

Newton-Krylov-Verfahren für dreidimensionale Über- und Hyperschallströmungen im thermochemischen Nichtgleichgewicht

Von der Fakultät Luft- und Raumfahrttechnik und Geodäsie
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Ferdinand Olawsky

aus Pforzheim

Hauptberichterin : Prof. Dr.-Ing. habil. Monika Auweter-Kurtz
Mitberichter : Prof. Dr.-Ing. habil. Dr. h.c. Wolfgang Wendland
Tag der mündlichen Prüfung : 17. Mai 2005

Institut für Raumfahrtsysteme

Universität Stuttgart

2005

Kurzfassung

Für die Simulation von dreidimensionalen Über- und Hyperschallströmungen um Raumfahrzeuge während des Wiedereintritts in die Erdatmosphäre wird in dieser Arbeit ein numerisches Verfahren vorgestellt. Die der Simulation zugrunde liegenden Navier-Stokes-Gleichungen bestehen aus fünf Massenerhaltungsgleichungen für die Spezies N_2 , O_2 , NO , N und O , drei Impulserhaltungsgleichungen, einer Gleichung für die Gesamtenergie, sowie drei Gleichungen für die Vibrationsenergien der molekularen Spezies N_2 , O_2 und NO .

Für die Diskretisierung der instationären, kompressiblen Navier-Stokes-Gleichungen wird ein zellzentriertes Finite Volumen Verfahren auf strukturierten Netzen eingesetzt. Die stationäre Lösung wird mit einem impliziten Newton-Verfahren mit approximierter Jacobi-Matrix berechnet. In jedem Zeitschritt des Newton-Verfahrens muss ein lineares Gleichungssystem gelöst werden. Hierzu wird in dieser Arbeit das Jacobi-Linienrelaxationsverfahren mit vorkonditionierten Krylov-Raum-Verfahren verglichen. Bei der Entwicklung der Vorkonditionierer wird besonderer Wert auf deren Vektorisierbarkeit gelegt. Um die Konvergenz zu beschleunigen, wird das Newton-Verfahren zu einem Newton-Krylov-Verfahren erweitert. Ausgehend von einem matrixfreien Verfahren werden weitere Verfahren entwickelt, um die Approximationen in der Jacobi-Matrix zu reduzieren und um den Einfluss der Approximationen auf die Konvergenz zu untersuchen.

Abstract

For the simulation of three-dimensional supersonic and hypersonic nonequilibrium flows around reentry vehicles a Navier-Stokes solver is developed. For an accurate modeling of thermochemical relaxation processes the Navier-Stokes equations consist of 5 continuity equations for N_2 , O_2 , NO , N and O , 3 momentum equations, the total energy equation and 3 vibrational energy equations for the molecular species N_2 , O_2 and NO .

The unsteady, compressible Navier-Stokes equations are discretized in space using cell-centered finite volume approach on structured grids. To calculate the steady-state solution of the Navier-Stokes equations an implicit Newton method with an approximated Jacobian is used. In each step of the Newton scheme a linear system has to be solved. Therefore, the Jacobi Line Relaxation method is compared with preconditioned Krylov subspace methods. Special attention is paid to the development of vectorizable preconditioners. In order to accelerate the convergence, the Newton scheme is enhanced to an exact Newton-Krylov-method. Starting from a matrix-free implicit method, several schemes are developed to reduce approximations in the Jacobian and to study the influence of the approximations on the convergence.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Angestellter am Institut für Raumfahrtsysteme, Fakultät Luft- und Raumfahrttechnik und Geodäsie der Universität Stuttgart, im Sonderforschungsbereich 259 *Hochtemperaturprobleme rückkehrfähiger Raumtransportsysteme* im Teilprojekt *Numerische Wiedereintrittsaerothermodynamik*.

Bei Prof. Dr.-Ing. habil. Monika Auweter-Kurtz bedanke ich mich ganz herzlich für die Übernahme des Hauptberichts und für die Unterstützung meines Promotionsvorhabens. Ihr stetes Interesse an meiner Arbeit war für mich bis zum Schluss eine wichtige Motivation. Prof. Dr.-Ing. habil. Dr. h.c. Wendland danke ich für die Übernahme des Mitberichts.

Bei meinen Kollegen in meiner Arbeitsgruppe bedanke ich mich für die gute Zusammenarbeit und die fachliche Unterstützung. Besonderer Dank gilt Herrn Markus Fertig für die vielen fachlichen Diskussionen und die Nachhilfestunden, in denen er mir unermüdlich die physikalischen Zusammenhänge erklärt hat. Herrn Farid Infed danke ich ganz herzlich für die kollegiale Zusammenarbeit, die fachlichen Diskussionen und auch für die privaten Gespräche, die mir sehr geholfen haben, die schwere Zeit während der Promotion zu überstehen.

Ganz herzlich bedanke ich mich bei meinen Arbeitskollegen am Institut für Raumfahrtsysteme und am Höchstleistungsrechenzentrum der Universität Stuttgart für die angenehme Arbeitsatmosphäre.

Ferdinand Olawsky

Inhaltsverzeichnis

Nomenklatur	8
1 Einleitung	10
2 Physikalische Strömungsmodellierung	14
2.1 Strömung um Wiedereintrittsfahrzeuge	14
2.2 Luftmodell	15
2.3 Strömungsmodell	16
3 Numerische Modellierung	22
3.1 Finite-Volumen-Diskretisierung	22
3.2 Reibungsfreie Flüsse	23
3.2.1 Roe-Verfahren	24
3.2.2 AUSMDV	27
3.2.3 Verfahren höherer Ordnung	33
3.3 Viskose Flüsse	36
3.4 Quellterme	37
3.5 Rand- und Anfangswerte	38
4 Implizites Lösungsverfahren	40
4.1 Zeitschritt-Newton-Verfahren	41
4.2 Inkonsistentes Newton-Verfahren	43
4.3 Lokaler Zeitschritt	44
4.4 Lösungsverfahren für lineare Gleichungssysteme	45
4.4.1 Iterative Lösungsverfahren	45
4.4.2 Jacobi-Linienrelaxationsverfahren	46
4.4.3 Krylov-Raum-Verfahren	51
4.4.4 Vorkonditionierung	56
4.4.5 Vektorisierbarkeit der Gleichungslöser	63
5 Konvergenzbeschleunigung	67
5.1 Inkonsistentes Newton-Verfahren	67
5.2 Konsistentes Newton-Verfahren	69
5.3 Matrix-Vektor-Multiplikationen	69
5.3.1 Matrixfreies Newton-Krylov-Verfahren	69
5.3.2 Finite Differenzen für Links- und Rechtszustände	71

5.3.3	Finite Differenzen für die Steigungen	72
5.3.4	Exakte Matrix-Vektor-Multiplikation	73
5.3.5	Exakte Matrix-Vektor-Multiplikation für die Jacobi-Matrix der viskosen Flüsse	74
6	Ergebnisse	76
6.1	Umströmung der MIRKA	76
6.2	Probenumströmung im Plasmawindkanal	80
6.3	MIRKA-Vorkörperströmung	83
6.4	Umströmung der Rückkehrkapsel COLIBRI	90
6.5	Umströmung der X-38	99
7	Zusammenfassung	100
A	Fluss-Jacobi-Matrizen	102
B	Pseudocodes der Gleichungslöser	104
	Literatur	108
	Abstact	116
	Lebenslauf	118

Nomenklatur

Physikalische Bezeichnungen

c	Schallgeschwindigkeit
c_v^{tr}	Spezifische Wärmekapazität für Rotation und Translation
c_v^r	Spezifische Wärmekapazität für Rotation
c_v^t	Spezifische Wärmekapazität für Translation
e	Innere Gesamtenergie
e_t	Gesamtenergie
e_{vib,N_2}	Vibrationsenergie von N_2
e_{vib,O_2}	Vibrationsenergie von O_2
$e_{vib,NO}$	Vibrationsenergie von NO
$\vec{E}, \vec{F}, \vec{G}$	Reibungsfreie Flüsse in x -, y - und z -Richtung
$\vec{E}_v, \vec{F}_v, \vec{G}_v$	Viskose Flüsse in x -, y - und z -Richtung
h_t	Totale Enthalpie
$h_{0,s}$	Bindungsenthalpie
p	Statischer Druck
P	Vektor der Primitivvariablen
Q	Erhaltungsvektor
$Q_{t-v,k}$	Austauschterm für Translation/Vibration
$Q_{v-v,k}$	Austauschterm für Vibration/Vibration
$Q_{c-v,k}$	Austauschterm für Chemie/Vibration
S	Quelltermvektor
t	Zeit
Δt	Diskreter Zeitschritt
T_t	Translationstemperatur
T_r	Rotationstemperatur
T_{tr}	Translations-/Rotationstemperatur
T_{vib,N_2}	Vibrationstemperatur von N_2
T_{vib,O_2}	Vibrationstemperatur von O_2
$T_{vib,NO}$	Vibrationstemperatur von NO
R_s	Spezifische Gaskonstante
x, y, z	Kartesisches Koordinatensystem
u, v, w	Geschwindigkeitskomponenten in x -, y - und z -Richtung
u_{\perp}	Normalgeschwindigkeit, $u_{\perp} = n_x u + n_y v + n_z w$
V	Volumenmatrix

α	Anstellwinkel
φ	Fluss
ρ	Gesamtdichte
ρ_s	Partialdichten
ψ_s	Molanteile
τ	Schubspannungstensor
ξ_s	Massenanteile
$\dot{\omega}$	Quellterm für chemische Reaktionen

Mathematische Symbole

CFL	Courant-Friedrichs-Lewy-Zahl
A	Matrix eines linearen Gleichungssystems
D	(Block-) Diagonalmatrix
E	Einheitsmatrix
L	Untere (<i>lower</i>) Dreiecksmatrix
U	Obere (<i>upper</i>) Dreiecksmatrix
N_i	Matrix der beiden Blocknebendiagonalen für die ξ -Richtung
N_j	Matrix der beiden Blocknebendiagonalen für die η -Richtung
N_k	Matrix der beiden Blocknebendiagonalen für die ζ -Richtung
P	Vorkonditionierungsmatrix
P_l	Matrix für linksseitige Vorkonditionierung
P_r	Matrix für rechtsseitige Vorkonditionierung
r_k	Residuum des linearen Gleichungssystems für die Näherungslösung x_k
x_k	Näherungslösung nach k Iterationen
K_m	m -dimensionaler Krylov-Raum
ξ	Koordinatenrichtung im Rechenraum
η	Koordinatenrichtung im Rechenraum
ζ	Koordinatenrichtung im Rechenraum
n_x	x -Komponente des Normalenvektors
n_y	y -Komponente des Normalenvektors
n_z	z -Komponente des Normalenvektors
ω	Relaxationsparameter

Indizes

l, r	Links-/Rechtszustand an einer Zellfläche
s, k	Gasphasenkomponente
tr	translatorisch-rotatorisch
v	viskos
vib	Vibration
i, j, k	Netzrichtung im strukturierten Netz
∞	Zuströmbedingung

Kapitel 1

Einleitung

Hintergrund und Umfeld der Arbeit

Neben den analytischen und den experimentellen Untersuchungen stellen numerische Simulationsverfahren ein drittes Glied in der Erforschung physikalischer, chemischer, mathematischer oder technischer Phänomene dar. Computersimulationen können z. B. teure Experimente ersetzen oder wichtige Vorarbeiten für experimentelle Untersuchungen leisten. Ebenso ist es mittels Computersimulationen möglich, Phänomene zu untersuchen, die aufgrund ihrer Komplexität oder aufgrund zu hoher Kosten weder experimentell noch analytisch erforscht werden können. Hierzu zählen zum Beispiel Simulationen zur Untersuchung langfristiger Klimaveränderungen, Simulationen der Bewegungen der Erdkruste oder Simulationen komplexer Strömungen. Eine umfangreiche Zusammenstellung wissenschaftlicher Simulationsverfahren aus unterschiedlichen Anwendungsgebieten ist in [48] zu finden.

Für ein effizientes Simulationsprogramm ist die Kombination von robuster Numerik mit modernsten Computern erforderlich. Die notwendige Rechenleistung für komplexe wissenschaftliche und industrielle Simulationsrechnungen kann heutzutage nur von Höchstleistungsrechnern erbracht werden. Durch den Einsatz spezieller Prozessoren (z. B. Vektorprozessoren) bzw. durch den Einsatz mehrerer Prozessoren (Parallelrechner) kann die Rechenzeit für aufwändige Simulationsverfahren deutlich reduziert werden. Durch die rasante Computerentwicklung der letzten Jahrzehnte können Computerprogramme immer größere Probleme zunehmend detaillierter simulieren. Parallel zur Computerentwicklung wurden auch auf dem Gebiet der Numerik in den letzten Jahrzehnten große Fortschritte erzielt, so dass mittlerweile für viele Bereiche der Ingenieurwissenschaften numerische Standardverfahren existieren. Da jedoch komplexe Anwendungsprobleme nur auf Höchstleistungsrechnern gelöst werden können, müssen schon bei der Entwicklung numerischer Verfahren die eingesetzten Computertypen berücksichtigt werden, da nicht jedes numerische Verfahren auf jedem Computer die gleiche Performance zeigt. Für den Einsatz auf einem Vektorrechner muss ein numerisches Verfahren speziell für Vektorprozessoren entwickelt, optimiert und implementiert werden. Ebenso müssen für einen Parallelrechner ggf. die numerischen Methoden angepasst oder erweitert werden, damit das Verfahren den Parallelrechner effizient nutzen kann.

Im Sonderforschungsbereich 259 *Hochtemperaturprobleme rückkehrfähiger Raumtransportsysteme* wurde am Institut für Raumfahrtsysteme der Universität Stuttgart im Teilprojekt C3 *Numerische Wiedereintrittsaerothermodynamik* das Simulationsprogramm **URANUS** (**Upwind Relaxation Algorithm for Nonequilibrium Flows of the University of Stuttgart**) entwickelt, welches die Nichtgleichgewichtsströmung um Wiedereintrittsfahrzeuge sowie deren thermische Belastung beim Wiedereintritt in die Erdatmosphäre berechnet. Für die Entwicklung von wiederverwendbaren Raumtransportsystemen ist die detaillierte Vorhersage der zu erwartenden thermischen Belastungen beim Wiedereintritt unerlässlich. Einerseits muss das Hitzeschutzsystem so ausgelegt werden, dass das Fahrzeug den thermischen Belastungen beim Wiedereintritt standhält, andererseits sollte das Hitzeschutzsystem nur eine geringe Masse haben. Um ein Hitzeschutzsystem optimal auslegen zu können, müssen schon beim Entwurf eines Raumfahrzeugs die thermischen Belastungen beim Wiedereintritt genau bekannt sein. Eine detaillierte Vorhersage der thermischen Belastung kann nur mit Hilfe eines numerischen Simulationsverfahrens erfolgen, welches sämtliche relevanten Realgas- und Katalyseeffekte sowie die komplexe Geometrie des Flugkörpers berücksichtigt.

Stand der Forschung

Für die Simulation von Stömungen um Wiedereintrittsfahrzeuge wurde beim Deutschen Zentrum für Luft- und Raumfahrt DLR das Simulationsverfahren CEVCATS entwickelt. Das zugrunde liegende punktimplizite Verfahren versucht mittels Mehrgittertechnik die Konvergenz zu beschleunigen, was jedoch für diese Art von Strömungen nicht problemlos gelingt ([72],[49],[71]). Die Probleme der Mehrgittertechnik bei reaktiven Hyperschallströmungen wurden sehr genau von P. Gerlinger analysiert ([23],[24]). Er simulierte reaktive Hyperschallverbrennungen und schaffte es, durch geschicktes Anpassen des Mehrgitterverfahrens diese Strömungen berechnen zu können. Der von der NASA eingesetzte LAURA-Code, der ebenso für die Berechnung von reaktiven Hyperschallströmungen entwickelt wurde, basiert ebenfalls auf einem punktimpliziten Verfahren.

Außerhalb des engen Gebiets der reaktiven Hyperschallströmungen gibt es viele numerische Verfahren, welche eine schnelle Berechnung von stationären Strömungslösungen versprechen. Neben den Mehrgitterverfahren werden in vielen Veröffentlichungen insbesondere die matrixfreien Newton-Krylov-Verfahren vorgestellt, welche durch geschickte Kombination des Newton-Verfahrens mit einem Krylov-Raum-Verfahren eine Vereinfachung und auch eine Beschleunigung des Verfahrens erreichen. In [68] ist eine umfangreiche Liste von Simulationsverfahren aus den Jahren 1985 bis 1997 zu finden, die sowohl Newton- als auch Krylov-Raum-Verfahren einsetzen. Leider ist in diesem Zusammenhang kein Simulationsverfahren für reaktive Hyperschallströmungen zu finden.

Die seit Anfang der neunziger Jahre aufkommenden Krylov-Raum-Verfahren werden immer häufiger auch in komplexeren Strömungssimulationsverfahren eingesetzt. Doch die dabei so entscheidene Frage nach einer geeigneten Vorkonditionierung kann nicht pauschal beantwortet werden. Für jeden Matrixtyp, für jede Art von Differentialgleichung und für jede zugrundeliegende Diskretisierung muss erneut nach einer geeigneten Vorkonditionierung gesucht werden. Auf dem Gebiet der numerischen Strömungsmechanik setzte sich vor allem die ILU-Vorkonditionierung

durch, [56]. Doch schon bei der Suche nach Vorkonditionierungstechniken für Vektorrechner sind nur wenige Methoden bekannt, [6]. Systematische Untersuchungen dazu sind in der Literatur nicht zu finden.

Vorarbeiten

Zu Beginn der vorliegenden Arbeit standen zwei Programmversionen des URANUS-Verfahrens zur Verfügung:

- ein dreidimensionales Idealgasverfahren und
- ein zweidimensionales/axialsymmetrisches Nichtgleichgewichtsverfahren.

Das dreidimensionale Idealgasverfahren wurde von Schöll für Schaufelreihenströmungen von Turbomaschinen entwickelt, [75]. Dieses Verfahren wurde von Gerlinger in seiner Diplomarbeit zu einem Simulationsverfahren für Idealgasströmungen um Wiedereintrittsfahrzeuge erweitert, [25].

Das zweidimensionale/axialsymmetrische Nichtgleichgewichtsverfahren stammt ursprünglich von Jonas [38] und wurde von Daiß, Knab und Kanne in der Gasphasen- und in der Oberflächenmodellierung, sowie durch die Modellierung der viskosen Flüsse ausgebaut ([14],[17], [40]). Dieses Verfahren wurde in den letzten Jahren innerhalb mehrerer Projekte erfolgreich eingesetzt. So wurde die gesamte numerische Nachflugauswertung bzgl. der Aerothermodynamik des MIRKA-Flugexperiments mit dem URANUS-Verfahren durchgeführt, [5]. Hierbei ergab sich eine sehr gute Übereinstimmung der Simulationsergebnisse mit den Flugmessdaten.

Zielsetzung der Arbeit

Um komplexere Fahrzeuggeometrien in Simulationen berücksichtigen zu können, soll in dieser Arbeit aufbauend auf den beiden existierenden URANUS-Programmversionen ein Simulationsverfahren für *dreidimensionale* Strömungen im thermochemischen Nichtgleichgewicht um Wiedereintrittsfahrzeuge aus dem niedrigen Erdorbit entwickelt werden. Hierbei soll die Strömungs- und Gasphasenmodellierung des zweidimensionalen/axialsymmetrischen Programms beibehalten und auf ein 5-Komponenten / 4-Temperaturen-Luftmodell angepasst werden. Da das Simulationsprogramm für Anwendungsrechnungen zur Verfügung stehen soll, muss ein robustes und effizientes numerisches Verfahren gewählt werden, das trotz komplexer Strömungen und steifer Quellterme die Berechnung der stationären Lösung ermöglicht. Daher soll das bereits im zweidimensionalen/axialsymmetrischen Programm eingesetzte Zeitschritt-Newton-Verfahren auch in das neu entwickelte Verfahren für dreidimensionale Strömungen übernommen werden.

Aufgrund des deutlich höheren Speicherplatz- und Rechenzeitbedarfs eines Verfahrens für dreidimensionale Strömungen im Vergleich zu einem Verfahren für zweidimensionale Strömungen soll in dieser Arbeit als Schwerpunkt die genaue Untersuchung des numerischen Verfahrens

gesetzt werden mit dem Ziel, den Speicherplatzbedarf und die Rechenzeit zu reduzieren. Hierzu gehört eine genaue Analyse der eingesetzten numerischen Methoden. Insbesondere soll das Jacobi-Linienrelaxationsverfahren zum Lösen der linearen Gleichungssysteme mit den in der Literatur weit verbreiteten Krylov-Raum-Verfahren mit ILU-Vorkonditionierung verglichen werden. Desweiteren soll eine Konvergenzbeschleunigung des Zeitschritt-Newton-Verfahren erreicht werden. Aufgrund des hohen Speicherplatz- und Rechenzeitbedarfs wird der Einsatz des Vektorrechners NEC SX-5 angestrebt. Daher wird großer Wert auf die Vektorisierbarkeit des Simulationsprogramms gelegt.

Gliederung

In den Kapiteln 2 und 3 werden das Luft- und das Strömungsmodell vorgestellt, sowie die räumliche Diskretisierung der inhomogenen Navier-Stokes-Gleichungen beschrieben.

In Kapitel 4 wird das Zeitschritt-Newton-Verfahren hergeleitet, sowie die praktikable Umsetzung dieses Verfahrens erläutert. Für die linearen Gleichungssysteme wird das eingesetzte Jacobi-Linienrelaxationsverfahren genauer untersucht und mit vorkonditionierten Krylov-Raum-Verfahren verglichen. Zusätzlich werden Methoden untersucht, die die Vektorisierbarkeit der Vorkonditionierungstechniken verbessern.

Kapitel 5 beschäftigt sich mit Untersuchungen zur Konvergenzbeschleunigung. Dazu wird das Newton-Verfahren mit den Krylov-Raum-Verfahren zu einem so genannten Newton-Krylov-Verfahren kombiniert.

In Kapitel 6 werden anhand unterschiedlicher Anwendungs- und Testfälle die verschiedenen numerischen Verfahren verglichen. Außerdem wird die Leistungsfähigkeit des Verfahrens demonstriert.

In Kapitel 7 erfolgt eine Zusammenfassung der Arbeit.

Kapitel 2

Physikalische Strömungsmodellierung

Die physikalischen Phänomene, die beim Wiedereintritt eines Raumfahrzeugs in die Erdatmosphäre auftreten, sind sehr komplex, so dass sie in einem Simulationsprogramm nicht vollständig berücksichtigt werden können. Es ist daher zwingend notwendig, sich bei der Modellierung auf die relevanten Phänomene zu beschränken. In diesem Kapitel werden ein Luftmodell und ein Strömungsmodell hergeleitet, welche für die genaue Berechnung des Oberflächenwärmestroms während des Wiedereintritts eines Raumfahrzeugs aus einem niedrigen Erdorbit ausreichend sind.

2.1 Strömung um Wiedereintrittsfahrzeuge

Beim Wiedereintritt aus einem niedrigen Erdorbit in die Erdatmosphäre hat ein Raumfahrzeug eine Anfangsgeschwindigkeit von ca. 7,9 km/s. Diese hohe Geschwindigkeit bestimmt zusammen mit der dünnen Atmosphäre die beim Wiedereintritt auftretenden aerothermodynamischen Phänomene.

Entlang der Wiedereintrittstrajektorie durchfliegt der Flugkörper unterschiedliche Strömungsbereiche, die mit der dimensionslosen Knudsenzahl charakterisiert und eingeteilt werden können. Die Knudsenzahl Kn ist als das Verhältnis der mittleren freien Weglänge $\bar{\lambda}$ und der charakteristischen Länge L des Wiedereintrittskörpers definiert,

$$Kn = \frac{\bar{\lambda}}{L}.$$

Die mittlere freie Weglänge bezeichnet die Strecke, die ein Molekül oder Atom im Mittel zwischen zwei Zusammenstößen mit anderen Teilchen zurücklegt.

In den äußeren Atmosphärenschichten oberhalb einer Höhe von 120 km ist aufgrund der geringen Dichte die mittlere freie Weglänge sehr groß. In diesem Bereich mit $Kn > 10$ wird die Strömung als freie Molekülströmung betrachtet, da zwischen den Molekülen so gut wie keine Stöße stattfinden. Mit abnehmender Flughöhe und zunehmender Dichte beeinflussen Stöße

zwischen den Teilchen die Strömung. Dieser Bereich mit $10^{-2} < Kn < 10$ wird als Übergangsbereich bezeichnet. Ist die mittlere freie Weglänge so klein, dass sie zur Beschreibung der makroskopischen Strömungsphänomene nicht mehr aufgelöst werden muss, so kann von einer Kontinuumsströmung ausgegangen werden. Dieser Bereich der Kontinuumsströmung wird dann angenommen, wenn die Knudsenzahl kleiner 10^{-2} ist, was typischerweise unterhalb einer Höhe von ungefähr 90 km gilt. Die Strömung wird dort aufgrund der hohen Gastemperaturen stark durch sogenannte Realgaseffekte beeinflusst, die im nächsten Abschnitt näher erläutert werden. Nach der heißen Phase des Wiedereintritts in einer Höhe von ca. 30 km treten aufgrund der geringen Fluggeschwindigkeit keine relevanten Realgaseffekte mehr auf, so dass die Strömung als reine Idealgasströmung betrachtet werden kann.

2.2 Luftmodell

Die vorliegende Arbeit beschäftigt sich mit der heißen Phase des Wiedereintritts aus einem niedrigen Erdorbit. Dieser Bereich liegt zwischen 30 km und 90 km Höhe, also in dem Bereich der durch Realgaseffekte beeinflussten Strömung, die als Kontinuumsströmung angesehen werden kann. Ziel des in dieser Arbeit vorgestellten Verfahrens ist die Berechnung des Oberflächenwärmestroms.

Aufgrund der hohen Geschwindigkeit entsteht vor dem Flugkörper ein senkrechter Verdichtungsstoß (Abb. 2.1). Über diesen Stoß erhöhen sich sprunghaft Druck, Dichte und Temperatur des zuströmenden Gases. Der Druck kann im Stoß auf das 500 bis 1000-fache ansteigen. Gleichzeitig wird die Strömungsgeschwindigkeit von Überschall nach Unterschall verzögert. Durch den Druckanstieg und der damit verbundenen Temperaturzunahme im Stoß entstehen die bereits erwähnten Realgaseffekte. Unter dem Begriff 'Realgaseffekte' werden Rotations- und Schwingungsanregung der Moleküle sowie die gesamten chemischen Prozesse wie Dissoziation, Ionisation und Rekombination zusammengefasst. Nach dem Stoß beginnt der thermochemische Relaxationsprozess, in dem sich aufgrund von Teilchenstößen die thermochemische Zusammensetzung einem Gleichgewichtszustand annähert. Aufgrund der niedrigen Dichte finden jedoch zu wenige Teilchenstöße statt, so dass das Gas nicht ins Gleichgewicht kommt. Daher muss die Strömung als Nichtgleichgewichtsströmung modelliert werden.

Das zuströmende Gasgemisch besteht im wesentlichen aus ca. 20% molekularem Sauerstoff O_2 und ca. 80% molekularem Stickstoff N_2 . Aufgrund der hohen Temperaturen im und nach dem Stoß beginnen die zuströmenden Moleküle zu dissoziieren, so dass auch atomarer Sauerstoff O und atomarer Stickstoff N in der Strömung auftreten. Durch Rekombinationsreaktionen oder durch Austauschreaktionen entsteht Stickoxid NO . Desweiteren werden die Gaskomponenten ionisiert, so dass zusätzlich N_2^+ , O_2^+ , NO^+ , N^+ und O^+ sowie freie Elektronen e^- in der Strömung auftreten. Für das vorgestellte Verfahren wird jedoch nur der Wiedereintritt aus einem niedrigen Erdorbit betrachtet. Hierbei haben die in der Strömung auftretenden Ionen so gut wie keine Auswirkung auf den Oberflächenwärmestrom. Somit kann im Luftmodell die Ionisation vernachlässigt werden. Das Luftmodell beinhaltet damit nur die fünf Komponenten N_2 , O_2 , NO , N und O .

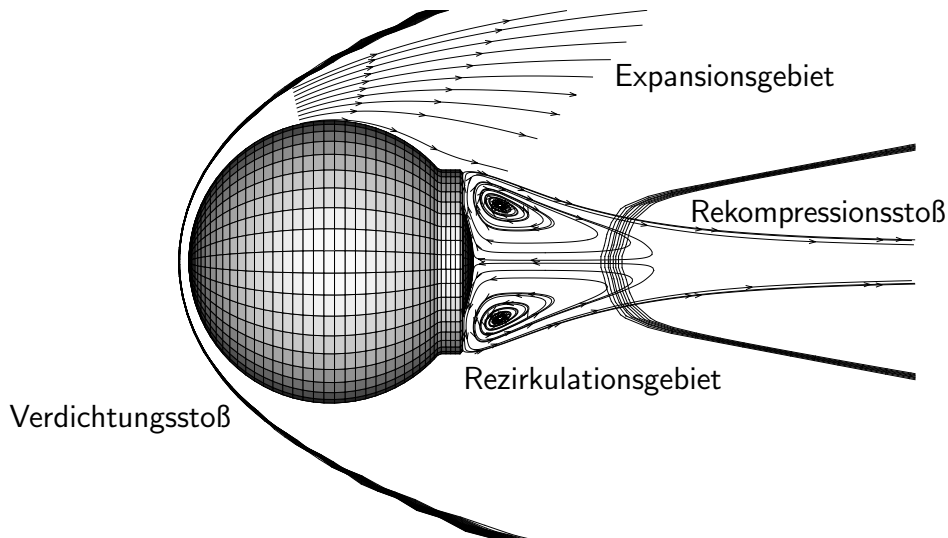


Abbildung 2.1: Strömung um MIRKA mit senkrechtem Verdichtungsstoß, Rezirkulations- und Expansionsgebieten sowie Rekompansionsstoß

Zusätzlich zu den chemischen Prozessen findet auch eine Anregung der inneren Freiheitsgrade der Moleküle statt. Die kinetische Energie der Strömung wird zu einem erheblichen Teil in Rotations- und Vibrationsenergien umgewandelt. Die Rotationsenergie gerät nach weniger als 100 Molekülstößen ins thermische Gleichgewicht mit der Translationsenergie. Daher wird im vorgestellten Verfahren die Rotationsenergie als im thermischen Gleichgewicht mit der Translationsenergie betrachtet. Dies bedeutet, dass die Rotationstemperatur T_r und die Translationsenergie T_t durch eine gemeinsame Rotations-/Translationstemperatur

$$T_{tr} = T_t = T_r \quad (2.1)$$

modelliert werden. Im Gegensatz dazu benötigen aber die Vibrationsenergien der Moleküle N_2 , O_2 und NO ungefähr 1000 Stöße, um ins Gleichgewicht mit der Translationsenergie zu kommen. Aufgrund der geringen Dichte finden nicht genügend Teilchenstöße statt, was zu einem Nichtgleichgewichtszustand führt, welcher bei den betrachteten Strömungen eine Rolle spielen kann. Daher werden die Vibrationsenergien als im thermischen Nichtgleichgewicht modelliert.

Zusammenfassend ergibt sich ein Luftmodell mit den 5 Komponenten N_2 , O_2 , NO , N und O und den 4 Temperaturen T_{tr} , T_{vib,N_2} , T_{vib,O_2} und $T_{vib,NO}$.

2.3 Strömungsmodell

Die heiße Phase des Wiedereintritts befindet sich in dem Bereich, in dem die Strömung als Kontinuumsströmung betrachtet werden kann. Daher wird die Strömung durch die als Navier-

Stokes-Gleichungen bekannten Erhaltungsgleichungen für Masse, Impuls und Energie modelliert, welche für die Erhaltungsgrößen

- Partialdichten: $\rho_{N_2}, \rho_{O_2}, \rho_{NO}, \rho_N, \rho_O$
- Impulse: $\rho u, \rho v, \rho w$
- Gesamtenergie: ρe_t
- Vibrationsenergien: $\rho_{N_2} e_{vib,N_2}, \rho_{O_2} e_{vib,O_2}, \rho_{NO} e_{vib,NO}$

aufgestellt werden müssen, wobei im Folgenden sämtliche Energien massenspezifisch angegeben werden. Hierbei ist ρ die Gesamtdichte

$$\rho = \sum_{s=1}^5 \rho_s \quad \text{mit} \quad \rho_1 = \rho_{N_2}, \quad \rho_2 = \rho_{O_2}, \quad \rho_3 = \rho_{NO}, \quad \rho_4 = \rho_N, \quad \rho_5 = \rho_O \quad (2.2)$$

und u, v, w sind die Geschwindigkeitskomponenten in x -, y - und z -Richtung des kartesischen Koordinatensystems. Der Index s wird im Folgenden zur Nummerierung der Spezies eingesetzt: $s = 1$ entspricht N_2 , $s = 2$ entspricht O_2 , $s = 3$ entspricht NO , $s = 4$ entspricht N , $s = 5$ entspricht O . Die Gesamtenergie e_t setzt sich aus der inneren Energie e und der kinetischen Energie zusammen

$$e_t = e + \frac{1}{2}(u^2 + v^2 + w^2). \quad (2.3)$$

Zur inneren Energie e gehören die Translationsenergien, die Rotationsenergien, die Bindungsenthalpien sowie die Vibrationsenergien

$$e = \sum_{s=1}^5 \xi_s c_{v,s}^t T_t + \sum_{s=1}^3 \xi_s c_{v,s}^r T_r + \sum_{s=1}^5 \xi_s h_{0,s} + \sum_{s=1}^3 \xi_s e_{vib,s}. \quad (2.4)$$

Hierbei gilt für den Massenbruch

$$\xi_s = \frac{\rho_s}{\rho} \quad (s = 1, \dots, 5). \quad (2.5)$$

Die Bindungsenthalpien werden mit $h_{0,s}$ bezeichnet. Da Translations- und Rotationsenergie im thermischen Gleichgewicht modelliert werden, können die spezifische Wärmekapazität der Translationsenergie $c_{v,s}^t$ und die spezifische Wärmekapazität der Rotationsenergie $c_{v,s}^r$ zusammengefasst werden zu einer gemeinsamen Wärmekapazität $c_{v,s}^{tr} = c_{v,s}^t + c_{v,s}^r$. Damit gilt für die innere Energie

$$e = \sum_{s=1}^5 \xi_s c_{v,s}^{tr} T_{tr} + \sum_{s=1}^5 \xi_s h_{0,s} + \sum_{s=1}^3 \xi_s e_{vib,s}. \quad (2.6)$$

Die Vibrationsenergien werden durch einen abgeschnittenen harmonischen Oszillator modelliert,

$$e_{vib,s} = \frac{R_s \Theta_s}{e^{\frac{\Theta_s}{T_{vib,s}}} - 1} - \frac{R_s D_s}{e^{\frac{D_s}{T_{vib,s}}} - 1} \quad (s = 1, \dots, 3). \quad (2.7)$$

Hierbei ist R_s die spezifische Gaskonstante, Θ_s die charakteristische Schwingungstemperatur und D_s die charakteristische Dissoziationstemperatur.

Die Erhaltungswerte werden zum Erhaltungsvektor zusammengefasst, welcher im Folgenden durch

$$Q = (\rho_{N_2}, \rho_{O_2}, \rho_{NO}, \rho_N, \rho_O, \rho u, \rho v, \rho w, \rho e_t, \rho_{N_2} e_{vib, N_2}, \rho_{O_2} e_{vib, O_2}, \rho_{NO} e_{vib, NO})^T \quad (2.8)$$

abgekürzt wird. Die Navier-Stokes-Gleichungen für diesen Erhaltungsvektor bestehen somit aus Erhaltungsgleichungen für

- 5 Massen,
- 3 Impulse,
- 1 Gesamtenergie,
- 3 Vibrationsenergien.

Im Folgenden werden die inhomogenen Navier-Stokes-Gleichungen aufgestellt.

Euler-Gleichungen

Die homogenen, instationären Euler-Gleichungen beschreiben eine reibungsfreie Stömung für eine Mischung idealer Gase. In einem kartesischen Koordinatensystem haben sie für eine dreidimensionale, kompressible Strömung die Form

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = 0. \quad (2.9)$$

Hierbei sind E , F und G die reibungsfreien Flüsse in x -, y - und z -Richtung:

$$E = \begin{pmatrix} \rho_s u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u h_t \\ \rho_{s'} u e_{vib, s'} \end{pmatrix} \quad F = \begin{pmatrix} \rho_s v \\ \rho v^2 + p \\ \rho vw \\ \rho vw \\ \rho v h_t \\ \rho_{s'} v e_{vib, s'} \end{pmatrix} \quad G = \begin{pmatrix} \rho_s w \\ \rho vw \\ \rho vw \\ \rho w^2 + p \\ \rho w h_t \\ \rho_{s'} w e_{vib, s'} \end{pmatrix} \quad (s = 1, \dots, 5; s' = 1, \dots, 3). \quad (2.10)$$

Für die totale Enthalpie h_t gilt

$$h_t = e_t + \frac{p}{\rho}. \quad (2.11)$$

Hierbei ist p der statische Druck. Das System von 12 Euler-Gleichungen beinhaltet nun 13 Unbekannte, da als zusätzliche Größe der Druck p eingeführt wurde. Zur Schließung des Systems ist daher eine weitere Gleichung notwendig. Diese Gleichung ist die kalorische Zustandsgleichung für eine Mischung idealer Gase

$$p = \sum_{s=1}^5 \rho_s R_s T_{tr}. \quad (2.12)$$

Navier-Stokes-Gleichungen

Durch Hinzunahme der viskosen Flüsse E_v , F_v und G_v entstehen aus den Euler-Gleichungen (2.9) die Navier-Stokes-Gleichungen.

$$\frac{\partial Q}{\partial t} + \frac{\partial(E - E_v)}{\partial x} + \frac{\partial(F - F_v)}{\partial y} + \frac{\partial(G - G_v)}{\partial z} = 0 \quad (2.13)$$

Für die Mehrkomponentenmodellierung werden die viskosen Flüsse durch Massendiffusion aufgrund von Konzentrations- und Druckgradienten, als auch durch einen zusätzlichen Energie-transport aufgrund der Massendiffusion erweitert.

$$E_v = \begin{pmatrix} \rho D_s \psi_{s,x} \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ q_x + u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \\ \lambda_{vib,k} T_{vib,k,x} + h_{t,k} \rho D_k \psi_{k,x} \end{pmatrix} \quad F_v = \begin{pmatrix} \rho D_s \psi_{s,y} \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ q_y + u\tau_{yx} + v\tau_{yy} + w\tau_{yz} \\ \lambda_{vib,k} T_{vib,k,y} + h_{t,k} \rho D_k \psi_{k,y} \end{pmatrix} \quad (2.14)$$

$$G_v = \begin{pmatrix} \rho D_s \psi_{s,z} \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ q_z + u\tau_{zx} + v\tau_{zy} + w\tau_{zz} \\ \lambda_{vib,k} T_{vib,k,z} + h_{t,k} \rho D_k \psi_{k,z} \end{pmatrix}.$$

Für die Massendiffusionsterme in den Massenerhaltungsgleichungen werden die Diffusionskoeffizienten D_s benötigt. Der Massendiffusionsstrom

$$\begin{pmatrix} j'_{s,x} \\ j'_{s,y} \\ j'_{s,z} \end{pmatrix} = \rho D_s \nabla \psi_s$$

muss zusätzlich korrigiert werden, damit die Summe der Flüsse gleich Null ist:

$$\vec{j}_s = \vec{j}'_s - \frac{\rho_s}{\rho} \sum_{i=1}^5 \vec{j}'_i. \quad (2.15)$$

Der Wärmestrom mit Wärmeübertragung durch Massendiffusion lautet

$$\begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = \lambda_{tr} \nabla T_{tr} + \sum_{s=1}^3 \lambda_{vib,s} \nabla T_{vib,s} + \sum_{s=1}^5 \rho h_{t,s} D_s \nabla \psi_s. \quad (2.16)$$

Mit $h_{t,s}$ wird die massenspezifische Enthalpie bezeichnet. Die translatorisch/rotatorische Wärmeleitfähigkeit wird mit λ_{tr} , die vibratorischen Wärmeleitfähigkeiten werden mit $\lambda_{vib,s}$ bezeichnet. Der Schubspannungstensor lautet bei Annahme der Stokes-Hypothese

$$\begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} = \mu \cdot \begin{pmatrix} 2\frac{\partial u}{\partial x} - \frac{2}{3}\nabla^T \cdot \vec{v} & \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} & 2\frac{\partial v}{\partial y} - \frac{2}{3}\nabla^T \cdot \vec{v} & \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} & \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} & 2\frac{\partial w}{\partial z} - \frac{2}{3}\nabla^T \cdot \vec{v} \end{pmatrix}. \quad (2.17)$$

Hierbei ist μ die Viskosität.

Navier-Stokes-Gleichungen mit Quelltermen

Für die Modellierung der thermochemischen Nichtgleichgewichtsströmung werden die inhomogenen Navier-Stokes-Gleichungen betrachtet.

$$\frac{\partial Q}{\partial t} + \frac{\partial(E - E_v)}{\partial x} + \frac{\partial(F - F_v)}{\partial y} + \frac{\partial(G - G_v)}{\partial z} = S \quad . \quad (2.18)$$

Hierbei bezeichnet S die chemischen und energetischen Quellterme

$$S = \begin{pmatrix} \dot{\omega}_s \\ 0 \\ 0 \\ 0 \\ 0 \\ Q_{t-v,k} + Q_{v-v,k} + Q_{c-v,k} \end{pmatrix} . \quad (2.19)$$

Die chemischen Quellterme $\dot{\omega}_s$ in den Massenerhaltungsgleichungen modellieren die chemischen Reaktionen in der Gasphase. $Q_{t-v,k}$, $Q_{v-v,k}$ und $Q_{c-v,k}$ sind die energetische Quellterme. Die drei Impulserhaltungsgleichungen sowie die Gesamtenergieerhaltungsgleichung haben keine Quellterme.

