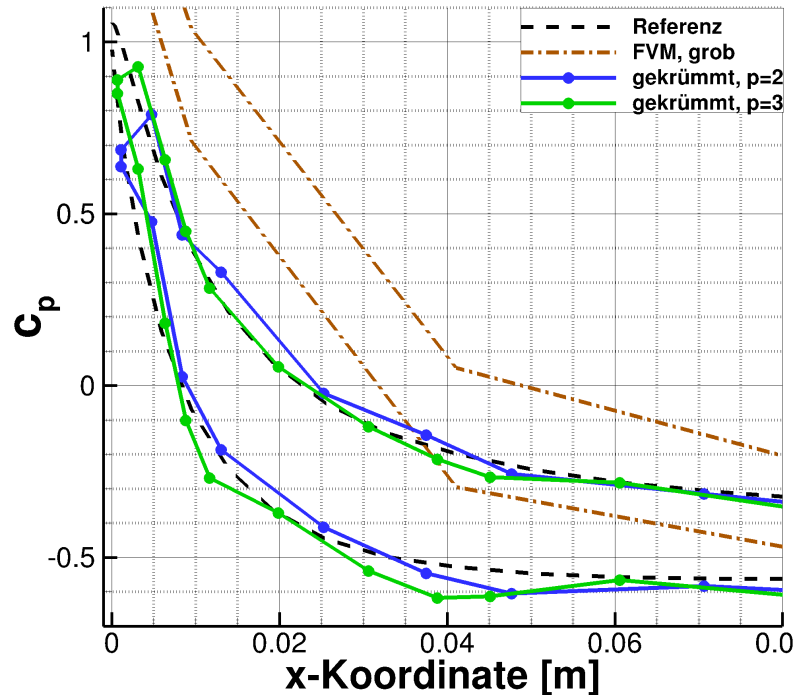


Abbildung 5.21: Vergleich der Druckbeiwerte am NACA-Profil verschiedener Approximationsordnungen ausgewertet in den Integrationspunkten

punkten deutlich neben der Referenzkurve. Erhöht man die Approximationsordnung bei Verwendung gekrümmter Elemente nähern sich die Werte denen der Referenzlösung an. Für $p = 3$ (grüne Kurve) ergibt sich eine gute Übereinstimmung. Dies konnte schon bei der Betrachtung der Entropie beobachtet werden und bestätigt sich hiermit. Aufgrund der wenigen Auswertungspunkte kann der genaue Verlauf der Kurven nicht verglichen werden. Hierzu sollen nun die Integrationspunkte der jeweiligen Ordnung als Auswertungspunkte dienen. Die sich ergebenden Verläufe sind in Abbildung 5.21 dargestellt.

Der Verlauf ist nun durch die erhöhte Anzahl an Auswertungspunkten deutlich glatter. Jedoch ergeben sich aufgrund der Unstetigkeiten zwischen den Elementen leichte Wellen im Verlauf. Allgemein lässt sich feststellen, dass die kubischen Ansatzordnung (grüne Kurve) dem erwarteten c_p -Wert im Staupunkt am nächsten erreicht. Auch der Druckverlauf unter dem Profil wird sehr gut getroffen. Oberhalb des Profils (kleineres c_p) entstehen jedoch größere Wellen im Verlauf, die zu einer Abweichung von der Referenzkurve führen. Ein Grund dafür könnte die noch nicht vollständig auskonvergierte Lösung bei kubischen Ansätzen sein. Durch die explizite Zeitintegration und der hohen Approximationsordnung benötigt das Erreichen der Maschinengenauigkeit eine gewisse Zeit und konnte innerhalb dieser Arbeit nicht mehr realisiert werden.

Abschließend lässt sich festhalten, dass durch eine Erhöhung der Ordnung eine deutliche Annäherung an die Referenzkurve erreicht werden konnte. Insbesondere bei Verwendung kubischer Ansätze können große Übereinstimmungen festgestellt werden. Lediglich die lange Simulationszeit bis zur auskonvergierten Lösung ist ein Nachteil, welcher sich jedoch durch den Einsatz von impliziten Verfahren zur Zeitdiskretisierung aufheben kann. Grundsätzlich zeigen sich hier die großen Vorteile für die Genauigkeit der Simulation, die die DG-Diskretisierung mit der Möglichkeit zur Erhöhung der Approximationsordnung bietet.

