Hadronisierung des Quark-Gluon-Plasmas im chromodielektrischen Modell

Inauguraldissertation
zur
Erlangung des Doktorgrades
der
Naturwissenschaftlichen Fakultät
der Justus-Liebig-Universität Gießen
Fachbereich 13 - Physik

vorgelegt von
Christoph Traxler
aus Marburg/Lahn

Gießen 1999

Dekan: Prof. Dr. Karl-Heinz Schartner
I. Berichterstatter: Prof. Dr. Ulrich Mosel
II. Berichterstatter: Prof. Dr. Werner Scheid
Tag der mündlichen Prüfung: 13. 4. 1999

Inhaltsverzeichnis

1 Einleitung **5**
1.1 Aufbau der Materie 5
1.2 Einordnung und Motivation dieser Arbeit 10
1.3 Gliederung 13

2 Hadronen, Strings und Quantenchromodynamik **17**
2.1 Stringmodell der Hadronen 17
2.2 Quarks 21
2.3 Quantenchromodynamik (QCD) 23
2.4 Confinement 26
2.5 Modelle des Confinements 30

3 Das chromodielektrische Modell **37**
3.1 Lagrangedichte und Eichinvarianz 37
3.2 Bewegungs- und Kontinuitätsgleichungen 38
3.3 Klassische Approximation 39
3.3.1 Naiver nichtabelscher Ansatz 40
3.3.2 Diskussion zur Verwendung ausgedehnter Teilchen 41
3.3.3 Klassische Farbdynamik im nichtabelschen Fall . . 42
3.3.4 Funktionierender Ansatz in abelscher Approximation 44
3.3.5 Wahl der Farbladungen 45
3.4 Farbconfinement 48
3.5 Irreduzible Weiße Cluster (IWCs) und Hadronenspektrum 52
3.6 Vollständigkeitsbeweis der IWC-Klassifikation 55
3.7 Analoge Betrachtung in der QCD 56
3.8 Diskussion eines möglichen Quark-σ-Kopplungsterms . . . 61

4 Computersimulation des Modells **63**
4.1 Bewegungsgleichungen und Approximationen 64
4.1.1 Bewegungsgleichungen und Approximationen des Modells 64

4.1.2 Approximationen der numerischen Simulation . . . 65
4.2 Behandlung des skalaren Feldes und der Teilchen 69
4.2.1 Explizite Methode für die Zeitschritte 69
4.2.2 Berücksichtigung des Pauliprinzips 70
4.3 Behandlung des Farbfeldes 73
4.3.1 Problemstellung 73
4.3.2 Diskretisierung mit der Methode der finiten Elemente 74
4.3.2.1 Variationsverfahren 74
4.3.2.2 Finite Elemente als Basisfunktionen . . . 75
4.3.2.3 Simplexkoordinaten und lineare Basis . . 77
4.3.2.4 Quadratische Ansatzfunktionen 80
4.3.2.5 Diskussion 82
4.3.3 Adaptive Verfeinerung 85
4.3.3.1 Anforderungen an ein Verfeinerungsschema 85
4.3.3.2 Bisektionsschema in d Dimensionen 87
4.3.3.3 Gleichmäßige Partitionsverfeinerung . . . 89
4.3.3.4 Adaptive (lokale) Partitionsverfeinerung . 91
4.3.3.5 Simplexverfeinerung in MUESLEA 92
4.3.3.6 Zäune im Simplexwald 94
4.3.4 Hierarchisches Mehrgitterverfahren 97
4.3.4.1 Problem und Lösungsidee 97
4.3.4.2 Lösung der Gleichungen auf dem groben Gitter 100
4.3.4.3 Einschub: Positivität des diskretisierten Differentialoperators 101
4.3.4.4 Grobe und feine Relaxation auf zwei Gitterstufen 107
4.3.4.5 Volles Mehrgitterverfahren 114
4.3.4.6 Grobstruktur und Präzisionssteuerung . . 116
4.3.4.7 Berechnung der Matrizen und Relaxation 117
4.3.4.8 Relax-Datenstrukturen 121
4.3.4.9 Behandlung der Randbedingungen 122
4.3.5 Der letzte Schliff: Optimierung 122
4.3.6 Rechenzeit und Speicherplatzbedarf 124
4.4 Diskussion 128
4.5 Clustererkennung und Hadronisierung in CDM 132

5 Resultate **135**
5.1 Wahl der Modellparameter 135
5.2 Ein einzelnes Quark 139
5.3 Stringspannung 142
5.4 String-String-Wechselwirkung 146
5.5 Hadronisierung 146
5.5.1 Initialisierung eines Anfangszustands 148

5.5.2 Analyse des Endzustands 150
5.5.3 Schnappschüsse der Simulation 151
5.5.4 Energieerhaltung, Hadronisierungszeit und Temperatur 154
5.5.5 Massenverteilungen 158
5.5.6 Rapiditäts- und Impulsspektren 160

6 Zusammenfassung und Ausblick 165
6.1 Innovationen und Resultate 165
6.2 Ausblick 167
6.2.1 Miteinbeziehung von Pionen 167
6.2.2 Studium des Phasenübergangs und andere Projekte 168

A Ergänzung zum Beweis der IWC-Klassifikation 171

B Wegweiser durch das Programm 177
B.1 CDM, MUESLEA, DISP und GUI 177
B.1.1 Programmaufruf und Kommunikationswege 177
B.1.2 Hinweise zur Benutzung und Weiterarbeit 182
B.2 Klassenstruktur des Programms CDM 186

Literaturverzeichnis 201

Danksagung 209

“Ob mir durch Geistes Kraft und Mund
Nicht manch Geheimnis würde kund;
Daß ich nicht mehr mit saurem Schweiß
Zu sagen brauche, was ich nicht weiß;
Daß ich erkenne, was die Welt
Im Innersten zusammenhält,
Schau alle Wirkenskraft und Samen,
Und tu nicht mehr in Worten kramen.”
J. W. VON GOETHE, *“Faust I”*

PETRUS: *Forscht, wo ihr was zum Forschen findet.*
Das Unerforschbare lasst unergründet.
Wir kennen euch. Ihr seid entrüstet,
wenn euch etwas verboten ist...
JOHNNY: *Sankt Peter übertreibt entschieden.*
Wir sind gar nicht so wissbegierig.
Den meisten wär's auch viel zu schwierig.
E. KÄSTNER *[1]*

1 Einleitung

1.1 Aufbau der Materie

Die Suche nach dem *Quark-Gluon-Plasma* (QGP) ist seit etwa zwei Jahrzehnten eine der größten wissenschaftlichen Anstrengungen in der Physik der starken Wechselwirkung. In den letzten Jahren sind $200A \cdot GeV$ Pb-Strahlen am SPS-Beschleuniger (CERN, Genf) verfügbar gemacht worden. Mit zukünftigen Beschleunigern wird man noch höhere Energien erreichen können, nämlich $(200 + 200)A \cdot GeV$ am RHIC (BNL, Brookhaven) und $(6300 + 6300)A \cdot GeV$ am LHC (CERN). Diese Energien korrespondieren zu geschätzten maximalen Energiedichten von $\epsilon = 5GeV/fm^3$ bzw. $\epsilon = 9GeV/fm^3$ und Temperaturen von etwa $T = 230MeV$ bzw. $T = 260MeV$. Im SPS beträgt die erreichte Temperatur etwa $T = 160MeV$ [2]. Zum Vergleich: Der Sonnenkern hat eine Temperatur von etwa 15 Millionen Grad Kelvin, das entspricht $1.2\,keV$. Der Phasenübergang hadronischer Materie zum QGP wird bei einer Temperatur erwartet, die etwa dem 200.000-fachen dieser Temperatur entspricht.

Die experimentellen Aufbauten kosten viel Geld. Der erwähnte LHC wird über eine Milliarde Euro gekostet haben, wenn er um das Jahr 2010 mit der Arbeit beginnen wird. Das sind 4 Euro von jedem Einwohner Europas. Betriebskosten und Gehälter der involvierten CERN-Wissenschaftler sind darin noch nicht enthalten. Wozu diese gewaltigen Anstrengungen? Die Physiker versuchen, der Antwort zu einer wichtigen Frage der Menschheit näherzukommen: woraus Materie besteht, und warum sie die uns bekannten Eigenschaften aufweist.

Lange ist bekannt, dass Materie aus Kristallen oder amorphen Zusammensetzungen von Molekülen oder Atomen besteht. Seit Rutherfords berühmtem α-Streuversuch an einer Goldfolie (1911) wissen wir, dass Atome aus einer Elektronenhülle und einem positiv geladenen Atomkern aufgebaut sind. Die *Schrödingergleichung* zur Beschreibung der Elektronenhülle des Wasserstoffatoms (1926) wird heute bereits in Schulbüchern vorgestellt. Die typische Ionisierungsenergie eines Atoms (Bindungsenergie des äußersten Elektrons) beträgt dabei einige meV bis einige eV. Die Struktur des Atomkerns ist qualitativ längst verstanden: er besteht aus *Protonen* und *Neutronen* (letztere entdeckt durch Chadwick 1932), die durch eine starke, aber kurzreichweitige *Kernkraft* (*hadronische Wechselwirkung*) zusammengehalten werden. Es ist diese Kernkraft, die vom Menschen zur Energiegewinnung in Kernreaktoren (und leider auch in Kernwaffen) benutzt wird. Die typische Energieskala liegt hier im MeV-Bereich, ist also wesentlich höher als die Energieskala chemischer Reaktionen. Die Substrukturteilchen der Protonen und Neutronen, die *Quarks*, wurden zunächst als Hilfsmittel zur Systematisierung des Hadronspektrums von Gell-Mann und Zweig 1963 eingeführt. Experimente, bei denen hochenergetische Elektronen an Protonen gestreut wurden [3], haben das Quarkmodell experimentell belegt; heute sind die Quarks integraler Bestandteil des Standardmodells der Elementarteilchenphysik.

Bis heute unverstanden ist aber die Natur der Bindungskräfte zwischen den Quarks. Es wurden nie freie Quarks beobachtet; sie scheinen äußerst fest in den Protonen und Neutronen gebunden zu sein. Diesen Umstand nennt man *Confinement*, Gefangenschaft, der Quarks. Tatsächlich kennt man heute neben den Protonen und Neutronen ein reiches Spektrum von gebundenen Quarkzuständen, die *Hadronen* [4]. Immerhin verfügen wir mit der *Quantenchromodynamik* (QCD) über eine Theorie der Wechselwirkungen der Quarks. Die QCD ist eine lokale Eichfeldtheorie, ähnlich der Quantenelektrodynamik (QED), die die Kräfte zwischen Elektronen beschreibt. Die Rolle der elektrischen Ladung spielt in der QCD die *Farbe*, die man den Quarks zuordnet; ein Quark kann rot, grün oder blau sein, ein Antiquark ist antirot, antigrün oder antiblau. Die Rolle des Photons als Wechselwirkungsboson der QED übernimmt in der perturbativen QCD (pQCD) das Gluon, das es in acht verschiedenfarbigen Versionen gibt [5].

Die QCD hat beachtliche Erfolge zu verzeichnen. Man konnte beispielsweise die Massen der leichtesten acht Hadronen korrekt aus der Theorie berechnen. Im Bereich hochenergetischer Elektron-Proton-Kollisionen hat die perturbative QCD (pQCD) eine hohe Vorhersagekraft unter Beweis stellen können.

Die Situation ist trotz der Erfolge noch lange nicht befriedigend, weil man den *Mechanismus* der Farbwechselwirkung noch nicht verstanden hat: die Identifikation der (hoffentlich wenigen) *relevanten Freiheitsgrade* steht noch aus, und somit entzieht sich die Phänomenologie der QCD noch weitgehend jeglicher Intuition. Warum beispielsweise existieren nur drei Sorten von Hadronen, die *Mesonen* mit je einem Quark und einem Antiquark, die (Anti-)*Baryonen* mit drei (Anti-)Quarks, und die *Gluebälle*[1]? Als Erklärung dafür verweist man in der Standardliteratur[2] auf die Farbeigenschaft der Quarks. Der Begriff der Farbe kommt der Intuition entgegen: Das Confinement der Quarks läßt sich anschaulich so beschreiben, dass Hadronen "weiß"[3] sein müssen. So können Hadronen entweder aus drei Quarks verschiedener Farbe bestehen, oder aus einem Quark und einem Antiquark mit der entsprechenden Antifarbe. Man spricht deshalb auch von *Farbconfinement*. Es wird nun postuliert, dass Hadronen grundsätzlich Farbsingletts sein müssen. Dies "erklärt", warum keine Hadronen mit vier oder fünf Quarks existieren, aber warum Sechs-Quark-Hadronen nicht beobachtet werden, bleibt unklar. Außerdem ist ein Postulat eher eine Beschreibung denn eine Erklärung (ähnlich dem Bohrschen Postulat ganzzahliger Bahndrehimpulse in der Atomphysik). Der genaue Mechanismus des Confinements wird wohl noch einige Zeit Forschungsgegenstand bleiben.