Klassifizierung der inhomogenen Navier-Stokes-Gleichungen

Gemäß ihrer physikalischen bzw. mathematischen Eigenschaften werden partielle Differentialgleichungen in die drei Typen *elliptisch*, *parabolisch* und *hyperbolisch* eingeteilt ([4], [19]). Die Auswahl eines numerischen Verfahrens zur Lösung der Gleichungen wird vom Typ der Differentialgleichung bestimmt.

Die stationären, inhomogenen Euler-Gleichungen

$$\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = S, \quad (2.20)$$

sind in Überschallgebieten hyperbolisch, in Unterschallgebieten jedoch elliptisch. Treten in einer Strömung sowohl Überschall- als auch Unterschallgebiete auf, so müsste ein numerisches Lösungsverfahren für diese Gleichungen ein hyperbolisch-elliptisches Problem lösen. Dazu gibt es bis heute noch keine stabilen numerischen Verfahren. Aus diesem Grund kann die Lösung der stationären Euler-Gleichungen nicht direkt berechnet werden.

Die *instationären* Euler-Gleichungen

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = S \quad (2.21)$$

sind sowohl in Überschall- als auch in Unterschallgebieten hyperbolisch in der Zeit. Die Bezeichnung 'hyperbolisch in der Zeit' soll verdeutlichen, dass die Evolutionsrichtung die Zeitachse ist.

Aus dem gemischten hyperbolisch-elliptischen Problem (2.20) wird nun ein rein hyperbolisches (2.21), welches numerisch wesentlich leichter zu behandeln ist.

Die instationären Navier-Stokes-Gleichungen (2.18) sind in Überschallgebieten hyperbolisch, in Unterschallgebieten parabolisch. Die parabolischen Gebiete können jedoch mit denselben numerischen Verfahren berechnet werden wie die hyperbolischen Gebiete, so dass hier keine Probleme bei der numerischen Berechnung entstehen.

Die Berechnung der *stationären* Lösung der Navier-Stokes-Gleichungen kann mit den *instationären* Navier-Stokes-Gleichungen erfolgen, falls die stationäre Lösung *asymptotisch stabil* ist. Beginnend von einer physikalisch sinnvollen Anfangsströmung wird eine instationäre Lösung so lange berechnet, bis sie gegen die stationäre Lösung konvergiert ist. Die Existenz einer asymptotisch stabilen stationären Lösung kann jedoch nicht garantiert werden, wird aber in dieser Arbeit stets vorausgesetzt.

Kapitel 3

Numerische Modellierung

Nachdem im Kapitel 2 die Navier-Stokes-Gleichungen aufgestellt wurden, befasst sich dieses Kapitel mit der räumlichen Diskretisierung dieser Gleichungen. Bei der räumlichen Diskretisierung ist darauf zu achten, dass die physikalischen Eigenschaften der Strömung erhalten bleiben und keine unphysikalischen Seiteneffekte entstehen.

3.1 Finite-Volumen-Diskretisierung

Für die Diskretisierung der Navier-Stokes-Gleichungen wird eine zellzentrierte Finite-Volumen-Diskretisierung gewählt. Die Navier-Stokes-Gleichungen lauten in Finite-Volumen-Formulierung für ein Volumenelement V mit Rand ∂V und Randnormalenvektor $\vec{n} = (n_x, n_y, n_z)$

$$\int_V \frac{\partial Q}{\partial t} dV + \oint_{\partial V} [(E - E_v)n_x + (F - F_v)n_y + (G - G_v)n_z] dA = \int_V S dV. \quad (3.1)$$

Für die räumliche Diskretisierung werden strukturierte Netze eingesetzt, um einerseits die Strömung in der Grenzschicht und den Oberflächenwärmestrom genau berechnen zu können, andererseits aber auch um auf einem Vektorrechner eine hohe Rechenleistung zu erzielen. Für eine Körperumströmung entsteht bei einem strukturierten Einblocknetz eine metrische Singularität, wie sie in Abbildung 3.2 zu sehen ist.

Angewandt auf eine hexaederförmige Zelle (Abb. 3.1) in einem strukturierten Netz mit dem Volumen $V_{i,j,k}$ und den Flächeninhalten A_1, \dots, A_6 lautet Gleichung (3.1)

$$V_{i,j,k} \cdot \frac{\partial Q_{i,j,k}}{\partial t} + \sum_{l=1}^6 A_l \varphi_l(Q) - \sum_{l=1}^6 A_l \varphi_{v,l}(Q) = V_{i,j,k} \cdot s(Q_{i,j,k}). \quad (3.2)$$

Hierbei hängen die reibungsfreien Flüsse φ_l und die viskosen Flüsse $\varphi_{v,l}$ von der lokalen Zelle und von einigen Zellnachbarn ab, die Quellterme s sind dagegen nur Funktionen des lokalen Erhaltungsvektors $Q_{i,j,k}$. Durch die räumliche Diskretisierung reduziert sich das System der Navier-Stokes-Gleichungen auf ein gewöhnliches Differentialgleichungssystem.

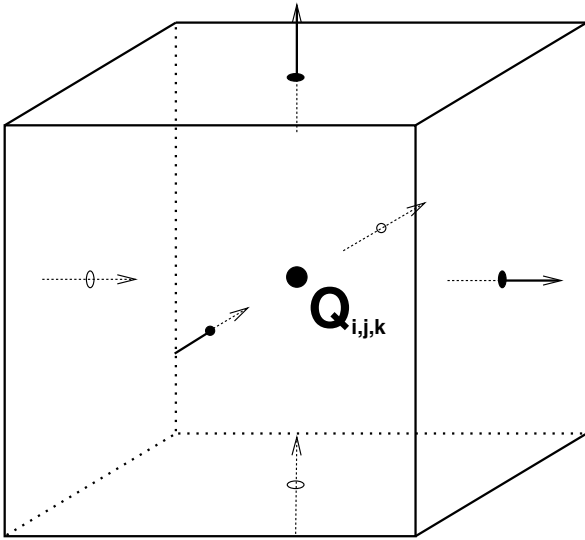
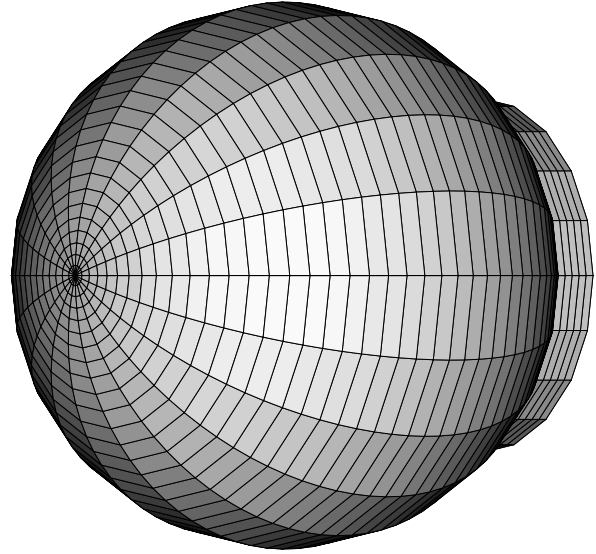


Abbildung 3.1: Hexaederförmige Zelle eines strukturierten Netzes


 Abbildung 3.2: Oberflächennetz der Wieder-
eintrittskapsel MIRKA mit metrischer Singu-
larität

3.2 Reibungsfreie Flüsse

In den Euler-Gleichungen spielt bei Über- und Hyperschallströmungen die gerichtete Informationsausbreitung entlang von Charakteristiken eine entscheidende Rolle. Daher wird für die Berechnung der reibungsfreien Flüsse der Godunov-Ansatz verwendet [82]. Dieser geht davon aus, dass innerhalb einer Zelle die Erhaltungsgrößen stetig (z. B. konstant) sind. Unstetigkeiten wie zum Beispiel ein Stoß können nur an den Zellrändern auftreten. Durch diesen Ansatz ergeben sich an den Zellrändern unstetige Übergänge. Diese Unstetigkeiten an den Rändern werden als eindimensionale Riemann-Probleme

$$\begin{cases} \frac{\partial Q}{\partial t} + \frac{\partial \varphi}{\partial x} = 0 \\ Q(x) = Q^l & \text{für } x < 0 \\ Q(x) = Q^r & \text{für } x > 0 \end{cases} \quad (3.3)$$

interpretiert. Hierbei sind Q^l und Q^r die Links- bzw. Rechtszustände der Erhaltungsgrößen an der Unstetigkeit. Der Fluss über eine Zellfläche wird als Lösung des lokalen Riemann-Problems betrachtet. In Gleichung (3.3) ist φ der eindimensionale Fluss

$$\varphi = \begin{pmatrix} \rho_s u_{\perp} \\ \rho u_{\perp} u + p n_x \\ \rho u_{\perp} v + p n_y \\ \rho u_{\perp} w + p n_z \\ \rho u_{\perp} h_t \\ u_{\perp} \rho_{s'} e_{vib,s'} \end{pmatrix} \quad s = 1, \dots, 5 \quad s' = 1, \dots, 3 \quad (3.4)$$

mit der Normalgeschwindigkeit $u_{\perp} = n_x u + n_y v + n_z w$. Die exakte Berechnung der Lösung des Riemann-Problems (3.3) ist für die Strömungssimulation zu aufwändig. Daher werden approximative Lösungsverfahren zur Berechnung der reibungsfreien Flüsse eingesetzt. Das bekannteste Verfahren hierfür ist das Roe-Verfahren, welches sehr genau und dissipationsarm ist. Als weiteres Verfahren wird in dieser Arbeit das AUSMDV-Verfahren eingesetzt.

3.2.1 Roe-Verfahren

Im Roe-Verfahren wird das nichtlineare Riemann-Problem

$$\frac{\partial Q}{\partial t} + \frac{\partial \varphi}{\partial x} = 0 \quad (3.5)$$

durch ein lineares Problem mit konstanten Koeffizienten

$$\frac{\partial Q}{\partial t} + A \frac{\partial Q}{\partial x} = 0 \quad (3.6)$$

approximiert und anschließend exakt gelöst. Von der Matrix A werden folgenden Eigenschaften verlangt:

- A ist hyperbolisch, d. h. A ist diagonalisierbar und hat reelle Eigenwerte
- $A = \frac{\partial \varphi}{\partial Q}$ für $Q^r = Q^l$
- $\varphi(Q^r) - \varphi(Q^l) = A(Q^r - Q^l)$.

Das Roe-Verfahren ist ein allgemeines Verfahren für beliebige hyperbolische Erhaltungsgleichungen. Allerdings ist es nicht möglich, für jedes System von Erhaltungsgleichungen eine Matrix A zu finden, die den obigen Bedingungen genügt. Für die Euler-Gleichungen kann jedoch eine Matrix A bestimmt werden [82]. Diese ergibt sich aus der Jacobi-Matrix $\frac{\partial \varphi}{\partial Q}$, ausgewertet an den sogenannten Roe-Mittelwerten.

Die exakte Jacobi-Matrix $\frac{\partial \varphi}{\partial Q}$ ist in Anhang A wiedergegeben. Die Roe-Mittelwerte werden nach Abgrall berechnet [1]:

$$\tilde{\rho} = \sqrt{\rho_r \rho_l} \quad (3.7)$$

$$\tilde{\xi}_s = \frac{\sqrt{\rho_r} \xi_{s,r} + \sqrt{\rho_l} \xi_{s,l}}{\sqrt{\rho_r} + \sqrt{\rho_l}} \quad (3.8)$$

$$\tilde{u}_s = \frac{\sqrt{\rho_r} u_r + \sqrt{\rho_l} u_l}{\sqrt{\rho_r} + \sqrt{\rho_l}} \quad (3.9)$$

$$\tilde{v}_s = \frac{\sqrt{\rho_r} v_r + \sqrt{\rho_l} v_l}{\sqrt{\rho_r} + \sqrt{\rho_l}} \quad (3.10)$$

$$\tilde{w}_s = \frac{\sqrt{\rho_r} w_r + \sqrt{\rho_l} w_l}{\sqrt{\rho_r} + \sqrt{\rho_l}} \quad (3.11)$$

$$(3.12)$$

$$\tilde{h}_t = \frac{\sqrt{\rho_r} h_{t,r} + \sqrt{\rho_l} h_{t,l}}{\sqrt{\rho_r} + \sqrt{\rho_l}} \quad (3.13)$$

$$\tilde{\xi}_s e_{vib,s} = \frac{\sqrt{\rho_r} \xi_{s,r} e_{vib,s,r} + \sqrt{\rho_l} \xi_{s,l} e_{vib,s,l}}{\sqrt{\rho_r} + \sqrt{\rho_l}} \quad (3.14)$$

Die mit diesen Mittelwerten berechnete Jacobi-Matrix erfüllt obige Bedingungen der Roe-Matrix. Für die Berechnung der exakten Lösung des linearen Riemann-Problems (3.6) werden die Zerlegung der Jacobi-Matrix in

$$\frac{\partial \varphi}{\partial Q} = R \Lambda R^{-1} \quad (3.15)$$

benötigt. Die Matrix Λ ist eine Diagonalmatrix mit den Eigenwerten der Jacobi-Matrix. Die Eigenwerte sind die Geschwindigkeiten der Charakteristiken

$$\Lambda = \text{diag}\{u_{\perp} - c, u_{\perp}, \dots, u_{\perp}, u_{\perp} + c\}. \quad (3.16)$$

Hierbei ist c die Schallgeschwindigkeit

$$c^2 = \sum_{s=1}^5 \xi_s \beta_{2,s} + \beta_1 \left(h_t - (u^2 + v^2 + w^2) - \sum_{s'=1}^3 \xi_{s'} e_{vib,s'} \right) \quad (3.17)$$

mit

$$\beta_1 = \frac{\sum_{s=1}^5 \rho_s R_s}{\sum_{s=1}^5 c_{v,s}^{tr} \rho_s} \quad (3.18)$$

$$\beta_{2,s} = \beta_1 \left(\frac{u^2 + v^2 + w^2}{2} - c_{v,s}^{tr} T - h_{0,s} \right) + R_s T. \quad (3.19)$$

Die Matrix R besteht aus den Eigenvektoren der Jacobi-Matrix. Die Matrizen R und R^{-1} sind im Anhang A wiedergegeben.

Die exakte Lösung des linearen Riemann-Problems (3.6) lautet

$$\varphi_{Roe} = \frac{1}{2} \left(\varphi(Q_r) + \varphi(Q_l) + \tilde{R} |\tilde{\Lambda}| \tilde{R}^{-1} (Q_r - Q_l) \right). \quad (3.20)$$

Hierbei ist $|\tilde{\Lambda}| = \text{diag}\{|u_{\perp} - c|, |u_{\perp}|, \dots, |u_{\perp}|, |u_{\perp} + c|\}$, ausgewertet an den Roe-Mittelwerten. Die Matrizen \tilde{R} sowie \tilde{R}^{-1} sind die Matrizen R und R^{-1} ausgewertet an den Roe-Mittelwerten. Eine genaue Herleitung der Lösung des linearen Riemann-Problems ist in [82] beschrieben.

Um die aufwändigen 12×12 -Matrizen R und R^{-1} nicht vollständig berechnen zu müssen, wird das Roe-Verfahren umgeschrieben

$$\tilde{R} |\tilde{\Lambda}| \tilde{R}^{-1} = \tilde{R} \left(|u_{\perp}| \cdot E - |\hat{\Lambda}| \right) \tilde{R}^{-1} = |u_{\perp}| \cdot E - \tilde{R} |\hat{\Lambda}| \tilde{R}^{-1} \quad (3.21)$$

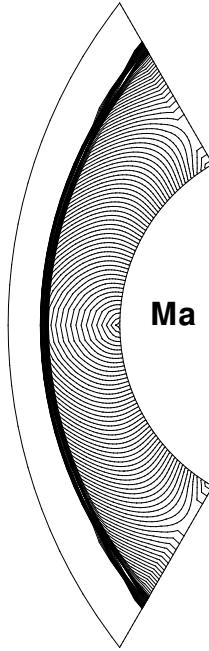


Abbildung 3.3: Reibungsfreie Zylinderumströmung, Ma=20, Roe-Verfahren mit Entropie-Korrektur

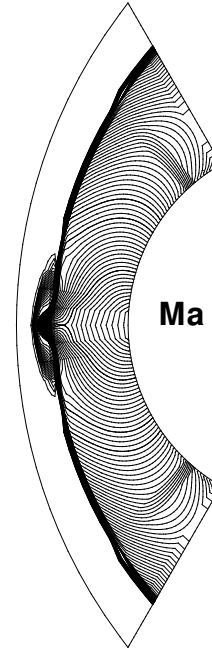


Abbildung 3.4: Reibungsfreie Zylinderumströmung, Ma=20, Roe-Verfahren mit Verdünnungsstößen und Carbuncle-Phänomen wenige Iterationen nach Ausschalten der Entropie-Korrektur

mit der Einheitsmatrix E und der Diagonalmatrix $|\hat{\Lambda}| = \text{diag}\{|u_{\perp} - c| - |u_{\perp}|, 0, \dots, 0, |u_{\perp} + c| - |u_{\perp}|\}$. Da die Matrix $|\hat{\Lambda}|$ nur zwei Einträge ungleich Null hat, müssen von der Matrix \tilde{R} nur die erste und letzte Spalte, von der Matrix \tilde{R}^{-1} nur die erste und letzte Zeile berechnet werden. Damit reduziert sich der Rechenaufwand für die Matrizen \tilde{R} und \tilde{R}^{-1} .

Der hergeleitete approximative Riemann-Löser ist in der bisherigen Form nicht anwendbar, da er unphysikalische Verdünnungsstöße entstehen lässt. Daher ist es erforderlich, eine Entropie-Korrektur vorzunehmen. Hierzu werden die Eigenwerte der Fluss-Jacobi-Matrix korrigiert. Die Eigenwerte λ werden ersetzt durch

$$\tilde{\lambda} = \begin{cases} \lambda & \text{für } \lambda > \mu \\ \frac{\lambda^2}{2\mu} + \frac{\mu}{2} & \text{sonst.} \end{cases} \quad (3.22)$$

Für die Eigenwerte $\lambda_1 = u_{\perp} + c$ und $\lambda_{12} = u_{\perp} - c$ wird der Parameter μ bestimmt über

$$\mu = \epsilon \cdot (|\tilde{u}_{\perp}| + \tilde{c}). \quad (3.23)$$

Dadurch werden Verdünnungsstöße erfolgreich vermieden.

Bei der Berechnung von Verdichtungsstößen leidet das Roe-Verfahren unter dem Carbuncle-Phänomen, wie es in Abbildung 3.4 zu sehen ist. Der senkrechte Verdichtungsstoß wird nicht korrekt berechnet. Daher wird in diesem Verfahren beim Roe-Verfahren zusätzlich zu den beiden

Eigenwerten $\lambda_1 = u_{\perp} + c$ und $\lambda_{12} = u_{\perp} - c$ auch die Eigenwerte λ_i ($i=2,\dots,11$) korrigiert. Dazu wird μ gesetzt auf

$$\mu = \epsilon \cdot (|\tilde{u}_{\parallel}| + \tilde{c}). \quad (3.24)$$

Aufgrund numerischer Tests wird der freie Parameter ϵ für alle Eigenwerte konstant auf 0,35 gesetzt. Mit diesem Parameterwert können Verdünnungstöße und das Carbuncle-Phänomen erfolgreich verhindert werden.

In Abbildung 3.3 ist eine Simulation einer Zylinder-Vorkörperströmung zu sehen. Berechnet wurde eine reibungsfreie Idealgasströmung mit dem Roe-Verfahren einschließlich Entropie-Korrektur. Die Zuströmdaten entsprechen den Daten der Wiederintrittskapsel MIRKA zum Zeitpunkt der maximalen Wärmelast. Die Zuströmdaten sind in Abschnitt 6.1 aufgelistet. In Abbildung 3.4 ist die Strömung wenige Zeitschritte nach Ausschalten der Entropie-Korrektur zu sehen. Es bilden sich zwei Verdünnungstöße aus, gleichzeitig entsteht das Carbuncle-Phänomen.

Für die Berechnung von dreidimensionalen Hyperschallströmungen hat das in Abschnitt 3.2.1 beschriebene Roe-Verfahren nicht die nötige Robustheit. Insbesondere bei den in Kapitel 5 erläuterten Konvergenzuntersuchungen stellte sich heraus, dass das Roe-Verfahren aufgrund seiner Instabilität die schnelle Konvergenz zur stationären Lösung sehr behindert. Aus diesem Grund musste im Rahmen dieser Arbeit nach einer robusteren Alternative gesucht werden.

3.2.2 AUSMDV

In [94] stellen Wada und Liou einen AUSMDV-Riemann-Löser mit einer Stoßstabilisierung und einer Entropie-Korrektur vor, welcher in den letzten Jahren immer häufiger in numerischen Verfahren für Über- und Hyperschallströmungssimulationen eingesetzt wurde ([15], [56],[41]). In dieser Arbeit wird das AUSMDV-Verfahren für das eingesetzte Luftmodell erweitert.

Herleitung des AUSMDV-Verfahrens

Ausgangspunkt des AUSDMV-Riemann-Lösers ist eine Variante des von Liou und Steffen entwickelten AUSM-Verfahrens (**A**dvection **U**pstream **S**plitting **M**ethod). Der numerische Fluss wird hierbei über

$$\varphi_{1/2} = \frac{1}{2} [(\rho u_{\perp})_{1/2}(\Psi_r + \Psi_l) - |(\rho u_{\perp})_{1/2}|(\Psi_r - \Psi_l)] + p_{1/2} \quad (3.25)$$

berechnet mit

$$\Psi = (\xi_s, u, v, w, h_t, e_{vib,s})^T \quad s = 1, \dots, 5 \quad s' = 1, \dots, 3.$$

Die Besonderheit dieser Variante des AUSM-Verfahrens ist die Flussaufteilung mit Hilfe des Massenflusses $(\rho u_{\perp})_{1/2}$ und nicht wie sonst üblich mit Hilfe der Geschwindigkeit u_{\perp} (u -splitting) oder nach der Machzahl (M -splitting). Wada und Liou bestimmen den Massenfluss über

$$(\rho u_{\perp})_{1/2} = u_l^+ \rho_l + u_r^- \rho_r$$

mit

$$u_l^+ = \alpha_l \left\{ \frac{(u_{\perp,l} + c_m)^2}{4c_m} \right\}_l + (1 - \alpha_l) \frac{u_{\perp,l} + |u_{\perp,l}|}{2} \quad (3.26)$$

$$u_r^- = \alpha_r \left\{ -\frac{(u_{\perp,r} - c_m)^2}{4c_m} \right\}_r + (1 - \alpha_r) \frac{u_{\perp,r} + |u_{\perp,r}|}{2}. \quad (3.27)$$

Durch die Parameterwahl

$$\alpha_l = \frac{2p_l}{p_l + p_r}, \quad \alpha_r = \frac{2p_r}{p_l + p_r}, \quad \text{und} \quad c_m = \max(c_l, c_r) \quad (3.28)$$

wird erreicht, dass die im AUSM-Verfahren enthaltene Dissipation an stationären und bewegten Kontaktunstetigkeiten verschwindet. Der Druckterm

$$p_{1/2} = p_l^+ + p_r^- \quad (3.29)$$

wird berechnet über

$$p_l^+ = \begin{cases} p_l \frac{(u_{\perp,l} + c_m)^2}{4c_m^2} \left(2 - \frac{u_{\perp,l}}{c_m} \right) & \text{wenn } |u_{\perp,l}| \leq c_m, \\ p_l \frac{u_{\perp,l} + |u_{\perp,l}|}{2u_{\perp,l}} & \text{sonst,} \end{cases} \quad (3.30)$$

$$p_r^- = \begin{cases} p_r \frac{(u_{\perp,r} + c_m)^2}{4c_m^2} \left(2 - \frac{u_{\perp,r}}{c_m} \right) & \text{wenn } |u_{\perp,r}| \leq c_m, \\ p_r \frac{u_{\perp,r} + |u_{\perp,r}|}{2u_{\perp,r}} & \text{sonst.} \end{cases} \quad (3.31)$$

Dieses Verfahren wird als AUSMD-Verfahren bezeichnet und hat seine Stärke in der genauen Berechnung von Kontaktunstetigkeiten. An Verdichtungsstößen erzeugt jedoch das Verfahren Oszillationen. Diese Oszillationen treten beim AUSMV-Verfahren nicht auf, welches den Impulsfluss durch

$$(\rho \vec{u} u_{\perp})_{\text{AUSMV}} = u_l^+ (\rho \vec{u})_l + u_r^+ (\rho \vec{u})_r \quad (3.32)$$

berechnet. Durch Kombination des AUSMD-Impulsflusses mit dem AUSMV-Impulsfluss entsteht das AUSMDV-Verfahren

$$(\rho \vec{u} u_{\perp})_{\text{AUSMDV}} = \frac{1}{2}(1 + s)(\rho \vec{u} u_{\perp})_{\text{AUSMV}} + \frac{1}{2}(1 - s)(\rho \vec{u} u_{\perp})_{\text{AUSMD}}. \quad (3.33)$$

Der Parameter s wird hierbei über den Druckgradienten ermittelt

$$s = \min \left(1, K \frac{|p_r - p_l|}{\min(p_l, p_r)} \right), \quad (3.34)$$

wobei für K der von Wada und Liou in [94] vorgeschlagene Wert $K = 10$ gewählt wird. Durch den Schaltparameter s wird in Stoßumgebung dem AUSMDV-Riemann-Löser durch den AUSMV-Impulsfluss die nötige Robustheit verliehen, andererseits behält das AUSMDV-Verfahren in glatten Strömungsgebieten die Genauigkeit des AUSMD-Verfahrens.

Stoßsensor und Stoßstabilisierung

In [94] zeigen Wada und Liou eine leichte Tendenz des AUSMDV-Verfahrens zum Carbuncle-Phänomen und schlagen daher eine Stoßstabilisierung mit Hilfe des Flux-Vektor-Splitting-Verfahrens von Hänel vor, welches sehr ähnliche Eigenschaften wie das AUSMDV-Verfahren hat, jedoch etwas dissipativer ist.

Zur Stabilisierung des Stoßes wird zuerst die Stoßposition durch einen Stoßsensor ermittelt. Den von Wada und Liou vorgestellten Stoßsensor für eindimensionale Strömungen wurde in [56] für zweidimensionale Strömungen eingesetzt. Dabei wird zur Bestimmung der Stoßposition kontrolliert, ob die Normalgeschwindigkeit an der Zellfläche von Überschall nach Unterschall wechselt. Für eine Zellfläche in η -Richtung lautet der Sensor

$$\begin{aligned}
 \text{wenn } u_{\perp,l} > c_l \text{ und } u_{\perp,r} < c_r & : S_{\eta,i,j,k} = 1 \\
 \text{wenn } -u_{\perp,l} < c_l \text{ und } -u_{\perp,r} > c_r & : S_{\eta,i,j,k} = 1 \\
 \text{sonst} & : S_{\eta,i,j,k} = 0.
 \end{aligned} \tag{3.35}$$

In der oberen Hälfte von Abbildung 3.5 ist die Mach-Zahl-Verteilung einer reibungsfreie Strömung um eine Kugel dargestellt. Die Zuström��daten sind Tabelle 3.1 aufgelistet.

Zuström��druck	432 Pa	Partialdichte N_2	$5,855 \cdot 10^{-4} \text{ kg/m}^3$
Zuström��machzahl	8,25	Partialdichte O_2	$4,667 \cdot 10^{-3} \text{ kg/m}^3$
Vibrationstemperatur von N_2	250,21 K	Partialdichte NO	$2,334 \cdot 10^{-17} \text{ kg/m}^3$
Vibrationstemperatur von O_2	250,21 K	Partialdichte N	$1,628 \cdot 10^{-19} \text{ kg/m}^3$
Vibrationstemperatur von NO	250,21 K	Partialdichte O	$1,062 \cdot 10^{-18} \text{ kg/m}^3$

Tabelle 3.1: Zuström��daten für Stoßsensoruntersuchungen

In der unteren Hälfte von Abbildung 3.5 sind die Zellen gekennzeichnet, in denen der Stoßsensor aktiv wird. Wie zu sehen ist, wird dieser Stoßsensor auch in Regionen aktiv, in denen eine Stabilisierung der Strömung durch das Hänel-Verfahren weder notwendig noch gewünscht ist. Durch die Verwendung der Normalgeschwindigkeiten $u_{\perp,l}$ und $u_{\perp,r}$ wird festgestellt, ob im *lokalen* Riemann-Problem ein Stoß vorliegt. Das Carbuncle-Phänomen ist jedoch kein Resultat eines lokalen Riemann-Problems, sondern tritt nur bei Dimensionsaufspaltung in mehrdimensionalen Strömungen auf. Daher wird in der vorliegenden Arbeit der Stoßsensor abgeändert zu:

$$\begin{aligned}
 \text{wenn } \text{sign}(u_{\perp,l}) \cdot \|\vec{u}_l\| > c_l \text{ und } 0 < \text{sign}(u_{\perp,r}) \cdot \|\vec{u}_r\| < c_r & : S_{\eta,i,j,k} = 1 \\
 \text{wenn } 0 < -\text{sign}(u_{\perp,l}) \cdot \|\vec{u}_l\| < c_l \text{ und } -\text{sign}(u_{\perp,r}) \cdot \|\vec{u}_r\| > c_r & : S_{\eta,i,j,k} = 1 \\
 \text{sonst} & : S_{\eta,i,j,k} = 0.
 \end{aligned} \tag{3.36}$$

Hierbei sind $u_{\perp,l}$ und $u_{\perp,r}$ wieder die Normalgeschwindigkeiten und $\|\vec{u}_l\|$ sowie $\|\vec{u}_r\|$ die Beträge der extrapolierten Geschwindigkeiten an den Zellflächen. Dieser Stoßsensor sucht den tatsächlichen Stoß der Strömung, wie in Abbildung 3.6 zu sehen ist. Da bei einem Stoß, der diagonal im Netz liegt, das Carbuncle-Phänomen nicht auftritt, wird vorgeschlagen, beide Stoßsensoren

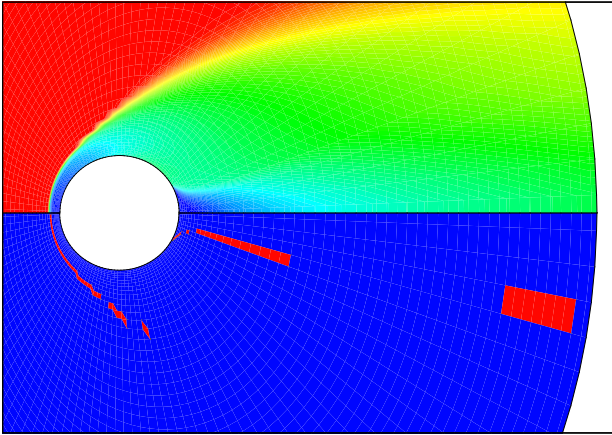


Abbildung 3.5: oben: Ma-Zahl-Verteilung einer reibungsfreien Strömung, unten: Gebiete, in denen der Stoßsensor nach [56] aktiv wird

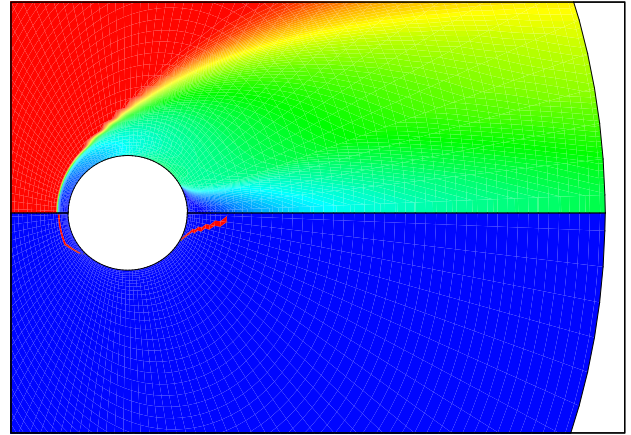


Abbildung 3.6: oben: Ma-Zahl-Verteilung einer reibungsfreien Strömung, unten: Gebiete, in denen der Stoßsensor (3.36) aktiv wird

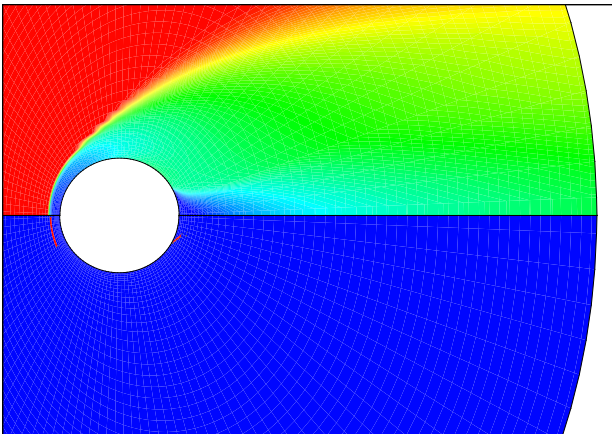


Abbildung 3.7: oben: Ma-Zahl-Verteilung einer reibungsfreien Strömung, unten: Gebiete, in denen der kombinierte Stoßsensor (3.35) und (3.36) aktiv wird

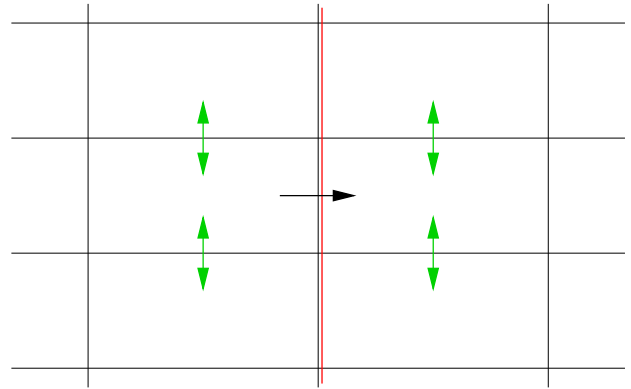


Abbildung 3.8: Einsatz des Hänel-Verfahrens zur Stoßstabilisierung: signalisiert an einer Zellfläche (schwarzer Pfeil) der Stoßsensor einen Stoß (rote Linie), so wird parallel zum Stoß (grüne Pfeile) das Hänel-Verfahren zur Stabilisierung eingesetzt

zu kombinieren: eine Stoßstabilisierung erfolgt nur dann, wenn *beide* Stoßsensoren einen Stoß anzeigen. Das Resultat dieses kombinierten Sensors ist in Abbildung 3.7 zu sehen.

Mit Hilfe dieses Stoßsensors kann nun gemäß Abbildung 3.8 das Hänel-Verfahren parallel zum Stoß zur Stabilisierung eingesetzt werden. Wird z. B. der Fluss in η -Richtung berechnet, so wird an jeder Zellfläche mit

$$S_{\xi,i,j,k} + S_{\xi,i-1,j,k} + S_{\xi,i,j+1,k} + S_{\xi,i-1,j+1,k} + S_{\zeta,i,j,k} + S_{\zeta,i,j,k-1} + S_{\zeta,i,j+1,k} + S_{\zeta,i,j+1,k-1} \neq 0$$

der Fluss mit dem dissipativere Hänel-Verfahren berechnet. Das Flux-Vector-Splitting-Verfahren

rens von Hänel ist durch

$$\varphi_{1/2} = u_l^+(\rho\Psi)_l + u_r^-(\rho\Psi) + p_{1/2} \quad (3.37)$$

gegeben. Hierbei werden die Geschwindigkeiten u_l^+ und u_r^- durch

$$u^\pm = \begin{cases} \pm \frac{1}{4c} (u_\perp \pm c)^2 & \text{für } |u_\perp| \leq c, \\ \frac{1}{2}(u_\perp + |u_\perp|) & \text{sonst} \end{cases} \quad (3.38)$$

und der Druckterm $p_{1/2} = p_l^+ + p_r^-$ durch

$$p^\pm = pu^\pm \cdot \begin{cases} \frac{1}{c} \left(\pm 2 - \frac{u_\perp}{c} \right) & \text{für } |u_\perp| \leq c, \\ \frac{1}{u_\perp} & \text{sonst} \end{cases} \quad (3.39)$$

berechnet.

Entropie-Korrektur

Das AUSMDV-Verfahren hat nach [94] leichte Tendenzen zum Bilden eines unphysikalischen Verdünnungsstoß (siehe Abb. 3.9). Aus diesem Grund führen Wada und Liou eine zusätzliche numerische Dissipation ein. Hierzu werden die beiden folgenden Fälle betrachtet

$$\begin{cases} \text{Fall A: } u_{\perp,l} - c_l < 0 & \text{und } u_{\perp,r} - c_r > 0 \\ \text{Fall B: } u_{\perp,l} + c_l < 0 & \text{und } u_{\perp,r} + c_r > 0. \end{cases}$$

Sind beide Fälle gleichzeitig erfüllt oder ist keiner der beiden Fälle erfüllt, so bleibt die AUSMDV-Flussberechnung unverändert. Trifft jedoch *entweder* Fall A *oder* Fall B zu, so wird

$$\text{für Fall A durch: } \varphi_{\text{AUSMDV,E-Korr.}} = \varphi_{\text{AUSMDV}} - C\Delta(u_\perp - c)\Delta(\rho\Psi)$$

$$\text{für Fall B durch: } \varphi_{\text{AUSMDV,E-Korr.}} = \varphi_{\text{AUSMDV}} - C\Delta(u_\perp + c)\Delta(\rho\Psi)$$

numerische Dissipation hinzugefügt. Hierbei ist $\Delta(\cdot) = (\cdot)_r - (\cdot)_l$. Im Rahmen dieser Arbeit wird C auf den in [94] vorgeschlagenen Wert 0,125 gesetzt.

Für die Berechnung einer stationären Strömung führt der obige Sensor zu Oszillationen in der Strömungsberechnung, da der Sensor nur jene Flüsse mit einer Entropie-Korrektur beaufschlagt, an denen die Strömung von Unterschall nach Überschall wechselt. Für ein stabiles Verfahren ist es notwendig, den Einsatz der Entropie-Korrektur zu erweitern. Dazu wird der Parameter C im obigen Sensor variiert zu $s \cdot C$.

Betrachtet werden alle Zellflächen, an denen eine Beschleunigung der Strömung stattfindet, es muss also gelten

$$0 < u_{\perp,l} < u_{\perp,r} \quad \text{oder} \quad u_{\perp,l} < u_{\perp,r} < 0.$$

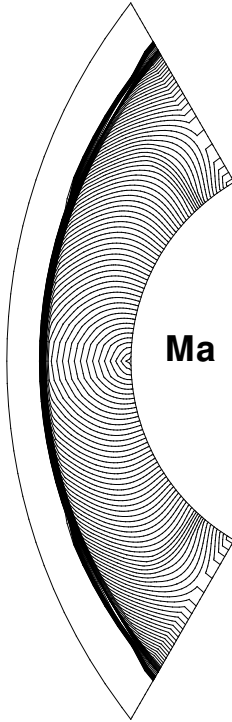


Abbildung 3.9: AUSMDV ohne Entropie-Korrektur, leichte Tendenz zu einem Verdünnungsstoß

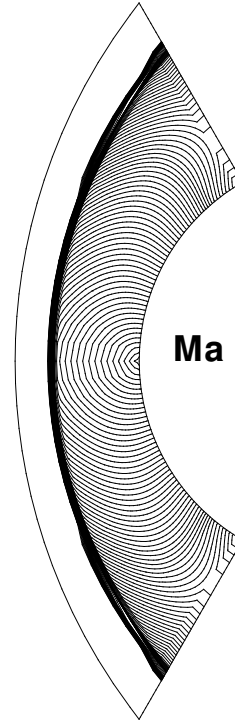


Abbildung 3.10: AUSMDV mit Entropie-Korrektur

Wie im AUSMDV-Verfahren wird eine Referenzschallgeschwindigkeit $c_m = \max(c_l, c_r)$ gewählt, mit der links und rechts vom Riemann-Problem die lokalen Machzahlen berechnet werden

$$\text{Ma}_l = \frac{u_{\perp,l}}{c_m}, \quad \text{Ma}_r = \frac{u_{\perp,r}}{c_m}.$$

Damit wird der lokale Parameter s bestimmt durch

$$\begin{aligned} \text{Ma}_l > 1 > \text{Ma}_r &: s = 1 \\ \text{Ma}_r > 1 > \text{Ma}_l &: s = 1 \end{aligned} \tag{3.40}$$

$$\text{sonst} : s = 1 - \max\left(0, \frac{\min(|\text{Ma}_r - 1|, |\text{Ma}_l - 1|)}{\Delta\text{Ma}}\right). \tag{3.41}$$

Der Parameter s bewirkt ein lineares Ansteigen der Entropie-Korrektur, sobald $|\text{Ma}_l - 1| < \Delta\text{Ma}$ oder $|\text{Ma}_r - 1| < \Delta\text{Ma}$ ist. Um nicht zu viel Dissipation in das Verfahren hinzuzufügen, sollte der freie Parameter ΔMa nicht zu groß gewählt werden. In dieser Arbeit wurde stets der Wert $\Delta\text{Ma} = 0,2$ eingesetzt. Der korrigierte AUSMDV-Fluss lautet dann

$$\varphi_{\text{AUSMDV,E-Korr.}} = \varphi_{\text{AUSMDV}} - s \cdot C \Delta(u_{\perp}) \Delta(\rho\Psi). \tag{3.42}$$

Durch die eingeführte Entropie-Korrektur werden Verdünnungsstöße erfolgreich vermieden (Abb. 3.10). Außerdem wird durch die Erweiterung der Entropiekorrektur die Berechnung der stationären Strömung ermöglicht.