6 Fazit und Ausblick

Im Rahmen dieser Arbeit wurde der DG-Strömungslöser für die Euler-Gleichungen um Ansätze für Hexaeder-Elemente bis zur vierten Ordnung erweitert. Die Implementierungen wurden darüber hinaus in verschiedenen Testfällen verifizieren. An diesen Beispielen konnten die Potentiale der DG-Diskretisierung demonstriert werden.

Die durchgeführte Konvergenzstudie belegt die vorausgesagte Konvergenz von $\mathcal{O}(h^{p+1})$ für DGNODAL und DGSEM und zeigt die Abhängigkeit der Genauigkeits- von der Approximationsordnung p . Weiterhin wurde belegt, dass im Bereich hoher Genauigkeiten eine Erhöhung der Ordnung effizienter ist als die Verfeinerung des Netzes. Dies ist einer der Hauptgründe für die Verwendung von Verfahren höherer Ordnungen. Bei der Gegenüberstellung von DGSEM und DGNODAL konnte des Weiteren ein Vorteil in der Simulationsgeschwindigkeit für DGSEM festgestellt werden, der sich bei höheren Ordnungen verdeutlicht. Aus diesen Gründen ist für das DG-Verfahren mit Hexaeder-Elementen DGSEM zu bevorzugen. Ein Nachteil ist jedoch die komplizierte Übertragen des DGSEM-Verfahrens auf andere Elementtypen.

In einem nächsten Beispiel wurde die Notwendigkeit von gekrümmten Rändern bei der DG-Diskretisierung gezeigt und deren Implementierung für DGNODAL validiert. Hierzu wurde die nicht-viskose Strömung um einen Zylinder mit geraden und gekrümmten Randelementen betrachtet. Die Vernetzung mit geraden Randelementen erzeugt dabei eine unphysikalische Lösung mit der Bildung eines Nachlaufs. Im Vergleich dazu ergibt die Verwendung gekrümmter Ränder einen deutlich gleichmäßigeren Strömungsverlauf. Dies bestätigt sich auch in der Darstellung der Entropie-Konturen und in den Verläufen der Druckbeiwerte am Zylinder.

Als ein anwendungsnaher Testfall sollte die reibungsfreie Strömung um ein Tragflügelprofil untersucht werden. Dabei wurde die DG-Diskretisierung auf einem groben Netz mit der Referenzlösung des FV-Solvers TRACE auf einem feinen Netz verglichen. Mit steigender Ansatzordnung nähert sich die DG-Lösung an die Referenzlösung an. Dies konnte insbesondere bei der Visualisierung der Druckbeiwerte am Profil verdeutlicht werden. In der Darstellung der Entropie-Konturen zeigt die DG-Lösung vierter Ordnung sogar ein besseres Ergebnis als die Referenz.

Aufgrund der Stabilitätsbedingung expliziter Zeitdiskretisierungsverfahren konnten in den numerischen Untersuchungen nur kleine Zeitschritte verwendet werden. Dies führte insbesondere bei hohen Approximationsordnungen zu sehr langen Rechenzeiten

bis zur Erreichung einer auskonvergierten Lösung. Durch die Implementierung eines impliziten Verfahrens zur Zeitintegration könnte sich die Rechenzeit für stationäre Simulationen deutlich verringern.

Der nächste logische Schritt in der Weiterentwicklung des DG-Strömungslösers ist die Betrachtung der viskosen Terme. Hierzu wäre die Diskretisierung der RANS-Gleichungen mit dem DG-Verfahren und die Kopplung mit verschiedenen Turbulenzmodellen ein erstes Teilziel. Die großen Potentiale der DG-Ortsdiskretisierung liegen aber in Verbindung mit hochauflösenden Verfahren wie zum Beispiel der Large Eddy Simulation (LES), der Detached Eddy Simulation (DES) oder der Direct Numerical Simulation (DNS), weshalb diese das Hauptziel sein sollten. In diesem Zusammenhang ist die Erhöhung der implementierten Ansatzordnungen eine weitere denkbare Aufgabe.