Das Farbconfinement scheint aber nicht wirklich absolut zu sein: Statische QCD-Gitterrechnungen [6, 7, 8], in denen das Verhalten von Farbfeldern bei hoher Temperatur untersucht wurde, geben Anlass zu der Vermutung, dass bei Überschreitung einer kritischen Temperatur oder einer kritischen Energiedichte ein Phasenübergang stattfindet: Die Hadronen verdampfen gewissermaßen und geben ihre Bestandteile, die Quarks und Gluonen, frei. Den dadurch entstehenden "Dampf" aus Quarks und Gluonen nennt man das *Quark-Gluon-Plasma* (QGP), den Phasenübergang nennt man *Deconfinement*. Vorhersagen für die kritische Temperatur liegen bei $150 - 200 MeV$ (200.000-fache Sonnenkerntemperatur), die kritische Energiedichte wird bei etwa der sechsfachen Atomkerndichte vermutet (das sind $2.5\, GeV/fm^3$).

Kann man ein QGP experimentell herstellen oder irgendwo im Weltall

[1] Reine gluonische Zustände (Gluebälle) sind zwar bis heute nicht eindeutig experimentell nachgewiesen, man vermutet aber als Ursache einfach ihre Kurzlebigkeit. Es sind jedenfalls Resonanzen gefunden worden, die als Kandidaten für Gluebälle gelten.

[2] siehe z. B. Halzen und Martin [5] oder ein beliebiges anderes Lehrbuch der theoretischen Elementarteilchenphysik.

[3] genauer: weiße Farbsingletts, was eine strengere Bedingung darstellt.

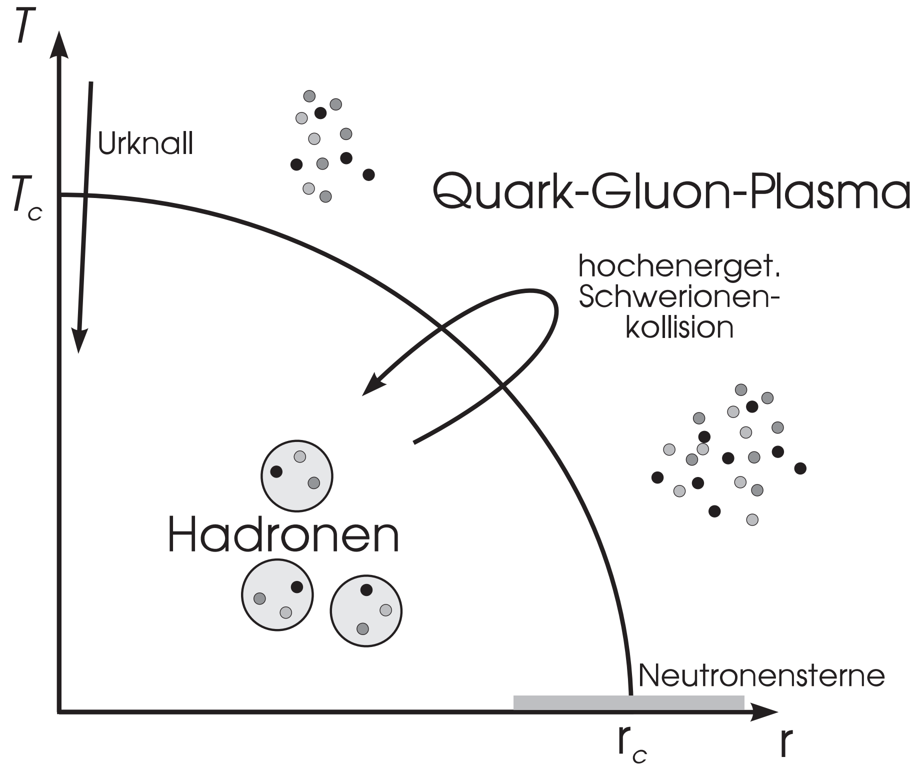


Abbildung 1.1: *Phasendiagramm hadronischer Materie.*

beobachten? In Abb. 1.1 sieht man drei mögliche Szenarien, in denen ein QGP vorkam oder vorkommen könnte. Kurz nach dem Urknall kondensierte ein Gas von Quarks und Gluonen (QGP) zu Hadronen. Dieses Szenario ist für uns nicht direkt beobachtbar.

Ein anderer möglicher Zugang zum Plasma ist die Bildung sogenannter Quarksterne, das sind massereiche Sterne, die nach Beendigung ihres nuklearen Brennens kollabieren. Da der thermische Gasdruck dann entfällt (das Brennen ist ja beendet), kann den Stern nur noch wenig am totalen Gravitationskollaps hindern. Zum Beispiel könnte das Pauliprinzip der Elektronen den Kollaps stoppen; die entstehende Sternleiche nennt man einen *Weißen Zwerg*. Falls der Gravitationsdruck sehr groß ist, wird es energetisch günstiger für die Elektronen sein, mit Protonen zu reagieren und Neutronen zu bilden. Der Stern kollabiert weiter und kann eventuell durch das Pauliprinzip der Neutronen gestoppt werden: ein *Neutronenstern* entsteht. Solche Neutronensterne sind teilweise als *Pulsare* der Beobachtung zugänglich. Es wird vermutet, dass der Kern eines Neutronensterns aus Quarkmaterie bestehen könnte; in gewissen Sinne wäre also der Kern eines Neutronensterns ein einziges riesiges Hadron, ein kaltes Quark-Gluon-Plasma (QGP).

Ein dritter möglicher Weg wurde am Anfang dieser Einleitung bereits angesprochen: Durch ultrarelativistische Schwerionenkollisionen sind sowohl hohe Dichten als auch hohe Temperaturen experimentell verfügbar, allerdings nur auf sehr kleinem Raum und nur für sehr kurze Zeit. So reichen heutige Beschleunigeranlagen wohl noch nicht aus, um ein Quark-Gluon-Plasma lange genug leben zu lassen, um es eindeutig zu identifizieren. Man erreicht bei Schwerpunktsenergien von $4{-}5A{\cdot}GeV$ $(Au{-}Au)$ am AGS (BNL, Brookhaven) maximale Energiedichten (im thermischen Gleichgewicht) von $0.9GeV/fm^3$ und Temperaturen von $150MeV$; bei Schwerpunktsenergien von $20A \cdot GeV$ $(S - Pb)$ am SPS (CERN, Genf) erreicht man maximale Energiedichten von $2.4GeV/fm^3$ und Temperaturen von $190MeV$ — das liegt gerade so an der Grenze. Bedeutend besser sind die Aussichten da am RHIC (ebenfalls BNL), der 1999 in Betrieb gehen soll: dort wird man Schwerpunktsenergien von $200A \cdot GeV$ $(Au - Au)$ erreichen, was zu Energiedichten von $5GeV/fm^3$ und Temperaturen von $230MeV$ führt. Eindeutig oberhalb des berechneten Phasenübergangs werden die Temperaturen am geplanten LHC (wieder CERN) liegen, der im nächsten Jahrzehnt fertiggestellt werden soll: Schwerpunktsenergien von über $6A \cdot TeV$ $(Pb - Pb)$ sollen Energiedichten von $9GeV/fm^3$ und Temperaturen von $260MeV$ bereitstellen [9].

Die Lebensdauer eines Plasmas am LHC wird auf 3-$50fm$ geschätzt. Das ist so kurz, dass den Experimentatoren keine Zeitauflösung möglich ist. Bevor das Studium der Quark-Gluon-Plasma-Physik überhaupt ernsthaft beginnen kann, muss also erst einmal nachgewiesen werden, ob während der Kollisionsreaktion überhaupt ein Quark-Gluon-Plasma als Zwischenzustand auftrat. Man hofft, dass das zwischenzeitliche Plasma im Endzustand der Reaktion eine "Signatur" hinterlässt, durch die die auseinanderspritzenden Hadronen verraten, ob sie aus einem Plasma stammen, oder ob die Reaktion "rein hadronisch" war.

Mögliche Signaturen des Plasmas können prinzipiell in zwei Klassen unterteilt werden: solche, die auf elektromagnetischen Reaktionen beruhen (direkte Photonen, Dileptonen), oder stark wechselwirkende Proben (Strangeness, Charm, etc.). Der Vorteil der ersteren ist, dass der Plasma-Feuerball für Photonen oder Elektronen transparent ist, man sozusagen direkt die Reaktionszone beobachten kann. Der Vorteil der letzteren ist, dass sie wesentlich höhere Statistik aufweisen, Daten mit brauchbarer Statistik also "billiger" verfügbar sind — tatsächlich genügt eine einzige ultrarelativistische Schwerionenkollision, um Hunderte oder Tausende von Pionen zu erzeugen, während gleichzeitig vielleicht nur ein einziges direktes Photon aus dem Plasma entsteht. Stark wechselwirkende Teilchen erlauben also sogar die Analyse einzelner Ereignisse (*Events*) — und es mag ja tatsächlich so sein, dass in einer bestimmten Reaktion (zentrale Kollision) ein Plasma entsteht, während dies in anderen Events nicht der Fall ist.

Nun entstehen Hadronen aus dem Plasma als Resultat des *Confinement-Phasenübergangs* vom Plasma zu hadronischer Materie, bei der die vorher freie Farbladungen sich zu Farbsingletts zusammenschließen. Dieser Phasenübergang steht zwischen der Quark-Gluon-Plasma-Physik

und den in Detektoren erfassbaren Größen. Deshalb ist es von entscheidender Bedeutung, den Phasenübergang zu verstehen, will man das Plasma untersuchen. Abgesehen davon ist das Studium des Phasenübergangs selbst bereits interessant, da hier die (wenig verstandene) langreichweitige Farbwechselwirkung die entscheidende Rolle spielt. Für die Untersuchung der Abkühlung des Plasmafeuerballs und des Confinementübergangs bieten sich dynamische (zeitlich aufgelöste) Computersimulationen an. Da aber eine dynamische Simulation des Plasmafeuerballs auf Basis der QCD auf heutigen Computern nicht realisierbar ist, sind wir für eine dynamische Simulation auf Modellbildung angewiesen. Die QCD steht hierbei gewissermaßen für die “exakte”, aber zu komplizierte Theorie, und es geht darum, ein einfaches Modell zu konstruieren, das die relevanten Eigenschaften der QCD widerspiegelt.
Im Rahmen der vorliegenden Dissertation wurde ein solches Modell, das *chromodielektrische Modell* (CDM), zur Untersuchung von Hadronisierungsszenarien eingesetzt. Dazu musste zunächst ein geeignetes Computerprogramm (Programmname CDM) entwickelt werden, das als Werkzeug für die vorgestellten Studien diente (und für zukünftige Aufgaben weiterhin bereitsteht).
Am Ende dieser Einleitung soll die anfängliche Frage nach dem Nutzen der gewaltigen Forschungsanstrengungen (nach Meinung des Autors) beantwortet werden: Der enorme Forschungsaufwand im Rahmen relativistischer Schwerionenexperimente und verwandter theoretischer Untersuchungen zielt nicht auf Methoden zur Energiegewinnung oder andere zur Mehrung unseres Wohlstands anwendbare Resultate ab. Es geht um die *kulturell bedeutsame* Frage, warum die Materie in der uns bekannten Form überhaupt existiert. Dass diese Fragestellung eine kulturelle Dimension besitzt, mag man dem Goethe-Zitat im Kopf dieses Kapitels entnehmen.

1.2 Einordnung und Motivation dieser Arbeit

Ultrarelativistische Schwerionenstrahlen (z. B. Au- oder Pb-Kerne) stellen das beste Mittel dar, um dem Ziel eines thermalisierten Quark-Gluon-Plasmas in Beschleunigerexperimenten nahezukommen. Schwerionenkollisionen machen extrem hohe Energie- und Baryonenzahldichten gleichzeitig verfügbar. Diese Vorteile auf der experimentellen Seite implizieren leider Schwierigkeiten bei der theoretischen Behandlung: auch wenn der thermodynamische Zugang zur Quarkmaterie bereits weit entwickelt wurde [6, 8, 10], fehlt mancher Einblick in die Reaktionsdynamik. Auch das Studium von Szenarien bei nichtverschwindender Baryonenzahldichte steckt noch in den Anfängen [11, 12]. Die Lösung der theoretischen Probleme, also der Realzeitbeschreibung und der Konsequenzen der nichtverschwindenden Baryonenzahldichte, ist von der Erfindung neuer nichtperturbativer Techniken zur Behandlung komplexer effektiver Wirkungen abhängig. Andererseits gibt es mehrere phänomenologische Modelle, die darauf abzielen, genau diese dynamischen Eigenschaften der QCD zu simulieren.