3.2.3 Verfahren höherer Ordnung

Für die Berechnung der reibungsfreien Flüsse werden die Links- und Rechtszustände Q^l und Q^r an den Zellflächen benötigt (vgl. (3.3)). Bei einer konstanten Verteilung der Stömungsgrößen innerhalb einer Zelle

$$Q_{i+\frac{1}{2}}^l = Q_i \quad Q_{i-\frac{1}{2}}^r = Q_i, \quad (3.43)$$

entsteht ein Verfahren erster Ordnung in der räumlichen Genauigkeit.

Zur Erhöhung der räumlichen Genauigkeit wird eine MUSCL-Extrapolation (**M**onotonic **U**pwind **S**cheme for **C**onservation **L**aws) eingesetzt. Hierbei wird von einer linearen Verteilung der Strömungsgrößen innerhalb der Zellen ausgegangen. Damit werden die Links- und Rechtszustände an den Zellflächen Q_r und Q_l durch eine lineare Extrapolation

$$Q_{l,i+\frac{1}{2}} = Q_i + s(Q) \frac{\Delta x}{2}, \quad Q_{r,i-\frac{1}{2}} = Q_i - s(Q) \frac{\Delta x}{2} \quad (3.44)$$

berechnet. Aus Stabilitätsgründen werden in Anwendungsprogrammen nicht die Erhaltungsgrößen Q , sondern Primitivvariablen P extrapoliert

$$P_{i+\frac{1}{2}}^l = P_i + s(P) \frac{\Delta x}{2}, \quad P_{i-\frac{1}{2}}^r = P_i - s(P) \frac{\Delta x}{2}.$$

Für die Berechnung der Steigungen $s(P)$ in einer Zelle werden einseitige Differenzen und die zentrale Differenz genommen. Für eine beliebige Strömungsgröße a lauten diese auf einem nicht-äquidistanten Netz

$$\delta^- a = 2 \cdot \frac{a_{i+1} - a_i}{\Delta x_{i+1} + \Delta x_i} \quad (3.45)$$

$$\delta^+ a = 2 \cdot \frac{a_i - a_{i-1}}{\Delta x_i + \Delta x_{i-1}} \quad (3.46)$$

$$\delta^0 a = 2 \cdot \frac{(\Delta x_{i-1} + \Delta x_i)^2 (a_{i+1} - a_i) + (\Delta x_{i+1} + \Delta x_i)^2 (a_i - a_{i-1})}{(\Delta x_{i-1} + \Delta x_i)(\Delta x_{i+1} + \Delta x_i)(\Delta x_{i-1} + 2 \cdot \Delta x_i + \Delta x_{i+1})}. \quad (3.47)$$

Aus diesen drei Steigungen wird mit Hilfe von so genannten Limitern die Steigung $s(P)$ bestimmt. In der vorliegenden Arbeit werden die folgenden Limiter eingesetzt.

Der MC-Limiter (*monotonized central-difference limiter*) von van Leer

$$\delta_{MC} = \begin{cases} 0 & \text{für } \delta^+ \cdot \delta^- < 0 \\ \text{sign}(\delta^0) \cdot \min(|\delta^-| \cdot \kappa, |\delta^0|, |\delta^+| \cdot \kappa) & \text{sonst} \end{cases} \quad (3.48)$$

mit $1 < \kappa < 2$ ist ein sehr flexibler TVD-Limiter. Haben die einseitigen Differenzen δ^- und δ^+ unterschiedliches Vorzeichen, so wird die Steigung δ_{MC} wie bei jedem TVD-Limiter auf null gesetzt. Durch dem Parameter κ kann der Limiter dem Strömungsproblem angepasst werden. Für $\kappa > 1,5$ ist eine Strömungsberechnung meist sehr instabil. Für $\kappa = 1$ entspricht der MC-Limiter dem minmod-Limiter. Für diesen Parameterwert ist für die meisten Strömungssimulationen eine stationäre Strömungslösung berechenbar.

Der MC-Limiter ist zwar für die Strömungsmodellierung ein durchaus einsetzbarer Limiter, jedoch gab es bei Konvergenzuntersuchungen, die in den Kapiteln 5 und 6 dargestellt sind, einige Probleme mit diesem Limiter. Aus diesem Grund wurden weitere Limiter in das Verfahren integriert. Die folgenden Limiter bestimmen die Steigung innerhalb einer Zelle durch

$$\delta = \alpha_1 \cdot \Phi(r)\delta^- + \alpha_2 \cdot \Phi\left(\frac{1}{r}\right)\delta^+ \quad (3.49)$$

mit

$$\alpha_1 = \frac{(\Delta x_{i-1} + \Delta x_i)}{(\Delta x_{i-1} + 2 \cdot \Delta x_i + \Delta x_{i+1})} \quad (3.50)$$

$$\alpha_2 = \frac{(\Delta x_{i+1} + \Delta x_i)}{(\Delta x_{i-1} + 2 \cdot \Delta x_i + \Delta x_{i+1})} \quad (3.51)$$

$$r = \frac{\delta^+}{\delta^-}. \quad (3.52)$$

Dieser Ansatz ist so gewählt, dass für $\Phi(r) = 1$ und $\Phi(\frac{1}{r}) = 1$ die Steigung δ exakt der zentralen Differenz δ^0 entspricht. Für die Limiter-Funktion Φ wurden zwei Varianten implementiert:

Van Leer-Limiter:

$$\Phi(r) = \frac{r + |r|}{1 + r}, \quad (3.53)$$

Van Albada-Limiter:

$$\Phi(r) = \frac{r + r^2}{1 + r^2}. \quad (3.54)$$

Die beschriebenen Limiter können im vorgestellten Verfahren sowohl auf die Steigungen der primitiven Variablen, als auch auf die Steigungen der charakteristischen Variablen angewandt werden.

Steigungstransformation

Für die Limitierung der Steigungen der *charakteristischen* Variablen müssen zuerst die Steigungen der Primitivvariablen in die Steigungen der charakteristischen Variablen transformiert werden. Nach der Limitierung erfolgt die Rücktransformation in die Steigungen der Primitivvariablen.

Der Vektor s_P der Steigungen der Primitivvariablen P wird in den Vektor s_W der Steigungen der charakteristischen Variablen W durch

$$s_W = R^{-1} \frac{\partial Q}{\partial P} \cdot s_P$$

transformiert. Entsprechend lautet die Rücktransformation

$$s_P = \left(\frac{\partial Q}{\partial P} \right)^{-1} R \cdot s_W.$$

Hierbei ist R wieder die Matrix der Eigenvektoren der Fluss-Jacobi-Matrix (Anhang A). Mit dem Erhaltungsvektor Q und dem Vektor P der Primitivvariablen

$$Q = \begin{pmatrix} \rho_s \\ \rho u \\ \rho v \\ \rho w \\ \rho e_t \\ \rho_{s'} e_{vib,s'} \end{pmatrix} \quad P = \begin{pmatrix} \rho_s \\ u \\ v \\ w \\ p \\ \rho_{s'} e_{vib,s'} \end{pmatrix}$$

entstehen die Transformationsmatrizen

$$R^{-1} \frac{\partial Q}{\partial P} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -\frac{\rho n_x}{2c} & -\frac{\rho n_y}{2c} & -\frac{\rho n_z}{2c} & \frac{1}{2c^2} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\xi_1}{c^2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\xi_2}{c^2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -\frac{\xi_3}{c^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\frac{\xi_4}{c^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\frac{\xi_5}{c^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\rho b_x^1}{c} & \frac{\rho b_y^1}{c} & \frac{\rho b_z^1}{c} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\rho b_x^2}{c} & \frac{\rho b_y^2}{c} & \frac{\rho b_z^2}{c} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\xi_1 e_{vib,1}}{c^2} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\xi_2 e_{vib,2}}{c^2} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\xi_3 e_{vib,3}}{c^2} & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \frac{\rho n_x}{2c} & \frac{\rho n_y}{2c} & \frac{\rho n_z}{2c} & \frac{1}{2c^2} & 0 & 0 & 0 \end{pmatrix}$$

sowie

$$\frac{\partial Q}{\partial P}^{-1} R = \begin{pmatrix} \xi_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_1 \\ \xi_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_2 \\ \xi_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_3 \\ \xi_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_4 \\ \xi_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi_5 \\ -\frac{c}{\rho} n_x & 0 & 0 & 0 & 0 & 0 & \frac{c b_x^1}{\rho} & \frac{c b_x^2}{\rho} & 0 & 0 & 0 & \frac{c}{\rho} n_x \\ -\frac{c}{\rho} n_y & 0 & 0 & 0 & 0 & 0 & \frac{c b_y^1}{\rho} & \frac{c b_y^2}{\rho} & 0 & 0 & 0 & \frac{c}{\rho} n_y \\ -\frac{c}{\rho} n_z & 0 & 0 & 0 & 0 & 0 & \frac{c b_z^1}{\rho} & \frac{c b_z^2}{\rho} & 0 & 0 & 0 & \frac{c}{\rho} n_z \\ c^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c^2 \\ \xi_1 e_{vib,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \xi_1 e_{vib,1} \\ \xi_2 e_{vib,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \xi_2 e_{vib,2} \\ \xi_3 e_{vib,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \xi_3 e_{vib,3} \end{pmatrix}.$$

Hierzu werden die Vektoren (b_x^1, b_y^1, b_z^1) und (b_x^2, b_y^2, b_z^2) benötigt, welche zusammen mit dem Flächennormalenvektor (n_x, n_y, n_z) ein Orthonormalsystem bilden.

3.3 Viskose Flüsse

Für die viskosen Flüsse (2.14) werden die räumlichen Ableitungen der Primitivvariablen $u, v, w, T, T_{vib,s}$ und ψ an den Zellflächen benötigt. Um diese Ableitungen zu berechnen, werden zuerst die räumlichen Ableitungen nach ξ, η und ζ im Rechenraum mittels zentraler Differenzen, an den Netzrändern mittels einseitiger Differenzen mit einer Genauigkeit zweiter Ordnung berechnet. Über die Beziehungen

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial}{\partial \zeta} \quad (3.55)$$

$$\frac{\partial}{\partial y} = \frac{\partial \xi}{\partial y} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial y} \frac{\partial}{\partial \zeta} \quad (3.56)$$

$$\frac{\partial}{\partial z} = \frac{\partial \xi}{\partial z} \frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial z} \frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial z} \frac{\partial}{\partial \zeta}. \quad (3.57)$$

werden die Ableitungen nach ξ, η und ζ zurück in den physikalischen Raum transformiert. Für die dabei benötigten metrischen Ableitungen gilt

$$\begin{aligned} \frac{\partial \xi}{\partial x} &= \frac{1}{J} \left(\frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \eta} \right) & \frac{\partial \eta}{\partial x} &= \frac{1}{J} \left(\frac{\partial y}{\partial \zeta} \frac{\partial z}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \zeta} \right) & \frac{\partial \zeta}{\partial x} &= \frac{1}{J} \left(\frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \xi} \right) \\ \frac{\partial \xi}{\partial y} &= \frac{1}{J} \left(\frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \zeta} \right) & \frac{\partial \eta}{\partial y} &= \frac{1}{J} \left(\frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial z}{\partial \xi} \right) & \frac{\partial \zeta}{\partial y} &= \frac{1}{J} \left(\frac{\partial x}{\partial \eta} \frac{\partial z}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} \right) \\ \frac{\partial \xi}{\partial z} &= \frac{1}{J} \left(\frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \eta} \right) & \frac{\partial \eta}{\partial z} &= \frac{1}{J} \left(\frac{\partial x}{\partial \zeta} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \zeta} \right) & \frac{\partial \zeta}{\partial z} &= \frac{1}{J} \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \right) \end{aligned} \quad (3.58)$$

mit

$$J = \det \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix}.$$

Die Werte der Primitivvariablen $u, v, w, T, T_{vib,s}$ und ψ auf den Zellflächen sind die Mittelwerte der Zellzentrumswerte,

$$u_{i+\frac{1}{2}} = \frac{1}{2}(u_i + u_{i+1}).$$

Am Abströmrand, Zuströmrand und Festkörperperrand liegen die Primitivvariablen bereits auf der Zellfläche.

In den viskosen Flüssen treten die Diffusionskoeffizienten D_s , die Viskosität μ , die translatorisch-rotatorische Wärmeleitfähigkeit λ_{tr} sowie die vibratorische Wärmeleitfähigkeit $\lambda_{vib,s}$ auf. Die

Berechnung dieser Transportkoeffizienten werden für das 5 Komponenten / 4 Temperaturen-Luftmodell im Temperaturbereich von 250 K bis 30000 K nach Chapman/Cowling, die Kollisionsintegrale nach Capitelli berechnet, [17]. Die Herleitung und Implementierung der Transportkoeffizienten in das hier vorgestellte Verfahren wurde von Fertig durchgeführt.

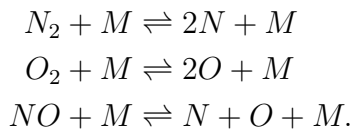
3.4 Quellterme

Die Quellterme

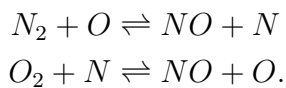
$$S = \begin{pmatrix} \dot{\omega}_s \\ 0 \\ 0 \\ 0 \\ 0 \\ Q_{t-v,k} + Q_{v-v,k} + Q_{c-v,k} \end{pmatrix} \quad (3.59)$$

aus Gleichung (2.18) beinhalten die chemischen Quellterme $\dot{\omega}_s$ sowie die energetischen Quellterme $Q_{t-v,k}$, $Q_{v-v,k}$ und $Q_{c-v,k}$.

Die chemischen Quellterme $\dot{\omega}_s$ in den Massenerhaltungsgleichungen modellieren die chemischen Reaktionen in der Gasphase. Auf das Luftmodell mit den drei molekularen Spezies N_2 , O_2 , NO und den beiden atomaren Spezies N , O wird das Reaktionsmodell von Park angewandt ([64],[38]). Jedes der drei Moleküle N_2 , O_2 und NO kann durch einen Stoßpartner $M \in \{N_2, O_2, NO, N, O\}$ dissoziiert werden. Daraus ergeben sich 15 Dissoziationsreaktionen,



Hinzu kommen zwei Austauschreaktionen



Somit besteht das Reaktionsmodell aus 17 chemischen Reaktionen.

Der Quellterm S hat keine Einträge in den Impulserhaltungsgleichungen. Ebenso bleibt die Gesamtenergie durch die Quellterme erhalten, so dass auch die Gesamtenergiegleichung keinen Quellterm hat.

Die Vibrationsenergiegleichungen beinhalten die energetischen Quellterme, welche die Energieaustauschmechanismen zwischen Translation und Schwingung (Q_{t-v}), Schwingung und Schwingung (Q_{v-v}) und Schwingung und Chemie (Q_{v-c}) modellieren.

Für eine genaue Modellierung der thermochemischen Relaxationsprozesse ist es wichtig, dass alle Quellterme physikalisch konsistent zueinander modelliert werden. Hierbei müssen der Einfluss

der Chemie auf die Anregung innerer Freiheitsgrade und auch der Einfluss der inneren Freiheitsgrade auf die Chemie verträglich zueinander modelliert werden. Dazu wird ein sehr fortschrittliches CVCV-Modell (Coupled Vibration-Chemistry-Vibration Model) eingesetzt, welches von Knab entwickelt wurde, [43].

Die Quellterme wurden für das hier vorgestellte Verfahren aus dem zweidimensionalen/axialsymmetrischen Verfahren übernommen und auf das hier beschriebene Luftmodell angepasst.

3.5 Rand- und Anfangswerte

Für die Berechnung der Navier-Stokes-Gleichungen müssen noch Randbedingungen modelliert und Anfangswerte gesetzt werden.

Randbedingungen

Über den Zuströmrand strömt das Gasgemisch in das Rechengebiet hinein, am Abströmrand verlässt die Strömung das Rechengebiet. Für diese beiden Ränder werden einfache Überschallrandbedingungen angesetzt. Da in Überschallgebieten sämtliche Charakteristiken in eine Richtung laufen, können am Überschallzuströmrand sämtliche Zuströmparameter fest vorgegeben werden, da keine Charakteristiken aus dem Strömungsfeld auf den Zuströmrand zulaufen. Am Überschallabströmrand werden sämtliche Zellzentrumswerte der letzten Zelle auf den Rand extrapoliert.

Weitere Ränder des strukturierten Netzes sind die Symmetrieränder und die metrische Singularität. An den Symmetrierändern wird der Erhaltungsvektor in Pseudozellen gespiegelt. An der metrischen Singularität werden die Zellzentrumswerte der Nachbarzellen arithmetisch gemittelt.

Im vorgestellten Verfahren spielt der Festkörpertrand eine besondere Rolle, da die Modellierung dieses Randes sich signifikant auf den berechneten Oberflächenwärmestrom und damit auf die Oberflächentemperatur auswirkt. Aus diesem Grund wurde für die Festkörpertrandbedingung von Daiß ein sehr detailliertes Modell, basierend auf einer vollständigen gaskinetischen Flussberechnung, entwickelt (siehe [14]), welches von Infed in das hier vorgestellte Verfahren übertragen wurde [34]. Als Festkörpertrandbedingungen sind implementiert:

1. $\frac{\partial p}{\partial n} = 0$
2. nicht-, voll- oder teilkatalytisches Verhalten des Oberflächenmaterials
3. Haft- oder Gleitrandbedingung
4. konstante oder strahlungsadiabate Oberflächentemperatur
5. thermisches Gleichgewicht und Nichtgleichgewicht an der Oberfläche

Anfangswerte

Für die instationären Navier-Stokes-Gleichungen werden Anfangswerte benötigt. Da die stationäre Strömungslösung berechnet werden soll, ist der gewählte Anfangswert nicht entscheidend für die Lösung. Daher ist es lediglich notwendig, einen Anfangswert zu wählen, von dem aus das numerische Verfahren in der Lage ist, die stationäre Lösung zu berechnen.

Als Anfangswert wird das gesamte Strömungsfeld auf die Zuströmdaten gesetzt. Lediglich in den letzten sechs Zellen vor dem Festkörperperrand wird die Geschwindigkeit reduziert, so dass auf dem Festkörperperrand die Geschwindigkeit zu Null wird, womit die Haftrandbedingung erfüllt wäre. Gleichzeitig wird in der Umgebung des Festkörperperrandes die Translationstemperatur erhöht.

Kapitel 4

Implizites Lösungsverfahren

Die Navier-Stokes-Gleichungen in Finite-Volumen-Formulierung (3.1) haben die Gestalt

$$V \frac{\partial Q}{\partial t} = R(Q). \quad (4.1)$$

In der Funktion R sind die reibungsfreien und viskosen Flüsse sowie die Quellterme zusammengefasst. Die Matrix V ist eine Diagonalmatrix, deren Einträge aus den Zellvolumina bestehen. Zusätzlich zu den Navier-Stokes-Gleichungen müssen Randbedingungen betrachtet werden, welche unabhängig vom Typ in der Form

$$R_r(Q) = 0 \quad (4.2)$$

geschrieben werden können. Für die Gleichungen (4.1) und (4.2) soll die stationäre Lösung berechnet werden, also die Lösung Q^* , für die gilt

$$\left. \frac{\partial Q}{\partial t} \right|_{Q^*} = 0 \quad \stackrel{(4.1)}{\iff} \quad R(Q^*) = 0$$
$$R_r(Q^*) = 0.$$

Es wird vorausgesetzt, dass eine *eindeutige* stationäre Lösung existiert und dass diese *asymptotisch stabil* ist. Dies bedeutet, dass die stationäre Lösung stabil ist, und dass für jede instationäre Lösung $Q(t)$ mit $Q(0)$ in einer Umgebung von Q^* gilt

$$\lim_{t \rightarrow \infty} Q(t) = Q^*.$$

Bei der Auswahl des numerischen Verfahrens für die Berechnung der stationären Lösung müssen die auftretenden physikalischen Phänomene berücksichtigt werden. In der Strömung treten Verdichtungs- und Rekompansionsstöße sowie Rezirkulations- und Expansionsgebiete auf. Diese Strömungsphänomene beeinflussen sich auch bei räumlicher Trennung gegenseitig. Hinzu kommen steife Quellterme, die die Berechnung der stationären Lösung erschweren.

Ebenso müssen bei der Auswahl des numerischen Verfahrens die Anforderungen an die numerische Lösung berücksichtigt werden. In Strömungsprogrammen ist es üblich, für die Berechnung

einer stationären Lösung die Residuennorm $\|R\|$ um drei bis vier Größenordnungen abzusenkten. Erfahrungsgemäß muss jedoch beim vorgestellten Verfahren für eine genaue Berechnung des Oberflächenwärmestroms die Residuennorm um mindestens *sechs* Größenordnungen reduziert werden. Nicht jedes numerische Verfahren ist dazu in der Lage.

Explizite Verfahren erfüllen die erwähnten Anforderungen nicht, da sie sowohl wegen der steifen Quellterme als auch wegen der räumlichen Ausbreitung der Strömung zu langsam konvergieren. Zumindest die Quellterme sollten wegen ihrer Steifheit implizit behandelt werden.

Heutzutage werden in vielen Strömungsprogrammen Mehrgittertechniken zur Konvergenzbeschleunigung erfolgreich eingesetzt. Im Bereich der reaktiven Hyperschallströmungen ist der Einsatz von Mehrgittertechniken mittlerweile zwar möglich, aber nach wie vor sehr schwierig. Einerseits gibt es Probleme in der Behandlung von Quelltermen mit Mehrgittertechniken. Andererseits kann ein starker Stoß mit Mehrgittertechniken nicht berechnet werden, da ein starker Stoß zu sensibel auf einen Gitterwechsel reagiert. Aus diesem Grund sind nur sehr wenige Mehrgitterverfahren für reaktive Hyperschallströmungen in der Literatur zu finden ([23],[24],[72],[42]).

In dieser Arbeit wurde als Lösungsverfahren ein Zeitschritt-Newton-Verfahren ausgewählt, das zwar einerseits sehr aufwändig ist, andererseits aber eine große Stabilität aufweist. In diesem Kapitel wird nach der Herleitung dieses impliziten Verfahrens eine praktikable Umsetzung des Verfahrens erläutert. Desweiteren werden Lösungsverfahren für lineare Gleichungssysteme vorgestellt.

4.1 Zeitschritt-Newton-Verfahren

Das Zeitschritt-Newton-Verfahren entsteht aus einer impliziten Euler-Zeitdiskretisierung von Gleichung (4.1),

$$V \frac{Q^{n+1} - Q^n}{\Delta t} = R(Q^{n+1}). \quad (4.3)$$

Nach einer Linearisierung der Fluss- und Quelltermbilanz R ergibt sich

$$V \frac{Q^{n+1} - Q^n}{\Delta t} = R(Q^n) + \frac{\partial R}{\partial Q}(Q^{n+1} - Q^n). \quad (4.4)$$

Die Randbedingungen (4.2) können sowohl explizit

$$R_r(Q^n) = 0, \quad (4.5)$$

als auch implizit

$$R_r(Q^{n+1}) = 0 \quad (4.6)$$

formuliert werden. Aufgrund der deutlich besseren Konvergenzeigenschaften ist eine implizite Formulierung wenn möglich vorzuziehen. Im Fall von impliziten Randbedingungen werden diese ebenfalls linearisiert:

$$0 = R_r(Q^{n+1}) \approx R_r(Q^n) + \frac{\partial R_r}{\partial Q}(Q^{n+1} - Q^n). \quad (4.7)$$

Durch Umsortieren der Gleichungen (4.4) und (4.7) entsteht das Zeitschritt-Newton-Verfahren

$$\left(V \frac{1}{\Delta t} - \frac{\partial R}{\partial Q} \right) \Delta Q = R(Q^n) \quad (4.8)$$

$$-\frac{\partial R_r}{\partial Q} \Delta Q = R_r(Q^n) \quad (4.9)$$

mit

$$Q^{n+1} = Q^n + \Delta Q. \quad (4.10)$$

Die Randbedingungsgleichungen (4.9) unterscheiden sich von den Navier-Stokes-Gleichungen (4.8) nur durch den fehlenden instationären Term $V \frac{1}{\Delta t}$. Daher werden im Folgenden die Funktionen R und R_r zusammengefasst und die Matrix V in den Randgleichungen mit Null-Einträgen erweitert. Damit lassen sich Gleichung (4.8) und Gleichung (4.9) zusammenfassen. Das daraus resultierende Iterationsverfahren ist in Algorithmus 4.1 dargestellt.

Zeitschritt-Newton-Verfahren	
$\left(V \frac{1}{\Delta t} - \frac{\partial R}{\partial Q} \right) \Delta Q = R(Q^n)$	(4.11)
$Q^{n+1} = Q^n + \omega \Delta Q$	$0 < \omega \leq 1$ (4.12)

Algorithmus 4.1: Zeitschritt-Newton-Verfahren

Zur Stabilisierung des Verfahrens wurde im Alg. 4.1 ein Relaxationsparameter ω eingefügt, auf dessen Bedeutung später eingegangen wird.

Das Verfahren wird beim Übergang $\Delta t \rightarrow \infty$ zum reinen Newton-Verfahren für die Funktion R , welches bekanntlich eine quadratische Konvergenz hat. Allerdings konvergiert das Newton-Verfahren nur dann, wenn der Startwert der Iterationen nahe genug bei der stationären Lösung liegt. Daher werden bei diesem Verfahren zuerst kleine Zeitschritte gewählt, um in die Nähe der stationären Lösung zu gelangen. Anschließend kann der Zeitschritt zumindest theoretisch beliebig erhöht werden, was zu einer quadratischen Konvergenz führen kann.

Das Iterationsverfahren gemäß Alg. 4.1 besteht damit aus folgenden Schritten:

- Aufstellen der Funktion $R(Q^n)$
- Aufstellen der Jacobi-Matrix $\frac{\partial R}{\partial Q}$
- Bestimmen des Zeitschrittes Δt
- Lösen des linearen Gleichungssystems (4.11)
- Berechnen des neuen Erhaltungsvektors Q^{n+1} gemäß (4.12)

Die Berechnung der Funktion R wurde in Kapitel 2 detailliert beschrieben. Im Folgenden wird auf die Berechnung der Jacobi-Matrix und auf das Lösen der linearen Gleichungssysteme eingegangen.

4.2 Inkonsistentes Newton-Verfahren

Im Algorithmus 4.1 muss die Berechnung von $R(Q^n)$ so genau wie möglich erfolgen, da die Funktion R die Strömungslösung bestimmt.

Zwar steigt der Speicherplatzbedarf der Jacobi-Matrix $\frac{\partial R}{\partial Q}$ nur linear mit der Anzahl der Zellen, doch ist der für die exakte Berechnung der Jacobi-Matrix benötigte Speicherplatz- und Rechenzeitbedarf indiskutabel. Die Quellterm- und Flussbilanz R einer Innenfeldzelle des strukturierten Netzes hängt von insgesamt 25 Zellen ab. Daher bestünde die Jacobi-Matrix bei exakter Berechnung aus 25 Blockdiagonalen. Aufgrund der 12 Erhaltungswerte bzw. der 12 Erhaltungsgleichungen pro Zelle entspräche der Speicherplatzbedarf für die Jacobi-Matrix $\frac{\partial R}{\partial Q}$ dem 300-fachen des Speicherplatzbedarfs des Vektors $R(Q^n)$. Die Jacobi-Matrix hat jedoch keine Auswirkungen auf die Strömungslösung, sondern nur auf die Konvergenz des Newton-Verfahrens. Somit ist es möglich, diese Matrix nur approximativ zu berechnen. Dabei werden im vorgestellten Verfahren folgende Approximationen gemacht:

- Die Jacobi-Matrix der reibungsfreien Flüsse wird nur mit den Flüssen erster Ordnung räumlicher Genauigkeit berechnet. Im AUSMDV-Verfahren wird dazu der numerische Fluss (3.25) und (3.32) analytisch nach dem Erhaltungsvektor (2.8) abgeleitet. Für das Roe-Verfahren (3.20) werden weitere Vereinfachungen vorgenommen, indem die Matrizen R , Λ und R^{-1} nicht abgeleitet werden.
- Für die Berechnung der Jacobi-Matrix der viskosen Flüsse wird für diese eine Thin-Shear-Layer-Approximation (TSL) durchgeführt. Bei einer TSL-Approximation z. B. des Flusses E_v in ξ -Richtung werden sämtliche Ableitungen nach η und ζ vernachlässigt.
- Die Transportkoeffizienten werden als konstant betrachtet.

Durch diese Approximationen reduziert sich die Jacobi-Matrix auf 7 Blockdiagonalen. Damit benötigt die Jacobi-Matrix nur noch das 84-fache des Speicherplatzbedarfs des Vektors $R(Q^n)$.

Um das Besetzungsschema der Matrix beizubehalten, müssen auch bei den Randbedingungen Vereinfachungen getroffen werden. Dazu werden die Jacobi-Matrizen für die Oberflächenrandbedingung ebenfalls nur von den Randbedingungen mit räumlicher Genauigkeit erster Ordnung berechnet. Außerdem kann die Behandlung der metrischen Singularität nicht implizit erfolgen, somit bleibt diese Randbedingung als einzige explizit. Da die mit den genannten Approximationen gebildete Jacobi-Matrix inkonsistent zur Funktion R ist, wird in der vorliegenden Arbeit dieses Verfahren als *inkonsistentes Newton-Verfahren* bezeichnet.

Trotz approximativer Berechnung der Jacobi-Matrix bleibt der hohe Speicherplatzbedarf ein Nachteil des numerischen Verfahrens. Dieser hohe Speicherplatzbedarf entsteht durch die Jacobi-Matrix sowie durch das Lösungsverfahren für das lineare Gleichungssystem. Für ein großes Anwendungsproblem mit einem Netz aus 1.000.000 Zellen wird daher ein Speicherplatz von ca. 25 GB benötigt. Um den Speicherplatzbedarf zu reduzieren wird bei der Implementierung des Verfahrens ein programmiertechnischer Trick angewandt. Für eine hinreichende Konvergenz ist es zwingend erforderlich, dass das Computerprogramm mit doppelter Genauigkeit rechnet. Dies bedeutet, dass pro Gleitkommazahl 8 Byte Speicherplatz benötigt werden. Sämtliche Variablen müssen mit doppelter Genauigkeit berechnet und abgespeichert werden, insbesondere alle Variablen und Berechnungen, die für das Aufstellen des Vektors $R(Q^n)$ notwendig sind. Wie jedoch bereits erwähnt wurde, ist es nicht erforderlich, die Jacobi-Matrix exakt zu berechnen. Eine approximativ berechnete Jacobi-Matrix braucht aber auch nicht mit doppelter Genauigkeit abgespeichert zu werden. Stattdessen wird die Matrix zwar mit doppelter Genauigkeit berechnet, aber nur mit einfacher Genauigkeit abgespeichert. Da eine Gleitkommazahl in einfacher Genauigkeit nur 4 Byte Speicherplatz benötigt, halbiert sich durch diesen programmiertechnischen Trick der Speicherplatzbedarf für die Jacobi-Matrix.

4.3 Lokaler Zeitschritt

Für den Zeitschritt Δt ist es in einem Zeitschritt-Newton-Verfahren üblich, einen lokalen Zeitschritt über eine CFL-Zahl zu bestimmen. Hierzu wird durch Vorgabe einer CFL-Zahl mit Hilfe der lokalen Geschwindigkeit $|u| + c$ und der Netzgröße Δx der lokale Zeitschritt

$$\Delta t = \frac{\Delta x \cdot \text{CFL}}{|u| + c} \quad (4.13)$$

berechnet. Für die hier betrachteten Hyperschallströmungen ist es erforderlich, zuerst den Zeitschritt klein zu halten ($\text{CFL} < 10$), bis der Stoß positioniert ist. Anschließend kann der Zeitschritt erhöht werden, um die Konvergenz des Verfahrens zu beschleunigen.

Hierbei ist zu beachten, dass das vorgestellte Zeitschritt-Newton-Verfahren an keine CFL-Bedingung gebunden ist wie etwa explizite Verfahren.

4.4 Lösungsverfahren für lineare Gleichungssysteme

In diesem Abschnitt werden Lösungsverfahren für lineare Gleichungssysteme und spezielle Vorkonditionierungstechniken für Vektorrechner vorgestellt.

4.4.1 Iterative Lösungsverfahren

Im Zeitschritt-Newton-Verfahren muss in jedem Iterationsschritt (= Zeitschritt) ein lineares Gleichungssystem (LGS)

$$\left(V \frac{1}{\Delta t} - \frac{\partial R}{\partial Q} \right) \Delta Q = R(Q^n) \quad (4.14)$$

mit einer schwach besetzten Matrix gelöst werden (vgl. (4.11)). Im Folgenden wird das LGS (4.14) auch mit $Ax = b$ abgekürzt.

Iterative Verfahren sind für große, schwach besetzte lineare Gleichungssysteme, wie sie bei der Diskretisierung partieller Differenzialgleichungen entstehen, wesentlich effizienter als direkte Verfahren. Beim vorgestellten Zeitschritt-Newton-Verfahren ist es außerdem nicht erforderlich, das lineare Gleichungssystem exakt zu lösen. Für die Konvergenz des Newton-Verfahrens reicht es aus, das LGS soweit zu lösen, dass die Bedingung

$$\left\| R(Q^n) - \left(V \frac{1}{\Delta t} - \frac{\partial R}{\partial Q} \right) \Delta Q \right\| \leq \eta \left\| R(Q^n) \right\| \quad (4.15)$$

für ein hinreichend kleines η erfüllt ist. Diese Version wird in der englischsprachigen Literatur als *inexact Newton-method* bezeichnet. Iterative Verfahren berechnen eine Folge von Näherungslösungen des LGS, welche gegen die exakte Lösung konvergiert. Es gibt zwei Klassen von iterativen Lösungsverfahren für lineare Gleichungssysteme:

1. **Stationäre Iterationsverfahren** sind Verfahren der Form

$$x_{n+1} = Mx_n + Nb.$$

Ein stationäres Verfahren ist konvergent, wenn gilt $\|M\| < 1$, wobei $\|\cdot\|$ eine beliebige induzierte Operatornorm ist. Das Verfahren ist konsistent zum linearen Gleichungssystem, wenn die exakte Lösung x^* des LGS Fixpunkt des Verfahrens ist. Es muss also gelten

$$x^* = Mx^* + Nb \quad \text{für} \quad x^* = A^{-1}b.$$

2. **Projektionsverfahren** bestimmen eine Näherungslösung x_k des LGS in einem k -dimensionalen Unterraum $\text{span}\{v_1, v_2, \dots, v_k\}$

$$x_k = x_0 + \sum_{i=1}^k \alpha_i v_i \quad \alpha_i \in \mathbb{R}, \quad v_i \in \mathbb{R}^n.$$

Im ersten Schritt eines Projektionsverfahren werden die Projektionsrichtungen v_i festgelegt. Im zweiten Schritt werden die Koeffizienten α_i bestimmt. Zur Gruppe der Projektionsverfahren gehört die Klasse der Krylov-Raum-Verfahren.

Das Residuum r_k des LGS für eine Näherungslösung x_k wird als

$$r_k = b - Ax_k \tag{4.16}$$

definiert. Ziel eines Iterationsverfahrens ist es, möglichst nahe an die exakte Lösung x^* zu kommen, d.h. die Norm $\|x^* - x_k\|$ soll klein werden. Diese Norm kann jedoch nicht berechnet werden, da die exakte Lösung x^* nicht bekannt ist. Dagegen ist es möglich, die Norm $\|A(x^* - x_k)\|$ zu berechnen:

$$\|A(x^* - x_k)\| = \|Ax^* - Ax_k\| = \|b - Ax_k\| = \|r_k\|. \tag{4.17}$$

Mit dieser Residuennorm kann ein Abbruchkriterium für das iterative Lösungsverfahren definiert werden. Das Iterationsverfahren soll bei Erfüllen der Bedingung

$$\frac{\|r_k\|}{\|r_0\|} < \eta \tag{4.18}$$

für ein hinreichend kleines η beendet werden (vgl. mit (4.15)). Hierbei ist r_0 das Residuum des Startwerts x_0 . Für das hier vorgestellte Verfahren war es ausreichend, η auf 10^{-4} zu setzen. Größere Werte für η verhindern die Konvergenz des Newton-Verfahrens. Zu kleine Werte für η bewirken ein genaueres Lösen des linearen Gleichungssystems und führen daher lediglich zu einer längeren Rechenzeit, während die Konvergenz des Newton-Verfahrens unverändert bleibt.

Um eine Verwechslung mit den Newton-Iterationen zu vermeiden, werden im Folgenden die Iterationen des LGS-Lösungsverfahrens als *Subiterationen* bezeichnet.

4.4.2 Jacobi-Linienrelaxationsverfahren

Als Lösungsverfahren für das lineare Gleichungssystem wurde von Jonas und Schöll in [38] und [75] ein Jacobi-Linienrelaxationsverfahren (JLR) mit einer Blockdiagonalvorkonditionierung und Subiterationen vorgestellt. Als Abbruchkriterium wurde die relative Änderung $\frac{x_{k+1} - x_k}{x_k}$ in einem einzigen Punkt des Strömungsfeldes betrachtet. Zusätzlich wurde die Anzahl der Iterationen auf maximal 8 bis 10 beschränkt. Grundlegende Untersuchungen zu diesem Verfahren wurden bisher allerdings nicht durchgeführt, so dass über die Eigenschaften dieses Verfahrens sowie dessen Effizienz nichts bekannt war.

Voraussetzung für das JLR-Verfahren ist eine Diskretisierung auf strukturierten Gittern und eine blockheptadiagonale Matrixstruktur, wie sie in Abschnitt 4.2 hergeleitet wurde. Für die Herleitung des JLR-Verfahrens wird das lineare Gleichungssystem (4.14) in der Form

$$\left(V \frac{1}{\Delta t} + D + N_i + N_j + N_k \right) x = b \tag{4.19}$$

geschrieben. Hierbei ist $V \frac{1}{\Delta t}$ der instationäre Term des Zeitschritt-Newton-Verfahrens und D die Blockhauptdiagonale der Jacobi-Matrix. Die Matrizen N_i , N_j und N_k bestehen aus jeweils zwei Blocknebendiagonalen, die zu einer Netzrichtung i , j oder k gehören.

Für das JLR-Verfahren wird das lineare Gleichungssystem (4.19) in die drei Netzrichtungen aufgeteilt, indem jeweils die Blocknebendiagonalen zu zwei Netzrichtungen auf die rechte Seite des linearen Gleichungssystems gebracht werden:

$$\left(V \frac{1}{\Delta t} + D + N_k \right) x = b - (N_i + N_j) x \quad (4.20)$$

$$\left(V \frac{1}{\Delta t} + D + N_j \right) x = b - (N_i + N_k) x \quad (4.21)$$

$$\left(V \frac{1}{\Delta t} + D + N_i \right) x = b - (N_j + N_k) x. \quad (4.22)$$

Diese linearen Gleichungssysteme werden mit der Inversen der Blockhauptdiagonalmatrix

$$K = \left(V \frac{1}{\Delta t} + D \right)^{-1} \quad (4.23)$$

multipliziert und nacheinander gelöst. Dazu wird auf der rechten Seite eines LGS die zuvor berechnete Näherungslösung für x eingesetzt. Nach dem Lösen des dritten LGS wird der Vorgang wiederholt. Damit entsteht das im Algorithmus 4.2 zusammengefasste Jacobi-Linienrelaxationsverfahren.

Jacobi-Linienrelaxationsverfahren

do

$$K \left(V \frac{1}{\Delta t} + D + N_k \right) x' = Kb - K (N_i + N_j) x_n \quad (4.24)$$

$$K \left(V \frac{1}{\Delta t} + D + N_j \right) x'' = Kb - K (N_i + N_k) x' \quad (4.25)$$

$$K \left(V \frac{1}{\Delta t} + D + N_i \right) x_{n+1} = Kb - K (N_j + N_k) x'' \quad (4.26)$$

until $\frac{\|r_k\|}{\|r_0\|} < \eta$

Algorithmus 4.2: Jacobi-Linienrelaxationsverfahren mit Subiterationen

Die drei blocktridiagonalen linearen Gleichungssysteme (4.24), (4.25) und (4.26) werden exakt durch Block-LU-Zerlegungen gelöst. Für das erste LGS (4.24) wird die Zellnumerierung im strukturierten Netz von (i, j, k) zu (k, j, i) gewechselt. Ebenso wird für das zweite LGS (4.25) eine Umsortierung zu (j, i, k) vorgenommen. Durch diese Änderung der Zellnumerierung wird erreicht, dass die Blocknebendiagonalen direkt neben der Blockhauptdiagonalen liegen. Damit können die LU-Zerlegungen effizient und speicherplatzsparend berechnet werden, wobei diese Berechnung nur einmal erfolgen muss, da sich während der Subiterationen die Matrizen nicht ändern. Der folgende Satz soll dieses heuristisch hergeleitete Iterationsverfahren etwas erläutern:

Satz 4.1

Das JLR-Verfahren (4.24), (4.25) und (4.26) ist ein stationäres Iterationsverfahren und konsistent zum linearen Gleichungssystem (4.19). Im Falle expliziter Randbedingungen konvergiert

das Verfahren für hinreichend kleine Zeitschritte $\Delta t > 0$. Die Konvergenzrate ist unabhängig von der Matrix K .

