

SYMBOLISCHE ABLEITUNGEN IM RAHMEN NUMMERISCHER STRAHLUNGSHYDRODYNAMIK

Diplomarbeit,
eingereicht von

Matthias Kittel

zur Erlangung des akademischen Grades
Magister der Naturwissenschaften
an der Fakultät für Naturwissenschaften und Mathematik
der Universität Wien.

Wien, im Juni 2004

Institut für Astronomie
Türkenschanzstr. 17
A-1180 Wien

Inhaltsverzeichnis

Zusammenfassung (abstract)	ix
Prolog	1
1 Einleitung	3
1.1 Einführung	3
1.2 über diese Arbeit	4
1.3 Motivation	5
1.4 persönliche Anmerkungen	6
2 Physik der SHD	7
2.1 Vereinfachungen	7
2.2 SHD-Gleichungen	8
2.2.1 Gleichungen für die Materie	9
2.2.2 Strahlungs-Gleichungen	10
2.3 Eddington-Näherung	11
2.4 Wechselwirkung zwischen Strahlung und Materie	12
2.4.1 Opazität	13
2.4.2 Zustandsgleichung	14
3 Mathematik der SHD	16
3.1 Definitionen	16
3.2 Reynolds Transport Theorem	19
3.3 SHD-Gleichungen in unterschiedlichen Bezugssystemen	22
3.3.1 SHD-Gleichungen mit totaler Zeitableitung (Lagrange)	23
3.3.2 SHD-Gleichungen mit partieller Zeitableitung (Gittersystem)	23
3.3.3 SHD-Gleichungen mit partieller Zeitableitung (Euler)	24
3.4 Künstliche Viskosität	25
3.4.1 Künstliche Viskosität in der Bewegungsgleichung	28
3.4.2 künstliche Viskosität in der Energiegleichung	30
3.5 Weitere notwendige mathematisch-nummerische Konstruktionen	31
3.5.1 Adaptive Gitter	31
3.5.2 Räumliche und zeitliche Glättung	32

3.5.3	Zeitliche Zentrierung	32
3.6	SHD-Gleichungen in Integralform	33
4	Diskretisierung	35
4.1	symbolische Notation	35
4.2	Diskretisierungsregeln	37
4.2.1	Operatoren	37
4.2.2	Volumens- und Oberflächenterme	40
4.3	Diskretisierung einer vektoriellen Gleichung - Bewegungsgleichung	45
4.4	Diskretisierung einer skalaren Gleichung - Energiegleichung	47
4.5	diskretisierte SHD-Gleichungen	49
5	Lösung des SHD-Systems	50
5.1	Allgemeines	50
5.2	Lösungsverfahren	50
5.3	Jacobi-Matrix	53
5.3.1	besondere Form der Jacobimatrix	53
6	MATHEMATICA-Notebook	56
6.1	Vorgehensweise	57
6.2	Eigenschaften von MATHEMATICA	58
6.2.1	Gleichungen	59
6.2.2	Muster und Blanks	60
6.2.3	Ableitungen	61
6.2.4	Ersetzungsregeln	61
6.2.5	automatische Vereinfachungen	62
6.2.6	Datenausgabe	63
6.2.7	Nachbearbeitung	63
6.3	Struktur	64
6.4	Steuerparameter	66
6.5	Gleichungen in MATHEMATICA-Notation	68
6.5.1	Anschreiben einer vektoriellen Gleichung in MATHEMATICA - Bewegungsgleichung	68
6.5.2	Anschreiben einer skalaren Gleichung in MATHEMATICA - Energiegleichung	69
6.6	Ein Beispiel - die Poissongleichung	69
6.7	Modul ShdNom.m	71
6.8	Reinigen	71
7	Ausblick	73
7.1	Ausbau und Erweiterungen	73
7.2	Zweidimensionale SHD	75

7.3 Weiterführung des Notebooks	75
A Index strahlungshydrodynamischer und numerischer Begriffe	79
B Intrinsische und selbstdefinierte MATHEMATICA-Befehle	87
C MATHEMATICA-Notebook zur Berechnung der Ableitungen	95
D ShdNom.m	111
E Output des MATHEMATICA-Notebooks	115
E.1 hivek_mathe.f90	117
E.2 mm.f90	119
E.3 md.f90	120
E.4 mu.f90	122
E.5 me.f90	125
F Naturkonstanten und Umrechnungsfaktoren	127
Sachregister	130
Literaturverzeichnis	133
Danksagungen	138
Lebenslauf	139

Abbildungsverzeichnis

1	Titelblatt <i>De Determinantibus functionalibus</i>	2
2.1	OPAL und Alexander Opazitäten	13
2.2	thermodynamische Effekte einer typischen Zustandsgleichung . . .	15
4.1	Zellenbelegung	37
4.2	Indexverlauf im SHD-Code	37
4.3	Operator zur Berechnung zeitlicher Differenzen $\Delta \mathbf{X}$	39
4.4	Operator zur Berechnung räumlicher Differenzen $\overline{\mathbf{X}}$	40
4.5	Operator zur Berechnung von räumlichen Mitteln	40
4.6	Advektion in einer skalaren Zelle	41
4.7	Advektion in einer vektoriellen Zelle	41
4.8	Berechnungsvorschrift für V_S - das skalare Volumen	42
4.9	Berechnungsvorschrift für V_V - das vektorielle Volumen	42
5.1	graphische Darstellung des eindimensionalen Newton-Verfahrens .	52
5.2	besondere Matrizen	54
5.3	Struktur der Jacobi-Matrix	54
7.1	MATHEMATICA-Notebook zur Berechnung der Ableitungen der Gittergleichung in η -Richtung im zweidimensionalen Fall	77
C.1	Notebook dev_8.2.nb	96

Tabellenverzeichnis

4.1	Bedeutung der in den SHD-Gleichungen verwendeten Größen. . .	36
4.2	von der Art der Strahlungshydrodynamikgleichungen	38
4.3	Operatoren	38
6.1	Variablenkonvention des MATHEMATICA-Notebooks	59
6.2	Gleichungsbezeichnung innerhalb des Notebooks	60
6.3	Steuerparameter des MATHEMATICA-Notebooks	67
B.1	Textkonventionen im Anhang B	88
F.1	astronomische und physikalische Konstanten	128
F.2	Umrechnungsfaktoren cgs- auf SI-Einheiten	129
F.3	Umrechnungsfaktoren SI- auf cgs-Einheiten	129

Astronomie! Bist narrisch?¹

S: The question is, what is a Mahnahmahna?²

W: The question is, who cares!

**Numerical analysis is the study of algorithms for the problems of
continuous mathematics.³**

¹Gängige Äußerung zur Wahl meines Studiums.

²Der Gedanke an Mahnahmahna half oft, Lösungen zu finden.

³Trefethen, L., 1992, *The Definition of Numerical Analysis*, SIAM News

Diese Arbeit wurde in neuer Rechtschreibung abgefasst und die allgemein üblichen Schrift **Antiqua** verwendet. Formeln werden wenn möglich in einer Zeile angeschrieben, dadurch kann es zu Überlängen kommen, des Weiteren gelten folgende Textkonventionen:

Font	Bedeutung
fett	wichtige Textstellen
<i>kursiv</i>	Hervorhebungen sowie Attribute von MATHEMATICA-Befehlen
typewriter	MATHEMATICA-Befehle

Als Textverarbeitungsprogramm wurde $\text{\LaTeX}^4$ verwendet.

In dieser Arbeit gelten folgende Abkürzungen:
SHD - Strahlungshydrodynamik
WSHDC - Wiener Strahlungshydrodynamik-Code

⁴Sehr hilfreich waren:

Güthner K., 1996, *Einführung in $\text{\LaTeX}2\epsilon$* , dpunkt

Knappen J., et al., 1995, *$\text{\LaTeX}2\epsilon$ -Kurzbeschreibung*, Zentrum für Datenverarbeitung, Johannes Gutenberg-Universität Mainz

Mösgen P., 1998, *Sachregister erstellen mit $\text{\LaTeX}$* , Universitätsrechenzentrum Katholische Universität Eichstätt

Zusammenfassung (abstract)

In this master thesis a new way of calculation of derivatives for the Jacobian-matrix used in the Vienna Radiation Hydrodynamics (RHD) - Code is presented. Due to the increasing size and complexity of the equations of RHD caused by newly implemented physical quantities and numerical adjustments, building up and calculating the Jacobian-matrix, which is necessary to solve problems of RHD, becomes more and more difficult and complicated. The symbolic mathematics-software MATHEMATICA is used to calculate these derivatives automatically in a way the user has to make as few as possible corrections by hand. The notebook with documentation, all used programmes and the results are presented.

Moreover the discretized equations of RHD are presented in a new notation and parts of these equations are listed in a way a beginner in working with the Vienna-RHD-code is able to understand them in a more clearly way. Furthermore detailed mathematical deduction of distinct parts of RHD are reviewed to put them into clearer light for the user.

In addition there is a list of numerical terms which helps a new user to find one's way through the Vienna-RHD-code.

Prolog

In hac Commentatione supponam, elementa Determinantis esse differentialia partialia systematis functionum totidem variabilium, harum variabilium respectu sumta. ⁵

*In dieser Abhandlung werde ich voraussetzen, die Elemente der Determinante seien die partiellen Differentiale eines **Systems von Functionen von eben so vielen Veränderlichen, nach diesen Veränderlichen genommen.*** ⁶

Da C.G.J.Jacobi der erste war, der über diese Art von Determinanten schrieb, wurde es bald gebräuchlich von der Funktional- oder Jacobi-Determinante zu sprechen. Die Matrix, aus der die Jacobi-Determinante berechnet wird, nennt man Funktional- oder eben **Jacobi-Matrix**. Um die Gleichungen der Strahlungshydrodynamik (ab hier als SHD bezeichnet) lösen zu können, muss obengennante Jacobi-Matrix aufgestellt werden. Das geschieht im Rahmen eines **impliziten** Lösungsverfahrens bei dem zur Lösung alle partielle Differentiale der Variablen nach allen Variablen berechnet werden müssen. Da durch die stetige Zunahme von physikalischen Vorgängen und Ausreizung numerischer Methoden die Gleichungen immer komplexer werden, ist es nahezu unmöglich, die Ableitungen händisch mit Bleistift und Papier zu bestimmen. Aus diesem Grund war es notwendig einen neuen Weg in Sachen Ableitungen einzuschlagen; Berechnung mittels des Softwarepackages MATHEMATICA. Des Weiteren soll diese Arbeit als Einstieg in den Wiener Strahlungshydrodynamik-Code (kurz WSHDC) dienen, als Nachschlagewerk und Referenz. Dazu ist im Anhang ein Index über die wichtigsten mathematischen-numerischen und physikalischer Begriffe der Strahlungshydrodynamik angefügt. Darüberhinaus wurde versucht, einige Lücken in der Herleitung der diskretisierten SHD-Gleichungen zu schließen, sowie einige grundlegende Aspekte für Anfänger klarer darzustellen und eine neue kompakte Schreibweise der diskretisierten SHD-Gleichungen entwickelt.

⁵aus Jacobis Originalartikel, siehe [18]

⁶deutsche Übersetzung von [18], siehe [39]

12.

De Determinantibus functionalibus *).

(Auct. C. G. J. Jacobi prof. ord. math. Regiom.)

1.

In Commentatione anteriore proprietates praecipuas Determinantium enarravi, quae ad quodcunque elementorum systema pertinet. In hac Commentatione supponam, elementa Determinantis esse differentialia partialia systematis functionum totidem variabilium, harum variabilium respectu sumta. Eiusmodi Determinantia per totam Analyticam gravissimas partes agere constat, quin etiam in variis quaestionibus ad systema functionum plurium variabilium pertinentibus similes vices gerere atque quotientem differentialem functionis unius variabilis. Quod egregie declarant varia theoremata quae de Determinantibus illis aliis occasionibus proposui. Qua de re fortasse convenit ea Determinantia propria appellatione *Determinantium functionalium* insignire. Quemadmodum autem Determinantium functionalium proprietates ex iis quae de Determinantibus algebraicis constant derivabimus, ita Determinantium proprietates algebraicorum vice versa e Determinantium functionalium proprietatibus deduci possunt. Statuendo enim, ipsas

$$f, f_1, f_2 \dots f_n$$

esse functiones lineares variabilium $x, x_1 \dots x_n$,

$$f_k = a_k x + a'_k x_1 + a''_k x_2 \dots + a_k^{(n)} x^n,$$

fit

$$\frac{\partial f_k}{\partial x_i} = a_k^{(i)},$$

ideoque Determinans, ad systema elementorum $a_k^{(i)}$ quodcunque pertinens, haberi potest pro Determinante ad systema differentialium partialium,

$$\frac{\partial f_k}{\partial x_i},$$

pertinente sive pro Determinante functionalis.

*) Quae e theoria aequationum linearium et Determinantium algebraicorum nota supponuntur, demonstrata inveniuntur in Commentatione praecedente „*De Determinantibus*“; ad hanc pertinent Commentationum paragraphi quas asteriscis superpositis notavi.

Kapitel 1

Einleitung

1.1 Einführung

Unendliche Weiten und Ewigkeiten - mit diesen Begriffen verbindet man im Allgemeinen die Astronomie. Es stimmt, dass der Weltraum Heimat der größten Zeit- und Längenskalen ist, nicht so bekannt ist das Existieren vom Allerkleinsten, ins Submikroskopische reichende. Ein Stern ist geprägt durch die Zeit, die das Licht vom Kern bis zum Rand benötigt, geprägt durch die Zeit, in der er pulsiert oder sich auf Grund innerer Vorgänge verändert; bis hin zu jener Zeit, in der sich die eigentlichen astronomischen Vorgänge bewegen, in den Jahrmilliarden in denen der Brennstoff der Sterne aufgebraucht wird, jener Zeitspanne in der die nuklearen Vorgänge im Sterninneren ablaufen.

Im Gegensatz zu den tiefen Weiten des Alls stehen die (sub)atomaren Skalen, auf denen sich die nuklearen Vorgänge abspielen, auf denen die Interaktion der Strahlung mit der Materie stattfindet. Die Entwicklung eines Sternes ist so beschaffen, dass sie in Raum (vom Beginn als Gaswolke im Raum, bis zu seinem Ende als zumindest Weißer Zwerg) und Zeit (Lebensspannen von mehreren Milliarden Jahren sind keine Seltenheit) mehrere Größenordnungen überstreicht. So faszinierend diese Tatsache ist, so schwierig ist die Beschreibung dieser riesigen Zeit- und Raumskalen mittels eines einzigen Programms, das alle physikalischen Aspekte bearbeiten kann.

Eines dieser Programme ist der WSHDC, der das Verhalten von Sternen, insbesondere deren (nichtlineare) Pulsationen berechnet. In diesen Code fließen die Eigenschaften und die Physik von Materie und Strahlung ein.

Ein weiterer enorm wichtiger Schritt von der Beobachtung zur theoretischen Rechnung ist die Übersetzung physikalischer Zusammenhänge in die Sprache der Mathematik, die Transkription von zum Beispiel Erhaltungssätzen in Differentialgleichungen. Neben dieser Transformation ist das Zusammenspiel von Mathematik und Computer, sprich von Numerik beziehungsweise finite Differenzengleichungen wichtig.

Die Physik liefert ein System von gekoppelten, partiellen Differentialgleichungen, welches durch eine spezielle Methode gelöst wird. Dieses Verfahren bezeichnet man als eine **implizite** Methode, das bedeutet, der Zeitschritt ist nicht durch die Courant-Friedrichs-Lewy- Bedingung (kurz CFL-Bedingung) beschränkt. Implizite Verfahren zeichnen sich durch eine größere Komplexität und gesteigerten Rechenaufwand als explizite aus, besitzen aber einen höheren Grad an Stabilität und eben die Fähigkeit, größere Zeitschritte während der Berechnung zu verwenden. Dieses implizite Vorgehen führt zu einem nicht-linearen Gleichungssystem, das mit einem Newton-Raphson-Verfahren (siehe Kapitel 5) gelöst wird. Im Laufe dieses iterativen Verfahrens ist es notwendig die Jacobi-Matrix aufzustellen, in der eine Vielzahl von Ableitungen vorkommen. Diese Ableitungen korrekt und in zeitlich vertretbarer Weise berechnen zu können, verlangt ein großes Maß an Aufwand. Diesen zu reduzieren und in Vorausschau weiter zu verkleinern ist Ziel dieser Arbeit.

1.2 über diese Arbeit

Diese Arbeit gliedert sich in sieben Kapitel. **Kapitel 1** beschreibt die Idee, die hinter dieser Arbeit steckt, welche Hintergründe zur Verwirklichung geführt haben, sowie die Zielsetzungen die verwirklicht hätten werden sollen.

In **Kapitel 2** wird auf die physikalischen Aspekte des Systems der SHD-Gleichungen eingegangen, sowie die wesentlichen Bestandteile erklärt und beschrieben, die in die Gestalt der Gleichungen eingehen und auf ihren Aufbau Einfluss haben.

Eine ausführliche Beschreibung der mathematischen Sichtweise der SHD-Gleichungen wird in **Kapitel 3** vorgestellt, es wird auf die unterschiedlichen Formulierung der Gleichungen in verschiedenen Bezugssystemen eingegangen, sowie spezielle mathematischen Eigenschaften der SHD-Gleichungen untersucht.

Kapitel 4 beschäftigt sich mit der Diskretisierung der SHD-Gleichungen unter Bezugnahme neuartiger Wege der Diskretisierung. Desweiteren wird versucht, den Vorgang der Diskretisierung zu erklären und eine neue übersichtliche Art der Notation vorgestellt.

Kapitel 5 bildet einen weiteren Schwerpunkt und bespricht das Aufstellen der Jacobi-Matrix, sowie die im WSHDC verwendeten Lösungsverfahren.

Das entwickelte Programm zur Berechnung der Ableitungen der Jacobi-Matrix mit Hilfe der symbolischen Mathematik-Software MATHEMATICA wird in **Kapitel 6** vorgestellt und detailliert besprochen.

Kapitel 7 gibt über in Zukunft geplante Arbeiten Auskunft.

Des Weiteren ist ein umfangreicher **Anhang** beigelegt (siehe Seite 78).

1.3 Motivation

Nach der Entwicklung einer Diskretisierung, erdacht durch Dorfi (siehe [14]), stellte sich die Frage, wie diese Neuerung am besten in den WSHDC eingebaut werden könne. Da es sich um eine tiefgreifende Veränderung des bisherigen Konzeptes handelt, mussten neue Wege erdacht und gefunden werden, um diesen neuen Herausforderungen begegnen zu können. Besonders aufwändig schien es, die benötigten Ableitungen der SHD-Gleichungen im Rahmen der Erstellung der Jacobi-Matrix zu berechnen, da jede Gleichung innerhalb dieses nichtlinearen Gleichungssystems nach jeder Variablen abgeleitet werden muss, und sich daraus ein enormer Arbeitsaufwand mit hohem Schwierigkeitsgrad ergibt. Aus diesem Anlass wurde die Idee geboren, diese Differentiationen mittels einer symbolischen Mathematiksoftware durchzuführen. Schon in den Anfangszeiten dieser Art von Programmen, die in die Erstellungszeit des WSHDC fiel, wurde ein solches Programm verwendet, um einen Teil der benötigten Ableitungen zu generieren. Als Software standen einige Programme zur Auswahl; die bekanntesten unter ihnen sind wohl DERIVE, MAPLE und eben MATHEMATICA, welches ausgewählt wurde. Ausschlaggebend dafür, war die Erfahrung mit dieser Software, obwohl eher bescheiden, sowie die leichte Verfügbarkeit über den Zentralen Informatikdienst der Universität Wien.

Neben der Erstellung der Ableitungen der SHD-Gleichungen sollte das Problem in einer globaleren Perspektive betrachtet und bearbeitet werden. Die Lösung für die gestellte Frage musste so formuliert werden, dass sie **jede** Änderung der SHD-Gleichungen verarbeiten, und die richtigen Ableitungen in einer sofort in den WSHDC einbaubaren Weise ausgeben kann. Folgendes muss das MATHEMATICA-Programm liefern, um als Lösung akzeptabel zu sein:

1. Die Korrektheit der Ableitungen muss gewährleistet sein, was durch das getestete symbolische Mathematik-Paket gegeben ist.
2. Der Aufwand der Erstellungen der Ableitungen ist durch die Komplexität und Größe des Gleichungssystems einfach zu hoch, um sie noch händisch durchführen zu können, aus diesem Grund sollte eine einfache Handhabung möglich sein.
3. Alle Änderungen der SHD-Gleichungen in physikalischer oder mathematisch-nummerischer Art sollen automatisch und ohne zusätzlichen Aufwand, das heißt ohne nachträgliche händische Bearbeitung, generiert werden.

4. Die Ausgabe des Programmes soll in einer Art erfolgen, dass sie ohne zusätzlichen Aufwand in den WSHDC eingebaut werden kann.

Oben genannte Punkte waren sozusagen der Arbeitsauftrag für jenes Programm (ab jetzt Notebook, wie Programme in *Matheamtica* genannt werden), das mit Hilfe der ausgesuchten Software zu erstellen war.

1.4 persönliche Anmerkungen

Am Ende dieser Einleitung möchte ich noch einige persönliche Bemerkungen tätigen. Diese Arbeit besitzt einen eher theoretisch-mathematisch Charakter, und hat mich aus diesem Grund zu einem intensiven Studium vieler Teilgebiete der Numerik animiert. Der WSHDC ist ein hochkomplexes Gebilde, in dem Physik und Numerik in beeindruckenswerter Weise zusammengeführt sind. Es war mir wichtig, für mich komplizierte und schwer nachzuvollziehende Aspekte in dieser Arbeit klarer und verständlicher aufzuarbeiten. Des Weiteren soll diese Arbeit für Studenten, die neu in Berührung mit dem WSHDC kommen, als Ausgangspunkt für ihren weiteren Studienweg dienen, wenn sie sich mit der Mathematik und Numerik des Quellcodes vertraut machen wollen.

Kapitel 2

Physik der SHD

In diesem Abschnitt wird auf jene physikalischen Zusammenhänge näher eingegangen, die zur Formulierung der bekannten Gleichungen der SHD führen. Dies soll ohne längliche mathematische Ableitungen der **SHD-Gleichungen** geschehen, aber einen Überblick über die Verschränkung von Materie und Strahlung in mathematischer Sichtweise bieten.

Da die Physik logischerweise die Gestalt der Gleichungen bestimmt, und sich diese durch Hinzufügen neuer Eigenschaften ändert, werden hier die SHD-Gleichungen vorgestellt. Welche Annahmen, welche Vereinfachungen und welche Art von notwendigen physikalischen Überlegungen führen zum jetzigen Aussehen der SHD-Gleichungen, das soll Ziel dieses Abschnitts sein.

Eine umfassende Einführung in die Physik der Sterne liefern [6], [7], [10], [11] und natürlich [29]. Speziell die Hydrodynamik behandelt [30], auf die Strahlung gehen besonders [36] und [52] ein. Die Verwendung des WSHDC bei der Berechnung von Sternpulsationen ist ausführlich in [15], [19], [20], [21], [22], [23] dargestellt. Eine hier nicht erläuterte Erweiterung der Physik durch Stauberzeugung und Massenverlust ist in [28] erklärt.

2.1 Vereinfachungen

Bei der Berechnung von astrophysikalischen Problemen ist es unablässlich präzise und realitätsnah zu programmieren, nichtsdestotrotz müssen Vereinfachungen getroffen werden, um Rechnungen mit vernünftigem Aufwand treiben zu können. Folgende *Zugeständnisse* werden im Rahmen der Lösung der SHD-Gleichungen im WSHDC an die Numerik gemacht:

- **Kontinuumstheorie** - Strahlung und Materie werden als zwei getrennte interagierende Medien beschrieben.
- **Geometrie** - Alle gravitierenden Massekonzentrationen lassen sich unter Vernachlässigung bestimmter Effekte (Rotation, etc.) als Kugeln beschrei-

ben. Aus diesem Grund werden die Rechnungen in Kugelsymmetrie durchgeführt, was zu erheblichen Erleichterungen in mathematisch-nummerischer Hinsicht führt.

- **Physik** - Wie schon erwähnt, werden bestimmte physikalische Eigenschaften von astrophysikalischen Objekten, wie die Rotation oder bestehende Magnetfelder vernachlässigt, um die SHD-Gleichungen so einfach wie möglich zu halten. Diese Vereinfachungen wird in Zukunft aber nicht mehr haltbar sein, da das Verhalten von Sternen und deren Entwicklung grundlegend auf oben genannten Effekten basiert. Ein Ziel dieser Arbeit ist es, diese bestehenden Vereinfachungen nicht mehr notwendig zu machen, damit eine realitätsnähere Beschreibung von Sternen, Planeten und Galaxien mit dem WSHDC möglich wird.
- **Strahlungstransport** - Der Strahlungstransport wird nur in der grauen Näherung und unter Annahme der Eddington-Näherung gelöst.

2.2 SHD-Gleichungen

Im Folgenden werden hier die SHD-Gleichungen⁷ wiedergegeben, die im WSHDC zur Berechnung astrophysikalischer Effekte Verwendung finden. Wie der Name schon sagt, beschreiben die SHD-Gleichungen das Verhalten von Strahlung und Materie (Hydrodynamik), sowie deren Wechselwirkungen. Die Herleitung der Gleichungen des Strahlungstransportes in der sogenannten Eddington-Approximation (eine detaillierte und übersichtliche Herleitung findet man in [19, Seite 39 bis 56]) ist ein Bestandteil der SHD-Gleichungen. Diese sind an die Gleichungen der Hydrodynamik gekoppelt, wobei zwei Einschränkungen wirksam werden. Erstens erfolgt die Behandlung der Gleichungen nur in sphärischer Symmetrie, das heißt, die Gleichungen hängen neben der Zeit nur mehr vom Radius ab, was der astronomischen Realität wohl nahe kommt, zweitens stellen Strahlung und Materie zwei unterschiedliche, aber in Wechselwirkung stehende Fluide dar.

Somit ergibt sich ein System von mindestens sieben nichtlinearen, partiellen, gekoppelten Differentialgleichungen erster Ordnung in Raum und Zeit (inklusive Gittergleichung; bei Staubrechnungen kommen fünf Gleichungen hinzu, ebenso erweitert sich das System bei Hinzunahme der Konvektion).

Die hier angegebenen Gleichungen sind eindimensional und in sphärischer Symmetrie formuliert, das verwendete Koordinatensystem ist Eulerisch (genauers dazu in 3.2). Die verwendeten Symbole beschreiben die Hydrodynamik wie folgt: ρ entspricht der Gasdichte, u_{rel} der Relativgeschwindigkeit zwischen Gitter (siehe

⁷Die Gleichungen sind hier in einer spezifischen Form angeschrieben und zwar in Kugelsymmetrie in der Euler-Formulierung.

Abschnitt 3.5.1) und Gas, u der Gasgeschwindigkeit, P dem Gasdruck, e der inneren Energie und m der integrierten Masse bis zum Radius r . J, H, K sind die Variablen, welche die Strahlung beschreiben und entsprechen den Momenten der Strahlungstransportgleichung. ∂_t ist äquivalent $\frac{\partial}{\partial t}$, ∂_r zu $\frac{\partial}{\partial r}$. Eine Beschreibung aller auftretenden Variablen findet sich in der Tabelle auf Seite 36, die Definitionen aller verwendeten Operatoren auf Seite 18.

Da die SHD-Gleichungen konservativ formuliert sind, müssen sich alle bekannten Erhaltungssätze in ihnen wiedergefunden werden können. Jede Gleichung ist einem Erhaltungssatz zugeordnet.

2.2.1 Gleichungen für die Materie

Zuerst werden jene vier Gleichungen vorgestellt, die das Verhalten der Materie beschreiben. Diese sind an die Gleichungen der Strahlung durch Hereinnahme der Terme mit H, J , und K gekoppelt.

Kontinuitätsgleichung (Massenerhaltung)

$$\partial_t \rho + \frac{1}{r^2} \partial_r (r^2 \rho \vec{u}_{rel}) = 0 \quad (2.1)$$

Die Kontinuitätsgleichung ist die Erhaltungsgleichung für die Masse. Sie sagt aus, dass die zeitliche Änderung der Dichte in einem Volumen, der Massenstrom über den Rand des Volumens entsprechen muss.

Bewegungsgleichung (Impulserhaltung)

$$\partial_t (\rho \vec{u}) + \frac{1}{r^2} \partial_r (r^2 \rho \vec{u} \vec{u}_{rel}) + \partial_r P + \frac{G\rho m}{r^2} - \frac{4\pi}{c} \rho \kappa_R \vec{H} - \vec{u}_Q = 0 \quad (2.2)$$

Die Bewegungsgleichung ist äquivalent der Impulserhaltung für Materie und beschreibt im Großen und Ganzen das zweite Newton'sche Axiom. Das heißt, dass die zeitliche Änderung des Impulses, der Summe aller angreifenden Kräfte entsprechen muss. Die ersten zwei Terme stehen für die zeitliche Änderung der Impulsdichte mit der Zeit. $\partial_r P$ beschreibt die Kraftdichte durch den Gradienten des Gasdruckes, die Kraftdichte durch die Gravitation ist durch

$$\frac{G\rho m}{r^2}$$

gegeben. Der Term

$$\frac{4\pi}{c} \rho \kappa_R \vec{H}$$

steht für die Kraftdichte des Strahlungsdruckes, $\vec{u}_Q$ ergibt sich durch Einführung einer künstlichen Viskosität (siehe 3.4).

Energiegleichung (Energieerhaltung)

$$\begin{aligned} \partial_t (\rho e) &+ \frac{1}{r^2} \partial_r (r^2 \rho e \vec{u}_{rel}) + \\ &+ P \frac{1}{r^2} \partial_r (r^2 \vec{u}) + 4\pi \rho \kappa_R (J - S) - \rho (\epsilon_{nuc} + \epsilon_Q) = 0 \end{aligned} \quad (2.3)$$

Dies ist die Erhaltungsgleichung für die innere Energie der Materie. Wiederum stehen die ersten zwei Terme für die zeitliche Änderung der Variable.

$$P \frac{1}{r^2} \partial_r (r^2 \vec{u})$$

steht für die von der Strömung geleistete mechanische Arbeit,

$$4\pi \rho \kappa_R (J - S)$$

für den Energieaustausch zwischen Materie und Strahlung. Der letzte Term bezieht sich wieder auf die künstliche Viskosität ϵ_Q (siehe 3.4), sowie für die Energieerzeugung durch Fusion ϵ_{nuc} (nukleare Vorgänge).

Poissongleichung (Eigengravitation)

$$\partial_r m = 4\pi \rho r^2 \quad (2.4)$$

Diese Gleichung wird aus der Poissongleichung durch einmalige Integration und Gleichsetzen mit dem Gravitationsgesetz ermittelt und beschreibt die integrierte Masse m bis zum Radius r .

2.2.2 Strahlungs-Gleichungen

Hier sind jene zwei Gleichungen angeschrieben, die das Verhalten der Strahlung veranschaulichen (siehe nächster Abschnitt).

0. Moment der Strahlungstransportgleichung (Strahlungsenergiegleichung)

$$\begin{aligned} \frac{1}{c} \partial_t J &+ \frac{1}{c} \frac{1}{r^2} \partial_r (r^2 J \vec{u}_{rel}) + \frac{1}{r^2} \partial_r (r^2 \vec{H}) + \\ &+ \frac{1}{c} K \frac{1}{r^2} \partial_r (r^2 \vec{u}) - \frac{1}{c} \frac{3K - J}{r} \vec{u} + \rho \kappa_P (J - S) = 0 \end{aligned} \quad (2.5)$$

Diese Gleichung wird durch einmalige Integration über alle Raumwinkel der Strahlungstransportgleichung gewonnen (siehe [36]), sie beschreibt die Energiebilanz für das Strahlungsfeld und kann daher als Energiegleichung für das Strahlungsfeld gesehen werden.

1. Moment der Strahlungstransportgleichung (Strahlungsflussgleichung)

$$\begin{aligned} \frac{1}{c} \frac{1}{r^2} \partial_t \vec{H} + \frac{1}{c} \partial_r (r^2 \vec{H} \vec{u}_{rel}) &+ \partial_r K + \frac{3K - J}{r} + \\ &+ \frac{1}{c} \vec{H} \frac{1}{r^2} \partial_r (r^2 \vec{u}) + \rho \kappa_R \vec{H} = 0 \end{aligned} \quad (2.6)$$

Dieser Gleichung entspricht das erste Moment der Strahlungstransportgleichung und kann als Erhaltungsgleichung für den Impulsfluss der Strahlung angesehen werden.

Diese sechs Gleichungen (und die Gittergleichung, siehe Abschnitt (3.5.1)) bilden das System der SHD-Gleichungen und damit das Rückgrat des WSHDC. Sie werden für jedes Problem, welches sich mit der Interaktion von Strahlung und Materie beschäftigt, herangezogen. Das ist der Grund für ihre vielfältige Verwendung in der stellaren, sowie in der (extra)galaktischen Astronomie.

2.3 Eddington-Näherung

Die letzten beiden Gleichungen (Strahlungsenergie- und Strahlungsflussgleichung) des vorigen Abschnitts bezeichnet man auch als **graue Momente** der Spezifischen Intensität $I_\nu(r, \mu, t)$. Sie werden aus

$$\{J, H, K\}_\nu(r, t) = \frac{1}{2} \int_{-1}^1 I_\nu(r, \mu, t) \mu^{\{0,1,2\}} d\mu \quad (2.7)$$

berechnet (K steht für den Strahlungsdruck). Eine übersichtliche Herleitung dieser Momente findet sich in [19, Seiten 39-56], mehr ins Detail geht [36].

Da jede Momentengleichung zusätzlich das nächstfolgende beinhaltet, erfolgt die Einführung einer zusätzliche Gleichung, um das Gleichungssystem eindeutig lösen zu können. Anstatt das komplette Strahlungsfeld zu berechnen, bedient man sich der sogenannten **Eddington-Approximation** oder **Eddington-Näherung**. Der Faktor f_ν , um das System zu schließen, lautet

$$f_\nu = \frac{K_\nu}{J_\nu} = \frac{1}{3}, \quad (2.8)$$

als Verhältnis zwischen dem nullten und zweiten Moment.

Der Wert von $f_\nu = \frac{1}{3}$, der für Computerberechnungen verwendet wird, ergibt

sich in Folge hoher Isotropie des Strahlungsfeldes und kann als physikalisch motivierter Grenzwert interpretiert werden.

Eine weitere Herleitung und Beschreibung des Eddington-Faktors basierend auf der geometrischen Näherung einer ausgedehnten statische Atmosphäre ist in [33] und [34] gegeben.

2.4 Wechselwirkung zwischen Strahlung und Materie

Durch die SHD-Gleichungen werden die Eigenschaften der Materie gleichzeitig mit denen der Strahlung beschrieben. Diese Wechselwirkung führt zu einem Impuls- und Energieaustausch zwischen diesen beiden Medien, der nicht auf lokale Effekte begrenzt ist, und innerhalb des Sternes an unterschiedlichen, weit entfernten Positionen auftreten kann. Alle Effekte wie Heizung oder Kühlung durch Photonen sind ausführlich in [36] abgeleitet und dargestellt.

Als besonders wichtiger Effekt für die Entstehung von stellaren Pulsationen sei hier der sogenannte κ -Mechanismus genannt, der als Teil des Strahlungsflusses Pulsationen in den Sternhüllen anregt.

Um die Strahlungstransportgleichung lösen zu können, müssen wie beschrieben, deren Momente berechnet werden. Da diese aber frequenzabhängig sind, wird zusätzlich noch über alle Frequenzen integriert. Die Kopplungsterme (beziehungsweise Proportionalitätsfaktoren) zwischen Energie und Strahlung gehen mit unterschiedlichen Gewichtungen in die SHD-Gleichungen über die frequenzabhängige Opazität κ_ν ein. Für diese gewichteten Opazitäten existieren gewisse Grenzwerte, wie das Planck-Mittel κ_P

$$\frac{1}{\kappa_P} = \frac{\int_0^\infty \kappa_\nu B_\nu d\nu}{\int_0^\infty B_\nu d\nu}, \quad (2.9)$$

für den optisch dünnen Fall und dem Rosseland-Mittel κ_R

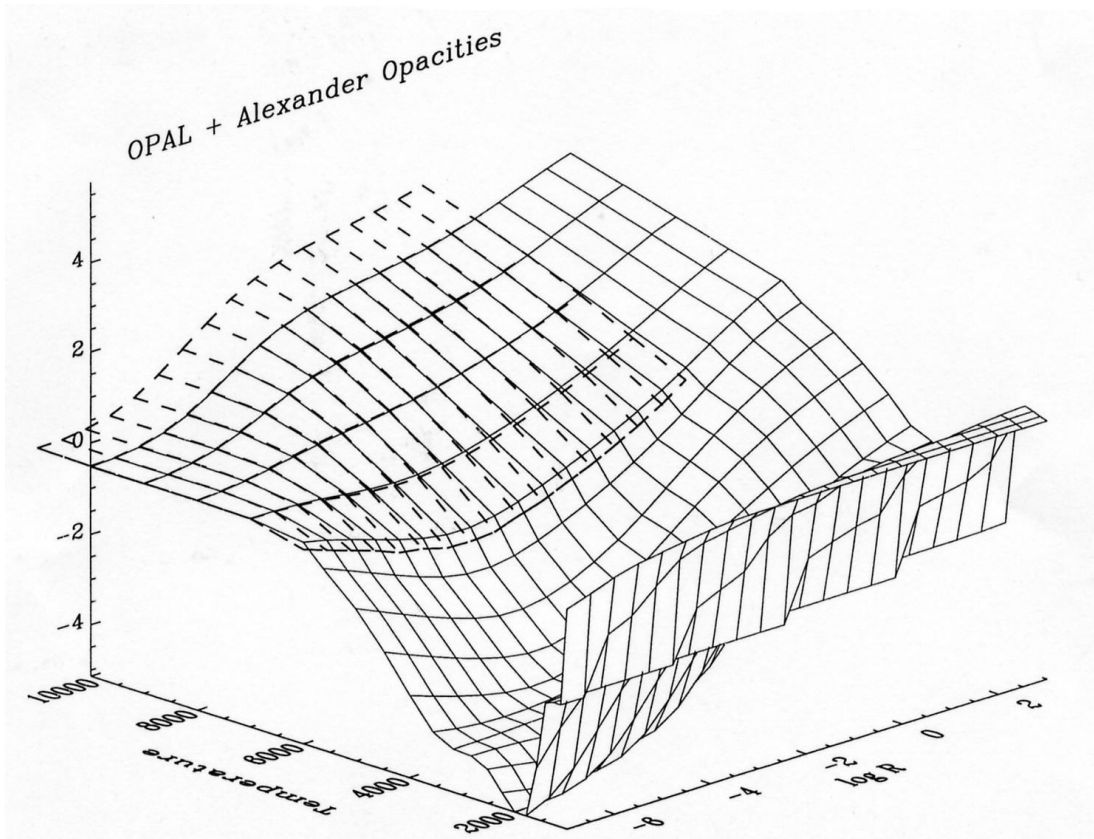
$$\frac{1}{\kappa_R} = \frac{\int_0^\infty \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu}{\int_0^\infty \frac{\partial B_\nu}{\partial T} d\nu}, \quad (2.10)$$

(B_ν entspricht in beiden Fällen der Planckfunktion) für den optisch dicken Fall im Sterninneren. Trotz des Existierens dieser Grenzwerte wird in astrophysikalischen Berechnungen vor allem das Rosseland-Mittel κ_R verwendet, da nur dieses Mittel für unterschiedliche chemische Zusammensetzungen in Tabellenform vorliegend ist. Physikalisch wäre es allerdings besser, die korrekten Grenzwerte für κ_ν zu verwenden.

2.4.1 Opazität

Einen besonderen Einfluss auf stellare Phänomene, die mit den SHD-Gleichungen berechnet werden, hat die chemische Zusammensetzung der Sternmaterie, die durch (X,Y,Z) gegeben ist, wobei X den Massenanteil des Wasserstoffes, Y den von Helium und Z den der *Metalle* (in der Astronomie wird jedes Element schwerer als Helium als *Metal* bezeichnet) beschreibt (Anteile in Prozent). Die Rosseland-Mittel für unterschiedliche chemische Zusammensetzungen sind berechnet und in Tabellenform für die Dichte und Temperatur angegeben. Abbildung (2.1) zeigt so eine geplottete Tabelle für ein typisches Rosseland-Mittel. Opazitätsdaten können über das Internet unter [24], [37] und [38] bezogen werden.