Das Ziel theoretischer Arbeiten ist es, den Mechanismus des Farbconfinements zu entschlüsseln. In diesem Rahmen bedeutet das Verständnis der Hadronisierung, den Confinementmechanismus in einem experimentell präparierbaren Szenario kennenzulernen. Im folgenden geben wir einen kurzen Überblick über die momentan existenten (und uns bekannten) Hadronisierungsmodelle. Diese Modelle lassen sich nach der Zeitdauer, die für die Hadronisierung eines Plasmafeuerballs benötigt wird, grob in drei Kategorien einteilen: *instantane Hadronisierung* (weniger als $1fm$ Hadronisierungszeit), *Gleichgewichtshadronisierung* (mehr als $20fm$ Hadronisierungszeit), und *evolutionäre Hadronisierung* (dazwischen).

String- und Partonfragmentationsmodelle [13, 14] sind die ältesten der ersten Kategorie (instantane Hadronisierung); sie wurden bereits bei der Beschreibung elementarer Kollisionen (pp, $p\bar{p}$ und e^+e^-) eingesetzt. Mit Hilfe des Glaubermodells [15] werden die Ergebnisse elementarer pp-Kollisionen dann für eine Schwerionenkollision aufaddiert. Verhältnisse zwischen Hadronenzahlen hängen dabei von phänomenologischen Fragmentationsfunktionen ab, eine Konstruktion, die auf Feynman zurückgeht [16]. Auch wenn diese Parameter sich auf die QCD im nichtperturbativen Grenzfall beziehen, fehlt jede Erklärung oder Herleitung für ihre Werte.

Weitere statistische und kombinatorische Modelle verbinden Quarkzahlen und (Prä-)Hadronenzahlen, indem sie Koaleszenz, d. h. Nähe im Phasenraum, zugrundelegen. Das *Quarkclustermodell*, das bspw. im Partonkaskadenmodell [17] oder auch in Stringmodellen wie VENUS [18] und (U)RQMD [19] zum Einsatz kommt, ist stark von Quarkeigenschaften (bspw. effektiven Massen) im nichtperturbativen Grenzfall abhängig. Das Budapester *ALCOR-Modell* [20] beruht auf der Annahme einer extrem schnellen Hadronisierung massiver Quarks fern vom thermischen Gleichgewicht, wobei Gluonen zunächst in Quarks fragmentieren. Auch wenn die individuelle Hadronenformationszeit in diesen Modellen verschwindet, ist der Moment, an dem ein Hadron entsteht, über einem Bereich von $0 - 30fm$ variabel. In dieser Art von Modellen ist die Hadronentstehung ein Prozess, der nur beschrieben, nicht aber erklärt wird. Entsprechend beschränkt ist die Vorhersagekraft.

Das Modell gemischter Phasen (*mixed phase model*) — einer der neuesten Vertreter der zweiten Kategorie — geht von einem thermalisierten Mehrphasenzustand aus. Zu diesem wird eine *nichtideale Zustandsgleichung* konstruiert, die Teilchenkorrelationen effektiv miteinbezieht und Confinementeigenschaften berücksichtigt. Auch wenn dies ein vielversprechender Ansatz sein mag, ist der Ausgangspunkt (thermisches Gleichgewicht) eine gewagte Annahme, speziell in der Anwendung auf relativistische Schwerionenkollisionen, wo die charakteristische Zeit der Quarkclusterformation vergleichbar der Äquilibrierungszeit ist, m. a. W. ein thermisches Gleichgewicht der Quarkphase wohl nie errreicht wird. Auch fehlen Diquarks und Baryonen bisher in diesem Modell.

Es gibt mehrere Ansätze der Beschreibung des Hadronisierungsphasenübergangs, die auf *chemischen Ratengleichungen* beruhen. Hier beträgt die Hadronisierungszeit einige *fm*. Nimmt man eine massive, nichtrelativistische Quarkphase an, kann die Hadronisierung mit Hilfe phänomenologischer Confinementpotentiale oder auch einfach dem störungstheoretischen Coulombpotential beschrieben werden. Die Bindung der Farbladungen zu farbneutralen Clustern wird hier durch Mediumeffekte wie die Stringformation verursacht. Die Zeitentwicklung des gemischten Quark-Hadron-Systems wird durch einen Satz chemischer Ratengleichungen beschrieben. Ein Ansatz dieser Art wurde bereits vor einem Jahrzehnt publiziert [21] und wurde vor kurzem in das bereits erwähnte ALCOR-Modell integriert[22].

Im Rahmen des *Nambu-Jona-Lasinio-Modells* (NJL) wurden "Hadronisierungsquerschnitte" (*hadronization cross sections*) berechnet [23, 24]. In diesem Modell sind aber nur die leichtesten Mesonen überhaupt gebunden. Deshalb kann in diesem Modell nur direkte Übergänge von $q\bar{q}$-Paaren zu Paaren leichter (gebundener) Mesonen, z. B. $\pi^+\pi^-$, studieren. Die Hadronisierung ist folglich im NJL-Modell nicht von einer hadronischen Kaskade gefolgt, und der Hadronisierungsmechanismus bleibt im Verborgenen. Schlimmer noch: die effektive Masse eines freien Quarks bleibt auch bei $T = 0$ noch endlich. Farbconfinement ist somit im NJL-Modell nicht vorhanden.

Das ursprünglich von Friedberg und Lee [25, 26, 27] vorgeschlagene *chromodielektrische Modell* (CDM) beschreibt Hadronen als gebundene Cluster farbgeladener Quarks in einem phänomenologischen skalaren Potential. Zu diesem Modell gibt es eine beachtliche Menge an Arbeiten (und sogar ein Lehrbuch), die stationäre hadronische Zustände mit quantisierten Quarks studieren [28, 29, 30, 31]. Die dynamische Realzeitsimulation ist dagegen relativ neu [31, 32, 33, 34, 35]. Die letzteren Arbeiten konstruieren eine *semiklassische Transporttheorie*, wobei Quarks wie klassische Ladungswolken behandelt werden. Es sind diese zuletzt zitierten Arbeiten, die der vorliegenden Dissertation geistig Pate standen, auch wenn wir von einer transporttheoretischen Beschreibung auf Basis der Einteilchen-Phasenraumverteilung mittlerweile wieder Abstand nehmen mussten [34, 35].

Die vorliegende Arbeit berichtet über einen vollständig klassischen, "molekulardynamischen" Hadronisierungsansatz auf Quarkebene. Zur dynamischen Beschreibung wird eine klassische Approximation des CDM benutzt, wobei Quarks und Gluonen als ausgedehnte, aber starre Farb-

ladungsverteilungen ("Tennisbälle") behandelt werden. Dazu kommen ein skalares σ-Feld und zwei abelsche Vektorfelder, die zu den beiden Erzeugern $\lambda^{3,8}$ einer maximalen abelschen Untergruppe der $SU(3)_C$-Symmetrie der QCD korrespondieren. Die Vektorfelder werden mittels zweier farbelektrostatischer Potentiale $\phi^{3,8}$ behandelt. Die sechs verbleibenden "geladenen" Gluonen werden wieder als Teilchen ("Tennisbälle") behandelt. Wir nehmen diese Vereinfachungen in Kauf, um eine molekulardynamische Simulation von etwa 500 Quarks und Gluonen in akzeptabler Rechenzeit durchführen zu können. Der wesentliche Confinementmechanismus dieses Modells ist in der Selbstwechselwirkung $U(\sigma)$ des skalaren Felds und der Konstruktion der chromodielektrischen Zahl $\kappa(\sigma)$ begründet. Diese beiden Funktionen imitieren die erwartete nichtperturbative Phänomenologie der QCD.
Zur Vorhersagekraft einer solchen Untersuchung ist folgendes zu sagen: Mikroskopische Studien wie diese könnten trotz ihrer Kompromisse in der Lage sein, Aussagen über die Zeitskala des Hadronisierungsprozesses zu machen, über die Frage, ob die Hadronisierungszeit kürzer oder länger ist als die charakteristische Zeit zur Erreichung des chemischen oder thermischen Gleichgewichts. Die Vorhersagen bezüglich Hadronzahlverhältnisse können mit denen der anderen Modelle verglichen werden. Allerdings stecken die vorgestellten Untersuchungen noch in den Anfängen.
Die Nachteile unseres Ansatzes, also die klassische Behandlung der Farbladungen und der Cluster, die abelsche Approximation in den langreichweitigen Feldern und die Benutzung des physikalisch unklaren σ-Feldes, werden nach Meinung des Autors bei weitem überdeckt durch die Vorteile: die Ermöglichung einer *event-by-event*, mikroskopischen, dynamischen Simulation, die farbelektrische Flussschläuche simuliert, die dem Modell nicht etwa elementar sind, sondern durch einen dynamischen Prozess formiert werden. Die in dieser Dissertation vorgestellte Simulation kann (in weiterentwickelter Form) dazu dienen, Hadronisierungsszenarien zu studieren, bis echte dynamische QCD-Gitterrechnungen mit Computern machbar werden. Wie lange es dauern wird, bis letzteres möglich sein wird, ist noch unbestimmt.

1.3 Gliederung

- Der erste Abschnitt dieser Einleitung zielte darauf ab, dem Laien die fachliche Einordnung dieser Dissertation zu erleichtern. Der zweite Abschnitt sollte die gleiche Aufgabe für den Fachmann erfüllen. Dieser dritte Abschnitt dient als Wegweiser für die restliche Dissertation.

- Das zweite Kapitel mit dem Titel *"Hadronen, Strings und Quantenchromodynamik"* soll die für diese Arbeit relevanten Ideen und

Entwicklungen der Hadronen- und Elementarteilchenphysik im Überblick darstellen.

- Das dritte Kapitel trägt die Überschrift *"Das chromodielektrische Modell"*. Hier sollen grundlegende Eigenschaften des chromodielektrischen Modells (CDM) erläutert werden; so wird z. B. das Farbconfinement im CDM explizit nachgewiesen. Außerdem werden einige neue Gedankengänge verfolgt, die teilweise sogar erst während des Aufschreibens entstanden sind, nichtsdestotrotz aber von großer Bedeutung für unsere Interpretation des Modells sind. Das dritte Kapitel spielt damit eine zentrale Rolle in dieser Dissertation.

- Im vierten Kapitel werden wir technischer. Hinter dem Titel *"Computersimulation des Modells"* verbirgt sich eine mit 60 Seiten immer noch knappe Erläuterung der Probleme und Lösungsansätze, die während der mehrjährigen Entwicklung des Computerprogramms CDM aufgetaucht sind. Nach der Wiederholung der Bewegungsgleichungen und Modellapproximationen werden die für die Numerik unabdingbaren Näherungen diskutiert. Die Verfahren zur Propagation der Farbladungen und des σ-Feldes sind etabliert, trivial und werden kurz zusammengefasst. Danach findet der Leser eine Einführung in das adaptive hierarchische Finite-Elemente-Mehrgitterverfahren, das in CDM zum Einsatz kommt. Hier werden die drei wesentlichen Ingredienzien der Methode in ihrem Zusammenspiel erklärt: (i) Die Finite-Elemente-Methode, (ii) die adaptive Netzverfeinerung mittels Simplex-Bisektion (der vorgestellte Algorithmus ist erst während dieses Projekts als "Nebenprodukt" entstanden), und (iii) das hierarchische Mehrgitterverfahren. Dabei wird besondere Betonung auf ein intuitives Verständnis der Methode gelegt. Es finden sich auch zahlreiche Hinweise auf die konkrete Realisierung in CDM, in der Regel mit expliziten Verweisen auf Programmstellen.

- Das fünfte Kapitel "Resultate" stellt, nach der Erläuterung unserer Wahl aller Modellparameter, die Forschungsergebnisse vor, die als Teil dieser Dissertation in [36] publiziert wurden und über die in mehreren Vorträgen berichtet wurde. Ergänzt wird dies durch zusätzliche Grafiken und Betrachtungen.

- Im sechsten Kapitel "Zusammenfassung und Ausblick" findet der Leser ebendies: ein Resümee dieser Arbeit, gefolgt von einer Diskussion der weiteren evolutionären (oder revolutionären) Entwicklung, die sich an diese Dissertation anschließen könnte.

- Anhang A enthält den detaillierten Beweis des Klassifikationssatzes irreduzibler weißer Cluster (IWCs), der in Kapitel 3 (Abb. 3.5) diskutiert wird. Die langwierige Fallunterscheidung, die dem Beweis zugrundeliegt, wurde zum Teil in diesen Anhang verlegt.