Beweis:

1. Stationäres Iterationsverfahren:

Aus den Gleichungen (4.24), (4.25) und (4.26) können die Zwischenwerte x' und x'' eliminiert werden. Nach Auflösen der daraus resultierenden Gleichung nach x_{n+1} entsteht die zum Algorithmus 4.2 äquivalente Formulierung

$$x_{n+1} = (V \frac{1}{\Delta t} + D + N_i)^{-1} (b - (N_j + N_k) (V \frac{1}{\Delta t} + D + N_j)^{-1} \cdot (b - (N_i + N_k) (V \frac{1}{\Delta t} + D + N_k)^{-1} (b - (N_i + N_j) x_n))).$$

Dies entspricht der üblichen Form stationärer Iterationsverfahren

$$x_{n+1} = Mx_n + Nb \tag{4.27}$$

mit

$$M = -(V \frac{1}{\Delta t} + D + N_i)^{-1} (N_j + N_k) (V \frac{1}{\Delta t} + D + N_j)^{-1} (N_i + N_k) (V \frac{1}{\Delta t} + D + N_k)^{-1} (N_i + N_j) \tag{4.28}$$

und

$$Nb = (V \frac{1}{\Delta t} + D + N_i)^{-1} (b - (N_j + N_k) (V \frac{1}{\Delta t} + D + N_j)^{-1} (b - (N_i + N_k) (V \frac{1}{\Delta t} + D + N_k)^{-1} b)) \tag{4.29}$$

bzw.

$$Nb = \left[(V \frac{1}{\Delta t} + D + N_i)^{-1} - (V \frac{1}{\Delta t} + D + N_i)^{-1} (N_j + N_k) (V \frac{1}{\Delta t} + D + N_j)^{-1} + (V \frac{1}{\Delta t} + D + N_i)^{-1} (N_j + N_k) (V \frac{1}{\Delta t} + D + N_j)^{-1} (N_i + N_k) (V \frac{1}{\Delta t} + D + N_k)^{-1} \right] b. \tag{4.30}$$

Die hierbei auftretenden Matrixinvertierungen werden exakt durch LU-Zerlegungen berechnet.

2. Konsistenz:

Sei $x^* = A^{-1}b$ die exakte Lösung von (4.19). Durch Einsetzen von x^* für x_n in Gleichung (4.24)

entsteht

$$K \left(V \frac{1}{\Delta t} + D + N_k \right) x' = Kb - K (N_i + N_j) x^* \quad (4.31)$$

$$\left(V \frac{1}{\Delta t} + D + N_k \right) x' = b - (N_i + N_j) x^* \quad (4.32)$$

$$\begin{aligned} &= b - \left(V \frac{1}{\Delta t} + D + N_i + N_j + N_k \right) x^* \\ &\quad + \left(V \frac{1}{\Delta t} + D + N_k \right) x^* \end{aligned} \quad (4.33)$$

$$\left(V \frac{1}{\Delta t} + D + N_k \right) x' = \left(V \frac{1}{\Delta t} + D + N_k \right) x^* \quad (4.34)$$

$$x' = x^* \quad (4.35)$$

Entsprechend gilt $x'' = x^*$, sowie $x_{n+1} = x^*$. Damit ist die exakte Lösung x^* des LGS ein Fixpunkt des Iterationsverfahrens (4.24), (4.25) und (4.26), womit die Konsistenz bewiesen ist.

3. Konvergenz:

Für die Konvergenzaussage geht der Satz von expliziten Randbedingungen aus. Damit sind die Randgleichungen nicht Bestandteil des LGS.

Das stationäre Verfahren (4.27) ist genau dann konvergent, wenn gilt

$$\|M\| < 1,$$

wobei $\|\cdot\|$ eine beliebige induzierte Operatornorm ist. Mit Hilfe der Dreiecksungleichung, der Neumann-Reihe und der Summenformel für geometrische Reihen erhält man unter der Voraussetzung $\|\Delta t V^{-1}(D + N_k)\| < 1$:

$$\left\| \left(V \frac{1}{\Delta t} + D + N_k \right)^{-1} (N_i + N_j) \right\| = \left\| (E + \Delta t V^{-1}(D + N_k))^{-1} \Delta t V^{-1}(N_i + N_j) \right\| \quad (4.36)$$

$$\leq \left\| (E + \Delta t V^{-1}(D + N_k))^{-1} \right\| \left\| \Delta t V^{-1}(N_i + N_j) \right\| \quad (4.37)$$

$$= \left\| \sum_{n=0}^{\infty} (-\Delta t V^{-1}(D + N_k))^n \right\| \left\| \Delta t V^{-1}(N_i + N_j) \right\| \quad (4.38)$$

$$\leq \left(\sum_{n=0}^{\infty} \Delta t^n \left\| V^{-1}(D + N_k) \right\|^n \right) \left\| \Delta t V^{-1}(N_i + N_j) \right\| \quad (4.39)$$

$$= \frac{1}{1 - \Delta t \left\| V^{-1}(D + N_k) \right\|} \left\| \Delta t V^{-1}(N_i + N_j) \right\|. \quad (4.40)$$

Für hinreichend kleines Δt kann sowohl die Voraussetzung

$$\|\Delta t V^{-1}(D + N_k)\| < 1 \quad (4.41)$$

als auch die Ungleichung

$$\frac{1}{1 - \Delta t \|V^{-1}(D + N_k)\|} \|\Delta t V^{-1}(N_i + N_j)\| < 1 \quad (4.42)$$

für beliebige Matrizen V , D , N_i , N_j und N_k erfüllt werden. Insgesamt kann damit für ein hinreichend kleines Δt erreicht werden, dass gilt

$$\left\| \left(V \frac{1}{\Delta t} + D + N_k \right)^{-1} (N_i + N_j) \right\| < 1 \quad (4.43)$$

$$\left\| \left(V \frac{1}{\Delta t} + D + N_j \right)^{-1} (N_i + N_k) \right\| < 1 \quad (4.44)$$

$$\left\| \left(V \frac{1}{\Delta t} + D + N_i \right)^{-1} (N_j + N_k) \right\| < 1. \quad (4.45)$$

Daraus folgt, dass $\|M\| < 1$ ist, womit die Konvergenz der JLR-Verfahrens für hinreichend kleine Δt gezeigt wurde.

4. Konvergenzrate:

Die linearen Gleichungssysteme (4.24),(4.25) und (4.26) werden exakt durch LU-Zerlegungen gelöst. Daher hat die Multiplikation der Gleichungssysteme mit die Matrix K keinen Einfluss auf die Konvergenz. \square

Der Satz über das JLR-Verfahren bedarf einiger Erläuterungen.

Bemerkung 4.2 Die Konsistenzaussage des Satzes 4.1 besagt, dass das Jacobi-Linienrelaxationsverfahren im Falle einer Konvergenz gegen die exakte Lösung des linearen Gleichungssystems (4.19) konvergiert. Damit hat die im JLR-Verfahren durchgeführte Dimensionsaufspaltung keine Auswirkungen auf die Lösung des LGS und damit auch keine Auswirkungen auf das Newton-Verfahren.

Die von Jonas in [38] eingeführte Vorkonditionierungsmatrix K hat keinen Einfluss auf die Konvergenz des JLR-Verfahrens. Daher ist die Bezeichnung 'Vorkonditionierung' nicht gerechtfertigt. Die Skalierung des linearen Gleichungssystems mit der Matrix K erzeugt auf der Hauptdiagonalen Einheitsmatrizen, welche den Algorithmus für die LU-Zerlegungen der tridiagonalen Matrizen vereinfachen.

Im Satz 4.1 wurde ein sehr allgemeines Konvergenzkriterium angegeben, welches für die Matrizen D , N_i , N_j und N_k keine Voraussetzungen benötigt. Maßgeblich für die Konvergenz des Verfahrens ist jedoch ausschließlich die Bedingung

$$\|M\| < 1.$$

Diese Bedingung kann bei entsprechenden Matrizen D , N_i , N_j und N_k unabhängig von der Wahl des Zeitschrittes Δt immer erfüllt sein, so dass das angegebene Konvergenzkriterium keine Einschränkung des Zeitschrittes darstellen muss. Die Aussage des Satzes soll lediglich verdeutlichen, dass der Zeitschritt für dieses Verfahren eine Bedeutung haben könnte. Untersuchungen dazu werden in Kapitel 6 durchgeführt.

Das JLR-Verfahren wurde im Rahmen dieser Arbeit durch eine Residuenberechnung erweitert, so dass in jedem Iterationsschritt die Residuennorm $\|r_k\| = \|b - Ax_k\|$ berechnet wird. Dies bedeutet zwar zusätzlichen Rechenaufwand, dafür kann aber mit dieser Norm das Abbruchkriterium (4.18) angewendet werden und das Verfahren mit den folgenden Krylov-Raum-Verfahren verglichen werden.

4.4.3 Krylov-Raum-Verfahren

Zu den effizientesten Lösungsverfahren für sehr große, schwach besetzte lineare Gleichungssysteme gehören die Krylov-Raum-Verfahren. Diese Verfahren setzen keine besondere Struktur der Matrix voraus, weshalb sie vor allem in Strömungssimulationsprogrammen eingesetzt werden, welche mit unstrukturierten Netzen rechnen.

Krylov-Raum-Verfahren gehören zur Klasse der Projektionsverfahren. Im Rahmen dieser Arbeit wurden folgende Krylov-Raum-Verfahren ausgewählt:

- TFQMR (Transposed Free Quasi Minimal Residual)
- QMRCGStab (Quasi-Minimal Residual Conjugate Gradients Stabilized)
- CGS (Conjugate Gradients Squared)
- BiCGStab (Bi-Conjugate Gradients Stabilized)
- GMRES(l) (restarted General Minimal Residual)

Die Auswahl erfolgte im Hinblick auf die Weiterentwicklung des Newton-Verfahrens, die im Kapitel 5 beschrieben wird. Es wurden daher nur Krylov-Raum-Verfahren eingesetzt, in denen lediglich Matrix-Vektor-Multiplikationen mit der Matrix A , jedoch keine Matrix-Vektor-Multiplikation mit der transponierten Matrix A^T auftreten. Das CGS-Verfahren wurde bereits von Jonas im zweidimensionalen/axialsymmetrischen Programm getestet, aber aufgrund einer schlechten Vorkonditionierung und der daraus resultierenden schlechten Konvergenz des Verfahrens nicht weiter eingesetzt, [38].

Der Schwerpunkt dieses Kapitels liegt weniger in der Untersuchung von Krylov-Raum-Verfahren als in der Untersuchung von *Vorkonditionierungstechniken* für Krylov-Raum-Verfahren. Aus diesem Grund wird im Folgenden nur das GMRES(l)-Verfahren vorgestellt, welches das robusteste und auf dem Gebiet der Strömungsmechanik am häufigsten eingesetzte Verfahren ist. Dieses Verfahren wird im nächsten Abschnitt detailliert dargestellt. Wegen der umfangreichen Herleitung der Krylov-Raum-Verfahren TFQMR, QMRCGStab, CGS und BiCGStab wird im Rahmen dieser Arbeit auf eine detaillierte Darstellung verzichtet. Eine vollständige Herleitung dieser Verfahren ist z. B. in [58] zu finden. Die in dieser Arbeit eingesetzten Algorithmen für TFQMR, QMRCGStab, CGS und BiCGStab sind im Anhang B aufgelistet.

Restarted General Minimal Residual - GMRES(l)

Das GMRES-Verfahren ist ein iteratives Lösungsverfahren für ein lineares Gleichungssystem $Ax = b$ mit *beliebiger, regulärer* $n \times n$ -Matrix A . Ausgehend von einem beliebigen Startvektor x_0 wird im m -ten Iterationsschritt eine Näherungslösung der Form

$$x_m = x_0 + \lambda_1 v_1 + \dots + \lambda_m v_m \quad \text{mit} \quad \lambda_i \in \mathbb{R}, \quad v_i \in \mathbb{R}^n \quad (4.46)$$

berechnet. Werden die Vektoren v_1, \dots, v_m in einer Matrix V_m und die Skalare $\lambda_1, \dots, \lambda_m$ in einem Vektor $\Lambda_m = (\lambda_1, \dots, \lambda_m)^T$ zusammengefasst, so entsteht aus dem Ansatz (4.46)

$$x_m = x_0 + V_m \Lambda_m. \quad (4.47)$$

Im GMRES-Verfahren bilden die Vektoren v_1, \dots, v_m eine Orthonormalbasis des Krylov-Raums

$$K_m(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}. \quad (4.48)$$

Hierbei ist r_0 das Residuum zum Startvektor x_0 , d. h. $r_0 = b - Ax_0$. Es kann leicht gezeigt werden, dass für die gesuchte Lösung stets gilt

$$x^* \in x_0 + K_m(A, r_0) \quad \text{für hinreichend großes } m.$$

Die Berechnung der Orthonormalbasis erfolgt mit dem im Algorithmus 4.3 zusammengefassten Arnoldi-Verfahren, welches lediglich eine numerisch stabilere Variante des Orthogonalisierungsverfahrens von Gram-Schmidt darstellt.

Arnoldi-Orthogonalisierungsverfahren
$v_1 = \frac{r_0}{\ r_0\ _2}$
for $j = 1, \dots, m$
$\omega_j = Av_j$
for $i = 1, \dots, j - 1$
$h_{i,j} = (\omega_j, v_i)$
$\omega_j = \omega_j - h_{i,j} \cdot v_i$
$h_{j+1,j} = \ \omega_j\ _2$
$v_{j+1} = \frac{\omega_j}{h_{j+1,j}}$

Algorithmus 4.3: Arnoldi-Orthogonalisierungsverfahren

Der Eintrag h_{21} ist somit zu Null geworden. Entsprechend können die Einträge h_{32}, h_{43}, \dots aus der Matrix eliminiert werden. Damit entsteht die QR -Zerlegung von \bar{H}_m

$$\bar{H}_m = Q_m R_m \quad \text{mit} \quad Q_m^T = G_m \cdot G_{m-1} \cdot \dots \cdot G_1. \quad (4.59)$$

Hierbei ist Q_m eine $(m+1) \times (m+1)$ -Orthogonalmatrix. Die $(m+1) \times m$ -Matrix R_m hat in der letzten Zeilen nur Nulleinträge, während die ersten m Zeilen eine obere Dreiecksmatrix darstellen. Damit kann die Residuennorm nach Gleichung (4.55) weiter umgeformt werden:

$$\|r_m\|_2 = \|w - \bar{H}_m \Lambda_m\|_2 \quad (4.60)$$

$$= \|w - Q_m R_m \Lambda_m\|_2 \quad (4.61)$$

$$= \|Q_m(Q_m^T w - R_m \Lambda_m)\|_2 \quad (4.62)$$

$$= \|Q_m^T w - R_m \Lambda_m\|_2 \quad (4.63)$$

Das Minimierungsproblem $\|r_m\|_2 \rightarrow \min!$ bzw.

$$\|Q_m^T w - R_m \Lambda_m\|_2 \rightarrow \min! \quad (4.64)$$

kann nach diesen Umformungen direkt gelöst werden. Eine schematische Darstellung von Gleichung (4.63) für $m = 3$ verdeutlicht den Sachverhalt:

$$\|Q_3^T w - R_3 \Lambda_3\|_2 = \left\| \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \begin{pmatrix} \|r_0\| \\ 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} \right\|_2. \quad (4.65)$$

Für die Minimierung der Residuennorm muss der Vektor $\Lambda_m = (\lambda_1, \dots, \lambda_m)^T$ so gewählt werden, dass die ersten m Zeilen des Terms $Q_m^T w - R_m \Lambda_m$ zu Null werden. Die Bestimmung von Λ_m erfolgt über Rückwärtssubstitution. Dagegen hat der Vektor Λ_m keinen Einfluss auf die letzte Zeile im Term $Q_m^T w - R_m \Lambda_m$. In dieser letzten Zeile steht somit die minimale Residuennorm, welche durch

$$\begin{aligned} \min \|r_m\|_2 &= \min \|Q_m^T w - R_m \Lambda_m\|_2 \\ &= \|r_0\|_2 \cdot |s_1 \cdot s_2 \cdot \dots \cdot s_m| \end{aligned} \quad (4.66)$$

berechnet werden kann, wobei mit s_k die Sinuswerte der Givens-Rotationen bezeichnet sind.

Mit Gleichung (4.66) kann die Residuennorm $\|r_m\|_2$ bestimmt werden, ohne dass zuvor die Näherungslösung x_m berechnet wurde. Dies kann insbesondere bei vielen GMRES-Iterationen dazu führen, dass die mit Gleichung (4.66) berechnete Residuennorm aufgrund von verschiedenen numerischen Problemen ungleich der tatsächlichen Residuennorm $\|b - Ax_k\|_2$ ist. Andererseits hat Gleichung (4.66) genau den Vorteil, dass zur Bestimmung der Residuennorm die Näherungslösung x_m sowie der Vektor Λ_m nicht berechnet werden müssen. Diese Tatsache wird im Algorithmus berücksichtigt: Das GMRES-Verfahren berechnet während der Iterationen keine Näherungslösung x_m . Erst wenn die Residuennorm $\|r_m\|_2$ klein genug ist, wird das Iterationsverfahren beendet und die Näherungslösung x_m berechnet.

GMRES-Verfahren

```

Startwert  $x_0$ 
 $r_0 = b - Ax_0$ 
 $\beta = \|r_0\|_2$ 
 $v_1 = \frac{r_0}{\beta}$ 
 $m = 0$ 
while  $\frac{\|r_m\|}{\|r_0\|} > \epsilon$  do
   $m = m + 1$ 
   $\omega = Av_m$ 
  for  $i = 1, \dots, m$ 
     $h_{i,m} = (\omega, v_i)$ 
     $\omega = \omega - h_{i,j} \cdot v_i$ 
   $h_{m+1,m} = \|\omega\|_2$ 
   $v_{m+1} = \frac{\omega}{h_{m+1,m}}$ 
  berechne QR-Zerlegung  $H_m = Q_m \cdot R_m$ 
   $\|r_m\| = \beta \cdot |s_1 \cdot s_2 \cdot \dots \cdot s_m|$ 
end(while)
bestimme  $\Lambda_m = (\lambda_1, \dots, \lambda_m)$  aus  $R_m \cdot \Lambda_m = \beta Q^T e_1$ 
 $x_j = x_0 + \sum_{i=1}^m \lambda_i \cdot v_i$ 

```

Algorithmus 4.4: Unvorkonditioniertes GMRES-Verfahren

Die Orthogonalisierung der Vektoren v_1, \dots, v_m , die Matrix \bar{H}_m , sowie die QR -Zerlegung müssen nicht in jeder GMRES-Iteration neu berechnet werden. Stattdessen werden in der m -ten Iteration nur die Terme neu berechnet, welche durch den Vektor v_{m+1} neu hinzukommen.

Da im GMRES-Verfahren im m -ten Iterationsschritt die Näherungslösung x_m als

$$x_m = x_0 + \lambda_1 v_1 + \dots + \lambda_m v_m$$

berechnet wird, nimmt der Speicherplatzbedarf in jedem Iterationsschritt um einen weiteren Vektor v_{m+1} zu. Um den Speicherplatzbedarf zu begrenzen ist es daher üblich, das GMRES-Verfahren nach l Iterationen neu zu starten. Dies bedeutet, dass das GMRES(l)-Verfahren nach spätestens l Iterationen die Näherungslösung x_l berechnet. Ist die gewünschte Genauigkeit der

Lösung noch nicht erreicht, so wird das GMRES-Verfahren anschließend wieder gestartet, jedoch mit dem neuen Startvektor x_l . Diese Version des GMRES-Verfahrens wird als *Restarted-GMRES* und l als *restart-level* bezeichnet. Nach einem Neustart wird das Residuum neu berechnet, womit ein möglicher numerischer Fehler aus Gleichung (4.66) korrigiert werden kann. Dies wird bei späteren Untersuchungen im Kapitel 5 eine wesentliche Rolle spielen.

Die Konvergenz des GMRES-Verfahrens ist durch das Minimierungsprinzip sichergestellt, da dadurch das Verfahren zumindest theoretisch nach *endlich* vielen Iterationen die exakte Lösung des linearen Gleichungssystems berechnet. Allerdings ist schon für das Restarted-GMRES-Verfahren die Konvergenz ungewiss. Ebenso lassen sich über die Konvergenzgeschwindigkeit des Verfahrens nur sehr schwer Aussagen treffen. Eine Konvergenzaussage ausschließlich mit Hilfe der Konditionszahl der Matrix A ist für beliebige, reguläre Matrizen nicht möglich.

4.4.4 Vorkonditionierung

Für das hier vorgestellte Zeitschritt-Newton-Verfahren zur Berechnung dreidimensionaler Strömungen konvergieren die Krylov-Raum-Verfahren viel zu schlecht oder überhaupt nicht. Daher müssen zur Konvergenzbeschleunigung Vorkonditionierungstechniken eingesetzt werden. Durch einer Vorkonditionierung des linearen Gleichungssystems wird versucht, den Krylov-Raum $K_m(A, r_0)$ und damit die Projektionsrichtungen v_1, \dots, v_m so zu verändern, dass die Krylov-Raum-Verfahren schneller konvergieren. Anstelle des linearen Gleichungssystems

$$Ax = b$$

wird das vorkonditionierte System

$$P_l^{-1}AP_r^{-1}y = P_l^{-1}b \quad \text{mit} \quad P_r^{-1}y = x \quad (4.67)$$

gelöst. P_l und P_r sind Vorkonditionierungsmatrizen für die linksseitige bzw. rechtsseitige Vorkonditionierung. Durch die Vorkonditionierung verändert sich zwar das lineare Gleichungssystem und damit auch der Krylov-Raum,

$$K_m((P_l^{-1}AP_r^{-1}), r_0) = \{r_0, (P_l^{-1}AP_r^{-1})r_0, (P_l^{-1}AP_r^{-1})^2r_0, \dots, (P_l^{-1}AP_r^{-1})^{m-1}r_0\}, \quad (4.68)$$

die Lösung des LGS bleibt jedoch unverändert. In Gleichung (4.68) ist r_0 das Anfangsresiduum des vorkonditionierten Systems, also

$$r_0 = P_l^{-1}b - P_l^{-1}AP_r^{-1}y_0. \quad (4.69)$$

In dieser Arbeit wird ausschließlich die linksseitige Vorkonditionierung eingesetzt, es gilt also stets $P_r = E$. Daher wird im Folgenden anstatt P_l nur noch P geschrieben. Die Matrizen-Multiplikation $P^{-1}A$ im linksseitig vorkonditionierten LGS

$$P^{-1}Ax = P^{-1}b \quad (4.70)$$

muss nicht explizit berechnet werden. Sämtliche in dieser Arbeit untersuchten Krylov-Raum-Verfahren benötigen die Matrix $P^{-1}A$ des LGS nur in Form von Matrix-Vektor-Multiplikationen. Diese können schrittweise erfolgen:

$$P^{-1}A \cdot v = P^{-1} \cdot (Av) = P^{-1}v'. \quad (4.71)$$

Anstelle *einer* Matrix-Vektor-Multiplikation werden nacheinander *zwei* Matrix-Vektor-Multiplikationen durchgeführt. Bei manchen Vorkonditionierern stellt $P^{-1}v'$ keine einfache Matrix-Vektor-Multiplikation dar, da die Matrix P^{-1} nicht explizit gegeben ist. Oft liegt nur die Matrix P und nicht die Matrix P^{-1} vor. Auf diesen Punkt wird später bei den einzelnen Vorkonditionieren nochmals eingegangen.

Die Entwicklung einer Vorkonditionierungsmatrix P stellt immer einen Kompromiss zwischen Effektivität und Rechenaufwand dar. Die effektivste Vorkonditionierung wäre

$$P^{-1} = A^{-1},$$

was natürlich aufgrund der notwendigen Rechenzeit indiskutabel ist und schon die Lösung des LGS ergäbe. Eine Vorkonditionierung P stellt stets einen Kompromiss aus den folgenden vier Punkten dar:

- Zum einen sollte die Vorkonditionierungsmatrix P^{-1} die Matrix A^{-1} möglichst gut approximieren.
- Außerdem sollte die Matrix P bzw. P^{-1} wenig Speicherplatz benötigen.
- Die Matrix P bzw. P^{-1} sollte schnell berechenbar sein.
- Der Term $P^{-1} \cdot v$ sollte ebenfalls schnell berechenbar sein.

Im Rahmen dieser Arbeit wurden folgende Vorkonditionierer getestet:

- Blockdiagonalskalierung
- LUSGS (Lower Upper Symmetric Gauss-Seidel)
- ILU (Incomplete Lower Upper Factorization)
- LLU (Line Lower Upper Factorization)
- Jacobi-Linienrelaxationsverfahren

Diese Vorkonditionierer werden im Folgenden hergeleitet.

Blockdiagonalvorkonditionierung

Die einfachste Vorkonditionierungsmatrix ist die Blockhauptdiagonale

$$P_{BD} = \left(V \frac{1}{\Delta t} + D \right). \quad (4.72)$$

Diese Vorkonditionierung wurde aus dem JLR-Verfahren übernommen, siehe Gleichung (4.23). Für das JLR-Verfahren war es notwendig, das Produkt $P_{BD}^{-1}A$ explizit zu berechnen. Diese explizite Multiplikation der Vorkonditionierungsmatrix P_{BD}^{-1} mit der Matrix A wurde hier für die Krylov-Raum-Verfahren beibehalten.

LUSGS-Vorkonditionierer

Der LUSGS-Vorkonditionierer (*Lower-Upper Symmetric Gauss-Seidel*) wird aus dem symmetrischen Gauss-Seidel-Verfahren abgeleitet. Für eine Zerlegung der Matrix A in die Form

$$A = L + D + U, \quad (4.73)$$

wobei L eine untere und U eine obere Dreiecksmatrix, sowie D eine (Block-)Diagonalmatrix ist, lautet das Gauss-Seidel-Verfahren

$$(L + D)x_{n+1} = b - Ux_n. \quad (4.74)$$

Das dazu symmetrische Verfahren entsteht durch Vertauschen von L und U :

$$(D + U)x_{n+1} = b - Lx_n. \quad (4.75)$$

Werden die beiden Versionen (4.74) und (4.75) abwechselnd eingesetzt, so entsteht das symmetrische Gauss-Seidel-Verfahren, welches sich als

$$x_{n+1} = (D + U)^{-1}L(L + D)^{-1}Ux_n + (D + U)^{-1}D(L + D)^{-1}b \quad (4.76)$$

schreiben lässt. Dies ist wiederum die klassische Form eines stationären Iterationsverfahren $x_{n+1} = Mx_n + Nb$. Die Matrix N eignet sich als Vorkonditionierungsmatrix für ein Krylov-Raum-Verfahren, woraus die Vorkonditionierung

$$P_{LUSGS} = (L + D)D^{-1}(D + U) \quad (4.77)$$

entsteht. Gelegentlich wird diese Vorkonditionierung in der Literatur auch als ILU-Vorkonditionierung bezeichnet, da durch einfaches Umformen die Matrix

$$P_{LUSGS} = (LD^{-1} + E)(D + U)$$

die Gestalt einer unvollständigen LU-Zerlegung hat. Dies entspricht jedoch nicht der sonst üblichen ILU-Vorkonditionierung, die mit der Gauss-Elimination berechnet wird.

Für den Einsatz in einem Krylov-Raum-Verfahren muss das Matrix-Vektor-Produkt

$$x = P_{LUSGS}^{-1} \cdot v$$

berechnet werden. Dies erfolgt in zwei Schritten über Vorwärts- und Rückwärtssubstitution

$$(L + D)v' = v \quad (4.78)$$

$$(D + U)x = Dv'. \quad (4.79)$$

ILU-Vorkonditionierung

Eine Zerlegung einer $n \times n$ -Matrix A als Produkt zweier Matrizen L und U ,

$$A = LU, \tag{4.80}$$

wobei die Matrix L eine untere (*lower*) Dreiecksmatrix und U eine obere (*upper*) Dreiecksmatrix ist, heißt LU -Zerlegung. Wird zusätzlich vorausgesetzt, dass $L_{ii} = 1$ ist,

$$L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ * & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ * & \cdots & * & 1 \end{pmatrix} \quad U = \begin{pmatrix} * & \cdots & \cdots & * \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * \end{pmatrix},$$

so sind durch diese Vorgaben die Matrizen L und U eindeutig bestimmt. Gleichung (4.80) kann geschrieben werden als

$$A_{ij} = \sum_{m=1}^n L_{im}U_{mj} = \sum_{m=1}^{\min(i,j)} L_{im}U_{mj} \quad (n = \text{Dimension der Matrix}). \tag{4.81}$$

Daraus ist es möglich, die Einträge L_{ij} und U_{ij} zu ermitteln:

$$L_{ij} = \left(A_{ij} - \sum_{m=1}^{j-1} L_{im}U_{mj} \right) U_{jj}^{-1} \tag{4.82}$$

$$U_{ij} = A_{ij} - \sum_{m=1}^{i-1} L_{im}U_{m,j}. \tag{4.83}$$

In dieser Arbeit wird die Berechnung der LU -Zerlegung zeilenweise durchgeführt. Da die Matrix des Zeitschritt-Newton-Verfahren eine Blockstruktur hat, werden die Einträge A_{ij} , L_{ij} und U_{ij} aus den Gleichungen (4.82) und (4.83) als Blöcke einer Blockmatrix betrachtet. Somit entsteht eine Block- LU -Zerlegung. Der Algorithmus zur Berechnung der vollständigen Block- LU -Zerlegung ist in Algorithmus 4.5 zusammengefasst.

Selbst für eine dünnbesetzte Matrix A ist die Berechnung der LU -Zerlegung aufgrund des enormen Speicherplatzbedarfs praktisch nicht durchführbar. Aus diesem Grund wird die LU -Zerlegung nur approximativ berechnet. Bei der unvollständigen (*incomplete*) LU -Zerlegung (ILU) werden nur die Einträge L_{ij} bzw. U_{ij} berechnet, für die auch A_{ij} ungleich Null ist. Damit wird das Besetzungsschema der Matrix A auf die Matrizen L und U übertragen. Durch diese Approximation wird der Speicherplatzbedarf deutlich reduziert. Wird mit M_A die Menge aller Indizes (i, j) bezeichnet, für die $A_{ij} \neq 0$ ist,

$$M_A(0) = \{(i, j) | A_{ij} \neq 0\}, \tag{4.84}$$

so kann der LU -Algorithmus 4.5 zum ILU-Algorithmus 4.6 abgewandelt werden. Für strukturierte Netze kann die Menge M_A exakt angegeben werden, wodurch sich der Algorithmus

Block-LU-Zerlegung

```

do  $i = 1, n$ 
  do  $j = 1, i - 1$ 
    
$$L_{ij} = \left( A_{ij} - \sum_{m=1}^{j-1} L_{im}U_{mj} \right) U_{jj}^{-1}$$

  enddo
  do  $j = i, n$ 
    
$$U_{ij} = A_{ij} - \sum_{m=1}^{i-1} L_{im}U_{mj}$$

  enddo
enddo
    
```

Algorithmus 4.5: Block-LU-Zerlegung

vereinfacht. Die Matrix A besteht aufgrund der strukturierten 3D-Netze aus sieben Blockdiagonalen. Damit bestehen die Matrizen L und U der ILU-Zerlegung jeweils aus vier Blockdiagonalen, wobei die Hauptdiagonale von L aus Einheitsmatrizen besteht. Desweiteren entspricht die Indizierung der Blöcke in der Matrix A der Indizierung im strukturierten Netz. Daher ist jeder Block durch drei Indizes (i, j, k) bestimmt. Somit müsste ein Matrixeintrag A_{ij} wie folgt indiziert werden:

$$A_{ij} = A_{(i_i, j_i, k_i), (i_j, j_j, k_j)}.$$

Ein Matrixeintrag A_{ij} ist genau dann ungleich Null, wenn die zugehörigen Zellen (i_i, j_i, k_i) und (i_j, j_j, k_j) im Netz benachbarte Zellen sind, d. h. es muss gelten

$$|i_i - i_j| + |j_i - j_j| + |k_i - k_j| \leq 1. \tag{4.85}$$

Damit ist die Indexmenge M_A bestimmt. Das Produkt $L_{im}U_{mj}$ im Algorithmus 4.6 muss nur berechnet werden, wenn $(i, m) \in M_A$ und $(m, j) \in M_A$, es muss also gelten:

$$|i_i - i_m| + |j_i - j_m| + |k_i - k_m| \leq 1 \tag{4.86}$$

$$|i_m - i_j| + |j_m - j_j| + |k_m - k_j| \leq 1. \tag{4.87}$$

Aus den Bedingungen (4.86) und (4.87) und unter der Berücksichtigung der Summationsgrenzen kann der Alogrithmus 4.6 deutlich vereinfacht werden. Die Berechnung der daraus resultierenden ILU-Zerlegung für strukturierte Netze erfolgt mit dem im Algorithmus 4.7 zusammengefassten Verfahren. Wie im Algorithmus 4.7 zu sehen ist, sind die Matrizen $U_{(i,j,k),(i+1,j,k)}$, $U_{(i,j,k),(i,j+1,k)}$ und $U_{(i,j,k),(i,j,k+1)}$ nur Kopien der oberen drei Blockdiagonalen von A . Da diese Matrizen nicht

Algorithmus: ILU-Zerlegung, (zeilenweise)

```

do  $i = 1, n$ 
  do  $j = 1, i - 1$ 
    if  $(i, j) \in M_A$ 
      
$$L_{ij} = \left( A_{ij} - \sum_{\substack{m=1 \\ (i,m),(m,j) \in M_A}}^{j-1} L_{im}U_{mj} \right) U_{jj}^{-1}$$

    endif
  enddo
  do  $j = i, n$ 
    if  $(i, j) \in M_A$ 
      
$$U_{ij} = A_{ij} - \sum_{\substack{m=1 \\ (i,m),(m,j) \in M_A}}^{i-1} L_{im}U_{mj}$$

    endif
  enddo
enddo

```

Algorithmus 4.6: ILU-Zerlegung

gesondert gespeichert werden brauchen, benötigt die ILU-Zerlegung nur Speicherplatz für vier Blockdiagonalen.

Die somit erhaltene ILU-Zerlegung wird als Vorkonditionierung

$$P_{ILU} = \tilde{L}\tilde{U} \quad (4.88)$$

in einem Krylov-Raum-Verfahren eingesetzt. Dazu muss das Produkt $P_{ILU}^{-1} \cdot v$ berechnet werden. Da die Matrizen \tilde{L} und \tilde{U} Dreiecksmatrizen sind, erfolgt die Berechnung von $P_{ILU}^{-1} \cdot v$ durch Vorwärts- und Rückwärtssubstitution.

LLU-Vorkonditionierung

Für die Herleitung des LUSGS-Vorkonditionierers wurde die Matrix A als Summe $L + D + U$ geschrieben. Wird dagegen die Matrix entsprechend der Zerlegung im JLR-Verfahren zerlegt

$$A = V \frac{1}{\Delta t} + D + N_i + N_j + N_k, \quad (4.89)$$

Block-ILU-Zerlegung für strukturierte Netze
do $k = 0, n_k + 1$
do $j = 0, n_j + 1$
do $i = 0, n_i + 1$

$$L_{(i,j,k),(i,j,k-1)} = A_{(i,j,k),(i,j,k-1)} \cdot U_{(i,j,k-1),(i,j,k-1)}^{-1}$$

$$L_{(i,j,k),(i,j-1,k)} = A_{(i,j,k),(i,j-1,k)} \cdot U_{(i,j-1,k),(i,j-1,k)}^{-1}$$

$$L_{(i,j,k),(i-1,j,k)} = A_{(i,j,k),(i-1,j,k)} \cdot U_{(i-1,j,k),(i-1,j,k)}^{-1}$$

$$L_{(i,j,k),(i,j,k)} = E$$

$$\begin{aligned} U_{(i,j,k),(i,j,k)} &= A_{(i,j,k),(i,j,k)} - L_{(i,j,k),(i,j,k-1)} \cdot U_{(i,j,k-1),(i,j,k)} \\ &\quad - L_{(i,j,k),(i,j-1,k)} \cdot U_{(i,j-1,k),(i,j,k)} \\ &\quad - L_{(i,j,k),(i-1,j,k)} \cdot U_{(i-1,j,k),(i,j,k)} \end{aligned}$$

$$U_{(i,j,k),(i+1,j,k)} = A_{(i,j,k),(i+1,j,k)}$$

$$U_{(i,j,k),(i,j+1,k)} = A_{(i,j,k),(i,j+1,k)}$$

$$U_{(i,j,k),(i,j,k+1)} = A_{(i,j,k),(i,j,k+1)}$$

enddo
enddo
enddo

Algorithmus 4.7: Block-ILU-Zerlegung für strukturierte Netze

so kann in Anlehnung an den LUSGS-Vorkonditionierer eine Linien-LU-Vorkonditionierung definiert werden:

$$P_{LLU} = (V \frac{1}{\Delta t} + D + N_k)(V \frac{1}{\Delta t} + D)^{-1}(V \frac{1}{\Delta t} + D + N_j)(V \frac{1}{\Delta t} + D)^{-1}(V \frac{1}{\Delta t} + D + N_i). \quad (4.90)$$

Für diesen Vorkonditionierer müssen die Matrix-Vektor-Produkte

$$(V \frac{1}{\Delta t} + D + N_k)^{-1}v \quad (V \frac{1}{\Delta t} + D + N_j)^{-1}v \quad (V \frac{1}{\Delta t} + D + N_i)^{-1}v \quad (4.91)$$

berechnet werden. Diese Produkte werden analog zum JLR-Verfahren über exakte LU-Zerlegungen der blocktridiagonalen Matrizen berechnet.

JLR-Vorkonditionierung

Als Vorkonditionierung für ein Krylov-Raum-Verfahren kann auch ein iteratives Lösungsverfahren benutzt werden. Im Vorkonditionierungsschritt muss eine Matrix-Vektor-Multiplikation

$$P^{-1}v$$

berechnet werden. Diese Multiplikation kann als lineares Gleichungssystem interpretiert werden:

$$Px = v. \tag{4.92}$$

Da die Vorkonditionierungsmatrix P^{-1} möglichst der Matrix A^{-1} entsprechen sollte, entspricht der Vektor x in Gleichung (4.92) einer Näherungslösung des linearen Gleichungssystems $Ax = v$. Daher kann anstatt von Gleichung (4.92) das lineare Gleichungssystem $Ax = v$ näherungsweise durch ein paar Iterationsschritte eines iterativen Lösungsverfahrens gelöst werden. Als Iterationsverfahren kann das zuvor beschriebene JLR-Verfahren benutzt werden. Hierfür wird die Anzahl der durchzuführenden JLR-Iterationen festgesetzt. Im Folgenden bedeutet GMRES(10) mit einer JLR(2)-Vorkonditionierung ein GMRES-Verfahren mit *restart-level* 10 und einer Vorkonditionierung mit 2 JLR-Iterationen. JLR(0) bedeutet eine Vorkonditionierung mit der Matrix N aus Gleichung (4.30).

Die vorgestellten Vorkonditionierer werden in Kapitel 6 miteinander verglichen.

4.4.5 Vektorisierbarkeit der Gleichungslöser

Das in dieser Arbeit vorgestellte Strömungssimulationsverfahren soll auf Vektorrechnern zum Einsatz kommen. Vektorprozessoren können jedoch nur ihre volle Rechenleistung bei vektorisierbaren Algorithmen erbringen. Das JLR-Verfahren aus Abschnitt 4.4.2 wurde schon durch Umkopieren der tridiagonalen Gleichungssysteme in eine gut vektorisierbare Form gebracht. Die Krylov-Raum-Verfahren bestehen im Wesentlichen nur aus Vektoradditionen, Skalarprodukten und Matrix-Vektor-Multiplikationen. Bei diesen Rechenoperationen treten bzgl. Vektorisierbarkeit ebenfalls keine Schwierigkeiten auf, so dass diese Verfahren die Rechenleistung eines Vektorrechners hervorragend ausnutzen können. Die im Abschnitt 4.4.4 erläuterten Blockdiagonalskalierung ist ebenso vektorisierbar. Die LLU- und JLR-Vorkonditionierung wurden aus dem JLR-Verfahren abgeleitet, welches wie bereits erwähnt in einer vektorisierbaren Form entwickelt wurde.

Die LUSGS-, sowie die ILU-Vorkonditionierung sind dagegen *rekursive* Algorithmen, welche in der angegebenen Form nicht vektorisierbar sind. Aus diesem Grund müssen diese Vorkonditionierer speziell für Vektorrechner optimiert werden. Beim LUSGS-Vorkonditionierer besteht das Problem in der Vorwärts- und Rückwärtssubstitution (4.78) und (4.79), während die Berechnung der Vorkonditionierung vektorisierbar ist. Im Gegensatz dazu sind jedoch bei der ILU-Vorkonditionierung sowohl die Berechnung der ILU-Zerlegung als auch die Vorwärts- und Rückwärtssubstitution nicht vektorisierbar.

Im Folgenden werden verschiedene Möglichkeiten untersucht, die Vektorisierbarkeit der LUSGS- und der ILU-Vorkonditionierung zu verbessern.