Abbildung 2.1: OPAL und Alexander Opazitäten



Dieses aus [16] entnommene Bild zeigt einen Teil eines typischen Rosseland-Mittels κ_R als Funktion der Temperatur T und $R = \rho/T_6^3$ mit $T_6 = T/10^6$. In dieser Tabelle sind Opazitäten zweier unterschiedlicher Quellen übereinandergelegt, die durchgezogenen Linien sind Alexander Opazitäten (siehe [3]), die strichlierten Linien stammen von [37].

2.4.2 Zustandsgleichung

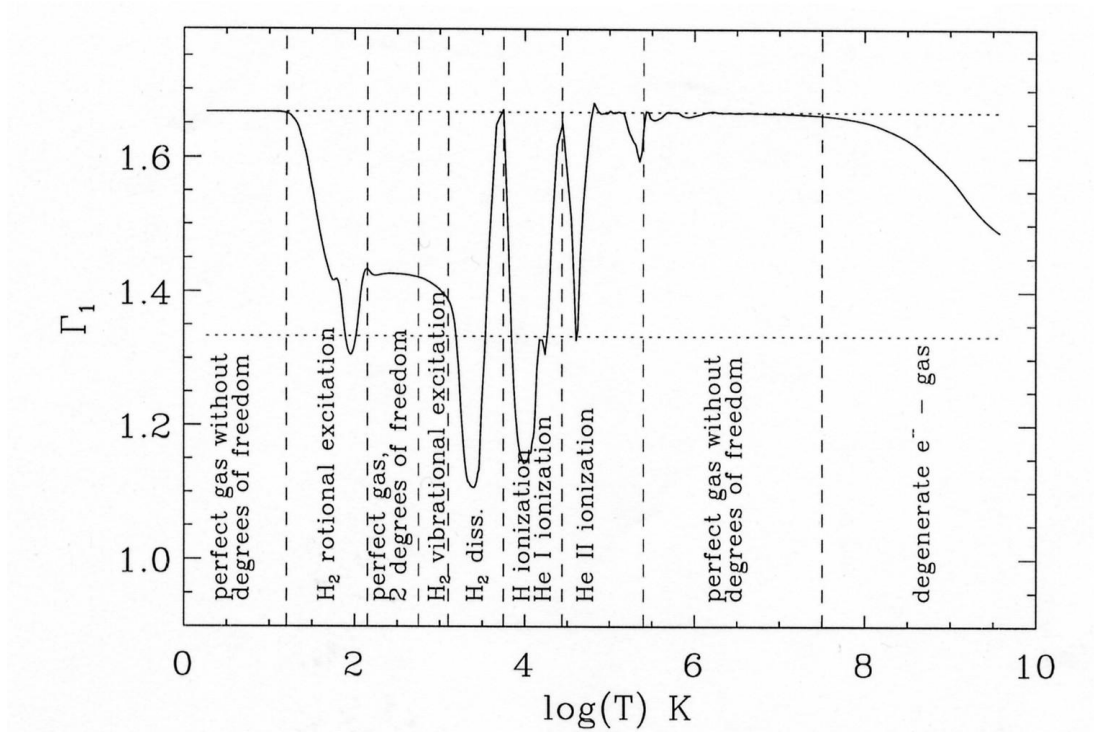
Die Zustandsgleichung des Gases bestimmt die thermodynamischen Eigenschaften eines Gases. Bei einer Vielzahl von astrophysikalischen Anwendungen ist die Annahme eines idealen Gases nicht mehr zulässig, da in es in strahlenden Sternplasmen zu Effekten wie Ionisation von Atomen, Dissoziation von Molekülen, sowie zur Degeneration der Elektronen kommt. Auf Grund dieser physikalischen Zusammenhänge müssen die thermodynamischen Variablen für alle in den Gleichungen vorhandenen Parameter berechnet oder wiederum eine Art von Tabelle verwendet werden.

Als Beispiel sei hier der thermodynamische Adiabatenindex Γ_1 vorgestellt:

$$\Gamma_1 = \frac{\rho}{P} \frac{\partial P}{\partial \rho} = \frac{\partial \ln P}{\partial \ln \rho} \quad (2.11)$$

Die in Abb. 2.2 gezeigte spezielle Zustandsgleichung wurde in [68] berechnet, beinhaltend H, H₂ und He, auf Basis der solaren Metallhäufigkeiten. Es ist Γ_1 gegen ein konstantes ρ von $10^{-13} \text{ kg m}^{-3}$ geplottet.

Abbildung 2.2: thermodynamische Effekte einer typischen Zustandsgleichung



Diese Abbildung (siehe [16]) stellt die thermodynamischen Effekte einer typischen Zustandsgleichung dar. Der Adiabatenindex Γ_1 ist gegen die Temperatur bei einer konstanten Dichte von $\rho = 10^{-13} \text{ kg m}^{-3}$ geplottet. Die gestrichelten waagrechten Linien entsprechen Werten von $\Gamma_1 = 5/3$, sowie $\Gamma_1 = 4/3$. Die bei unterschiedlichen Temperaturen auftretenden thermodynamischen Erscheinungen sind in der Abbildung als Text eingefügt.

Kapitel 3

Mathematik der SHD

Zur Lösung der SHD-Gleichungen ist ein hohes Maß an **mathematischen Kenntnissen** und **Werkzeugen** von Nöten. Als erstes sind alle Definitionen angegeben, die notwendig sind, um die Gleichungen bearbeiten zu können. Danach wird auf die Darstellung derselben in unterschiedlichen Bezugssystemen eingegangen, da es die Physik verlangt, die SHD-Gleichungen unter verschiedenen Blickwinkeln zu betrachten.

3.1 Definitionen

In den folgenden Definitionen und Gleichungen (eine gute Einführung liefert [5]) werden alle Variablen X , die Vektoren darstellen mit $\vec{X}$ bezeichnet, alle anderen Größen X sind Skalare. Eine Beschreibung der Größen der SHD und weitere Erläuterungen findet sich in der Tabelle auf Seite 36.

Der Nabla-Operator $\vec{\nabla}$ auf einen Skalar ψ angewendet, ergibt in Kugelkoordinaten Folgendes:

$$\vec{\nabla}\psi = \begin{pmatrix} \frac{\partial\psi}{\partial r} \\ \frac{1}{r} \frac{\partial\psi}{\partial\theta} \\ \frac{1}{r \sin\theta} \frac{\partial\psi}{\partial\phi} \end{pmatrix} \quad (3.1)$$

Der Nabla-Operator $\vec{\nabla}$ auf einen Vektor

$$\vec{A} = \begin{pmatrix} A_1 \\ A_2 \\ A_3 \end{pmatrix} \quad (3.2)$$

angewendet, ergibt in Kugelkoordinaten

$$\vec{\nabla} \cdot \vec{A} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 A_1) + \frac{1}{r \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta A_2) + \frac{1}{r \sin \theta} \frac{\partial A_3}{\partial \theta}. \quad (3.3)$$

Als letzter Operator sei noch der Laplace-Operator Δ erwähnt, der in Kugelkoordinaten auf einen Skalar ψ angewendet, untenstehendes Ergebnis liefert:

$$\Delta \psi = \vec{\nabla}^2 \psi = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \psi}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial \psi}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 \psi}{\partial \theta^2} \quad (3.4)$$

Das *dyadische Produkt* mit dem Operationszeichen $\otimes$ ist eine Verknüpfung zwischen zwei Vektoren, die folgendermaßen definiert ist. Sind die Komponenten der zwei Vektoren $\vec{a}$ und $\vec{b}$ als

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad (3.5)$$

und

$$\vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (3.6)$$

bezeichnet, dann gilt

$$\vec{a} \otimes \vec{b} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{pmatrix}. \quad (3.7)$$

Weiters gelten für die SHD-Gleichungen folgende Definitionen:

- ∂_t ... partielle zeitliche Ableitung, entspricht einem ortsfesten Koordinatensystem (Euler'sche Formulierung). Es gilt für eine Funktion $F(\vec{x}, t)$

$$\frac{\partial}{\partial t} F(\vec{x}, t) = \lim_{\Delta t \rightarrow 0} \frac{F(\vec{x}, t + \Delta t) - F(\vec{x}, t)}{\Delta t}. \quad (3.8)$$

Da es sich wie gesagt um ein sich zeitlich nicht veränderndes Koordinatensystem handelt, gilt

$$\lim_{\Delta t \rightarrow 0} \frac{\Delta \vec{x}}{\Delta t} = 0. \quad (3.9)$$

- D_t ... totale zeitliche Ableitung, entspricht dem mit dem Gas mitbewegtem Koordinatensystem (Lagrange'sche Formulierung). Es gilt für eine Funktion $F(\vec{x}, t)$

$$\frac{D}{Dt} F(\vec{x}, t) = \lim_{\Delta t \rightarrow 0} \frac{F(\vec{x} + \Delta \vec{x}, t + \Delta t) - F(\vec{x}, t)}{\Delta t}, \quad (3.10)$$

über eine Taylorreihenentwicklung für $F(\vec{x}, t)$ gelangt man zu

$$\frac{D}{Dt} F(\vec{x}, t) = \lim_{\Delta t \rightarrow 0} \frac{F(\vec{x}, t) + \frac{\partial F}{\partial t} \Delta t + \frac{\partial F}{\partial \vec{x}} \Delta \vec{x} - F(\vec{x}, t)}{\Delta t}, \quad (3.11)$$

was

$$\frac{D}{Dt} F(\vec{x}, t) = \frac{\partial}{\partial t} F(\vec{x}, t) + \vec{u}_{gas} \vec{\nabla} F \quad (3.12)$$

mit der Gasgeschwindigkeit

$$\lim_{\Delta t \rightarrow 0} \frac{\Delta \vec{x}}{\Delta t} = \vec{u}_{gas} \quad (3.13)$$

entspricht, da sich das örtliche Koordinatensystem mit dem Gas mitbewegt und so die zeitliche Änderung der Koordinaten mit eben dieser Geschwindigkeit erfolgt.

- d_t ... partielle zeitliche Ableitung, entspricht dem mit dem Gitter (siehe 3.5.1) mitbewegtem Koordinatensystem. Es gilt für eine Funktion $F(\vec{x}, t)$

$$\frac{d}{dt} F(\vec{x}, t) = \frac{\partial}{\partial t} F(\vec{x}, t) + \vec{u}_{grid} \vec{\nabla} F, \quad (3.14)$$

wobei

$$\lim_{\Delta t \rightarrow 0} \frac{\Delta \vec{x}}{\Delta t} = \vec{u}_{grid} \quad (3.15)$$

in diesem Fall der Gittergeschwindigkeit entspricht, da sich die Änderung des Koordinatensystems hier auf den Gitterindex bezieht, das heißt jene Geschwindigkeit, mit der sich die Gitterpunkte an die Funktion anpasst.

- $\partial_r \dots$ partielle räumliche Ableitung
- $\vec{\nabla} \dots$ Nablaoperator
- $\Delta \dots$ Laplaceoperator, entspricht $\vec{\nabla} \cdot \vec{\nabla} = \vec{\nabla}^2$

Des Weiteren wird die eindimensionale Eddington-Näherung ($3K_\nu = J_\nu$ oder $f_\nu = \frac{1}{3}$) angenommen, da der Strahlungsdruck K im allgemeinen Fall ein Tensor wäre.

3.2 Reynolds Transport Theorem

Bei der Formulierung der physikalischen Gleichungen auf einem adaptiven Gitter muss natürlich dem Prinzip der konservativen Formulierung genüge getan werden, sodass sie auch auf einem sich beliebig bewegenden Gitter noch immer Geltung besitzt. Da die physikalischen Variablen als Eulerisch definiert sind, müssen die drei bereits beschriebenen zeitlichen Ableitungen ineinander überführt werden können; das passiert mit Hilfe des **Reynolds Transport Theorems** (siehe [36]), das die Grundlage einer konservativen Formulierung der SHD-Gleichungen darstellt.

$$\begin{aligned}
\frac{D}{Dt} \int_V F(\vec{x}, t) dV &= \int_V \frac{D}{Dt} F(\vec{x}, t) + F(\vec{x}, t) \vec{\nabla} \cdot \vec{u}_{gas} dV = \\
&= \int_V \frac{\partial}{\partial t} F(\vec{x}, t) + \vec{u}_{gas} \vec{\nabla} F(\vec{x}, t) + F(\vec{x}, t) \vec{\nabla} \cdot \vec{u}_{gas} dV = \\
&= \int_V \frac{\partial}{\partial t} F(\vec{x}, t) + \vec{\nabla} (F(\vec{x}, t) \vec{u}_{gas}) dV = \\
&= \int_V \frac{\partial}{\partial t} F(\vec{x}, t) dV + \int_{\partial V} F(\vec{x}, t) \vec{u}_{gas} d\vec{A}
\end{aligned} \tag{3.16}$$

Analog gilt natürlich:

$$\begin{aligned}
\frac{d}{dt} \int_V F(\vec{x}, t) dV &= \int_V \frac{d}{dt} F(\vec{x}, t) + F(\vec{x}, t) \vec{\nabla} \cdot \vec{u}_{grid} dV = \\
&= \int_V \frac{\partial}{\partial t} F(\vec{x}, t) + \vec{u}_{grid} \vec{\nabla} F(\vec{x}, t) + F(\vec{x}, t) \vec{\nabla} \cdot \vec{u}_{grid} dV = \\
&= \int_V \frac{\partial}{\partial t} F(\vec{x}, t) + \vec{\nabla} (F(\vec{x}, t) \vec{u}_{grid}) dV = \\
&= \int_V \frac{\partial}{\partial t} F(\vec{x}, t) dV + \int_{\partial V} F(\vec{x}, t) \vec{u}_{grid} d\vec{A}
\end{aligned} \tag{3.17}$$

Setzt man nun (3.16) und (3.17) gleich, erhält man folgenden Zusammenhang:

$$\begin{aligned}
\frac{d}{dt} \int_V F(\vec{x}, t) dV &= \underbrace{\int_V \frac{\partial}{\partial t} F(\vec{x}, t) dV}_{\frac{D}{Dt} \int_V F(\vec{x}, t) dV} + \int_{\partial V} F(\vec{x}, t) \vec{u}_{grid} d\vec{A} = \\
&= \frac{D}{Dt} \int_V F(\vec{x}, t) dV - \int_{\partial V} F(\vec{x}, t) \vec{u}_{gas} d\vec{A} \\
= \frac{d}{dt} \int_V F(\vec{x}, t) dV &= \frac{D}{Dt} \int_V F(\vec{x}, t) dV + \int_{\partial V} F(\vec{x}, t) (\vec{u}_{grid} - \vec{u}_{gas})^8 d\vec{A} \quad (3.18)
\end{aligned}$$

Analog gilt:

$$\frac{D}{Dt} \int_V F(\vec{x}, t) dV = \frac{d}{dt} \int_V F(\vec{x}, t) dV + \int_{\partial V} F(\vec{x}, t) (\vec{u}_{gas} - \vec{u}_{grid})^8 d\vec{A} \quad (3.19)$$

Als nützlich hat sich auch folgender Satz aus [36] erwiesen:

$$\begin{aligned}
\frac{D}{Dt} \int_V \rho F(\vec{x}, t) dV &= \int_V \frac{D}{Dt} (\rho F(\vec{x}, t)) + \rho F \vec{\nabla} \cdot \vec{u}_{gas} dV = \\
&= \int_V \rho \frac{D}{Dt} F(\vec{x}, t) + \underbrace{F(\vec{x}, t) \frac{D}{Dt} \rho + \rho F(\vec{x}, t) \vec{\nabla} \cdot \vec{u}_{gas}}_{F(\vec{x}, t) \left[\frac{D}{Dt} \rho + \rho \vec{\nabla} \cdot \vec{u}_{gas} \right]} dV = \\
&= \int_V \rho \frac{D}{Dt} F(\vec{x}, t) dV \quad (3.20)
\end{aligned}$$

Kontinuitätsgleichung

Ausgangspunkt ist die Kontinuitätsgleichung in Lagrange'scher Formulierung.

$$\frac{D}{Dt} \int_V \rho dV = 0 \quad (3.21)$$

Wird nun Gleichung (3.19) angewendet, erhält man:

$$\stackrel{(3.19)}{\longrightarrow} \frac{D}{Dt} \int_V \rho dV = \frac{d}{dt} \int_V \rho dV + \int_{\partial V} \rho (\vec{u}_{gas} - \vec{u}_{grid}) d\vec{A} = 0 \quad (3.22)$$

⁸ $\vec{u}_{gas} - \vec{u}_{grid} = \vec{u}_{rel}$

Bewegungsgleichung

Wiederum ist die Gleichung in Lagrange'scher Schreibweise Ausgangspunkt.

$$\begin{aligned}
 \frac{D}{Dt} \int_V \rho \vec{u} dV &= \int_{\partial V} -P d\vec{A} = \int_V -\vec{\nabla} P dV \\
 \xrightarrow{(3.19)} \frac{d}{dt} \int_V \rho \vec{u} dV + \int_{\partial V} \rho \vec{u} (\vec{u}_{gas} - \vec{u}_{grid}) d\vec{A} &= \int_{\partial V} -P d\vec{A} \\
 \xrightarrow{(3.22)} \frac{D}{Dt} \int_V \rho \vec{u} dV &= \int_V \rho \frac{D}{Dt} \vec{u} dV = \int_V -\vec{\nabla} P dV
 \end{aligned} \tag{3.23}$$

Auf Grund von Gleichung (3.23) gilt für jedes Volumen V

$$\rho \frac{D}{Dt} \vec{u} = -\vec{\nabla} P. \tag{3.24}$$

Energiegleichung

Auch hier wird von der Lagrange-Formulierung der Gleichung ausgegangen.

$$\frac{D}{Dt} \int_V \rho e_{tot} dV = \int_{\partial V} -P \vec{u} d\vec{A}, \tag{3.25}$$

mit $e_{tot} = e_{int} + \frac{1}{2} \vec{u}^2$. (e_{tot} entspricht der spezifischen Gesamtenergie (*tot* ... total ... gesamt), e_{int} der inneren Energie (*int* ... internal ... innere) und $\frac{1}{2} \vec{u}^2$ steht für die spezifische Bewegungsenergie.

$$\begin{aligned}
 \frac{D}{Dt} \int_V \rho e_{int} dV + \underbrace{\frac{D}{Dt} \int_V \rho \frac{1}{2} \vec{u}^2 dV}_{\int_V \vec{u} \rho \frac{D}{Dt} \vec{u} dV} &= \int_V -\vec{\nabla} (P \vec{u}) dV
 \end{aligned} \tag{3.26}$$

Aus der Bewegungsgleichung können nun folgende Relationen abgeleitet werden.

$$\int_V \vec{u} \rho \frac{D}{Dt} \vec{u} dV = \int_V \vec{u} (-\vec{\nabla} P) dV \tag{3.27}$$

$$\int_V \rho \frac{1}{2} \frac{D}{Dt} \vec{u}^2 dV = \int_V \vec{u} \rho \frac{D}{Dt} \vec{u} dV \tag{3.28}$$

Setzt man nun Gleichung (3.26) in die Energiegleichung ein, erhält man

$$\begin{aligned}
& \xrightarrow{(3.26)} \frac{D}{Dt} \int_V \rho e_{int} dV + \int_V -\vec{u}(\vec{\nabla} P) dV = \int_V -\vec{\nabla}(P\vec{u}) dV \\
& \frac{d}{dt} \int_V \rho e_{int} dV + \int_{\partial V} \rho e_{int} (\vec{u}_{gas} - \vec{u}_{grid}) d\vec{A} = \int_V -[\vec{\nabla}(\vec{u}) - \vec{u}\vec{\nabla}P] dV \\
& \frac{d}{dt} \int_V \rho e_{int} dV + \int_{\partial V} \rho e_{int} (\vec{u}_{gas} - \vec{u}_{grid}) d\vec{A} = \int_V -P\vec{\nabla}\vec{u} dV \quad (3.29)
\end{aligned}$$

All diese Zusammenhänge werden verwendet, um die SHD-Gleichungen von einem Bezugssystem in das andere umrechnen zu können.

3.3 SHD-Gleichungen in unterschiedlichen Bezugssystemen

Auf Grund der im vorigen Abschnitt hergeleiteten Zusammenhänge können die SHD-Gleichungen, abhängig von ihrem Bezugssystem, in unterschiedlichen Schreibweisen angegeben werden. Allen ist gemein, dass sie in Kugelkoordinaten angeschrieben sind, von denen jeweils nur der Radialterm berücksichtigt wird.

Je nach Problemstellung bevorzugt man eines der besprochenen Bezugssysteme. In einigen Fällen ist es von Vorteil, wenn man anstatt eines adaptiven Gitters die Euler-Beschreibung einer örtlich fixierten Zelle verwendet, bei anderen Untersuchungen ist es sinnvoll sich mit dem strömenden Gas mitzubewegen.

Diese Flexibilität in der Betrachtungsweise führt allerdings zu einer mathematischen Mehrarbeit, da beim Übergang von einem System in ein anderes Zusatzterme berechnet werden müssen, um die Transformation konsistent abschließen zu können. Diese Terme werden **Advektion** genannt. Geht man zum Beispiel von einem System, das sich mit dem Gas mitbewegt, zu einem ortsfesten System über, beschreiben die so neu entstandenen Advektionsterme den Fluss über die Oberfläche in das nun neu entstandene Volumen (das auf Grund der Folgen der Diskretisierung entsteht, siehe Seite 35), welches in ersterer Betrachtungsweise nicht vorkommt, da man sich ja mit dem System in der Strömung mitbewegt. Es werden hier drei Bezugssysteme vorgestellt, sowie die Terme beschrieben, die entstehen, wenn man von einem System ins andere wechselt. Dabei handelt es sich um ein

- System mit ortsfesten Koordinaten (Euler),
- System im mitbewegtem Gas (Lagrange),
- System im mitbewegtem Gitter.

Ein weiteres Kennzeichen der Transformation zwischen den Systemen ist das Auftreten dreier Geschwindigkeiten; jene des Gases, jene des sich auf das Problem adaptierenden Gitters und die daraus resultierende Relativgeschwindigkeit.

3.3.1 SHD-Gleichungen mit totaler Zeitableitung (Lagrange)

Die totale Zeitableitung, auch Lagrange-Formulierung genannt, beschreibt jenes System, in dem sich die Koordinaten mit dem Gas mitbewegen. Das bedeutet, dass sich das Koordinatensystem mit der Gasgeschwindigkeit gegenüber der Umgebung verändert, man *sieht* das Problem aus der Sicht eines sich bewegendes Gasteilchens, man *sitzt* sozusagen auf dem Teilchen. Die bestimmende Geschwindigkeit ist jene des Gases $\vec{u}_{gas}$, die bei Transformation in anderen Systeme Eingang in die Advektionsterme findet.

Die Gleichungen (3.30) bis (3.35) bilden das System der SHD-Gleichungen mit totaler Zeitableitung.

Kontinuitätsgleichung:

$$D_t \rho + \rho \vec{\nabla} \cdot \vec{u} = 0 \quad (3.30)$$

Bewegungsgleichung:

$$D_t(\rho \vec{u}) + (\rho \vec{u}) \vec{\nabla} \cdot \vec{u} + \vec{\nabla} P + \rho \vec{\nabla} \phi - \frac{4\pi}{c} \rho \kappa_P \vec{H} - \vec{u}_Q = 0 \quad (3.31)$$

Energiegleichung:

$$D_t(\rho e) + (\rho e) \vec{\nabla} \cdot \vec{u} + P \vec{\nabla} \cdot \vec{u} + 4\pi \rho \kappa_P (J - S) - \rho (\epsilon_{nuc} + \epsilon_Q) = 0 \quad (3.32)$$

Poissongleichung:

$$\Delta \phi = 4\pi G \rho \quad (3.33)$$

0. Moment der Strahlungstransportgleichung:

$$\begin{aligned} \frac{1}{c} D_t J + \frac{1}{c} J \vec{\nabla} \cdot \vec{u} + \vec{\nabla} \cdot \vec{H} + \\ + \frac{1}{c} K \vec{\nabla} \cdot \vec{u} - \frac{1}{c} \frac{3K - J}{r} \vec{u} + \rho \kappa_P (J - S) = 0 \end{aligned} \quad (3.34)$$

1. Moment der Strahlungstransportgleichung:

$$\frac{1}{c} D_t \vec{H} + \frac{1}{c} \vec{H} \vec{\nabla} \cdot \vec{u} + \vec{\nabla} K + \frac{3K - J}{r} + \frac{1}{c} \vec{H} \vec{\nabla} \cdot \vec{u} + \rho \kappa_P \vec{H} = 0 \quad (3.35)$$

3.3.2 SHD-Gleichungen mit partieller Zeitableitung (Gittersystem)

Die partielle Zeitableitung im Gittersystem entspricht jenem System, in dem man sich mit dem adaptierenden Gitter mitbewegt. Man befindet sich sozusagen

auf dem Gitter und führt alle Bewegungen desselben aus. Die charakteristische Geschwindigkeit ist die Gittergeschwindigkeit $\vec{u}_{rel}$, die bei den Umrechnungen in die Advektionsterme eingeht.

Das Set (3.36) bis (3.41) entspricht den SHD-Gleichungen im Gittersystem.

Kontinuitätsgleichung:

$$d_t \rho + \vec{\nabla} \cdot (\rho \vec{u}_{rel}) = 0 \quad (3.36)$$

Bewegungsgleichung:

$$d_t (\rho \vec{u}) + \vec{\nabla}(\rho \vec{u} \otimes \vec{u}_{rel}) + \vec{\nabla} P + \rho \vec{\nabla} \phi - \frac{4\pi}{c} \rho \kappa_P \vec{H} - \vec{u}_Q = 0 \quad (3.37)$$

Energiegleichung:

$$d_t (\rho e) + \vec{\nabla}(\rho e \vec{u}_{rel}) + P \vec{\nabla} \cdot \vec{u} + 4\pi \rho \kappa_P (J - S) - \rho (\epsilon_{nuc} + \epsilon_Q) = 0 \quad (3.38)$$

Poissongleichung:

$$\Delta \phi = 4\pi G \rho \quad (3.39)$$

0. Moment der Strahlungstransportgleichung:

$$\begin{aligned} \frac{1}{c} d_t J &+ \frac{1}{c} \vec{\nabla} \cdot (J \vec{u}_{rel}) + \vec{\nabla} \cdot \vec{H} + \\ &+ \frac{1}{c} K \vec{\nabla} \cdot \vec{u} - \frac{1}{c} \frac{3K - J}{r} \vec{u} + \rho \kappa_P (J - S) = 0 \end{aligned} \quad (3.40)$$

1. Moment der Strahlungstransportgleichung:

$$\frac{1}{c} d_t \vec{H} + \frac{1}{c} \vec{\nabla}(\vec{H} \vec{u}_{rel}) + \vec{\nabla} K + \frac{3K - J}{r} + \frac{1}{c} \vec{H} \vec{\nabla} \cdot \vec{u} + \rho \kappa_P \vec{H} = 0 \quad (3.41)$$

Bei dem Übergang zwischen zwei Systemen, die durch die Gasgeschwindigkeit und Gittergeschwindigkeit definiert sind, kommt es zu Geschwindigkeitsunterschieden, die durch die Relativgeschwindigkeit $\vec{u}_{rel}$ gegeben sind.

3.3.3 SHD-Gleichungen mit partieller Zeitableitung (Euler)

Die Euler-Formulierung, die *normale* partielle Zeitableitung ist die wohl älteste und bekannteste Betrachtungsweise eines Koordinatensystems. Hierbei handelt es sich um ein ortsfestes Koordinatensystem, man ist also in der Raumkoordinate fest und betrachtet das Gas, das sich durch das fixierte Koordinatensystem bewegt. Das Gas bewegt sich relativ zu den Zellen mit der Gasgeschwindigkeit $\vec{u}_{gas} = \vec{u}$.

Kontinuitätsgleichung:

$$\partial_t \rho + \frac{1}{r^2} \partial_r (r^2 \rho \vec{u}) = 0 \quad (3.42)$$

Bewegungsgleichung:

$$\partial_t (\rho \vec{u}) + \frac{1}{r^2} \partial_r (r^2 \rho \vec{u} \vec{u}) + \partial_r P + \frac{G \rho m}{r^2} - \frac{4\pi}{c} \rho \kappa_P \vec{H} - \vec{u}_Q = 0 \quad (3.43)$$

Energiegleichung:

$$\begin{aligned} \partial_t (\rho e) + \frac{1}{r^2} \partial_r (r^2 \rho e \vec{u}) + \\ + P \frac{1}{r^2} \partial_r (r^2 \vec{u}) + 4\pi \rho \kappa_P (J - S) - \rho (\epsilon_{nuc} + \epsilon_Q) = 0 \end{aligned} \quad (3.44)$$

Poissonsgleichung:

$$\partial_r m = 4\pi \rho r^2 \quad (3.45)$$

0. Moment der Strahlungstransportgleichung:

$$\begin{aligned} \frac{1}{c} \partial_t J + \frac{1}{c} \frac{1}{r^2} \partial_r (r^2 J \vec{u}) + \frac{1}{r^2} \partial_r (r^2 \vec{H}) \\ + \frac{1}{c} K \frac{1}{r^2} \partial_r (r^2 \vec{u}) - \frac{1}{c} \frac{3K - J}{r} \vec{u} + \rho \kappa_P (J - S) = 0 \end{aligned} \quad (3.46)$$

1. Moment der Strahlungstransportgleichung:

$$\begin{aligned} \frac{1}{c} \frac{1}{r^2} \partial_t \vec{H} + \frac{1}{c} \partial_r (r^2 \vec{H} \vec{u}) + \partial_r K + \frac{3K - J}{r} \\ + \frac{1}{c} \vec{H} \frac{1}{r^2} \partial_r (r^2 \vec{u}) + \rho \kappa_P \vec{H} = 0 \end{aligned} \quad (3.47)$$

3.4 Künstliche Viskosität

Neben physikalischen Effekten (molekulare Viskositäten verbreitern Stoßfronten oder Kontaktdiskontinuitäten im Bereich der mittleren Weglänge durch Wärmeleitung) müssen auch numerische Aspekte bei der Berechnung von Stößen in Betracht gezogen werden. Unendlich schmale Stöße oder Kontaktdiskontinuitäten sind mit der Methode der finiten Differenzen (siehe Abschnitt 4) nicht ohne Weiteres aufzulösen, da der Abstand zwischen zwei Gitterpunkten endlich ist. Ohne Einführung **der künstlichen Viskosität** kommt es zu numerischen Instabilitäten in Form von ungewollten Schwingungen, die bei der Berechnung von Stoßwellen auf keinen Fall auftreten sollten, um Ergebnisse nicht zu verfälschen.

Aus diesem Grund wird die hier beschriebene künstliche Viskosität, als numerische Notwendigkeit (für weitere numerische Konstrukte siehe nächster Abschnitt) eingeführt, welche die Stoßfront oder Kontaktdiskontinuität über einige Zellen verschmiert (verbreitert) und so Unstetigkeiten in kontinuierliche Bereiche umwandelt, die sich numerisch besser behandeln lassen. Die physikalische Entsprechung ist der Übergang von den Gleichungen für ein ideales Fluid zu den NAVIER-STOKES-Gleichungen. An diese Art von Viskosität werden drei Anforderungen gestellt:

1. Die künstliche Viskosität soll Stöße über mehrere Gitterpunkte verbreitern und nur in deren Umgebung wirken.
2. Die künstliche Viskosität darf keinen Einfluss auf expandierende Strukturen haben.
3. Homologe Kontraktionen sollen unbeeinflusst stattfinden.

Die genuine Idee der künstlichen Viskosität wurde von *von Neumann* und *Richtmyer* 1950 (siehe [57]) als

$$q = \begin{cases} l^2 \rho \left(\frac{\partial u}{\partial r} \right)^2 & \text{für } \frac{\partial u}{\partial r} < 0 \\ 0 & \text{sonst} \end{cases}$$

publiziert und von *Tscharnutter* und *Winkler* 1979 (siehe [52]) mit oben genannten Eigenschaften auf beliebige Geometrien verallgemeinert. Die in diesem Abschnitt verwendete Notationen stammen aus der Arbeit von *Tscharnutter* und *Winkler*.

Grundlegend ist, dass das Konzept der künstlichen Viskosität geometrieunabhängig formuliert ist und daher für beliebige Geometrien (planar, zylindrisch, sphärisch, Flussröhrengeometrie) gelten kann. Wichtig für astrophysikalische Anwendungen ist von jeher jene für Kugeln.

Die Idee der künstlichen Viskosität basiert auf der Einführung eines viskosen Drucktensors Q_l^m , der sich wie folgt anschreiben lässt (für die Definition der ε -Terme siehe [52, Seite 31]), hier seien nur die Einträge der Spur unter Gleichung (3.56) angeführt, l ist ein Normierungsparameter, der sich auf das Gitter bezieht und angibt, um welchen Betrag eine auftretende Stoßfront verbreitert wird.

$$Q_l^m = l^2 \rho \operatorname{div} \vec{u} = \begin{pmatrix} \varepsilon_r^r - \frac{1}{3} \operatorname{div} \vec{u} & \varepsilon_\theta^r & \varepsilon_\phi^r \\ \varepsilon_r^\theta & \varepsilon_\theta^\theta - \frac{1}{3} \operatorname{div} \vec{u} & \varepsilon_\phi^\theta \\ \varepsilon_r^\phi & \varepsilon_\theta^\phi & \varepsilon_\phi^\phi - \frac{1}{3} \operatorname{div} \vec{u} \end{pmatrix} \quad (3.48)$$

Durch diese Definition der künstlichen Viskosität ergeben sich zusätzliche Terme in der Bewegungs- und Energiegleichung. In die Bewegungsgleichung geht die durch die Viskosität ausgeübte Kraft, in die Energiegleichung, die Umwandlung von Bewegungsenergie in thermische Energie ein. Den viskosen Kraftbeitrag erhält man durch Divergenzbildung des viskosen Drucktensors.

$$\vec{F}_Q = \frac{\partial Q_l^m}{\partial m} \quad \text{oder} \quad \vec{F}_Q = \vec{\nabla} \cdot \mathbf{Q} \quad (3.49)$$

Die dissipierte Energie ergibt sich wie folgt (wobei ε_l^m der Tensor des symmetrisierten Geschwindigkeitsfeldes ist {siehe [52, Seite 31]}), die zweiten Schreibweisen stammen aus [16, Seite 285]).

$$\epsilon_Q = -\frac{1}{\rho} Q_l^m \varepsilon_l^m \quad \text{oder} \quad \epsilon_Q = -\frac{1}{\rho} \mathbf{Q} \cdot (\vec{\nabla} \cdot \vec{u}) \quad (3.50)$$

Zur leichteren Verständlichkeit der nachstehenden Herleitungen seien einige (Operator-)Beziehungen in den folgenden Boxen zusammengefasst. Auf diese wird immer wieder zurückgegriffen; ihre Verwendung ist an den Ziffern über den Gleichheitszeichen zu erkennen.

I - Operatorbeziehung 1

$$\frac{\partial}{\partial r^3} = \frac{4\pi}{3} \frac{\partial}{\partial V}$$

II - Operatorbeziehung 2

X ist Platzhalter für eine beliebige Größe

$$\begin{aligned} \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 X) &= \frac{1}{r^2} \left(\frac{\partial r^2}{\partial r} X + r^2 \frac{\partial X}{\partial r} \right) = \\ &= \frac{1}{r^2} 2r X + \frac{1}{r^2} r^2 \frac{\partial X}{\partial r} = \frac{2}{\mathbf{r}} \mathbf{X} + \frac{\partial \mathbf{X}}{\partial \mathbf{r}} \\ 3 \frac{\partial}{\partial r^3} (r^2 X) &= 3 \left(\frac{\partial r^2}{\partial r^3} X + r^2 \frac{\partial X}{\partial r^3} \right) = \\ &= 3 \left(\frac{\partial (r^3)^{\frac{2}{3}}}{\partial r^3} X + r^2 \frac{\partial X}{\partial r^3} \right) = 3 \left(\frac{2}{3} \frac{1}{r} X + r^2 \frac{\partial X}{\partial r^3} \right) = \\ &= \frac{2}{\mathbf{r}} \mathbf{X} + 3 \mathbf{r}^2 \frac{\partial \mathbf{X}}{\partial \mathbf{r}^3} \\ \text{somit gilt} \quad \frac{\partial X}{\partial r} &= 3 r^2 \frac{\partial X}{\partial r^3} \\ \text{oder} \quad \frac{1}{r^2} \frac{\partial X}{\partial r} &= 3 \frac{\partial X}{\partial r^3} \\ \text{oder} \quad \frac{1}{3} \frac{1}{r^2} \frac{\partial X}{\partial r} &= \frac{\partial X}{\partial r^3} \end{aligned}$$

III - Operatorbeziehung 3

X ist Platzhalter für eine beliebige Größe

$$\begin{aligned}\frac{3}{r} \frac{\partial(r^3 X)}{\partial r^3} &= \frac{3}{r} \left(\frac{\partial r^3}{\partial r^3} X + r^3 \frac{\partial X}{\partial r^3} \right) = \\ &= \frac{3}{r} X + \frac{3}{r} r^3 \frac{\partial X}{\partial r^3} = \frac{3}{r} X + 3r^2 \frac{\partial X}{\partial r^3}\end{aligned}$$

IV - Operatorbeziehung 4

X ist Platzhalter für eine beliebige Größe

$$\begin{aligned}\frac{\partial u}{\partial r} - \frac{\partial(r^2 u)}{\partial r^3} &\stackrel{(\text{II})}{=} \frac{\partial u}{\partial r} - \frac{1}{3} \frac{1}{r^2} \frac{\partial(r^2 u)}{\partial r} = \\ &= \frac{\partial u}{\partial r} - \frac{1}{3} \frac{1}{r^2} \left(\frac{\partial r^2}{\partial r} u + r^2 \frac{\partial u}{\partial r} \right) = \frac{1}{3} \frac{1}{r^2} \left(2ru + r^2 \frac{\partial u}{\partial r} \right) \\ &= \frac{\partial u}{\partial r} - \frac{1}{3} \frac{1}{r^2} 2ru - \frac{1}{3} \frac{1}{r^2} r^2 \frac{\partial u}{\partial r} = \frac{2}{3} \frac{\partial u}{\partial r} - \frac{2}{3} \frac{u}{r} \\ &= \frac{2}{3} \left(\frac{\partial u}{\partial r} - \frac{u}{r} \right)\end{aligned}$$

V - Spurfreiheit des viskosen Drucktensors

entspricht Bedingung 3 der künstlichen Viskosität

$$\begin{aligned}Q_\theta^\theta + Q_\phi^\phi + Q_r^r &= 0 \\ Q_r^r &= -(Q_\theta^\theta + Q_\phi^\phi)\end{aligned}$$

3.4.1 Künstliche Viskosität in der Bewegungsgleichung

Der Zusatzterm des viskosen Druckes für die Bewegungsgleichung lässt sich nun als der radiale Term in sphärischer Symmetrie, das heißt Ersetzung von

$$\text{div } \vec{u} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u), \quad (3.51)$$

sowie das Vernachlässigen aller Ableitungen ungleich derer nach r , und unter entsprechender Ersetzung der Operatoren wie folgt anschreiben.

$$\frac{\partial Q_l^m}{\partial m} = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 Q_r^r) - \frac{Q_\theta^\theta + Q_\phi^\phi}{r}, \quad (3.52)$$

wobei die Q -Terme, die Einträge im viskosen Drucktensor repräsentieren.