- Der Anhang B stellt einen Wegweiser durch den CDM-Programmcode dar. Natürlich ist es nicht möglich, den gesamten Programmcode abzudrucken — mit über 700 Seiten hat er die mehrfache Länge des vorliegenden Textes. Der Überblick soll dem Benutzer und Weiterentwickler des Codes dazu dienen, sich in den über 37.000 Programmzeilen, 107 Files und 125 Klassen zurechtzufinden. Dabei wird naturgemäß auf die Grobstruktur des Programms Wert gelegt; die Feinheiten sollten sich durch Lektüre des kommentierten Quelltextes nachvollziehen lassen. Dieser Anhang fasst auch zusammen, welche Voraussetzungen gegeben sein müssen, um CDM benutzen zu können und welche Schritte ggf. zu unternehmen sind, um es zur Arbeit zu bewegen. Auch werden einige Hilfsprogramme (Tools) vorgestellt, die im CDM-Kontext entwickelt wurden.

- Nach der obligatorischen Literaturliste und einem hoffentlich nützlichen Index findet der Leser in der Danksagung eine Würdigung derjenigen Menschen, die mir bei der Anfertigung dieser Arbeit mit Rat und Tat zur Seite standen.

"Why QCD? Because it is the only game in town",
L. WILETS *[37]*

"I had the sound first, without the spelling... Then, in one of my occasional perusals of 'Finnegans Wake', by James Joyce, I came across the word 'quark' in the phrase 'Three quarks for Muster Mark.'",
M. GELL-MANN *[38]*

2 Hadronen, Strings und Quantenchromodynamik

Dieses Kapitel ist ein knapper Abriss derjenigen Entwicklungen und Konzepte der Elementarteilchenphysik, die für diese Arbeit relevant sind. Für ein ausführlicheres, einführendes Werk sei auf [5] und die darin gegebenen Referenzen verwiesen.

2.1 Stringmodell der Hadronen

Nach der Entdeckung des Neutrons 1932 und des Pions 1947 versuchte man, die starken, kurzreichweitigen Kernkräfte nach Yukawa durch den Austausch virtueller Pionen zu beschreiben. Hochenergetische Streuexperimente zeigten aber bald auf, dass das leichte Pion ($139 MeV$) zur Beschreibung der Wechselwirkung nicht ausreichte. Austauschteilchen höheren Spins und höherer Masse zeigten sich als Resonanzen im Experiment. Schließlich wurden weitere Teilchen direkt beobachtet, die *seltsamen* Teilchen, deren Lebensdauer ausreicht, um sichtbare Spuren in Blasenkammern zu hinterlassen. In den frühen sechziger Jahren war das reiche Massenspektrum der stark wechselwirkenden Teilchen, der *Hadronen*, das größte Rätsel der Elementarteilchenphysik. Die Hadronen werden heute begrifflich in *Mesonen* mit ganzzahligem Spin und *Baryonen* mit halbzahligem Spin eingeteilt.

Allein durch die Analyse der experimentellen Daten entstand die *Regge-Poltheorie* [39], derzufolge gewisse Serien von Resonanzen als Austauschteilchen dienen, die auf sog. *Regge-Trajektorien* liegen. Die führende (leichteste) Trajektorie[1] folgt dabei dem empirischen Gesetz [40]

$$J \propto M^2 \quad . \tag{2.1}$$

Mit anderen Worten, zu gegebenem Spin J hat die leichteste Austauschresonanz eine invariante Masse M^2, die proportional zu diesem Spin ist. Wir können das mit den heute bekannten Hadronendaten leicht nachvollziehen (Abb. 2.1).
Diese Beobachtung lässt sich dadurch erklären, dass man sich die Hadronen als *Strings* vorstellt, das sind geschlossene oder offene "Saiten" ohne jede innere Struktur, die schwingen oder rotieren können. Weder Länge noch Masse noch Drehimpuls (Spin) des Strings ist dabei a priori festgelegt[2]; der String hat lediglich eine bestimmte Masse pro Länge, die *Stringspannung* σ.

Wir können die Beziehung zwischen J und M^2 in einem einfachen klassischen Stringmodell reproduzieren: Ein klassischer String kann rotieren und vibrieren (schwingen). Soll er bei festem Drehimpuls eine möglichst geringe Masse M besitzen, so darf er nur rotieren und nicht zusätzlich auch schwingen. Wir betrachten daher einen schnurgeraden String der Länge s, der mit der Kreisfrequenz ω um sein Zentrum rotiert; der Drehimpuls steht dabei senkrecht auf dem String (Abb. 2.2). Der String hat die Masse

$$\begin{aligned} M &= \int dm \\ &= \int_{-s/2}^{s/2} \frac{\sigma dr}{\sqrt{1-\omega^2 r^2}} \\ &= \frac{2\sigma}{\omega} \arcsin\left(\frac{\omega s}{2}\right) \\ &= \frac{2\sigma}{\omega} \arcsin k \end{aligned} \tag{2.2}$$

und den Drehimpuls

$$\begin{aligned} L &= \int_{-s/2}^{s/2} \frac{r(\omega r)\,\sigma dr}{\sqrt{1-\omega^2 r^2}} \\ &= \frac{\sigma}{\omega^2}\left(\arcsin k - k\sqrt{1-k^2}\right) \end{aligned} \tag{2.3}$$

mit $k := \omega s/2$. Minimieren wir nun die Masse M mit der Nebenbedingung, dass der Drehimpuls $L = J$ bekannt ist, so ergibt sich mit der Methode der Lagrangemultiplikatoren das Gleichungssystem

$$\frac{\partial}{\partial(k,\omega,\lambda)}(M - \lambda(L-J)) = 0 \quad , \tag{2.4}$$

[1]Andere Trajektorien liegen parallel dazu, sind aber zu größeren Massen hin verschoben.

[2]Natürlich unterliegt der Spin des quantenmechanischen Strings den Bedingungen der quantenmechanischen Drehimpulsalgebra, die ihn auf ganz- oder halbzahlige Werte beschränkt.

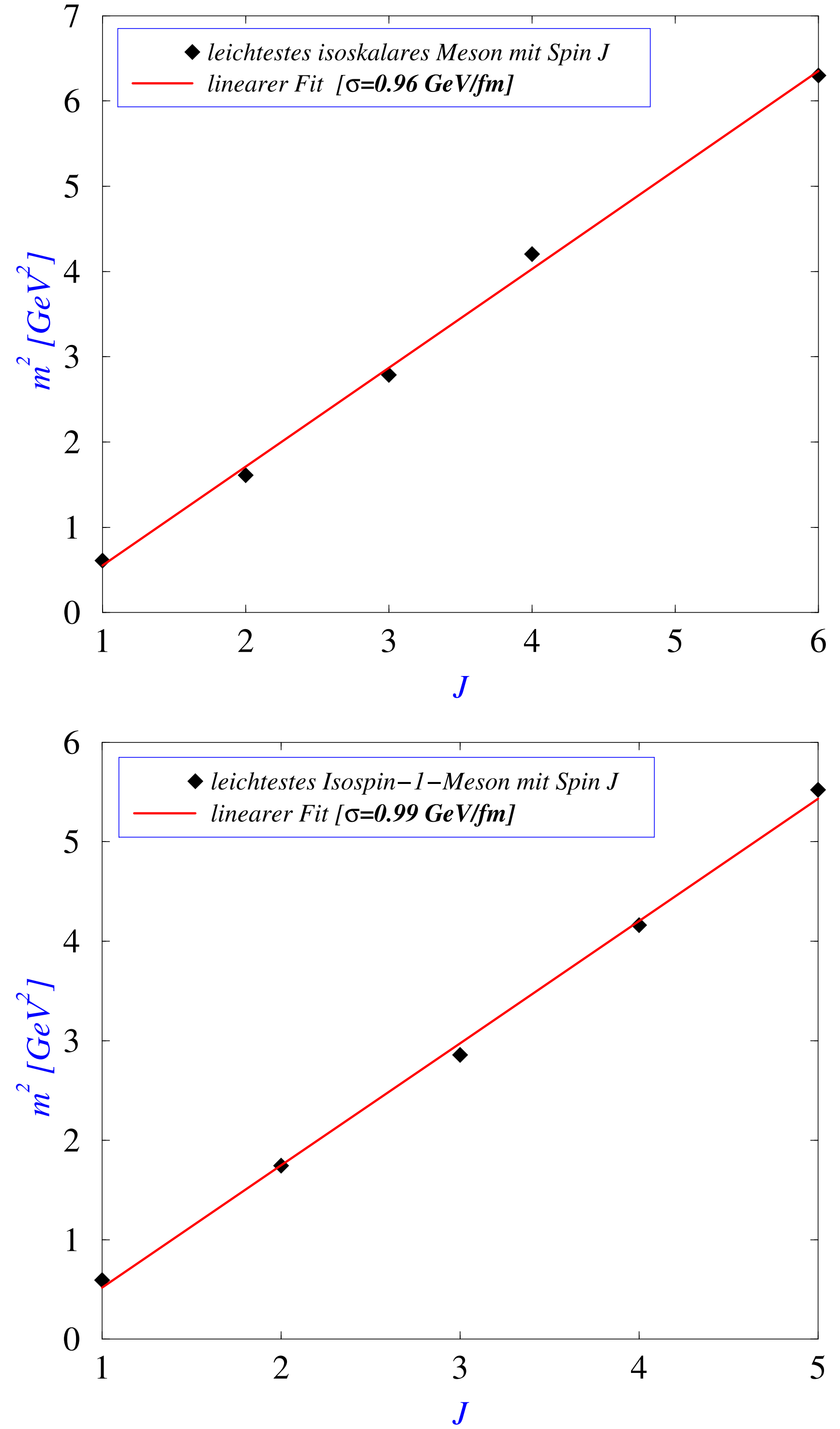


Abbildung 2.1: Zu gegebenem Teilchenspin ist jeweils die invariante Masse des leichtesten Mesons aufgetragen (Chew-Frautschi-Plot); oben für isoskalare, unten für isovektorielle Mesonen. Daten aus [4].

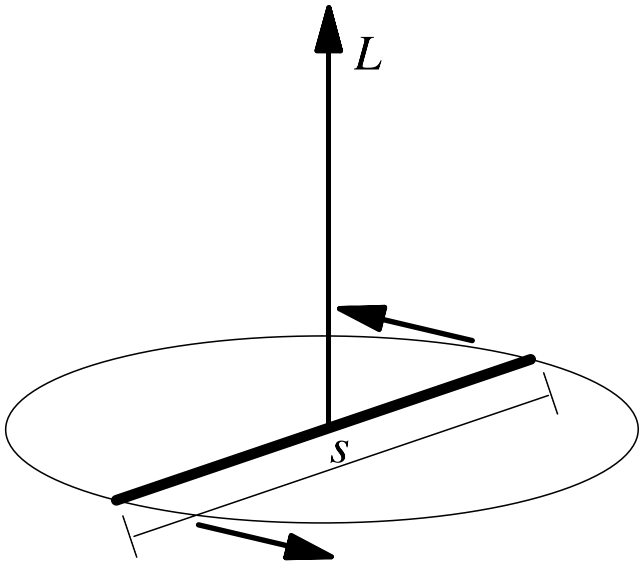


Abbildung 2.2: Ein rotierender klassischer String.

das ausgeschrieben und etwas vereinfacht lautet:

$$\left| \begin{array}{rcl} k\sqrt{1-k^2}\,\lambda - (\lambda-\omega)\arcsin k &=& 0 \\ k^2\lambda - \omega &=& 0 \\ \frac{\sigma}{\omega^2}\left(\arcsin k - k\sqrt{1-k^2}\right) &=& J. \end{array} \right| \tag{2.5}$$

Aus der zweiten Gleichung sieht man $\omega = \lambda k^2$; setzt man das in die erste Gleichung ein, so folgt wieder nach Vereinfachung

$$k\sqrt{1-k^2} + \left(k^2-1\right)\arcsin k = 0 \quad . \tag{2.6}$$

Wie man an Abb. 2.3 sieht, hat diese Gleichung nur die Lösungen $k = 0$ und $k = \pm 1$. Aus physikalischen Gründen gilt $k > 0$, also folgt $k = 1$.
Wir wissen bereits $\omega = \lambda k^2$, also folgt $\omega = \lambda$. Setzen wir alles in die dritte Gleichung des Systems (2.5) und in die Gleichung (2.2) ein, so folgt

$$J = \frac{\pi\sigma}{2\lambda^2} \tag{2.7}$$

$$M = \frac{\pi\sigma}{\lambda} \quad ; \tag{2.8}$$

woraus wir schließen können, dass

$$M^2 = 2\pi\sigma J \tag{2.9}$$

gilt. Man vergleiche das mit der empirisch gewonnenen Beziehung (2.1). Aus den linearen Regressionsgeraden in Abb. 2.1 lässt sich die Stringspannung recht genau zum Wert

$$\sigma = 1\,GeV/fm \tag{2.10}$$

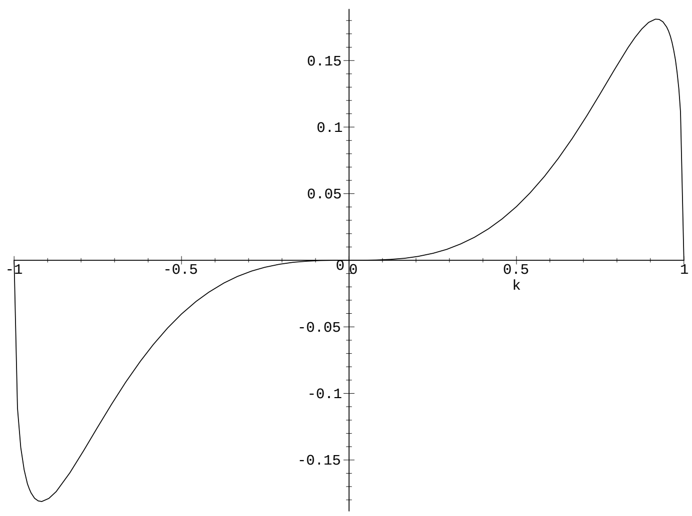


Abbildung 2.3: *Die linke Seite von* (2.6).

ermitteln.