Neumann-Reihe für die Vorwärts- und Rückwärtssubstitution

Eine in der Literatur oft vorgeschlagene Möglichkeit, die Vorwärts- und Rückwärtssubstitution in eine vektorisierbare Form zu bekommen, ist die Anwendung der Neumann-Reihe. Für eine

strikte obere Dreiecksmatrix L lautet die Neumann-Reihe für die Matrix $(E - L)^{-1}$

$$(E - L)^{-1} = E + L + L^2 + L^3 + L^4 + \dots \quad (4.93)$$

Da eine strikte obere Dreiecksmatrix nilpotent ist, d.h. es gibt ein k mit $L^k = 0$, ist in diesem Fall die Neumann-Reihe endlich. Das Produkt $(E - L)^{-1}v$ kann approximiert werden durch:

$$(E - L)^{-1}v \approx \sum_{m=0}^l L^m v, \quad (4.94)$$

wobei l nicht dem Nilpotenzgrad k entsprechen muss, sondern kleiner gewählt werden kann. Durch Anwenden der Neumann-Reihe können somit die Vorwärts- und Rückwärtssubstitutionen durch vektorisierbare Matrix-Vektor-Multiplikationen ersetzt werden. Hierbei ist jedoch zu beachten, dass es sich um approximative Berechnungen der Vorwärts- und Rückwärtssubstitutionen handelt, d.h. dass die Vorkonditionierung durch das Anwenden der Neumann-Reihe verändert wurde, was sich auf die Konvergenz des Krylov-Raum-Verfahrens auswirken kann.

Approximierte ILU-Berechnung

Die Berechnung der ILU-Zerlegung nach Algorithmus 4.7 ist nicht vektorisierbar, da der Algorithmus in der Berechnung der U_{ii} eine Rekursion enthält. Eine Möglichkeit, diesen Algorithmus in eine vektorisierbare Form zu bringen, ist die Vereinfachung des Algorithmus. Bei der Berechnung der Hauptdiagonalen

$$U_{ii} = A_{ii} - \sum A_{im} U_{mm}^{-1} A_{mi} \quad (4.95)$$

kann die Summe vernachlässigt werden:

$$U_{ii} = A_{ii}. \quad (4.96)$$

Die dadurch entstandene Vorkonditionierungsmatrix entspricht exakt der LUSGS-Vorkonditionierung, die im Abschnitt 4.4.4 beschrieben wurde.

Iterative ILU-Berechnung

Ein weiterer, in dieser Arbeit erstmals durchgeführter Versuch, eine nichtrekursive Berechnung der ILU-Zerlegung zu entwickeln, besteht aus der iterativen Berechnung. Die Matrizen U_{ii} werden durch

$$U_{ii} = A_{ii} - \sum A_{im} U_{mm}^{-1} A_{mi}$$

berechnet. Diese Formel wird nun als Iterationsansatz verwendet: im n -ten Iterationsschritt werden auf der rechten Seite die U_{ii} vom Iterationsschritt $n - 1$ eingesetzt und damit die U_{ii} neu berechnet:

$$(U^n)_{ii} = A_{ii} - \sum A_{im} (U^{n-1})_{mm}^{-1} A_{mi}. \quad (4.97)$$

Zwar ist der Rechenaufwand *pro Iterationsschritt* genauso groß wie für die gesamte ILU-Berechnung nach Algorithmus 4.7, aber dafür ist diese Berechnung vektorisierbar.

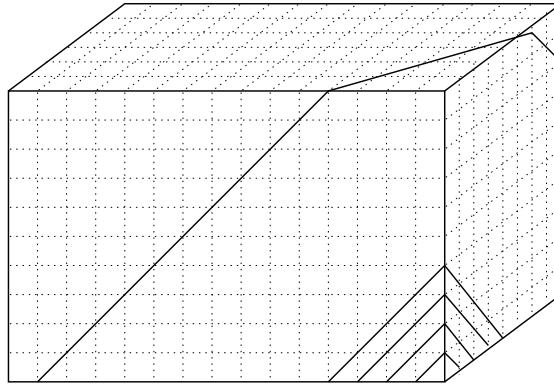


Abbildung 4.1: Strukturiertes Netz mit Hyperebenen.
Jede Hyperebene ist durch $i + j + k = \text{const.}$ bestimmt.

Hyperebenenformat

Da das hier vorgestellte Verfahren auf strukturierten Netzen rechnet, können die Vorwärts- und Rückwärtssubstitution durch Ändern der Berechnungsreihenfolge zu vektorisierbaren Algorithmen umgeschrieben werden. Die Vorwärtssubstitution beginnt in der Ecke $(0,0,0)$ des strukturierten Netzes. Ist dieser Eckpunkt berechnet, so können die drei Nachbarpunkte $(1,0,0)$, $(0,1,0)$ und $(0,0,1)$ gleichzeitig berechnet werden, da diese nicht rekursiv voneinander abhängen. Anschließend können die Punkte $(2,0,0)$, $(0,2,0)$, $(0,0,2)$, $(1,1,0)$, $(1,0,1)$ und $(0,1,1)$, die alle in einer Ebene liegen, gleichzeitig berechnet werden. Durchläuft die Ebene den ganzen Block, so können sämtliche Punkte, die auf einer solchen Hyperebene (HE) liegen, durch einen vektorisierbaren Code berechnet werden. Sämtliche Punkte (i,j,k) , die auf einer solchen Hyperebene liegen, können über

$$i + j + k = \text{konstant}$$

bestimmt werden.

Um den ILU-Algorithmus in diese Form bringen zu können, wird Speicherplatz und Rechenzeit benötigt. Der zusätzliche Speicherplatz wird benötigt, um die Vorkonditionierungsmatrizen L und U in das Hyperebenenformat umsortieren zu können. Bisher brauchten die Matrizen $U_{(i,j,k),(i+1,j,k)}$, $U_{(i,j,k),(i,j+1,k)}$ und $U_{(i,j,k),(i,j,k+1)}$ nicht gesondert abgespeichert zu werden, da sie mit Teilen der Jacobi-Matrix übereinstimmen. Für die vektorisierbare ILU-Version müssen jedoch diese Matrizen im Hyperebenenformat vorliegen und damit gesondert gespeichert werden.

Hinzu kommt, dass auch die Vektoren im Krylov-Raum-Verfahren nicht im richtigen Datenformat vorliegen. Daher muss vor jeder Multiplikation eines Vektors mit der Vorkonditionierungsmatrix dieser erst einmal in das Hyperebenenformat umkopiert werden, was zusätzlich Rechenzeit benötigt.

Mit dieser beschriebenen Methode ist es möglich, sowohl die Berechnung der ILU-Zerlegung, als auch die Vorwärts- und Rückwärtssubstitution bei der LUSGS- und bei der ILU-Vorkonditionie-

rung in eine vektorisierbare Form zu bringen. Hierbei ist zu beachten, dass durch den Wechsel in das Hyperebenenformat am ILU-Algorithmus und an den Vorwärts- und Rückwärtssubstitution lediglich die Berechnungsreihenfolge geändert wird. Dieser Wechsel hat keine Auswirkungen auf die Algorithmen und damit auch keine Auswirkungen auf die Konvergenz der Krylov-Raum-Verfahren.

Kapitel 5

Untersuchungen zur Konvergenzbeschleunigung

Im Kapitel 4 wurde ein Zeitschritt-Newton-Verfahren mit approximierter Jacobi-Matrix, sowie verschiedene iterative Lösungsverfahren für lineare Gleichungssysteme vorgestellt. Das beschriebene Verfahren ist sehr robust und konvergiert bei sachgemäßer Netzauflösung sogar für komplexe Hyperschallströmungen im thermochemischen Nichtgleichgewicht bis auf Maschinengenauigkeit. Beispiele für komplexe Strömungssimulationen sind in Kapitel 6 dargestellt.

Das vorgestellte Verfahren ist soweit ausgereift, dass große Anwendungsrechnungen durchgeführt werden können (siehe Abschnitt 6.5). Mit zunehmender Komplexität der Strömung und zunehmender Netzgröße in Strömungssimulationen für dreidimensionale Flugkörper steigt jedoch entsprechend auch die Gesamtrechenzeit an. Daher soll in diesem Kapitel untersucht werden, ob es Möglichkeiten gibt, das Zeitschritt-Newton-Verfahren zu beschleunigen.

5.1 Inkonsistentes Newton-Verfahren

Das in Kapitel 4 vorgestellte Zeitschritt-Newton-Verfahren benutzt für die Berechnung der Jacobi-Matrix nur die reibungsfreien Flüsse mit räumlicher Genauigkeit erster Ordnung, während die Strömungslösung mit einer Genauigkeit zweiter Ordnung berechnet wird. Ebenso wird bei den viskosen Flüssen die Jacobi-Matrix nur approximativ berechnet. Damit ist die Jacobi-Matrix inkonsistent zur Strömungsmodellierung.

Wird im vorgestellten Verfahren eine Strömung mit einer räumlichen Genauigkeit erster Ordnung berechnet, so entsteht ein *konsistentes* Zeitschritt-Newton-Verfahren, da für diesen Fall die Jacobi-Matrix exakt berechnet wird. Für eine reibungsfreie Zylinder-Vorkörperströmung mit $Ma_\infty = 2,5$ und den Zuströmdaten aus Tabelle 5.1 ist der Konvergenzverlauf des Verfahrens in Abbildung 5.1 zu sehen. Das konsistente Verfahren lässt in der Konvergenzphase einen Zeitschritt mit $CFL=10^6$ und $\omega = 1,0$ zu.

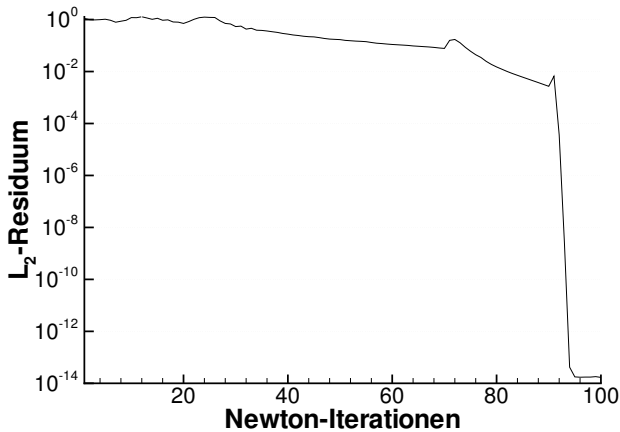


Abbildung 5.1: Konvergenzverlauf für eine reibungsfreie Zylinder-Vorkörperströmung, AUSMDV mit räumlicher Genauigkeit erster Ordnung, Relaxationsparameter $\omega = 1, 0$, maximale CFL-Zahl 10^6

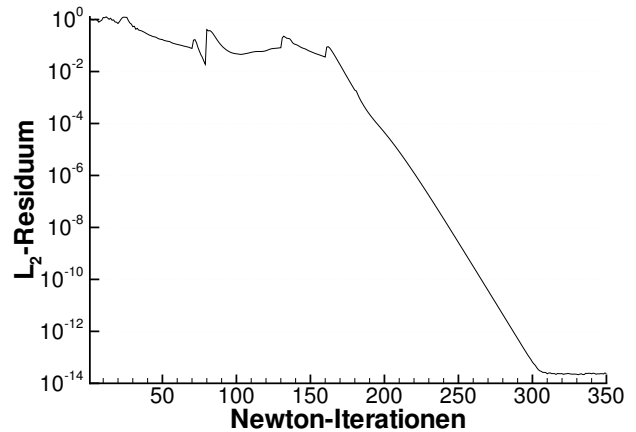


Abbildung 5.2: Konvergenzverlauf für eine reibungsfreie Zylinder-Vorkörperströmung, AUSMDV mit räumlicher Genauigkeit zweiter Ordnung, Relaxationsparameter $\omega = 0, 8$, maximale CFL-Zahl 10^6

Wird dagegen die Strömung mit einer räumlichen Genauigkeit zweiter Ordnung berechnet, so verlangsamt sich trotz hoher CFL-Zahl die Konvergenz deutlich, wie in Abbildung 5.2 zu sehen ist. Außerdem muss ein Relaxationsparameter kleinergleich 0,9 gewählt werden, um überhaupt die stationäre Lösung berechnen zu können.

Zuströmdruck	2490 Pa
Zuströmmachzahl	2.5
Partialdichte N_2	$2,984 \cdot 10^{-2} \text{ kg/m}^3$
Partialdichte O_2	$9,152 \cdot 10^{-3} \text{ kg/m}^3$
Partialdichte NO	$4,054 \cdot 10^{-14} \text{ kg/m}^3$
Partialdichte N	$1,893 \cdot 10^{-14} \text{ kg/m}^3$
Partialdichte O	$2,161 \cdot 10^{-14} \text{ kg/m}^3$
Vibrationstemperatur von N_2	221,7 K
Vibrationstemperatur von O_2	221,7 K
Vibrationstemperatur von NO	221,7 K

Tabelle 5.1: Zuströmdaten für reibungsfreie Zylinderströmung

Daher liegt die Vermutung nahe, dass der Grund für die schlechtere Konvergenz die Approximationen in der Jacobi-Matrix sind. Um das Verfahren zu beschleunigen, wird in diesem Abschnitt versucht, aus dem inkonsistenten Zeitschritt-Newton-Verfahren ein konsistentes Verfahren zu erstellen. Da jedoch das exakte, explizite Berechnen der Jacobi-Matrix zu viel Rechenzeit und Speicherplatz benötigen würde, wird hier ein Newton-Krylov-Verfahren angestrebt, welches durch Kombination des Newton-Verfahrens mit den Krylov-Raum-Verfahren das explizite Berechnen der Jacobi-Matrix umgeht.

5.2 Konsistentes Newton-Verfahren

Das Newton-Verfahren stellt ein LGS

$$\left(V \frac{1}{\Delta t} - \frac{\partial R}{\partial Q} \right) \Delta Q = R(Q^n) \quad (5.1)$$

auf, dessen Matrix im wesentlichen aus der Jacobi-Matrix $\frac{\partial R}{\partial Q}$ besteht. Die implementierten Krylov-Raum-Verfahren benötigen jedoch die Matrix nur in Form von *Matrix-Vektor-Multiplikationen* (MVM). Für die Weiterentwicklung wird die Matrix aufgeteilt, wodurch sich eine MVM schreiben lässt als

$$\left(V \frac{1}{\Delta t} - \frac{\partial R}{\partial Q} \right) \cdot v = V \frac{1}{\Delta t} \cdot v - \frac{\partial \varphi}{\partial Q} \cdot v + \frac{\partial \varphi_v}{\partial Q} \cdot v + \frac{\partial s}{\partial Q} \cdot v. \quad (5.2)$$

Mit φ werden die reibungsfreien Flüsse, mit φ_v die viskosen Flüsse und mit s die Quellterme bezeichnet. Der Vektor v wird vom Krylov-Raum-Verfahren bestimmt.

Das Produkt $V \frac{1}{\Delta t} \cdot v$ kann problemlos exakt berechnet werden, da die Matrix $V \frac{1}{\Delta t}$ eine Diagonalmatrix ist und somit sehr wenig Speicherplatz benötigt. Da die Quellterme s stark nichtlinear sind, können sie mit den folgenden Methoden nicht behandelt werden. Andererseits besteht die Quellterm-Jacobi-Matrix $\frac{\partial s}{\partial Q}$ nur aus einer Blockdiagonalen. Diese kann daher bei einem noch akzeptablen zusätzlichen Speicherplatzbedarf gesondert abgespeichert werden.

Im Folgenden werden Methoden dargestellt und hergeleitet, wie die Matrix-Vektor-Multiplikationen $\frac{\partial \varphi}{\partial Q} \cdot v$ und $\frac{\partial \varphi_v}{\partial Q} \cdot v$ konsistenter, d. h. mit weniger oder gar keinen Approximationen, berechnet werden können. *Ziel der Weiterentwicklung ist ein konsistentes Newton-Krylov-Verfahren mit einer quadratischen Konvergenz.* Dazu muss es möglich sein, in der Konvergenzphase im Algorithmus 4.1 den Zeitschritt Δt beliebig zu erhöhen und den Relaxationsparameter ω auf 1,0 zu setzen.

Da sich jedoch bei der Entwicklung dieser konsistenten Verfahren sehr große Probleme einstellen, wird im Folgenden untersucht, ob die quadratische Konvergenz *ungeachtet des Rechenzeit- und Speicherplatzbedarfs* erreicht werden kann. Eine Bewertung der Verfahren erfolgt im Kapitel 6.

5.3 Matrix-Vektor-Multiplikationen

5.3.1 Matrixfreies Newton-Krylov-Verfahren

Die Grundidee des klassischen Newton-Krylov-Verfahrens ist der Finite Differenzen Ansatz für die Berechnung der Matrix-Vektor-Multiplikation

$$\begin{aligned} \frac{\partial \varphi}{\partial Q} \cdot v &= \lim_{\varepsilon \rightarrow 0} \frac{\varphi(Q + \varepsilon v) - \varphi(Q)}{\varepsilon} \\ &\approx \frac{\varphi(Q + \varepsilon v) - \varphi(Q)}{\varepsilon} \quad (\varepsilon \text{ klein}). \end{aligned} \quad (5.3)$$

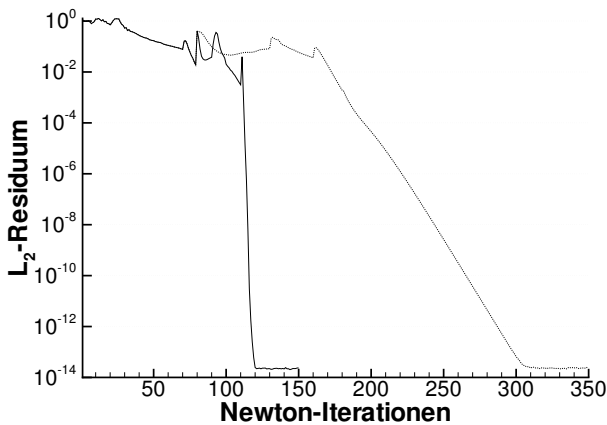


Abbildung 5.3: Konvergenzverlauf des inkonsistenten Newton-Verfahrens (Punkte) und konsistenten Newton-Verfahrens mit FD- φ -Ansatz mit $\varepsilon = 10^{-5}$ (Linie), Relaxationsparameter $\omega = 1,0$

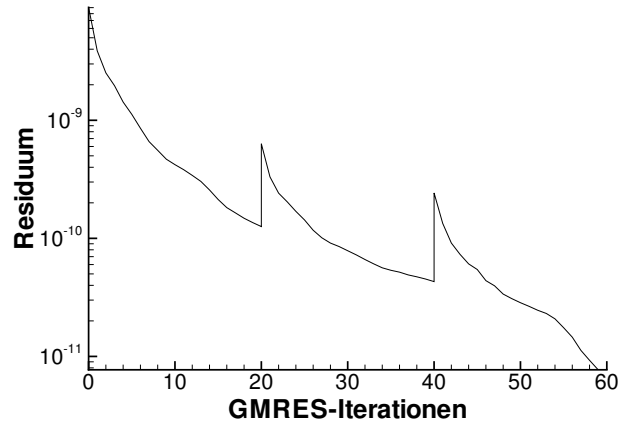


Abbildung 5.4: Konvergenz des GMRES-Verfahrens in der 123. Newton-Iteration, AUSMDV mit MC-Limiter, $\kappa=1,0$

Zwar stellt dies nur eine Approximation der Matrix-Vektor-Multiplikation dar, jedoch werden für die Finite Differenzen die Flüsse φ mit einer räumlichen Genauigkeit zweiter Ordnung benutzt. Im Folgenden wird dieses Verfahren mit FD- φ (Finite Differenzen für φ) bezeichnet. Derselbe Ansatz kann natürlich genauso für die viskosen Flüsse φ_v verwendet werden.

Die Wahl des Parameters ε in Gleichung (5.3) ist entscheidend für die Genauigkeit der Finiten Differenz. Dieser Parameter sollte einerseits sehr klein gewählt werden, um die Finite Differenz möglichst genau zu berechnen. Andererseits hat ein Computer nur eine endliche Rechengenauigkeit. Wird der Parameter ε zu klein gewählt, so ist der berechnete Wert nur auf wenige Stellen genau. Der Parameter ε wird über

$$\varepsilon = \frac{\sqrt{\varepsilon_0}}{\|v\|}, \tag{5.4}$$

bestimmt, wobei mit ε_0 die Rechengenauigkeit des Computers bezeichnet wird. Diese liegt bei doppelt genauer Berechnung je nach Computertyp bei ca. 10^{-12} .

Dieses FD- φ -Verfahren wird als *matrixfreies Verfahren* bezeichnet, da die unvorkonditionierten Krylov-Raum-Verfahren die Jacobi-Matrix nicht mehr explizit benötigen. Die Matrix $\frac{\partial R}{\partial Q}$ bräuchte also nicht berechnet werden. Wie jedoch in Kapitel 4 erläutert wurde, benötigen die Krylov-Raum-Verfahren eine Vorkonditionierung. Für die Vorkonditionierung wird daher weiterhin die approximierten Jacobi-Matrix berechnet.

In Abb. 5.3 sind Konvergenzverläufe für die Zylindervorkörperströmung mit den Zuströmdaten aus Tabelle 5.1 dargestellt. Zu sehen sind die Konvergenzverläufe des inkonsistenten und des konsistenten Newton-Verfahrens. Das konsistente FD- φ -Verfahren konvergiert deutlich schneller als das inkonsistente Verfahren. Doch bei genauer Analyse des Verfahrens deuten sich schon

Probleme an. Der Finite Differenzen Ansatz beinhaltet stets einen Approximationsfehler. Dieser Fehler kann durch das Restarted-GMRES-Verfahren sichtbar gemacht werden. Bei jedem Neustart des GMRES-Verfahrens wird das Residuum neu berechnet. Der dabei auftretende Residuensprung (siehe Abb. 5.4) entsteht durch die Approximationsfehler des Finite Differenzen Ansatzes. Im demonstrierten Beispiel stellen die Unregelmäßigkeiten in der GMRES-Konvergenz keine Probleme dar. Sollen jedoch stärkere Stöße berechnet werden, so werden die Residuensprünge im GMRES-Verfahren immer größer. Das GMRES-Verfahren kann aufgrund der zu großen Approximationsfehler der Finiten Differenzen das LGS nicht mehr genau genug lösen, was zu einem Zusammenbruch des Newton-Verfahrens bzw. zu einer drastischen Zeitschrittbeschränkung führt. Um den Approximationsfehler zu reduzieren, muss daher der Parameter ϵ optimal gewählt werden. Jedoch ist dies nur für gleichmäßig skalierte Funktionen φ möglich. Daher versagt das matrixfreie Verfahren bei starken Stößen.

5.3.2 Finite Differenzen für Links- und Rechtszustände

Um ein stabileres Verfahren zu erhalten, darf der Finite Differenzen Ansatz nur auf gleichmäßig skalierte Funktionen angewendet werden. Da die reibungsfreien Flüsse φ Funktionen der Links- und Rechtszustände an den Zellflächen sind, also $\varphi = \varphi(Q^l, Q^r)$, kann durch Anwenden der Kettenregel die Matrix-Vektor-Multiplikation umgeformt werden zu:

$$\frac{\partial \varphi}{\partial Q} \cdot v = \frac{\partial \varphi}{\partial Q^l} \cdot \frac{\partial Q^l}{\partial Q} \cdot v + \frac{\partial \varphi}{\partial Q^r} \cdot \frac{\partial Q^r}{\partial Q} \cdot v. \quad (5.5)$$

Die Jacobi-Matrizen $\frac{\partial \varphi}{\partial Q^l}$ und $\frac{\partial \varphi}{\partial Q^r}$ sind die Ableitungen der Euler-Flüsse nach den Links- bzw. Rechtszuständen Q^l und Q^r . Diese Jacobi-Matrizen sind exakt die Ableitungen der Euler-Flüsse mit einer Genauigkeit erster Ordnung, welche schon für das inkonsistente Zeitschritt-Newton-Verfahren benötigt wurden. Die Matrix-Vektor-Produkte

$$\frac{\partial Q^l}{\partial Q} \cdot v \quad \text{und} \quad \frac{\partial Q^r}{\partial Q} \cdot v$$

werden durch den Finite-Differenzen-Ansatz approximiert:

$$\frac{\partial Q^l}{\partial Q} \cdot v \approx \frac{Q^l(Q + \epsilon v) - Q^l(Q)}{\epsilon} \quad \text{und} \quad \frac{\partial Q^r}{\partial Q} \cdot v \approx \frac{Q^r(Q + \epsilon v) - Q^r(Q)}{\epsilon}. \quad (5.6)$$

Diese MVM-Variante, bestehend aus Gleichungen (5.5) und (5.6), wird im Folgenden mit mit FD-Q bezeichnet. Als Testfall wird eine reibungsfreie Zylindervorkörperströmung mit $Ma_\infty = 8,25$ gewählt. Die Isomach-Linien sind in Abb. 5.5 zu sehen. Die Zuströmbedingungen für diesen Testfall sind in Tabelle 5.2 zusammengestellt.

In Abb. 5.6 sind drei Konvergenzverläufe dargestellt

- Simulation mit einer räumlichen Diskretisierung erster Ordnung (schwarze Linie)

Zuströmdruck	2490,56 Pa	Partialdichte N_2	$2,984 \cdot 10^{-2} \text{ kg/m}^3$
Zuströmmachzahl	8,25	Partialdichte O_2	$9,152 \cdot 10^{-3} \text{ kg/m}^3$
Vibrationstemperatur von N_2	221,7 K	Partialdichte NO	$4,054 \cdot 10^{-14} \text{ kg/m}^3$
Vibrationstemperatur von O_2	221,7 K	Partialdichte N	$1,893 \cdot 10^{-14} \text{ kg/m}^3$
Vibrationstemperatur von NO	221,7 K	Partialdichte O	$2,162 \cdot 10^{-14} \text{ kg/m}^3$

Tabelle 5.2: Zuström��daten für eine Zylindervorkörperströmung

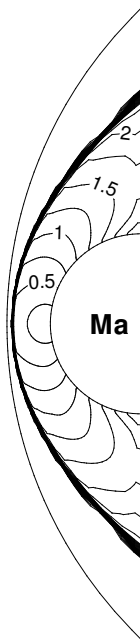


Abbildung 5.5: Zylinder-Vorkörperströmung, Isomach-Linien

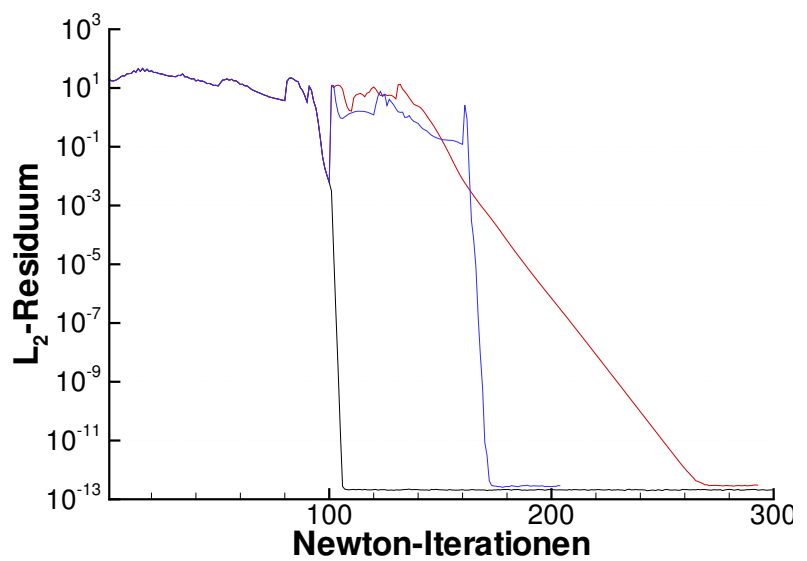


Abbildung 5.6: Konvergenzverlauf für eine reibungsfreie Zylinder-Vorkörperströmung mit AUSMDV, $Ma_\infty=8.25$, berechnet mit FD-Q-Ansatz

- Simulation mit einer räumlichen Diskretisierung zweiter Ordnung, inkonsistentes Newton-Verfahren (rote Linie)
- Simulation mit einer räumlichen Diskretisierung zweiter Ordnung, konsistentes FD-Q Verfahren (blaue Linie)

Mit diesem FD-Q Ansatz konnte eine deutliche Konvergenzbeschleunigung erreicht werden, jedoch ist dieses Verfahren für $Ma_\infty \approx 20$ immer noch nicht einsetzbar.

5.3.3 Finite Differenzen für die Steigungen

Der zuvor gemachten FD-Q Ansatz wird in diesem Abschnitt fortgesetzt, indem die Links- und Rechtszustände Q^l und Q^r an den Zellflächen als Funktionen der extrapolierten Primitivva-

riablen P^l und P^r , und diese wiederum als Funktionen vom Erhaltungsvektor Q betrachtet werden. So entsteht durch weiteres Anwenden der Kettenregel der folgende Ausdruck:

$$\frac{\partial \varphi}{\partial Q} \cdot v = \left(\frac{\partial \varphi}{\partial Q^l} \cdot \frac{\partial Q^l}{\partial P^l} \cdot \frac{\partial P^l}{\partial P} + \frac{\partial \varphi}{\partial Q^r} \cdot \frac{\partial Q^r}{\partial P^r} \cdot \frac{\partial P^r}{\partial P} \right) \frac{\partial P}{\partial Q} \cdot v. \quad (5.7)$$

Der Vektor P ist der Vektor der Variablen, die extrapoliert werden (vgl. Abschnitt 3.2.3),

$$P = (\rho_s, u, v, w, p, \rho_s e_{vib,s}).$$

Die Matrix-Vektor-Produkte mit den Matrizen

$$\frac{\partial \varphi}{\partial Q^l}, \quad \frac{\partial Q^l}{\partial P^l}, \quad \frac{\partial \varphi}{\partial Q^r}, \quad \frac{\partial Q^r}{\partial P^r} \quad \text{und} \quad \frac{\partial P}{\partial Q}$$

werden exakt berechnet, d. h. diese Matrizen werden berechnet und gesondert abgespeichert.

Die Matrix-Vektor-Produkte

$$\frac{\partial P^l}{\partial P} \cdot v \quad \text{und} \quad \frac{\partial P^r}{\partial P} \cdot v$$

werden über den Finite-Differenzen-Ansatz berechnet, wobei die Berechnung der Links- und Rechtszustände

$$P^l = P + \frac{\Delta x}{2} \cdot s(P) \quad P^r = P - \frac{\Delta x}{2} \cdot s(P) \quad (5.8)$$

berücksichtigt wird. Damit ergibt sich

$$\frac{\partial P^l}{\partial P} \cdot v \approx v + \frac{\Delta x}{2} \cdot \frac{s(P + \epsilon v) - s(P)}{\epsilon} \quad \text{und} \quad \frac{\partial P^r}{\partial P} \cdot v \approx v - \frac{\Delta x}{2} \cdot \frac{s(P + \epsilon v) - s(P)}{\epsilon}. \quad (5.9)$$

Dieses Verfahren wird im Folgenden mit FD-S (Finite-Differenzen für Steigungen) bezeichnet.

Dieser Ansatz baut wieder auf den Finite Differenzen Ansatz auf. Dadurch entsteht wiederum die erwähnte ϵ -Problematik. Dieser Ansatz ist jedoch nicht stabiler als der FD-Q-Ansatz.

5.3.4 Exakte Matrix-Vektor-Multiplikation

Durch die bisher hergeleiteten konsistenten Newton-Verfahren ist es möglich, Überschallströmungen berechnen zu können. Allerdings haben diese Verfahren noch nicht die notwendige Robustheit und erreichen für Hyperschallströmungen nicht die gewünschte quadratische Konvergenz. Um die Ursachen hierfür zu finden, wird in diesem Abschnitt ein sehr aufwändiges Verfahren hergeleitet, welches die Matrix-Vektor-Multiplikation exakt berechnet, ohne dabei eine Finite-Differenzen-Approximation einzusetzen.

Der FD-S-Ansatz kann leicht zur exakten Matrix-Vektor-Multiplikation erweitert werden, in dem auch die Jacobi-Matrix $\frac{\partial s}{\partial P}$ exakt berechnet wird. Dies ist allerdings nur für eine Extrapolation der Primitivvariablen ohne Transformation in die charakteristischen Steigungen leicht zu realisieren. Damit entsteht die exakte Matrix-Vektor-Multiplikation

$$\frac{\partial \varphi}{\partial Q} \cdot v = \left(\frac{\partial \varphi}{\partial Q^l} \cdot \frac{\partial Q^l}{\partial P^l} \cdot \left[E + \frac{\Delta x}{2} \frac{\partial s}{\partial P} \right] + \frac{\partial \varphi}{\partial Q^r} \cdot \frac{\partial Q^r}{\partial P^r} \cdot \left[E - \frac{\Delta x}{2} \frac{\partial s}{\partial P} \right] \right) \frac{\partial P}{\partial Q} \cdot v \quad (5.10)$$

Dieses Verfahren ist für Anwendungsrechnungen nicht mehr praktikabel, da sowohl der Speicherplatzbedarf als auch der Rechenaufwand zu groß sind. Anhand dieses Ansatzes ist es jedoch möglich, die Probleme und Schwierigkeiten eines konsistenten Newton-Verfahrens zu untersuchen.

Mit diesem Verfahren ist es jetzt erstmals möglich, auch Hyperschallströmungen mit $Ma_\infty=20$ zu berechnen (siehe Kapitel 6).

5.3.5 Exakte Matrix-Vektor-Multiplikation für die Jacobi-Matrix der viskosen Flüsse

Wie bereits erwähnt, kann für die MVM mit der Jacobi-Matrix der viskosen Flüsse der FD- φ -Ansatz benutzt werden,

$$\frac{\partial \varphi_v}{\partial Q} \cdot v \approx \frac{\varphi_v(Q + \varepsilon v) - \varphi_v(Q)}{\varepsilon} \quad (\varepsilon \text{ klein}). \quad (5.11)$$

Um jedoch für die Konvergenzuntersuchungen die Unsicherheiten bei der Finiten-Differenzen-Approximation auszugrenzen, wird hier ein anderer Ansatz verfolgt.

Barth benutzt für die MVM mit der Jacobi-Matrix der reibungsfreien Flüsse in [8] einen Ansatz, der aufgrund einer anderen Extrapolation im hier vorgestellten Verfahren so nicht angewendet werden kann. Allerdings kann die Idee von Barth bei der MVM mit der Jacobi-Matrix der viskosen Flüsse eingesetzt werden. Für die viskosen Flüsse

$$E_v = \begin{pmatrix} \rho D_s \psi_{s,x} \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ q_x + u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \\ \lambda_{v,s} \frac{\partial T_{v,s}}{\partial x} + h_{t,s} \rho D_s \psi_{s,x} \end{pmatrix} \quad F_v = \begin{pmatrix} \rho D_s \psi_{s,y} \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ q_y + u\tau_{xy} + v\tau_{yy} + w\tau_{yz} \\ \lambda_{v,s} \frac{\partial T_{v,s}}{\partial y} + h_{t,s} \rho D_s \psi_{s,x} \end{pmatrix}$$

$$G_v = \begin{pmatrix} \rho D_s \psi_{s,z} \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ q_z + u\tau_{xz} + v\tau_{yz} + w\tau_{zz} \\ \lambda_{v,s} \frac{\partial T_{v,s}}{\partial z} + h_{t,s} \rho D_s \psi_{s,z} \end{pmatrix}$$

werden zwei Variablen-Vektoren eingeführt:

$$S = (\rho_m, u_m, v_m, w_m, e_{vib,N_2,m}, e_{vib,O_2,m}, e_{vib,NO,m}, T_{tr,m}, \quad (5.12)$$

$$T_{vib,N_2,m}, T_{vib,O_2,m}, T_{vib,NO,m}, h_{s,m}, \psi_{s,m}, \xi_{s,m}, \quad (5.13)$$

$$u_x, u_y, u_z, v_x, v_y, v_z, w_x, w_y, w_z, \psi_{s,x}, \psi_{s,y}, \psi_{s,z}, T_x, T_y, T_z \quad (5.14)$$

$$T_{vib,N_2,x}, T_{vib,O_2,y}, T_{vib,NO,z}, \tau_{xx}, \tau_{xy}, \tau_{xz}, \tau_{yy}, \tau_{yz}, \tau_{zz}) \quad (5.15)$$

$$P = (\rho, u, v, w, p, e_{vib,N_2}, e_{vib,O_2}, e_{vib,NO}, T_{vib,N_2}, T_{vib,O_2}, T_{vib,NO}, T_{tr}, \psi_s, h_s). \quad (5.16)$$

Der Vektor S besteht aus den Mittelwerten (Indizes m) der Primitivvariablen auf den Zellflächen sowie den räumlichen Ableitungen (Indizes x, y und z) der Primitivvariablen. Die Primitivvariablen sind im Vektor P zusammengefasst. Mit der Kettenregel ergibt sich für die Matrix-Vektor-Multiplikation

$$\frac{\partial \varphi_v}{\partial Q} \cdot v = \frac{\partial \varphi_v}{\partial S} \cdot \frac{\partial S}{\partial P} \cdot \frac{\partial P}{\partial Q} \cdot v. \quad (5.17)$$

Die Jacobi-Matrix $\frac{\partial P}{\partial Q}$ ist wieder die Ableitung der Primitivvariablen P nach dem Erhaltungsvektor Q . Diese Matrix wird zum Teil schon bei der MVM mit der Jacobi-Matrix der Euler-Flüsse benötigt. Der Variablenvektor S ist so gewählt worden, dass die Matrix $\frac{\partial \varphi_v}{\partial S}$ problemlos berechnet werden kann. Somit können die Matrizen $\frac{\partial P}{\partial Q}$ und $\frac{\partial \varphi_v}{\partial S}$ leicht berechnet und abgespeichert werden.

Im Gegensatz dazu braucht die Jacobi-Matrix $\frac{\partial S}{\partial P}$ *nicht* berechnet zu werden. Die Vektoren S und P sind so gewählt worden, dass S eine *lineare Funktion* in den Variablen P ist. Damit ist die Idee von Barth anwendbar. Wegen

$$\frac{\partial S(P)}{\partial P} \cdot v = S(v)$$

wird anstelle einer Matrix-Vektor-Multiplikation die Funktion S an der Stelle v ausgewertet. Dies kann im Computerprogramm sehr einfach realisiert werden, da dieselben Unterprogramme, die $S(P)$ für die viskosen Flüsse berechnen, genauso für $S(v)$ eingesetzt werden können.

Bei den bisherigen Überlegungen werden die Transportkoeffizienten noch nicht mitberücksichtigt. Nach Vorschlägen von Fertig werden diese approximativ abgeleitet, in dem die Beziehung

$$\lambda \propto \sqrt{T}$$

benutzt wird. Dies ist die einzige Approximation in der Matrix-Vektor-Multiplikation.

Durch die exakten Matrix-Vektor-Multiplikationen steht nun ein sehr aufwändiges, aber dafür exaktes Newton-Verfahren zur Verfügung. Mit diesem Verfahren ist es möglich, die Konvergenz der konsistenten Newton-Verfahren bei der Berechnung von Hyperschallströmungen genau zu analysieren. Die Ergebnisse dieser Analysen sind in Kapitel 6 zusammengefasst.

Kapitel 6

Ergebnisse

In diesem Kapitel wird die Leistungsfähigkeit des vorgestellten Simulationsverfahrens anhand typischer Testfälle demonstriert. Außerdem erfolgt dabei die Bewertung der verschiedenen numerischen Verfahren, die in den vorangegangenen Kapiteln vorgestellt wurden.

Das AUSMDV-Verfahren, das in dieser Arbeit für Hyperschallströmungen im thermochemischen Nichtgleichgewicht angepasst wurde, wird mit dem Roe-Verfahren verglichen. Die im Kapitel 4 vorgestellten Lösungsverfahren für lineare Gleichungssysteme einschließlich der Vorkonditionierungsmethoden werden detailliert untersucht. Desweiteren werden das inkonsistente und die konsistenten Newton-Krylov-Verfahren aus den Kapiteln 4 und 5 für die Berechnung von Hyperschallströmungen im thermochemischen Nichtgleichgewicht getestet.

6.1 Umströmung der MIRKA

Im Kapitel 3 wurden das Roe-Verfahren und das AUSMDV-Verfahren vorgestellt. In diesem Abschnitt soll anhand der Wiedereintrittskapsel MIRKA das AUSMDV-Verfahren mit dem Roe-Verfahren verglichen werden. Für die Simulation wurde ein Netz mit 72000 Zellen erstellt. Berechnet wird der Zeitpunkt der maximalen Wärmelast während des Wiedereintritts der Kapsel in die Erdatmosphäre. Die Zuströmbedingungen sind in der Tabelle 6.1 aufgelistet.