Mittels der in den Boxen angeschriebenen Beziehungen lässt sich nun der gesuchte Zusatzterm in der Bewegungsgleichung ableiten:

$$\frac{\partial Q_l^m}{\partial m} \stackrel{(V)}{=} \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 Q_r^r) + \frac{Q_r^r}{r} \quad (3.53)$$

Führt man die Differentiation durch, gelangt man zu

$$\begin{aligned} \frac{1}{r^2} \frac{\partial r^2}{\partial r} Q_r^r + \frac{1}{r^2} r^2 \frac{\partial Q_r^r}{\partial r} + \frac{Q_r^r}{r} &= \frac{1}{r^2} 2r Q_r^r + \frac{\partial Q_r^r}{\partial r} + \frac{Q_r^r}{r} = \\ &= \frac{2}{r} Q_r^r + \frac{\partial Q_r^r}{\partial r} + \frac{Q_r^r}{r} \stackrel{(II)}{=} \frac{3}{r} Q_r^r + 3r^2 \frac{\partial Q_r^r}{\partial r^3}. \end{aligned} \quad (3.54)$$

Das Ergebnis von (3.54) lässt sich weiters unter der Verwendung von (III) zu

$$(3.54) \stackrel{(III)}{=} \frac{3}{r} \frac{\partial (r^3 Q_r^r)}{\partial r^3} \quad (3.55)$$

umformen. Setzt man nun die entsprechenden Terme aus den Einträgen des viskosen Drucktensors ein, wobei

$$\begin{aligned} Q_r^r = \varepsilon_r^r - \frac{1}{3} \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u) \quad \text{mit} \quad \varepsilon_r^r &= \frac{\partial u}{\partial r}, \quad \text{sowie} \quad \varepsilon_\theta^\theta = \frac{u}{r} + \frac{1}{r} \frac{\partial v}{\partial \theta} = 0 \quad \text{und} \\ \varepsilon_\phi^\phi &= \frac{u}{r} + \frac{v}{r} \cot \theta + \frac{1}{\partial r \sin \theta} \frac{\partial w}{\partial \phi} = 0 \end{aligned} \quad (3.56)$$

gilt, ein, ergibt sich aus Gleichung (3.55) Folgendes:

$$\begin{aligned} (3.55) &= \frac{3}{r} \frac{\partial}{\partial r^3} \left(r^3 l^2 \rho \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u) \left[\frac{\partial u}{\partial r} - \frac{1}{3} \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u) \right] \right) = \\ &\stackrel{(II)}{=} \frac{3}{r} \frac{\partial}{\partial r^3} \left(r^3 l^2 \rho 3 \frac{\partial}{\partial r^3} (r^2 u) \left[\frac{\partial u}{\partial r} - \frac{1}{3} \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u) \right] \right) = \\ &\stackrel{(I,IV)}{=} \frac{8\pi}{3r} \frac{\partial}{\partial V} \left(r^3 l^2 \rho 4\pi \frac{\partial}{\partial V} (r^2 u) \left[\frac{\partial u}{\partial r} - \frac{u}{r} \right] \right) \end{aligned} \quad (3.57)$$

Wird nun der Term

$$l^2 4\pi \frac{\partial}{\partial V} (r^2 u) \quad (3.58)$$

herausgehoben und mit einem linearen Term (siehe [19, Seite 39]) kombiniert, erhält man den Koeffizienten der künstlichen Viskosität μ_Q

$$\mu_Q = -l_1 c_s + l_2^2 4\pi \frac{\partial}{\partial V} (r^2 u). \quad (3.59)$$

Der lineare Term skaliert mit der Schallgeschwindigkeit c_s und dient dazu, kleine Oszillationen in der Nähe von Kontaktdiskontinuitäten zu dämpfen. Der quadratische Term wirkt nur im kompressiven und nicht homologen Fall, anders ausgedrückt, nur wenn

$$l_2^2 \frac{\partial}{\partial V} (r^2 u) < 0 \quad (3.60)$$

gilt.

Die endgültige Form dieses Terms in der Bewegungsgleichung ist nun:

$$(3.57) = \frac{8\pi}{3r} \frac{\partial}{\partial V} \left(r^3 \rho \left[-l_1 c_s + l_2^2 4\pi \frac{\partial}{\partial V} (r^2 u) \right] \left[\frac{\partial u}{\partial r} - \frac{u}{r} \right] \right). \quad (3.61)$$

Integriert man nun diesen Ausdruck und setzt die Vorzeichen richtig, erhält man

$$\int_V u_Q dV = - \int_V \frac{8\pi}{3r} \frac{\partial}{\partial V} \left(r^3 \rho \left[-l_1 c_s + l_2^2 4\pi \frac{\partial}{\partial V} (r^2 u) \right] \left[\frac{\partial u}{\partial r} - \frac{u}{r} \right] \right) dV, \quad (3.62)$$

und mit Gleichung (3.59) findet man

$$\int_V u_Q dV = - \int_V \frac{8\pi}{3r} \frac{\partial}{\partial V} \left[\mu_Q r^3 \rho \left(\frac{\partial u}{\partial r} - \frac{u}{r} \right) \right] dV. \quad (3.63)$$

Der zusätzliche Term in der Bewegungsgleichung lautet diskretisiert (siehe Kapitel 4) nun:

$$(3.63) \mapsto -\frac{8\pi}{3r} \Delta \left[\mu_Q \rho \bar{r}^3 \left(\frac{\Delta u}{\Delta r} - \frac{\bar{u}}{\bar{r}} \right) \right] \quad (3.64)$$

3.4.2 künstliche Viskosität in der Energiegleichung

Die durch die künstliche Viskosität dissipierte Energie kann durch

$$\epsilon_Q = -l^2 \operatorname{div} u \left\{ \frac{1}{3} [(\epsilon_r^r - \epsilon_\theta^\theta)^2 + (\epsilon_r^r - \epsilon_\phi^\phi)^2 + (\epsilon_\phi^\phi - \epsilon_\theta^\theta)^2] + 2[(\epsilon_\phi^r \epsilon_r^\phi + \epsilon_\theta^r \epsilon_r^\theta + \epsilon_\theta^\phi \epsilon_\phi^\theta)] \right\} \quad (3.65)$$

angegeben werden. Verwendet man wieder Gleichung (3.51) und bezieht man wieder nur die radialen Terme mit ein, erhält man

$$\begin{aligned} \epsilon_Q &\stackrel{(I)}{=} -l^2 \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u) \left\{ \frac{1}{3} \left[\left(\frac{\partial u}{\partial r} - \frac{u}{r} \right)^2 + \left(\frac{\partial u}{\partial r} - \frac{u}{r} \right)^2 \right] \right\} = \\ &= -\frac{2}{3} l^2 \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 u) \left\{ \left(\frac{\partial u}{\partial r} - \frac{u}{r} \right)^2 \right\}. \end{aligned} \quad (3.66)$$

Verwendet man nun die Beziehungen (IV) in umgekehrter Richtung (Achtung auf das Quadrat der Klammer und damit, dass statt dem Faktor $\frac{2}{3}$, der von $\frac{4}{9}$ herausgehoben werden kann) und (II), gelangt man zur Formulierung wie in [52] auf Seite 35 mit

$$\epsilon_Q = \frac{9}{2} l^2 \frac{\partial}{\partial r^3} (r^2 u) \left[\frac{\partial u}{\partial r} - \frac{\partial(r^2 u)}{\partial r^3} \right]^2 \quad (3.67)$$

angegeben. Nun wird Operatorbeziehung (I) angewendet und Gleichung (3.59) eingesetzt und man erhält

$$(3.67) = -\frac{2}{3} \mu_Q \rho \left[\frac{\partial u}{\partial r} - \frac{u}{r} \right]^2. \quad (3.68)$$

Integriert lautet dieser Term wie folgt

$$\int_V \epsilon_Q dV = - \int_V \frac{2}{3} \mu_Q \rho \left[\frac{\partial u}{\partial r} - \frac{u}{r} \right]^2 dV, \quad (3.69)$$

was zu der diskretisierten Version

$$(3.69) \quad \mapsto \quad -\frac{2}{3} \mu_Q \rho V_S \left(\frac{\Delta u}{\Delta r} - \frac{\bar{u}}{\bar{r}} \right)^2 \quad (3.70)$$

führt.

3.5 Weitere notwendige mathematisch-nummerische Konstruktionen

Neben der künstlichen Viskosität sind noch einige weitere Konstrukte mathematisch-nummerischer Natur von Nöten, um die SHD-Gleichungen mit hoher Genauigkeit lösen zu können (siehe [59]). Die drei wichtigsten seien hier genannt, dabei handelt es sich um das Konzept des adaptiven Gitters, die räumliche und zeitliche Glättung, sowie das Prinzip der zeitlichen Zentrierung.

3.5.1 Adaptive Gitter

Um steile Gradienten beziehungsweise Stoßfronten und Kontaktdiskontinuitäten gut auflösen zu können, ist es notwendig, dass in diesen Bereichen *genauer* gerechnet wird, als an Stellen, wo sich recht wenig in den Variablen ändert. Diese Anforderungen können mit einem sogenannten adaptiven Gitter gelöst werden, dass an sensiblen Stellen des Raumes die Gitterpunkte anhäuft und so die Gradienten besser auflösen kann. Die ursprüngliche Idee wurde erstmals in [13] vorgestellt.

Die zentrale Idee des adaptiven Gitters (ausgezeichnete Literatur zu Gittern und deren Generierung findet sich in [50], [58], [59] und [60]) ist jene, dass die zeitliche Entwicklung der räumlichen Gitterpunktverteilung parallel mit dem System der SHD-Gleichungen mitgelöst wird und durch die Struktur der Strömung bestimmt wird. Dort wo es zu steilen Gradienten kommt, sollen sich die Punkte ansammeln, während sie an flachen Gradienten auseinandergezogen werden. Die Gittergleichung sagt nun im Grunde aus, dass die tatsächliche Punktkonzentration n_i proportional zur gewünschten Auflösung R_i sein soll, also

$$n_i = \frac{X_i}{r_{i+1} - r_i} \propto R_i, \quad (3.71)$$

mit $i = 1, \dots, N$ gilt, wobei i den Stützstellenindex markiert, N die Gesamtanzahl der Gitterpunkte repräsentiert und X eine typische Längenskala des Problems darstellt. Die gewünschte Punktkonzentration wird erreicht, indem man fordert,

dass sich die Punkte äquidistant entlang der Bogenlänge der aufzulösenden Funktionen $f_1, \dots, f_M$ anordnen sollen, womit

$$R_i = \sqrt{1 + \sum_{k=1}^M \left(\frac{\Delta f_k}{\Delta r} \right)_i^2} \quad (3.72)$$

gilt. Der Ausdruck $\left(\frac{\Delta f_k}{\Delta r} \right)_i$ repräsentiert dabei die diskretisierte erste Ableitung der Funktion f_k an der Stelle i . Das führt zu einer Gittergleichung der Form laut [13]

$$\frac{\hat{n}_{i-1}}{R_{i-1}} = \frac{\hat{n}_i}{R_i} \quad (3.73)$$

mit

$$\hat{n}_i = \tilde{n}_i + \tau \frac{\delta \tilde{n}_i}{\delta t} \quad (3.74)$$

$$\tilde{n}_i = n_i - \alpha(\alpha - 1)(n_{i+1} - 2n_i + n_{i-1}) \quad (3.75)$$

Für die Parameter τ und α siehe folgende Unterabschnitte.

3.5.2 Räumliche und zeitliche Glättung

Die Anpassungsfähigkeit des Gitters an die Funktionen sollte aus Gründen der numerischen Stabilität, weder in räumlicher noch in zeitlicher Hinsicht allzu groß sein. Deswegen verwendet man eine räumliche und zeitliche Glättung unter Einführung der Gittersteifeparameter α und τ , die die Änderung der Punktkonzentration von einem Zeitschritt zum nächsten beschränken. Der Parameter α engt die Änderung der Punktkonzentration auf 20 bis 30 Prozent von einem Zeitschritt zum nächsten ein. Der zeitliche Gittersteifeparameter τ hängt von den Zeitskalen des betrachteten Problems ab, muss aber um einiges kleiner als die kürzeste physikalische Zeitskala sein, damit das Gitter in der Lage sein kann, sich auf diese einstellen zu können. Die endgültige Version der Gittergleichung stellt Gleichung (3.73) dar (für eine detaillierte Herleitung sei auf [13] verwiesen).

3.5.3 Zeitliche Zentrierung

Die Berechnung von Pulsationslösungen bedingt ein besonderes Augenmerk auf die zeitliche Entwicklung des Problems. Insbesondere muss die numerische Dämpfung bezüglich dieser Entwicklung auf ein Minimum reduziert werden. Dies geschieht durch die zeitliche Zentrierung; die Ableitungen in der Zeit können von den radialen Termen getrennt werden und eine Taylor-Reihen-Entwicklung für die Anwendung der finite-Differenzen-Methode durchgeführt werden. Eine genaue Ableitung unten angeführter Gleichungen findet sich in [16, Seite 279].

Oben zitierte Ableitung führt zu einem linearen Operator, der auf die physikalischen Variablen angewendet Folgendes ergibt:

$$\begin{aligned}\langle X(t) \rangle &= \theta X(t + \delta t) + (1 - \theta)X(t) \\ &= \theta X^{(neu)} + (1 - \theta)X^{(alt)} \quad \text{mit } 0 \leq \theta \leq 1\end{aligned}\tag{3.76}$$

Setzt man $\theta = 1$ ergibt sich ein voll implizites Schema. Als typischer Wert gilt $\theta = 0,51$ (siehe [21] und [22]).

Beim Verfahren der zeitlichen Zentrierung bestehen nun zwei Möglichkeiten:

- zeitliche Zentrierung der Variablen und anschließende Operatorenbildung
- zeitliche Zentrierung des Operators und anschließende Anwendung auf die Variablen

Um die beiden Möglichkeiten der zeitlichen Zentrierung vergleichen zu können, muss jede der beiden Formulierungen in die SHD-Gleichungen implementiert und astrophysikalische Rechnungen mit dem Code durchgeführt werden.

Da aber das Testen der zwei Möglichkeiten ebenfalls der Aufstellung der Jacobi-Matrix und dadurch der Erstellung der Ableitungen der Gleichungen bedarf, ist die zeitliche Zentrierung ein Paradebeispiel für die Verwendung des erstellten Notebooks. Die Berechnung der Ableitungen erfolgt nun automatisch und durch Wegfall eines erklärglichen Teil des Zeitaufwandes kann nun die ideale zeitliche Zentrierung für unterschiedliche astrophysikalische Fragestellungen ohne großen Aufwand gefunden werden.

3.6 SHD-Gleichungen in Integralform

Um ein stabiles, mit den Erhaltungssätzen konsistentes Diskretisierungsverfahren anwenden zu können, wird die sogenannte **finite-Volumen**-Diskretisierung verwendet (siehe [61]). Deshalb ist es notwendig, die SHD-Gleichungen umzuformulieren. Dazu verwendet man das zeitabhängige Volumen V innerhalb einer Oberfläche ∂V und integriert die Gleichungen mit Hilfe des Integralsatzes von Gauss, der verwendet wird, um die Divergenzterme in Flüsse über die Oberfläche umzuwandeln. Dieser Satz liefert den Zusammenhang zwischen einem Integral über ein Volumen V mit dem Integral über die Oberfläche ∂V , welches das vorhergenannte Volumen einschließt, für Näheres siehe [5, Seite 545]. Dabei ist darauf zu achten, dass sich die zeitliche Ableitung jetzt auf einen fixen Gitterpunkt beziehen.

Die Gleichungen (3.77) bis (3.82) bilden das SHD-System in Integralform.

Kontinuitätsgleichung:

$$\frac{\partial}{\partial t} \int_{V(t)} \rho dV + \int_{\partial V(t)} \rho \vec{u}_{rel} d\vec{A} = 0 \quad (3.77)$$

Bewegungsgleichung:

$$\begin{aligned} & \frac{\partial}{\partial t} \int_{V(t)} \rho \vec{u} dV + \int_{\partial V(t)} \rho \vec{u} \vec{u}_{rel} d\vec{A} + \int_{V(t)} \vec{\nabla} P dV + \\ & + \int_{V(t)} \frac{G\rho m}{r^2} dV - \frac{4\pi}{c} \int_{V(t)} \rho \kappa_P \vec{H} dV - \int_{V(t)} \vec{u}_Q dV = 0 \end{aligned} \quad (3.78)$$

Energiegleichung:

$$\begin{aligned} & \frac{\partial}{\partial t} \int_{V(t)} \rho e dV + \int_{\partial V(t)} \rho e \vec{u}_{rel} d\vec{A} + \int_{V(t)} P \vec{\nabla} \cdot \vec{u} dV + \\ & 4\pi \int_{V(t)} \rho \kappa_P (J - S) dV - \int_{V(t)} \rho (\epsilon_{nuc} + \epsilon_Q) dV = 0 \end{aligned} \quad (3.79)$$

Poissonsgleichung:

$$m = \int_0^r 4\pi r'^2 \rho dr' \quad (3.80)$$

0. Moment der Strahlungstransportgleichung:

$$\begin{aligned} & \frac{1}{c} \frac{\partial}{\partial t} \int_{V(t)} J dV + \frac{1}{c} \int_{\partial V(t)} J \vec{u}_{rel} d\vec{A} + \\ & + \int_{V(t)} \vec{\nabla} \cdot \vec{H} dV + \frac{1}{c} \int_{V(t)} K \vec{\nabla} \cdot \vec{u} dV - \\ & + \frac{1}{c} \int_{V(t)} \frac{3K - J}{r} \vec{u} dV + \int_{V(t)} \rho \kappa_P (J - S) dV = 0 \end{aligned} \quad (3.81)$$

1. Moment der Strahlungstransportgleichung:

$$\begin{aligned} & \frac{1}{c} \frac{\partial}{\partial t} \int_{V(t)} \vec{H} dV + \frac{1}{c} \int_{\partial V(t)} \vec{H} \vec{u}_{rel} d\vec{A} + \int_{V(t)} \vec{\nabla} K dV + \\ & + \int_{V(t)} \frac{3K - J}{r} dV + \frac{1}{c} \int_{V(t)} \vec{H} \vec{\nabla} \cdot \vec{u} dV + \int_{V(t)} \rho \kappa_P \vec{H} dV = 0 \end{aligned} \quad (3.82)$$

Dieser Schritt der Integration ist Voraussetzung für eine konservative Formulierung der diskretisierten Gleichungen. Jetzt kann die nachfolgende Diskretisierung begonnen werden, indem man immer das Volumen V zwischen zwei benachbarten Zellen verwendet; aus diesem Grund heißt diese Methode eben finite Volumen Methode.

Kapitel 4

Diskretisierung

Um Differentialgleichungen mittels Computer lösen zu können, müssen diese zuerst diskretisiert werden. Es handelt sich dabei um einen Übergang von kontinuierlicher Mathematik zu einer diskreten Formulierung, da es mittels Computernumerik nur möglich ist, Gleichungen, Differentiale und Integrale über diskreten Punkten auszuwerten. Für eine grundlegende Betrachtung vom Übergang kontinuierlicher Mathematik zur diskreten Numerik siehe [42]. Eine philosophische Übersicht über die Problematik liefert [51].

Beim Vorgang der Diskretisierung, von dem es je nach Problem, mathematischer Genauigkeit und systembezogener Parameter, unterschiedliche Möglichkeiten gibt, müssen bestimmte Regeln befolgt werden die im folgenden Abschnitt beschrieben sind. Der Vorgang der Diskretisierung ist ein sensibler Vorgang, bei dem viel Einfluss auf die zu erwartende Performace eines Programms genommen werden kann, Fehler beim Diskretisieren können zur Verletzung physikalische Annahmen, sowie der konservativen Formulierung der SHD-Gleichungen führen.

4.1 symbolische Notation

Alle in den (diskretisierten) SHD-Gleichungen verwendeten Variablen und Abkürzungen, sowie ihre Art (Vektor oder Skalar) einschließlich ihrer Dimension finden sich in der Tabelle (4.1).

Bei der diskretisierten Form der Variablen gibt es zwei unterschiedliche Arten, nämlich

- skalare (Tensor gerader Stufe) Variablen - definiert in der Zellmitte
- vektorielle (Tensor ungerader Stufe) Variablen - definiert am Zellrand .

Die Notation aller Variablen ist in Abbildung (4.1) dargestellt. Diese Art von Formulierung nennt *staggered mesh* oder gestaffeltes Gitter. Die Nummerierung der Stützstellen wird mit i bezeichnet, mit $i = 1, \dots, N$, wobei der erste Gitterpunkt am **Aussenrand** liegt, der letzte am Innenrand (siehe Abbildung (4.2)).

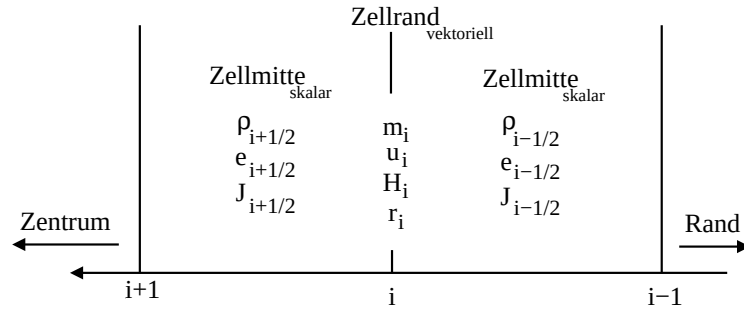
Tabelle 4.1: Bedeutung der in den SHD-Gleichungen verwendeten Größen.

Symbol	Art	Bezeichnung	Dimension (SI)
c	Skalar	Lichtgeschwindigkeit	m s^{-1}
$d\vec{A}$	Vektor	Oberflächenelement von ∂V	m^2
dV	Vektor	Volumenelement von V	m^3
∂V	Vektor	Oberfläche von V	m^2
e	Skalar	spezifische innere Gasenergie	J kg^{-1}
ϵ_{nuc}	Skalar	nukleare Energieerzeugung	$\text{m}^2 \text{s}^{-3}$
ϵ_Q	Skalar	viskose Energiedissipation	$\text{m}^2 \text{s}^{-3}$
G	Skalar	Gravitationskonstante	$\text{N m}^2 \text{kg}^{-2}$
H	Vektor	1-tes Moment des Strahlungsfeldes	$\text{J m}^{-2} \text{s}^{-1}$
$4\pi H$	Vektor	Strahlungsfluss	$\text{J m}^{-2} \text{s}^{-1}$
J	Skalar	0-tes Moment des Strahlungsfeldes	$\text{J m}^{-2} \text{s}^{-1}$
$\frac{4\pi}{c} J$	Skalar	Strahlungsenergiedichte	J m^{-3}
K	Skalar	2-tes Moment des Strahlungsfeldes	$\text{J m}^{-2} \text{s}^{-1}$
$\frac{4\pi}{c} K$	Skalar	Strahlungsdruck	J m^{-3}
κ_P	Skalar	Opazität (Planck-Mittel)	$\text{m}^2 \text{kg}^{-1}$
m	Vektor	integrierte Masse	kg
P	Skalar	Gasdruck	N m^{-2}
ϕ	Skalar	Gravitationspotential	$\text{kg m}^2 \text{s}^{-2}$
π	Skalar	Kreiszahl	—
r	Vektor	Radius	m
ρ	Skalar	Massedichte	kg m^{-3}
S	Skalar	Quellterm	$\text{J m}^{-2} \text{s}^{-1}$
t	Skalar	Zeit	s
$\vec{u}_{\text{gas}}$	Vektor	Gasgeschwindigkeit	m s^{-1}
$\vec{u}_{\text{grid}}$	Vektor	Gittergeschwindigkeit	m s^{-1}
$\vec{u}_{\text{rel}}$	Vektor	Geschwindigkeit zwischen Gas und Gitter	m s^{-1}
u_Q	Vektor	viskoser Impulsübertrag	$\text{kg m}^{-2} \text{s}^{-1}$
V_S	Skalar	skalares Volumen	m^3
V_V	Vektor	vektorielles Volumen	m^3

Neben der Einteilung der Variablen in zwei Klassen, ist dies auch bei den Gleichungen möglich. Auch diese werden in skalare und vektorielle Gleichungen eingeteilt (siehe Tabelle (4.2)). Jene Gleichungen die eine skalare Variable beschreiben werden skalar genannt, Entsprechendes gilt für die vektorielle Klasse (für die einzige Ausnahme siehe Fußnote 9).

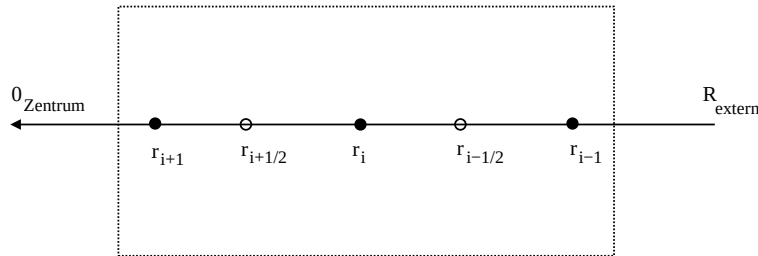
⁹Die diskretisierte Poissongleichung hat skalaren Charakter, die Masse ist aber am Zellrand definiert und daher ein Vektor. Das ist nur scheinbar ein Widerspruch, da die Poissongleichung erst **nach** ein Mal erfolgter Integration diskretisiert wird, *verschiebt* sich die Gleichung sozusagen um einen halben Indexschritt, dadurch verändert sich die Art der Gleichung. Aus

Abbildung 4.1: Zellenbelegung



Die unbekannten Variablen werden auf einem sogenannten *staggered mesh* formuliert, bei dem skalarartige Werte wie die Dichte ρ oder die innere Energie e in der Zellmitte, und vektorielle Werte wie die Geschwindigkeit u oder die Masse m am Zellrand definiert werden.

Abbildung 4.2: Indexverlauf im SHD-Code



Aus historischen Gründen wird der Außenrand des Sterns mit dem Index 0 bezeichnet, hin zum Zentrum erhöht sich der Index. Das wurde aus dem Grund einer weniger speicherintensiven Implementation der äußeren Randbedingungen eingeführt.

4.2 Diskretisierungsregeln

4.2.1 Operatoren

Als Ausgangspunkt der Diskretisierung dienen die SHD-Gleichungen in Integralform von Seite 33. Die räumlichen und zeitlichen Differentiale werden nun auf dem staggered mesh entsprechend diskretisiert. Dabei kommen sieben ver-

diesem Grund beschreibt eine skalare Gleichung eine vektorielle Variable. Die Poissongleichung ist die einzige Ausnahme, alle anderen Gleichungen und zugehörigen Variablen verhalten sich konsistent.

Tabelle 4.2: von der Art der Strahlungshydrodynamikgleichungen

Gleichung	Art	Variable
Gitter	vektoriell	Radius
Poisson (Eigengravitation)	skalar	integrierte Masse ⁹
Kontinuitäts (Masse)	skalar	Dichte
Bewegungs (Impuls)	vektoriell	Gasgeschwindigkeit
Energie	skalar	innere Energie
0.Moment des Strahlungsfeldes	skalar	Strahlungsenergie
1.Moment des Strahlungsfeldes	vektoriell	Strahlungsfluss

schiedene Operatoren (siehe 4.3) zur Anwendung.

Tabelle 4.3: Operatoren

Bezeichnung	Bedeutung	Berechnung
δX	zeitliche Differenz	$X_i^{(neu)} - X_i^{(alt)}$
ΔX	räumliche Differenz, skalar	$X_{i-\frac{1}{2}} - X_{i+\frac{1}{2}}$
	räumliche Differenz, vektoriell ¹⁰	$X_i - X_{i+1}$
$\overline{X}$	Mittelung, skalar ¹¹	$\frac{1}{2}(X_{i-\frac{1}{2}} + X_{i+\frac{1}{2}})$
	Mittelung, vektoriell	$\frac{1}{2}(X_i + X_{i+1})$
$\tilde{X}$	advektierte Größe, skalar ¹²	$\alpha X_{i+\frac{1}{2}} + (1 - \alpha) X_{i-\frac{1}{2}}$
	advektierte Größe, vektoriell ¹²	$\alpha X_i + (1 - \alpha) X_{i-1}$

Der Operator δ für das zeitliche Differential wird mittels der Variablen zum neuen und alten Zeitpunkt an der Stützstelle i berechnet, es ist unabhängig, ob sich dieser Operator in einer skalaren oder vektoriellen Gleichung befindet, die Art der Gleichung hat keinen Einfluss auf die Diskretisierung des zeitlichen

¹⁰Eine Ausnahme bildet hier der Ausdruck r^3 , hier wird zuerst potenziert und danach erst die Differenz gebildet.

¹¹Die Mittelung der Dichte wird volumensnormiert berechnet und fällt aus diesem Berechnungsschema, siehe Gleichung (4.8).

¹²Der kleinere Index wird verwendet, wenn $\vec{u}_{rel} < 0$ gilt, der größere in jedem anderen Fall.

Differentials.

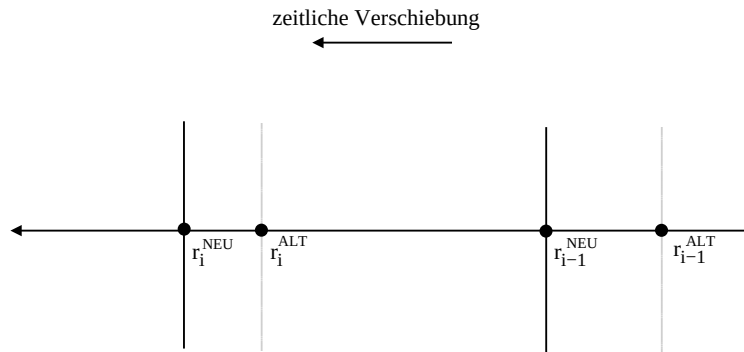
Dies ist anders bei der Diskretisierung des räumlichen Differentials, bei der die Klasse der Gleichung sehr wohl eine Rolle spielt, aus diesem Grund entspricht der Operator Δ jeweils einer Differenz an unterschiedlichen Orten, entweder zwischen zwei Zellrändern (i und $i + 1$) oder zwischen zwei Zellmitten ($i - \frac{1}{2}$ und $i + \frac{1}{2}$).

Dasselbe gilt für Mittelungen von Variablen $\bar{X}$: In skalaren Gleichungen müssen Variablen, die am Zellrand definiert sind, gemittelt werden, um sie sozusagen in eine skalare Größe umwandeln zu können, Entsprechendes gilt selbstverständlich umgekehrt. Es ist ebenfalls geboten, die Advektionsterme $\tilde{X}$ entsprechend ihrer Gleichungsklasse zu diskretisieren, der verwendete Gitterindex wird *upstream* (Strömungsrichtung entscheidet) genommen und hängt von der Relativgeschwindigkeit zwischen Gas und Gitter ab und wird durch einen Schalter indiziert. Es kommt ein einfaches *donor-cell*-Advektionsschema erster Ordnung zur Anwendung, um zu garantieren, dass immer nur Größen stromaufwärts in eine Zelle advektiert werden.

Die Operatoren Δ , $\bar{X}$ und $\tilde{X}$ bedingen eine *Verschiebung* der Klasseneigenschaften. Wird die Differenz eines Skalars ermittelt, ist das Ergebnis ein Vektor und umgekehrt. Es ist daher bei den Gleichungen darauf zu achten, dass nach Anwendung aller Operatoren die Variablen von gleicher Klasse sind. Es kommt vor, dass mehrere Operatoren hintereinander auf eine Variable angewendet werden, hier ist besondere Genauigkeit gefragt, wann und wie oft die Variable ihre Klasse verändert.

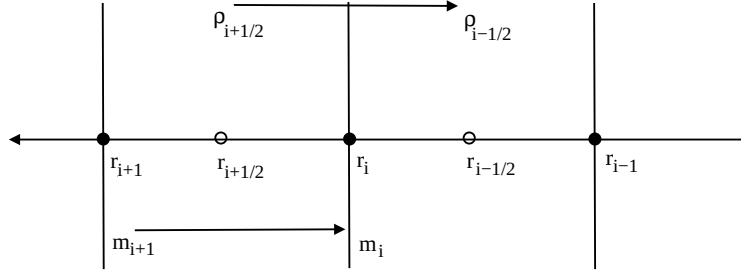
Oben beschriebene Operatorbeziehungen sind in den Abbildungen (4.3), (4.4), (4.5), (4.6) und (4.7) graphisch veranschaulicht.

Abbildung 4.3: Operator zur Berechnung zeitlicher Differenzen ΔX



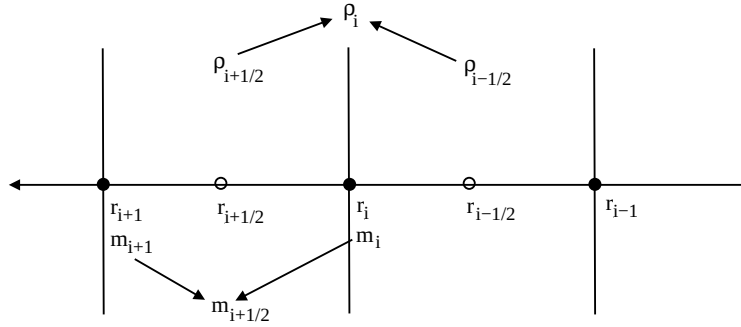
Die Differenz zwischen neuen und alten Werten zwischen den Gitterindizes i und $i + 1$ entspricht dem zeitlichen Differential.

Abbildung 4.4: Operator zur Berechnung räumlicher Differenzen $\overline{\mathbf{X}}$



Das räumliche Differential entspricht je nach Gleichungsklasse einer Differenz zwischen den Gitterindizes i und $i + 1$ oder $i - \frac{1}{2}$ und $i + \frac{1}{2}$.

Abbildung 4.5: Operator zur Berechnung von räumlichen Mitteln



Räumliche Mittel müssen berechnet werden, um skalare Größen auf vektoriell *Gleichungsniveau* zu bringen und vice versa.

4.2.2 Volumens- und Oberflächenterme

Zur Diskretisierung der Volumenelemente dV muss man wieder zuerst die Klasse der Gleichung betrachten. Für eine skalare Gleichung gilt der Zusammenhang

$$V_S = \frac{4\pi}{3}(r_{i-1}^3 - r_i^3). \quad (4.1)$$

Dieser Zusammenhang ist graphisch in Abbildung (4.8) gezeigt. Das skalare Volumen V_S ergibt sich durch eine Differenz zwischen den Gitterpunkten $i - 1$ und i und *lebt* daher in der Zellmitte auf dem Index $i - \frac{1}{2}$.

Das vektorielle Volumen V_V am Gitterindex i wird durch Mittelbildung zweier skalarer Volumina V_S berechnet (siehe auch Abbildung (4.9):

Abbildung 4.6: Advektion in einer skalaren Zelle

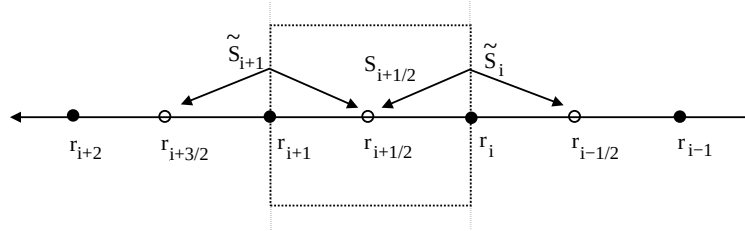
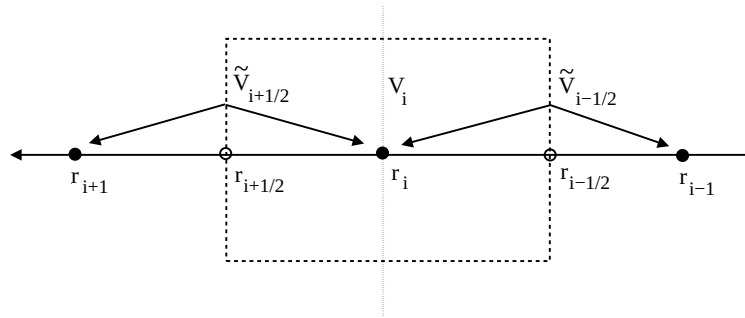


Abbildung 4.7: Advektion in einer vektoriellen Zelle



$$\begin{aligned}
 V_V &= \frac{4\pi}{3}(r_{i-\frac{1}{2}}^3 - r_{i+\frac{1}{2}}^3) = \\
 &= \frac{4\pi}{3} \left[\frac{1}{2}(r_{i-1}^3 + r_i^3) - \frac{1}{2}(r_i^3 + r_{i+1}^3) \right] = \\
 &= \frac{4\pi}{3} \frac{1}{2}(r_{i-1}^3 + r_i^3 - r_i^3 - r_{i+1}^3) = \\
 &= \frac{4\pi}{6}(r_{i-1}^3 - r_{i+1}^3)
 \end{aligned} \tag{4.2}$$

Nach diesen Definitionen können folgende fünf Diskretisierungsregeln definiert werden, mit denen die Volumens- und Oberflächenintegrale, sowie die Differenzterme der SHD-Gleichungen einfach diskretisiert werden können.

Abbildung 4.8: Berechnungsvorschrift für V_S - das skalare Volumen

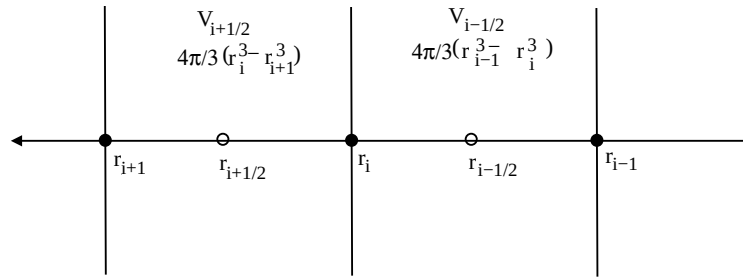
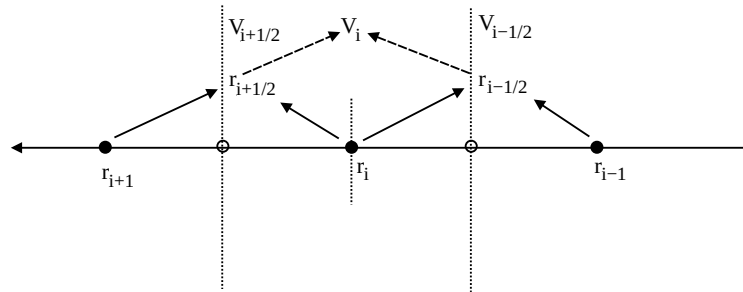


Abbildung 4.9: Berechnungsvorschrift für V_V - das vektorielle Volumen



V_V wird durch Mittelbildung der skalaren Volumina links und rechts vom gewünschten Index berechnet.

Diskretisierungsregel 1

$$\frac{\partial}{\partial t} \int_{V(t)} X_S dV \mapsto \frac{\delta}{\delta t} X_S V_S \quad (4.3)$$

Diskretisierungsregel 2

$$\frac{\partial}{\partial t} \int_{V(t)} X_V dV \mapsto \frac{\delta}{\delta t} X_V V_V \quad (4.4)$$

Diskretisierungsregel 3

$$\int_{\partial V(t)} X \vec{u}_{rel} dV \mapsto 4\pi \Delta(r^2 \tilde{X} \vec{u}_{rel}) \quad (4.5)$$

Diskretisierungsregel 4

$$\begin{aligned} \int_{V(t)} \vec{\nabla} \cdot X dV &= \int_{V(t)} \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 X) dV = \\ &= \int_{V(t)} 3 \frac{\partial}{\partial r^3} (r^2 X) dV \mapsto 3 \frac{1}{r^3} \Delta(r^2 X) \frac{4\pi}{3} r^3 = \\ &= 4\pi \Delta(r^2 X) \end{aligned} \quad (4.6)$$

Diskretisierungsregel 5

$$\int_{V(t)} \vec{\nabla} X dV \mapsto 4\pi r^2 \Delta X \quad (4.7)$$

Da bei der Diskretisierung der SHD-Gleichungen sehr viele Indizes vorkommen, kann es leicht passieren, dass man den einen oder anderen vergisst oder verwechselt. Aus diesem Grund wird hier eine neue Form der Beschreibung der diskretisierten SHD-Gleichungen vorgestellt, um den genannten Problemen aus dem Weg zu gehen. Aus diesem Grund müssen einige weitere Regeln festgelegt werden, damit die Diskretisierung konsistent durchgeführt werden kann. Dabei ist auf Folgendes bei der Diskretisierung der SHD-Gleichungen zu achten:

- Differenzen werden immer nach dem Schema *außen - innen* gebildet und hängen von der Klasse der Gleichung ab.
- Skalare Gleichungen *leben* auf halbzahligen Indizes¹³, das heißt der maßgebende Gitterindex liegt bei $i + \frac{1}{2}$
- Vektorielle Gleichungen *leben* auf ganzzahligen Indizes, der Hauptindex liegt bei i .
- Dementsprechend liegen die Ränder einer skalaren Zelle auf ganzzahligen Indizes,

¹³Da es in FORTRAN keine halbzahligen Indizes gibt, werden die halbzahligen Indizes auf ganzzahligen abgespeichert, und zwar um einen Halbschritt tiefer $i + \frac{1}{2}$ wird zu i und $i - \frac{1}{2}$ zu $i - 1$.

- und die Ränder einer vektoriellen Zelle auf halbzahligen Indizes.
- Mittelungen müssen entsprechend der Klasse der Gleichung durchgeführt werden.