Der Erfolg des Stringbildes zur Erklärung des hadronischen Massenspektrums begründete einen neuen Zweig der theoretischen Physik, die *Stringtheorie*. Zunächst als fundamentale Theorie der Hadronen gedacht, konnte diese jedoch nie mehr als die Grobstruktur der Spektren und der Wechselwirkungen bei hohen Energien erklären. Schließlich wurde sie vom *statischen Quarkmodell* abgelöst, das alle bekannten Hadronen in *Multipletts* klassifizieren konnte und dabei sogar die Existenz weiterer Hadronen vorhersagen konnte. Das statische Quarkmodell hielt auch der experimentellen Überprüfung durch hochenergetische Elektron-Proton-Streuversuche stand, zumindest insofern, dass die Quarks im Proton mit ihren drittelzahligen elektrischen Ladungen nachgewiesen werden konnten.

2.2 Quarks

Das statische Quarkmodell erklärt Hadronen als zusammengesetzte Systeme von *Quarks*. Baryonen (halbzahliger Spin) bestehen aus drei Quarks, Mesonen (ganzzahliger Spin) aus einem Quark und einem Antiquark. Die Quarks kommen dabei in verschiedenen *Flavors* (Sorten) vor, denen man solche Werte für die additiven Quantenzahlen zuordnet, dass sich die gemessenen Werte der Hadronen direkt erklären lassen. Zum Beispiel ist das Neutron (Quarkinhalt udd) elektrisch neutral, während das Proton (uud) die Ladung 1 besitzt. Zu den Quantenzahlen der Quarks siehe Tab. 2.1; für die Antiquarks $\bar{u}$, $\bar{d}$, $\bar{s}$, $\bar{c}$, $\bar{b}$, $\bar{t}$ müssen bis auf die Massen alle Werte mit -1 multipliziert werden.
Die Existenz nahezu punktförmiger drittelzahliger elektrischer Ladun-

Tabelle 2.1: *Quarks und ihre wichtigsten Quantenzahlen [4].*

Flavor	*Ladung*	*Isospin$_z$*	*Strangeness*	*Charm*	*Bottom*	*Top*	*Masse*
up	$+2/3$	$+1/2$	0	0	0	0	$2-8MeV$
down	$-1/3$	$-1/2$	0	0	0	0	$5-15MeV$
strange	$-1/3$	0	-1	0	0	0	$100-300MeV$
charm	$+2/3$	0	0	1	0	0	$1.0-1.6GeV$
bottom	$-1/3$	0	0	0	-1	0	$4.1-4.5GeV$
top	$+2/3$	0	0	0	0	1	$180GeV$

gen im Proton ist auf eindrucksvolle Weise in Experimenten zur tiefinelastischen Elektron-Proton-Streuung nachgewiesen; selbst die Größe dieser Ladungen kann man nachmessen [41].
Die Massen der leichten Hadronen (Mesonen- und Baryonenoktett) lassen sich aber im Konstituentenquarkmodell nicht einfach durch Addition der Quarkmassen erklären. Welche Werte auch immer man den *up*- und *down*-Quarks zuordnet, man kann nicht gleichzeitig $938MeV$ für das Proton (uud), $939.5MeV$ für das Neutron (udd) und $139MeV$ für das π^+ ($u\bar{d}$) einstellen. Aus den ersten beiden Zahlen folgt nämlich eine etwa gleiche Masse von $313MeV$ für beide Quarksorten, und die geringe π^+-Masse ist dafür bereits viel zu leicht. Schaleneffekte können dies nicht erklären, denn für die leichtesten Hadronen eines bestimmten Spin-/Isospinkanals muss natürlich gelten, dass die räumliche Quarkwellenfunktion vollständig symmetrisch in einem $1s$-Grundzustand ist.
Also müssen die erheblichen Massendifferenzen mit den starken Bindungskräften zwischen den Quarks zusammenhängen. Diese Bindungskräfte sind immerhin so stark, dass Quarks nie als freie Teilchen beobachtet wurden.
Für das Dekuplett der leichtesten Spin-3/2-Baryonen funktioniert ein Konstituentenquarkmodell allerdings recht gut: diese Baryonen haben alle eine Masse von rund $1250 \pm 70MeV$, und nur das $\Omega(sss)$ fällt mit $1600MeV$ heraus. Die Annahme, dass die Quarks dieser Baryonen alle im $1s$-Grundzustand sind, führt jedoch z. B. beim $\Delta^{++}(uuu)$ zu einem Problem: Dessen drei Quarks befinden sich offenbar in einem Zustand, der symmetrisch bezüglich Flavor, Spin und Ortsraum[3] ist. Da die Quarks Spin-1/2-Teilchen sind, kann dies nur dann mit Paulis *Ausschließungsprinzip* in Einklang stehen, wenn es eine weitere Quantenzahl gibt, in der die drei Quarks sich unterscheiden. Eine solche abstrakte Quantenzahl, die *Farbe*, wurde bereits in den sechziger Jahren po-

[3] Die recht nahe beieinanderliegenden Massen des Baryondekupletts lassen darauf schließen, dass die Quantenzahlen der Ortsraumwellenfunktionen gleich sind; d.h. alle Quarks befinden sich in einem “$1s$-Zustand”. Die Massenunterschiede der Baryonen des Dekupletts lassen sich recht gut durch die unterschiedlichen Quarkmassen erklären.

stuliert. Quarks nehmen eine der drei Farben rot, grün oder blau an; Antiquarks die jeweiligen Antifarben. Der Beitrag der Quarks[4] zur Lagrangedichte des Quarkmodells

$$\mathcal{L} = \sum_{\text{flavor } f,\text{ color } i} \bar{\Psi}_i^f \left(i\partial\!\!\!/ - m^f\right) \Psi_i^f \tag{2.11}$$

ist exakt invariant unter *globalen* $SU(3)_C$*-Transformationen*

$$\begin{aligned} q_i &\rightarrow Uq \\ \bar{q}_i &\rightarrow \bar{q}U^\dagger. \end{aligned} \tag{2.12}$$

Hier steht U für eine beliebige $SU(3)$-Matrix, und der *Feynman-Dagger*

$$\partial\!\!\!/ := \gamma^\mu \partial_\mu \tag{2.13}$$

mit den bekannten Diracschen γ^μ-Matrizen wurde als Abkürzung verwendet. Diese Symmetrie im Farbraum nennt man $SU(3)_C$*-Symmetrie*. Das Postulat, dass die physikalischen Hadronen Farbsingulettzustände sein müssen, erklärt mit Hilfe der Farbe auf anschauliche Weise, warum keine freien Quarks oder Diquarks im Experiment beobachtet werden. Eine Begründung für dieses Postulat kann das statische Quarkmodell allerdings nicht bieten.
Unter $SU(3)$-Transformationen im dreidimensionalen Flavorraum der *up*, *down* und *strange*-Quarks ist (2.11) nur näherungsweise invariant — der Massenterm (und eine mögliche Ankopplung des elektromagnetischen Feldes) brechen diese $SU(3)_F$*-Symmetrie*. Die größere Flavorsymmetriegruppe $SU(4)_F$, die das *charm*-Quark miteinbezieht, ist wegen dessen hoher Masse stark gebrochen.

2.3 Quantenchromodynamik (QCD)

Die Theorie der *Quantenelektrodynamik* (QED), die in den vierziger Jahren von Feynman, Schwinger, Tomonaga und anderen geschaffen wurde, hat seitdem viele signifikante Erfolge erfahren. So konnte man mit ihrer Hilfe beispielsweise den *Lamb shift* im Wasserstoffspektrum verstehen, die Rayleighstreuung (die Ursache für das Himmelsblau) erklären, Lebensdauern atomarer Zustände akkurat berechnen und die

[4]Eigentlich ist (2.11) die Lagrangedichte freier Quarks ohne Farbfelder; die Lagrangedichte des statischen Quarkmodells enthielte noch Terme, die die Kräfte (Potentiale) zwischen den Quarks beschreibt. Alternativ könnte auch (2.11) die vollständige Lagrangedichte des statischen Quarkmodells sein, und Randbedingungen am "Hadronbag" würden die Kräfte ersetzen. Aber dies ist vage; eine eindeutige Lagrangedichte des statischen Quarkmodells existiert nicht, da das Modell keine Felddynamik besitzt. Dies soll hier aber nicht vertieft werden; wir verwenden (2.11) auch nur zur Erläuterung der globalen $SU(3)_C$-Symmetrie und somit zur heuristischen "Herleitung" der QCD-Lagrangedichte.

Vakuumpolarisation verstehen. Die QED ist eine lokale Eichfeldtheorie mit der abelschen Eichgruppe $U(1)$. Ihre Erweiterung auf die nichtabelsche Eichgruppe $U(1) \times SU(2)$ wurde Ende der sechziger Jahre von Glashow, Weinberg und Salam vorgenommen und ist als Theorie der elektroschwachen Wechselwirkung ähnlich erfolgreich gewesen — ein Ergebnis war bspw. die Vorhersage der $W^{\pm}$- und Z^0-Bosonen, die schließlich am CERN nachgewiesen wurden.
Um die starken Kräfte zu beschreiben, die die Quarks im Hadron binden, wurde das erfolgreiche Konzept der Eichfeldtheorie erneut aufgegriffen. Als Eichgruppe diente die Farbsymmetriegruppe $SU(3)_C$, die bereits als globale Symmetriegruppe im statischen Quarkmodell vorhanden ist. Durch Hinzufügen eines Eichfeldes, dessen Teilchen *Gluonen* genannt wurden, entstand so die Lagrangedichte der QCD,

$$\mathcal{L} = \sum_{\text{flavor } f} \bar{q}^f \left(iD - m_f\right) q^f - \frac{1}{4} F^a_{\mu\nu} F^{a\mu\nu}. \tag{2.14}$$

Die Quarkspinoren tragen hier neben dem versteckten Spinorindex noch einen ebenfalls unterdrückten Farbvektorindex, der die Werte $\{1,2,3\}$ durchläuft. Es handelt sich also um drei Felder im Farbraum, die gemeinsam einen Vektor der Fundamentaldarstellung der $SU(3)$ formen.
Der *kovariante Ableitungsoperator*

$$iD_\mu := i\partial_\mu - gA_\mu \tag{2.15}$$

enthält eine *minimale Kopplung* an das Farbfeld

$$A_\mu = A^a_\mu t^a \quad . \tag{2.16}$$

Dabei sind die t^a $(a = 1 \ldots 8)$ Erzeugende der Liegruppe $SU(3)$. In der Fundamentaldarstellung gleichen sie den Gell-Mann-Matrizen λ^a bis auf einen Faktor $1/2$:

$$t^a = \frac{\lambda^a}{2} \quad . \tag{2.17}$$

Die Gell-Mann-Matrizen wiederum sind üblicherweise wie folgt definiert:

$$\begin{aligned}
\lambda^1 &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \lambda^2 &= \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
\lambda^4 &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} & \lambda^5 &= \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix} \\
\lambda^6 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} & \lambda^7 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix} \\
\lambda^3 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \lambda^8 &= \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.
\end{aligned} \tag{2.18}$$

Es gilt die $SU(3)_C$-Lie-Algebra

$$\left[t^a, t^b\right] = i f^{abc} t^c \tag{2.19}$$

mit den vollständig antisymmetrischen Strukturkonstanten f^{abc} der $SU(3)_C$,

$$\begin{aligned} f^{123} &= 1 \\ f^{147} &= f^{246} = f^{257} = f^{345} = \frac{1}{2} \\ f^{156} &= f^{367} = -\frac{1}{2} \\ f^{458} &= f^{678} = \frac{\sqrt{3}}{2} \quad . \end{aligned} \tag{2.20}$$