Zuströmdruck	37,88 Pa	Partialdichte N_2	$3,859 \cdot 10^{-4} \text{ kg/m}^3$
Zuströmmachzahl	20,43	Partialdichte O_2	$1,184 \cdot 10^{-4} \text{ kg/m}^3$
Vibrationstemperatur von N_2	260,72 K	Partialdichte NO	$5,244 \cdot 10^{-14} \text{ kg/m}^3$
Vibrationstemperatur von O_2	260,72 K	Partialdichte N	$2,448 \cdot 10^{-14} \text{ kg/m}^3$
Vibrationstemperatur von NO	260,72 K	Partialdichte O	$2,796 \cdot 10^{-14} \text{ kg/m}^3$

Tabelle 6.1: Zuströmdaten für die Wiedereintrittskapsel MIRKA zum Zeitpunkt der maximalen Wärmelast

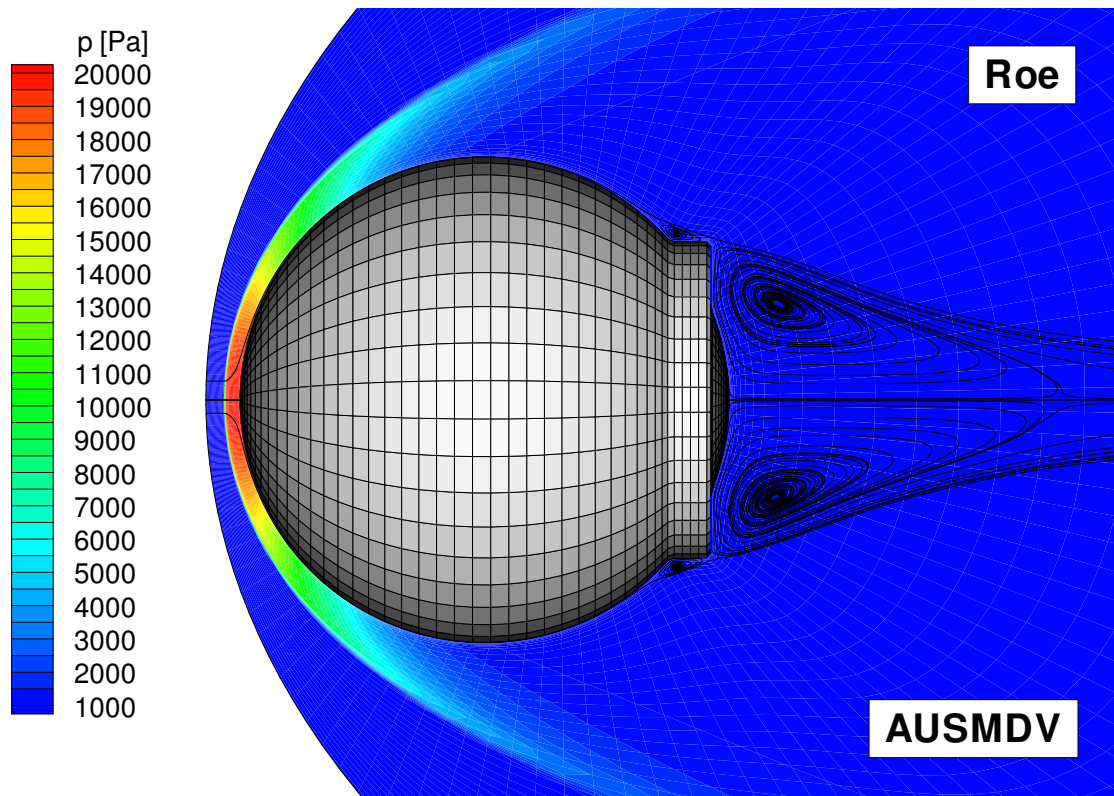


Abbildung 6.1: MIRKA-Kapsel, Druck- und Stromlinienvergleich

Der Vergleich der Isobaren und der Wirbelgröße (Abb. 6.1) sowie der Isomachlinien (Abb. 6.2) zeigt eine sehr gute Übereinstimmung der beiden Lösungen. Damit scheint das AUSMDV-Verfahren für die Simulation von Wiedereintrittsströmungen eine einsetzbare Alternative zum Roe-Verfahren zu sein.

Da jedoch das vorgestellte Verfahren die Aufgabe hat, den Oberflächenwärmestrom zu berechnen, ist zu überprüfen, ob diese sensible Größe durch das AUSMDV-Verfahren genauso berechnet werden kann wie mit dem Roe-Verfahren. In Abbildung 6.4 werden die Oberflächenwärmeströme miteinander verglichen. Zum leichteren Verständnis sind in Abbildung 6.3 ausgewählte Punkte der Oberflächenkontur markiert, die in Abbildung 6.4 wiederzufinden sind. Erstaunlicherweise führt das AUSMDV-Verfahren zu einem etwas niedrigeren Wärmestrom als das Roe-Verfahren. Dieses Resultat ist deshalb so erstaunlich, da der niedrigere Wärmestrom der AUSMDV-Lösung auf ein dissipationsärmeres Verfahren zurückschließen lässt, obwohl jedoch das Roe-Verfahren als das dissipationsärmste Verfahren zählt. Dies zeigt, dass im Roe-Verfahren die zusätzliche Entropiekorrektur für die mittleren Eigenwerte, welche zur Stoßstabilisierung eingeführt wurde, eine deutliche Auswirkung auf den Oberflächenwärmestrom hat.

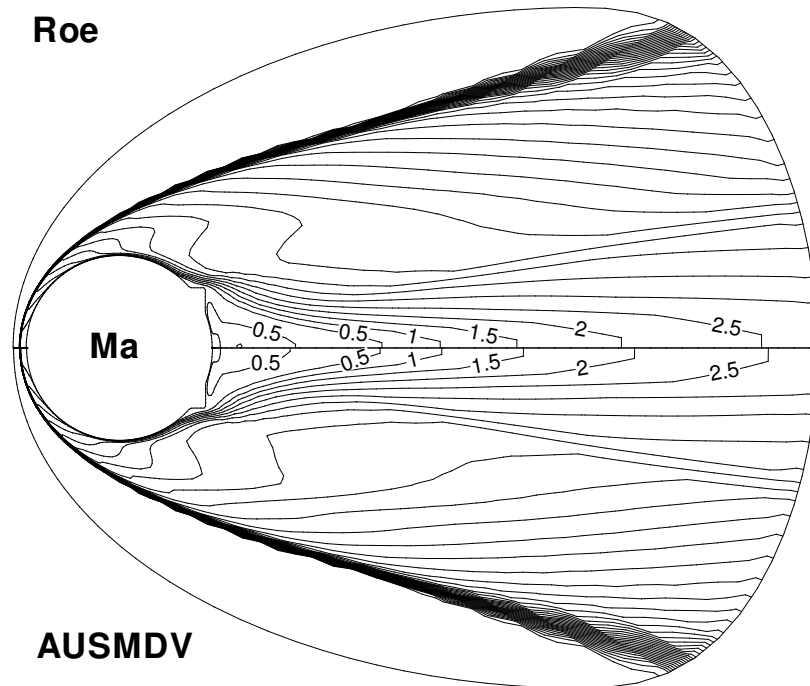


Abbildung 6.2: Vergleich der Isomach-Linien um die Wieder-
eintrittskapsel MIRKA

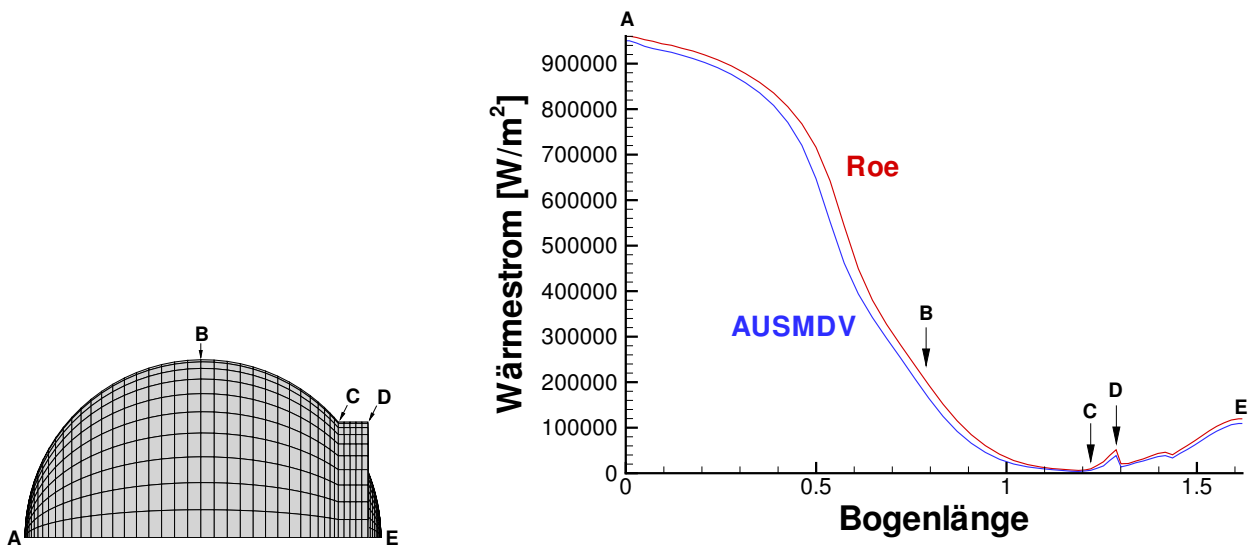


Abbildung 6.3: MIRKA mit
markierten Punkten für Abbil-
dung 6.4

Abbildung 6.4: Vergleich des Oberflächenwärmestroms der
MIRKA

Abschließende Bewertung

Das AUSMDV-Verfahren ist ein robustes und zugleich sehr genaues Verfahren und damit für den Einsatz in einem Simulationsverfahren für Hyperschallströmungen im thermochemischen Nichtgleichgewicht geeignet. Damit konnte im Rahmen dieser Arbeit eine stabile Alternative zum Roe-Verfahren gefunden werden.

6.2 Probenumströmung im Plasmawindkanal

Um die vorgestellten Newton-Verfahren zu testen und um weitere Einsatzmöglichkeiten des Programms zu demonstrieren, wird als nächster Testfall eine Probenumströmung in einem Plasmawindkanal berechnet. Für diesen numerischen Testfall wird als Zuströmbedingung ein Gas im Gleichgewicht mit einer Geschwindigkeit von $Ma_\infty=2,0$ und einer Gesamtenthalpie von $h_t = 20\text{MJ/kg}$ bei einem Umgebungsdruck von 1000 Pa gewählt. Aus diesen Vorgaben ergeben sich die Zuströmbedingungen, die in Tabelle 6.2 aufgelistet sind. In Abbildung 6.5 und 6.6 sind der Druck und die Vibrationstemperatur T_{vib,N_2} der stationären Lösung dargestellt.

Zuströmdruck	1000 Pa	Partialdichte N_2	$3,351 \cdot 10^{-4}\text{ kg/m}^3$
Zuströmmachzahl	2,0	Partialdichte O_2	$1,750 \cdot 10^{-8}\text{ kg/m}^3$
Vibrationstemperatur von N_2	4880,46 K	Partialdichte NO	$1,163 \cdot 10^{-6}\text{ kg/m}^3$
Vibrationstemperatur von O_2	4880,46 K	Partialdichte N	$6,986 \cdot 10^{-5}\text{ kg/m}^3$
Vibrationstemperatur von NO	4880,46 K	Partialdichte O	$1,225 \cdot 10^{-4}\text{ kg/m}^3$

Tabelle 6.2: Zuströmdaten für die Probenumströmung

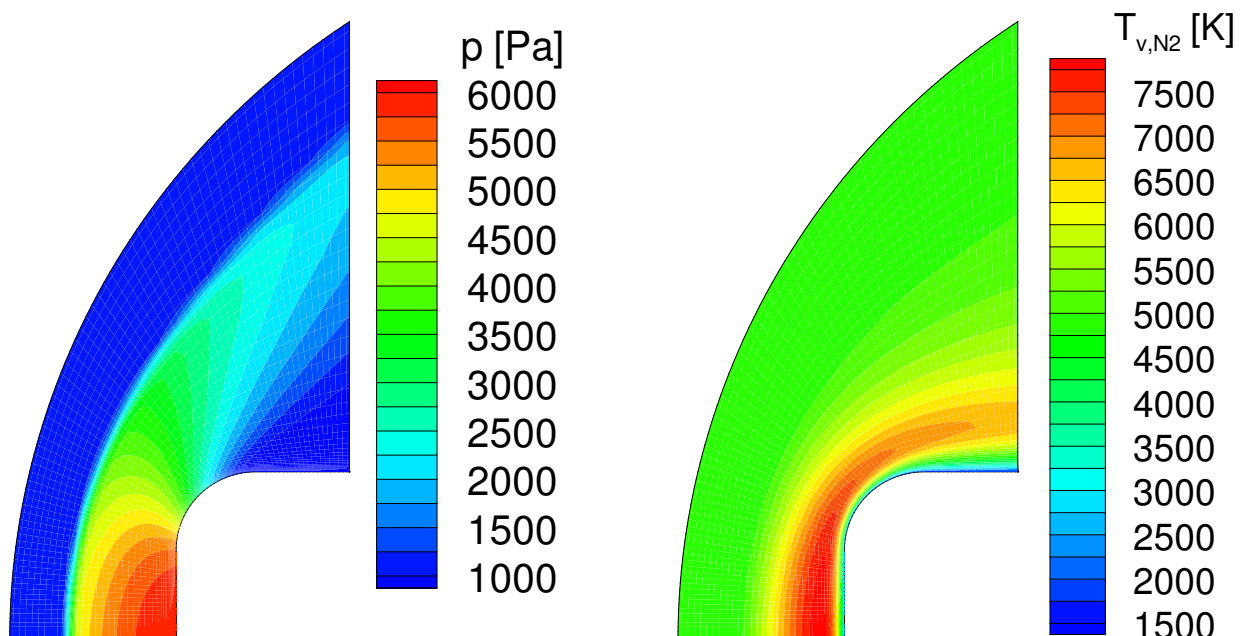


Abbildung 6.5: Druckverteilung bei der Probenumströmung

Abbildung 6.6: Vibrationstemperatur von N_2 bei der Probenumströmung

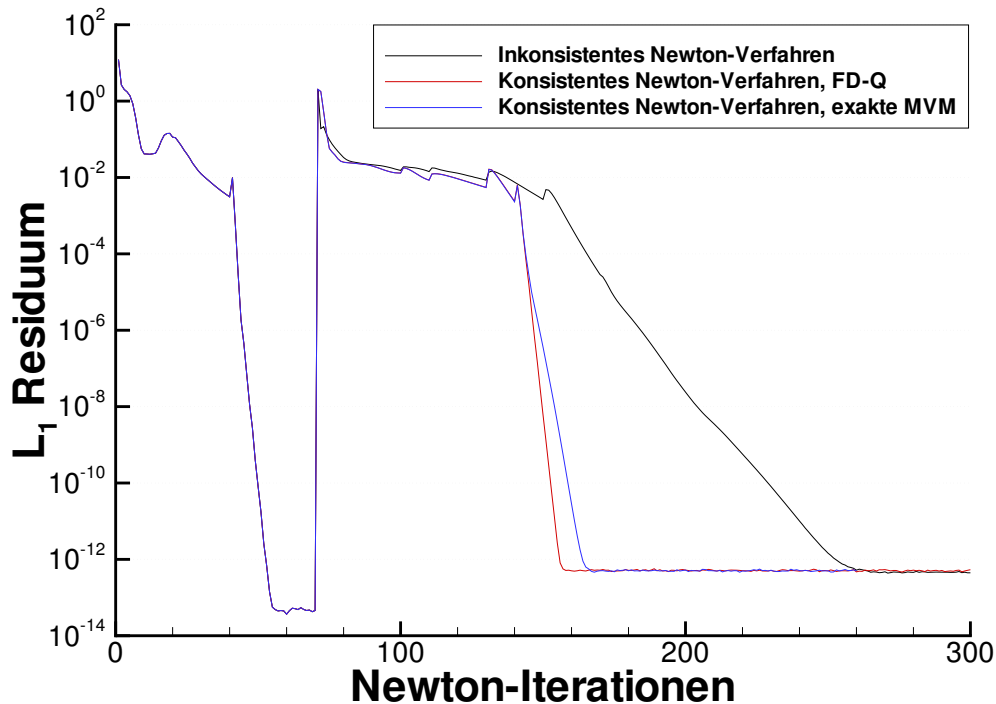


Abbildung 6.7: Konvergenzverlauf der Probenumströmung. Vergleich von inkonsistenten und konsistenten Newton-Verfahren

Parametersteuerung

In Abbildung 6.7 sind Konvergenzverläufe der getesteten Newton-Verfahren zu sehen. Während der ersten 70 Iterationen werden die reibungsfreien Flüsse mit einer räumlichen Genauigkeit erster Ordnung berechnet. Ab der 40. Iteration kann ein reines Newton-Verfahren ($CFL \rightarrow \infty$) eingesetzt werden, was zu einer quadratischen Konvergenz führt.

Nach der 70. Iteration werden die reibungsfreien Flüsse mit einer räumlichen Genauigkeit zweiter Ordnung diskretisiert. Eingesetzt wurde der van-Albada-Limiter. Die stationäre Strömungslösung wurde mit verschiedenen Newton-Krylov-Verfahren berechnet.

Zuerst wurde ein *inkonsistentes Newton-Verfahren* eingesetzt. Das Verfahren benötigt für die Konvergenz eine Unterrelaxation mit $\omega = 0.8$. Während der Konvergenzphase konnte die CFL-Zahl maximal auf 500 gesetzt werden. Das *matrixfreie FD- φ Verfahren* konvergierte bei diesem Strömungsfall nicht und ist deshalb in Abbildung 6.7 nicht dargestellt. Das *FD-Q Verfahren* hingegen konvergierte mit einem Relaxationsparameter von 1,0 und einer maximalen CFL-Zahl von 10^5 . Schließlich wurde noch die *exakte Matrix-Vektor-Multiplikation* eingesetzt, für die ebenfalls die Parameter $\omega = 1,0$ und $CFL=10^5$ möglich waren. Für diese Strömung ist damit ein konsistentes Verfahren mit $\omega = 1,0$ und $CFL=10^5$ möglich. Wie in Abbildung 6.7 zu sehen ist, konvergieren die beiden konsistenten Newton-Verfahren deutlich schneller als das inkonsistente Verfahren.

Desweiteren ist zu beachten, dass für die Simulationen trotz Hochenthalpieströmung die CFL-Zahl im konsistenten Newton-Verfahren sehr hoch gesetzt werden kann. Die Quellterme erzwingen also während der Konvergenzphasen keine Zeitschrittbeschränkung.

Abschließende Bewertung

Das vorgestellte Verfahren, welches hauptsächlich für die Simulation von Hyperschallströmungen entwickelt wurde, ist auch für die Simulation von Strömungen in Plasmawindkanälen einsetzbar. Außerdem konnte gezeigt werden, dass Hochenthalpieströmungen mit einem relativ schwachen Stoß mit den konsistenten Newton-Verfahren berechnet werden können. Die Konvergenz des Newton-Verfahrens konnte durch die Konsistenzsteigerung beschleunigt werden.

6.3 MIRKA-Vorkörperströmung

In diesem Abschnitt werden die vorgestellten Lösungsverfahren für lineare Gleichungssysteme und die Vorkonditionierer anhand eines einfachen Strömungsfalles detailliert verglichen. Als Testfall für die Lösungsverfahren wurde eine viskose Nichtgleichgewichts-Vorkörperströmung einer Kugel mit einem Meter Durchmesser gewählt. Die Zuströmdaten entsprechen den Daten des MIRKA-Wiedereintritts in 70 km Höhe.

Zuströmdruck	9,71 Pa	Partialdichte N_2	$1,1280 \cdot 10^{-4} \text{ kg/m}^3$
Zuströmmachzahl	24,25	Partialdichte O_2	$3,4596 \cdot 10^{-5} \text{ kg/m}^3$
Vibrationstemperatur von N_2	228,59K	Partialdichte NO	$1,5328 \cdot 10^{-16} \text{ kg/m}^3$
Vibrationstemperatur von O_2	228,59K	Partialdichte N	$7,1552 \cdot 10^{-17} \text{ kg/m}^3$
Vibrationstemperatur von NO	228,59K	Partialdichte O	$8,1727 \cdot 10^{-17} \text{ kg/m}^3$

Tabelle 6.3: Zuströmdaten für die MIRKA-Vorkörperströmung

Es wurde eine nichtkatalytische Oberfläche mit Hafttrandbedingungen und mit einer konstanten Oberflächentemperatur von 1500 K simuliert. Die Strömung wurde nur mit einer räumlichen Genauigkeit erster Ordnung berechnet. In Abbildung 6.3 ist das stoßangepasste Rechennetz mit 23088 Zellen zu sehen.

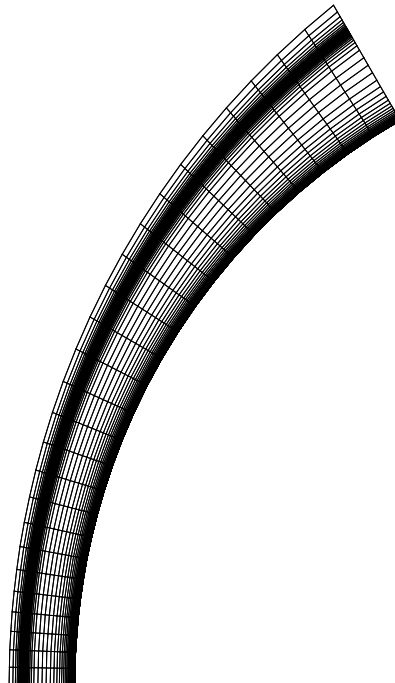


Abbildung 6.8: Stoßadaptiertes Rechennetz für die Vorkörperströmung

Speicherplatzbedarf

Für das hier vorgestellte Strömungssimulationsverfahren ist der Speicherplatzbedarf des linearen Gleichungssystems und der Vorkonditionierer von zentraler Bedeutung. Sämtliche Algorithmen für die Berechnung der Flüsse und der Quellterme lassen sich nahezu stufenlos von teilweise vektorisierbaren und speicherplatzsparenden bis hin zu vollvektorisierbaren und speicherplatzaufwändigen Algorithmen implementieren. Somit besteht bei diesen Verfahren stets die Möglichkeit, durch relativ geringfügige Rechenzeitzunahme den Speicherplatz zu reduzieren. Im Gegensatz dazu kann beim linearen Gleichungssystem kein Speicherplatz eingespart werden. Lediglich bei den Vorkonditionierern gibt es Unterschiede im Speicherplatzbedarf. Das lineare Gleichungssystem und die Vorkonditionierung benötigen innerhalb des Simulationsprogramm den meisten Speicherplatz und bestimmen damit den minimalen Speicherplatzbedarf des gesamten Verfahrens.

Der Speicherplatzbedarf kann in der Anzahl an Blockdiagonalen (BD) gemessen werden. Im linearen Gleichungssystem $Ax = b$ besteht die Matrix A aus 7 Blockdiagonalen. Wird die Matrix A mit doppelter Genauigkeit abgespeichert, so benötigt eine Blockdiagonale den 12-fachen Speicherplatz des Vektors b . Bei einfacher Genauigkeit der Matrix A hat eine Blockdiagonale nur den sechsfachen Speicherplatzbedarf von b .

Vorkonditionierer	Vektorisierung	Speicherplatzbedarf
BD	-	1 BD
LUSGS	-	1 BD
LUSGS	Neumann	1 BD
LUSGS	Hyperebenen	7 BD
ILU	-	4 BD
ILU	Neumann	4 BD
ILU	Hyperebenen	7 BD
LLU	-	8 BD
JLR(l)	-	8 BD

Tabelle 6.4: Vergleich des zusätzlichen Speicherplatzbedarfs für die verschiedenen Vorkonditionierer

In Tabelle 6.4 ist der Speicherplatzbedarf für die Vorkonditionierer aufgelistet, der *zusätzlich* zum Speicherplatzbedarf für das lineare Gleichungssystem benötigt wird.

Das JLR-Verfahren benötigt zusätzlich zur Matrix A Speicherplatz für weitere 8 Blockdiagonalen und 3 Vektoren. Die Krylov-Raum-Verfahren benötigen mit Ausnahme des GMRES-Verfahrens noch zusätzlichen Speicherplatz von weniger als 11 Vektoren. Beim GMRES-Verfahren steigt der Speicherplatz mit jeder Iteration an. Der zusätzliche Speicherplatzbedarf beträgt $(i + 5)$ Vektoren, wobei i die Anzahl der Iterationen ist. Soweit nicht ausdrücklich anderes angegeben ist, wird in dieser Arbeit der *restart-level* auf 20 gesetzt, so dass der maximale zusätzliche Speicherplatzbedarf für das GMRES-Verfahren bei 25 Vektoren liegt. In einer Simulationsrechnung mit gut aufgelöstem Rechnetz und guter Vorkonditionierung sind jedoch in der Regel

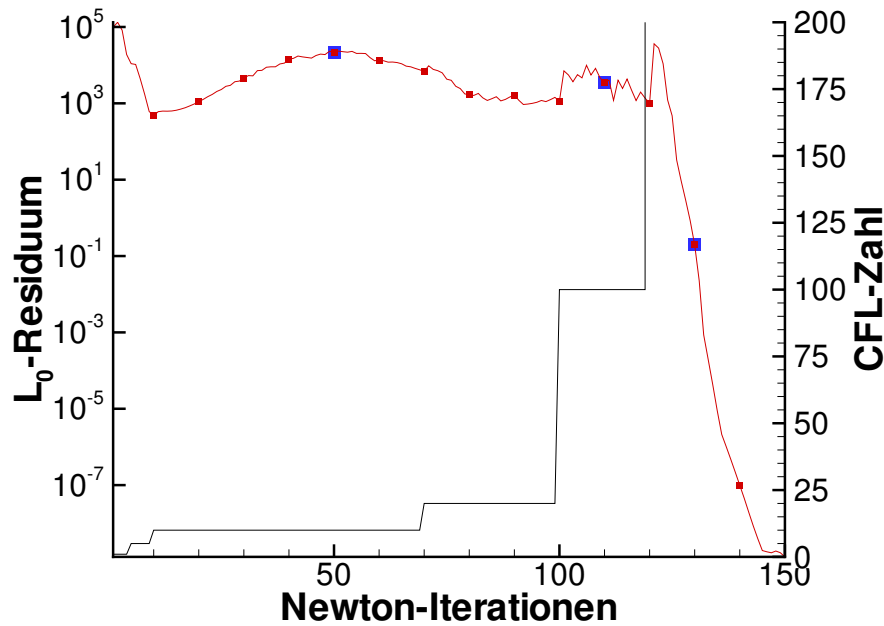


Abbildung 6.9: CFL-Zahl-Steuerung (schwarze Linie) und Konvergenzverlauf (rote Linie) des Zeitschritt-Newton-Verfahrens, ausgewählte Iterationen sind blau markiert

keine 20 GMRES-Iterationen zum Lösen des linearen Gleichungssystems notwendig.

Testfälle

Im Konvergenzverlauf des Zeitschritt-Newton-Verfahrens (siehe Abb. 6.3) wurden drei für das Gesamtverfahren typische Newton-Iterationen ausgewählt:

- Newton-Iteration Nr. 50: Diese Iteration gehört zur Startphase, in der der Stoß im Netz bewegt wird. Daher darf in dieser Iteration nur eine kleine CFL-Zahl (CFL=10) gewählt werden.
- Newton-Iteration Nr. 110: Um den Stoß im Netz zu positionieren, muss die CFL-Zahl erhöht werden (CFL=100). Dies ist die kritischste Phase während des Simulationslaufes.
- Newton-Iteration Nr. 130: In der Konvergenzphase kann hier ein reines Newton-Verfahren eingesetzt werden (CFL $\rightarrow \infty$).

Die linearen Gleichungssysteme, die in diesen ausgewählten Newton-Iterationen gelöst werden müssen, unterscheiden sich sowohl aufgrund der unterschiedlichen CFL-Zahlen als auch aufgrund der unterschiedlichen Strömungssituationen. Damit ist nicht zu erwarten, dass die Lösungsverfahren für diese ausgewählten Newton-Iterationen stets gleich konvergieren.

In diesen drei Newton-Iterationen wurden folgende Gleichungslöser getestet:

- JLR
- GMRES ohne Vorkonditionierung
- GMRES mit Blockdiagonalskalierung
- GMRES mit ILU-Vorkonditionierung
- GMRES mit LLU-Vorkonditionierung
- GMRES mit LUSGS-Vorkonditionierung
- GMRES mit JLR(0)-, JLR(1)-, JLR(2)-, JLR(3)-, JLR(5)- und JLR(10)-Vorkonditionierung

Das GMRES-Verfahren wurde für diese Tests ohne *restarts* eingesetzt. Als Startwerte für die Iterationsverfahren wurde die Lösung des linearen Gleichungssystems der vorhergehenden Newton-Iteration gewählt.

Rechenzeitvergleich

Das *unvorkonditionierte* GMRES-Verfahren konvergiert in keinem der drei untersuchten Iterationen. Das GMRES-Verfahren mit der Blockdiagonalskalierung konvergiert zwar, aber die Konvergenzrate lässt stark nach, so dass nicht sichergestellt ist, dass das lineare Gleichungssystem in der notwendigen Genauigkeit gelöst werden kann. Da bei gleichen Testrechnungen das BiCGStab-Verfahren mit Blockdiagonalskalierung gar nicht konvergierte, scheint diese Vorkonditionierung für diese Simulation nicht geeignet zu sein. Die Vorkonditionierer JLR(2), JLR(3), JLR(5) und JLR(10) reduzieren zwar erwartungsgemäß deutlich die Anzahl der GMRES-Iterationen, die Rechenzeit nimmt jedoch zu, so dass das GMRES-Verfahren mit diesen Vorkonditionierern mehr Rechenzeit benötigt als mit JLR(0) und JLR(1).

Aus den genannten Gründen werden in den folgenden Diagrammen nur JLR, GMRES-ILU, GMRES-LLU, GMRES-LUSGS, GMRES-JLR(0) und GMRES-JLR(1) miteinander verglichen.

Für die Konvergenz des Newton-Verfahrens ist es nicht erforderlich, das lineare Gleichungssystem sehr genau zu lösen. Innerhalb der ersten Rechenphase, in der der Stoß im Netz positioniert wird (50. Iteration), reicht eine Reduktion der Residuennorm um 2 bis 3 Größenordnungen aus. In der Stabilisierungsphase (110. Iteration) sollte die Residuennorm des linearen Gleichungssystems um 4 Größenordnungen abgesenkt werden. Dagegen reicht in der Konvergenzphase (130. Iteration) für diesen Rechenfall eine Reduktion der Residuennorm um 1 bis 2 Größenordnungen aus.

In den Abbildungen 6.10, 6.11 und 6.12 sind die Konvergenzverläufe über der Zeit aufgetragen, wobei die Simulationen auf einem PC mit einer Prozessorleistung von 1,7 GHz durchgeführt

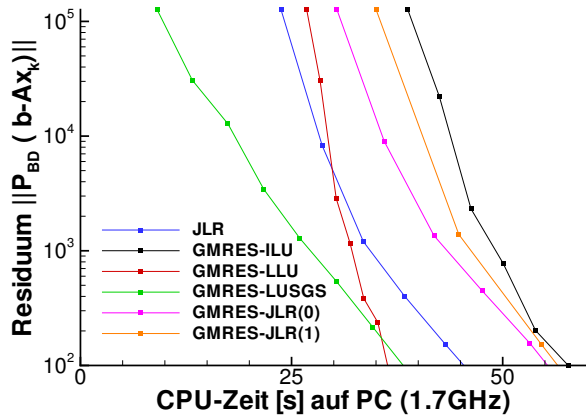


Abbildung 6.10: Konvergenzverläufe in der 50. Newton-Iteration auf einem PC

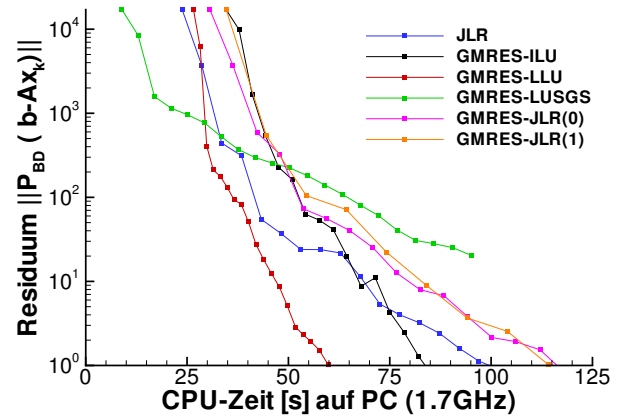


Abbildung 6.11: Konvergenzverläufe in der 110. Newton-Iteration auf einem PC

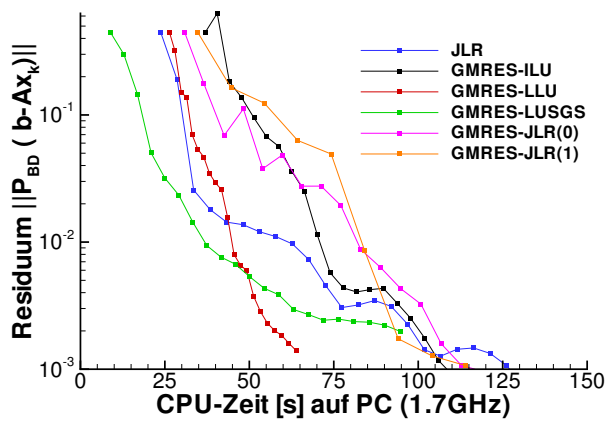


Abbildung 6.12: Konvergenzverläufe in der 130. Newton-Iteration auf einem PC

wurden. Um die Konvergenzverläufe miteinander vergleichen zu können, müssen vergleichbare Residuen herangezogen werden. Daher ist in den Abbildungen einheitlich für alle Verfahren die Residuennorm $\|P_{BD}(b - Ax_k)\|_2$ dargestellt, obwohl das GMRES-Verfahren je nach Vorkonditionierung eine andere Norm minimiert. Die Zeit wurde ab dem Zeitpunkt gemessen, ab dem das lineare Gleichungssystem ohne Vorkonditionierung vollständig aufgestellt war. Somit geht die Zeit für die Berechnung der Vorkonditionierung in die Zeitmessung und damit auch in die Diagramme mit ein.

Auffallend ist das schlechte Abschneiden von GMRES-ILU. Lediglich in der 110. Iteration führt die ILU-Vorkonditionierung zu einer schnelleren Konvergenz des GMRES-Verfahrens. Der Grund für das schlechte Abschneiden liegt in der aufwändigen Berechnung der ILU-Zerlegung. Dagegen ist in allen drei untersuchten Fällen GMRES-LLU schneller als GMRES-ILU. Das GMRES-LUSGS-Verfahren benötigt zwar mehr Iterationen als GMRES-ILU oder GMRES-LLU, aber aufgrund der geringen Zeit für die Berechnung der Vorkonditionierung ist es in der

50. und der 130. Newton-Iteration eines der schnellsten Verfahren. In der 110. Newton-Iteration ist GMRES-LUSGS jedoch das langsamste Verfahren. Das JLR-Verfahren gehört mit zu den schnellsten Verfahren, dagegen sind GMRES-JLR(0) und GMRES-JLR(1) stets die langsameren Varianten. Zusammenfassend kann gesagt werden, dass alle in den Diagrammen dargestellte Verfahren die linearen Gleichungssysteme lösen können und damit prinzipiell zum Einsatz im Zeitschritt-Newton-Verfahren geeignet sind. Für den PC scheint das beste Verfahren GMRES mit LLU-Vorkonditionierung zu sein. Da jedoch auf einem PC in der Regel der zur Verfügung stehende Speicherplatz doch recht beschränkt ist, könnte der LUSGS-Vorkonditionierer mit seinem geringen Speicherplatzbedarf durchaus eine einsetzbare Variante sein.

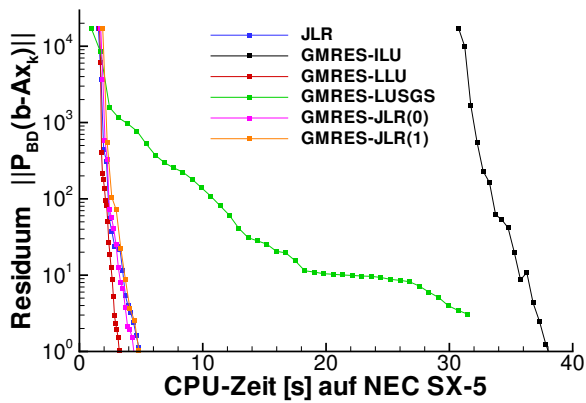


Abbildung 6.13: Konvergenzverläufe in der 110. Newton-Iteration auf NEC SX-5, ILU und LUSGS nichtvektorisierbar

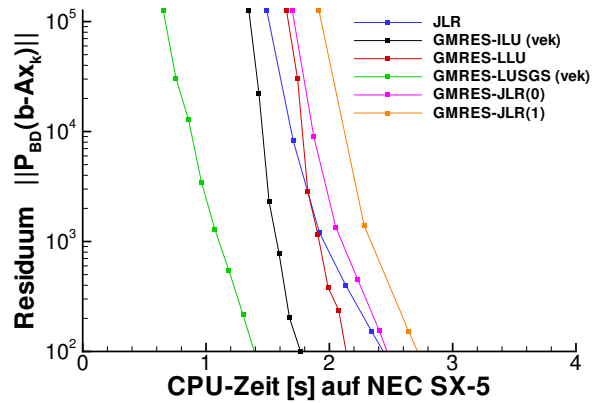


Abbildung 6.14: Konvergenzverläufe in der 50. Newton-Iteration auf NEC SX-5, ILU und LUSGS vektorisierbar

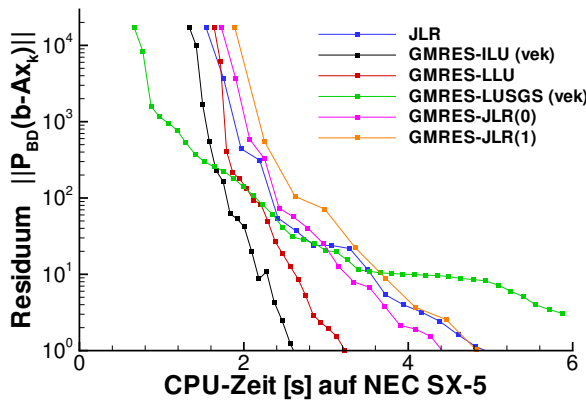


Abbildung 6.15: Konvergenzverläufe in der 110. Newton-Iteration auf NEC SX-5, ILU und LUSGS vektorisierbar

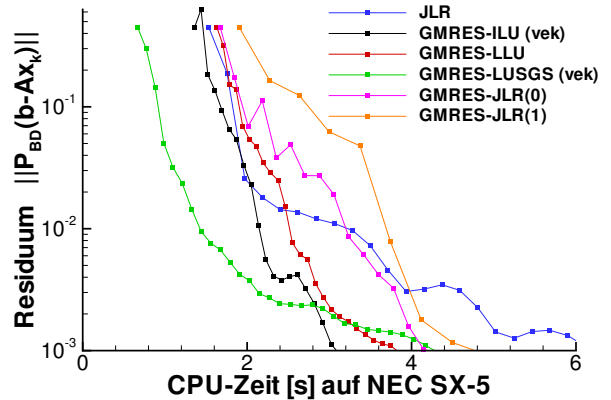


Abbildung 6.16: Konvergenzverläufe in der 130. Newton-Iteration auf NEC SX-5, ILU und LUSGS vektorisierbar

Auf dem Vektorrechner NEC SX-5 sehen dagegen die Konvergenzverläufe derselben Verfahren anders aus. In Abbildung 6.13 sind die Konvergenzverläufe für die 110. Newton-Iteration dargestellt, jedoch wurden die Simulationen auf dem Vektorrechner durchgeführt. Die nicht

vektorisierbare Berechnung der ILU-Vorkonditionierung benötigt über 30 Sekunden, erst dann kann das GMRES-Verfahren beginnen, das lineare Gleichungssystem zu lösen. Andere Verfahren haben dagegen schon nach 3 Sekunden das lineare Gleichungssystem gelöst. Ebenso fällt auf, dass das GMRES-Verfahren mit der LUSGS-Vorkonitionierung deutlich langsamer ist als die restlichen Verfahren. Daher werden für die ILU- und für die LUSGS-Vorkonditionierung die vektorisierbaren Varianten mit Hilfe des Hyperebenenformats eingesetzt.

Wie in den Abbildungen 6.14, 6.15 und 6.16 zu sehen ist, schneidet GMRES mit ILU(HE) am besten ab. Lediglich das GMRES-LUSGS-Verfahren ist in der 50. und 130. Newton-Iteration noch schneller als GMRES-ILU. GMRES-LLU ist für diesen Strömungsfall auf einem Vektorrechner langsamer als GMRES-ILU. Die langsamsten Verfahren sind JLR, GMRES-JLR(0) und GMRES-JLR(1).

Abschließende Bewertung

In diesem Abschnitt wurde gezeigt, dass die Lösungsverfahren JLR, GMRES-ILU, GMRES-LLU, GMRES-LUSGS, GMRES-JLR(0) und GMRES-JLR(1) für den Einsatz im vorgestellten Strömungssimulationsverfahren geeignet sind. Insbesondere stellte sich heraus, dass das JLR-Verfahren genauso effizient ist wie das GMRES-Verfahren. Die besten Vorkonditionierer für das GMRES-Verfahren sind ILU, LUSGS und LLU, wobei auf einem Vektorrechner eine vektorisierbare Variante der ILU- und LUSGS-Vorkonditionierer eingesetzt werden muss. Der LUSGS-Vorkonditionierer ist eine speicherplatzsparende Alternative zum ILU-Vorkonditionierer und in einigen Fällen sogar die schnellere Variante.

6.4 Umströmung der Rückkehrkapsel COLIBRI

Die verschiedenen Lösungsverfahren für das lineare Gleichungssystem sowie die konsistenten und das inkonsistente Newton-Verfahren werden anhand einer COLIBRI-Wiedereintrittssimulation getestet. Die Nichtgleichgewichtsströmung, der starke Stoß sowie die Grenzschicht können mit dem vorgestellten Simulationsverfahren problemlos in weniger als 400 Iterationen berechnet werden.