Eine spannende Problematik wäre außerdem die Simulation verschiedener Elementtypen innerhalb eines Netzes oder die Betrachtung von hängenden Knoten. Um das volle Potential der Methode zu nutzen, wäre schlussendlich auch die Implementierung von verschiedenen Adaptionstrategien denkbar.

Literaturverzeichnis

- [1] Trace-portal. <http://www.trace-portal.de/wiki/index.php/TRACE>, 2013. Online; Zugriff 25. Januar 2015.
- [2] J. Anderson. *Computational fluid dynamics*. McGraw-Hill Science/Engineering/Math, 1 edition, February 1995.
- [3] M.A. Augustin. *Numerische Untersuchungen eines unstetigen Galerkin-Verfahrens zur Lösung der Konvektions-Diffusions-Gleichung*. Diplomarbeit, Naturwissenschaftliche-Technische Fakultät der Universität des Saarlandes, 2009.
- [4] F. Bassi und S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations. *J. Comput. Phys.*, 131(2):267–279, 1997.
- [5] C.E. Baumann und J. T. Oden. A discontinuous hp finite element method for convection—diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175(3–4):311 – 341, 1999.
- [6] J. Blazek. *Computational fluid dynamics: principles and applications; 2nd ed.* Elsevier, Burlington, MA, 2006.
- [7] S. Carstens. *Diskontinuierliche Galerkin-Verfahren in Raum und Zeit zur Simulation von Transportprozessen*. Schriftenreihe Fachgebiet Baumechanik/Baudynamik. Kassel University Press, 2013.
- [8] S. Cherednichenko, C. Frey, und G. Ashcroft. On the application of the discontinuous Galerkin method to turbomachinery flows. *6th European Congress on Computational Methods in Applied Sciences and Engineering*, pages 2359–2375, 2012.
- [9] B. Cockburn. Discontinuous Galerkin methods for convection dominated problems. School of Mathematics, University of Minnesota.
- [10] B. Cockburn und C.-W. Shu. The Runge-Kutta local projection P1-discontinuous-Galerkin finite element method for scalar conservation laws. *IMA Preprint Series*, 388, 1988.

- [11] B. Cockburn und C.-W. Shu. Runge- Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2002.
- [12] M.O. Deville, P.F. Fischer, und E.H. Mund. *High-order methods for incompressible fluid flow*. Cambridge University Press, 2002.
- [13] S. Drapkina, C. Frey, und G. Ashcroft. On the integration of high-order boundary elements in a 3d discontinuous Galerkin method for turbomachinery flows. In *European Conference on Computational Fluid Dynamics - ECFD VI*, Juli 2014.
- [14] M. Dumbser und M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221(2):693 – 723, 2007.
- [15] J.H. Ferziger und M. Peric. *Numerische Strömungsmechanik*. Springer Berlin, Heidelberg, 2008.
- [16] P.F. Fischer. Galerkin methods for incompressible flow simulation. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, 2009.
- [17] G. Gassner. *Discontinuous-Galerkin-Verfahren*. Vorlesungsskript. Institut für Aerodynamik und Gasdynamik der Universität Stuttgart, 2013.
- [18] T. Grätsch. *Eine auf dem Skalarprodukt basierende Darstellung der modernen Statik unter besonderer Betrachtung der Methode der finiten Elemente*. Dissertation, Fachgebiet Baustatik der Universität Kassel, 2002.
- [19] M. Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. Teubner-Verlag, Wiesbaden, 3. edition, 2009.
- [20] H. Harbrecht. *Numerische Mathematik*. Vorlesungsskript. Institut für Angewandte Analysis und Numerische Simulation der Universität Stuttgart, 2009.
- [21] A. Harten und J.M. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50(2):235 – 269, 1983.
- [22] R. Hartmann und P. Houston. Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comp.*, 24:979–1004, 2002.
- [23] J.S. Hesthaven und T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Publishing Company, Incorporated, 1st edition, 2007.