Im übrigen gilt die Normierung

$$\mathrm{tr}\left(\frac{\lambda^a}{2}\frac{\lambda^b}{2}\right) = \frac{\delta^{ab}}{2} \quad . \tag{2.21}$$

Die drei Farben der Quarks koppeln also an die acht Gluonenfelder A_μ^a über einen Kopplungsterm in der kovarianten Ableitung.
Der gluonische Feldtensor in (2.14) ist durch den Kommutator der kovarianten Ableitungen definiert:

$$\begin{aligned} F_{\mu\nu} &= \frac{i}{g}\left[D_\mu, D_\nu\right] \\ &=: F_{\mu\nu}^a t^a \quad . \end{aligned} \tag{2.22}$$

$F_{\mu\nu}$ ist, wie A_μ, im Farbraum eine 3×3-Matrix, die sich in die t^a entwickeln lässt. Die Koeffizienten $F_{\mu\nu}^a$ bzw. A_μ^a formen dann einen achtdimensionalen Vektor der *adjungierten Darstellung* der $SU(3)$.
Die resultierende Lagrangedichte (2.14) ist skalar sowohl unter Lorentztransformationen als auch unter *lokalen $SU(3)_C$-Transformationen*

$$\begin{aligned} q &\to e^{i\epsilon_a(x)t^a} q \\ \bar{q} &\to \bar{q} e^{-i\epsilon_a(x)t^a} \\ A_\mu &\to e^{i\epsilon_a(x)t^a}\left(A_\mu - \frac{1}{g}\left(\partial_\mu \epsilon_a(x)\right) t^a\right) e^{-i\epsilon_a(x)t^a} \\ \boldsymbol{D} q^f &\to e^{i\epsilon_a(x)t^a} \boldsymbol{D} q^f \\ \bar{q}^f \boldsymbol{D} q^f &\to \bar{q}^f \boldsymbol{D} q^f \\ F_{\mu\nu}^a F^{a\mu\nu} &\to F_{\mu\nu}^a F^{a\mu\nu} \end{aligned} \tag{2.23}$$

forminvariant (die Ableitung ∂_μ wirkt hier nur auf die nachfolgende Eichfunktion ϵ_a).

Wir haben die kovariante Ableitung

$$\begin{aligned} D_\mu q &= (\partial_\mu + igA_\mu)q \\ F_{\mu\nu} &= \frac{i}{g}\,[D_\mu, D_\nu] \end{aligned} \tag{2.24}$$

in ihrer Wirkung auf Vektoren der Fundamentaldarstellung der $SU(3)_C$ definiert. Der Vergleich mit dem kovarianten Ableitungsoperator in der Differentialgeometrie

$$\begin{aligned} \nabla_\mu A_\nu &= \partial_\mu A_\nu + \Gamma^\sigma_{\mu\nu} A_\sigma \\ R_{\mu\nu} &= [\nabla_\mu, \nabla_\nu] \end{aligned} \tag{2.25}$$

zeigt eine enge Analogie auf. Die Eichfelder A_μ korrespondieren zu Verbindungskoeffizienten, und der Feldtensor zum Riemannschen Krümmungstensor. Wir wollen diese Analogie hier nicht weiter verfolgen; es sei nur angemerkt, dass die Wirkung der kovarianten Ableitung von der Tensorstruktur des Arguments abhängt: in der Differentialgeometrie ist die kovariante Ableitung von Skalaren, Vektoren und Tensoren zweiter Stufe beispielsweise durch

$$\begin{aligned} \nabla_\mu S &= \partial_\mu S \\ \nabla_\mu A_\nu &= \partial_\mu A_\nu + \Gamma^\sigma_{\mu\nu} A_\sigma \\ \nabla_\mu T_{\nu\lambda} &= \partial_\mu T_{\nu\lambda} + \Gamma^\sigma_{\mu\nu} T_{\sigma\lambda} + \Gamma^\sigma_{\mu\lambda} T_{\nu\sigma} \end{aligned} \tag{2.26}$$

gegeben. In der Eichfeldtheorie ist analog die kovariante Ableitung von Farbskalaren (Singletts), dreidimensionalen Vektoren der Fundamentaldarstellung und achtdimensionalen Vektoren $F^a_{\mu\nu}$ der adjungierten Darstellung:

$$\begin{aligned} \mathcal{D}_\mu S &= \partial_\mu S \\ \mathcal{D}_\mu q &= D_\mu q \\ &= (\partial_\mu + igA_\mu)q \\ \mathcal{D}_\mu F^a &= [D_\mu, F^a] \\ &= \partial_\mu F^a + igA^b_\mu f^{abc} F^c \quad . \end{aligned} \tag{2.27}$$

Hier übernehmen die Strukturkonstanten die Rolle, die in der Fundamentaldarstellung die Einträge der Gell-Mann-Matrizen gespielt haben.

2.4 Confinement

Der gluonische Feldtensor aus (2.14) wurde in (2.22) definiert als

$$F_{\mu\nu} = F^a_{\mu\nu} t^a := \frac{i}{g}[D_\mu, D_\nu] \quad .$$

Da die Gell-Mann-Matrizen nicht kommutieren, ist der Feldtensor nichtlinear in den Potentialen. Dies führt bereits auf der klassischen Ebene

Abbildung 2.4: *Links: Gebündelte farbelektrische Feldlinien der QCD. Rechts: normale elektrische Feldlinien (schematisch).*

zu einer Selbstwechselwirkung der Farbfelder — die Felder tragen selbst Ladung. Eine schwerwiegende Folge dessen ist die erhebliche Verkomplizierung der Feldgleichungen: das nichtabelsche Gegenstück der *Maxwellgleichungen*, die *Yang-Mills-Gleichungen*, ist nichtlinear und daher auch numerisch schwer zu lösen [35].

Auf quantenmechanischer Ebene führt die Selbstwechselwirkung der Farbfelder (Gluonen) zu neuen Phänomenen, beispielsweise der asymptotischen Freiheit: die farbelektrische Ladung wird vom umgebenden Vakuum verstärkt statt, wie die elektrische Ladung, abgeschirmt. Das hat zur Folge, dass auf hohen Energieskalen die effektive Kopplungskonstante kleiner wird, da dort die nichtperturbativen Vakuumeffekte eine geringere Rolle spielen. Diesen Effekt nennt man asymptotische Freiheit: auf hohen Energieskalen wechselwirken die Quarks schwächer. Perturbative Berechnungen sind daher in der QCD sehr erfolgreich zum Verständnis hochenergetischer e^+e^--Kollisionen eingesetzt worden. Betrachtet man bei solchen Experimenten nämlich nur inklusive hadronische Observablen, so ist nur der erste "harte" Prozess $e^+e^- \rightarrow q\bar{q}$ maßgeblich, und für diesen ist eine perturbative Rechnung zulässig.

Auf dem Niveau perturbativer Rechnungen konnte man jedoch keine Erklärung für das *Confinement*, die Gefangenschaft der Quarks in Hadronen, finden. Das leuchtet ein: schließlich ist das Confinement ein Phänomen, das eine langreichweitige, starke Wechselwirkung voraussetzt, und der Erfolg des hadronischen Stringmodells legt nahe, dass diese Wechselwirkung ein mit der Entfernung linear ansteigendes Potential besitzt (konstante Stringspannung). Eine solche Wechselwirkung kann nicht störungstheoretisch behandelt werden — Störungstheorie im diagrammatischen Stil, wie sie von Feynman geprägt wurde, kann nur auf schwach wechselwirkende Teilchen angewandt werden.

Es ist heute eine allgemein akzeptierte Vorstellung, dass quantenmechanische Effekte die farbelektrischen Feldlinien zwischen Farbladungen zu einem Flussschlauch "bündeln" (Abb. 2.4). Das erklärt den Erfolg des Stringbildes ebenso wie das Quarkconfinement. Tatsächlich wurde

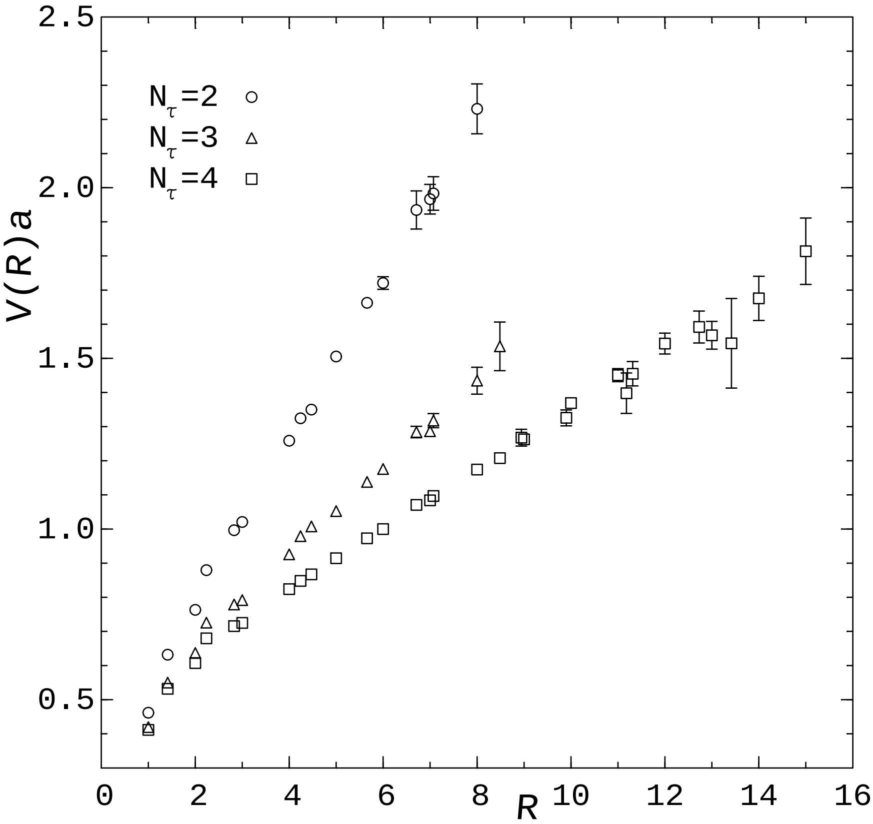


Abbildung 2.5: Statische QCD-Gitterrechnung zur Stringkonstanten. Das linear ansteigende $q\bar{q}$-Potential deutet auf das Vorhandensein eines Flussschlauchs hin. Das verwendete Gitter hat die Größe $32^3 \times N_\tau$, wobei Graphen zu mehreren Werten von N_τ hier zusammengefasst sind. (Abb. aus [8] entnommen)

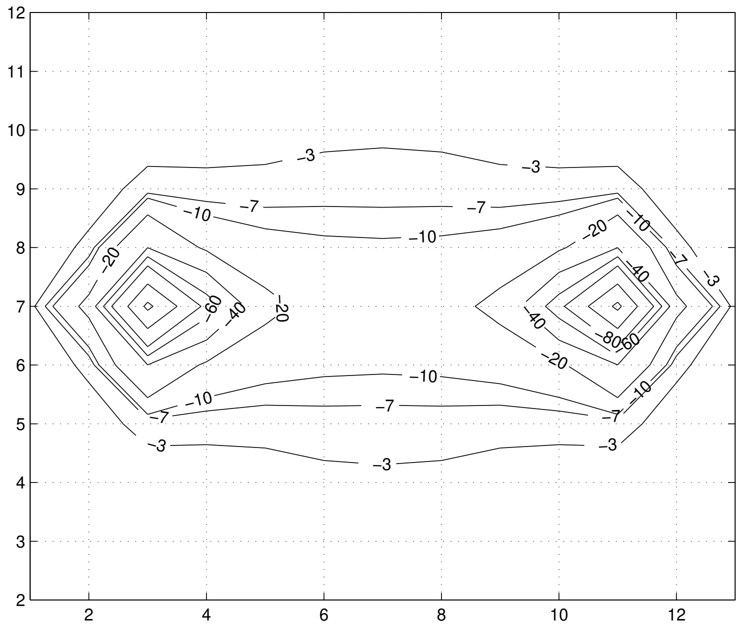


Abbildung 2.6: *Statische QCD-Gitterrechnung zur Energiedichte in einem $q\bar{q}$-Flussschlauch. (Abb. aus [42] entnommen)*

ein linear ansteigendes Quarkpotential in *QCD-Gitterrechnungen* nachgewiesen (siehe [8], dem Abb. 2.5 entnommen wurde), und sogar die schlauchförmige Verteilung der Feldenergie konnte nachgewiesen werden [42, 43] (siehe Abb. 2.6).

Allerdings sind QCD-Gitterrechnungen numerisch extrem aufwendig, und obwohl Flussschläuche nachgewiesen wurden, konnten die wesentlichen, dabei beteiligten Freiheitsgrade nicht identifiziert werden: der *Mechanismus* des Confinement ist nach wie vor unklar. Wir wissen lediglich, dass das Confinement ein quantenmechanischer Effekt ist — in der klassischen Chromodynamik existiert er nicht.