Zuströmbedingungen

Die Kapsel wird in einer Flughöhe von 66 km mit einer Geschwindigkeit von $Ma_\infty=18,2$ und einem Anstellwinkel von $\alpha = 22^\circ$ simuliert. Die Zuströmbedingungen sind in Tab. 6.5 aufgelistet.

Zuströmdruck	9,28 Pa	Partialdichte N_2	$1,083 \cdot 10^{-4} \text{ kg/m}^3$
Zuströmmachzahl	18,2	Partialdichte O_2	$3,321 \cdot 10^{-5} \text{ kg/m}^3$
Vibrationstemperatur von N_2	227,51 K	Partialdichte NO	$1,472 \cdot 10^{-13} \text{ kg/m}^3$
Vibrationstemperatur von O_2	227,51 K	Partialdichte N	$6,869 \cdot 10^{-14} \text{ kg/m}^3$
Vibrationstemperatur von NO	227,51 K	Partialdichte O	$7,846 \cdot 10^{-14} \text{ kg/m}^3$

Tabelle 6.5: Zuströmdaten für COLIBRI

Die reibungsfreien Flüsse werden mit dem AUSMDV-Verfahren und dem Limiter von van Albada berechnet. Das zugrundeliegende strukturierte Netz mit metrischer Singulartität besteht aus 42570 Zellen. Die Isomachlinien der stationären Lösung sind in Abbildung 6.17 zu sehen.

Parametersteuerung

In Abb. 6.18 ist der Konvergenzverlauf des inkonsistenten Newton-Verfahrens zu sehen. Dieser Konvergenzverlauf ist typisch für Zeitschritt-Newton-Verfahren zur Simulation von Überschallströmungen. Der Simulationsverlauf unterteilt sich in drei Phasen, die im Folgenden näher erläutert werden sollen.

In der ersten Phase der Simulation muss das Verfahren, ausgehend von einer konstanten Anfangsströmung, den Stoß im Netz bewegen, bis er ungefähr die richtige Position erreicht hat. Für die hier durchgeführte COLIBRI-Simulation erstreckt sich diese Phase bis ungefähr zur 100. Iteration. Der sich bewegende Stoß erfordert eine kleine CFL-Zahl ($CFL \leq 5$). Zur Stabilisierung des Verfahrens wird in dieser Phase eine räumliche Genauigkeit erster Ordnung in den reibungsfreien Flüssen verwendet. Der Relaxationsparameter ω wird auf 1.0 gesetzt. Während der Iterationen 1 bis 99 wird der Genauigkeitsparameter η für das lineare Gleichungssystem auf 10^{-2} gesetzt (vgl. (4.18)).

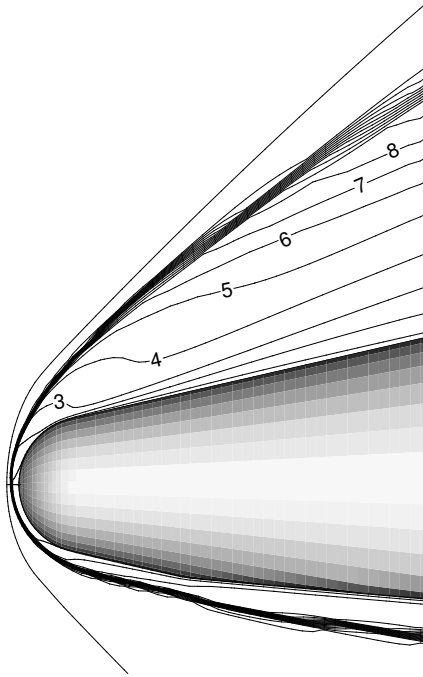


Abbildung 6.17: COLIBRI-Kapsel mit Iso-machlinien

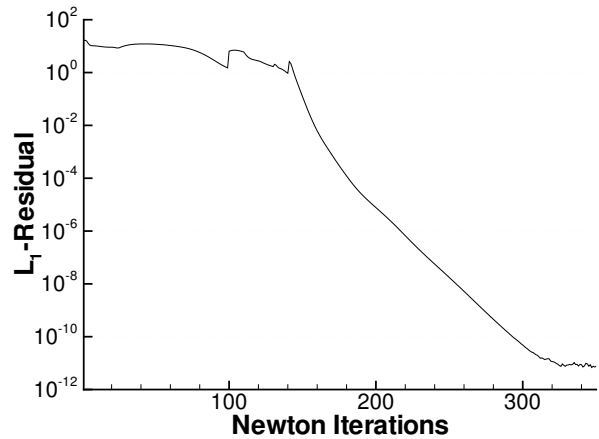


Abbildung 6.18: Konvergenzverlauf der COLIBRI-Simulation

In der zweiten Phase der Simulation werden die Strömungs- und Oberflächenmodelle auf die gewünschte Simulation eingestellt. In dieser COLIBRI-Simulation bedeutet dies, dass ab der 100. Iteration die reibungsfreien Flüsse mit einer Genauigkeit zweiter Ordnung berechnet werden und die Oberflächendiskretisierung auf zweite Ordnung umgeschaltet wird. In dieser Phase müssen sämtliche Strömungsphänomene (insbesondere die exakte Stoßposition) ausgebildet werden. Diese Rechenphase ist die sensibelste und instabilste Phase der gesamten Simulation. Der Relaxationparameter ω muss auf 0,8 herabgesetzt werden. Außerdem muss in der 100. Iteration die CFL-Zahl auf 1 herabgesetzt werden. Um jedoch die entgültige Stoßposition zu erreichen, wird die CFL-Zahl behutsam erhöht. In der hier vorgestellten COLIBRI-Simulation konnte CFL=20 eingesetzt werden. In den Iteration 100 bis 140 wird η auf 10^{-4} gesetzt.

In der dritten Phase der Simulation, wenn die Strömung schon nahe an der stationären Lösung ist, kann die CFL-Zahl deutlich weiter erhöht werden, um die Konvergenz des Verfahrens zu beschleunigen. Ab der 140. Iteration kann die CFL-Zahl auf 50, später sogar auf 500 gesetzt werden. Höhere CFL-Zahlen wären zwar möglich, führen jedoch in diesem Fall nicht zu einer schnelleren Konvergenz und machen das inkonsistente Newton-Verfahren eher instabiler. Nachdem schließlich ab der 140. Iteration der Stoß die richtige Position hat, kann der Parameter η auf 10^{-3} gesetzt werden.

Die hier gewählte CFL-Zahl-Steuerung wurde unter Verwendung des GMRES-Verfahrens optimiert, um testen zu können, ob das JLR-Verfahren dieselben Zeitschritte verträgt wie die Krylov-Raum-Verfahren.

Die beschriebene Parametersteuerung von η sollte die Konvergenz des Newton-Verfahrens nicht beeinflussen, d. h. auch durch ein genaueres Lösen der linearen Gleichungssysteme (z. B. mit $\eta = 10^{-10}$) konvergiert das Newton-Verfahren nicht schneller. Erstaunlicherweise musste für ein paar Vorkonditionierer die Wahl des Parameters η geändert werden, da das Newton-Verfahren nicht konvergierte.

Vergleich der Lösungsverfahren für das LGS

Die COLIBRI-Simulation mit der oben beschriebenen Parametersteuerung wurde mit verschiedenen Lösungsverfahren für die linearen Gleichungssysteme auf dem Vektorrechner NEC SX-5 durchgeführt. Um die verschiedenen Vorkonditionierungstechniken vergleichen zu können, musste das GMRES-Verfahren erweitert werden. Das Konvergenzkriterium des GMRES-Verfahrens benutzt das Residuum des *vorkonditionierten* LGS. Um jedoch die Rechenzeiten bei verschiedenen Vorkonditionierern vergleichen zu können, muss ein von der Vorkonditionierung unabhängiges Konvergenzkriterium eingesetzt werden. Daher wurde GMRES so erweitert, dass es für das Konvergenzkriterium das Residuum des unvorkonditionierten LGS berechnet. Dazu ist es erforderlich, in jeder GMRES-Iteration die Näherungslösung x_k und das Residuum r_k zu berechnen. Die zusätzliche Rechenzeit ist jedoch vernachlässigbar.

In Tab. 6.6 sind die Rechenzeiten aufgelistet, die für das Lösen der linearen Gleichungssysteme benötigt werden. Für den Vergleich der Gleichungssystemlöser wurden jeweils drei Rechenzeiten angegeben. Zuerst werden die Rechenzeiten für die ersten 100 Iterationen angegeben, da in dieser Simulationsphase nur kleine CFL-Zahlen eingesetzt werden können. Als zweiter Vergleichswert wird die Rechenzeit bis zur 230. Iteration aufgelistet. Bis zur 230. Iteration wird das Residuum um sechs Größenordnungen abgesenkt, was für die Bestimmung des genauen Oberflächenwärmestroms notwendig und ausreichend ist. Als letzte Rechenzeit ist jeweils die Zeit bis zur 330. Iteration angegeben. In der 330. Iteration ist die Simulation auskonvergiert, ein weiteres Konvergieren ist aufgrund der Rechengenauigkeit der Computer nicht mehr möglich.

Es sei darauf hingewiesen, dass in Tab. 6.6 nur die Rechenzeiten aufgelistet sind, die zum Lösen der linearen Gleichungssysteme benötigt werden. Die Rechenzeit für das Aufstellen der linearen Gleichungssysteme ist nicht eingeschlossen. Die CPU-Zeit für das Berechnen der reibungsfreien und viskosen Flüsse, der Quellterme und der Jacobi-Matrizen betrug für die ersten 100 Iterationen 991 s, für die Iterationen 1-230 2345 s und für die Iteration 1-330 3385 s. Die Zeitmessung erfolgte auf einem Superrechner mit geteiltem Arbeitsspeicher (*shared memory*). Durch unterschiedlich starke Auslastung des Computers können sich daher in den gemessenen Rechenzeiten kleine Abweichungen ergeben. Dies bedeutet, dass die gemessenen Rechenzeiten um 1-2% schwanken können.

In der vierten Spalte von Tabelle 6.6 wird die Berechnungsart der Vorkonditionierung angegeben. Diese Berechnung kann nichtvektorisierbar (nv), iterativ (it(l)) oder mit Hilfe von Hyperebenen (HE) erfolgen. Für die Berechnung der Vorwärts- und Rückwärtssubstitutionen bei der LUSGS- oder bei der ILU-Vorkonditionierung wird die Berechnungsart in Spalte fünf angegeben. Die Berechnung der Vorwärts- und Rückwärtssubstitutionen (VRS) kann nichtvektorisierbar (nv), mit Hilfe von abgeschnittenen Neumann-Reihen (Neumann(l)) oder mit Hilfe

	Löser	Vorkonditionierung		VRS	Zeit[s]	Zeit[s]	Zeit[s]
					100 Its.	230 Its.	330 Its.
1.	JLR	–	–	–	232	863	1354
2.	GMRES(20)	BD	–	–	–	–	–
3.	GMRES(20)	LUSGS	–	nv	778	15408	23945
4.	GMRES(20)	LUSGS	–	Neumann(10)	448	13865	20760
5.	GMRES(20)	LUSGS	–	Neumann(15)	695	15409	23829
6.	GMRES(20)	LUSGS	–	Neumann(20)	895	19230	29503
7.	GMRES(20)	LUSGS	–	HE	171	16885	3954
8.	GMRES(20)	ILU	nv	nv	5427	13348	19473
9.	GMRES(20)	ILU	nv	HE	5149	12016	17275
10.	GMRES(20)	ILUiter	it(3)	HE	350	1118	1656
11.	GMRES(20)	ILUiter	it(4)	HE	410	1210	1799
12.	GMRES(20)	ILUiter	it(5)	HE	470	1327	1970
13.	GMRES(20)	ILUiter	it(6)	HE	531	1445	2137
14.	GMRES(20)	ILUiter	it(10)	HE	746	1917	2821
15.	GMRES(20)	ILUiter	it(3)	Neumann(15)	555	2884	4462
16.	GMRES(20)	ILUiter	it(4)	Neumann(15)	620	2858	4443
17.	GMRES(20)	ILUiter	it(5)	Neumann(15)	679	2888	4447
18.	GMRES(20)	ILU	HE	Neumann(10)	–	–	–
19.	GMRES(20)	ILU	HE	Neumann(11)	408	3383	5434
20.	GMRES(20)	ILU	HE	Neumann(13)	415	3712	5936
21.	GMRES(20)	ILU	HE	Neumann(15)	452	3614	1516
22.	GMRES(20)	ILU	HE	Neumann(20)	548	2592	4254
23.	GMRES(20)	ILU	HE	HE	254	770	1166
24.	GMRES(20)	LLU	–	–	270	887	1393
25.	GMRES(20)	JLR(0)	–	–	292	987	1544
26.	GMRES(20)	JLR(1)	–	–	337	1079	1672
27.	GMRES(20)	JLR(2)	–	–	393	1192	1845
28.	GMRES(20)	JLR(3)	–	–	455	1317	2008
29.	GMRES(20)	JLR(5)	–	–	567	1557	2335
30.	GMRES(20)	JLR(10)	–	–	881	2249	3277

Tabelle 6.6: Vergleich der CPU Zeit für JLR und GMRES(20) mit verschiedenen Vorkonditionierern und verschiedenen Vektorisierungstechniken

von Hyperebenen (HE) erfolgen.

Rechenzeitvergleich

Das **JLR**-Verfahren (Zeile 1 von Tab. 6.6) ist ein sehr schnelles Verfahren. Die Befürchtung von Bemerkung 4.2, dass eventuell ein kleinerer Zeitschritt für die Konvergenz notwendig sein könnte, hat sich als falsch herausgestellt. Eine Reduktion des Zeitschrittes für die Konvergenz des JLR-Verfahrens ist nicht notwendig.

Die Blockdiagonalskalierung braucht zwar nur sehr wenig Speicherplatz, aber dafür musste die Simulation mit **GMRES-BD** (Zeile 2) wegen zu langer Rechenzeit abgebrochen werden. Für hohe CFL-Zahlen benötigte GMRES mehr als 1000 Iterationen, um das LGS in der gewünschten Genauigkeit zu lösen. Damit ist GMRES-BD kein geeignetes Verfahren für derartige Wiedereintrittssimulationen.

Der **LUSGS**-Vorkonditionierer (Zeilen 3-7) benötigt genau so viel Speicherplatz wie die Blockdiagonalskalierung. Auf einem Vektorrechner muss jedoch eine gut vektorisierbare Vorwärts- und Rückwärtssubstitution implementiert werden, welche den Speicherplatzbedarf deutlich erhöht. Die abgebrochenen Neumann-Reihen (Zeilen 4-6) ergibt kein zufriedenstellendes Ergebnis. Der Einsatz von Hyperebenen (Zeile 7) führt zum besten Ergebnis. Für kleine CFL-Zahlen ist GMRES-LUSGS das schnellste Verfahren, da die Berechnung der Vorkonditionierung wenig Rechenzeit benötigt. Für hohe CFL-Zahlen aber benötigt das GMRES-Verfahren sehr viele Iterationen, was die Gesamtsimulation deutlich verlangsamt.

Die **ILU**-Vorkonditionierung (Zeilen 8-23) ist für den Einsatz auf Vektorrechner am problematischsten. Sowohl für die Berechnung der ILU-Zerlegung, als auch für die Vorwärts- und Rückwärtssubstitutionen müssen geeignete Vektorisierungstechniken gefunden werden, da die nichtvektorisierbare Version (Zeile 8) zu viel Rechenzeit benötigt. Am schnellsten war wiederum der Einsatz des Hyperebenenformats sowohl für die ILU-Berechnung, als auch für die Vorwärts- und Rückwärtssubstitutionen (Zeile 23). Allerdings führt das Hyperebenenformat zu einem höheren Speicherplatzbedarf. Die abgebrochenen Neumann-Reihen konvergierten dagegen sehr schlecht, insbesondere mit GMRES-ILU-Neumann(l) mit $l \leq 10$ konvergierte das Newton-Verfahren gar nicht mehr. Die iterative Berechnung der ILU-Zerlegung (Zeile 10-17) ist im Vergleich zur nichtvektorisierbaren Version (Zeile 8) eine deutliche Beschleunigung, doch nicht so schnell wie GMRES-ILU im Hyperebenenformat (Zeile 23). Hierbei ist zu beachten, dass die Berechnung der iterativen ILU-Zerlegung sehr schnell ist. Werden jedoch nur wenige ILU-Iterationsschritte (Zeile 10-13) durchgeführt, so steigt die Anzahl der notwendigen GMRES-Iterationen deutlich an. Erst bei GMRES-ILUiter(10) (Zeile 14) benötigt GMRES ungefähr gleich viele Iterationen wie GMRES-ILU.

Der in dieser Arbeit neu vorgestellte **LLU**-Vorkonditionierer (Zeile 24) benötigt viel Speicherplatz, ist aber auf einem Vektorrechner einer der schnellsten Vorkonditionierer.

Die Vorkonditionierung mit dem **JLR**-Verfahren (Zeilen 25-30) reduziert zwar die Anzahl der GMRES-Iterationen deutlich, dafür ist jedoch das JLR-Verfahren als Vorkonditionierer viel zu aufwändig.

	Löser	Vorkonditionierer		VRS	Zeit [s] 100 Its.	Zeit [s] 230 Its.	Zeit [s] 330 Its.
31.	GMRES(20)	ILU	HE	Neumann(3)	520	3046	4682
32.	GMRES(20)	ILU	HE	Neumann(4)	573	3021	4599
33.	GMRES(20)	ILU	HE	Neumann(5)	493	2920	4516
34.	GMRES(20)	ILU	HE	Neumann(6)	522	2982	4601
35.	GMRES(20)	ILU	HE	Neumann(10)	557	3307	5241
36.	GMRES(20)	ILU	HE	HE	278	771	1158

Tabelle 6.7: CPU-Zeit für GMRES-ILU-Neumann(1), Genauigkeit der LGS-Lösung während der ersten 100 Iterationen hochgesetzt

Zusammenfassend kann gesagt werden, dass für GMRES die besten Vorkonditionierer LLU und ILU sind, wobei für die ILU-Vorkonditionierung das Hyperebenenformat gewählt werden sollte. Das JLR-Verfahren ist ebenso ein sehr schnelles Verfahren und daher für strukturierte Netze eine gleichwertige Alternative zu GMRES-ILU und GMRES-LLU.

Da die abgebrochenen Neumann-Reihen bei der ILU-Vorkonditionierung ein instabiles Verhalten zeigen, werden die GMRES-ILU-Neumann(l)-Verfahren nochmals getestet, dieses Mal wird die Genauigkeit der LGS-Lösung während der ersten 100 Iterationen auf 10^{-4} gesetzt. Die Ergebnisse sind in Tab. 6.7 dargestellt. Der Einsatz von Neumann-Reihen ist also möglich, aber im Vergleich zu anderen Vektorisierungstechniken zu langsam.

Rechenzeitvergleich der Krylov-Raum-Verfahren

Bisher wurde ausschließlich das GMRES(20) Verfahren mit dem JLR-Verfahren verglichen. Für den Vergleich der Krylov-Raum-Verfahren TFQMR, QMRCGStab, CGS, BiCGStab und GMRES werden die beiden Vorkonditionierer LLU und ILU eingesetzt. Zusätzlich wird das GMRES-Verfahren mit verschiedenen *restart-levels* getestet.

Wie in Tabelle 6.8 zu sehen ist, sind alle Krylov-Raum-Verfahren mit den ausgewählten Vorkonditionierern einsetzbar. Auch das von Jonas in [38] getestete CGS-Verfahren konnte in dieser Arbeit durch den Einsatz einer geeigneten Vorkonditionierung erfolgreich für Simulationen genutzt werden. Die Unterschiede in der CPU-Zeit sind sehr klein im Vergleich zu den Unterschieden, die durch verschiedene Vorkonditionierungstechniken entstehen (vgl. Tab. 6.6 mit Tab. 6.8). Dies liegt vor allem an den für diesen Vergleich ausgewählten hervorragenden Vorkonditionierern LLU und ILU. Somit ist die Wahl des Krylov-Raum-Verfahrens nicht so wichtig wie die Wahl der Vorkonditionierung.

Konsistentes Newton-Verfahren

Anhand der COLIBRI-Simulation sollen die Grenzen des konsistenten Newton-Verfahrens aufgezeigt werden. Für die Konvergenzuntersuchungen wird die vorige COLIBRI-Simulation ver-

	Löser	Zeit [s] 100 Its.	Zeit [s] 230 Its.	Zeit [s] 330 Its.
37.	GMRES(5)-ILU	216	640	994
38.	GMRES(10)-ILU	213	622	969
39.	GMRES(15)-ILU	216	621	955
40.	GMRES(20)-ILU	216	618	949
41.	CGS-ILU	232	727	1111
42.	BiCGStab-ILU	226	626	958
43.	TFQMR-ILU	233	715	1095
44.	QMRCGStab-ILU	225	638	975
45.	GMRES(5)-LLU	223	723	1173
46.	GMRES(10)-LLU	222	701	1118
47.	GMRES(15)-LLU	212	701	1130
48.	GMRES(20)-LLU	223	695	1103
49.	CGS-LLU	230	851	1350
50.	BiCGStab-LLU	227	725	1139
51.	TFQMR-LLU	230	835	1343
52.	QMRCGstab-LLU	226	733	1159

Tabelle 6.8: Rechenzeitvergleich der Krylov-Raum-Verfahren

einfacht, in dem eine konstante Oberflächentemperatur vorgegeben wird. Dadurch soll der Einfluss der komplexen Oberflächenrandbedingung auf die Konvergenz so weit wie möglich reduziert werden.

Die Simulation wurde mit der zuvor beschriebenen Parametersteuerung durchgeführt. Eingesetzt wurden bei dieser Simulation vier verschiedene Limiter:

- van-Albada-Limiter
- MC-Limiter mit $\kappa = 1,0$
- MC-Limiter mit $\kappa = 1,2$
- MC-Limiter mit $\kappa = 1,5$

Die Konvergenzverläufe des Simulationsverfahrens für diese vier Limiter sind in Abb. 6.19 dargestellt. Alle vier Simulationen konnten problemlos mit dem inkonsistenten Newton-Verfahren durchgeführt werden. Lediglich die Simulation mit dem MC-Limiter mit $\kappa = 1,5$ zeigt ein paar Instabilitäten.

Dieselben Simulationen mit identischer Parametersteuerung für ω und CFL-Zahl wurden mit dem konsistenten Newton-Verfahren mit *exakter* Matrix-Vektor-Multiplikation wiederholt. Die Konvergenzverläufe sind in Abb. 6.20 dargestellt. Sämtliche Simulationen brechen nach wenigen

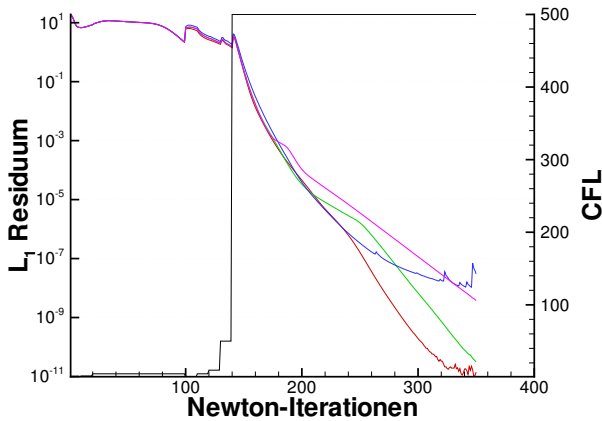


Abbildung 6.19: Konvergenzverläufe für vier Limiter, inkonsistentes Newton-Verfahren

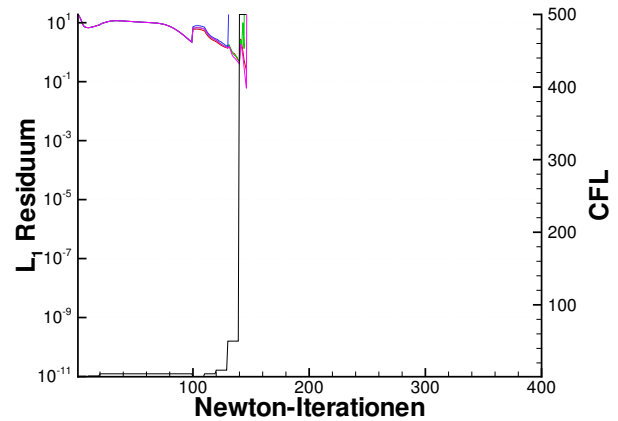


Abbildung 6.20: Konvergenzverläufe für vier Limiter, konsistentes Newton-Verfahren

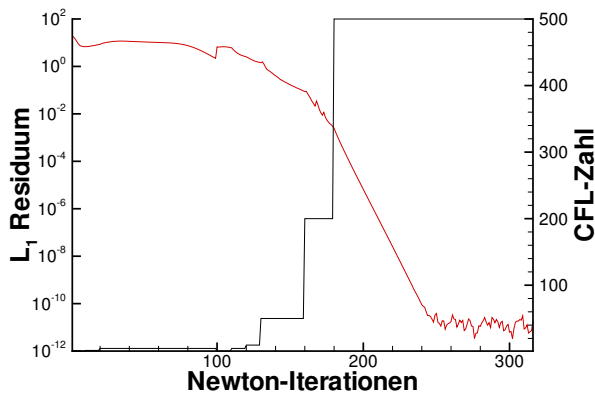
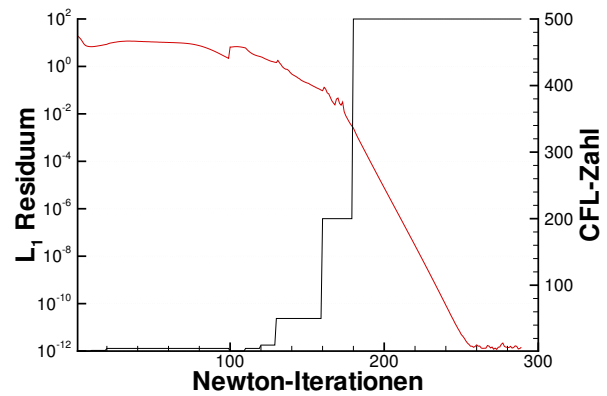


Abbildung 6.21: Konvergenzverläufe für van-Albada-Limiter mit konsistentem Newton-Verfahren

Abbildung 6.22: Konvergenzverläufe für MC-Limiter mit $\kappa = 1,2$ und konsistentem Newton-Verfahren

Iterationen ab. Dieses Phänomen konnte bei allen Hyperschallsimulationen mit *konsistentem* Newton-Verfahren beobachtet werden. Um ein stabiles Verfahren zu erhalten, darf die CFL-Zahl nicht so schnell erhöht werden wie im inkonsistenten Newton-Verfahren. Bei diesem verhältnismäßig einfachen Strömungsfall reichte eine kleine Änderung in der CFL-Zahl Steuerung aus, um das Verfahren für den van-Albada-Limiter und für den MC-Limiter mit $\kappa = 1,2$ zur Konvergenz zu bringen (siehe Abb. 6.21 und 6.22).

Für den MC-Limiter mit $\kappa = 1,0$ und $\kappa = 1,5$ bleibt das konsistente Verfahren sehr instabil und konvergiert nicht mehr so weit wie mit dem inkonsistenten Newton-Verfahren (Abb. 6.23). Diese Tatsache ist in sofern erstaunlich, da der MC-Limiter mit $\kappa = 1,0$ die stabilste Strömungsmodellierung mit einer räumlichen Genauigkeit zweiter Ordnung darstellt. Doch beim Einsatz des konsistenten Newton-Verfahrens verhindert dieser Limiter eine vollständige Konvergenz. Für den MC-Limiter mit $\kappa = 1,5$ ist eine Konvergenz des Newton-Verfahrens gar nicht mehr möglich. Wie in Abb. 6.24 zu sehen ist, bricht das Verfahren selbst bei einer maximalen CFL-Zahl

von 20 ab. Es sei nochmals darauf hingewiesen, dass für die beiden Limiterparameter $\kappa = 1,0$ und $\kappa = 1,5$ das *inkonsistente* Newton-Verfahren konvergiert!

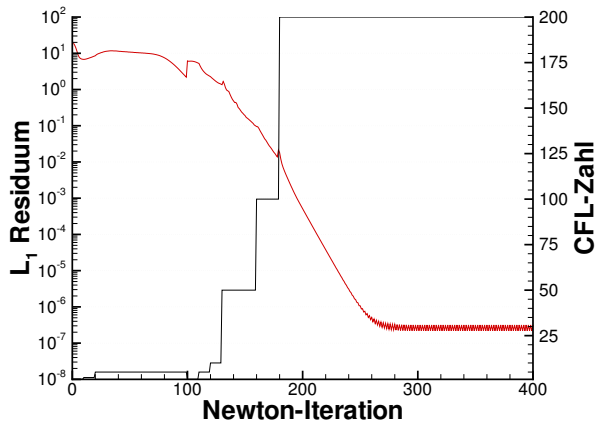


Abbildung 6.23: Konvergenzverläufe für den MC-Limiter mit $\kappa = 1,0$ mit konsistentem Newton-Verfahren, keine vollständige Konvergenz

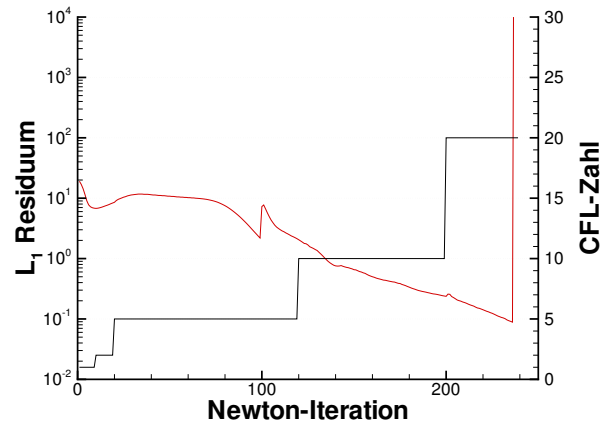


Abbildung 6.24: Konvergenzverläufe für MC-Limiter mit $\kappa = 1,5$ und konsistentem Newton-Verfahren, Abbruch des Verfahrens trotz $CFL_{max} = 20$

Schon anhand dieser einfachen COLIBRI-Simulation wird deutlich, dass das konsistente Newton-Verfahren nicht zur gewünschten quadratischen Konvergenz führt. Die Konvergenz hängt sehr stark von den eingesetzten Limitern ab. Das konsistente Verfahren ist instabiler als das inkonsistente.

Abschließende Bewertung

Die Testrechnungen ergaben, dass das GMRES-Verfahren mit den Vorkonditionierern ILU oder LLU sowie das JLR-Verfahren die schnellsten Verfahren sind. Das JLR-Verfahren ist demnach für dieses Verfahren ein sehr effizientes Lösungsverfahren.

Der Vergleich der konsistenten Newton-Verfahren machte deutlich, dass die Konsistenzsteigerung nicht zwingend zu einer Konvergenzsteigerung führt. Stattdessen zeigte das konsistente Newton-Verfahren mit der exakten Matrix-Vektor-Multiplikation deutliche Instabilitäten, die durch kleinere CFL-Zahlen und einem Relaxationsparameter stabilisiert werden muss. Somit ist für Hyperschallströmungen mit starken Stößen ein konsistentes Newton-Verfahren mit quadratischer Konvergenz nicht möglich. Das stabilste Verfahren für Hyperschallströmungen im thermochemischen Nichtgleichgewicht ist und bleibt das inkonsistente Newton-Verfahren.

6.5 Umströmung der X-38

Das vorgestellte Verfahren ist mittlerweile so ausgereift und robust genug, so dass es für große Anwendungsrechnungen eingesetzt werden kann. Die für das TETRA-Programm von Infed gerechneten X38-Simulationen (siehe Abb. 6.25) wurden mit dem in dieser Arbeit vorgestellten Verfahren durchgeführt. Hierfür wurden weitere Modelle und Verfahren eingesetzt, die außerhalb der hier vorgestellten Arbeit in das Simulationsprogramm integriert und deshalb nicht in der vorliegenden Arbeit beschrieben wurden. Die abgebildete Simulation wurde auf einem Rechnernetz mit 888832 Zellen durchgeführt, welches von der NASA zur Verfügung gestellt wurde.

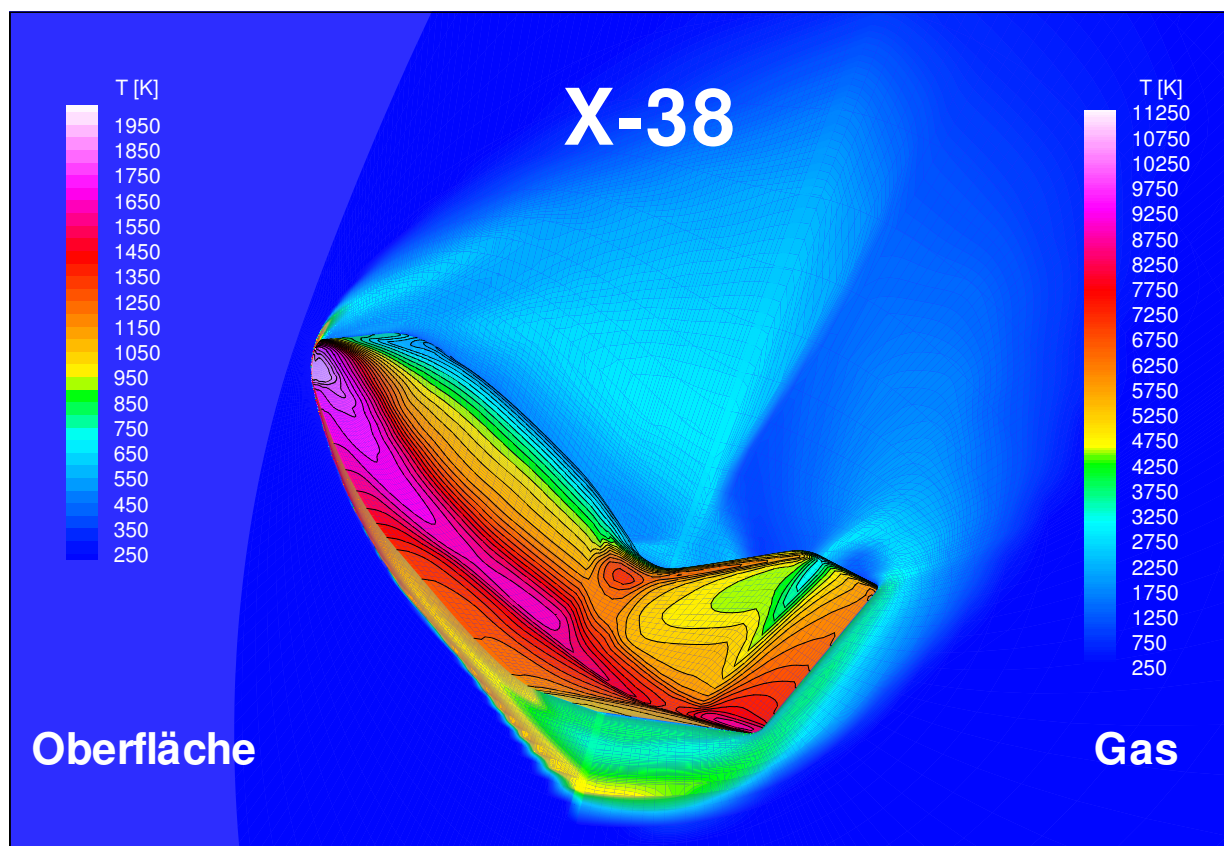


Abbildung 6.25: Ergebnisse einer X-38 Simulation mit einem Rechnernetz aus 888832 Zellen

Die genauen Auswertungen dieser Simulation sind in [22], [31] und [32] veröffentlicht.

Abschließende Bewertung

Das Simulationsverfahren mit der in dieser Arbeit vorgestellten Numerik ist für großen Anwendungsrechnungen geeignet.

Kapitel 7

Zusammenfassung

In der vorliegenden Arbeit wurde ein Simulationsverfahren für dreidimensionale Über- und Hyperschallströmungen im thermochemischen Nichtgleichgewicht entwickelt, mit welchem die Stömung um Raumfahrzeuge beim Wiedereintritt aus einem niedrigen Erdorbit in die Erdatmosphäre berechnet werden kann. Dazu wurde ein Luftmodell mit den fünf Komponenten N_2 , O_2 , NO , N und O sowie den vier Temperaturen T_{tr} , T_{vib,N_2} , T_{vib,O_2} und $T_{vib,NO}$ verwendet, für welches die instationären Navier-Stokes-Gleichungen aufgestellt wurden.

Die Navier-Stokes-Gleichungen werden in Finite-Volumen-Diskretisierung auf strukturierten Netzen gelöst. Für die Modellierung der reibungsfreien Flüsse wurden der approximative Riemann-Löser nach Roe und das AUSMDV-Verfahren eingesetzt. Vergleichsrechnungen ergaben, dass das AUSMDV-Verfahren robuster und dissipationsärmer als das Roe-Verfahren ist. Mit dem AUSMDV-Verfahren konnte damit ein robustes und zugleich sehr genaues Verfahren zur Berechnung der reibungsfreien Flüsse gefunden werden. Die viskosen Flüsse wurden standardmäßig mittels Finite Differenzen in das Verfahren integriert.

Für die Berechnung der stationären Lösung wird ein inkonsistentes Zeitschritt-Newton-Verfahren eingesetzt, welches bei sachgemäßer Strömungsmodellierung eine Reduktion des Residuums bis auf Maschinengenauigkeit garantiert. Die in jedem Iterationsschritt entstehenden linearen Gleichungssysteme werden iterativ gelöst. Als Lösungsverfahren wurde das von Schöll und Jonas entwickelte Jacobi-Linienrelaxationsverfahren sowie die Krylov-Raum-Verfahren CGS, BiCG-Stab, TFQMR, QMRGStab und GMRES(l) eingesetzt und miteinander verglichen. Als Vorkonditionierer für die Krylov-Raum-Verfahren wurde eine Blockdiagonalskalierung, eine ILU-Vorkonditionierung, sowie eine LUSGS-Vorkonditionierung getestet. Ebenso wurden aus der dem Jacobi-Linienrelaxationsverfahren die Vorkonditionierer JLR(l) und LLU abgeleitet.

Bei der Entwicklung der Vorkonditionierungstechniken wurde großer Wert auf die Vektorsierbarkeit gelegt. Daher wurden verschiedene Techniken getestet, mit welchen die ILU- und LUSGS-Vorkonditionierer auf Vektorrechner optimaler eingesetzt werden können. Bei den Vorkonditionierern waren ILU und LLU die besten Methoden. Das LUSGS-Verfahren ist jedoch auf PCs mit kleinem Arbeitsspeicher durchaus eine alternative Vorkonditionierung, bei der jedoch ggf. kleinere CFL-Zahlen in Kauf genommen werden müssen.

Desweiteren wurden Konvergenzanalysen durchgeführt mit dem Ziel, Möglichkeiten für eine Konvergenzbeschleunigung zu finden. Dabei wurde versucht, das inkonsistente Newton-Verfahren zu einem konsistenten Verfahren zu erweitern, was bisher nur für Idealgasströmungen mit $Ma \leq 2$ in der Literatur zu finden ist. Ausgehend von einem matrixfreien Newton-Krylov-Verfahren wurden weitere, stabilere Methoden entwickelt, um das konsistente Newton-Verfahren auch auf Hyperschallströmungen anwenden zu können. Durch die aufwändige Entwicklung einer exakten Matrix-Vektor-Multiplikation wurde deutlich, dass ein reines Newton-Verfahren mit $CFL \rightarrow \infty$ und $\omega = 1,0$ für Hyperschallströmungen im thermochemischen Nichtgleichgewicht nicht garantiert werden kann. Zur Stabilisierung des Verfahrens ist daher stets ein Zeitschritt $CFL < 1000$ und ein Relaxationsparameter $\omega < 0,8$ notwendig.

Anhang A

Fluss-Jacobi-Matrizen

Der für die Strömungsmodellierung zugrundeliegende Erhaltungsvektor lautet

$$Q = \begin{pmatrix} \rho_s \\ \rho u \\ \rho v \\ \rho w \\ \rho e_t \\ \rho_{s'} e_{vib,s'} \end{pmatrix}.$$

Der reibungsfreie Flussvektor über eine Zellfläche mit Normalenvektor (n_x, n_y, n_z) lautet

$$\varphi = \begin{pmatrix} \rho_s u_\perp \\ \rho u_\perp u + p n_x \\ \rho u_\perp v + p n_y \\ \rho u_\perp w + p n_z \\ \rho u_\perp h_t \\ u_\perp \rho_{s'} e_{vib,s} \end{pmatrix}$$

mit $s = 1, \dots, 5$ und $s' = 1, \dots, 3$. Hierbei ist u_\perp die Normalgeschwindigkeit bzgl. der Zellfläche. Für das Roe-Verfahren wird die Jacobi-Matrix

$$\frac{\partial \varphi}{\partial Q} \tag{A.1}$$

und deren Zerlegung

$$\frac{\partial \varphi}{\partial Q} = R \Lambda R^{-1}. \tag{A.2}$$

benötigt. Hierbei soll Λ eine Diagonalmatrix sein. Diese Zerlegung kann für obigen Flussvektor vollständig analytisch durchgeführt werden.