- [24] R. Höchstetter. Adaptive Discontinuous-Galerkin-Verfahren zur Lösung der Flachwassergleichungen mit verschiedenen Flussfunktionen. Diplomarbeit, Fakultät für Mathematik, Technische Universität München, 2009.
- [25] C. Johnson und J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comput.*, 46(173):1–26, January 1986.
- [26] G. Karniadakis und S.J. Sherwin. *Spectral/hp element methods for CFD*. Numerical mathematics and scientific computation. Oxford University Press, 1999.
- [27] P. Kaufmann, S. Martin, M. Botsch, und M. Gross. Flexible simulation of deformable models using discontinuous Galerkin fem. *Graph. Models*, 71(4), 2009.
- [28] D. Kuhl und G. Meschke. *Finite Elemente Methoden I & II*. Vorlesungsskript. Lehrstuhl für Statik und Dynamik der Ruhr-Universität Bochum, 2002.
- [29] B. Landmann. *A parallel discontinuous Galerkin code for the Navier-Stokes and Reynolds-averaged Navier-Stokes equations*. Dissertation, Fachbereich Luft- und Raumfahrttechnik der Universität Stuttgart, 2007.
- [30] C.B. Laney. *Computational gasdynamics*. Cambridge University Press, 1998. Cambridge Books Online.
- [31] C. Lübon. *Turbulenzmodellierung und Detached-Eddy-Simulationen mit einem Discontinuous-Galerkin-Verfahren hoher Ordnung*. Dissertation, Fakultät der Luft- und Raumfahrttechnik der Universität Stuttgart, 2011.
- [32] B. Van Leer, J.L. Thomas, P.L. Roe, und R.W. Newsome. A comparison of numerical flux formulas for the Euler and Navier-Stokes equations. volume 87-1104 of *AIAA PAPER*, 1987.
- [33] S.K. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16 – 42, 1992.
- [34] P. Lesaint und P.A. Raviart. *On a finite element method for solving the neutron transport equation*. Univ. Paris VI, Labo. Analyse Numérique, 1974.
- [35] B.Q. Li. *Discontinuous finite elements in fluid dynamics and heat transfer*. Computational Fluid and Solid Mechanics. Springer, 2006.
- [36] M. Möller. *Hochauflösende FEM-FCT-Verfahren zur Diskretisierung von konvektionsdominanten Transportproblemen mit Anwendung auf die kompressiblen Eulergleichungen*. Diplomarbeit, Fachbereich Mathematik der Universität Dortmund, 2003.

- [37] X. Nogueira, L. Cueto-Felgueroso, I. Colominas, H. Gmeza, F. Navarrina, und M. Casteleiro. On the accuracy of finite volume and discontinuous Galerkin discretizations for compressible flow on unstructured grids. *International Journal for Numerical Methods in Engineering*, 00:1–6, 2000.
- [38] W.H. Reed und T.R. Hill. *Triangular mesh methods for the neutron transport equation*. 1973.
- [39] E. Rose. *Implizite Runge-Kutta-Verfahren und ihre Anwendung auf Steuerungsprobleme*. Diplomarbeit, Fakultät der Mathematik und Physik der Universität Bayreuth, 2007.
- [40] C. Tam und J.C. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of Computational Physics*, 107(2):262 – 281, 1993.
- [41] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer, Berlin, 2nd edition, 1999.
- [42] Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.O. Persson, B. van Leer, und M. Visbal. High-order CFD methods: current status and perspective. *Int. J. Num. Meth. Fluids*, 72(8):811–845, 2013.
- [43] A. Wolkov, C. Hirsch, und B. Leonard. Discontinuous Galerkin method on unstructured hexahedral grids for 3d euler and navier-stokes equations. *AIAA Paper 2007-4078*, 2007.



Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.

Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: Bergmann

Vorname: Michael

dass ich die vorliegende Masterarbeit bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Erweiterung eines 3D Discontinuous Galerkin Strömungslösers

ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der Masterarbeit ist erfolgt durch:

Ort

Datum

Unterschrift im Original