Eine letzte notwendige Größe ist die vektorielle Dichte. Da die Dichte eine skalare Größe ist, muss sie gemittelt werden, um einen Dichtewert am Zellrand zu erhalten. Die Dichte wird laut [14] volumensgemittelt berechnet, um Massenerhaltung für V_S und V_V zu gewährleisten:

$$\begin{aligned}
\rho_i V_V &= \frac{1}{2}(\rho_{i+\frac{1}{2}} V_{S,i+\frac{1}{2}} + \rho_{i-\frac{1}{2}} V_{S,i-\frac{1}{2}}) \\
\rho_i &= \frac{\frac{1}{2}(\rho_{i+\frac{1}{2}} V_{S,i+\frac{1}{2}} + \rho_{i-\frac{1}{2}} V_{S,i-\frac{1}{2}})}{V_V} = \\
&= \frac{\frac{1}{2}[\rho_{i+\frac{1}{2}} \frac{4\pi}{3}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}} \frac{4\pi}{3}(r_{i-1}^3 - r_i^3)]}{\frac{4\pi}{6}(r_{i-1}^3 - r_{i+1}^3)} = \\
&= \frac{\frac{4\pi}{6}[\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)]}{\frac{4\pi}{6}(r_{i-1}^3 - r_{i+1}^3)} = \\
\bar{\rho} &= \frac{\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)}{(r_{i-1}^3 - r_{i+1}^3)} \tag{4.8}
\end{aligned}$$

Zum Schluss sei noch die Diskretisierung des Termes

$$\delta t r_i^2 u_{rel,i} \tilde{\rho}_i \longrightarrow [\delta t r_i^2 u_i - \frac{1}{3}\delta(r_i^3)] \tilde{\rho}_i \tag{4.9}$$

laut [14] angegeben.

Betrachtet man nun die diskretisierten Gleichungen und die für die Berechnung verwendeten Gitterpunkte erkennt man, dass im Fall der hier verwendeten SHD-Gleichungen folgende Stützstellen vorkommen: $i - 2$, $i - 1$, i , $i + 1$ und $i + 2$ (oder einer Verschiebung um einen Halbschritt nach rechts). Das Set der zur Diskretisierung verwendeten Gitterpunkte nennt man **Stempel** (englisch stencil), im hier gezeigten Fall ist es ein Fünferstempel.

Um den Vorgang der Diskretisierung zu veranschaulichen und die Komplexität der am Ende der Ableitungen stehenden Gleichungen anschaulich zu machen, sei hier exemplarisch je eine vektorielle und eine skalare Gleichung vollständig diskretisiert angeschrieben.

Die Vorgehensweise gliedert sich in vier Schritte:

1. Ausgangspunkt ist die Gleichung in Integralform
2. Diskretisierung laut der in diesem Abschnitt vorgestellten Regeln (neue **Kompaktschreibweise ohne Indizes**)
3. Multiplikation der Gleichung mit dem Faktor δt
4. Einsetzen aller Variablen mit zugehörigen Indizes

4.3 Diskretisierung einer vektoriellen Gleichung - Bewegungsgleichung

Bei der Bewegungsgleichung handelt es sich um eine vektorielle Gleichung, weil sie die Geschwindigkeit u beschreibt, die am Zellrand definiert ist. Skalare Größen müssen also gemittelt werden, damit sie in das Schema dieser Gleichung passen. Auf die besondere Art der Dichtemittelung sei noch ein Mal hingewiesen (siehe Gleichung (4.8)). Große Vorsicht ist bei der Diskretisierung der Advektionsterme geboten, da diese Teile der Gleichungen im Allgemeinen am fehleranfälligsten bei der Diskretisierung sind. Desweiteren ist auf die Verschachtelung der Operatoren zu achten, da dies die Komplexität der Terme noch erhöht.

Ausgangspunkt ist die Integralform der Bewegungsgleichung:

$$\begin{aligned} & \frac{\partial}{\partial t} \int_{V(t)} \rho \vec{u} dV + \int_{\partial V(t)} \rho \vec{u} \vec{u}_{rel} dA + \int_{V(t)} \vec{\nabla} P dV + \\ & + \int_{V(t)} \frac{G \rho m}{r^2} dV - \frac{4\pi}{c} \int_{V(t)} \rho \kappa_R \vec{H} dV - \int_{V(t)} \vec{u}_Q dV = 0 \end{aligned}$$

Danach erfolgt die Diskretisierung

$$\begin{aligned} & \xrightarrow{\text{Diskretisierung}} \frac{\delta}{\delta t} (u \bar{\rho} V_V) + 4\pi \Delta((\widetilde{\bar{\rho}u}) \overline{r^2 u^{rel}}) + 4\pi r^2 (\Delta P) + \\ & \frac{Gm}{r^2} \bar{\rho} V_V - \frac{4\pi}{c} \bar{\kappa}_R H \bar{\rho} V_V + \vec{u}_Q V_V = 0, \end{aligned}$$

sowie die Multiplikation mit δt und man erhält die Gleichung diskretisiert in der neuen kompakten Schreibweise:

$$\begin{aligned} & \xrightarrow{\text{Multiplikation mit } \delta t} \underbrace{\delta(u \bar{\rho} V_V)}_A + \underbrace{4\pi \Delta((\widetilde{\bar{\rho}u}) \overline{r^2 u^{rel}} \delta t)}_B + \underbrace{4\pi r^2 (\Delta P) \delta t}_C + \\ & + \underbrace{\frac{Gm}{r^2} \bar{\rho} V_V \delta t}_D - \underbrace{\frac{4\pi}{c} \bar{\kappa}_R H \bar{\rho} V_V \delta t}_E + \underbrace{\frac{8\pi}{3r} \Delta \left\{ \mu_Q \rho \bar{r}^3 \delta t \left(\frac{\Delta u}{\Delta r} - \frac{\bar{u}}{\bar{r}} \right) \right\}}_F = 0 \end{aligned}$$

Um nun auf die Diskretisierung mittels Indizes zu gelangen, müssen nur die zuvor abgeleiteten Variablen eingesetzt werden. Da es sich um eine Vektorgleichung handelt, wird das vektorielle Volumen V_V eingesetzt und alle skalaren Größen gemittelt. Die Gleichung lebt auf dem Index i , deswegen müssen alle Mittelungen sich auf diesen Index beziehen. Der Operator Δ im Advektionsterm wird wie folgt diskretisiert: Mittelwertbildung auf den Index $i - \frac{1}{2}$ über i und $i - 1$ der einen Komponente der Differenz und Mittelwertbildung auf $i + \frac{1}{2}$ für die andere Komponente, damit bei der Differenzbildung durch das diskretisierte räumliche Differential wieder auf den Index i gelangt wird. Die Schalter α für die Advektion sind ebenfalls mit einem Index versehen, der dem Charakter der Gleichung entspricht.

Die einzelnen Terme sind in obiger Gleichung mit den Bezeichnungen A bis F versehen und anschließend einzeln in diskretisierte Form angeschrieben:

$$\begin{aligned}
& \xrightarrow{\text{Term A}} \left(\left[u_i \frac{\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)}{(r_{i-1}^3 - r_{i+1}^3)} \frac{4\pi}{6} (r_{i-1}^3 - r_{i+1}^3) \right]^{(neu)} - \right. \\
& \quad \left. + \left[u_i \frac{\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)}{(r_{i-1}^3 - r_{i+1}^3)} \frac{4\pi}{6} (r_{i-1}^3 - r_{i+1}^3) \right]^{(alt)} \right) + \\
& \xrightarrow{\text{Term B}} + 4\pi \left\{ \frac{1}{2} \left[\left((u_i r_i^2 \delta t) - \frac{1}{3} \left[(r_i^3)^{(neu)} - (r_i^3)^{(alt)} \right] \right) + \left((u_{i-1} r_{i-1}^2 \delta t) - \right. \right. \right. \\
& \quad \left. \left. \left. \frac{1}{3} \left[(r_{i-1}^3)^{(neu)} - (r_{i-1}^3)^{(alt)} \right] \right) \right] * \right. \\
& \quad * \left[\alpha_i \frac{\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)}{(r_{i-1}^3 - r_{i+1}^3)} u_i + \right. \\
& \quad \left. (1 - \alpha_i) \frac{\rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3) + \rho_{i-\frac{3}{2}}(r_{i-2}^3 - r_{i-1}^3)}{(r_{i-2}^3 - r_i^3)} u_{i-1} \right] + \\
& \quad + \frac{1}{2} \left[\left((u_i r_i^2 \delta t) - \frac{1}{3} \left[(r_i^3)^{(neu)} - (r_i^3)^{(alt)} \right] \right) + \left((u_{i+1} r_{i+1}^2 \delta t) - \right. \right. \\
& \quad \left. \left. \left. \frac{1}{3} \left[(r_{i+1}^3)^{(neu)} - (r_{i+1}^3)^{(alt)} \right] \right) \right] * \right. \\
& \quad * \left[\alpha_{i+1} \frac{\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)}{(r_{i-1}^3 - r_{i+1}^3)} u_i + \right. \\
& \quad \left. (1 - \alpha_{i+1}) \frac{\rho_{i+\frac{3}{2}}(r_{i+1}^3 - r_{i+2}^3) + \rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3)}{(r_i^3 - r_{i+2}^3)} u_{i+1} \right] \left. \right\} +
\end{aligned}$$

$$\begin{aligned}
& \xrightarrow{\text{Term C}} + 4\pi r_i^2 (P_{i-\frac{1}{2}} - P_{i+\frac{1}{2}}) \delta t + \\
& \xrightarrow{\text{Term D}} + \frac{Gm_i}{r_i^2} \frac{\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)}{(r_{i-1}^3 - r_{i+1}^3)} \frac{4\pi}{6} (r_{i-1}^3 - r_{i+1}^3) \delta t + \\
& \xrightarrow{\text{Term E}} + \frac{4\pi}{c} \frac{1}{2} (\kappa_{R,i-\frac{1}{2}} - \kappa_{R,i+\frac{1}{2}}) H_i \frac{\rho_{i+\frac{1}{2}}(r_i^3 - r_{i+1}^3) + \rho_{i-\frac{1}{2}}(r_{i-1}^3 - r_i^3)}{(r_{i-1}^3 - r_{i+1}^3)} \frac{4\pi}{6} (r_{i-1}^3 - r_{i+1}^3) \delta t + \\
& \xrightarrow{\text{Term F}} - \frac{8\pi}{3r_i} \left\{ \left[\mu_{Q,i-\frac{1}{2}} \rho_{i-\frac{1}{2}} \frac{1}{2} (r_i^3 + r_{i+1}^3) \delta t \left(\frac{u_{i-1} - u_i}{r_{i-1} - r_i} - \frac{\frac{1}{2}(u_{i-1} - u_i)}{\frac{1}{2}(r_{i-1} - r_i)} \right) \right] - \right. \\
& \quad \left. \left[\mu_{Q,i+\frac{1}{2}} \rho_{i+\frac{1}{2}} \frac{1}{2} (r_i^3 + r_{i+1}^3) \delta t \left(\frac{u_i - u_{i+1}}{r_i - r_{i+1}} - \frac{\frac{1}{2}(u_i - u_{i+1})}{\frac{1}{2}(r_i - r_{i+1})} \right) \right] \right\}
\end{aligned}$$

4.4 Diskretisierung einer skalaren Gleichung - Energiegleichung

Bei der Energiegleichung handelt es sich um eine skalare Gleichung wie auch die innere Energie e eine skalare Variable bezeichnet. Die Gleichung lebt auf den halbzahligen Index $i + \frac{1}{2}$, alle vektoriellen Größen müssen auf diesen Index hin gemittelt werden. Wiederum ist auch hier besonderes Augenmerk auf den Advektionsterm zu richten, der hier aber um einiges unkomplizierter ist, als in der Bewegungsgleichung. Die Schlter für den Advektionsterm haben wiederum dem der Gleichung entsprechen halbzahligen Index.

Noch ein Mal sei darauf hingewiesen, dass alle halbzahligen Indizes in der Speicherung des FORTRAN-Codes auf ganzzahlige (und zwar um einen Halbschritt tieferen) Indizes abgespeichert werden. $\rho_{i+\frac{1}{2}}$ wird auf ρ_i abgespeichert.

Wiederum wird von der Integralform ausgegangen

$$\begin{aligned}
& \frac{\partial}{\partial t} \int_{V(t)} \rho e dV + \int_{\partial V(t)} \rho e \vec{u}_{rel} dA + \int_{V(t)} P \vec{\nabla} \cdot \vec{u} dV + \\
& + 4\pi \int_{V(t)} \rho \kappa_R (J - S) dV - \int_{V(t)} \rho (\epsilon_{nuc} + \epsilon_Q) dV = 0
\end{aligned}$$

und die entsprechenden Diskretisierungsregeln angewendet.

$$\xrightarrow{\text{Diskretisierung}} \frac{\delta}{\delta t} \rho e V_S + 4\pi \Delta((\tilde{\rho} e) r^2 u^{rel}) + 4\pi P \Delta(r^2 u) + \\ 4\pi \rho \kappa_R (J - S) V_S - \rho(\epsilon_{nuc} + \epsilon_Q) V_S = 0$$

und man gelangt zur Kompaktschreibweise der Energiegleichung:

$$\xrightarrow{\text{Multiplikation mit } \delta t} \underbrace{\delta(\rho e V_S)}_A + \underbrace{4\pi \Delta((\tilde{\rho} e) r^2 u^{rel} \delta t)}_B + \underbrace{4\pi P \Delta(r^2 u)}_C + \\ \underbrace{4\pi \rho \kappa_R (J - S) V_S \delta t}_D - \underbrace{\frac{2}{3} \mu_Q \rho V_S \delta t \left(\frac{\Delta u}{\Delta r} - \frac{\bar{u}}{\bar{r}} \right)^2}_E = 0$$

Im skalaren Fall wird nun V_S für das Volumen eingesetzt. Wieder sind die einzelnen Terme der Gleichung mit A bis E beschriftet und die Diskretisierung sieht wie folgt aus:

$$\xrightarrow{\text{Term A}} \left[\rho_{i+\frac{1}{2}} e_{i+\frac{1}{2}} \frac{4\pi}{3} (r_i^3 - r_{i+1}^3)^{(neu)} - \rho_{i+\frac{1}{2}} e_{i+\frac{1}{2}} \frac{4\pi}{3} (r_i^3 - r_{i+1}^3)^{(alt)} \right] +$$

$$\xrightarrow{\text{Term B}} + 4\pi \left\{ \left[u_i r_i^2 \delta t - \frac{1}{3} (r_i^3)^{(neu)} - (r_i^3)^{(alt)} \right] \left[\alpha_{i+\frac{1}{2}} \rho_{i+\frac{1}{2}} e_{i+\frac{1}{2}} + (1 - \alpha_{i+\frac{1}{2}}) \rho_{i-\frac{1}{2}} e_{i-\frac{1}{2}} \right] - \right. \\ \left. \left[u_{i+1} r_{i+1}^2 \delta t - \frac{1}{3} (r_{i+1}^3)^{(neu)} - (r_{i+1}^3)^{(alt)} \right] \left[\alpha_{i+\frac{3}{2}} \rho_{i+\frac{3}{2}} e_{i+\frac{3}{2}} + (1 - \alpha_{i+\frac{3}{2}}) \rho_{i+\frac{1}{2}} e_{i+\frac{1}{2}} \right] \right\} +$$

$$\xrightarrow{\text{Term C}} + 4\pi P_{i+\frac{1}{2}} (r_i^2 u_i - r_{i+1}^2 u_{i+1}) \delta t +$$

$$\xrightarrow{\text{Term D}} + 4\pi \kappa_{i+\frac{1}{2}} \rho_{i+\frac{1}{2}} (J_{i+\frac{1}{2}} - S_{i+\frac{1}{2}}) \frac{4\pi}{3} (r_i^3 - r_{i+1}^3) \delta t -$$

$$\xrightarrow{\text{Term E}} + \frac{2}{3} \mu_{Q,i+\frac{1}{2}} \rho_{i+\frac{1}{2}} \frac{4\pi}{3} (r_i^3 - r_{i+1}^3) \delta t \left(\frac{u_{i-1} - u_i}{r_{i-1} - r_i} - \frac{\frac{1}{2}(u_i - u_{i+1})}{\frac{1}{2}(r_i - r_{i+1})} \right)^2$$

4.5 diskretisierte SHD-Gleichungen

Das System der SHD-Gleichungen in diskretisierter Form schließt das Kapitel über Diskretisierung ab. Alle Gleichungen sind nach den abgeleiteten Regeln bearbeitet worden und in Kompaktschreibweise nach der Multiplikation mit δt angegeben. Im Gegensatz bis zur dato üblichen Schreibweise sind die Gleichungen **nicht** durch 4π dividiert worden. Dieselben Gleichungen finden sich zusammengefasst auf Seite ?? im Anhang.

Kontinuitätsgleichung:

$$\delta(\rho V_S) + 4\pi \Delta(\tilde{\rho} r^2 u^{rel} \delta t) = 0 \quad (4.10)$$

Bewegungsgleichung:

$$\begin{aligned} \delta(u \bar{\rho} V_V) + 4\pi \Delta((\tilde{\rho} u) \bar{r}^2 \overline{u_{rel}} \delta t) + 4\pi r^2 (\Delta P) \delta t + \frac{Gm}{r^2} \bar{\rho} V_V \delta t - \\ \frac{4\pi}{c} \overline{\kappa_R} H \bar{\rho} V_V \delta t + \frac{2\pi}{3r} \Delta \left\{ \mu_Q \rho \bar{r}^3 \delta t \left(\frac{\Delta u}{\Delta r} - \frac{\bar{u}}{\bar{r}} \right) \right\} = 0 \end{aligned} \quad (4.11)$$

Energiegleichung:

$$\begin{aligned} \delta(\rho e V_S) + 4\pi \Delta((\tilde{\rho} e) r^2 u_{rel} \delta t) + 4\pi P \Delta(r^2 u) \delta t + \\ + 4\pi \kappa_R \rho (J - S) V_S \delta t - \frac{2}{3} \mu_Q \rho V_S \delta t \left(\frac{\Delta u}{\Delta r} - \frac{\bar{u}}{\bar{r}} \right)^2 = 0 \end{aligned} \quad (4.12)$$

Poissongleichung:

$$\Delta m = \rho V_S \quad (4.13)$$

0. Moment der Strahlungstransportgleichung:

$$\begin{aligned} \frac{1}{c} \delta(J V_S) + \frac{4\pi}{c} \delta t \Delta(r^2 u^{rel} \tilde{J}) + 4\pi \delta t \Delta(r^2 H) + \frac{4\pi}{c} \delta t K \Delta(r^2 u) - \\ + \frac{1}{c} \delta t (3K - J) \frac{\bar{u}}{\bar{r}} V_S + \kappa_R \rho \delta t (J - S) V_S = 0 \end{aligned} \quad (4.14)$$

1. Moment der Strahlungstransportgleichung:

$$\begin{aligned} \frac{1}{c} \delta(H V_V) + \frac{4\pi}{c} \delta t \Delta(\bar{r}^2 \tilde{H} \overline{u^{rel}}) + 4\pi r^2 \delta t \Delta K + \delta t \frac{3\bar{K} - \bar{J}}{r} V_V + \\ + \frac{4\pi}{c} \delta t H \Delta(\bar{r}^2 \bar{u}) + \overline{\kappa_R} \delta t H \bar{\rho} V_V = 0 \end{aligned} \quad (4.15)$$

Kapitel 5

Lösung des SHD-Systems

In diesem Kapitel werden grundlegende Eigenschaften der Jacobi-Matrix, sowie das Lösen des SHD-Gleichungssystems besprochen. Eine gute Einführung in das Thema numerisches Lösen von Gleichungssystemen liefern [53] und [54], ein umfassendes Bild auf hohem Niveau geben [17] und [69].

5.1 Allgemeines

Um nun das SHD-Gleichungssystem, bei dem es sich um ein System impliziter, nichtlinearer, partieller, diskretisierter Differentialgleichung handelt, lösen zu können, verwendet man das iterative Newton-Raphson-Verfahren. In diesem Verfahren muss die sogenannte Jacobi-Matrix (siehe [18]) erstellt werden. Bei dieser Matrix handelt es sich um die Funktional-Matrix, das bedeutet, dass die Koeffizienten der Matrix die Ableitungen aller Gleichungen des Systems nach allen Variablen des Systems repräsentieren.

Auf Grund des diskreten Differenzenschemas besitzt die Jacobi-Matrix eine besondere Struktur mit einer hohen Anzahl von Null-Einträgen (siehe Abschnitt 5.3.1).

Ein weiterer Schritt des Lösungsverfahrens beinhaltet die Invertierung der Jacobi-Matrix. Der dabei auftretende Rechenaufwand ist beträchtlich, darüberhinaus ist die Inversion einer Matrix mit vielen Einträgen gleich Null eine besondere Herausforderung an die verwendete Numerik.

5.2 Lösungsverfahren

In diesem Abschnitt wird das Verfahren, mit dem man das diskretisierte SHD-Gleichungssystem lösen kann, näher beschrieben.

Definiert man alle unbekannten Variablen X am Gitterpunkt i zum Zeitpunkt k durch

$$\mathbf{X} = X_i^k = (X_{1,i}^k, \dots, X_{M,i}^k) = (\rho_i^k, u_i^k, e_i^k, J_i^k, \dots), \quad (5.1)$$

kann das System der nichtlinearen Differentialgleichungen wie folgt angeschrieben werden:

$$G_m(\langle X_i \rangle) = 0 \quad (5.2)$$

mit m von 1 bis M . Dieses System ist hoch nichtlinear und enthält M Gleichungen und N Gitterpunkte, was einer Anzahl von $M \times N$ Unbekannten entspricht. Grundsätzlich kann so ein Gleichungssystem explizit oder implizit gelöst werden, abhängig davon, wie die Größen zum neuen Zeitpunkt $(n + 1)$ berechnet werden. Bei einem expliziten Verfahren werden die Werte zum neuen Zeitpunkt durch Vorwärtsextrapolation mit dem Zeitschritt δt aus den Werten zum alten Zeitpunkt errechnet. Dieser Zeitschritt ist allerdings nicht beliebig wählbar, sondern wird durch die CFL-Bedingung (siehe [9]) limitiert. Diese Beschränkung von δt ist der entscheidende Nachteil von expliziten Lösungsverfahren.

Bei einem **impliziten** Verfahren wird bei der Berechnung der Werte zum neuen Zeitpunkt bereits eben diese verwendet. Das dadurch entstandene nichtlineare, gekoppelte, algebraische Gleichungssystem der Unbekannten zum neuen Zeitpunkt lässt sich nur durch ein Näherungsverfahren berechnen. Ein solches System muss iterativ gelöst werden, was mittels des **Newton-Raphson-Verfahrens** (die eindimensionale Entsprechung ist in [45] beschrieben, die graphische Entsprechung ist in Abb. 5.1 dargestellt) geschieht. Der Zeitschritt δt ist nur mehr von der Konvergenz des Verfahrens abhängig und nicht mehr durch die CFL-Bedingung beschränkt, was höhere Zeitschritte ermöglicht. Findet das Verfahren keine Nullstelle, wird der Zeitschritt so lange verkleinert, bis Konvergenz eintritt.

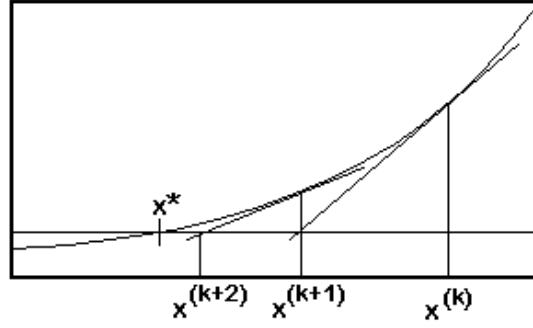
Das Newton-Raphson-Verfahren wird wie folgt durchgeführt. Zuerst wird eine Taylorreihenentwicklung um den neuen Zeitpunkt $\mathbf{X}^{(i+1)}$ durchgeführt

$$G(\mathbf{X}^{(i+1)}) = G(\mathbf{X}^{(i)}) + \frac{\partial G}{\partial \mathbf{X}^{(i)}}(\mathbf{X}^{(i+1)} - \mathbf{X}^{(i)}) = 0, \quad (5.3)$$

wobei eben davon ausgegangen wird, dass $\mathbf{X}^{n+1}$ bereits eine Lösung des Systems darstellt.

Setzt man nun beispielsweise $G(\mathbf{X}^{(i+1)})$ gleich Null und führt die Ersetzung

Abbildung 5.1: graphische Darstellung des eindimensionalen Newton-Verfahrens



Das Newton-Raphson-Verfahren ist ein Lösungsverfahren für nichtlineare Gleichungssysteme, das hier dargestellte Newton-Verfahren ist die eindimensionale Entsprechung. $x^{(k)}$ entspricht dem Startwert und x^* dem Fixpunkt, der Nullstelle.

$$\delta \mathbf{X} = \mathbf{X}^{(i+1)} - \mathbf{X}^{(i)} \quad (5.4)$$

durch, gelangt man zu

$$\begin{aligned} 0 &= G(\mathbf{X}^{(i)}) + \frac{\partial G}{\partial \mathbf{X}^{(i)}} \delta \mathbf{X} \\ -G(\mathbf{X}^{(i)}) &= \frac{\partial G}{\partial \mathbf{X}^{(i)}} \delta \mathbf{X} \\ \delta \mathbf{X} &= -\left(\frac{\partial G}{\partial \mathbf{X}}\right)^{-1} G(\mathbf{X}^{(i)}), \end{aligned} \quad (5.5)$$

wobei $\left(\frac{\partial G}{\partial \mathbf{X}}\right)^{-1}$ für die inverse Jacobi-Matrix steht. Wird Gleichung (5.4) in umgekehrter Richtung angewandt, ergibt sich

$$\mathbf{X}^{(i+1)} - \mathbf{X}^{(i)} = -\left(\frac{\partial G}{\partial \mathbf{X}}\right)^{-1} G(\mathbf{X}^{(i)}), \quad (5.6)$$

und daraus die Lösung zum neuen Zeitpunkt durch

$$\mathbf{X}^{(i+1)} = \mathbf{X}^{(i)} - \left(\frac{\partial G}{\partial \mathbf{X}}\right)^{-1} G(\mathbf{X}^{(i)}). \quad (5.7)$$

Das Newton-Raphson-Verfahren wird, wie alle iterativen Verfahren, so lange durchgeführt, bis eine Abbruchbedingung erreicht wird. Für genauere

Information zu dieser Abbruchbedingung siehe [16].

Die zwei *Knackpunkte* in diesem Verfahren ist die Aufstellung von $\left(\frac{\partial G}{\partial \mathbf{X}}\right)^{-1}$, der Jacobi-Matrix und die Inversion derselben, die durch eine Gauss-Elimination durchgeführt wird.

5.3 Jacobi-Matrix

Die Jacobi-Matrix ist eine der speziellsten Matrizen, die in der Numerik Verwendung finden. In ihr sind als Einträge die Ableitungen aller Gleichungen nach allen Variablen eingetragen, aus diesem Grund ergibt sich eine besondere Struktur der Matrixelemente.

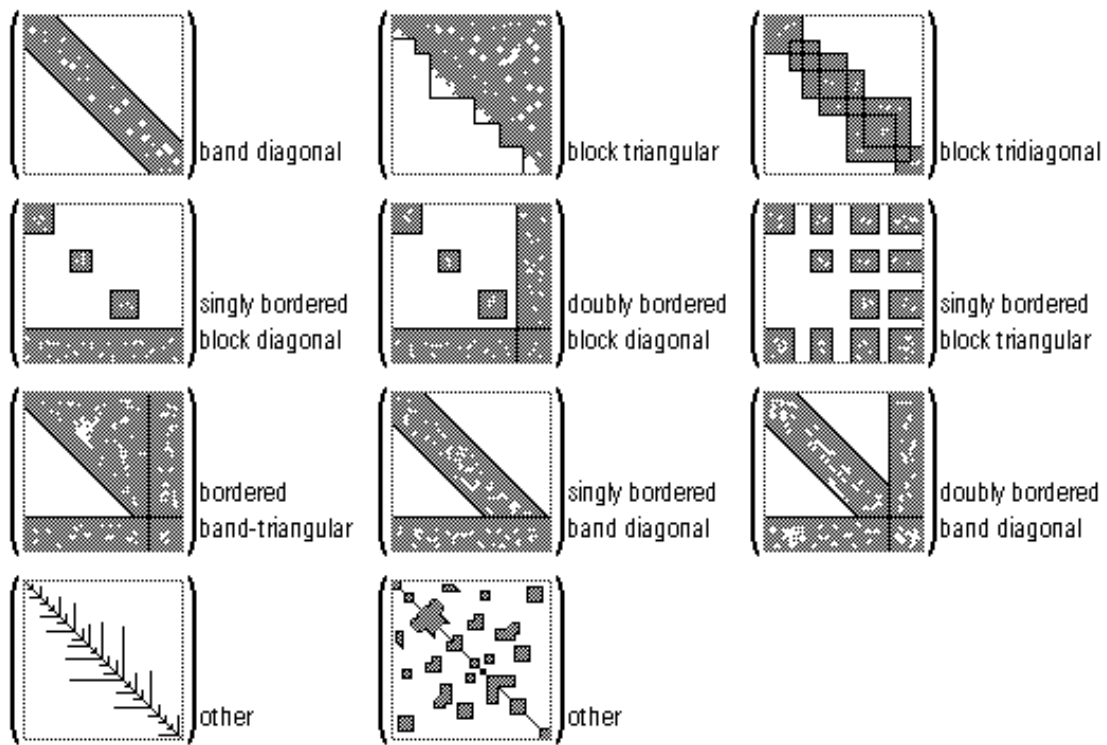
5.3.1 besondere Form der Jacobimatrix

Je nach betrachtetem Problem ist die Struktur der Matrix unterschiedlich (siehe Abb. 5.2). Die besondere Form der Jacobi-Matrix der SHD geht auf die Berechnungen der Ableitungen (beziehungsweise auf das dazu verwendeten Differenzenschema) zurück. Zur Berechnung einer Ableitung am Gitterindex i werden die benachbarten Gitterpunkte $i - 2$, $i - 1$, $i + 1$ und $i + 2$ verwendet, das bedeutet, man braucht bis zu fünf Punkte, um eine korrekte Ableitung berechnen zu können (Verwendung eines Fünf-Punkt-Stempels). Diese Voraussetzung hat selbstverständlich Einfluss auf die Einträge in der Jacobi-Matrix. Die so nun entstehende Struktur in der Matrix entspricht der einer pentadiagonalen Bandmatrix mit maximal fünf Einträgen in jeder Spalte oder Zeile. Das bedeutet, dass man für jeden Gitterindex i je ein Zeile (also N Zeilen) mit fünf $(M \times M)$ -Untermatrizen erhält (siehe Abbildung 5.3). Der Großteil der Einträge in der Jacobi-Matrix sind allesamt Nullen. Auf Grund der großen Anzahl der Gitterpunkte und der vielen Nullen spricht man von **schwach besetzten Matrizen** (sparse matrices).

$$\begin{pmatrix} \ddots & & & & & & \\ & \ddots & & & & & \\ & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ \dots 0 & \frac{\partial G_{m,i}}{\partial x_{m,i-2}} & \frac{\partial G_{m,i}}{\partial x_{m,i-1}} & \frac{\partial G_{m,i}}{\partial x_{m,i}} & \frac{\partial G_{m,i}}{\partial x_{m,i+1}} & \frac{\partial G_{m,i}}{\partial x_{m,i+2}} & 0 \dots \\ & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ & & & & & & \ddots \end{pmatrix} \quad (5.8)$$

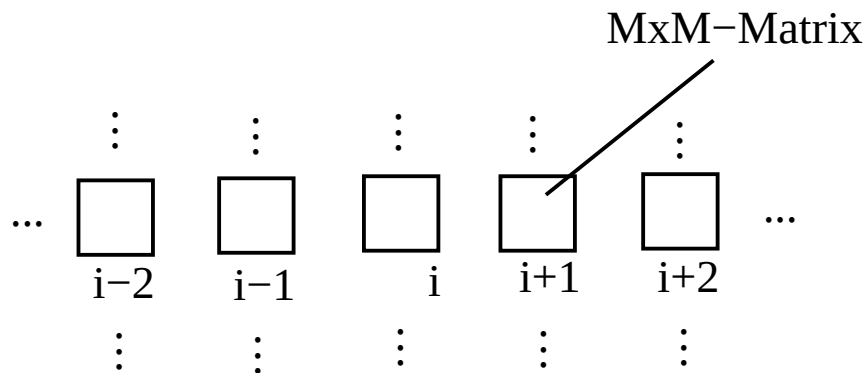
mit einer der fünf Untermatrizen

Abbildung 5.2: besondere Matrizen



Weiß Flächen innerhalb der Matrizen bezeichnen Nullen, grau unterlegte Flächen bezeichnen Einträge ungleich Null, die deutschen Namen können direkt aus den englischen abgeleitet werden (aus [17]).

Abbildung 5.3: Struktur der Jacobi-Matrix



Diese Skizze stellt die spezielle Blockform der Jacobi-Matrix dar. M entspricht der Anzahl der Gleichungen beziehungsweise Unbekannten.

$$\frac{\partial G_{m,i}}{\partial x_{m,i}} = \begin{pmatrix} \frac{\partial G_{1,i}}{\partial x_{1,i}} & \frac{\partial G_{1,i}}{\partial x_{2,i}} & \frac{\partial G_{1,i}}{\partial x_{3,i}} & \cdots \\ \frac{\partial G_{2,i}}{\partial x_{1,i}} & \frac{\partial G_{2,i}}{\partial x_{2,i}} & \frac{\partial G_{2,i}}{\partial x_{3,i}} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = S_{i,i}. \quad (5.9)$$

Mit dieser Abkürzung und Gleichung (5.4)

$$\delta X_i = (X_{1,i}^{(k+1)} - X_{1,i}^{(k)}, X_{2,i}^{(k+1)} - X_{2,i}^{(k)}, \dots \quad (5.10)$$

lässt sich das Gleichungssystem (5.3) als

$$S_{i,i-2}\delta X_{i-2} + S_{i,i-1}\delta X_{i-1} + S_{i,i}\delta X_i + \quad (5.11)$$

$$+ S_{i,i+1}\delta X_{i+1} + S_{i,i+2}\delta X_{i+2} = -G(X_i^{(k)}) \quad (5.12)$$

schreiben, wobei die S -Terme für $(M \times M)$ -Matrizen und die X für M-Vektoren stehen.

Nach dem Einbeziehen der Randwerte, lässt sich die Jacobi-Matrix invertieren und man erhält eine obere Dreiecks-Matrix in der Form

$$\delta X_i = U_i \delta X_{i+1} + V_i \delta X_{i+2} + w_i, \quad (5.13)$$

wo nach dem Vorwärtseinsetzen folgende Ausdrücke auftreten.

$$Y_i = ([S_{i,i-2}U_{i-2} + S_{i,i-1}]U_{i-1} + S_{i,i-2}V_{i-2} + S_{i,i})^{-1} \quad (5.14)$$

$$U_i = Y_i [[S_{i,i-2}U_{i-2} + S_{i,i-1}]V_{i-1} + S_{i,i+1}] \quad (5.15)$$

$$V_i = Y_i S_{i,i+2} \quad (5.16)$$

$$w_i = Y_i [-G(X_i^{(k)}) - [S_{i,i-2}U_{i-2} + S_{i,i-1}]w_{i-1} - S_{i,i-2}w_{i-2}]. \quad (5.17)$$

Die obere Dreiecksform kommt durch die Invertierung der $(M \times M)$ -Matrizen in jeder Zeile zustande. Mit $i = 1$ beginnend, lassen sich die Werte für Y_i , U_i , V_i und w_i berechnen. Dies wird solange durchgeführt bis $i = N$ erreicht ist. Danach wird die Rückeinsetzung durchgeführt und alle Werte $\delta \mathbf{X}$ berechnet um zu der Lösung der physikalischen Unbekannten $\mathbf{X}$ zu gelangen.

Tritt der Fall ein, dass das Lösungsverfahren kein Ergebnis liefert, kann dies aus zwei Gründen passieren. Entweder schlägt das Newton-Raphson-Verfahren fehl, weil es nicht konvergiert, oder die Jacobi-Matrix ist so schlecht konditioniert, dass die Invertierung fehlschlägt.

Kapitel 6

MATHEMATICA-Notebook

Aus dem Bereich der symbolischen Mathematik Softwarepackages ist das Produkt MATHEMATICA¹⁴ ausgewählt worden, um die Berechnung der Ableitung durchzuführen. Die Entscheidung für dieses Produkt erfolgte, auf Grund von Erfahrung in der Arbeit mit dieser Software im Rahmen ähnlicher Berechnungen in der Konvektionstheorie und der leichten und kostengünstigen Verfügbarkeit über den Zentralen Informatikdienst der Universität Wien.

Einführungen in MATHEMATICA, sowie Kurse zum Gebrauch, als auch Bücher zu einzelnen mathematischen Aspekten sind in großer Zahl in Bibliotheken vorhanden, oder aus dem Internet zu beziehen. Verweisen möchte ich hier auf das umfangreiche, alle Details behandelnde Handbuch zu MATHEMATICA (siehe [62]). Brauchbare Einführungen sind unter [4], [26], [27], [43], [44], [55] und [56] gegeben.

Besonders hilfreich war [1], wo der Umgang mit Differentialgleichungen mit MATHEMATICA beschrieben ist, sowie [35] als überschaubare Anleitung zum Programmieren in MATHEMATICA. Eine generelle Einführung in das Programmieren mit symbolischer Mathematik-Software bietet [49].

Das Internet bietet darüberhinaus eine Fülle von Seiten, die vom Vertreiber von MATHEMATICA, **Wolfram**, unterstützt werden, darunter sind die Homepage ([63]), das Forum ([64]), in welchem über aktuelle Probleme diskutiert werden kann, sowie die Dokumentationen ([65], [66] und [67]) zu den unterschiedlichen Versionen von MATHEMATICA.

Neben der Literatur über die Verwendung und den Aufbau von MATHEMATICA, finden sich auch Codes von Programmierern, die den Quellcode von MATHEMATICA so verändert oder erweitert haben, um spezielle mathematische, numerische oder logistische Probleme lösen zu können (siehe [46] und [47]). Als Quelle für die Lösung aktueller Probleme und einzelner Programme dient das Forum der deutschsprachigen MATHEMATICA-User unter [12]. Darüberhinaus gibt es eine große Anzahl von Vorlesungen an Universitäten über das Thema MATHEMATICA,

¹⁴In MATHEMATICA werden die Files Notebooks genannt.

da diese Software vermehrt im technisch-naturwissenschaftlichen Bereich verwendet wird (siehe [32]). Als letztes wäre noch die eingebaute Hilfedatei genannt, die äußerst übersichtlich, mit sehr vielen Beispielen versehen, die Funktionsweise von MATHEMATICA beschreibt.

6.1 Vorgehensweise

Ziel der Verwendung von MATHEMATICA war es, die in der Jacobi-Matrix der diskretisierten Gleichungen auftretenden Ableitungen automatisch zu generieren. Dies sollte so konstruiert sein, dass die Eingabe einfach und unkompliziert gehandhabt werden kann und dass die Ausgabe in einer Form erfolgt, die so wenig wie möglich händischer Bearbeitung bedarf. Die Struktur der Ableitungen sollte so kurz und so leserlich wie möglich sein. Die einzelnen in den Ableitungen vorkommenden Terme sollen so weit wie möglich physikalische Entsprechungen besitzen und dadurch als grundlegende Größen der SHD-Gleichungen identifiziert werden können.

Es werden sogenannte **primäre Variablen** eingeführt, die den Basisvariablen der SHD-Gleichungen entsprechen, das sind im rein hydrodynamischen Fall

- der Radius r_i
- die Masse m_i
- die Dichte $\rho_{i+\frac{1}{2}}$
- die Gasgeschwindigkeit u_i
- die spezifische innere Energie $e_{i+\frac{1}{2}}$

durch die Strahlung kommen noch

- das nullte Moment der Strahlungstransportgleichung ($\hat{=}$ Strahlungsenergiedichte) $J_{i+\frac{1}{2}}$
- das erste Moment der Strahlungstransportgleichung ($\hat{=}$ Strahlungsfluss) H_i

hinzu.

Es werden alle Ableitungen nach jeder der oben genannten Variablen berechnet. Darüberhinaus sind noch **sekundäre Variablen** eingeführt, die physikalischen Entsprechungen besitzen, wie zum Beispiel die in der Diskretisierung verwendeten Volumina V_S und V_V . Diese sind logischerweise aus primären Variablen aufgebaut. Darüberhinaus besteht die Möglichkeit, Variablen noch höherer Klasse zu definieren und in das Notebook einzubauen. Je komplexer die Gleichungen werden, desto ratsamer ist es zusätzliche (Hilfs-)Variablen zu definieren, um die Gleichungen und Ableitungen übersichtlich und leserlich zu halten. Dies ist im

eindimensionalen Fall möglich und auch gewünscht.

Bei Vergrößerung der Anzahl der primären Variablen und Ausbau der Terme (besonders im zweidimensionalen Fall) ist es ratsamer, die Einfachheit der Terme auf Kosten der (physikalischen) Lesbarkeit in Kauf zu nehmen.