Dies kann man sehr einfach mit folgender Überlegung einsehen: die nichtabelsche Eichgruppe $SU(3)_C$ der QCD besitzt abelsche Untergruppen, z.B. die von λ^3 erzeugte eindimensionale Liegruppe. Nimmt man nun eine klassische Ladungsverteilung an, die nur Ladungsdichten proportional λ^3 enthält, so reduzieren sich die Yang-Mills-Gleichungen auf einfache Maxwellgleichungen. Die Lösungen $F^a_{\mu\nu}$ dieser Feldgleichungen verschwinden dann überall für $a \neq 3$, und $F^3_{\mu\nu}$ ist einfach gleich dem entsprechenden Maxwellfeld, das zu dieser Ladungsverteilung gehört, ohne jede Bündelung von Feldlinien. Damit kann die klassische Chromodynamik keine Flussschläuche erzwingen und das Quarkconfinement nicht erklären.

Von QCD-Gitterrechnungen her ist allerdings bekannt, dass das Farbconfinement nicht wirklich absolut ist: bei Überschreitung einer sehr hohen kritischen Temperatur oder Energiedichte tritt der *Deconfinement-*

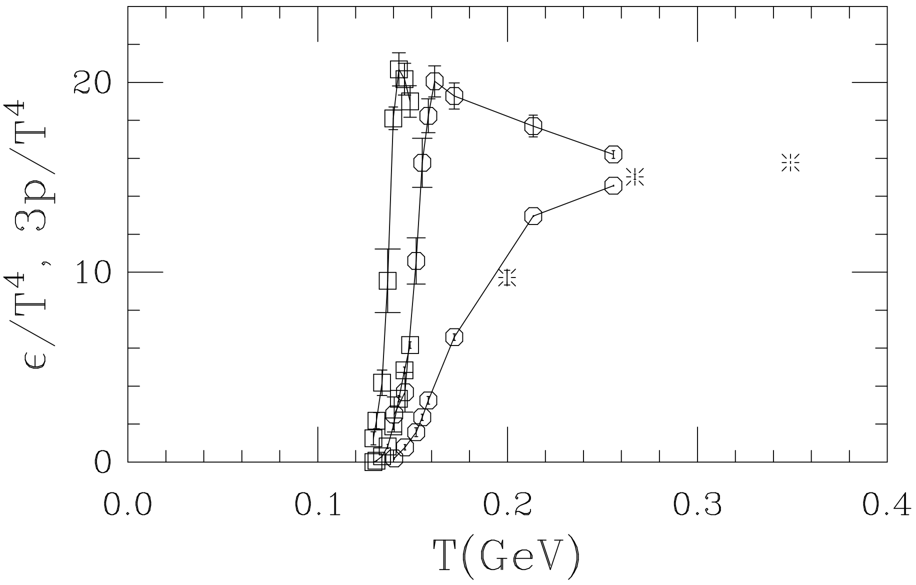


Abbildung 2.7: Gitterrechnung zur Energiedichte (obere Kurven) und freie Energiedichte (Druck, untere Kurve) des QCD-Vakuums bei hohen Temperaturen. Der Unterschied der beiden oberen Kurven ist die verwendete Quarkmasse: $m_q = T/4$ *für die quadratischen Datenpunkte, und* $m_q = T/10$ *für die achteckigen Datenpunkte, was für eine Temperatur* T *jenseits des Phasenübergangs realistischer ist. Die Sterne stellen die Extrapolation der freien Energiedichte hin zu verschwindender Quarkmasse dar. (Abb. aus [8] entnommen)*

Phasenübergang auf, oberhalb dessen die Quarks und Gluonen quasifrei sind. Diese Materiephase nennt man das *Quark-Gluon-Plasma* (QGP). Sie ist durch eine drastische Erhöhung der freien Energiedichte (=Vakuumdruck) gekennzeichnet, die auf die erhöhte Zahl der Freiheitsgrade in der Plasmaphase zurückzuführen ist. Man betrachte hierzu Abb. 2.7: deutlich erkennbar ist eine (ausgewaschene) Stufe in der freien Energiedichte bei einer kritischen Temperatur von etwa $150 - 200 MeV$.

2.5 Modelle des Confinements

Ein physikalisches Modell unterscheidet sich typischerweise von einer Theorie in folgender Hinsicht: das Modell ist einfacher, hat wesentlich weniger Freiheitsgrade, und eine gewisse Abweichung der Resultate von der Realität wird in Kauf genommen. Von der Theorie hingegen wird erwartet, dass sie die Wirklichkeit beschreibt, was in der heutigen Elementarteilchenphysik ziemlich komplizierte Ansätze verlangt. Ein gutes Modell ist viel einfacher als die Theorie, ist aber in der Lage, die wesentlichen Phänomene der Theorie zu reproduzieren. Verfügt der Physiker über gute Modelle, so kann er ein intuitives Verständnis der Theorie und damit der Natur erlangen.

Besitzt man nun ein Modell, so gibt es prinzipiell zwei Forschungsaufgaben im Kontext dieses Modells: Erstens kann man versuchen, die Gleichungen des Modells durch begründbare Approximationen aus der Theorie herzuleiten; dabei stellt sich oft heraus, dass das Modell nur in bestimmten Bereichen (Temperaturen, Dichten) Gültigkeit besitzt. Zweitens kann man die Phänomenologie des Modells studieren, in der Annahme, die Resultate lassen sich auf die Theorie übertragen.

QCD-Gitterrechnungen (die "Theorie" in unserem Falle) sind extrem rechenzeitaufwendig, weil sie auf der direkten Auswertung von Pfadintegralen mit Hilfe eines Monte-Carlo-Verfahrens beruhen. Selbst dieser Ansatz ist aber schon mit Approximationen behaftet: neben der kleinen Gittergröße ist vor allem die Einbeziehung von Quarks in die Rechnungen zu nennen, die nur näherungsweise geschieht (*quenched approximation*). Solange es praktisch nicht möglich ist, dynamische Simulationen der QCD auf hinreichend feinen Gittern durchzuführen, sind die Physiker daher auf die Modellbildung angewiesen.

Im Rahmen der Themenstellung dieser Arbeit sollen im folgenden vor allem Modelle der QCD diskutiert werden, die das Ziel verfolgen, das Farbconfinement der QCD dynamisch zu beschreiben. Diese Einschränkung schließt Modelle wie das Nambu-Jona-Lasinio-Modell [24] oder das chirale Bagmodell von unserer Betrachtung aus, denn diese beschreiben zwar einen chiralen Phasenübergang — ebenfalls ein QCD-Phänomen — aber nicht das hier interessierende Farbconfinement. Ebenso ist das MIT-Bag-Modell hier nicht von Interesse; es beschreibt zwar Confinement, aber auf eine statische Weise unter Zuhilfenahme von externen Randbedingungen, und es ist dadurch keine echte lokale Feldtheorie, die allein durch eine Lagrangedichte (mit Randbedingungen im Unendlichen) definiert wird. Einen Überblick über diese erwähnten Modelle findet man in [30].

Als naheliegendes "Modell" der QCD bietet sich zunächst die klassische Version der QCD an: die klassische Chromodynamik (CCD), die auch von Stefan Loh in seiner Dissertation betrachtet wurde [35]. Die Feldgleichungen der CCD sind die Yang-Mills-Gleichungen, die im Vergleich zu den Maxwellgleichungen einen nichtabelschen Charakter aufweisen. Es ist eine naheliegende Frage, ob dieser nichtabelsche Charakter, verbunden mit einer großen Kopplungskonstante, zu Confinement führen kann. Wir haben allerdings im letzten Abschnitt mit einer simplen Überlegung gezeigt, dass es in der CCD immer auch abelsche ("Coulomb-") Konfigurationen ohne Farbconfinement geben muss. Mandula und Magg [44, 45] fanden heraus, dass die Coulomblösung sogar stabil ist, solange die Kopplungskonstante klein genug ($\alpha_S < 3/2$) ist. Im niederenergetischen Bereich trifft dies nicht mehr zu; dort wird die Kopplungskonstante groß, die Coulomblösung folglich instabil und nichtabelsche Lösungen übernehmen, die auch schon in den 70er Jahren berechnet werden konnten [46, 47, 48]. Natürlich liegen diese energetisch sogar noch tiefer (sonst wäre die Coulomblösung ja stabil) und unterstützen damit kein Confinement, sondern dienen vielmehr der Ladungsabschirmung (*screening* statt *anti-screening*). Somit ist die CCD als “Modell” der QCD unbrauchbar, zumindest wenn man das Confinement studieren und verstehen will.

Der naheliegende Ausweg ist die Konstruktion eines Modells, das Strings (Flussschläuche) bereits auf der klassischen Ebene produziert und dabei an die CCD-Lagrangedichte angelehnt ist. Das Friedberg-Lee-Modell [25, 26, 27], eine relativistisch-dynamische Verallgemeinerung des MIT-Bag-Modells[5], beschreibt Hadronen als nontopologische Solitonenbags mit darin enthaltenen Quarks. Dabei werden die mutmaßlichen dielektrischen Eigenschaften des QCD-Vakuums als Ausgangspunkt für die Konstruktion des Modells genommen. In diesem Bild besitzt das physikalische QCD-Vakuum eine farbdielektrische Zahl und farbmagnetische Suszeptibilität von $\epsilon = 1/\mu = 0$. Dadurch ist bei endlichem farbelektrischen und -magnetischen Feldern $\vec{E}^a$, $\vec{B}^a$ stets gewährleistet, dass kein farbelektrischer Fluss $\vec{D}^a = \epsilon E^a$ durch das physikalische Vakuum fließen kann[6]. Dies führt zur Gefangenschaft von Farbladungen, da die Einzelquarkkonfiguration verboten ist. Zwischen entgegengesetzten Farbladungen muss aber ein farbelektrischer Fluss fließen dürfen, was im Friedberg-Lee-Modell nur unter Zerstörung der Struktur des physikalischen Vakuums möglich wird. Es bildet sich ein Flussschlauch aus, durch den das Farbfeld strömen kann.

Man kann hier eine Analogie zum bekannten Typ-II-Supraleiter bilden: dort kondensieren Cooperpaare von Elektronen, und das resultierende Medium verbietet Farbfeldlinien, so dass in solch einem Medium magnetische Monopole gefangen (*confined*) wären: der einzelne Monopol hätte unendliche Energie, und zwischen Monopolen würden sich magnetische Flusschläuche bilden, entlang derer die Supraleitung zusammenbrechen würde. Analog dazu nun das QCD-Bild des dualen Typ-II-Farbsupraleiters: hier kondensieren Cooperpaaren von farbmagnetischen Monopolen, und dafür werden farbelektrische Felder verdrängt, die einzelne Farbladung erhält eine unendliche Energie. Zwischen Farbladungen bilden sich farbelektrische Flussschläuche, wobei die supraleitende Eigenschaft des Vakuums entlang des Flusschlauchs zerstört wird. Auch hier wird also erst durch Änderung der Vakuumstruktur ein Farbfeldfluss zulässig.

Auch wenn das letztere Bild durch seine elegante Querverbindung zur Festkörperphysik vielleicht theoretisch ansprechender wirkt: es ist von seinen phänomenologischen Konsequequenzen her *identisch* mit dem

[5]Das MIT-Bag-Modell folgt als Grenzfall der Friedberg-Lee-Lagrangedichte [27].