Roe-Matrizen

$$\frac{\partial \varphi}{\partial Q} = \begin{pmatrix} u_{\perp}(1-\xi_1) & -\xi_1 u_{\perp} & -\xi_1 u_{\perp} & -\xi_1 u_{\perp} & -\xi_1 u_{\perp} & \xi_1 n_x & \xi_1 n_y & \xi_1 n_z & 0 & 0 & 0 & 0 \\ -\xi_2 u_{\perp} & u_{\perp}(1-\xi_2) & -\xi_2 u_{\perp} & -\xi_2 u_{\perp} & -\xi_2 u_{\perp} & \xi_2 n_x & \xi_2 n_y & \xi_2 n_z & 0 & 0 & 0 & 0 \\ -\xi_3 u_{\perp} & -\xi_3 u_{\perp} & u_{\perp}(1-\xi_3) & -\xi_3 u_{\perp} & -\xi_3 u_{\perp} & \xi_3 n_x & \xi_3 n_y & \xi_3 n_z & 0 & 0 & 0 & 0 \\ -\xi_4 u_{\perp} & -\xi_4 u_{\perp} & -\xi_4 u_{\perp} & u_{\perp}(1-\xi_4) & -\xi_4 u_{\perp} & \xi_4 n_x & \xi_4 n_y & \xi_4 n_z & 0 & 0 & 0 & 0 \\ -\xi_5 u_{\perp} & -\xi_5 u_{\perp} & -\xi_5 u_{\perp} & -\xi_5 u_{\perp} & u_{\perp}(1-\xi_5) & \xi_5 n_x & \xi_5 n_y & \xi_5 n_z & 0 & 0 & 0 & 0 \\ \beta_{2,1} n_x - u_{\perp} & \beta_{2,2} n_x - u_{\perp} & \beta_{2,3} n_x - u_{\perp} & \beta_{2,4} n_x - u_{\perp} & \beta_{2,5} n_x - u_{\perp} & u_{\perp} + u n_x (1-\beta_1) & u n_y - \beta_1 v n_x & u n_z - \beta_1 w n_x & n_x \beta_1 & -n_x \beta_1 & -n_x \beta_1 & -n_x \beta_1 \\ \beta_{2,1} n_y - v_{\perp} & \beta_{2,2} n_y - v_{\perp} & \beta_{2,3} n_y - v_{\perp} & \beta_{2,4} n_y - v_{\perp} & \beta_{2,5} n_y - v_{\perp} & v n_x - \beta_1 u n_y & u_{\perp} + v n_y (1-\beta_1) & v n_z - \beta_1 w n_y & n_y \beta_1 & -n_y \beta_1 & -n_y \beta_1 & -n_y \beta_1 \\ \beta_{2,1} n_z - w_{\perp} & \beta_{2,2} n_z - w_{\perp} & \beta_{2,3} n_z - w_{\perp} & \beta_{2,4} n_z - w_{\perp} & \beta_{2,5} n_z - w_{\perp} & w n_x - \beta_1 u n_z & w n_y - \beta_1 v n_z & u_{\perp} + w n_z (1-\beta_1) & n_z \beta_1 & -n_z \beta_1 & -n_z \beta_1 & -n_z \beta_1 \\ u_{\perp}(\beta_{2,1} - h_t) & u_{\perp}(\beta_{2,2} - h_t) & u_{\perp}(\beta_{2,3} - h_t) & u_{\perp}(\beta_{2,4} - h_t) & u_{\perp}(\beta_{2,5} - h_t) & h_t n_x - u_{\perp} \beta_1 u & h_t n_y - u_{\perp} \beta_1 v & h_t n_z - u_{\perp} \beta_1 w & u_{\perp}(1+\beta_1) & -u_{\perp} & -u_{\perp} & -u_{\perp} \\ -u_{\perp} e_{vib,1} \xi_1 & -u_{\perp} e_{vib,1} \xi_1 & -u_{\perp} e_{vib,1} \xi_1 & -u_{\perp} e_{vib,1} \xi_1 & -u_{\perp} e_{vib,1} \xi_1 & \xi_1 e_{vib,1} n_x & \xi_1 e_{vib,1} n_y & \xi_1 e_{vib,1} n_z & 0 & u_{\perp} & 0 & 0 \\ -u_{\perp} e_{vib,2} \xi_2 & -u_{\perp} e_{vib,2} \xi_2 & -u_{\perp} e_{vib,2} \xi_2 & -u_{\perp} e_{vib,2} \xi_2 & -u_{\perp} e_{vib,2} \xi_2 & \xi_2 e_{vib,2} n_x & \xi_2 e_{vib,2} n_y & \xi_2 e_{vib,2} n_z & 0 & 0 & u_{\perp} & 0 \\ -u_{\perp} e_{vib,3} \xi_3 & -u_{\perp} e_{vib,3} \xi_3 & -u_{\perp} e_{vib,3} \xi_3 & -u_{\perp} e_{vib,3} \xi_3 & -u_{\perp} e_{vib,3} \xi_3 & \xi_3 e_{vib,3} n_x & \xi_3 e_{vib,3} n_y & \xi_3 e_{vib,3} n_z & 0 & 0 & 0 & u_{\perp} \end{pmatrix}$$

$$R = \begin{pmatrix} \xi_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_1 \\ \xi_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_2 \\ \xi_3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_3 \\ \xi_4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \xi_4 \\ \xi_5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \xi_5 \\ u - c n_x & u & u & u & u & u & c b_x^1 & c b_x^2 & 0 & 0 & 0 & u + c n_x \\ v - c n_y & v & v & v & v & v & c b_y^1 & c b_y^2 & 0 & 0 & 0 & v + c n_y \\ w - c n_z & w & w & w & w & w & c b_z^1 & c b_z^2 & 0 & 0 & 0 & w + c n_z \\ h_t - c u_{\perp} & \langle \vec{u}, \vec{u} \rangle - \frac{\beta_{2,1}}{\beta_1} & \langle \vec{u}, \vec{u} \rangle - \frac{\beta_{2,2}}{\beta_1} & \langle \vec{u}, \vec{u} \rangle - \frac{\beta_{2,3}}{\beta_1} & \langle \vec{u}, \vec{u} \rangle - \frac{\beta_{2,4}}{\beta_1} & \langle \vec{u}, \vec{u} \rangle - \frac{\beta_{2,5}}{\beta_1} & c \langle \vec{u}, \vec{b}^1 \rangle & c \langle \vec{u}, \vec{b}^2 \rangle & 1 & 1 & 1 & h_t + c u_{\perp} \\ \xi_1 e_{vib,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \xi_1 e_{vib,1} \\ \xi_2 e_{vib,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \xi_2 e_{vib,2} \\ \xi_3 e_{vib,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \xi_3 e_{vib,3} \end{pmatrix}$$

$$R^{-1} = \frac{1}{c^2} \begin{pmatrix} \frac{1}{2}(\beta_{2,r} + c u_{\perp}) & -\frac{1}{2}(c n_x + \beta_1 u) & -\frac{1}{2}(c n_y + \beta_1 v) & -\frac{1}{2}(c n_z + \beta_1 w) & \frac{\beta_1}{2} & -\frac{\beta_1}{2} & -\frac{\beta_1}{2} & -\frac{\beta_1}{2} \\ \delta_{rs} c^2 - \xi_s \beta_{2,r} & \xi_s \beta_1 u & \xi_s \beta_1 v & \xi_s \beta_1 w & -\xi_s \beta_1 & \xi_s \beta_1 & \xi_s \beta_1 & \xi_s \beta_1 \\ -c \langle \vec{u}, \vec{b}^1 \rangle & c b_x^1 & c b_y^1 & c b_z^1 & 0 & 0 & 0 & 0 \\ -c \langle \vec{u}, \vec{b}^2 \rangle & c b_x^2 & c b_y^2 & c b_z^2 & 0 & 0 & 0 & 0 \\ -\xi_1 e_{vib,1} \beta_{2,r} & \xi_1 e_{vib,1} \beta_1 u & \xi_1 e_{vib,1} \beta_1 v & \xi_1 e_{vib,1} \beta_1 w & -\xi_1 e_{vib,1} \beta_1 & c^2 + \xi_1 e_{vib,1} \beta_1 & \xi_1 e_{vib,1} \beta_1 & \xi_1 e_{vib,1} \beta_1 \\ -\xi_2 e_{vib,2} \beta_{2,r} & \xi_2 e_{vib,2} \beta_1 u & \xi_2 e_{vib,2} \beta_1 v & \xi_2 e_{vib,2} \beta_1 w & -\xi_2 e_{vib,2} \beta_1 & \xi_2 e_{vib,2} \beta_1 & c^2 + \xi_2 e_{vib,2} \beta_1 & \xi_2 e_{vib,2} \beta_1 \\ -\xi_3 e_{vib,3} \beta_{2,r} & \xi_3 e_{vib,3} \beta_1 u & \xi_3 e_{vib,3} \beta_1 v & \xi_3 e_{vib,3} \beta_1 w & -\xi_3 e_{vib,3} \beta_1 & \xi_3 e_{vib,3} \beta_1 & \xi_3 e_{vib,3} \beta_1 & c^2 + \xi_3 e_{vib,3} \beta_1 \\ \frac{1}{2}(\beta_{2,r} - c u_{\perp}) & \frac{1}{2}(c n_x - \beta_1 u) & \frac{1}{2}(c n_y - \beta_1 v) & \frac{1}{2}(c n_z - \beta_1 w) & \frac{\beta_1}{2} & -\frac{\beta_1}{2} & -\frac{\beta_1}{2} & -\frac{\beta_1}{2} \end{pmatrix}$$

$r =$ Spaltenindex ($r = 1, \dots, 5$), $s =$ Zeilenindex ($s = 1, \dots, 5$),

$$\Lambda = \text{diag}\{u_{\perp} - c, u_{\perp}, \dots, u_{\perp}, u_{\perp} + c\}$$

Anhang B

Pseudocodes der Gleichungslöser

In diesem Abschnitt sind die Pseudocodes für die Krylov-Raum-Verfahren CGS, BiCGStab, TFQMR und QMRCGStab aufgelistet.

CGS-Algorithmus

$x_0; k = 0; \tilde{r}_0 = r_0 = b - Ax_0; q_0 = 0; p_{-1} = 0;$

while $\|r_k\| > \epsilon$ **do**

if $k = 0$

$\beta_k = \langle \tilde{r}_0, r_k \rangle;$

else

$$\beta_k = \frac{\langle \tilde{r}_0, r_k \rangle}{\langle \tilde{r}_0, r_{k-1} \rangle}$$

$u_k = r_k + \beta_k q_k$

$p_k = u_k + \beta_k (q_k + \beta_k p_{k-1})$

$$\alpha_k = \frac{\langle \tilde{r}_0, r_k \rangle}{\langle \tilde{r}_0, Ap_k \rangle}$$

$q_{k+1} = u_k - \alpha_k Ap_k$

$r_{k+1} = r_k - \alpha_k A(u_k + q_{k+1})$

$x_{k+1} = x_k - \alpha_k (u_k + q_{k+1})$

$k = k + 1;$

end

$x = x_k$

Algorithmus B.1: CGS

BiCGSTAB-Algorithmus

$$x_0; k = 0; r_0 = b - Ax_0;$$

$$q_0 = 0;$$

$$\rho_0 = 1; \alpha = 1; \omega_0 = 1;$$

$$v_0 = 0; p_0 = 0$$

while $\|r_k\| > \epsilon$ **do**

$$k = k + 1;$$

$$\rho_k = \langle \tilde{r}_0, r_{k-1} \rangle$$

$$\beta = \frac{\rho_k}{\rho_{k-1}} \frac{\alpha}{\omega_{k-1}}$$

$$p_k = r_{k-1} + \beta(p_{k-1} - \omega_{k-1}v_{k-1})$$

$$v_k = Ap_k$$

$$\alpha = \frac{\rho_k}{\langle \tilde{r}_0, v_k \rangle}$$

$$s = r_{k-1} - \alpha v_k$$

$$t = As$$

$$\omega = \frac{\langle t, s \rangle}{\langle t, t \rangle}$$

$$x_k = x_{k-1} + \alpha p_k + \omega_k s$$

$$r_k = s - \omega_k t$$

end

Algorithmus B.2: BiCGStab

TFQMR-Algorithmus

$$x_0; w_1 = y_1 = r_0 = b - Ax_0; \tau = \|r_0\|$$

$$v_0 = Ay_1; d_0 = 0; \eta_0 = \theta_0 = 0$$

for $j = 1, 2, 3, \dots$

$$\alpha_{j-1} = \frac{\langle w_{2j-1}, r_0 \rangle}{\langle v_{j-1}, r_0 \rangle}$$

$$y_{2j} = y_{2j-1} - \alpha_{j-1}v_{j-1}$$

for $m = 2j - 1, 2j$

$$w_{m+1} = w_m - \alpha_{j-1}Ay_m$$

$$\theta_m = \frac{\|w_{m+1}\|}{\tau_{m-1}}$$

$$c_m = (1 + \theta_m^2)^{-1/2}$$

$$d_m = y_m + \frac{\theta_{m-1}^2 \eta_{m-1}}{\alpha_{j-1}} d_{m-1}$$

$$\eta_m = c_m^2 \alpha_{j-1}$$

$$x_m = x_{m-1} + \eta_m d_m$$

$$\tau_m = \tau_{m-1} \theta_m c_m$$

if $\sqrt{m+1} \tau_m < \epsilon$ **then STOP**

next m

$$\beta_{j-1} = \frac{\langle w_{2j+1}, r_0 \rangle}{\langle w_{2j-1}, r_0 \rangle}$$

$$y_{2j+1} = w_{2j+1} + \beta_{j-1}y_{2j}$$

$$v_j = Ay_{2j+1} + \beta_{j-1}(Ay_{2j} + \beta_{j-1}v_{j-1})$$

next j

Algorithmus B.3: TFQMR

QMRCGStab-Algorithmus

$$x_0; j = 0; r_0 = b - Ax_0;$$

$$p = r_0; \tau = \|r_0\|; v = Ap$$

$$d_0 = 0, \eta = 0; \theta = 0$$

while $\|r_j\| > \epsilon$ **do**

$$j = j + 1$$

$$\alpha = \frac{\langle r_{j-1}, r_0 \rangle}{\langle v, r_0 \rangle}$$

$$s = r - \alpha v$$

$$\theta = \frac{\|s\|}{\tau}$$

$$c = \frac{1}{\sqrt{1 + \theta^2}}$$

$$d = p + \frac{\theta^2 \eta}{\alpha} \cdot d$$

$$\eta = c^2 \alpha$$

$$x = x + \eta \cdot d$$

$$\tau = \tau \theta c$$

$$\omega = \frac{\langle As, s \rangle}{\langle As, As \rangle}$$

$$r = s - \omega As$$

$$\theta = \frac{\|r\|}{\tau}$$

$$c = \frac{1}{\sqrt{1 + \theta^2}}$$

$$d = s + \frac{\theta^2 \eta}{\omega} \cdot d$$

$$\eta = c^2 \omega$$

$$x = x + \eta \cdot d$$

$$\tau = \tau \theta c$$

$$\beta = \frac{\alpha \langle r_j, r_0 \rangle}{\omega \langle r_{j-1}, r_0 \rangle}$$

$$p = r + \beta(p - \omega Ap)$$

$$v = Ap$$

end

Algorithmus B.4: QMRCGStab

Literaturverzeichnis

- [1] R. ABGRALL, *Hypersonic calculations by Riemann solvers technique*, Computer Physics Communications 65, 1-7, 1991
- [2] P. ADAMI, V. MICHELASSI, F. MARTELLI, *Performance of a Newton-Krylov Scheme Against Implicit and Multigrid Solvers for Inviscid Flows*, AIAA-98-2429, 1998
- [3] W. K. ANDERSON, R. D. RAUSCH, D. L. BOHNAUS, *Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids*, AIAA-95-1740, 1995
- [4] J. D. ANDERSON, JR., *Computational Fluid Dynamics*, McGraw-Hill, 1995
- [5] M. AUWETER-KURTZ, J. BURKHARDT, M. FERTIG, H.-H. FRÜHAUF, H. HABIGER, G. JAHN, E. MESSERSCHMID, U. SCHÖTTLE, *Flugdatenauswertung der MIRKA-Experimente HEATIN und PYREX*, Proceedings der DGLR Jahrestagung, Bremen, 5.-8. Oktober, 1998
- [6] R. BARRETT, M. BERRY, T. F. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. EIJKHOUT, R. POZO, C. ROMINE AND H. VAN DER VORST, *Templates for the solution of linear systems: building blocks for iterative methods*, SIAM, Philadelphia, 1994
<http://netlib2.cs.utk.edu/templates>
- [7] T. J. BARTH, *Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms*, AIAA 87-0595, 1987
- [8] T. J. BARTH, S. LINTON, *An Unstructured Mesh Newton Solver for Compressible Fluid Flow and its Parallel Implementation*, AIAA-95-0221, 33rd Aerospace Sciences Meeting & Exhibit, Kanuary 9-12, Reno, NV, 1995
- [9] M. A. BUSHY, P. CINNELLA, *Steps Toward more Accurate and Efficient Simulations of Reactive Flows*, AIAA-98-2425, 1998
- [10] T. F. CHAN, H. A. VAN DER VORST, *Approximate and Incomplete Factorizations*, in: D. E. Keyes, A. Sameh, V. Venkatakrishnan (Eds.), *Parallel Numerical Algorithms*, Vol. 4 of ICASE/LaRC Interdisciplinary Series in Science and Engineering, Kluwer Academic Publishers, Dordrecht, pp. 167-202, 1997

-
- [11] R. CHOQUET, *Accélération de l'algorithme de Newton-GMRES pour les équations de Navier-Stokes*, publication interne N° 831, institut national de recherche en informatique et en automatique, 1994
- [12] R. CHOQUET, *A matrix-free preconditioner applied to CFD*, rapport de recherche N° 2605, institut national de recherche en informatique et en automatique, 1995
- [13] A. DAGAN, *A Convergence Accelerator of Linear System of Equations Based Upon the Power Method*, AIAA-2000-2256, 2000
- [14] A. DAISS, H.-H. FRÜHAUF, E. W. MESSERSCHMID, *Modeling of Catalytic Reactions on Silicia Surfaces with Consideration of Slip Effects*, AIAA-1996-1903, 1996
- [15] G. DEGREGZ, D. V. ABEELE, P. BARBANTE, B. BOTTIN, *Numerical Simulation of Inductively Coupled Plasma Flows and Hypersonic (Re-)Entry Flows*, European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2000, Barcelona, 11.-14. September 2000
- [16] A. EBERLE, *Split EULER-Fluxes Based on Variable Reference Speeds*, Extended Abstract for the Seventh International Conference on Hyperbolic Problems, ETH Zürich, Switzerland, 9.-13. Februar, 1998
- [17] M. FERTIG, A. DOHR, H.-H. FRÜHAUF, *Transport Coefficients for High Temperature Nonequilibrium Air Flows*, AIAA-98-2937, 7th AIAA Joint Thermophysics and Heat Transfer Conference, Albuquerque, New Mexico, USA, 15.-18. Juni, 1998
- [18] M. FERTIG, H.-H.-FRÜHAUF, M. AUWETER-KURTZ, *Modelling of Reactive Processes at SiC Surfaces in Rarefield Nonequilibrium Airflows*, AIAA-2003-3102, 2003
- [19] C. A. J. FLETCHER, *Computational Techniques for Fluid Dynamics*, Volumes I and II, Springer-Verlag, 1988
- [20] D. R. FOKKEMA, G. L. G. SLEIJPEN, H. A. VAN DER VORST, *Accelerated Inexact Newton Schemes for Large Systems of Nonlinear Equations*, SIAM J. Sci. Comput., Vol. 19, No. 2, pp. 657-674, March 1998
- [21] R. W. FREUND, G. H. GOLUB, N. M. NACHTIGAL, *Iterative Solution of Linear Systems*, Acta Numerica, pp. 1-44, 1992
- [22] H.-H. FRÜHAUF, F. INFED, M. FERTIG, F. OLAWSKY, *Thermal Loads for the Experimental Vehicle X-38*, Proc. 4th European Symposium Aerothermodynamics for Space Applications, 15-18 October 2001, Capua, Italy, ESA SP-487, March 2002
- [23] P. GERLINGER, D. BRÜGGEMANN, *Multigrid Convergence Acceleration for Turbulent Supersonic Flows*, International Journal for Numerical Methods in Fluids, Vol. 24, pp. 1019-1035, 1997
- [24] P. GERLINGER, D. BRÜGGEMANN, *An Implicit Multigrid Scheme for the Compressible Navier-Stokes Equations with Low-Reynolds-Number Turbulence Closure*, ASME Fluids Engineering Division Summer Meeting FEDSM'97, June 22-26, 1997

- [25] U. GERLINGER, H.-H. FRÜHAUF, TH. BÖNISCH, *Implicit Upwind Navier-Stokes Solver for Reentry Nonequilibrium Flows*, AIAA-Paper 97-2547, 32nd AIAA Thermophysics Conference, Atlanta, Georgia, USA, June 23-25, 1997
- [26] P. GEUZAINÉ, I. LEPOT, F. MEERS, J.-A. ESSERS, *Multilevel Newton-Krylov Algorithms for Computing Compressible Flows on Unstructured Meshes*, AIAA-99-3341, 1999
- [27] I. M. G. GROTHOWSKY, *Efficient Time Integration of Navier-Stokes Equations*, Computer & Fluids, 28, pp. 243-263, 1999
- [28] J. HEIERMANN, M. AUWETER-KURTZ, A. EBERLE, U. IBEN, P. C. SLEZIONA, *Robuste hochauflösende Methoden zur Simulation magnetoplasmadynamischer Raketen-triebwerke*, DGLR-JT99-034, DGLR-Jahrbuch, S. 923-929, 1999
- [29] C. HIRSCH, *Numerical Computation of Internal and External Flows*, Volume I, John Wiley & Sons, 1988
- [30] E. ISSMAN, G. DEGREGZ, H. DECONINCK, *Implicit Upwind Residual-Distribution Euler and Navier-Stokes Solver on Unstructured Meshes*, AIAA Journal Vol. 34, No. 10, October 1996
- [31] F. INFED, M. FERTIG, H.-H. FRÜHAUF, F. OLAWSKY, *Oberflächen-Temperaturunsicherheiten beim X38-Demonstrator*, Preprint Nr. 59, Preprint-Reihe des SFB 259, 2001
- [32] F. INFED, M. FERTIG, *Realgas-, Katalyse- und Verdünnungseffekte auf aerothermodynamische Lasten (TETRA AP 21390)*, Endbericht IRS-02 IB 06, TET-IRS-21-TN-5030, Institut für Raumfahrtssysteme, Universität Stuttgart, Dezember 2002
- [33] F. INFED, M. AUWETER-KURTZ, *Simulation of Hypersonic Flows in Thermo Chemical Nonequilibrium Around Re-entry Vehicle X-38 with the URANUS Code*, Space Technology, Vol. 24, Iss. 2-3, pp. 81-93, 1 issue, 2004
- [34] F. INFED, F. OLAWSKY, M. AUWETER-KURTZ, *Stationary Coupling of 3D Hypersonic Nonequilibrium Flows and TPS Structure with URANUS*, AIAA-2003-3984, 16th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, June 23-26, 2003
angenommen zu Veröffentlichung im Journal of Spacecraft and Rockets unter dem Titel: *Stationary Coupling of Hypersonic Nonequilibrium Flows and Thermal Protection System Structure*
- [35] A. JAMESON, W. SCHMIDT, *Some Recent Developments in Numerical Methods for Transonic Flows*, Computer Methods in Applied Mechanics and Engineering 51, North-Holland, pp. 467-493, 1985
- [36] A. JAMESON, W. SCHMIDT, E. TURKEL, *Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes*, AIAA paper 81-1259, AIAA 14th Fluid and Plasma Dynamics Conference, June 23-25, Palo Alto, California, 1981

-
- [37] A. JAMESON, S. YOON, *Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations*, AIAA Journal, Vol. 25, No. 7, July 1987
- [38] S. JONAS, *Implizites Godunov-Typ-Verfahren zur voll gekoppelten Berechnung reibungsfreier Hyperschallströmungen im thermo-chemischen Nichtgleichgewicht*, Dissertation, Fakultät für Luft- und Raumfahrttechnik, Universität Stuttgart, VDI Fortschrittsbericht, Reihe 7, Nr. 221, 1993
- [39] W. JOUBERT, T. OPPE, R. JANARDHAN, W. DEARHOLT, *Fully Parallel Global M/ILU Preconditioning for 3-D Structured Problems*, LANL Technical Report, LAUR 98-2259, 1998
- [40] S. KANNE, *Zur thermo-chemischen Relaxation innerer Freiheitsgrade durch Stoß- und Strahlungsprozesse beim Wiedereintritt*, Dissertation, Fortschritts-Berichte VDI, Reihe 12, Nr. 429, 2000
- [41] B. VAN KEUK, *Numerische Analyse und Bewertung von Upwind-Verfahren für die Anwendung auf mehrdimensionale Über- und Hyperschallströmungen von Gasen*, Dissertation, RWTH Aachen, Shaker Verlag, 2000
- [42] S.S. KIM, C. KIM, O. H. RHO, *A Multigrid Algorithm for Computing Hypersonic, Chemically Reacting Flows*, AIAA-2000-2479, 2000
- [43] O. KNAB, *Konsistente Mehrtemperatur-Modellierung von thermochemischen Relaxationsprozessen in Hyperschallströmungen*, Dissertation, Fakultät Luft- und Raumfahrttechnik, Universität Stuttgart, 1996
- [44] O. KNAB, H.-H. FRÜHAUF, E. MESSERSCHMID, *Validation of a Physically Consistent Coupled Vibration-Chemistry-Vibration Model for Ionized Air in Thermo-Chemical Nonequilibrium*, AIAA-Paper 93-2866, 28th AIAA Thermophysics Conference, Orlando, Florida, USA, July 6-8, 1993
- [45] D. A. KNOLL, V. A. MOUSSEAU, W. J. RIDER, *On Coarse Grid Operators and Multigrid Preconditioned Newton-Krylov Methods in CFD*, AIAA-99-3340, 1999
- [46] D. A. KNOLL, W. J. RIDER, *A Multigrid Preconditioned Newton-Krylov Method*, SIAM J. Sci. Comput., Vol. 21, No. 2, pp. 691-710, 1999
- [47] D. A. KNOLL, W. B. VANDERHEYDEN, V. A. MOUSSEAU, D. B. KOTHE, *On Preconditioning Newton-Krylov Methods in Solidifying Flow Applications*, SIAM J. Sci. Comput., Vol. 23, No. 2, pp. 381-397, 2001
- [48] E. KRAUSE, W. JÄGER (EDS.), *High Performance Computing in Science and Engineering '99*, Springer, 1999
- [49] N. KROLL, R. RADESPIEL, C.-C. ROSSOW, *Accurate and Efficient Flow Solvers for 3D Applications on Structured Meshes*, Paper vorgestellt im Rahmen des AGARD-FDP-VKI Kurses über 'Parallel Computing in CFD', gehalten am VKI, Rhode-Saint-Genèse, Belgium, 15.-19. Mai 1995 und 16.-20. Oktober 1995 bei NASA Ames, USA, veröffentlicht in R-807

- [50] C. H. LEE, B. ZHENG, S. P. WU, *Numerical Computation of Hypersonic Flows over Complex Configuration*, AIAA-99-3687, 1999
- [51] R. J. LEVEQUE, H. C. YEE, *A Study of Numerical Methods for Hyperbolic Conservation Laws with Stiff Source Terms*, Journal of Computational Physics 86, pp. 187-210, 1990
- [52] H. LOMAX, T. H. PULLIAM, D. W. ZINGG, *Fundamentals of Computational Fluid Dynamics*, Scientific Computation, Springer, 2003
- [53] D. LOVELY, R. HAIMES, *Shock Detection from Computational Fluid Dynamics Results*, AIAA-99-3285, 1999
- [54] H. LUO, J. D. BAUM, R. LÖHNER, *A Fast, Matrix-Free Implicit Method for Compressible Flows on Unstructured Grids*, Journal of Computational Physics 146, 664-690, 1998
- [55] H. LUO, J. D. BAUM, R. LÖHNER, *A Fast, Matrix-Free Implicit Method for Computing Low Mach Number Flows on Unstructured Grids*, AIAA-99-3315, 1999
- [56] A. MEISTER, *Zur zeitgenauen numerischen Simulation reibungsbehafteter, kompressibler, turbulenter Strömungsfelder mit einer impliziten Finite-Volumen-Methode vom Box-Typ*, Forschungsbericht 96-08, Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V., 1996
- [57] A. MEISTER, *Comparison of Different Krylov Subspace Methods Embedded in an Implicit Finite Volume Scheme for the Computation of Viscous and Inviscid Flow Fields on Unstructured Grids*, Journal of Computational Physics 140, 311-345, 1998
- [58] A. MEISTER, *Numerik linearer Gleichungssysteme*, Vieweg Verlag, 1999
- [59] A. MEISTER, CH. VÖMEL, *Efficient Preconditioning of linear Systems arising from the Discretization of Hyperbolic Conservation Laws*, Hamburger Beiträge zur Angewandten Mathematik, Reihe F, Computational Fluid Dynamics and Data Anaysis 10, Universität Hamburg, July 1999
- [60] T. MAGIN, D. P. VANDEN ABBEELE, G. DEGREGZ, *An Implicit Multiblock Solver for Inductive Plasma Flows*, AIAA 2000-2480, 2000
- [61] L. MOTTURA, L. VIGEVANO, M. ZACCANTI, *Factorized Implicit Upwind Methods Applied to Inviscid Flows at High Mach Number*, AIAA 98-2822, 1998
- [62] F. OLAWSKY, F. INFED, M. AUWETER-KURTZ, *Preconditioned Newton Method for Computing Supersonic and Hypersonic Nonequilibrium Flows*, Journal of Spacecraft and Rockets, Vol. 41, No. 6, pp. 907-914, November-December 2004
- [63] C. W. OOSTERLEE, L. WASHIO, *Krylov Subspace Acceleration of Nonlinear Multigrid with Application to Recirculating Flows*, SIAM J. Sic. Comput., Vol. 21, No. 5, pp. 1670-1690, 2000

-
- [64] C. PARK, *Convergence of Computation of Chemical Reacting Flows*, Thermophysical Aspects of Reentry Flows, Progress in Astronautics and Aeronautics, 103, edited by J. N. Moss and C. D. Scott, AIAA, New York, pp. 478-513, 1986
- [65] M. PERNICE, M. D. TOCCI, *A Multigrid-Preconditioned Newton-Krylov Method for the Incompressible Navier-Stokes Equations*, SIAM J. Sci. Comput., Vol. 24, No. 2, pp. 398-418, 2001
- [66] M. PERNICE, H. F. WALKER, *NITSOL: a Newton Iterative Solver for Nonlinear Systems*, SIAM J. Sci. Comput., Vol. 19, No. 1, pp. 302-318, January 1998
- [67] A. PUEYO, D. W. ZINGG, *Improvements to a Newton-Krylov Solver for Aerodynamic Flows*, AIAA 98-0619, 1998
- [68] A. PUEYO, *An Efficient Newton-Krylov Method for the Euler and Navier-Stokes Equations*, Dissertation, Universität Toronto, 1998
- [69] TH. H. PULLIAM, S. ROGERS, T. J. BARTH, *Practical Aspects of Krylov-Subspace Iterative Methods in CFD*, AGARD 77th Fluid Dynamics Panel Symposium on Progress and Challenges in CFD Methods and Algorithms, Seville, Spain, 2-5 October 1995
- [70] T. H. PULLIAM, *Implicit Methods in CFD*, Presented at ICFD 1988 Conference on Numerical Methods for Fluid Dynamics, Oxford University, Oxford, UK, March 21-24, 1988
- [71] R. RADESPIEL, C. ROSSOW, *An Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations*, AIAA-89-1953, AIAA 9th Computational Fluid Dynamics Conference, Buffalo, New York, June 14-16, 1989
- [72] R. RADESPIEL, J. M. A. LONGO, S. BRÜCK, D. SCHWAMBORN, *Efficient Numerical Simulation of Complex 3D Flows with Large Contrast*, Preprint of paper presented at the 77th AGARD Fluid Dynamics Panel Meeting and Symposium on 'Progress and Challenges in CFD Methods and Algorithms', Seville, Spain, 2-5 October 1995
- [73] Y. SAAD, H. A. VAN DER VORST, *Iterative Solution of Linear Systems in the 29th Century*, Journal of Computational and Applied Mathematics, Vol. 123, Issue 1-2, Special issue on numerical analysis 2000 Vol. III: linear algebra pp. 1-33, 2000
- [74] Y. SAAD, K. WU, *DQGMRES: a direct Quasi-minimal residual algorithm based on incomplete orthogonalization*, Numerical Linear Algebra with Applications, Vol. 3, pp. 329-343, 1996
- [75] E. SCHÖLL, *Ein dreidimensionales Rechenverfahren für reibungsbehaftete Strömungen in Schaufelreihen von Turbomaschinen*, Dissertation, Universität Stuttgart, VDI Verlag, 1995
- [76] D. SHAROV, H. LUO, J. D. BAUM, R. LÖHNER, *Implementation of Unstructured Grid GMRES+LU-SGS Method on Shared-Memory, Cache-Based Parallel Computers*, AIAA-2000-0927, 2000

- [77] V. SIMONCINI, *On the Convergence of restarted Krylov Subspace Methods*, SIAM J. Matrix Anal. Appl., Vol. 22, No. 2, pp. 430-452, 2000
- [78] P. C. SLEZIONA, *Numerische Analyse der Strömungsvorgänge in magnetoplasmadynamischen Raumfahrtantrieben*, Dissertation, 1992
- [79] A. SOULAÏMANI, G. DA PONTE, N. B. SALAH, *Acceleration of GMRES Convergence for Three-Dimensional Compressible, Incompressible and MHD Flows*, AIAA-99-3381, 1999
- [80] K. STÜBEN, *Algebraic Multigrid (AMG): An Introduction with Applications*, GMD-Forschungszentrum Informationstechnik GmbH, GMD Report 70, 1999
<http://www.gmd.de>
- [81] T. TANG, *Convergence Analysis for Operator-Splitting Methods applied to Conservation Laws with Stiff Source Terms*, SIAM J. Numer. Anal. Vol. 35, No. 5, pp. 1939-1968, October 1998
- [82] E. F. TORO, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, 1999
- [83] H. A. VAN DER VORST, C. VUIK, *GMRESR: A Family of Nested GMRES Methods*, Numerical Linear Algebra with Applications, Vol. 1(4), pp. 369-386, 1994
- [84] H. A. VAN DER VORST, *Parallel Iterative Solution Methods for Linear Systems arising from Discretized PDE's*, Special Course on Parallel Computing in CFD, AGARD-R-807, AGARD, Neuilly-sur-Seine, France Workshop Lecture Notes, 1995
- [85] H. A. VAN DER VORST, *Bi-CGStab: a Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comput., Vol. 13, No. 2, pp.631-644, March 1992
- [86] V. VENKATAKRISHNAN, *On the Accuracy of Limiter and Convergence to Steady State Solutions*, AIAA Paper 93-0880, 1998
- [87] V. VENKATAKRISHNAN, *Perspective on Unstructured Grid Flow Solvers*, AIAA Journal Vol. 34, No. 3, March 1996
- [88] V. VENKATAKRISHNAN, *Improved Multigrid Performance of Compressible Navier-Stokes Solvers*, AIAA-98-2967, 1999
- [89] J. VIERENDEELS, K. RIEMSLAGH, E. DICK, *A Multigrid Method for Viscous Flows for All Mach Numbers and All Grid Aspect Ratios*, AIAA-200-2255, 2000
- [90] C. VUIK, *GMRESR: A family of nested GMRES methods*, Report 91-80, Faculty of Technical Mathematics and Informatics, TU Delft, 1991
- [91] C. VUIK, *Fast Iterative Solvers for the discretized incompressible Navier-Stokes equations*, Report 93-98, Faculty of Technical Mathematics and Informatics, TU Delft, 1993

- [92] C. VUIK, *Further experiences with GMRESR*, Report 92-12, Faculty of Technical Mathematics and Informatics, TU Delft, 1992
- [93] K. VUIK, *New insights in GMRES-like methods with variable preconditioners*, Report 93-10, Faculty of Technical Mathematics and Informations, TU Delft, 1993
- [94] Y. WADA, M.-S. LIOU, *An Accurate and Robust Flux Splitting Scheme for Shock and Contact Discontinuities*, SIAM J. Sci. Comput., Vol. 18, No. 3, pp. 633-657, May 1997
- [95] CHR. WAGNER, *Introduction to Algebraic Multigrid*, Course Notes of an Algebraic Multigrid Course at the University of Heidelberg in the Wintersemester 1998/99, Version 1.1, 1998
- [96] J. M. WEISS, *Calculation of Reactive Flowfields Involving Stiff Chemical Kinetics*, AIAA-99-3369, 1999
- [97] J. WENDT (ED.), *Computational Fluid Dynamics*, with contributions by J. D. Anderson, G. Degrez, E. Dick and R. Grundmann, Springer, 1992
- [98] M. J. WRIGHT, G. V. CANDLER, D. BOSE, *Data-Parallel Line Relaxation Method for the Navier-Stokes Equations*, AIAA Journal Vol. 36, No. 9, September 1998
- [99] S. ZENG, C. VUIK, P. WESSELING, *Solution of the Incompressible Navier-Stokes Equations in General Coordinates by Krylov Subspace and Multigrid Methods*, Report 93-64, TU-Delft, 1993
- [100] S. ZENG, C. VUIK, P. WESSELING, *Solution of the Incompressible Navier-Stokes Equations in General Coordinates by Krylov Subspace and Multigrid Methods*, Faculty of Technical Mathematics and Informatics Delft University of Technology, TUD Report 93-64, 1993
- [101] J. ZHANG, *Preconditioned Krylov Subspace Methods for Solving Nonsymmetric Matrices from CFD Applications*, Technical Report No. 280-98, Department of Computer Science, University of Kentucky, Lexington, KY, 1998

Abstract

Newton-Krylov-Method for Supersonic and Hypersonic Thermochemical Nonequilibrium Flows

For the simulation of three-dimensional supersonic and hypersonic nonequilibrium flows around reentry vehicles a Navier-Stokes solver was developed. For an accurate modeling of thermochemical relaxation processes the Navier-Stokes equations consist of 5 continuity equations for N_2 , O_2 , NO , N and O , 3 momentum equations, the total energy equation and 3 vibrational energy equations for the molecular species N_2 , O_2 and NO .

The unsteady, compressible Navier-Stokes equations are discretized in space using cell-centered finite volume approach on structured grids. The inviscid fluxes are formulated in the physical coordinate system and calculated by Roe's scheme or AUSMDV scheme. AUSMDV seems to be more robust and more accurate than Roe. The viscous fluxes are discretized in the computational domain using classical central differences formulas of second order accuracy.

To calculate the steady-state solution of the Navier-Stokes equations an implicit modified Newton method is used. To reduce memory requirement the Jacobian of the fluxes are computed using the 1st order inviscid fluxes and a thin shear layer approximation of the viscous fluxes, while the source terms Jacobian is implemented exactly. In each step of the Newton scheme a linear system has to be solved. The first solver tested in this scheme is the Jacobi Line Relaxation method which was developed by Schöll and Jonas. This solver has been analysed and compared with preconditioned Krylov subspace methods, which are the most powerful solvers for large, sparse linear systems. The Krylov subspace methods TFQMR, QMRCGStab, CGS, BiCGStab and GMRES(l) are used with a block diagonal scaling preconditioner, the LUSGS preconditioner and a block ILU preconditioner. A new preconditioner was developed called Line LU factorization. Also the Jacobi Line Relaxation method was used as a preconditioner.

Due to the large computational effort the vector supercomputer NEC SX-5 has been used for the computation. So special attention has been paid to the development of vectorizable preconditioners. For the forward and backward substitution the truncated Neumann series and the hyper plane ordering have been tested. Also a new method to compute the ILU factorization iteratively has been developed.

The best solvers for the linear systems are GMRES-ILU, GMRES-LLU and the Jacobi Line Relaxation method. Considering the vectorization the truncated Neumann series is not recommended. Best results have been achieved by using hyper planes.

For the described Newton scheme with an approximated Jacobian, a nearly quadratic convergence can be achieved for all Mach numbers, if the inviscid fluxes are computed with a 1st order spatial discretization. In order to accelerate the convergence for a 2nd order discretization, the Newton scheme was enhanced to an exact Newton-Krylov-method. Starting from a matrix-free implicit method, several schemes have been developed to reduce approximations in the Jacobian. Especially a method has been developed to compute the matrix-vector-products exactly without approximations. But also with this new scheme a fast convergence could only be achieved for some reconstruction schemes of the primitive variables. For hypersonic flows with strong shocks, a quadratic convergence of the Newton method is not always given. Thus it is impossible to guarantee a quadratic convergence of the Newton scheme for hypersonic reentry flows.

Lebenslauf

Ferdinand Olawsky

geboren am 31.3.1970 in Pforzheim

1976–1980	Grundschule in Wittlensweiler
1980–1989	Kepler-Gymnasium Freudenstadt
1989	Abitur
1989–1990	Grundwehrdienst
1990–1991	Studium der kath. Kirchenmusik an der Staatlichen Hochschule für Musik und Darstellende Kunst Stuttgart
1991–1997	Studium der Mathematik an der Universität Stuttgart
1997	Diplomarbeit <i>Isoenergetischer KAM-Satz für koisotrope hamiltonsche Vektorfelder und Anwendungen</i>
1997	Studienabschluss, Diplom–Mathematiker
1997-2003	Wissenschaftlicher Angestellter am Institut für Raumfahrtsysteme der Universität Stuttgart im Sonderforschungsbereich 259 <i>Hochtemperaturprobleme rückkehrfähiger Raumtransportsysteme</i> im Teilprojekt <i>Numerische Wiedereintrittsaerothermodynamik</i>
seit 2004	Wissenschaftlicher Angestellter am Fraunhofer Institut für Techno- und Wirtschaftsmathematik, Kaiserslautern