Eine weitere Forderung an das Notebook war, dass die Eingabe ohne großen Aufwand erfolgen kann und, was noch wichtiger ist, die Ausgabe der Ableitungen in einer Form erfolgt, dass sie ohne (größere) Nachbearbeitung direkt in den WSHDC eingebaut werden können. Dies geschieht durch den Output als FORTRAN-Subroutinen, die einfach in den Quellcode des WSHDC gelinkt werden.

6.2 Eigenschaften von MATHEMATICA

In diesem Abschnitt wird auf die im Ableitungs-Notebook verwendeten MATHEMATICA-Features eingegangen und die dahinterstehenden Prinzipien erklärt.

Die zwei grundlegenden Notwendigkeiten im Berechnungsvorgang ist die Fähigkeit der Software, zur Erstellung der Ableitungen und das Ersetzen ausgewählte Terme. Diese beiden Fähigkeiten sind Basiseigenschaften von symbolischer Mathematik-Software, also natürlich auch von MATHEMATICA. Diese Fähigkeiten bedürfen allerdings einiger Modifikationen, um den geforderten Ansprüchen genüge zu tun. Neben diesen Eigenschaften verfügt Mathematik die Fähigkeit, generierten Code in FORTRAN-Code umzuwandeln. Da dies aber nur in den FORTRAN77-Standard möglich ist, der WSHDC aber in FORTRAN90/95 programmiert ist, müssen einige Korrekturen angebracht werden, um diesen Output verwenden zu können.

Leider ist es in MATHEMATICA nicht möglich, Variablen in Indexschreibweise zu definieren, aus diesem Grund müssen die Variablen so geschrieben werden, dass ihr Name auf den Index hinweist. Der Radius an der Stützstelle i , r_i wird als `xlmr` geschrieben. Das `x` gibt an, dass es sich um eine primäre Variablen handelt, `l` bezeichnet den Stützstellenindex i (aus sentimentalen Gründen wurde die Notation aus dem WSHDC übernommen), `mr` bezeichnete den Radius (auch diese Art der Bezeichnung stammt aus dem Quellcode). Die konsistente Bezeichnung aus dem WSHDC ($X(L,MR)$) für die Variable Radius an der Stützstelle l , (also i) ließ sich nicht in MATHEMATICA übernehmen, da es Variablenbezeichnungen mit Klammern oder Rechenzeichen ($X(L-1,MR)$) wäre auch nicht erlaubt) nicht zulässt. Die Definition der Variablen ist in Tabelle (6.1) angeschrieben.

¹⁵Die Bezeichnung `xa` signalisiert, dass der Wert der Variablen zum alten Zeitpunkt genommen werden muss. Jede Variable hat natürlich eine Entsprechung zum alten Zeitpunkt und wird in derselben Notation angezeigt.

Tabelle 6.1: Variablenkonvention des MATHEMATICA-Notebooks

Variable	Entsprechung in Indexschreibweise
<code>xlmIImr</code>	r_{i-2}
<code>xlmImr</code>	r_{i-1}
<code>xlmr</code>	r_i
<code>xlpImr</code>	r_{i+1}
<code>xlpIImr</code>	r_{i+2}
<code>xlmIIHmd</code>	$\rho_{i-\frac{3}{2}}$
<code>xlmhmd</code>	$\rho_{i-\frac{1}{2}}$
<code>xlmphmd</code>	$\rho_{i+\frac{1}{2}}$
<code>xlpIIHmd</code>	$\rho_{i+\frac{3}{2}}$
<code>xlpVhmd</code>	$\rho_{i+\frac{5}{2}}$
<code>xlmm</code>	m_i
<code>xlmu</code>	u_i
<code>xlphme</code>	$e_{i+\frac{1}{2}}$
<code>xalmr</code>	$r_i^{(alt)} 15$
<code>dvolph</code>	V_S
<code>dvoll</code>	V_V
<code>vecmd</code>	$\rho_i = \bar{\rho}$
<code>murd12</code>	Advektionsterm
<code>sadsph</code>	Advektionsschalter an Stützstellenindex $i - \frac{1}{2}$
<code>sadv</code>	Advektionsschalter an Stützstellenindex i

6.2.1 Gleichungen

Gleichungen können in MATHEMATICA leicht durch das Zeichen = definiert werden. Links vom Gleichheitszeichen wird die Bezeichnung der Gleichung geschrieben, hier von `G[1]` bis `G[7]`, was den SHD-Gleichungen entspricht. Die Entsprechungen sind in unten stehender Liste angeführt.

Alle nach dem Gleichheitszeichen geschriebenen Variablen werden der Gleichung sozusagen zugeordnet. Eine Besonderheit in MATHEMATICA ist jedoch, dass alle sekundären Variablen, die irgendwo im Code definiert werden, während der Berechnungen durch primäre Variablen ersetzt werden.

Als Beispiel sei hier die Poissongleichung angeführt, die wie folgt angeschrieben wird:

$$G[2] = (xlmm - xlpImm) - xlphmd * dvolph \quad (6.1)$$

Die Gleichung besteht aus den primären Variablen `xlmm`, `xlpImm` und `xlphmd`, sowie aus der sekundären Variablen `dvolph`. Diese sekundäre Variable, die das

Tabelle 6.2: Gleichungsbezeichnung innerhalb des Notebooks

Bezeichnung	SHD-Entsprechung
G[1]	Gittergleichung
G[2]	Poissongleichung
G[3]	Kontinuitätsgleichung
G[4]	Bewegungsgleichung
G[5]	Energiegleichung
G[6]	Strahlungsenergiegleichung
G[7]	Strahlungsflussgleichung

skalare Volumen beschreibt, wird an anderer Stelle durch

$$\text{dvolph} = \frac{4\pi}{3}(\text{x1mr}^3 - \text{xlp1mr}^3) \quad (6.2)$$

definiert. Bei der Ableitung der Poissongleichung wird diese sekundäre Variable durch die primären Variablen ersetzt und nach diesen abgeleitet.

6.2.2 Muster und Blanks

Sogenannte Muster werden in MATHEMATICA mit einem `:=` geschrieben. Im Gegensatz zu normalen Funktionen (mit `=`) stehen solche Ausdrücke für Muster, die wiederholt angewendet werden können. Ausdrücke mit einem einfachen `=` wie

$$\mathbf{f} = 2 * \mathbf{var} \quad (6.3)$$

entsprechen einer Zuordnung, `f` wird einfach mit dem doppelten Wert von `var` belegt.

$$\mathbf{f}[\mathbf{var_}] := 2 * \mathbf{var} \quad (6.4)$$

steht für eine Funktion im mathematischen Sinne. `f` hängt von `var` ab. Verändert man den Wert des unabhängigen Teils `var`, ändert sich der Wert von `f` natürlich auch. Das Zeichen `_` wird Blank genannt und bedeutet, dass der Ausdruck `var_` für jeden beliebigen mathematischen Ausdruck stehen kann und dass es nur ein einzelner Ausdruck sein darf.

6.2.3 Ableitungen

Das Herzstück der gesamten Berechnung ist eine sehr simple MATHEMATICA-Funktion, nämlich `D[]`. `D[G[1],xlmr]` bedeutet, dass die erste Gleichung nach der Variablen `xlmr` abgeleitet wird. Durch mehrere Schleifenkonstruktionen wird nun jede Gleichung nach jeder Variablen abgeleitet und in einer Matrix abgespeichert, der Jacobi-Matrix. Man sieht, dass die eigentliche Hauptaufgabe des Notebooks, das Ableiten, ein recht einfacher und übersichtlicher Vorgang ist.

6.2.4 Ersetzungsregeln

Eine weitere Forderung an das Notebook war, dass nach der Berechnung der Ableitungen, die Möglichkeit gegeben sein sollte, Terme nach Gesichtspunkten der (physikalischen) Lesbarkeit ersetzen zu können. Das bedeutet, dass bestimmte Terme in den Ableitungen gesucht, zusammengefasst und ersetzt werden müssen. Zuvor ist es aber noch notwendig, auf einige Besonderheiten von MATHEMATICA einzugehen:

MATHEMATICA sortiert die einzelnen Variablen einer Gleichung nach dem Alphabet und nach aufsteigender Ordnung. Das heißt die Eingabe

$$a^2 - c^2 + a - d \quad (6.5)$$

wird in

$$a + a^2 - c^2 - d \quad (6.6)$$

umgewandelt. Aus diesem Grund kommt es zu unüblichen Schreibweisen innerhalb der Ausgabe wie zum Beispiel

$$-a + b \quad (6.7)$$

anstatt

$$b - a, \quad (6.8)$$

da nicht auf das Rechenzeichen, sondern nur auf die Variablennamen geachtet wird. Nichtsdestotrotz ist MATHEMATICA in der Lage, den Term `b-a` aus der Gleichung

$$-a + a^2 + b - c^2 - d \quad (6.9)$$

mit `test` zu

$$a^2 - c^2 - d + test \quad (6.10)$$

ersetzen. Dies passiert durch die Ersetzungsregeln, die man wie folgt anschreibt:

$$x /. (b - a) \rightarrow test \quad (6.11)$$

Dies ist die Ersetzungsregel für die obige durchgeführte Substitution. Da es die Möglichkeit gibt sehr viele Substitutionen durchzuführen, sind diese in der Funktion `change[x_]` (siehe Anhang) zusammengefasst.

Ein Problem bei der Ersetzung von Ausdrücken stellt deren Vorzeichen dar. Die Ersetzungsregel

$$x /. (b - a) \rightarrow \text{test} \quad (6.12)$$

führt bei dem Ausdruck

$$a + a^2 - b - c^2 - d \quad (6.13)$$

nicht zu dem Ergebnis

$$a^2 - (b - a) - c^2 - d, \quad (6.14)$$

sondern wird ignoriert. Das liegt an folgender Eigenschaft von MATHEMATICA: Es sind sogenannte **Heads** definiert, sie geben den Kopf des Ausdruckes wieder, sprich den übergeordneten Zusammenhang des Ausdruckes. Der Head von `5` liefert `Integer`, derjenige von `var symbol`, was einer Variablen entspricht. Allerdings liefert der Ausdruck `-var` nicht den Head `-symbol`, sondern `times`. Das bedeutet, dass die übergeordnete Operation der Multiplikation (englisch *times* - mal nehmen) `-1*var` die oberste Einheit für den Head bildet. Dadurch werden Variablen mit negativen Vorzeichen nicht ersetzt.

Um Abhilfe zu schaffen wurde das Muster

$$\text{vor}[\text{rule_}] := \text{Table}[(-1)^i \text{rule}[[1]] \rightarrow (-1)^i \text{rule}[[2]], \{i, 1, 2\}] \quad (6.15)$$

definiert, welches das Vorzeichenproblem löst. Es wird eine kleine Schleife durchlaufen, bei der zuerst die Ersetzung der Variable und dann dessen negativen Gegenstücks versucht wird.

6.2.5 automatische Vereinfachungen

Innerhalb von MATHEMATICA existiert der Befehl `Simplify[]`, der versucht, jeden zugewiesenen Ausdruck auf eine besonders einfache mathematische Form zu bringen. Diese Prozedur kann unter Umständen sehr zeitintensiv sein und zu komplizierteren Ausdrücken als der Ausgangsterm führen. Deshalb ist die Rechenzeit für diesen Befehl in den Steuerparametern begrenzbare. Neben dem beschriebenen Befehl existiert noch jener mit dem Namen `FullSimplify[]`, der weitere, bei `Simplify[]` nicht verwendete mathematische Vereinfachungsregeln unterstützt. Diese sind aber nicht dem Problem der SHD-Gleichungen angepasst und werden deshalb nicht verwendet. Die benötigte Rechenzeit liegt außerdem weit über der des Befehles `Simplify[]`.

Ein weiterer Punkt in der Verwendung des Vereinfachungsbefehls ist die Richtige Platzierung innerhalb des Notebooks. Entweder wird vor der Ersetzung der Hilfsvariablen die Vereinfachung durchgeführt oder danach. Aus Tests hat sich ergeben, dass es zu besseren Ergebnissen kommt, wenn erst **nach** dem Einsatz der Ersetzungsregeln die Vereinfachung der Ausdrücke durchgeführt wird. Dadurch ist gewährleistet, dass ein Höchstmaß an Ersetzungen innerhalb der Ableitungen getätigt wird.

6.2.6 Datenausgabe

Die Datenausgabe ist so strukturiert, dass eine größtmögliche Anpassung an den WSHDC gegeben ist, der in FORTRAN90/95 programmiert ist. Zu diesem Zweck ist das Muster `fortconst[x_]` definiert worden, das den MATHEMATICA-Output zuerst mittels der Funktion `FortranForm[]` in FORTRAN77-Code umwandelt und danach mittels `ToString[]` in ASCII-Text. Dadurch ist eine einfache spätere Manipulation durch String-Operationen möglich. Neben der Ausgabe der Ableitungen in verwertbarer Form, wird zusätzlich eine FORTRAN-File-Umgebung erzeugt, die der einer Subroutine entspricht. Das Notebook verwendet die Notation des WSHDCs, um die Ableitungen darzustellen. $G(MR, L)$ entspricht der Gittergleichung am Stützstellenindex i , $DG(MR, MR, IP, L)$ definiert die Ableitung der Gittergleichung nach der Variablen r am Stützstellenindex $i + 1$. Für eine genaue Beschreibung der WSHDC-Notation siehe Kommentare innerhalb des Notebooks oder [41].

Neben der Gleichung und aller Ableitungen **ungleich Null** (die Ausgabe der Ableitungen die Null ergeben wird unterdrückt, um Zeit und Speicherplatz zu sparen, im Code werden alle Einträge in der Jacobi-Matrix zuerst mit Null belegt und nur alle Einträge ungleich Null werden überschrieben), werden zusätzlich alle notwendigen FORTRAN-Bezeichnungen eingefügt, die notwendig sind, damit das File als Subroutine verwendet werden kann.

Bei der Verwendung des Musters `fortconst[x_]` wird gleichzeitig die Variablennotation innerhalb des MATHEMATICA-Notebooks (`xlmr`) in die FORTRAN-Notation ($X(L, MR)$) umgewandelt. Dies geschieht mit dem MATHEMATICA-Modul `SHDNom.m`, in dem alle Transformationen gespeichert sind (Genaueres dazu später).

6.2.7 Nachbearbeitung

Trotz aller Manipulationen innerhalb des Notebooks müssen die erzeugten Files noch einer Nachbearbeitungen unterzogen werden. Zwei im Output vorhandenen Zeichen können von FORTRAN-Compilern nicht interpretiert werden. Da die Texte als String ausgegeben werden, sind sie durch ein spezielles Maskierungszeichen begrenzt, welches in MATHEMATICA das Anführungszeichen (")

ist. Diese Zeichen müssen aus dem Text gelöscht werden. Das zweite Zeichen ist das Zeilenfortführungszeichen, das in MATHEMATICA der verkehrte Schrägstrich (den meisten als Backslash bekannt) ist, dasjenige in FORTRAN jedoch das Kaufmannsund (&). Die Ersetzungen werden nach Durchlaufen des Notebooks mittels eines Shell-Skripts durchgeführt.

6.3 Struktur

Notebooks sind in Zellen unterteilt, die durch Klammern am rechten Rand gekennzeichnet sind. Diese Zellen bilden Untereinheiten und können entweder editierbar oder gesperrt sein. Vorzugsweise sind sie so gestaltet, dass Teile, die der User unter keinen Umständen verändern soll, gesperrt sind. Im Fall des hier vorgestellten Notebooks sind nur die Zellen editierbar, in denen die Gleichungen oder Hilfsvariablen eingegeben werden sollen. Folgende Farbkonventionen gelten innerhalb des Notebooks:

- editierbare Zellen - grau unterlegt und schwarz gerahmt
- nicht editierbare Zellen - weiß mit schwarzem Text
- blauer Text - Kommentar
- gelb unterlegter Text - Anweisungen zur Eingabe

Am Anfang des Notebooks steht der zeitliche Verlauf der Entwicklung und erste Kommentare zur Verwendung. Danach können die Steuerparameter eingegeben werden, die im nächsten Abschnitt erklärt sind. Darauf erfolgt die Überprüfung der Eingaben, die Bestätigung wird innerhalb des Notebooks ausgegeben, um bei einer Fehleingabe die Berechnung unterbrechen zu können, und das Logfile im Ausgabeverzeichnis erstellt.

Darunter sind alle primären Variablen innerhalb einer Liste definiert, die für das ausgewählte Problem kennzeichnend sind (5 Variablen im Hydrodynamikfall, 7 bei gesamter SHD). Nach dieser Definition kann der User die diskretisierten Gleichungen in das Notebook eingeben. Für jede Gleichung der SHD ist eine eigene Zelle reserviert, die nach der Konvention laut Tabelle (6.2) bearbeitet werden muss.

Nach der ersten Eingabe werden jene Variablen- und Filebezeichner bestimmt, um eine konsistente Ausgabe mit dem WSHDC zu erreichen. Desweiteren werden die Namen für die Ausgabefiles belegt, die denen von FORTRAN-Subroutinen entsprechen. Danach wird die Vorzeichenwechselfunktion festgelegt, die oben definiert wurde.

Innerhalb der nächsten editierbaren Zelle wird jenes Muster definiert, das einen Ausdruck in verwertbaren FORTRAN-Code umwandelt. In jedem Ausdruck werden auftretende Brüche durch eine entsprechende Buchstabenkombination

ersetzt. Diese Vorgehensweise stammt noch aus grauer Urzeit, als konstante Werte nur ein Mal berechnet werden sollten. Aus diesem Grund werden diese Werte in den Gleichungen durch Variablen ersetzt, um bei der Berechnung auf einen in einem Hilfsvariablen-File berechneten Wert zugreifen zu können. Dies sparte wertvolle Rechenzeit. Dieser Schritt ist durch die rasche Entwicklung in der Computerbranche nicht mehr notwendige, aber aus Konsistenzgründen innerhalb des WSHDC trotzdem durchgeführt. Nach diesen Ersetzungen erzeugt die Funktion `N[]` eine numerische Form des Ausdruckes. `N[4]` liefert als Ergebnis 4., was unter FORTRAN einer Real-Zahl entspricht. Dies ist notwendig um die Variablendefinition innerhalb des WSHDCs einzuhalten. Erst dann erfolgt die Umwandlung in FORTRAN-Code und danach die Ausgabe als ASCII-String.

Folgendes Beispiel soll das Vorgehen illustrieren:

$$4x_{lmr}^2 - 1 \tag{6.16}$$

wird in den numerischen Wert

$$4.x_{lmr}^2 - 1. \tag{6.17}$$

umgewandelt, danach in FORTRAN-Code

$$4. * x_{lmr} * *2 - 1. \tag{6.18}$$

und zum Schluss als Text ausgegeben

$$"4. * x_{lmr} * *2 - 1." \tag{6.19}$$

In der nächsten Zelle erfolgt die Festlegung des Ersetzungsmusters. Hier muss allerdings ein kleiner Trick angewendet werden. Da das Notebook in die Ausgabefiles jeweils die Gleichung und die zugehörigen Ableitungen schreibt, kommt es zu einem Konflikt bei der Ersetzung von sekundären oder höhergeordneten Variablen. Diese sollen in den Gleichungen selbst nicht ersetzt, für die Ableitungen jedoch müssen sie sehrwohl substituiert werden, da ansonsten eine Differentiation nicht durchführbar ist. Darum die Ersetzung von sekundären und höherrangigen Variablen. Bei der Ausgabe der Gleichungen bleiben nun alle Variablen ungeändert stehen, für die Ableitungen werden aber nur Zusammensetzungen der Primärvariablen verwendet.

Im Anschluss erfolgt die Festlegung der File-Header der FORTRAN-Files und die Ausgabe der Gleichungen. In der nächsten editierbaren Zellen sind alle sekundären und andere höherrangigen Variablen eingetragen, die notwendig sind, um die Gleichungen formulieren zu können. Es ist darauf zu achten, dass wirklich jede Variable in dieser Zelle eingetragen wird, da es sonst nicht möglich ist, korrekte Ableitungen zu erhalten. In der folgenden Zelle wird das File der Hilfsvektoren definiert und alle Hilfsvariablen wie die Differenzen von kubischen

Ausdrücken hineingeschreiben, da es genauer ist, nur die Radiusdifferenz und nicht die Differenz der Volumina zu berechnen.

$$a^3 - b^3 \tag{6.20}$$

wird durch

$$(a - b)(a^2 + ab + b^2) \tag{6.21}$$

ersetzt werden.

Nach dem Fertigstellen des Hilfsvektorenfiles erfolgt die Erstellung der Jacobi-Matrix, die intern abgespeichert wird. Dieser Vorgang dauert je nach Umfang der Gleichungen zwischen einigen Sekunden und einigen Minuten, im zweidimensionalen Fall können Rechenzeiten bis zu 20 Minuten auftreten. Der vorletzte Punkt innerhalb des Notebooks ist noch die Ausgabe der Ableitungen in der Form, die konsistent mit dem WSHDC ist. Abschluss der Rechnung bildet das Löschen bestehender Zuordnung der sekundären Variablen, damit beim erneuten Durchlauf des Notebooks alte Verbindungen zu keinen Konflikten mit Neuerungen führen können.

6.4 Steuerparameter

In diesem Abschnitt sind die Steuerparameter des Notebooks in einer Liste zusammengefasst. Es besteht die Möglichkeit das Notebook unter verschiedenen Betriebssystemen laufen zu lassen (DOS oder Unix/Linux). Erfreulich ist die Tatsache, dass sich die Files (Notebook wie Output-Files) ohne Probleme zwischen Rechnern unterschiedlichen Betriebssystems transportieren lassen.

Das Setzen eines Arbeitsverzeichnisses ist unbedingt notwendig, da MATHEMATICA-Files ansonsten in den Tiefen der Directory-Struktur abspeichert. Bevor Rechnungen begonnen werden, ist es ratsam alle benötigten Module (eine Art Hilfsprogramm) zu laden, damit darauf auf keinen Fall vergessen wird.

Wichtig ist die Angabe der Zeilenlänge des Outputs. Es hängt nämlich vom Compiler ab, wie lang die Zeilen innerhalb des FORTRAN-Codes sein dürfen. FORTRAN77 beschränkt die Zeilenlänge auf 72 Zeichen, was von dem Befehl `FortranForm[]` eingehalten wird. Da neuere Compiler aber mehr Zeichen innerhalb einer Zeile verarbeiten können, modifiziert die Zeilenlängenangabe den Output auf die gewünschte Länge.

MATHEMATICA arbeitet in einer Hinsicht besonders benutzerfreundlich: Wird bei der Eingabe ein Fehler gemacht, erkennt das Programm den Fehler und gibt entsprechende Fehlermeldungen aus, die angeben, wie der Fehler zu beheben ist. Allerdings werden auch bei Namensgleichheiten Warnings ausgegeben, die

Tabelle 6.3: Steuerparameter des MATHEMATICA-Notebooks

Paramter	Zweck
SetDirectory	Setzen des Arbeitsverzeichnisses
<< X'	Lades des Moduls X
dos	Schalter für Betriebssystem 0 = UNIX/LINUX-Betriebssystem 1 = MS-DOS-Betriebssystem
modell	Modellnummer
pagewidth	Zeilenlänge
aus	Schalter für Ausgabe der Warnings 0 = keine Ausgabe 1 = Ausgabe
loeschen	Schalter für Löschung bestehender Files 0 = keine Löschung 1 = Löschung
logfile	Schalter für Erstellung eines Logfiles 0 = keine Erstellung 1 = Erstellung
logfile	Schalter für Verwendung von Winkelfunktion von Simplify[] 0 = keine Verwendung 1 = Verwendung
zeit	Angabe der der Rechenzeit für den Befehl Simplify[] in Sekunden
ng	Anzahl der Gleichungen
logtxt	Logfile Text

eben auf diesen Umstand hinweisen. Da die Variablennamen große Ähnlichkeiten aufweisen, gibt es unzählige Warnungen. Diese können ausgeschaltet werden (allerdings sollte man das nur, wenn man sich seiner Programmierung sehr sicher ist).

Weiters kann angegeben werden, ob man das Löschen bestehender Files oder die Erstellung eines Logfiles wünscht.

Für die Funktion `Simplify[]` existieren zwei Schalter, die dazu dienen, die Schnelligkeit der Vereinfachungen zu erhöhen. Erstens kann die Verwendung von Additionstheoremen bei der Vereinfachung unterdrückt werden, da in den SHD-Gleichungen im Moment keine Winkelfunktionen vorkommen. Als zweites kann die Vereinfachungszeit für jeden Term beschränkt werden. Ist nach einer bestimmten Zeit keine Vereinfachung gefunden worden, wird der Vorgang abgebrochen und der Term unverändert stehen gelassen.

Als letztes muss noch die Anzahl der Gleichungen und damit die der primären Variablen definiert werden.

6.5 Gleichungen in MATHEMATICA-Notation

Nach all den getätigten Definitionen sei hier jeweils ein Beispiel für eine vektorielle und skalare Gleichung in MATHEMATICA-Notation gegeben.

6.5.1 Anschreiben einer vektoriellen Gleichung in MATHEMATICA - Bewegungsgleichung

Hier sei als Beispiel für eine vektorielle Gleichung die Bewegungsgleichung genommen und gezeigt, wie der Übergang von der diskretisierten Form (siehe auch Seite 49) in die MATHEMATICA-Schreibweise aussieht. Man erkennt, dass die Volumensterme als sekundäre Variablen definiert sind und die Advektionsterme genau der Diskretisierung auf Seite 45 entsprechen. Anzumerken ist, dass im Moment nur die reine Hydrodynamik eingebaut ist. Die Gleichung ist in ihre Terme zerlegt und diese mit Großbuchstaben beschriftet. Die Umwandlung in MATHEMATICA-Notation ist Termweise angeschrieben.

$$G[4] = \underbrace{\delta(u \bar{\rho} V_V)}_A + \underbrace{4\pi \Delta((\widetilde{\rho u}) \overline{r^2 u^{rel}} \delta t)}_B + \underbrace{4\pi r^2 (\Delta P) \delta t}_C + \underbrace{\frac{Gm}{r^2} \bar{\rho} V_V \delta t}_D$$

$$\begin{aligned} \xrightarrow{\text{Mathematica-Notation}} A &= (\text{x l p h m d} * \text{d v o l p h} - \text{x a l p h m d} * \text{d v o l a p h}) \\ B &= 4 * \pi * (\text{m u r 2 d l} * (\text{x l p h m d} * \text{s a d s p h} + (1 - \text{s a d s p h}) * \text{x l m h m d}) - \\ &+ \text{m u r 2 d l p I} * (\text{s a d s p I I I h} * \text{x l p I I I h m d} + (1 - \text{s a d s p I I I h}) * \text{x l p h m d}) \\ C &= 4 * \pi * \text{x l m r}^2 * \text{t s t} * (\text{p r g z m h} - \text{p r g z p h}) \\ D &= \frac{G * \text{x l m m}}{\text{x l m r}^2} * \text{v e c m d} * \text{d v o l l} * \text{t s t} \end{aligned}$$

6.5.2 Anschreiben einer skalaren Gleichung in MATHEMATICA - Energiegleichung

Die Energiegleichung dient wieder als Beispiel für eine skalare Gleichung (wieder nur der rein hydrodynamische Fall). Die Transformation in Notebook-Schreibweise ist wieder für jeden Term angegeben.

$$G[5] = \underbrace{\delta(\rho e V_S)}_A + \underbrace{4\pi \Delta((\widetilde{\rho e}) r^2 u_{rel} \delta t)}_B + \underbrace{4\pi P \Delta(r^2 u) \delta t}_C$$

$$\begin{aligned} \xrightarrow{\text{Mathematica-Notation}} A &= (\text{x l p h m d} * \text{x l p h m e} * \text{d v o l p h} - \\ &\quad \text{x a l p h m d} * \text{x a l p h m e} * \text{d v o l a p h}) \\ B &= \text{m u r 2 d l} * (\text{x l p h m d} * \text{x l p h m e} * \text{s a d s p h} + (1 - \text{s a d s p h}) * \\ &\quad * \text{x l m h m d} * \text{x l m h m e}) - \text{m u r 2 d l p I} * \\ &\quad (\text{s a d s p I I I h} * \text{x l p I I I h m d} * \text{x l p I I I h m e} + \\ &\quad + (1 - \text{s a d s p I I I h}) * \text{x l p h m d} * \text{x l p h m e}) \\ C &= 4 * \pi * \text{p r g z p h} * \text{t s t} * (\text{x l m r}^2 * \text{x l m u} - \text{x l p I m r}^2 * \text{x l p I m u}) \end{aligned}$$

6.6 Ein Beispiel - die Poissongleichung

Die diskretisierte Poisson-Gleichung schreibt sich wie folgt:

$$\Delta m = \rho V_S, \quad (6.22)$$

daraus wird in MATHEMATICA-Schreibweise

$$G[2] = (\text{x l m m} - \text{x l p I m m}) - \text{x l p h m d} * \text{d v o l p h}. \quad (6.23)$$

Die sekundäre Variable `dvolph` wird durch

$$\frac{4\pi}{3}(\text{xlmr}^3 - \text{xlpImr}^3) \quad (6.24)$$

ersetzt und man gelangt zu:

$$\text{G}[2] = (\text{xlmm} - \text{xlpImm}) - \text{xlpHmd} * \frac{4\pi}{3}(\text{xlmr}^3 - \text{xlpImr}^3). \quad (6.25)$$

Nun wird nach allen primären Variablen abgeleitet. Beiträge liefern nur Ableitungen nach m_i , m_{i+1} , $\rho_{i+\frac{1}{2}}$, r_i und r_{i+1} , alle anderen Ableitungen sind gleich Null.

Folgende Ableitungen sind ungleich Null:

$$\begin{aligned} \frac{\partial \text{G}[2]}{\partial \text{xlmm}} &= 1 \\ \frac{\partial \text{G}[2]}{\partial \text{xlpImm}} &= -1 \\ \frac{\partial \text{G}[2]}{\partial \text{xlpHmd}} &= -\frac{4\pi}{3} * (\text{xlmr}^3 - \text{xlpImr}^3) \\ \frac{\partial \text{G}[2]}{\partial \text{xlmr}} &= 4 * \pi * \text{xlpHmd} * \text{xlmr}^2 \\ \frac{\partial \text{G}[2]}{\partial \text{xlpImr}} &= -4 * \pi * \text{xlpHmd} * \text{xlpImr}^2 \end{aligned}$$

danach erfolgt die Ersetzung durch sekundäre Variablen, was nur den dritten Eintrag verändert:

$$\frac{\partial \text{G}[2]}{\partial \text{xlpHmd}} = -\text{dvolscA}$$

Dann wird das Muster `fortconst[]` angewendet und man erhält

$$\begin{aligned} \frac{\partial \text{G}[2]}{\partial \text{xlmm}} &= "1." \\ \frac{\partial \text{G}[2]}{\partial \text{xlpImm}} &= "-1." \\ \frac{\partial \text{G}[2]}{\partial \text{xlpHmd}} &= "-\text{dvolscA}" \\ \frac{\partial \text{G}[2]}{\partial \text{xlmr}} &= "4 * \pi * \text{xlmd} * \text{xlpImr} * 2" \\ \frac{\partial \text{G}[2]}{\partial \text{xlpImr}} &= "-4 * \pi * \text{xlmr} * 2 * \text{xlmd}" \end{aligned}$$

6.7 Modul ShdNom.m

Im Modul *ShdNom.m* sind alle Ersetzungsregeln für die FORTRAN-Schreibweise des WSHDCs gespeichert. Nachdem alle Output-Terme in Strings umgewandelt wurden, werden alle Variablen von MATHEMATICA-Notation wieder in FORTRAN-Standard übersetzt. Darüberhinaus werden die Buchstabenkürzel für die Brüche wieder in Brüche umgewandelt und redundanter FORTRAN-Code, wie die Terme **1* herausgelöscht. Bei der Ersetzung der Variablennamen ist darauf zu achten, dass zuerst die langen Namen und danach erst die kurzen ersetzt werden, also zuerst $\mathbf{xlpImr} \rightarrow X(L+1,MR)$ und dann erst $\mathbf{xlmr} \rightarrow X(L,MR)$.

Schlussendlich erhält man für die Ableitungen folgende Ausdrücke:

$$\begin{aligned}\frac{\partial G[2]}{\partial \mathbf{x1mm}} &= \text{"1."} \\ \frac{\partial G[2]}{\partial \mathbf{xlpImm}} &= \text{" - 1."} \\ \frac{\partial G[2]}{\partial \mathbf{x1phmd}} &= \text{" - (DVOLSCA(L))"} \\ \frac{\partial G[2]}{\partial \mathbf{xlmr}} &= \text{"4 * PI * X(L, MD) * X(L + 1, MR) * *2"} \\ \frac{\partial G[2]}{\partial \mathbf{xlpImr}} &= \text{" - 4 * PI * X(L, MR) * *2 * X(L, MD)"}.\end{aligned}$$

Das Modul ist auf Seite 111 im Anhang beigelegt.

6.8 Reinigen

Der letzte Schritt bei der Erstellung der Ableitungen ist das *Reinigen* der Ausgabefiles, das heißt, dass die Anführungszeichen (") gelöscht und das Zeilenumbruchzeichen ersetzt wird. Für die Ableitungen der Poissongleichungen

erhält man:

$$\frac{\partial G[2]}{\partial x_{lmm}} = 1.$$

$$\frac{\partial G[2]}{\partial x_{lpImm}} = -1.$$

$$\frac{\partial G[2]}{\partial x_{lphmd}} = -(DVOLSCA(L))$$

$$\frac{\partial G[2]}{\partial x_{lmr}} = 4 * PI * X(L, MD) * X(L + 1, MR) ** 2$$

$$\frac{\partial G[2]}{\partial x_{lpImr}} = -4 * PI * X(L, MR) ** 2 * X(L, MD).$$

Das Notebook erzeugt für jede Gleichung ein FORTRAN-File, eine File der Hilfsvektoren und ein Logfile.

Für den rein hydrodynamischen Fall sind alle Output-Files im Anhang beigefügt. Desweiteren findet sich ein Anhang über **alle** Befehle inklusive Erklärungen, die im Notebook verwendet werden. Zusätzlich befindet sich im Anhang das gesamte entwickelte Notebook.

Kapitel 7

Ausblick

Alle von MATHEMATICA generierten FORTRAN-Files sind mit dem bestehenden SHD-Code verbunden und einfache Testrechnungen durchgeführt. Bestehende Ableitungen werden mit denen durch MATHEMATICA generierten verglichen, und ergeben exakt dasselbe Ergebnis. Rechnungen für den rein hydrodynamischen Fall liefern die erwarteten Resultate. Um das Zusammenspiel des bereits vorhandenen Codes mit dem neuen, besonders die Verwendung der neuen und die Kombination mit den alten Hilfsvektoren, zu testen, erfolgt der Einbau der Files nach der Reinigung von Maskierungszeichen und Austausch von Zeilenfortführungszeichen direkt in den SHD-Code.

Alle durchgeführten Veränderung des SHD-Codes, sowie der genaue Implementierungsvorgang sind in [41] beschrieben.

7.1 Ausbau und Erweiterungen

Das Ziel des erarbeiteten MATHEMATICA-Notebooks ist, alle Änderungen des SHD-Codes in numerischer und physikalischer Hinsicht, die Ableitungen betreffend, automatisieren zu können. Dadurch wird die Effizienz und Schnelligkeit der Berechnung von Ableitungen drastisch erhöht. Durch diese Zeitersparnis, sowie durch das Ausschließen von Ableitungsfehlern durch die Verwendung eines getesteten algebraischen Computersystems, ist die Möglichkeit gegeben, einfach und rasch unterschiedliche Gleichungsdiskretisierungen zu generieren und zu testen. Neben der kontinuierlichen Erweiterung des SHD-Codes durch physikalische Aspekte, wie

- Wechselwirkung mit Magnetfeldern
- Rotationseffekte
- Anpassung an neue Geometrien und astrophysikalische Problemstellungen

- Staubterme (hier sind fünf zusätzliche Gleichungen von Nöten, welche die zeitabhängige Entwicklung der Staubpartikel beschreiben)
- Erweiterung auf frequenzabhängigen Strahlungstransport und Opazitäten
- Erweiterung der SHD-Gleichungen auf ein Multi-fluid, um ein chemisches oder nukleares Netzwerk mitrechnen zu können
- Einbau von Gleichungen zur Beschreibung der Konvektion
- generelle Erweiterung des Codes auf weitere astrophysikalische Anwendungen, bei denen Strahlung und Materie wechselwirken

ist die Verbesserung und die Erhöhung der Genauigkeit der Numerik der zweite Teil, der sich auf Veränderungen der SHD-Gleichungen auswirkt.

Folgende Anpassungen der Numerik sollen mit Hilfe der von MATHEMATICA generierten Ableitungen untersucht werden:

- Erweiterung des in [14] vorgeschlagenen Diskretisierungsschemas auf die Kopplungsterme in den Materiegleichungen und die Strahlungsgleichungen, um Vergleichsrechnungen in einer astrophysikalischen Umgebung durchführen zu können. Diese Ergebnisse sind mit Rechnungen in der alten Diskretisierung zu vergleichen und es ist zu bestimmen, ob die neue Diskretisierung Resultate liefert, die bisher in divergenten Bereichen liegen und, ob sie genauer als die bisherigen sind.
- Untersuchung, ob sich der Übergang von zeitzentrierten Variablen auf zeit-zentrierte Operatoren auf die Ergebnisse auswirkt. Die Resultate sollen mithelfen zu entscheiden, welche Art der zeitlichen Zentrierung für welches astrophysikalisches System ideal geeignet ist.
- Da die Advektionsterme in den Gleichungen jene sind, die in besonderem Maße kompliziert wie wichtig sind, können mit MATHEMATICA unterschiedliche Advektionsschemata genauer untersucht werden. Die Terme solcher Schemata höherer Ordnung besitzen sehr umfangreiche und komplexe Ableitungen, die mit Hilfe des Notebooks leicht für unterschiedlichste Ausgangskombinationen berechnet werden können.
- Generelle Untersuchung von grundlegender Numerik, wie unterschiedliche Diskretisierungsvorschriften und Erweiterung jener Teile des WSHDC bei denen in Hinblick auf die zu große Komplexität der Ableitungen eine vereinfachte Formulierung gewählt wurde.

7.2 Zweidimensionale SHD

Aufbauend auf einer früheren Version des hier vorgestellten Notebooks wurde MATHEMATICA auch für die Erstellung der Ableitungen (siehe Abbildung 7.1) bei der Programmierung eines neuen zweidimensionalen SHD-Codes (siehe [48]) verwendet.

Auf Grund der höheren Anzahl von Variablen und der logischerweise auftretenden größeren Menge von Gitterpunkten ist hier mit einem viel umfangreicheren Output zu rechnen. Deshalb wurde für jede Gleichung aus dem SHD-System ein eigenes Notebook erstellt, das die Ableitungen nun separat für jede Gleichung berechnet, um den Überblick über die Ableitungen besser behalten zu können. Darüberhinaus sind die einzelnen Terme der Ableitungen um einiges umfangreicher als im eindimensionalen Fall. Dadurch muss verstärkt Augenmerk auf die Ersetzungsregeln gelegt, und weitere Hierarchien von Hilfsvariablen eingeführt werden.

7.3 Weiterführung des Notebooks

Das hier präsentierte MATHEMATICA-Notebook soll in keiner Hinsicht einen Abschluss im Bereich Erstellung der Ableitungen darstellen. Weitere Adaptionen, Erweiterungen und Verbesserungen sind bereits in Angriff genommen worden.

Das Handling des Notebooks bedarf noch einer strafferen Strukturierung, um die Eingabe der Hilfsvektoren und Ersetzungsregeln noch weiter zu vereinfachen.

Durch die Umstrukturierung des WSHDCs, die auf Grund der Einführung der neuen Art der Diskretisierung notwendig wurde, steht das Notebook in Zentrum des Verfahrens zur Gewinnung der Ableitungen. Es soll so in den Ablauf des WSHDCs eingebaut werden, dass es autonom für jede Art des Problems (Geometrie, Anfangswerte und -bedingungen) die benötigten Ableitungen liefert, die automatisch gelinkt werden können. Dazu ist es notwendig, die FORTRAN-Routinen des Quellcodes umzuschreiben und das Notebook noch enger an diese zu binden.

Einen weiteren extrem sensiblen Punkt in den Berechnungen bilden die Randbedingungen der SHD-Gleichungen. Die Berechnung dieser Bedingungen soll vom Quellcode in das Notebook übernommen werden und dort ebenfalls automatisch, nach Auswahl über Schalter, berechnet werden.

Zur Realisierung all dieser Punkte, ist es notwendig, den Input in das und den Output aus dem Notebook zu verbessern und zu vereinfachen.

Abbildung 7.1: MATHEMATICA-Notebook zur Berechnung der Ableitungen der Gittergleichung in η -Richtung im zweidimensionalen Fall

Ableitungen der 2D Gittergleichungen (eta-Richtung): Zentralbereich

author: stoekl
date: 02 09 03

Setzen des Rechenverzeichnis - Setdirectory
Einstellen der Anzahl der Zeichen pro Zeile (Pagewidth) - pagewidth
Ausschalten der Warnings und Fehlermeldungen - off: 0:aus, 1:ein
Loeschen der bestehenden Files im Arbeitsverzeichnis - loeschen: 0:aus, 1:ein
Anlegen eines Logfiles - logfile: 0:aus, 1:ein

```
SetDirectory["/velo/stoekl/2D-SHD/mathematica/data"];

pagewidth = 130;
off = 1;
loeschen = 1;
```

Abfrage der Parameterwerte und Kontrollausgabe

```
Print["Off wurde auf ", off, " gesetzt."];
If[off == 1, Off[]]
Print["Loeschen wurde auf ", loeschen, " gesetzt."]
If [loeschen == 1, Run["rm DG_grid_eta.f90"]];

Off wurde auf 1 gesetzt.

Loeschen wurde auf 1 gesetzt.
```

Liste der internen Variablen (#=nv) (Punkte #=np), nach denen abgeleitet wird.

```
nv = 4;
np = 13;
dev = {Xij, Ximlj, Xim2j, Xiplj, Xip2j, Xijm1, Xijm2, Xijp1, Xijp2,
Ximljm1, Ximljp1, Xipljm1, Xipljp1, Yij, Yimlj, Yim2j, Yiplj, Yip2j,
Yijm1, Yijm2, Yijp1, Yijp2, Yimljm1, Yimljp1, Yipljm1, Yipljp1,
Eij, Eimlj, Eim2j, Eiplj, Eip2j, Eijm1, Eijm2, Eijp1,
Eijp2, Eimljm1, Eimljp1, Eipljm1, Eipljp1,
rhoij, rhoimlj, rhoim2j, rhoiplj, rhoip2j, rhoijm1, rhoijm2,
rhoijp1, rhoijp2, rhoimljm1, rhoimljp1, rhoipljm1, rhoipljp1};
```

Hilfsvariablen:

$$\begin{aligned} \text{XnetahatN} = & \frac{1 + 2 \text{betas} + 2 \text{betap}}{\sqrt{(Xijp1 - Xij)^2 + (Yijp1 - Yij)^2}} - \\ & \frac{\text{betas}}{\sqrt{(Xijp2 - Xijp1)^2 + (Yijp2 - Yijp1)^2}} - \frac{\text{betas}}{\sqrt{(Xij - Xijm1)^2 + (Yij - Yijm1)^2}} - \\ & \frac{\text{betap}}{\sqrt{(Xip1jp1 - Xip1j)^2 + (Yip1jp1 - Yip1j)^2}} - \frac{\text{betap}}{\sqrt{(Xim1jp1 - Xim1j)^2 + (Yim1jp1 - Yim1j)^2}}; \end{aligned}$$

Dieses Bild zeigt die erste Seite eines MATHEMATICA-Notebooks, das für die Berechnung der Ableitungen der Gittergleichung in einer Raumrichtung des neuen 2d-SHD-Codes verwendet wird, für weitere Informationen siehe [48].

Beschreibung der Anhänge

Folgende Anhänge sind beigelegt:

- **Anhang A:** Stichwortliste SHD-relevanter Begriffe mit kurzer Erklärung und Literaturverweisen, erstellt auf Grund der oft langwierigen Suche nach bestimmten Einzelheiten im riesigen Konvolut erzeugten Papiers.
- **Anhang B:** Liste **aller** im MATHEMATICA-Notebook verwendeten Befehle und Erklärungen, um eine möglichst benutzerfreundliche Verwendung desselben garantieren zu können.
- **Anhang C:** Printout der aktuellsten Version des MATHEMATICA-Notebook.
- **Anhang D:** Printout des für die Berechnungen nötigen Moduls.
- **Anhang E:** Printout der generierten FORTRAN-Files der aktuellsten Version des MATHEMATICA-Notebooks.
- **Anhang F:** Kurzzusammenfassung der diskretisierten SHD-Gleichungen, kompakt auf eine Seite gebracht.
- **Anhang G:** Als Anhänger der SI-Einheiten hier die Umrechnungsfaktoren, in das in der Astronomie noch oft gebräuchliche cgs-System, sowie eine Auflistung aller im SHD-Code verwendeten Konstanten mit ihren Werten, relativen Fehlern und Dimensionen.

Diese Anhänge sollen als Nachschlagewerk dienen und der Orientierung in der Welt der SHD und des WSHDC hilfreich sein.

Anhang A

Index

strahlungshydrodynamischer und numerischer Begriffe

In diesem Anhang sind die wichtigsten Begriffe der SHD und der Mathematik beziehungsweise Numerik aufgelistet. Dies ist geschehen, um einen Überblick für Neueinsteiger zu schaffen, der helfen soll, sich schneller in die Materie und den Aufbau des Wiener SHD-Codes einlesen zu können. Für genauere Informationen über den Ablauf und Aufbau siehe [41].

Die Einträge in diesem Index sind wie folgt gestaltet:

Eintrag

Kurzbeschreibung des Eintrages mit maximal einigen Zeilen.

Optional ein oder mehrere Literaturhinweise den Eintrag betreffend.

Adaptives Transporttheorem

A

Wechseln man das Bezugssystem in der Beschreibung der Gleichungen, zum Beispiel von einem Koordinatensystem, das am Ort fixiert ist, auf eines, das sich auf ein Masseteilchen bezieht, treten bei dieser Transformation zusätzliche Terme in den Gleichungen auf. Das adaptive Transporttheorem beschreibt diese Beziehung und gibt an, wie man diese zusätzlichen Terme berechnet.

[30], [36], [52], [61]

Advektion

Wird eine Formulierung von hydrodynamischen Zusammenhängen verwendet, die nicht der Lagrange-Beschreibung entspricht, benötigt man ein Verfahren, mit dem der Transport von Größen von einer in die nächste Zelle beschrieben werden kann. Die entsprechenden Terme in den Gleichungen werden *Advektion* genannt. Der Transport der Größen wird mit Relativgeschwindigkeit durchgeführt, desweiteren muss dieses Verfahren konservativ formuliert sein, das bedeutet, dass der Verlust oder Gewinn aus der links anschließenden Zelle durch den entsprechenden Gewinn oder Verlust an der rechten Nachbarzelle ausgeglichen werden muss.

[52], [61]

Anfangsbedingungen

Um Rechnungen mit dem SHD-Code durchführen zu können, muss zuerst ein Anfangsmodell erstellt werden. Das bedeutet, man braucht ein Modell, in dem das System der Gleichungen voll gelöst ist und eine zugehörige Gitterpunkteverteilung vorgegeben ist. Erstellen von Anfangsmodellen ist eine eigene Wissenschaft, da bei impliziten Methoden Anfangswert- und Randwertprobleme oftmals vermischt sind.

[16]

B Bewegungsgleichung

Diese Gleichung beschreibt die Bewegung der Teilchen und erhält dabei den Impuls.

[30]

C Courant-Friedrichs-Levy (CFL) - Bedingung

Bei impliziten Rechnungen ist der Zeitschritt nicht durch die sogenannte CFL-Bedingung begrenzt, die sonst bei expliziten Rechnungen dem Zeitschritt ein Limit setzt, das nicht über-/unterschritten werden kann und darf.

[9]

D Diffusion

Im Set der SHD-Gleichungen ist die Diffusion nicht eingebaut, nichtsdestotrotz spielt sie in vielen Bereichen der Astrophysik eine charakteristische

Rolle (cosmic-rays bei der Entwicklung von Supernova-Überresten). Aus diesem Grund muss in die SHD-Gleichungen ein Diffusionsterm eingebaut werden.

[16, Seite 281]

Diskretisierung

Transformation der Differentialgleichungen eines Problems in eine diskretisierte Form. Das bedeutet, dass die Gleichungen nur an einzelnen Punkten ausgewertet werden. Differentialquotienten werden in Differenzenquotienten überführt. Die diskreten Punkte an denen die Differentialgleichungen ausgewertet werden bezeichnet man als Gitterpunkte.

[16], [58]

Donor-Cell

Advektionsschema erster Ordnung. Für die Bildung der Differenzen werden immer jene Werte genommen, die bezüglich der betrachteten Zelle *upwind*, also *stromaufwärts* liegen. Der Transport der Größen in oder durch eine Zelle erfolgt immer in die durch die Strömung gegebene Richtung.

[19, Seite 77]

Eddington Approximation

E

Da in den Momenten der Intensität immer das nächstfolgende in den Gleichungen vorkommt, muss dieses System *geschlossen* werden. Dies geschieht mit der sogenannten Eddington-Näherungen. Der Eddington-Faktor wird als Schließbedingung $\frac{K}{J} = \frac{1}{3}$ gesetzt.

[16], [36]

Energiegleichung

Diese Gleichung formuliert das erste Gesetz der Thermodynamik, die Erhaltung der Energie.

[30]

erstestes Moment

der spezifischen Intensität, entspricht dem radiativen Fluss.

[36]

Euler-Beschreibung

Der Fluss von Teilchen kann auf unterschiedliche Weise beschrieben werden. Jene, in der man örtlich fixierte Zellen betrachtet, die durchflossen werden, nennt man Euler-Formulierung. Gegensatz ist die Lagrange-Beschreibung.

explizit

Extrapolation der Werte der Variablen zum neuen Zeitpunkt durch die Werte zum alten Zeitpunkt. Die Lösung der Gleichungen ist durch direkte Verfahren möglich, der Zeitschritt bei dieser Methode ist allerdings durch die CFL-Bedingung begrenzt. Gegenteil von implizit.

[19, Seite 79]

F Finite-Volumen Diskretisierung

Jene numerische Methode, die beim Übergang von integraler zur diskretisierten Form der SHD-Gleichungen verwendet wird. Das Volumen, über das integriert wird, ist jenes zwischen zwei benachbarten Zellen und ist somit finit, also endlich.

[16], [58]

G Gittergleichung

Um Lösungen von Differentialgleichungen über viele Größenordnungen oder steile Gradienten darstellen zu können, bedarf es eines ausgereiften Gitters, das im Bereich der oben genannten Fälle eine hohe Auflösung besitzt. Im Wiener SHD-Code werden die genannten Probleme mit Hilfe der Gittergleichung gelöst, die die Gitterpunkte entlang kritischer Stellen anhäuft, um genauere Ergebnisse zu erhalten. Die Gleichung wird gleichzeitig mit den anderen Gleichungen des SHD-Systems gelöst.

[13]

H Henyey-Verfahren

Verfahren zur Invertierung der Jacobi-Matrix.

[29, Seite 79]

I implizit

Jene Art die Werte der Variablen zum neuen Zeitpunkt auszurechnen, bei dem nicht aus den Gleichungen zum alten Zeitpunkt interpoliert wird, sondern die Werte zum neuen Zeitpunkt eingehen, das heißt man muss ein iterativen Näherungsverfahren verwenden, um die Gleichungen lösen zu können. Vorteil ist, dass der Zeitschritt nicht durch die CFL-Bedingung limitiert ist. Gegenteil von explizit.

[19, Seite 79]

Jacobi-Matrix

J

Während des Lösungsverfahrens der SHD-Gleichungen wird die invertierte Jacobi-Matrix benötigt. Dabei handelt es sich um die Funktionalmatrix; die Einträge sind alle Variablen nach allen Variablen abgeleitet.

[18], [16]

Kappa-Mechanismus

K

Stellare Pulsationen werden durch Änderungen in der Opazität ausgelöst. Dieser Vorgang wird Kappa-Mechanismus genannt.

[19] bis [23]

konservativ

Jene Formulierung, die konsistet mit den Erhaltungsgrößen ist, also Verfahren, bei denen Impuls, Energie, etc. erhalten werden.

Kontinuitätsgleichung

Jene Gleichung aus dem Set der SHD-Gleichungen, die die Dichte beschreibt. Diese Gleichung steht für die Massenerhaltung.

[30]

künstliche Viskosität

Um Stoßfronten auf einem Gitter auflösen zu können, muss dieselbe über einige Gitterpunkte verschmiert werden. Aus diesem Grund wird ein Viskositätsterm eingebaut, um Diskontinuitäten künstlich zu verbreitern.

[52]

L Lagrange-Beschreibung

Eine Möglichkeit Flüsse durch Zellen zu beschreiben ist die Lagrange-Formulierung. Dabei befindet man sich im System der fließenden Teilchen, die sich durch die Zellen bewegen, das entspricht einer Fixierung der Masse, im Gegensatz zur Euler-Beschreibung.

N Newton-Raphson Iteration

Jenes Verfahren mit dem das Set der SHD-Gleichungen gelöst wird. Auf Grund der großen Nichtlinearität des zu lösenden algebraischen Systems wird dieses iterative Verfahren angewendet.

[16]

nulltes Moment

Hierbei handelt es sich um das nullte Moment der spezifischen Intensität. Es wird auch radiative Energiegleichung genannt.

[36]

O Opazität

Lichtdurchlässigkeit des Sternengases in Abhängigkeit von der chemischen Zusammensetzung, Dichte, Druck und Temperatur. Opak bedeutet lichtundurchlässig. Die Daten sind in Tabellen vorhanden, die in den SHD-Code gelinkt werden.

[16], [37], [38], [24]

P Planck-Mittel

Grenzwert des Kopplungstermes κ_ν zwischen Strahlung und Materie im Grenzwert des optisch dünnen Falles ist das Planck-Mittel.

$$\frac{1}{\kappa_P} = \frac{\int_0^\infty \kappa_\nu B_\nu d\nu}{\int_0^\infty B_\nu d\nu} \quad (\text{A.1})$$

Poisson-Gleichung

In astrophysikalischen Anwendungen spielt die Gravitationskraft naturgemäß eine sehr große Rolle. Um das Gravitationspotential berechnen zu können, muss die Poissongleichung gelöst werden. Sie ist im Set der SHD-Gleichungen diejenige, die die Masse als Variable beschreibt.

[30]

Randbedingungen

R

Am Sternrand und am Innenrand müssen bestimmte Bedingungen spezifiziert werden, um die Differentialgleichungen eindeutig lösen zu können, die se Bedingungen werden im Fall der SHD-Gleichungen Randbedingungen genannt.

[16]

räumliche Glättung

Die Gittergleichung reagiert augenblicklich auf die Veränderung der Variablen. Um zu verhindern, dass diese Veränderungen zu abrupt passieren, wird die Möglichkeit der Veränderung der Zellengröße auf einen bestimmten maximalen Wert begrenzt.

[13]

Relativgeschwindigkeit

Jene Geschwindigkeit zwischen Gasgeschwindigkeit und Gitterbewegung. Entsteht dadurch, da das Koordinatensystem nicht im Raum fixiert ist.

Rosseland-Mittel

Das Rosseland-Mittel ist der Grenzwert des Kopplungstermes κ_ν zwischen Strahlung und Materie im Grenzwert im optisch dicken Fall. Es wird in astrophysikalischen Rechnungen bevorzugt verwendet, da es in Tabellenform vorliegt. Wird wie folgt berechnet.
folgt berechnet:

$$\frac{1}{\kappa_R} = \frac{\int_0^\infty \frac{1}{\kappa_\nu} \frac{\partial B_\nu}{\partial T} d\nu}{\int_0^\infty \frac{\partial B_\nu}{\partial T} d\nu} \quad (\text{A.2})$$

Z zeitliche Glättung

entspricht einer Dämpfung der Veränderungsmöglichkeit in der Gittergleichung, damit keine extremen Sprünge in der zeitlichen Entwicklung auftreten können.

[13], [16, Seite 293]

zeitliche Zentrierung

Bei der Lösung von Pulsationen muss besonderes Augenmerk auf die zeitliche Entwicklung der Rechnung gelegt werden. Beim Verfahren der zeitlichen Zentrierung wird ein *Mischterm* aus Variablen zum alten und zum neuen Zeitpunkt verwendet, um Fehler in der zeitlichen Entwicklung zu minimieren.

[16]

Anhang B

Intrinsische und selbstdefinierte MATHEMATICA-Befehle

In diesem Anhang sind alle MATHEMATICA-Befehle und ihre Kurzbeschreibung aufgelistet, um das Verständnis des erstellten Notebooks **dev_8.2.nb** zu vereinfachen. Die Definitionen sind aus dem Anhang A.8 von [63] entnommen und sind ebenfalls eingebauten MATHEMATICA-Hilfedatei abrufbar. Weitere Informationen über die Verwendungsweise der aufgelisteten Befehle findet man in [12], [63], [64], [65] und [66]. Desweiteren sind die im Modul **ShdNom.m** genutzten selbstdefinierten Befehle angegeben.

Built-in Funktionen in MATHEMATICA beginnen immer mit einem Großbuchstaben, zusammengesetzte Befehlsnamen können innerhalb des Namens ebenfalls Großbuchstaben besitzen. Dies ist so geregelt, um intrinsische von selbstdefinierten Befehlen leichter unterscheiden zu können. Nichtsdestoweniger dürfen auch diese mit Großbuchstaben beginnen. Die in dieser Arbeit verwendeten selbstdefinierten Befehle beginnen mit Kleinbuchstaben.

Die Einträge sind wie folgt gestaltet:

Befehl
Beschreibung des Befehls.

Weiters gelten folgende Textkonventionen laut folgender Liste:

Tabelle B.1: Textkonventionen im Anhang B

Font	Bedeutung
Schreibmaschine	MATHEMATICA-Befehle
<i>kursiv</i>	Attribute von MATHEMATICA-Befehlen
<i>antiqua</i>	Erläuterungen und Erklärungen

Sonder- zeichen

" "

Anführungszeichen kennzeichnen beim MATHEMATICA-Output ASCII-Strings. Diese Gänsefüßchen werden durch das Shellsript **reinigen** gelöscht.

\

ist das Zeilenfortsetzungszeichen, das MATHEMATICA verwendet. Mittels **reinigen** werden diese Backslashes durch Kaufmann-Unds (&) ersetzt, dem Zeilenfortsetzungszeichen in FORTRAN90/95.

;

Der Strichpunkt wird verwendet, um den Output in das Notebook während der Berechnungen zu unterdrücken. Er steht immer am Ende einer Zeile.

B Begin[]

Dieser Befehl wird zum Beispiel verwendet, um in Modulen den Beginn desselben nach der Einleitung zu definieren.

`Begin["context"]` stellt den gerade aktiven Kontext zurück.

BeginPackage[]

Der Befehl steht typischerweise zu Beginn jedes Moduls und bildet seine Einleitung.

`BeginPackage["context"]` setzt die Kontexte *context* und *System* aktiv.

bruch[]

Selbstdefinierte Ersetzungsfunktion im Module *ShdNom.m*. Sie führt alle in den Gleichungen verwendeten Abkürzungen für Brüche in die realen Zahlenwerte über.

C change[]

Diese selbstdefinierte Funktion innerhalb des Notebooks überführt die Hilfsvariablen in die innerhalb des SHD-Codes verwendeten Variablennamen. Durch diese Namenssubstitution können die Hilfsvariablen nicht mehr durch primäre Variablen ersetzt werden. Darüberhinaus ist dadurch die SHD-Code Variablen-Konvention fixiert.

Clear[]

löscht die Werte und Definitionen, die mit *symbol* verbunden sind.

`Clear[symbol1, symbol2, ...]` löscht die Belegungen (Werte und Definitionen) von *symbol₁*, *symbol₂*, ...

Close[]

schließt die Ausgabedatei.

`Close["Ausgabedatei"]` schließt die Datei *Ausgabedatei*, in die MATHEMATICA schreibt.

D[]

D

Diese Funktion wird verwendet, um alle Ableitungen zu berechnen, ein sehr einfaches, aber mächtiges Werkzeug. Es ist sozusagen das Herzstück der Berechnung der Jacobi-Matrix.

`D[f, x]` berechnet die partielle Ableitung von *f* nach *x*.

`D[f, {x, n}]` berechnet die n-te Ableitung von *f* nach *xⁿ*.

`D[f, x1, x2, ...]` berechnet die Ableitungen von *f* nach *x₁*, *x₂*, ...

Die Ableitungen aller Einträge, die von *x* unabhängig sind, ergeben den Wert 0.

Date[]

gibt Datum und Uhrzeit wie auf dem Computer eingestellt wider.

`Date[]` ergibt Datum und Uhrzeit in folgender Formatierung: {Jahr, Monat, Tag, Stunde, Minute, Sekunde} als Liste.

Alle ausgegebenen Zahlen sind bis auf den Sekundenwert *Integers*.

Do[]

Wie in jeder anderen Programmiersprache bezeichnet dieser Befehl eine Schleifenberechnung über einen oder mehrere Indizes.

`Do[Ausdruck, {imax}]` evaluiert *Ausdruck* *imax* mal.

`Do[Ausdruck, {i, imin, imax, di}]` evaluiert *Ausdruck* *imax* mal, beginnend bei *imin* mit der Schrittweite *di*. `Do[Ausdruck, {i, imin, imax}, {j, jmin, jmax}]` evaluiert *Ausdruck* über die Indizes *i* und *j*.

End[]

E

Dieser Befehl wird zum Beispiel verwendet, um in Modulen das Ende desselben und damit den Abschluss zu definieren.

`End["context"]` gibt den gerade aktiven Kontext *context* aus und belegt den davor gültigen.

EndPackage[]

Ist das Ende eines Moduls und schließt es ab. Steht immer gemeinsam mit `BeginPackage[]`.

F False

Wird hier verwendet, um in `SetOptions[]` Optionen mit der Boole'schen Variable *Falsch* zu belegen.

zum Beispiel: `SetOptions[Simplify, Trig → False]` schaltet die Rechenregeln für trigonometrische Funktionen im `Simplify`-Befehl aus.

fortconst[]

ist ein selbstdefinierter Befehl, der sich aus mehreren intrinsischen Befehlen zusammensetzt. Zuerst werden alle numerischen Werte eines Ausdrucks wie Brüche oder *Real*-Werte in entsprechende Variablen umbenannt. Danach wird mittels `N[]` der numerische Wert des Ausdrucks bestimmt, durch `FortranForm[]` in FORTRAN-Code und abschließend durch `ToString[]` in einen ASCII-String umgewandelt.

FortranForm[]

ist der MATHEMATICA eingebaute Befehl, um Output in FORTRAN-Code umzuwandeln.

Beispiel: `FortranForm[ $\frac{1}{2}x^2$ ]=x**2/2.`

G General

ist ein Symbol, das systemimmanenten Befehlen zugeordnet ist. Mit den Tags `On` oder `Off` kann dieses Symbol ein- oder ausgeschaltet werden.

Beispiel: `Off[General::spell]` schaltet die Warnings von MATHEMATICA betreffend Namensgleichheit von Variablennamen aus.

I If[]

ist ein klassischer Programmierungsbefehl, der logische Verweise zulässt.

`If[Bedingung, w, f]` ergibt *w*, wenn *Bedingung* wahr ist, ansonsten wird *f* ausgeführt.

`If[Bedingung, w, f, e]` ergibt *e*, egal ob *Bedingung w* oder *f* erfüllt ist.

N N[]

`N[Ausdruck]` gibt den numerischen Wert von *Ausdruck* aus. `N[1]=1.`, `N[ $\frac{1}{2}$ ]=0.5`. Diese Funktion wird verwendet, um dem FORTRAN90/95-Standard näher zu kommen, da dort die Ziffer 1 als *Integer*, die Ziffer 1. aber als *Real* definiert ist.

nom???

Diese Funktionen, bei denen die Fragezeichen für Nummer zwischen 1 und

999 stehen, übersetzen die in MATHEMATICA verwendeten Variablennamen in jene, die im SHD-Code verwendet werden. Dies muss geschehen, da innerhalb von MATHEMATICA Klammern und Rechenzeichen (vorallem das im SHD-Code häufig verwendete Pluszeichen) nicht für Variablennamen erlaubt sind. Die Funktion wird auf Strings angewendet, deswegen sind um die Variablen immer Anführungszeichen zu setzen.

Beispiel: `nom2 = "x1mm"-> "X(L,MM)";` ersetzt die *Mathematica-Variable* `x1mm` in die SHD-Code-Varibale `X(L,MM)`.

num?

Diese drei Ersetzungsfunktionen (? steht für 1, 2 oder 3) sind notwendig, um den FORTRAN-Output zu verkürzen. Da auf alle Ergebnisse die Funktion `N[]` angewendet wird, sind alle Ergebnisse von numerischer Natur. Ausdrücke wie `(7+x)` werden manchmal als `1*(7+x)` dargestellt. Die Funktionen werden verwendet, um diese redundanten Rechenoperationen zu entfernen.

`num1 = "*z1 "-> "";` ersetzt zum Beispiel die von `FortranForm[]` erzeugte überflüssige Operation *Multiplikation mit 1*.

Off[]

O

schaltet bestimmte MATHEMATICA-Symbole aus, siehe Eintrag **General**.

`Off[symbol::tag]` schaltet den *tag* von *symbol* aus.

OpenAppend[]

öffnet eine Ausgabedatei und hängt an den bereits bestehenden Inhalt den Output an.

Format: `OpenAppend["Ausgabedatei"]`

OpenWrite[]

öffnet eine Ausgabedatei, löscht allenfalls bestehende Einträge und schreibt gewünschten Output von Beginn an in die Datei.

Format: `OpenWrite["Ausgabedatei"]`

PageWidth

P

bestimmt die Zeilenbreite (Zeilenlänge in Characters) der Ausgabedatei. Für den Befehl `FortranForm[]` ist diese Länge auf 72 voreingestellt, da das dem FORTRAN77-Standard entspricht. Um die Übersichtlichkeit zu erhöhen (der FORTRAN90/95-Intel-Compiler kann längere Zeilen verarbeiten) wurde der Wert auf 130 eingestellt, kann aber beliebig variiert werden, `SetOptions[Ausgabedatei, PageWidth -> Infinity]` erlaubt beliebig lange Ausgabezeilen. Ein für den 2D-SHD-Code getesteter Compiler ist in der Lage, 456 Characters pro Zeile zu verarbeiten.

SetDirectory[]

S

legt das Arbeitsverzeichnis fest, in dem MATHEMATICA seine Berechnungen ausführt, Input einliest und Output ausgibt.

Format: `SetDirectory["Arbeitsverzeichnis"]`

SetOptions[]

ist ein Befehl, der Optionen eines MATHEMATICA-Befehls einstellt oder verändert. Für jeden Befehl gibt es eine Optionenliste, die vor der Verwendung bearbeitet werden kann. Für jede dieser Optionen gibt es eine Default-Einstellung. Im verwendeten Programm wird die Option *Trig* von `Simplify[]` auf *False* gesetzt (siehe dort). Desweiteres kann `SetOptions[]` auch für Eingabe- oder Ausgabedateien gesetzt werden, siehe `PageWidth`.

shd[]

setzt sich aus den Funktionen `nom[]`, `num[]` und `bruch[]` genau in dieser Reihenfolge zusammen und ist als Befehl durch die Eigenschaften der drei Funktionen bestimmt (siehe dort).

Simplify[]

ist ein sehr mächtiger, aber schwer auszurechnender Befehl von MATHEMATICA. `Simplify[]` vereinfacht, nach den innerhalb von MATHEMATICA gespeicherten Vorgaben, mathematische Ausdrücke. Diese Berechnungen verbrauchen sehr viel Rechnerzeit und liefern zum Teil nur Ergebnisse, die im nachhinein bearbeitet werden müssen. Um die Geschwindigkeit des Befehls zu erhöhen, wurden die Optionen desselben verändert, siehe `SetOptions[]`. Der *große Bruder* dieses Befehls ist `FullSimplify[]`, bei dem weitere nicht oft verwendete Vereinfachungsregeln (zum Beispiel spezielle Logarithmenregeln) angewandt werden.

StringJoin[]

verbindet zwei oder mehr Strings zu einem. `StringJoin["di", "pl", "om"]` ergibt *"diplom"*.

StringReplace[]

ersetzt innerhalb von Strings einzelne Characters durch andere. Das Ergebnis von `StringReplace["Matrix", "x" → "zen"]` ist *"Matrizen"*.

StringTake[]

extrahiert aus einem gegebenen String einzelne Characters.

`StringTake["string", n]` extrahiert die ersten *n* Characters von *string*.
`StringTake["string", -n]` extrahiert die letzten *n* Characters von *string*.

`StringTake["string", {m,n}]` extrahiert die Characters von der Position *m* bis *n* in *string*.

Table []

T

In dem Programm wird `Table []` verwendet, um eine zweidimensionale Liste ($\hat{=}$ Jacobi-Matrix) zu erstellen. Hier wird `Table[Ausdruck, {i, imin, imax}, {j, jmin, jmax}]` verwendet, um die Matrix zu erstellen, was einer zweidimensionalen Schleife über die Einträge entspricht.

TableForm []

gibt die Elemente einer zweidimensionalen Liste in rechteckigen Zellen aus. `TableForm[Liste]` führt keine Berechnungen durch, sondern wird nur verwendet, um Listen in Matrixform ausgeben zu können.

```
TableForm[{{1,2},{3,4}}] =
```

1	2
3	4

TimeConstraint

ist eine Option des Befehls `Simplify[]`, mit dem die Dauer der Berechnungen limitiert werden können.

ToString []

konvertiert einen MATHEMATICA-Ausdruck in einen ASCII-String. `ToString[ $\frac{1}{2}u$ ]` = " $\frac{u}{2}$ ".

UpperCase []

U

Dieser Befehl wird auf Strings angewendet und verwandelt alle vorkommenden Kleinbuchstaben in Großbuchstaben. Diese Funktion wird verwendet, um den Standard der Großschreibung im SHD-Code zu gewährleisten.

`UpperCase["diPlom"]` ergibt "*DIPLOM*"

vor []

V

Diese sehr wichtige Funktion wird als Vorzeichenregelfunktion bezeichnet. Für MATHEMATICA besitzen die Ausdrücke `Variable` und `-Variable` unterschiedliche `Heads`, das bedeutet MATHEMATICA nimmt diese, nur durch das Vorzeichen unterschiedlichen Variablen, als unterschiedliche Ausdrücke wahr. `Variable` ist ein *Symbol*, während `-Variable` als $-1 * \textit{Symbol}$ und somit als Multiplikation interpretiert wird. Einfache Ersetzungen finden somit nicht statt, weil alle Variablen mit negativem Vorzeichen nicht *gefunden* werden. Diese Funktion bereinigt das Problem und ersetzt Variablen egal welchen Vorzeichens mit den entsprechen Ersetzungen.

`vor[z $\overline{e}d$ - >  $\frac{1}{3}$ ]` ersetzt die Variable *zed* mit $\frac{1}{3}$ und *-zed* mit $-\frac{1}{3}$.

Write []

W

schreibt Output in die Ausgabedatei.
Format: `Write["Ausgabedatei", Ausdruck]`

Anhang C

MATHEMATICA-Notebook zur Berechnung der Ableitungen

In diesem Anhang ist das verwendete MATHEMATICA-Notebook zur Berechnung der Ableitungen für die Erstellung der Jacobi-Matrix beigefügt. Es ist unter <http://www.astro.univie.ac.at/~dorfi> nach Beantragung eines Passwortes unter kittel@astro.univie.ac.at downloadbar.

Das Notebook ist selbsterklärend, Kommentare, Erläuterungen und Beschreibungen finden sich innerhalb des Codes. Für alle weiteren Fragen siehe Hauptteil dieser Arbeit.

Ableitungen des 1D-Modells mit geeignetem Output:

author: Kittel Matthias

date: 09 04 03

modified: 10 04 03
 =>gesamter Output,Poisson eingebaut
 14 04 03
 =>Sbleitungsliste eingebaut,Ableitungen eingebaut
 =>Dateiabfrage vor Rechnungsbeginn eingebaut
 15 04 03
 =>Einbau Ableitungen in Ausgabe-file
 28 04 03=>ToString der Ausgabe
 =>Variablensetzen nach ToString
 =>bei Ableitungswert=0,kein Output
 29 04 03
 =>Erweiterung logfile
 =>in Output.bei real-Zahlen eingefuegt
 =>Ausgabe der Gleichungen in die Output-Files
 17 06 03
 =>Modell mit 5 Gleichungen (Gitter-,Kontinuitaets-,Bewegungs,Energie-und Poisson-Gleichung)
 ohne Viskositaet und Strahlung
 =>Leerzeilen im Output nach jedem Ableitungseintrag
 18 06 03
 =>Filbezeichner,Subroutinebezeichner eingebaut
 =>Ausgabe in hivek_mathe
 20 06 03
 =>Reorganisation der einzelnen Code-Bereich
 =>Dokumentationsausweitung
 21 06 03
 =>Einbau Kontinuitaetsgleichung
 04 09 03
 =>Aenderung des Paramenters off in aus
 =>Variablenumbenennung
 =>Laden der Packages DoFor und ShdNom
 09 09 03
 =>Umstrukturierung und Vereinfachung
 =>Output-Verbesserung
 =>Einbau Vorzeichenregel
 =>Einbau Optionen von Simplify
 =>Log-file Erweiterung
 =>Dokumentationserweiterung
 22 09 03
 => weitere Umstrukturierungen
 30 09 03

zeitlicher Verlauf der Entstehung des Notebooks

=> Dokumentation, lay-out notebook
 01 10 03
 => Einbau G[4]
 02 10 03
 => Umkehrung Simplify- und change-Befehl
 => Einbau neuer Hilfsvariablen
 => Einbau der Anweisungen
 05 10 03
 => erneuerstes Modul DoFor.m
 => Gleichungseingabe geändert
 08 10 03
 => numerischer Output verbessert
 14 10 03
 => Stringlaenge fuer Dateiabschluss vergroessert, beginn kontinuetaetsgleichung
 29 10 03
 => Fertigstellung Gleichungssystem (Hydrodynamik, zeitlich zentriert)
 30 10 03
 => Erweiterung logbereich

purpose: errechnet Ableitungen (Jacobi-Matrix) fuer das 1D-Modell fuer die Gleichungen mr,md,mm,mu,me ohne Strahlung und ohne Viskositaet

- .) Setzen des Rechenverzeichnisses (in diesem Verzeichnis werden die erzeugten Files gespeichert) - Setdirectory
- .) Modellnummerbezeichnung (Bezeichnung optional fuer Dokumentation)- modell
- .) Einstellen der Anzahl der Zeichen pro Zeile (Pagewidth), bezieht sich auf den verwendeten FORTRAN-Compiler - pagewidth
- .) Ausschalten der Warnings und Fehlermeldungen (um unnoetige beeps zu vermeiden) - off: 0:aus, 1:ein
- .) Loeschen der bestehenden Files im Arbeitsverzeichnis (sollen die bestehen Files geloescht werden, wenn nicht bricht das Programm bei existierenden Files ab, Schalter, um versehentliches Loeschen der erzeugten Files zu vermeiden)- loeschen: 0:aus, 1:ein
- .) Anlegen eines Logfiles (in diesem Logfile werden Datum, Uhrzeit, Modellnummer und Pagewidth abgespeichert) - logfile: 0:aus, 1:ein
- .) Anzahl der Gleichungen (optional, im Moment nur ng=5 ohne Strahlungsterme fuer neue Diskretisierung verwendbar) - ng: {5,7,8}
- .) Fuer den Tag Simplify werden Optionen gesetzt: winkel...Winkelfunktionen werden bei der Berechnung beruecksichtigt: 0:aus, 1:ein, zeit: Zeit fuer die maximale Berechnungsdauer fuer Simplify in Sekunden
- .) Dieses notebook benoetigt folgedes Modul: **ShdNom.m**.

Alle Erlaeuterungen sind als Text innerhalb dieses Notebooks angegeben, **Kommentare sind blau unterlegt**, editierbare Zellen sind grau unterlegt und schwarz gerahmt. Alle weiss unterlegten Zellen sind gesperrt und sollten nicht edidiert werden.

Anweisungen zur Benutzung sind auf gelben Hintergrund gehalten.

Hier sind die Umgebungsvariablen fuer die Benuetzung des Notebooks einzugeben. Alle Variablen sind vor Start der Evaluation richtig zu setzen, siehe Kommentare.

Erläuterungen zur Benutzung

```

SetDirectory["/home/kittel/dipl/
    mathematica/1D/rho_neu/devdata"];
(* Arbeitsverzeichnis, ist einzugeben *)
<< ShdNom`;
(* Laden des Moduls ShdNom.m *)
dos = 0;
(* Switch fuer Art des Betriebssystems *)
modell = 1;          (* Modellnummer *)
pagewidth = 55;      (* Zeilenlaenge *)
aus = 1;
(* Switch fuer das Ausschalten der
    Warnings bei Namensgleichheiten *)
loeschen = 1;        (* Switch fuer das
    Loeschen der bereits exisiterenden Files *)
logfile = 1;
(* Switch fuer das Erstellen eines Logfiles *)
winkel = 0;          (* Switch
    fuer die Option Trig bei Simplify,
    Berechnung mit oder ohne Winkelfunktionen *)
zeit = 25;            (* Switch fuer die
    Option TimeConstraint bei Simplify *)
ng = 5;
(* Anzahl der Gleichungen *)
logtxt = "hydro, zz, ohne visko";
(* Infotext zum gerechneten
    Modell *)

```

Abfrage der Parameterwerte und Kontrollausgabe, Schreiben des Logfiles, Abfrage ob bestehende Ausgabefiles
 geloescht werden sollen

Einstellungen und Filehandling

```

If[logfile == 1,
  logout = OpenWrite["logfile"];
  Write[logout, "Datum:", Date[]];
  Write[logout, "Modellnummer:", modell];
  Write[logout, "Gleichungen:", ng];
  Write[logout, "Zeilenbreite:", pagewidth];
  Write[logout, "Off-Befehl:", aus, "-> 0...aus, 1...ein"];
  Write[logout, "Winkelfunktionen:", winkel, "-> 0...aus, 1...ein"];
  Write[logout, "Simplify-Zeit:", zeit, "Sekunden"];
  Close[logout];]
Print["Dos wurde auf ", dos, " gesetzt."];
Print["Off wurde auf ", aus, " gesetzt."];
If[aus == 1, Off[General::spell, General::spell1]]
Print["Loeschen wurde auf ", loeschen, " gesetzt."]
If [loeschen == 1 && dos == 0, Run["rm *.f90"],
  If[FileNames[] != {},
    Print["Speicherung der Files?!"]; Exit[]];]
If [loeschen == 1 && dos == 1, Run["del *.f90"]];
If[winkel == 0, SetOptions[Simplify, Trig -> False]];
SetOptions[Simplify, TimeConstraint -> zeit];

Dos wurde auf 0 gesetzt.

Off wurde auf 1 gesetzt.

Loeschen wurde auf 1 gesetzt.

```

Liste der internen Variablen (Stuetzstellen), nach denen abgeleitet wird. dev8...8 Gleichungen, dev7...7 Gleichungen, dev5...5 Gleichungen

xlmr ... x bedeutet Variable, l ist die Stuetzstelle, mr steht fuer Radius

lmII ... Stuetzstelle I-2, alle anderen entsprechend, p steht fuer +

mm ... Masse

md ... Dichte

mu ... Geschwindigkeit

me ... innere Energie

mj ... Strahlungsenergie

mh ... Strahlungsfluss

mk .. Konvektion

Logfile-Output und weitere Erläuterungen

```

dev8 = {xlmIImr, xlmImr, xlmr, xlpImr, xlpIImr,
        xlmIImm, xlmImm, xlm, xlpImm, xlpIImm,
        xlmIImd, xlmImd, xlm, xlpImd, xlpIImd,
        xlmIImu, xlmImu, xlm, xlpImu, xlpIImu,
        xlmIIme, xlmIme, xlm, xlpIme, xlpIIme,
        xlmIImj, xlmImj, xlmj, xlpImj, xlpIImj,
        xlmIImh, xlmImh, xlmh, xlpImh, xlpIImh,
        xlmIImk, xlmImk, xlmk, xlpImk, xlpIImk};
dev7 = {xlmIImr, xlmImr, xlmr, xlpImr, xlpIImr,
        xlmIImm, xlmImm, xlm, xlpImm, xlpIImm,
        xlmIImd, xlmImd, xlm, xlpImd, xlpIImd,
        xlmIImu, xlmImu, xlm, xlpImu, xlpIImu,
        xlmIIme, xlmIme, xlm, xlpIme, xlpIIme,
        xlmIImj, xlmImj, xlmj, xlpImj, xlpIImj,
        xlmIImh, xlmImh, xlmh, xlpImh, xlpIImh};
dev5 = {xlmIImr, xlmImr, xlmr, xlpImr, xlpIImr,
        xlmIImm, xlmImm, xlm, xlpImm, xlpIImm,
        xlmIIImhd, xlmhmd, xlpImhd, xlpIIImhd, xlpVhmd,
        xlmIImu, xlmImu, xlm, xlpImu, xlpIImu,
        xlmIIImhe, xlmhme, xlpImhe, xlpIIImhe, xlpVhme};
If[ng == 8, devanzahl = dev8,
   If[ng == 7, devanzahl = dev7, devanzahl = dev5]];

```

Hier sind nun die Gleichungen einzugeben. Fuer jede dieser Gleichungen gibt es ein grau unterlegtes Feld.

Gittergleichung:

G[1] = 0;

Poisson-Gleichung:

G[2] = (xlm - xlpImm) - xlpImd * dvolph;

Kontinuitaetsgleichung

Definition der Variablen nach denen abgeleitet wird, sowie die ersten beiden Gleichungen

```
G[3] = (xlphmd * dvolph - xalphmd * dvolaph) +
mur2dl *
(xlphmd * sadsph + (1 - sadsph) * xlmhmd) -
mur2dlpI * (sadsplIIh * xlpIIhmd +
(1 - sadsplIIh) * xlphmd) ;
```

Bewegungsgleichung:

```
G[4] = (xlmv * vecmd * dvoll - xalmv * vecamd * dvolal) +
mur2dmlmh * (sadv * (vecmd * xlmv) +
(1 - sadv) * ((vecmdmI * xlmImu))) -
mur2dmlph * (sadvpI * (vecmdpI * xlpImu) +
(1 - sadvpI) * (vecmd * xlmv) +
xlmr2 * tst * (prgzmh - prgzph) +
 $\frac{G * xlmv}{xlmr^2} * vecmd * dvoll * tst;$ 
```

Energiegleichung:

```
G[5] = (xlphmd * xlphme * dvolph -
xalphmd * xalphme * dvolaph) +
mur2dl * (xlphmd * xlphme * sadsph +
(1 - sadsph) * xlmhmd * xlmhme) -
mur2dlpI * (sadsplIIh * xlpIIhmd * xlpIIhme +
(1 - sadsplIIh) * xlphmd * xlphme) +
prgzph * tst * (xlmr2 * xlmv - xlpImr2 * xlpImu) ;
```

Strahlungsenergiegleichung:

```
G[6] = 0;
```

Strahlungsflussgleichung:

```
G[7] = 0;
```

Konvektionsgleichung:

Gleichungen 3 bis 7

```
G[8] = 0;
```

Variablenbezeichner (setzt die im SHD-Code verwendeten Bezeichnungen fuer die Art der Gleichung):

```
Vb[1] = "MR"; Vb[2] = "MM"; Vb[3] = "MD"; Vb[4] = "MU";  
Vb[5] = "ME"; Vb[6] = "MJ"; Vb[7] = "MH"; Vb[8] = "MK";
```

Filebezeichner (belegt die Namen der FORTRAN90/95-files, die in den Code gelinkt werden):

```
Fb[1] = "mr.f90"; Fb[2] = "mm.f90"; Fb[3] = "md.f90"; Fb[4] = "mu.f90";  
Fb[5] = "me.f90"; Fb[6] = "mj.f90"; Fb[7] = "mh.f90"; Fb[8] = "mk.f90";
```

Subroutinebezeichner (bezeichnet die Namen der Subroutinen, wie im SHD-Code verwendet):

```
Sb[1] = " EQGRID (L)"; Sb[2] = " EQMAS (L)";  
Sb[3] = " EQKONT (L)"; Sb[4] = " EQMOT (L)"; Sb[5] = " EQENE (L)";  
Sb[6] = " EQSENE (L)"; Sb[7] = " EQSFLX (L)"; Sb[8] = " EQKONV (L)";
```

Vorzeichenregel und Vorzeichenwechselfunktion:

```
vor[rule_] := Table[(-1)i rule[[1]] → (-1)i rule[[2]], {i, 1, 2}];  
(* Vorzeichenregel, erstellt eine Liste, in der die Ersetzungsregeln fuer  
Werte mit positiven und negativen Vorzeichen eingetragen sind *)
```

Hier werden alle numerischen Konstanten eingegeben, die ganzzahligen oder rationalen Wertem entsprechen und in PHYSKO (SHD-Code) definiert sind.

```
fortconst[x_] := ToString[FortranForm[  
  N[x /. vor[ $\frac{1}{3}$  → zed] /. vor[ $\frac{2}{3}$  → zzd] /.  
    vor[ $\frac{4}{3}$  → zvd] /. vor[ $\frac{5}{2}$  → zfv] /.  
    vor[ $\frac{5}{3}$  → zfd] /. vor[ $\frac{1}{4}$  → zev] /.  
    vor[ $\frac{1}{6}$  → zes]] /. vor[1. → z1] /.  
    vor[2. → z2] /. vor[3. → z3] /.  
    vor[4. → z4] /. vor[5. → z5] /.  
    vor[6. → z6] /. vor[7. → z7] ]];
```

Gleichung 8, Definitionen für den Output, Funktion fortconst[]

Alle Ersetzungsregeln sind hier einzugeben. Jeder Hilfsvariable wird eine neue Bezeichnung zugeordnet, um das spätere Ersetzen zu ermöglichen. Das passiert schon an dieser Stelle, damit die Ersetzungen auch schon Wirkung auf die Gleichungen selbst haben.
 Folgende Syntax ist zu befolgen: ... /. vor[ersetzt1 -> ersetzt2]

```
change[x_] :=
  x /. vor[dvolph → dvolasca] /. vor[dvoll →
    dvolvec] /.
    vor[vecmd → xvecmd] /.
    vor[vecmdmI → xvecmdmI] /.
    vor[vecmdpI → xvecmdpI] /.
    vor[drdreI → dkub] /.
    vor[drdreipI → dkubpI] /.
    vor[drdreimI → dkubmI] /.
    vor[differ1 → diff1] /.
    vor[differ2 → diff2] /.
    vor[differ3 → diff3] /.
    vor[mur2dl → ur2dl] /.
    vor[mur2dlpI → ur2dlpI] /.
    vor[dvolaph → dvolasca] /.
    vor[dvolal → dvolavec] /.
    vor[vecamd → xvecamd] /.
  vor[
    mur2dmlmh →
      ur2dmlmh] /.
  vor[mur2dmlph → ur2dmlph];
(* dazugehoerige Vorzeichenwechselfunktion,
  ersetzt Werte mit beiderlei Vorzeichen *)
```

Ausgabe der Gleichungen im FORTRAN90/95-Standard:

Funktion change[]

```

Do[outeqs = OpenWrite[Fb[i], PageWidth → pagewidth];
  Write[outeqs, "subroutine ", Sb[i]];
  Write[outeqs, ""];
  Write[outeqs, "!SHD-Gleichungen und Ableitungen nach den Variablen"];
  Write[outeqs, "!-----"];
  Write[outeqs, "!Ersetzung von Sonderzeichen:"];
  Write[outeqs, "!Backslash durch blank&"];
  Write[outeqs, "!Anfuhrungszeichen mit blank"];
  Write[outeqs, "!-----"];
  Write[outeqs, ""];
  Write[outeqs, "!-----"];
  Write[outeqs, "!Header:"];
  Write[outeqs, "!Modell = ", modell];
  Write[outeqs, StringJoin["!Info: ", logtxt]];
  Write[outeqs, "!Datum: ", Date[]];
  Write[outeqs, "!-----"];
  Write[outeqs, ""];
  Write[outeqs, "use alles"];
  Write[outeqs, "use PHYSKO"];
  Write[outeqs, ""];
  Write[outeqs, "G(", Vb[i], ", L)=",
    ToUpperCase[shd[fortconst[change[G[i]]]]];
  Write[outeqs, ""];
  Close[Fb[i]]; {i, 1, ng}]

```

Hilfsvariablen fuer die Gleichungen:

Alle gewuenschten Hilfsvariablen muessen hier eingetragen werden und zwar mit dem Variablennamen, die in den Gleichungen vorkommen beziehungsweise in den Ersetzungsregeln an erster Stelle stehen. Kommentare sin hier unbedingt notwendig.

```

vecmd = (xlphmd (xlmr3 - xlpImr3) +
          xlmhmd (xlmImr3 - xlmr3)) / (xlmImr3 - xlpImr3);
(* Dichte am Zellenrand, vektorielle Groesse *)
vecamd = (xalphmd (xalmr3 - xalpImr3) + xalmhmd
          (xalmImr3 - xalmr3)) / (xalmImr3 - xalpImr3);
(* Dichte am Zellenrand, vektorielle
  Groesse zum alten Zeitpunkt *)
dvoll =  $\frac{1}{6}$  (xlmImr3 - xlpImr3);
(* Volumen am Zellenrand, gemittelte Groesse *)
dvolal =  $\frac{1}{6}$  (xalmImr3 - xalpImr3);
(* Volumen am Zellenrand,

```

Header der Ausgabefiles der Ableitungen, Definiton der Hilfsvariablen

```

gemittelte Groesse zum alten Zeitpunkt *)
dvolph =  $\frac{1}{3}$  (xlmr3 - xlpImr3);
(* Volumen in der Zellmitte, skalare Groesse *)
dvolaph =  $\frac{1}{3}$  (xalmr3 - xalpImr3);
(* Volumen in der Zellmitte,
alte Zeit, skalare Groesse *)
drdreier = (xlmr3 - xalmr3);
(* temporaere Volumensaenderung *)
differ1 = (xlmImr3 - xlpImr3);
(* kubische Differenz 1 *)
differ2 = (xlmImr3 - xlmr3);
(* kubische Differenz 2 *)
differ3 = (xlmr3 - xlpImr3);
(* kubische Differenz 3 *)

mur2dl = (zz * xlmu + zm * xalmu) * tst *
      (zz * xlmr2 + zm * xalmr2) -  $\frac{1}{3}$  * drdreier;
mur2dlpI = (zz * xlpImu + zm * xalpImu) *
      tst * (zz * xlpImr2 + zm * xalpImr2) -
       $\frac{1}{3}$  * (xlpImr3 - xalpImr3);
mur2dlmI = (zz * xlmImu + zm * xalmImu) *
      tst * (zz * xlmImr2 + zm * xalmImr2) -
       $\frac{1}{3}$  * (xlmImr3 - xalmImr3);

mur2dmlmh =  $\frac{1}{2}$  (mur2dl + mur2dlmI);
mur2dmlph =  $\frac{1}{2}$  (mur2dl + mur2dlpI);

```

File der Hilfsvektoren: "hivek_mathe.f90"

weitere Hilfsvariablen

```

hivout = OpenWrite["hivek_mathe.f90", PageWidth → 130]; (* Kopf des Files *)
Write[hivout, "subroutine hivek_mathe"];
Write[hivout, ""];
Write[hivout, "Hilfsvektoren fuer die Ableitungen der Gleichungen"];
Write[hivout, "-----"];
Write[hivout, "Ersetzung von Sonderzeichen:"];
Write[hivout, "Backslash durch blank&"];
Write[hivout, "Anfuhrungszeichen mit blank"];
Write[hivout, "-----"];
Write[hivout, ""];
Write[hivout, "-----"];
Write[hivout, "Header:"];
Write[hivout, "Modell = ", modell];
Write[hivout, StringJoin["Info: ", logtxt]];
Write[hivout, "Datum: ", Date[]];
Write[hivout, "-----"];
Write[hivout, ""];
Write[hivout, "use alles"];
Write[hivout, "use PHYSKO"];
Write[hivout, ""];
Write[hivout, ""];
Write[hivout, "IMPLICIT NONE"];
Write[hivout, ""];
Close[hivout];

```

Hier wird das File fuer die Hilfsvariablen zusammengestellt. Die Variablen werden hier so benannt wie im SHD-Code verwendet (Überprüfung mit dem Modul ShdNom.m notwendig. Desweiteren ist auch darauf zu achten die richtigen Schleifengrenzen zu setzen. Vor den Hilfsvariablen ist jeweils shd[fort[]] zu setzen.

```

hivout = OpenAppend[
    "hivek_mathe.f90", PageWidth → pagewidth];
Write[hivout, "INTEGER :: L"];
Write[hivout, ""];
Write[hivout, "DO L=1,NPT"];
(* Schleife von 1 bis
    Maximalzahl (NPT, siehe SHD-Code) *)
Write[hivout, ""];
Write[hivout, "DRDREI (L)=", shd[fortconst[
    (xlmr - xalmr) (xlmr2 + xlmr * xalmr + xalmr2) ]]];
Write[hivout, ""];
Write[hivout, "ENDDO"];
Write[hivout, ""];
Write[hivout, "DO L=1,NPM"];
(* Schleife von 1 bis Maximalzahl-

```

Header der Hilfsvariablendatei, sowie Ausgabe der Hilfsvariablen

```

1 (NPM, siehe SHD-Code *)
Write[hivout, ""];
Write[hivout, "DVOLSCA (L)=",
  shd[fortconst[ $\frac{1}{3}$  (xlmr - xlpImr)
    (xlmr2 + xlmr * xlpImr + xlpImr2) ] ] ];
Write[hivout, ""];
Write[hivout, "DVOLASCA (L)=",
  shd[fortconst[ $\frac{1}{3}$  (xalmr - xalpImr)
    (xalmr2 + xalmr * xalpImr + xalpImr2) ] ] ];
Write[hivout, ""];
Write[hivout, "DIFF3 (L)=",
  shd[fortconst[ (xlmr - xlpImr)
    (xlmr2 + xlmr * xlpImr + xlpImr2) ] ] ];
Write[hivout, ""];
Write[hivout, "ENDDO"];
Write[hivout, ""];
Write[hivout, "DO L=2,NPM"];
(* Schleife von 2 bis Maximalzahl-
1 (NPM, siehe SHD-Code *)
Write[hivout, ""];
Write[hivout, "DVOLVEC (L)=",
  shd[fortconst[ $\frac{1}{6}$  (xlpImr - xlmImr)
    (xlpImr2 + xlpImr * xlmImr + xlmImr2) ] ] ];
Write[hivout, ""];
Write[hivout, "DVOLAVEC (L)=",
  shd[fortconst[ $\frac{1}{6}$  (xalpImr - xalmImr)
    (xalpImr2 + xalpImr * xalmImr + xalmImr2) ] ] ];
Write[hivout, ""];
Write[hivout, "DIFF1 (L)=",
  shd[fortconst[ (xlmImr - xlpImr)

```

Ausgabe weiterer Hilfsvariablen

```

      (xlmImr2 + xlmImr * xlpImr + xlpImr2) ] ] ] ;
Write[hivout, ""];
Write[hivout, "VECMD (L)=",
  shd[fortconst[(xlpHmd (xlmr - xlpImr)
    (xlmr2 + xlmr * xlpImr + xlpImr2) +
    xlmHmd (xlmImr - xlmr) (xlmImr2 +
      xlmImr * xlmr + xlmr2)) / ((xlmImr - xlpImr)
      (xlmImr2 + xlmImr * xlpImr + xlpImr2)) ] ] ] ;
Write[hivout, ""];
Write[hivout, "VECAMD (L)=",
  shd[fortconst[(xalpHmd (xalmr - xalpImr)
    (xalmr2 + xalmr * xalpImr + xalpImr2) +
    xalmHmd (xalmImr - xalmr)
    (xalmImr2 + xalmImr * xalmr + xalmr2)) /
    ((xalmImr - xalpImr) (xalmImr2 +
      xalmImr * xalpImr + xalpImr2)) ] ] ] ;
Write[hivout, ""];
Write[hivout, "ENDDO"];
Write[hivout, ""];
Write[hivout, "DO L=2,NPT"];
(* Schleife von 2 bis
   Maximalzahl (NPT, siehe SHD-Code) *)
Write[hivout, ""];
Write[hivout, "DIFF2 (L)=",
  shd[fortconst[(xlmImr - xlmr)
    (xlmImr2 + xlmImr * xlmr + xlmr2) ] ] ] ;
Write[hivout, ""];
Write[hivout, "ENDDO"];
Close[hivout];

```

```

hivout = OpenAppend["hivek_mathe.f90", PageWidth → 130];
(* Abschluss des Files *)
Write[hivout, ""];
Write[hivout, "end subroutine hivek_mathe"];
Close[hivout];

```

Aufstellen der Jacobi-Matrix!

Ausgabe der restlichen Hilfsvariablen, sowie Footer der Hilfsvariablendatei

```

m = Table[D[G[i], devanzahl[[j]]], {i, 1, ng}, {j, 1, 5*ng}];
(* Ableitung der Gleichungen nach allen Variablen,
   danach mathematische Vereinfachungen, zuletzt Aufstellen der Matrix *)

(*shd[fort[change[Simplify[D[G[4],devanzahl[[2]]]]]]]
   shd[fort[Simplify[change[D[G[4],devanzahl[[2]]]]]]]*)

```

Stuetzstellenindizes (notwendig fuer korrekte Berechnungen im SHD-Code):

```

IL ... L-2
IM ... L-1
IN ... L
IP ... L+1
IQ ... L+2

```

```
Ss[1] = "IL"; Ss[2] = "IM"; Ss[3] = "IN"; Ss[4] = "IP"; Ss[5] = "IQ";
```

Ausgabe in die zugehoerigen Files (Schleife ueber alle Ausgabefiles):

```

Do[
  If[Vb[i1] == "MR",
    outmr = OpenAppend[Fb[1], PageWidth → pagewidth];
    If[ToString[m[[i1, 5*(i2-1)+i3]]] ≠ "0",
      Write[outmr, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ")", "=",
        ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5*(i2-1)+i3]]]]]]];
      Write[outmr, ""];
    Close["mr.f90"];];
  If[Vb[i1] == "MM",
    outmm = OpenAppend[Fb[2], PageWidth → pagewidth];
    If[ToString[m[[i1, 5*(i2-1)+i3]]] ≠ "0",
      Write[outmm, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ")", "=",
        ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5*(i2-1)+i3]]]]]]];
      Write[outmm, ""];
    Close["mm.f90"];];
  If[Vb[i1] == "MD",
    outmd = OpenAppend[Fb[3], PageWidth → pagewidth];
    If[ToString[m[[i1, 5*(i2-1)+i3]]] ≠ "0",
      Write[outmd, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ")", "=",
        ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5*(i2-1)+i3]]]]]]];
      Write[outmd, ""];
    Close["md.f90"];];
  If[Vb[i1] == "MU",
    outmu = OpenAppend[Fb[4], PageWidth → pagewidth];
    If[ToString[m[[i1, 5*(i2-1)+i3]]] ≠ "0",
      Write[outmu, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ")", "=",
        ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5*(i2-1)+i3]]]]]]];
      Write[outmu, ""];
    Close["mu.f90"];];
  If[Vb[i1] == "ME",
    outme = OpenAppend[Fb[5], PageWidth → pagewidth];
    If[ToString[m[[i1, 5*(i2-1)+i3]]] ≠ "0",
      Write[outme, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ")", "=",
        ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5*(i2-1)+i3]]]]]]];
      Write[outme, ""];
    Close["me.f90"];];

```

Berechnung der Ableitungen und Ausgabe der Ergebnisse

```

If[Vb[i1] == "MJ",
  outmj = OpenAppend[Fb[6], PageWidth → pagewidth];
  If[ToString[m[[i1, 5 * (i2 - 1) + i3]]] ≠ "0",
    Write[outmj, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ") ", "=",
      ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5 * (i2 - 1) + i3]]]]]]];
    Write[outmj, ""];
  Close["mj.f90"];];
If[Vb[i1] == "MH",
  outmh = OpenAppend[Fb[7], PageWidth → pagewidth];
  If[ToString[m[[i1, 5 * (i2 - 1) + i3]]] ≠ "0",
    Write[outmh, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ") ", "=",
      ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5 * (i2 - 1) + i3]]]]]]];
    Write[outmh, ""];
  Close["mh.f90"];];
If[Vb[i1] == "MK",
  outmk = OpenAppend[Fb[8], PageWidth → pagewidth];
  If[ToString[m[[i1, 5 * (i2 - 1) + i3]]] ≠ "0",
    Write[outmk, "DG(", Vb[i1], ",", Vb[i2], ",", Ss[i3], ",", L, ") ", "=",
      ToUpperCase[shd[fortconst[Simplify[change[m[[i1, 5 * (i2 - 1) + i3]]]]]]];
    Write[outmk, ""];
  Close["mk.f90"];];
, {i1, 1, ng}, {i2, 1, ng}, {i3, 1, 5}];

```

Schliessen der Ausgabefiles:

```

Do[outeqs = OpenAppend[Fb[i], PageWidth → pagewidth];
  Write[outeqs, ""];
  Write[outeqs, ""];
  Write[outeqs, "end subroutine", StringTake[Sb[i], 7]];
Close[Fb[i]]; , {i, 1, ng}];

```

Loeschen der Variablenbelegung

Zum Abschluss sind die Hilfsvariablenbelegungen fuer den naechsten Rechengang zu loeschen, die Variablennamen sind die in den Gleichungen vorkommenden beziehungsweise, die in den Ersetzungsregeln an erster stelle stehenden.

```

Clear[dvolph, dvoll, vecmd, vecmdpI, vecmdmI,
  drdreip, drdreipI, drdreimI, differ1,
  differ2, differ3, mur2dlpI, mur2dl, dvolaph,
  dvolal, vecamd, mur2dmlmh, mur2dmlph];

```

restliche Ergebnisausgabe, Footer der Ausgabedatei sowie Löschen der Hilfsvariablenbelegung

Anhang D

ShdNom.m

Um die Variablenkonvention des SHD-Codes zu gewährleisten müssen die Variablennamen der in MATHEMATICA verwendeten Befehle umgewandelt werden. Da bestimmte Zeichen wie das Pluszeichen oder runde Klammern in *Matematica* nicht als Variablenbezeichnungen zugelassen sind, diese aber in FORTRAN erlaubt und im SHD-Code in Verwendung sind, ist es Aufgabe dieses Moduls, das im Notebook gelinkt ist, für die korrekte Ersetzung der Variablen zu sorgen. Für jede im Notebook verwendete Variable gibt es eine Entsprechung im SHD-Code, die hier aufgelistet sein muss. Besondere Achtsamkeit ist bei neu eingeführten Hilfsvariablen gegeben, da diese unbedingt hier eingetragen werden müssen.

```
BeginPackage["ShdNom`"]
```

```
shd::usage = "Transformation der internen Mathematicavariablen  
in die im Wiener SHD-Code verwendeten FORTRAN-Variablen,  
darueber hinaus Ersetzen von redundantem FORTRAN-Code wie **1"
```

```
(*  
Fuer die Ersetzungsregeln gibt es im Allgemeinen keine  
vorgeschriebene Reihenfolge. Auf zwei Dinge ist allerdings  
zu achten. Erstens, dass bei Ersetzungen von Variablen  
mit einer uebereinstimmenden Anzahl von Buchstaben, wie  
sadv und sadvpI, zuerst IMMER die LAENGERE Variable  
ersetzt wird. Zweitens sollten zuerst immer die Variablenname,  
dann der redundante FORTRAN-Code und zuletzt die Brueche  
ersetzt werden.  
)
```

```
Begin["Private`"]
```

```

nom1 = "xlmImr" -> "X(L-1,MR)";
nom2 = "xlmm" -> "X(L,MM)";
nom3 = "xlpImm" -> "X(L+1,MM)";
nom4 = "xlphmd" -> "X(L,MD)";
nom5 = "dvolvec" -> "DVOLVEC (L)";
nom6 = "xlpImr" -> "X(L+1,MR)";
nom7 = "dvol sca" -> "DVOLSCA (L)";
nom8 = "xlmr" -> "X(L,MR)";
nom9 = "xlpImm" -> "X(L+1,MM)";
nom10 = "xlmhmd" -> "X(L-1,MD)";
nom11 = "xlpImr" -> "X(L+1,MR)";
nom12 = "xlmIImr" -> "X(L-2,MR)";
nom13 = "xlpIImr" -> "X(L+2,MR)";
nom14 = "xalmImr" -> "XA(L-1,MR)";
nom15 = "xalmr" -> "XA(L,MR)";
nom16 = "xalpImr" -> "XA(L+1,MR)";

nom161 = "xalmhmd" -> "XA(L-1,MD)";
nom162 = "xalphmd" -> "XA(L,MD)";
nom163 = "xvecmdmI" -> "VECMD (L-1)";
nom164 = "dvolavec" -> "DVOLAVEC (L)";
nom165 = "xalmu" -> "XA(L,MU)";
nom166 = "xvecamd" -> "VECAMD (L)";

nom169 = "xvecmdpI" -> "VECMD (L+1)";
nom170 = "xvecmd" -> "VECMD (L)";
nom171 = "ur2dmlmh" -> "ur2dm (l-1)";
nom172 = "ur2dmlph" -> "ur2dm (l)";
nom173 = "sadvpI" -> "sadv (l+1)";
nom174 = "sadv" -> "sadv (l)";
nom175 = "xlphme" -> "X(L,ME)";
nom176 = "xalphme" -> "XA(L,ME)";
nom177 = "xalmImu" -> "XA(l-1,MU)";

nom116 = "xalmd" -> "XA(L,Md)";
nom117 = "xlmd" -> "X(L,MD)";
nom118 = "xlpImd" -> "X(l+1,md)";
nom119 = "xlpIImd" -> "X(l+2,md)";
nom120 = "xlmImd" -> "X(l-1,md)";
nom121 = "xlmIImd" -> "X(l-2,md)";

nom123 = "ur2dl" -> "ur2d (l)";
nom122 = "ur2dlpI" -> "ur2d (l+1)";

```

```
nom124 = "sadsph" -> "Sad (1)";
nom125 = "sadspIIh" -> "Sad (1+1)";
```

```
nom127 = "xalpImu" -> "XA(L+1,MU)";
nom128 = "xalmu" -> "XA(1,mu)";
```

```
nom17 = "xlpIIhmd" -> "X(L+1,MD)";
nom18 = "PI" -> "PI";
nom19 = "G" -> "GRAV";
nom20 = "alpha" -> "ALPHA";
nom21 = "beta" -> "BETA";
nom22 = "tst" -> "TST";
nom23 = "xvecmdmI" -> "VECMD(L-1)";
nom24 = "xvecmdpI" -> "VECMD(L+1)";
nom25 = "xvecmd" -> "VECMD(L)";
nom26 = "xlmU" -> "X(L,MU)";
nom27 = "xlpImu" -> "X(L+1,MU)";
nom28 = "xlmImu" -> "X(L-1,MU)";
nom29 = "dkubmI" -> "DDREI(L-1)";
nom30 = "dkubpI" -> "DDREI(L+1)";
nom31 = "dkub" -> "DR3(L)";
nom32 = "prgzph" -> "prgz (L)";
nom33 = "prgzmh" -> "prgz (L-1)";
nom34 = "xlmIIhmd" -> "X(L-2,MD)";
nom35 = "diff1" -> "DIFF1 (L)";
nom36 = "diff2" -> "DIFF2 (L)";
nom37 = "diff3" -> "DIFF3 (L)";
nom38 = "ddrei" -> "dr3 (1)";
nom39 = "dvolasca" -> "dvolasca (1)";
```

(* Variablenersetzungsregeln *)

```
num1 = "*z1 " -> "";
num2 = ")*z1" -> ")";
num3 = "z1*(" -> "(";
num4 = "12.566370614359172" -> "4.*PI";
num4 = "6.283185307179586" -> "2.*PI";
num4 = "3.141592653589793" -> "PI";
```

(* Ersetzungsregeln fuer redundanten FORTRAN-Code *)

```
bruch1 = "zed" -> "z13";
bruch2 = "zzd" -> "z23";
```

```

bruch3 = "zvd" -> "z43";
bruch4 = "zfv" -> "z52";
bruch5 = "zfd" -> "z53";
bruch6 = "zev" -> "z34";
bruch7 = "zes" -> "z16";

(* Ersetzungsregeln fuer spezielle PHYSKO-Variablen (Brueche) *)

nom[x_] := StringReplace[x, {nom1, nom2, nom3, nom4, nom5, nom6, nom7,
nom8, nom9, nom10, nom11, nom12, nom13, nom14, nom15, nom16, nom161,
nom162, nom163, nom164, nom165, nom166, nom169, nom170, nom171, nom172,
nom173, nom174, nom175, nom176, nom177, nom17, nom18, nom19, nom20, nom21,
nom22, nom23, nom24, nom25, nom26, nom27, nom28, nom29, nom30, nom31,
nom32, nom33, nom34, nom35, nom36, nom37, nom38, nom39, nom116, nom122,
nom123, nom124, nom125, nom127, nom128 }];

num[x_] := StringReplace[x, {num1, num2, num3, num4, num5, num6}];

bruch[x_] := StringReplace[x, {bruch1, bruch2, bruch3, bruch4,
bruch5, bruch6, bruch7}];

(* Ersetzungsfunktionen *)

shd[x_] := bruch[num[nom[x]]];

(* Transformationsfunktion *)

Print["Das Modul Shd.m wurde durchlaufen"];

(* Bestaetigung der Ausfuehrung des Moduls *)

End []

EndPackage []

```

Anhang E

Output des MATHEMATICA-Notebooks

Der Output des MATHEMATICA-Programms ist so gestaltet, dass im Fall der Maximalgleichungsanzahl acht (im reinen hydrodynamischen Fall fünf) Files für die Gleichungen und deren Ableitungen, sowie ein File für die Hilfsvektoren, die die anderen Files verwenden, erzeugt werden. Das Format der erzeugten Files entspricht dem Aufbau einer FORTRAN90/95-Subroutine. Dadurch ist es leicht möglich, diese Dateien in den SHD-Code einzubauen.

Die Namen der Files sind wie folgt:

- **hivek_mathe.f90** - File für die Hilfsvektoren
- **mr.f90** - Gittergleichung und Ableitungen
- **mm.f90** - Poisson-Gleichung und Ableitungen
- **md.f90** - Kontinuitätsgleichung und Ableitungen
- **mu.f90** - Geschwindigkeitsgleichung und Ableitungen
- **me.f90** - Energiegleichung und Ableitungen
- **mj.f90** - Strahlungsenergiedichtegleichung und Ableitungen
- **mh.f90** - Strahlungsflussgleichung und Ableitungen
- **mk.f90** - Strahlungsdruckgleichung und Ableitungen

Da der gesamte Output in Strings vom ASII-Format erfolgt, maskiert MATHEMATICA den gesamten Text mit Anführungszeichen ("), das Zeilenfortsetzungszeichen ist der Backslash (\). Um die Anführungszeichen zu eliminieren und das korrekte Zeilenfortsetzungszeichen für FORTRAN90/95 (&) einzusetzen

werden die Ausgabedateien nachbearbeitet. Mit dem Shellsript **reinigen** werde die entsprechenden Ersetzungen vorgenommen.

E.1 hivek_mathe.f90

```
subroutine hivek_mathe

!Hilfsvektoren fuer die Ableitungen der Gleichungen
!-----
!Ersetzung von Sonderzeichen:
!Backslash durch blank&
!Anfuehrungszeichen mit blank
!-----

!-----
!Header:
!Modell = 1
!Info: hydro, zz, ohne visko
!Datum: {2003, 10, 30, 21, 28, 35}
!-----

use alles
use PHYSKO

IMPLICIT NONE

INTEGER :: L

DO L=1,NPT

DRDREI (L)=(XA(L,MR)**2 + XA(L,MR)*X(L,MR) + &
X(L,MR)**2)*(X(L,MR) - XA(L,MR))

ENDDO

DO L=1,NPM

DVOLSCA (L)=(X(L,MR)**2 + X(L,MR)*X(L+1,MR) + &
X(L+1,MR)**2)*(X(L,MR) - X(L+1,MR))*z13

DVOLASCA (L)=(XA(L,MR)**2 + XA(L,MR)*XA(L+1,MR) + &
XA(L+1,MR)**2)*(XA(L,MR) - XA(L+1,MR))*z13

DIFF3 (L)=(X(L,MR)**2 + X(L,MR)*X(L+1,MR) + &
```

```

X(L+1,MR)**2)*(X(L,MR) - X(L+1,MR))

ENDDO

DO L=2,NPM

DVOLVEC (L)=(X(L-1,MR)**2 + X(L-1,MR)*X(L+1,MR) + &
X(L+1,MR)**2)*(X(L+1,MR) - X(L-1,MR))*z16

DVOLAVEC (L)=(XA(L-1,MR)**2 + XA(L-1,MR)*XA(L+1,MR) &
+ XA(L+1,MR)**2)*(XA(L+1,MR) - XA(L-1,MR))*z16

DIFF1 (L)=(X(L-1,MR)**2 + X(L-1,MR)*X(L+1,MR) + &
X(L+1,MR)**2)*(X(L-1,MR) - X(L+1,MR))

VECMD (L)=(X(L-1,MD)*(X(L-1,MR)**2 + &
X(L-1,MR)*X(L,MR) + X(L,MR)**2)*(X(L-1,MR) - X(L,MR)) &
+ X(L,MD)*(X(L,MR)**2 + X(L,MR)*X(L+1,MR) + &
X(L+1,MR)**2)*(X(L,MR) - X(L+1,MR)))/(X(L-1,MR)**2 + &
X(L-1,MR)*X(L+1,MR) + X(L+1,MR)**2)*(X(L-1,MR) - &
X(L+1,MR)))

VECAMD (L)=(XA(L-1,MD)*(XA(L-1,MR)**2 + &
XA(L-1,MR)*XA(L,MR) + XA(L,MR)**2)*(XA(L-1,MR) - &
XA(L,MR)) + XA(L,MD)*(XA(L,MR)**2 + &
XA(L,MR)*XA(L+1,MR) + XA(L+1,MR)**2)*(XA(L,MR) - &
XA(L+1,MR)))/(XA(L-1,MR)**2 + XA(L-1,MR)*XA(L+1,MR) + &
XA(L+1,MR)**2)*(XA(L-1,MR) - XA(L+1,MR)))

ENDDO

DO L=2,NPT

DIFF2 (L)=(X(L-1,MR)**2 + X(L-1,MR)*X(L,MR) + &
X(L,MR)**2)*(X(L-1,MR) - X(L,MR))

ENDDO

end subroutine hivek_mathe

```

E.2 mm.f90

```
subroutine EQMAS (L)

!SHD-Gleichungen und Ableitungen nach den Variablen
!-----
!Ersetzung von Sonderzeichen:
!Backslash durch blank&
!Anfuhrungszeichen mit blank
!-----

!-----
!Header:
!Modell = 1
!Info: hydro, zz, ohne visko
!Datum: {2004, 5, 14, 18, 29, 58}
!-----

use alles
use PHYSKO

G(MM,L)=X(L,MM) - DVOLSCA (L)*X(L,MD) - X(L+1,MM)

DG(MM,MR,IN,L)=-4*PI*X(L,MR)**2*X(L,MD)

DG(MM,MR,IP,L)=4*PI*X(L,MD)*X(L+1,MR)**2

DG(MM,MM,IN,L)=Z1

DG(MM,MM,IP,L)=-Z1

DG(MM,MD,IN,L)=- (DVOLSCA (L))

end subroutine EQMAS
```

E.3 md.f90

```
subroutine EQKONT (L)

!SHD-Gleichungen und Ableitungen nach den Variablen
!-----
!Ersetzung von Sonderzeichen:
!Backslash durch blank&
!Anfuhrungszeichen mit blank
!-----

!-----
!Header:
!Modell = 1
!Info: hydro, zz, ohne visko
!Datum: {2004, 5, 14, 18, 29, 58}
!-----

use alles
use PHYSKO

G(MD,L)=DVOLSCA (L)*X(L,MD) - DVOLASCA (L)*XA(L,MD) + 4*PI*(UR2D &
(L)*(SAD (L)*X(L,MD) + X(L-1,MD)*(Z1 - SAD (L))) - UR2D (L+1)*(SAD &
(L+1)*X(L+1,MD) + X(L,MD)*(Z1 - SAD (L+1))))

DG(MD,MR,IN,L)=4*PI*X(L,MR)*(X(L,MR)*X(L,MD) + &
(X(L-1,MD) + SAD (L)*X(L,MD) - SAD (L)*X(L-1,MD))*(-(X(L,MR)) + &
TST*Z2*ZZ*(XA(L,MU)*ZM + X(L,MU)*ZZ))

DG(MD,MR,IP,L)=4*PI*X(L+1,MR)*(-(X(L,MD)*X(L+1,MR)) - &
(X(L,MD) + SAD (L+1)*X(L+1,MD) - SAD (L+1)*X(L,MD))*(-(X(L+1,MR)) + &
TST*Z2*ZZ*(XA(L+1,MU)*ZM + X(L+1,MU)*ZZ))

DG(MD,MD,IM,L)=4*PI*(Z1 - SAD (L))*(-(DR3(L)*Z13) + &
TST*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ))

DG(MD,MD,IN,L)=DVOLSCA (L) + 4*PI*(UR2D (L+1)*(SAD &
(L+1) - Z1) - DR3(L)*SAD (L)*Z13 + SAD (L)*TST*(XA(L,MR)**2*ZM + &
X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ))

DG(MD,MD,IP,L)=-4*PI*SAD (L+1)*UR2D (L+1)
```

```

DG(MD,MU,IN,L)=4*PI*TST*(X(L-1,MD) + SAD (L)*X(L,MD) - &
SAD (L)*X(L-1,MD))*ZZ*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ)

```

```

DG(MD,MU,IP,L)=4*PI*TST*(X(L,MD)*(SAD (L+1) - Z1) - SAD &
(L+1)*X(L+1,MD))*ZZ*(XA(L+1,MR)**2*ZM + X(L+1,MR)**2*ZZ)

```

```

end subroutine EQKONT

```

E.4 mu.f90

```
subroutine EQMOT (L)

!SHD-Gleichungen und Ableitungen nach den Variablen
!-----
!Ersetzung von Sonderzeichen:
!Backslash durch blank&
!Anfuhrungszeichen mit blank
!-----

!-----
!Header:
!Modell = 1
!Info: hydro, zz, ohne visko
!Datum: {2004, 5, 14, 18, 29, 58}
!-----

use alles
use PHYSKO

G(MU,L)=(DVOLVEC (L)*GRAV*TST*X(L,MM)*VECMD (L))/X(L,MR)**2 + &
4*PI*TST*X(L,MR)**2*(PRGZ (L-1) - PRGZ (L)) + 4*PI*(DVOLVEC &
(L)*X(L,MU)*VECMD (L) - DVOLAVEC (L)*XA(L,MU)*VECAMD (L) + UR2DM &
(L-1)*(SADV (L)*X(L,MU)*VECMD (L) + X(L-1,MU)*VECMD (L-1)*(Z1 - SADV (L))) &
- UR2DM (L)*(SADV (L+1)*X(L+1,MU)*VECMD (L+1) + X(L,MU)*VECMD (L)*(Z1 - &
SADV (L+1))))

DG(MU,MR,IM,L)=2.PI*X(L-1,MR)*((GRAV*TST*X&
(L-1,MD)*X(L-1,MR)*X(L,MM))/X(L,MR)**2 + 4*PI*X(L-1,MD)*X(L-1,MR)*X(L,MU) &
+ (SADV (L)*X(L,MU)*VECMD (L) + X(L-1,MU)*VECMD (L-1) - SADV &
(L)*X(L-1,MU)*VECMD (L-1))*(-(X(L-1,MR)) + TST*Z2*ZZ*(XA(L-1,MU)*ZM + &
X(L-1,MU)*ZZ)) + (SADV (L)*X(L-1,MR)*X(L,MU)*(DIFF1 (L)*X(L-1,MD) - DIFF2 &
(L)*X(L-1,MD) - DIFF3 (L)*X(L,MD))*(XA(L-1,MR)**3 - DR3(L) - X(L-1,MR)**3+ &
TST*Z3*(XA(L-1,MR)**2*ZM + X(L-1,MR)**2*ZZ)*(XA(L-1,MU)*ZM + X(L-1,MU)*ZZ) &
+ TST*Z3*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + &
X(L,MU)*ZZ)))/DIFF1 (L)**2 + (X(L-1,MR)*X(L,MU)*(SADV (L+1) - Z1)*(DIFF1 &
(L)*X(L-1,MD) - DIFF2 (L)*X(L-1,MD) - DIFF3 (L)*X(L,MD))*(-(DR3(L)) + &
Z3*(UR2D (L+1) + TST*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + &
X(L,MU)*ZZ))))/DIFF1 (L)**2)

DG(MU,MR,IN,L)=PI*(12.*TST*X(L,MR)*(PRGZ &
```

$(L-1) - \text{PRGZ}(L)) - (\text{GRAV} * \text{TST} * X(L, \text{MM}) * (\text{DIFF2}(L) * X(L-1, \text{MD}) + \text{DIFF3}(L) * X(L, \text{MD})) * Z2) / X(L, \text{MR}) ** 3 - \text{GRAV} * \text{TST} * X(L, \text{MM}) * (X(L-1, \text{MD}) - X(L, \text{MD})) * Z3 + \&$
 $X(L, \text{MR}) * Z3 * (4 * \text{PI} * X(L, \text{MR}) * X(L, \text{MU}) * (X(L, \text{MD}) - X(L-1, \text{MD})) + (\text{SADV} \&$
 $(L) * X(L, \text{MU}) * \text{VECMD}(L) + X(L-1, \text{MU}) * \text{VECMD}(L-1) - \text{SADV}(L) * X(L-1, \text{MU}) * \text{VECMD} \&$
 $(L-1)) * (-X(L, \text{MR})) + \text{TST} * Z2 * Z3 * (X(A(L, \text{MU}) * ZM + X(L, \text{MU}) * ZZ)) - (\text{SADV} \&$
 $(L+1) * X(L+1, \text{MU}) * \text{VECMD}(L+1) + X(L, \text{MU}) * (\text{VECMD}(L) - \text{SADV}(L+1) * \text{VECMD} \&$
 $(L))) * (-X(L, \text{MR})) + \text{TST} * Z2 * Z3 * (X(A(L, \text{MU}) * ZM + X(L, \text{MU}) * ZZ)) + (\text{SADV} \&$
 $(L) * X(L, \text{MR}) * X(L, \text{MU}) * (X(L, \text{MD}) - X(L-1, \text{MD})) * (X(A(L-1, \text{MR}) ** 3 - \text{DR3}(L) - \&$
 $X(L-1, \text{MR}) ** 3 + \text{TST} * Z3 * (X(A(L-1, \text{MR}) ** 2 * ZM + X(L-1, \text{MR}) ** 2 * ZZ) * (X(A(L-1, \text{MU}) * ZM + \&$
 $X(L-1, \text{MU}) * ZZ) + \text{TST} * Z3 * (X(A(L, \text{MR}) ** 2 * ZM + X(L, \text{MR}) ** 2 * ZZ) * (X(A(L, \text{MU}) * ZM + \&$
 $X(L, \text{MU}) * ZZ))) / \text{DIFF1}(L) - (X(L, \text{MR}) * X(L, \text{MU}) * (\text{SADV}(L+1) - Z1) * (X(L-1, \text{MD}) - \&$
 $X(L, \text{MD})) * (-\text{DR3}(L)) + Z3 * (\text{UR2D}(L+1) + \text{TST} * (X(A(L, \text{MR}) ** 2 * ZM + \&$
 $X(L, \text{MR}) ** 2 * ZZ) * (X(A(L, \text{MU}) * ZM + X(L, \text{MU}) * ZZ))) / \text{DIFF1}(L))) * Z23$

$\text{DG}(\text{MU}, \text{MR}, \text{IP}, L) = 2 * \text{PI} * X(L+1, \text{MR}) * (-4 * \text{PI} * X(L, \text{M} \&$
 $\text{U}) * X(L, \text{MD}) * X(L+1, \text{MR}) - (\text{GRAV} * \text{TST} * X(L, \text{MM}) * X(L, \text{MD}) * X(L+1, \text{MR})) / X(L, \text{MR}) ** 2 + \&$
 $(\text{SADV}(L) * X(L, \text{MU}) * X(L+1, \text{MR}) * (\text{DIFF2}(L) * X(L-1, \text{MD}) + X(L, \text{MD}) * (\text{DIFF3}(L) - \&$
 $\text{DIFF1}(L))) * (X(A(L-1, \text{MR}) ** 3 - \text{DR3}(L) - X(L-1, \text{MR}) ** 3 + \&$
 $\text{TST} * Z3 * (X(A(L-1, \text{MR}) ** 2 * ZM + X(L-1, \text{MR}) ** 2 * ZZ) * (X(A(L-1, \text{MU}) * ZM + X(L-1, \text{MU}) * ZZ) \&$
 $+ \text{TST} * Z3 * (X(A(L, \text{MR}) ** 2 * ZM + X(L, \text{MR}) ** 2 * ZZ) * (X(A(L, \text{MU}) * ZM + \&$
 $X(L, \text{MU}) * ZZ))) / \text{DIFF1}(L) ** 2 - (\text{SADV}(L+1) * X(L+1, \text{MU}) * \text{VECMD}(L+1) + \&$
 $X(L, \text{MU}) * (\text{VECMD}(L) - \text{SADV}(L+1) * \text{VECMD}(L))) * (-X(L+1, \text{MR})) + \&$
 $\text{TST} * Z2 * Z3 * (X(A(L+1, \text{MU}) * ZM + X(L+1, \text{MU}) * ZZ)) + (X(L, \text{MU}) * X(L+1, \text{MR}) * (\text{SADV}(L+1) \&$
 $- Z1) * (\text{DIFF2}(L) * X(L-1, \text{MD}) + X(L, \text{MD}) * (\text{DIFF3}(L) - \text{DIFF1}(L))) * (-\text{DR3}(L)) + \&$
 $Z3 * (\text{UR2D}(L+1) + \text{TST} * (X(A(L, \text{MR}) ** 2 * ZM + X(L, \text{MR}) ** 2 * ZZ) * (X(A(L, \text{MU}) * ZM + \&$
 $X(L, \text{MU}) * ZZ))) / \text{DIFF1}(L) ** 2$

$\text{DG}(\text{MU}, \text{MM}, \text{IN}, L) = (\text{PI} * \text{GRAV} * \text{TST} * (\text{DIFF2} \&$
 $(L) * X(L-1, \text{MD}) + \text{DIFF3}(L) * X(L, \text{MD})) * Z23) / X(L, \text{MR}) ** 2$

$\text{DG}(\text{MU}, \text{MD}, \text{IM}, L) = \text{PI} * \text{DIFF2} \&$
 $(L) * ((\text{GRAV} * \text{TST} * X(L, \text{MM})) / X(L, \text{MR}) ** 2 + X(L, \text{MU}) * (4 * \text{PI} + (\text{SADV} \&$
 $(L) * (X(A(L-1, \text{MR}) ** 3 - \text{DR3}(L) - X(L-1, \text{MR}) ** 3 + \text{TST} * Z3 * (X(A(L-1, \text{MR}) ** 2 * ZM + \&$
 $X(L-1, \text{MR}) ** 2 * ZZ) * (X(A(L-1, \text{MU}) * ZM + X(L-1, \text{MU}) * ZZ) + \text{TST} * Z3 * (X(A(L, \text{MR}) ** 2 * ZM + \&$
 $X(L, \text{MR}) ** 2 * ZZ) * (X(A(L, \text{MU}) * ZM + X(L, \text{MU}) * ZZ))) / \text{DIFF1}(L) - ((\text{SADV}(L+1) - \&$
 $Z1) * (\text{DR3}(L) - Z3 * (\text{UR2D}(L+1) + \text{TST} * (X(A(L, \text{MR}) ** 2 * ZM + \&$
 $X(L, \text{MR}) ** 2 * ZZ) * (X(A(L, \text{MU}) * ZM + X(L, \text{MU}) * ZZ))) / \text{DIFF1}(L))) * Z23$

$\text{DG}(\text{MU}, \text{MD}, \text{IN}, L) = \text{PI} * \text{DIFF3} \&$
 $(L) * ((\text{GRAV} * \text{TST} * X(L, \text{MM})) / X(L, \text{MR}) ** 2 + X(L, \text{MU}) * (4 * \text{PI} + (\text{SADV} \&$
 $(L) * (X(A(L-1, \text{MR}) ** 3 - \text{DR3}(L) - X(L-1, \text{MR}) ** 3 + \text{TST} * Z3 * (X(A(L-1, \text{MR}) ** 2 * ZM + \&$
 $X(L-1, \text{MR}) ** 2 * ZZ) * (X(A(L-1, \text{MU}) * ZM + X(L-1, \text{MU}) * ZZ) + \text{TST} * Z3 * (X(A(L, \text{MR}) ** 2 * ZM + \&$
 $X(L, \text{MR}) ** 2 * ZZ) * (X(A(L, \text{MU}) * ZM + X(L, \text{MU}) * ZZ))) / \text{DIFF1}(L) - ((\text{SADV}(L+1) - \&$

```

Z1)*(DR3(L) - Z3*(UR2D (L+1) + TST*(XA(L,MR)**2*ZM + &
X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ))))/DIFF1 (L)))*Z23

```

```

DG(MU,MU,IM,L)=2.PI*(TST*(SADV &
(L)*X(L,MU)*VECMD (L) + X(L-1,MU)*VECMD (L-1) - SADV (L)*X(L-1,MU)*VECMD &
(L-1))*ZZ*(XA(L-1,MR)**2*ZM + X(L-1,MR)**2*ZZ) + VECMD (L-1)*(Z1 - SADV &
(L))*Z13*(XA(L-1,MR)**3 - DR3(L) - X(L-1,MR)**3+ TST*Z3*(XA(L-1,MR)**2*ZM &
+ X(L-1,MR)**2*ZZ)*(XA(L-1,MU)*ZM + X(L-1,MU)*ZZ) + TST*Z3*(XA(L,MR)**2*ZM &
+ X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ)))

```

```

DG(MU,MU,IN,L)=PI*(4*PI*(DIFF2 &
(L)*X(L-1,MD) + DIFF3 (L)*X(L,MD)) + TST*(SADV (L)*X(L,MU)*VECMD (L) + &
X(L-1,MU)*VECMD (L-1) - SADV (L)*X(L-1,MU)*VECMD &
(L-1))*Z3*ZZ*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ) - TST*(SADV &
(L+1)*X(L+1,MU)*VECMD (L+1) + X(L,MU)*(VECMD (L) - SADV (L+1)*VECMD &
(L)))*Z3*ZZ*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ) - VECMD (L)*(Z1 - SADV &
(L+1))*Z3*(UR2D (L+1) - DR3(L)*Z13 + TST*(XA(L,MR)**2*ZM + &
X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ)) + SADV (L)*VECMD &
(L)*(XA(L-1,MR)**3 - DR3(L) - X(L-1,MR)**3+ TST*Z3*(XA(L-1,MR)**2*ZM + &
X(L-1,MR)**2*ZZ)*(XA(L-1,MU)*ZM + X(L-1,MU)*ZZ) + TST*Z3*(XA(L,MR)**2*ZM + &
X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ)))*Z23

```

```

DG(MU,MU,IP,L)=4*PI*(-0.5*TST*(SADV &
(L+1)*X(L+1,MU)*VECMD (L+1) + X(L,MU)*(VECMD (L) - SADV (L+1)*VECMD &
(L)))*ZZ*(XA(L+1,MR)**2*ZM + X(L+1,MR)**2*ZZ) - 0.5*SADV (L+1)*VECMD &
(L+1)*(UR2D (L+1) - DR3(L)*Z13 + TST*(XA(L,MR)**2*ZM + &
X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ)))

```

```

end subroutine EQMOT

```

E.5 me.f90

```
subroutine EQENE (L)

!SHD-Gleichungen und Ableitungen nach den Variablen
!-----
!Ersetzung von Sonderzeichen:
!Backslash durch blank&
!Anfuhrungszeichen mit blank
!-----

!-----
!Header:
!Modell = 1
!Info: hydro, zz, ohne visko
!Datum: {2004, 5, 14, 18, 29, 58}
!-----

use alles
use PHYSKO

G(ME,L)=DVOLSCA (L)*X(L,MD)*X(L,ME) - DVOLASCA &
(L)*XA(L,MD)*XA(L,ME) + PRGZ (L)*TST*(X(L,MR)**2*X(L,MU) - &
X(L+1,MR)**2*X(L+1,MU)) + UR2D (L)*(SAD (L)*X(L,MD)*X(L,ME) + &
X(L-1,MD)*XLMHME*(Z1 - SAD (L))) - UR2D (L+1)*(SAD &
(L+1)*X(L+1,MD)*XLPPIIHME + X(L,MD)*X(L,ME)*(Z1 - SAD (L+1)))

DG(ME,MR,IN,L)=X(L,MR)*(4*PI*X(L,MR)*X(L,MD)*X(L,ME) + &
PRGZ (L)*TST*X(L,MU)*Z2 + (SAD (L)*X(L,MD)*X(L,ME) + X(L-1,MD)*(XLMHME - &
SAD (L)*XLMHME*Z1))*(-(X(L,MR)) + TST*Z2*ZZ*(XA(L,MU)*ZM + X(L,MU)*ZZ))

DG(ME,MR,IP,L)=X(L+1,MR)*(-4*PI*X(L,MD)*X(L,ME)*X(L+1,M&
R) - PRGZ (L)*TST*X(L+1,MU)*Z2 - (SAD (L+1)*X(L+1,MD)*XLPPIIHME + &
X(L,MD)*(X(L,ME) - SAD (L+1)*X(L,ME)))*(-(X(L+1,MR)) + &
TST*Z2*ZZ*(XA(L+1,MU)*ZM + X(L+1,MU)*ZZ))

DG(ME,MD,IM,L)=XLMHME*(Z1 - SAD (L))*(-(DR3(L)*Z13) + &
TST*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ))

DG(ME,MD,IN,L)=X(L,ME)*(DVOLSCA (L) + UR2D (L+1)*(SAD &
(L+1) - Z1) + SAD (L)*(-(DR3(L)*Z13) + TST*(XA(L,MR)**2*ZM + &
X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ)))
```

DG(ME,MD,IP,L)=- (SAD (L+1)*UR2D (L+1)*XLPIIIHME*Z1)

DG(ME,MU,IN,L)=PRGZ (L)*TST*X(L,MR)**2 + TST*(SAD &
(L)*X(L,MD)*X(L,ME) + X(L-1,MD)*(XLMHME - SAD &
(L)*XLMHME*Z1))*ZZ*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ)

DG(ME,MU,IP,L)=- (PRGZ (L)*TST*X(L+1,MR)**2*Z1) + &
TST*(X(L,MD)*X(L,ME)*(SAD (L+1) - Z1) - SAD &
(L+1)*X(L+1,MD)*XLPIIIHME*Z1)*ZZ*(XA(L+1,MR)**2*ZM + X(L+1,MR)**2*ZZ)

DG(ME,ME,IM,L)=X(L-1,MD)*(Z1 - SAD (L))*(-(DR3(L)*Z13) &
+ TST*(XA(L,MR)**2*ZM + X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ))

DG(ME,ME,IN,L)=X(L,MD)*(DVOLSCA (L) + UR2D (L+1)*(SAD &
(L+1) - Z1) + SAD (L))*(-(DR3(L)*Z13) + TST*(XA(L,MR)**2*ZM + &
X(L,MR)**2*ZZ)*(XA(L,MU)*ZM + X(L,MU)*ZZ))

DG(ME,ME,IP,L)=- (SAD (L+1)*UR2D (L+1)*X(L+1,MD))

end subroutine EQENE

Anhang F

Naturkonstanten und Umrechnungsfaktoren

Tabelle F.1: astronomische und physikalische Konstanten

Bezeichnung	Symbol	Wert	Dimension	Fehler
Astronomische Einheit	AU	$1.495\,978\,706\,60 \cdot 10^{11}$	m	$2.0 \cdot 10^{-1}$
Avogadro-Zahl	N_A	$6.022\,141\,99 \cdot 10^{23}$	mol^{-1}	$4.7 \cdot 10^{16}$
Boltzmann-Konstante	k	$1.380\,6503 \cdot 10^{-23}$	J K^{-1}	$2.4 \cdot 10^{-29}$
Elektronenmasse	m_e	$9.109\,381\,88 \cdot 10^{-31}$	kg	$7.2 \cdot 10^{-38}$
Elementarladung	e	$1.602\,176\,462 \cdot 10^{-19}$	C	$6.3 \cdot 10^{-27}$
Gaskonstante	$\mathfrak{R}$	8.314 472	$\text{J K}^{-1} \text{mol}^{-1}$	$1.5 \cdot 10^{-5}$
Gravitationskonstante	G	$6.672\,598\,5 \cdot 10^{-11}$	$\text{N m}^2 \text{kg}^{-2}$	$1.28 \cdot 10^{-4}$
Jahr (siderisch)	a_{sid}	$3.155\,814\,98 \cdot 10^7$	s	exakt
Jahr (tropisch)	a_{trop}	$3.155\,692\,52 \cdot 10^7$	s	exakt
Lichtgeschwindigkeit	c	$2.997\,924\,58 \cdot 10^8$	m s^{-1}	exakt
mittlere Sonnendichte	$\bar{\rho}_{\odot}$	$1.408\,4969 \cdot 10^3$	kg m^{-3}	—
Parsec	pc	$3.085\,677\,5807 \cdot 10^{16}$	m	$4.0 \cdot 10^5$
Planck'sches Wirkungsquantum	h	$6.626\,068\,76 \cdot 10^{-34}$	J s	$5.2 \cdot 10^{-41}$
	$\hbar$	$1.054\,571\,596 \cdot 10^{-34}$	J s	$8.2 \cdot 10^{-42}$
Protonenmasse	m_p	$1.672\,621\,58 \cdot 10^{-27}$	kg	$1.3 \cdot 10^{-34}$
Sonnenleuchtkraft	$L_{\odot}$	$3.846 \cdot 10^{26}$	W	$8.0 \cdot 10^{23}$
Sonnenmasse	$M_{\odot}$	$1.9889 \cdot 10^{30}$	kg	$3.0 \cdot 10^{25}$
Sonnenradius	$R_{\odot}$	$6.9598 \cdot 10^8$	m	—
Stefan-Boltzmann-Konstante	σ	$5.670\,400 \cdot 10^{-8}$	$\text{W m}^{-2} \text{K}^{-4}$	$4.0 \cdot 10^{-13}$

In dieser Tabelle sind alle Konstanten mit ihren Symbolen, ihren Werten und zugehörigen relativen Fehlern angegeben (siehe [2], [25] und [40]). — bezeichnet, dass kein relativer Fehler in der Literatur gefunden wurde.

Tabelle F.2: Umrechnungsfaktoren cgs- auf SI-Einheiten

Bezeichnung	cgs-Einheit	SI-Einheit
Druck	1 dyn cm ⁻²	10 ⁻² N m ⁻²
Energie	1 erg	10 ⁻⁷ J
Energie/Masse	1 erg g ⁻¹	10 ⁻⁴ J kg ⁻¹
Energie/Volumen	1 erg cm ⁻³	10 ⁻¹ J m ⁻³
Geschwindigkeit	1 cm s ⁻¹	10 ⁻² m s ⁻¹
Kraft	1 dyn	10 ⁻⁵ N
Länge	1 cm	10 ⁻² m
Leuchtkraft	1 erg s ⁻¹	10 ⁻⁷ J s ⁻¹
Masse	1 g	10 ⁻³ kg
Opazität	1 cm ² g ⁻¹	10 ⁻² m ² kg ⁻¹

In dieser Tabelle sind die Umrechnungsfaktoren vom cgs-System ins SI-System angegeben (siehe [8]).

Tabelle F.3: Umrechnungsfaktoren SI- auf cgs-Einheiten

Bezeichnung	SI-Einheit	cgs-Einheit
Druck	1 N m ⁻²	10 ² dyn cm ⁻²
Energie	1 J	10 ⁷ erg
Energie/Masse	1 J kg ⁻¹	10 ⁴ erg g ⁻¹
Energie/Volumen	1 J m ⁻³	10 ¹ erg cm ⁻³
Geschwindigkeit	1 m s ⁻¹	10 ² cm s ⁻¹
Kraft	1 N	10 ⁵ dyn
Länge	1 m	10 ² cm
Leuchtkraft	1 J s ⁻¹	10 ⁷ erg s ⁻¹
Masse	1 kg	10 ³ g
Opazität	1 m ² kg ⁻¹	10 ² cm ² g ⁻¹

In dieser Tabelle sind die Umrechnungsfaktoren vom SI-System ins cgs-System angegeben (siehe [8]).

Sachregister

Sonderzeichen	
;	88
””	88
\	88
0. Moment der Strahlungstransportgleichung	10, 23 ff., 34, 49
1. Moment der Strahlungstransportgleichung	11, 23 ff., 34, 49
A	
Ableitungen in Mathematica	61
Adaptive Gitter	31
Adaptives Transporttheorem	79
Advektion	80
Allgemeines	50
Anfangsbedingungen	80
Aufbau der Diplomarbeit	4
Ausbau und Erweiterungen	73
Ausblick	73
automatische Vereinfachungen	62
B	
Begin[]	88
BeginPackage[]	88
Beschreibung der Anhänge	78
besondere Form der Jacobimatrix	53
Bewegungsgleichung	9, 21, 23 ff., 34, 49, 68, 80
bruch[]	88
C	
C.G.J.Jacobi	1
change[]	88
Clear[]	89
Close[]	89
Courant-Friedrichs-Levy (CFL) - Bedingung	80
D	
D[]	89
Danksagungen	138
Date[]	89
Datenausgabe	63
De Determinantibus functionalibus	3
Definition der Operatoren	17
Diffusion	80
Diskretisierung	35, 81
Diskretisierung einer skalaren Gleichung - Energiegleichung	47
Diskretisierung einer vektoriellen Gleichung - Bewegungsgleichungsgleichung	45
Diskretisierungsoperatoren	38
Diskretisierungsregeln	37
Do[]	89
Donor-Cell	81
E	
Eddington Approximation	81
Eddington-Näherung	11
Eigenschaften von Mathematica	58
Einführung	3
Einleitung	3
End[]	89
EndPackage[]	90
Energiegleichung	10, 21, 23 ff., 34, 49, 69, 81

erstes Moment	81	Klasse der Strahlungshydrodynamik-	
Euler-Beschreibung	82	gleichungen	36
explizit	82	konservativ	83
F		Kontinuitätsgleichung	9, 49
False	90	Kontinuitätsgleichung..	20, 23 ff., 34,
Finite-Volumen Diskretisierung...	82	83	
fortconst[]	90	künstliche Viskosität	83
FortranForm[]	90	L	
G		Lösung des SHD-Systems	50
gamma	15	Lösungsverfahren	50
General	90	Lagrange-Beschreibung	84
Gittergleichung	82	Latex-Literatur	viii
Gleichungen der Materien	9	Lebenslauf	139 ff.
Gleichungen in Mathematica	59	Literaturverzeichnis	133–137
Gleichungen in Mathematica-		M	
Notation	68	Mathematica-Befehle	87
H		Mathematica-Notebook	56
Henyei-Verfahren	82	Mathematik der SHD	16
Herleitungen der SHD-Gleichungen	8	mathematisch-nummerische Kon-	
hivek_mathe.f90	117 f.	struktionen	31
I		mathematische Definitionen	16
If[]	90	md.f90	120 f.
implizit	83	me.f90	125 f.
Index strahlungshydrodynamischer		mm.f90	119
und numerischer Begriffe		Modul ShdNom.m	71, 111–114
79		Motivation	5
J		mu.f90	122 ff.
Jacobi-Matrix	53, 83	Muster und Blanks	60
K		N	
Künstliche Viskosität	25	N[]	90
Künstliche Viskosität in der Bewe-		Nachbearbeitung	63
gungsgleichung	28	Newton-Raphson Iteration	84
künstliche Viskosität in der Energie-		nom???	90
gleichung	30	Notebook dev_8.2.nb	95
Kappa-Mechanismus	83	nulltes Moment	84
		num?	91
		O	
		Off[]	91

Opazität 13, 84
 OpenAppend[] 91
 OpenWrite[] 91
 Output 115–126

P

Pagewidth 91
 persönliche Anmerkungen 6
 Physik der SHD 7
 Planck-Mittel 84
 Poisson-Gleichung 85
 Poissongleichung 10, 23 ff., 34, 49
 Prolog 1

R

Randbedingungen 85
 räumliche Glättung 85
 Räumliche und zeitliche Glättung 32
 Reinigen 71
 Relativgeschwindigkeit 85
 Reynolds Transport Theorem 19
 Rosseland-Mittel 85

S

Setdirectory[] 91
 SetOptions[] 92
 SHD-Gleichungen in Integralform 33
 SHD-Gleichungen in unterschiedli-
 chen Bezugssystemen 22
 SHD-Gleichungen mit partieller
 Zeitableitung (Euler) 24
 SHD-Gleichungen mit partieller
 Zeitableitung (Gittersystem)
 23
 SHD-Gleichungen mit totaler Zeita-
 bleitung (Lagrange) 23
 shd[] 92
 Simplify[] 92
 Steuerparameter 66
 Strahlungs-Gleichungen 10
 Strahlungsenergiegleichung 10
 Strahlungsflussgleichung 11

StringJoin[] 92
 StringReplace[] 92
 StringTake[] 92
 Struktur des Notebooks 64
 symbolische Notation der Diskretise-
 rung 35

T

Tabelle der Naturkonstanten 127
 Table[] 93
 TableForm[] 93
 TimeConstraint 93
 ToString[] 93

U

Umrechnungstabellen cgs/SI-
 Einheiten 127
 UpperCase[] 93

V

Vereinfachungen der SHD-Physik .. 7
 Volumens- und Oberflächenterme in
 der Diskretisierung 40
 vor[] 93
 Vorgehensweise bei der Noteboker-
 stellung 57

W

Wechselwirkung zwischen Strahlung
 und Materie 12
 Weiterführung des Notebooks 75
 Write[] 93

Z

zeitliche Glättung 86
 zeitliche Zentrierung 86
 Zeitliche Zentrierung 32
 Zusammenfassung (abstract) ix
 Zustandsgleichung 14
 Zweidimensionale SHD 75

Literaturverzeichnis

- [1] Abell, Martha L.; Braselton, James P., 1997, *Differential Equations with Mathematica*, Academic Press
- [2] Astrophysical Constants, 2000, Hypertext:
<http://pdg.lbl.gov/2000/astrorppbook.pdf> (01 05 04)
- [3] Alexaner, D.R.; Fergusen, J.W., 1994, *Astropysical Journal* **437**, 879
- [4] Berger, C.; Lang, C., 2001, *Kurze Einführung in Mathematica*,
Hypertext: <http://physik.uni-graz.at/~cbl/teaching/mmvl/EinfMath/EinfMath.pdf> (01 05 04)
- [5] Bronstein, I.; Semendjajew, K., 2000, *Taschenbuch der Mathematik*, Verlag Harri Deutsch
- [6] Castor, J.I., 1972, *Astrophysical Journal* **178**, 779
- [7] Clayton, Donald D., 1983, *Principles of Stellar Evolution and Nucleosynthesis*, The University of Chicago Press
- [8] Conversion factors for energy equivalents, 2000,
Hypertext: <http://physics.nist.gov/cuu/Constants/energy.html> (01 05 04)
- [9] Courant, R., Friedrichs, K., Lewy, H., 1928, *Mathematische Annalen* **100**, 32
- [10] Cox, J.P., 1968, *Principles of Stellar Structure, Vol I Physical Principles*, Gordon and breach Science Publishers
- [11] Cox, J.P., 1968, *Principles of Stellar Structure, Vol II Applications to Stars*, Gordon and breach Science Publishers
- [12] Deutschsprachige Mathematica User Group, Hypertext:
<http://www.mathematica.ch> (01 05 04)

- [13] Dorfi, E.A.; Drury, L.O., 1986, *Simple Adaptive Grids for 1-D Initial Value Problems*, Journal of Computational Physics, **69**, 175-195
- [14] Dorfi, E.A., et al., 2004, *Towards a more consistent discretization scheme for adaptive implicit RHD computations*, in Vorbereitung
- [15] Dorfi E.; Feuchtinger M., 1991, *Nonlinear stellar pulsations. I - Numerical methods, basic physics, initial models and first results*, Astronomy and Astrophysics, **249**, 417
- [16] Dorfi E.; 1998, *Computational Methods for Astrophysical Fluid Flows - Radiation Hydrodynamics: Numerical Aspects and Applications*, Saas-Fee Advanced Course 27, Springer
- [17] Dorn, Th., 1998, *Numerische Mathematik, interaktiv - Nach einer Idee von Christian Überhuber*, Seminar an der Technischen Universität Wien
Hypertext: <http://www.dorn.org/uni/sls/index.html> (01 05 04)
- [18] Jacobi, C.G.J., 1841, *De Determinantibus functionalibus*, Crelle-Journal, **22**, Heft 4 IV, 319
- [19] Feuchtinger, M., 1989, *Strahlungshydrodynamische Modellrechnungen für Sterne am asymptotischen Riesenast*, Diplomarbeit, Universität Wien
- [20] Feuchtinger, M., 1991, *Zur Pulsation der Cepheiden*, Dissertation, Universität Wien
- [21] Feuchtinger M.; Dorfi E., 1994, *Nonlinear pulsation. 2: Radiative models of RR-Lyrae stars*, Astronomy and Astrophysics **291**, 209
- [22] Feuchtinger M.; Dorfi E., 1996, *Nonlinear stellar pulsation: III. Fourier decomposition of theoretical RRab models*, Astronomy and Astrophysics **306**, 837
- [23] Feuchtinger M.; Dorfi E.; Höfner S., 1993, *Radiation hydrodynamics in atmospheres of long-period variables*, Astronomy and Astrophysics **273**, 513
- [24] Fortran routines: OPAL opacities,
Hypertext: <http://www.krl.caltech.edu/~aib/kappa.html> (01 05 04)
- [25] Fundamental Physical Constants from NIST, 1998, Hypertext: <http://physics.nist.gov/cuu/Constants/index.html/> (01 05 04)

- [26] Gray, John W., 1994, *Mastering Mathematica*, AP Professional
- [27] Hieb, D., *A Mathematica Primer*, 1995, Hypertext:
<http://www.urz.uni-heidelberg.de/Dokumentation/CompAlgebra/mathematica-MathPrimer.ps> (01 05 04)
- [28] Höfner, S., 1994 *Dust Driven Mass Loss from Long Period Variables*, Dissertation, Universität Wien
- [29] Kippenhahn, R.; Weigert, A., 1990, *Stellar Structure and Evolution*, Springer, Berlin
- [30] Landau L.D., Lifschitz E.M., 1981, *Lehrbuch der Theoretischen Physik VI, Hydrodynamik*, Akademischer Verlag, Berlin
- [31] Leveque, R.J.; Mihalas, D.; Dorfi, E.; Müller, E., 1997, *Computational Methods for Astrophysical Fluid Flow*, Saas-Fee Advanced Course **27**, Springer
- [32] Lorenz, R., 2003, *Einführung in das funktionale und symbolische Programmieren mit Mathematica*, Vorlesung an der Universität Wien
- [33] Lucy, L., 1971, *Astrophysical Journal* **163**, 95
- [34] Lucy, L., 1976, *Astrophysical Journal* **205**, 482
- [35] Maeder, Roman, 1991, *Programming in Mathematica*, Addison-Wesley Publishing Company
- [36] Mihalas, D.; Mihalas, B., 1984, *Foundations of Radiation Hydrodynamics*, Oxford University Press
- [37] Opacity Project, 2004, Hypertext: <http://vizier.u-strasb.fr/OP.html> (08 06 04)
- [38] OPAL OPACITY TABLES,
Hypertext: <http://www-phys.llnl.gov/Research/OPAL/existing.html> (08 06 04)
- [39] Ostwald's Klassiker der exakten Wissenschaft, 1896, **78**, *Ueber die Funktionaldeterminanten*, Leipzig, deutsche Übersetzung von [18], 1841
- [40] Physical Constants and Astronomical Data, 1996,
Hypertext: <http://www.astro.wisc.edu/~dolan/constants.html> (01 05 04)

- [41] Pikall, H.; 2004, *Dokumentation zum Wiener SHD-Code*, unveröffentlicht
- [42] Potter, D.;, 1980, *Computational Physics*, John Wiley & Sons, Chicago
- [43] Rechenzentrum Humboldt-Universität, 1997, *Mathematica*,
Hypertext: http://www.hu-berlin.de/rz/mstat/www_math.html (01 05 04)
- [44] Reents, G., 2003, *Einführung in Mathematica*,
Hypertext: <http://hubble.physik.uni-konstanz.de/msoehn/mamakurs/netzversionen/termin1/introduc.nb.pdf> (Seite nicht mehr in Betrieb)
- [45] Reichel, H.; et al., 2004, *Lehrbuch der Mathematik 7.Band*, öbv&hpt
- [46] Sofroniou, M, 1995, *An Efficient Symbolic-Numeri Enviroment By Extending Mathematica's Format Rules (Format.ps)*,
Hypertext: <http://library.wolfram.com/infocenter/MathSource/60>
(01 05 04)
- [47] Sofroniou, M, 1995, *C, FORTRAN77, Maple and TeX Code Generation Package (Format.m)*,
Hypertext: <http://library.wolfram.com/infocenter/MathSource/60>
(01 05 04)
- [48] Stökl, A., 2003, Dissertation, in Vorbereitung
- [49] Strnadl, C.; Lorenz, R.; Scholz, W., 2000, *Symbolische Mathematik in der Theoretischen Physik*, Vorlesung am Institut für Theoretische Physik, Technische Universität Wien
Hypertext: <http://magnet.atp.tuwien.ac.at/scholz/projects/MmaHTML/>
(01 05 04)
- [50] Thompson, J.; Wansi, Z.; Wayne, C.; 1985, *Numerical Grid Generation*, North-Holland
- [51] Trefethen, L., 1992, *The Definition of Numerical Analysis*, SIAM News
- [52] Tscharnuter, W.M.; Winkler, K.-H., 1979, *Computational Physics Communication* **18**, 171
- [53] Überhuber, Ch., 1995, *Computernumerik 1*, Springer, Berlin
- [54] Überhuber, Ch., 1995, *Computernumerik 2*, Springer, Berlin

- [55] Universitätsrechenzentrum Heidelberg, 2001, *Mathematica-Block-Kurs August 2001*,
Hypertext: <http://web.urz.uni-heidelberg.de/Ausbildung/Kurse/MathemBlk/AUG01> (01 05 04)
- [56] Universitäts-Rechenzentrum Trier, 1999, *Einführung in Mathematica 3.0*,
Hypertext: <http://www.uni-trier.de/urt/user/baltes/docs/mathe/mathe.html>
(01 05 04)
- [57] von Neumann, J.; Richtmyer, R.D., 1950, *Journal of Applied Physics* **21**, 232
- [58] Winkler, K.-H.; Mihalas, D.; Norman, M., 1985, *Adaptive-Grid Methods with Asymmetric Time-Filtering*, Computer Physics Communications, **36**, 121
- [59] Winkler, K.-H.; Norman, M.; Mihalas, D., 1984 *Implicit adaptive-grid radiation hydrodynamics*, MPA 112
- [60] Winkler, K.-H.; Norman, M.; Newman, M., 1984, *Adaptive Mesh Techniques for Fronts in Star Formation*, Physica, **12D**, 408-425
- [61] Winkler, K.-H.; Norman, M., 1986, NATO-ASI Series C., Vol.188, D.Reidel, Dordrecht
- [62] Wolfram, St., 1999, *The Mathematica Book*, Cambridge University Press
- [63] Wolfram-Research, 2003, Hypertext: <http://www.wolfram.com> (01 05 04)
- [64] Wolfram Research Forums, 2003, Hypertext: <http://forums.wolfram.com>
(01 05 04)
- [65] Wolfram Documentation Center, 2003, *Documentation Center Mathematica 4 in deutsch*, Hypertext: <http://documents.wolfram.com/v4-de> (01 05 04)
- [66] Wolfram Documentation Center, 2003, *Documentation Center Mathematica 5 in English*, Hypertext: <http://documents.wolfram.com/v5> (01 05 04)
- [67] Wolfram Web Resources, 2003, Hypertext: <http://www.wolfram.com/webresources.html> (01 05 04)
- [68] Wuchterl, G., 1995, Computational Physics Communication **89**, 119
- [69] Young, D.: 1971, *Iterative Solution of Large Linear Systems*, Academic Press

An dieser Stelle möchte ich allen danken, die mir ermöglicht haben, diese Diplomarbeit abzuschließen.

Einerseits meinen Eltern, die wohl nach anfänglichen Entsetzen über die Studienwahl, mich bedingungslos materiell und seelisch unterstützt haben. Dank gebührt auch dem Rest meiner Familie, darunter meiner Schwester Birgit und meinen Großmüttern.

Andererseits meinen Freunden und Studienkollegen, stellvertretend zu nennen Florian Freistetter und Stefan Reisinger.

Dankbar bin ich auch für die Unterstützung durch die TAG. Den jetzigen und ehemaligen Mitgliedern, die mitgeholfen haben den Wiener SHD-Code zu entwickeln. Besonderer Dank gebührt Holger Pikall, ohne den ich wohl im Source-Code untergegangen wäre und Alexander Stökl, der immer guten Rat parat hatte, wenn ich nicht mehr weiter wusste (und so manches 4π für mich suchte), sowie Christian Reimers für seine Gelassenheit und sein Wissen über Latex.

Zu Dank bin ich auch Bernhard Loew verpflichtet, der mich schonungslos in die Welt von *Mathematica* gestoßen, mich aber nicht untergehen hat lassen, Thomas Löger für seine Geduld bei Erklärungen, sowie Christoph Lhotka, der wohl in Zukunft bei *Wolfram* arbeiten wird.

Riesiger Dank gebührt vor allem Dr. Ernst A. Dorfi, der im wahrsten Sinne astronomische Geduld bewiesen und nie den Glauben verloren hat, dass diese Diplomarbeit je abgeschlossen werden könnte. Außerdem trotz der dem widerlichen mañana.

Dank der Schnucku!

Lebenslauf

Kittel Matthias
geboren 25 12 1975 in Wien
österreichischer Staatsbürger

wohnhaft
Florianigasse 1
2604 Theresienfeld

Vater - Kittel Christoph, geboren 03 10 1947, kaufmännischer Angestellter

Mutter - Kittel Gerlinde, geboren 26 11 1948 als Auer Gerlinde, diplomierte Krankenschwester

Schwester - Kittel Birgit, geboren 09 11 1977, Studium der Psychologie

- 1979/80 – 1981/82 Kindergarten Notre Dame de Sion, Burggasse 37, 1070 Wien
- 1982/83 – 1985/86 Volksschule Notre Dame de Sion, Burggasse 37, 1070 Wien
- 1986/87 – 1987/88 Bundesrealgymnasium Kandlgasse 39, 1070 Wien
- 1988/89 – 1993/94 Bundesrealgymnasium Gröhrmühlgasse 27, 2700 Wiener Neustadt
Reifeprüfung mit ausgezeichnetem Erfolg 06 1994
- 10 1994 – 12 1994 Ausbildung zum Sanitätsgehilfen in der Ostarrichi-Kaserne in Amstetten im Rahmen des Präsenzdienstes
- 12 1994 – 06 1995 Ableistung des restlichen Präsenzdienstes an der Theresianischen Militärakademie Wiener Neustadt
- seit 10 1995/96 Studium der Astronomie an der Universität Wien
seit 10 1995/96 Studium der Physik (Diplom) an der Universität Wien

- 1997 Stipendium des Senates der Universität Wien
- 1998 Stipendium des Senates der Universität Wien
- 1998 Leistungsstipendium der Fakultät für Naturwissenschaften der Universität Wien
- 02 1998 Abschluss des ersten Studienabschnittes Physik (Diplom) mit ausgezeichnetem Erfolg
- 10 1998 Abschluss des ersten Studienabschnittes Astronomie mit ausgezeichnetem Erfolg
- 1999 Stipendium des Senates der Universität Wien
- 10 1999 Abschluss des zweiten Studienabschnittes Astronomie mit ausgezeichnetem Erfolg
- 2000 Stipendium des Senates der universität Wien
- 2000 - 2001 Mitglied der Kommission zur Nachbesetzung der ordentlichen Universitätsprofessur *Theoretische Astronomie* als Studentenvertreter
- seit 03 2000 Studium der Mathematik (Diplom) an der Universität Wien
- seit 06 2001 Diplomarbeit am Institut für Astronomie der Universität Wien - Thema: *Symbolische Ableitungen im Rahmen numerischer Strahlungshydrodynamik*
- seit 10 2001 Lehramtsstudium der Physik und Mathematik an der Universität Wien
- 05 2002 erste Lehrtätigkeit am BORG 1, Hegelgasse 14, 1010 Wien im Rahmen des pädagogischen Praktikums der Lehramtsausbildung
- 09 – 12 2002 Lehrtätigkeit am BORG 1, Hegelgasse 14, 1010 Wien im Rahmen des Schulpraktikums (Mathematik) der Lehramtsausbildung
- 2003 Top-Stipendium der niederösterreichischen Landesregierung
- 02 2003 Lehrtätigkeit am BG/BRG Zehnergasse 17, 2700 Wr.Neustadt im Rahmen des Schulpraktikums (Physik) der Lehramtsausbildung
- 07 2003 Abschluss der pädagogischen Ausbildung im Rahmen der Lehramtsausbildung
- 09 2003 Abschluss des ersten Studienabschnittes Lehramt Physik - Mathematik

- seit 12 2003, Lehrtätigkeit im Fach Mathematik am BORG Herzog-Leopold-Stra ße, 2700 Wr.Neustadt
- 05 2004 Abschluss des zweiten Studienabschnitts Lehramt Physik - Mathematik