[6]Dieser Mechanismus wird im folgenden Kapitel 3 noch genauer beleuchtet werden. Hier sollen lediglich die verschiedenen Philosophien der Modelle miteinander verglichen werden, ohne allzusehr ins Detail zu gehen.

des Farbdielektrikums ($\epsilon_{\text{vac}} = 0$), das weiter oben erläutert wurde. Für beide Bilder fehlen aber rigorose Herleitungen aus der QCD, auch wenn Versuche unternommen wurden, die farbmagnetischen Monopole und den Ordnungsparameter der dualen Landau-Ginzburg-Theorie im Gluonenfeld der QCD zu identifizieren ([49, 50, 51, 52], siehe auch den Übersichtsartikel von Pirner [53]).
Es soll kurz näher auf zwei verschiedene Arten des chromodielektrischen Modells eingegangen werden, wobei wir teilweise auf das folgende Kapitel vorgreifen. Das Friedberg-Lee-Modell besitzt die Lagrangedichte

$$\begin{aligned}\mathcal{L}^{\text{FL}} &= \sum_f \bar{q}^f \left(iD - m_f - g_S\sigma\right) q^f \\ &\quad -\frac{\kappa(\sigma)}{4} F^a_{\mu\nu} F^{\mu\nu a} \\ &\quad +\frac{1}{2}(\partial_\mu\sigma)(\partial^\mu\sigma) - U(\sigma) \qquad (2.28)\end{aligned}$$

Das skalare Feld σ mit einer nichtlinearen Selbstwechselwirkung $U(\sigma)$ nimmt im Grundzustand den nichtverschwindenden Wert σ_{vac} an (Abb. 3.1). Über die dielektrische Funktion $\kappa(\sigma)$, die an σ_{vac} verschwindet, wird ein farbelektrischer Fluss durch das physikalische Vakuum verhindert. Der lokale Aufbruch dieses Zustands, z. B. für Farbflussschläuche, ist möglich, kostet aber eine gewisse Energiedichte, die physikalisch als Vakuumdruck interpretiert werden kann. Zum Friedberg-Lee-Modell gibt es eine beachtliche Menge an Arbeiten (und sogar ein Lehrbuch [37]), die stationäre hadronische Zustände mit quantisierten Quarks studieren [28, 29, 30, 31].
Das Nielsen-Patkos-Modell [51] wird durch die Lagrangedichte

$$\begin{aligned}\mathcal{L}^{\text{NP}} &= \sum_f \bar{q}^f \left(iD - m_f - \frac{g_S}{\chi}\right) q^f \\ &\quad -\frac{\chi^4}{4} F^a_{\mu\nu} F^{\mu\nu a} \\ &\quad +\frac{1}{2}(\partial_\mu\chi)(\partial^\mu\chi) - m^2\chi^2 \qquad (2.29)\end{aligned}$$

definiert. Wir fassen die Unterschiede und Gemeinsamkeiten beider Modelle kurz zusammen:

- Die enge Verwandtschaft des Friedberg-Lee-Modells (2.28) zum Nielsen-Patkos-Modell (2.29) ist mittels der Identifikation

  $$\begin{aligned}\sigma &= \sigma_{\text{vac}}(1-\chi) \\ \kappa &= \chi^4\end{aligned}$$

  leicht zu erkennen. Der Vakuumwert des χ-Felds beträgt 0, während bei Friedberg und Lee das dazu analoge σ-Feld den Vakuumwert

σ_{vac} annimmt. Dieser Unterschied mag eher technischer Natur sein als inhaltlich bedeutsam, jedoch nimmt man an, dass σ so etwas wie ein Glueball-Kondensat beschreibt (also eine Größe wie $avF^a_{\mu\nu}F^{\mu\nu a}$), das im Vakuum tatsächlich nicht verschwindet. Die Wahl eines Modells mit σ_{vac} ist damit m. E. die natürlichere.

- Das Friedberg-Lee-Modell ist bis auf den κ-Term renormierbar. Beim Nielsen-Patkos-Modell ist darüberhinaus der Term $\bar{q}(g_S/\chi)q$ nichtrenormierbar. Für effektive Feldtheorien mag diese Eigenschaft aber vielleicht als irrelevant angesehen werden.

- Absolutes Farbconfinement kommt im Friedberg-Lee-Modell durch die Form von $\kappa(\sigma)$, das im Vakuumzustand verschwindet, zustande. Das Nielsen-Patkos-Modell leistet das Gleiche über den Kopplungsterm $\chi^4 F^2$. Eine Ladungskonfiguration mit einem einzelnen Quark ist in beiden Modellen (selbst mit $g_S = 0$) mit einer unendlich hohen Energie behaftet. Dies wird in Abschnitt 3.4 gezeigt; der dort für das Friedberg-Lee-Modell gezeigte Beweis lässt sich direkt auf das Nielsen-Patkos-Modell übertragen.

- In der zitierten Form besitzt das Nielsen-Patkos-Modell auch ein absolutes Quarkconfinement, in dem Sinne, dass selbst unter Vernachlässigung des Farbfelds ein einzelnes Quark bereits eine unendlich große Masse im physikalischen Vakuum annimmt. Dies kommt durch den Kopplungsterm $\bar{q}(g_S/\chi)q$. Thomas Vetter hat in seiner Dissertation allerdings auch das Friedberg-Lee-Modell mit einem solchen nichtlinearen Kopplungsterm $\bar{q}g_{\text{eff}}(\sigma)q$ versehen, wobei

  $$g_{\text{eff}}(\sigma) \;=\; g_S\sigma_{\text{vac}}\left(\frac{1}{\kappa(\sigma)} - 1\right).$$

  Dieser Term führt dann auch im Friedberg-Lee-Modell ohne Gluonen zu absolutem Quarkconfinement [33]. Es ist aber fragwürdig, ob ein solches “Quarkconfinement ohne Farbe” (beschrieben durch eine divergente Quarkmasse im Vakuum) der korrekte Ansatz ist, das QCD-Confinement zu modellieren. Confinement rührt in der QCD von der Wechselwirkung der Farbladungen her und ist damit ein Phänomen, das seine natürliche Beschreibung nicht in einer divergenten Selbstenergie, sondern in der Quark-Quark-Wechselwirkung (und damit den Korrelationen) findet.

- Die komplizierte nichtkonvexe Form des Potentials $U(\sigma)$ führt zu einem Phasenübergang des σ-Felds bei hoher skalarer Quarkdichte selbst in Abwesenheit jeglicher lokaler Farbladungsdichte [54, 55]. Dieser Phasenübergang entsteht durch die nichtkonvexe[7] Form des

[7] gemeint ist der Vorzeichenwechsel in $U''(\sigma)$.

U-Potentials und ist daher im Nielsen-Patkos-Modell nicht vorhanden.

- Es fehlen rigorose Herleitungen beider effektiven Lagrangedichten aus der QCD.

Die Unterschiede zwischen den Modellen sind geringer als die Bandbreite der jeweiligen Modellversionen, so dass in der Literatur häufig nur vom "chromodielektrischen Modell" gesprochen wird, und die Wahl einer der beiden Lagrangedichten (2.28) und (2.29) eher eine Frage der persönlichen Präferenz darstellt. Da die beiden Modelle aufeinander abbildbar sind, möchte ich sie im folgenden auch unter dem Begriff *chromodielektrisches Modell* zusammenfassen.
An der dynamischen Realzeitsimulation des Friedberg-Lee-Modells ist bereits längere Zeit im Gießener Institut für Theoretische Physik gearbeitet worden. Dazu wurde eine semiklassische Transporttheorie konstruiert, zunächst in einer Dimension [31], dann in drei Dimensionen ohne Farbe [32, 33]. Von der transporttheoretischen Beschreibung auf Basis der Einteilchen-Phasenraumverteilung nach Thomas Vetter [33] musste mittlerweile wieder Abstand genommen werden, wie Stefan Loh in seiner Dissertation gezeigt hat [35]. Stefan Loh arbeitete mit dem Friedberg-Lee-Modell unter Einbeziehung der Farbfelder, allerdings in der abelschen und quasistatischen Näherung und unter Annahme zylindersymmetrischer Konfigurationen und gewissen weiteren Idealisierungen [34, 35]. Die vorliegende Arbeit erweitert dies zu einer vollen dreidimensionalen Simulation, allerdings immer noch in der quasistatischen und abelschen Approximation. Wir betrachten außerdem das Modell stets mit $g_S = 0$, aus Gründen, die in Abschnitt 3.8 erläutert werden.

Geordi: "Suddenly it's like the laws of physics went right out the window."
Q: "And why shouldn't they? They're so inconvenient!"
STAR TREK: THE NEXT GENERATION,
Episode "True Q", wie von Krauss [56] zitiert

3 Das chromodielektrische Modell

In diesem Kapitel soll das dieser Arbeit zugrundeliegende chromodielektrische Modell (CDM) erläutert werden. Vom theoretischen Aspekt her ist dieses Kapitel daher Zentrum der Arbeit. Es werden einige Eigenschaften des Modells diskutiert, z. B. die Flussschlauchformation und das Farbconfinement, das mathematisch präzise nachgewiesen wird. Vor allem werden aber wirklich originäre Überlegungen angestellt, insbesondere zu den Konsequenzen der klassischen Approximation für punktförmige und ausgedehnte Ladungen, zum Hadronenspektrum im Modell und verwandte Überlegungen für die QCD.

3.1 Lagrangedichte und Eichinvarianz

Die Lagrangedichte des chromodielektrischen Modells kann als Summe dreier Terme geschrieben werden:

$$\mathcal{L}^{qm} = \mathcal{L}_p^{qm} + \mathcal{L}_c + \mathcal{L}_\sigma \quad . \tag{3.1}$$

Der erste Term ist dabei der bilineare Quarkterm

$$\mathcal{L}_p^{qm} = \sum_f \bar{q}^f \left(i\not{D} - m_f \right) q^f \quad , \tag{3.2}$$

der formal identisch mit dem Quarkbeitrag der QCD-Lagrangedichte (2.14) ist. Der Beitrag der Farbfelder ist im Vergleich zur QCD-Lagrangedichte mit einer *dielektrischen Zahl* $\kappa(\sigma)$ multipliziert[1], so dass er

$$\mathcal{L}_c = -\frac{\kappa(\sigma)}{4} F^a_{\mu\nu} F^{\mu\nu a} \tag{3.3}$$

[1]Wir werden in Abschnitt 3.4 sehen, dass tatsächlich eine Analogie zur Elektrodynamik im Medium besteht, wobei κ die Rolle der Dielektrizitätszahl spielt.

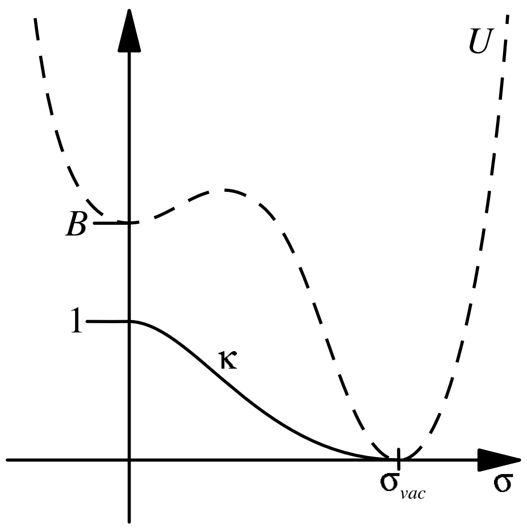


Abbildung 3.1: *Schematische Darstellung der Funktionen $U(\sigma)$ und $\kappa(\sigma)$. Obwohl beide Funktionen in einen Graph gezeichnet sind, ist U eine Energiedichte, während κ dimensionslos ist. $\sigma_{\rm vac}$ bezeichnet den Wert des σ-Feldes im physikalischen Vakuum (Grundzustand).*

lautet, mit dem bekannten gluonischen Feldtensor aus (2.22),

$$F_{\mu\nu} \;=\; F^a_{\mu\nu}\frac{\lambda^a}{2} \;:=\; \frac{i}{g}[D_\mu, D_\nu] \quad .$$

Die Abhängigkeit der Dielektrizitätszahl κ vom σ-Feld ist schematisch in Abb. 3.1 gezeigt. Die Eigendynamik des σ-Feldes ist schließlich durch den Beitrag

$$\mathcal{L}_\sigma \;=\; \frac{1}{2}(\partial_\mu\sigma)(\partial^\mu\sigma) - U(\sigma) \tag{3.4}$$

zur Lagrangedichte bestimmt. $U(\sigma)$ ist ebenfalls schematisch in Abb. 3.1 dargestellt.
Die Lagrangedichte (3.1) ist wie die QCD-Lagrangedichte invariant unter lokalen $SU(3)_C$–Transformationen der Form (2.23). Dabei wird σ als Singlett transformiert, d.h. bleibt unverändert.

3.2 Bewegungs- und Kontinuitätsgleichungen

Die aus (3.1) folgenden Bewegungsgleichungen lauten

$$[D_\mu, \kappa(\sigma)F^{\mu\nu}] \;=\; j^\nu \tag{3.5}$$

$$(i\not{D} - m_f)\, q^f \;=\; 0 \tag{3.6}$$

$$\partial_\mu\partial^\mu\sigma + U'(\sigma) \;=\; -\frac{\kappa'(\sigma)}{4}F^a_{\mu\nu}F^{\mu\nu a} \tag{3.7}$$

mit dem Farbstrom der Quarks,

$$j^\nu \;=\; j^{\nu a}\frac{\lambda^a}{2} \;:=\; g\sum_f \bar{q}^f\gamma^\nu\lambda^a q^f\frac{\lambda^a}{2} \tag{3.8}$$

Man beachte, dass dieser Strom alleine nicht erhalten ist: schreiben wir die Yang-Mills-Gleichung (3.5) um in adjungierte Form,

$$\partial_\mu(\kappa(\sigma)F^{\mu\nu a}) + igf^{abc}A^b_\mu\kappa(\sigma)F^{\mu\nu c} \;=\; j^{\nu a} \tag{3.9}$$

so folgt durch Kontraktion mit ∂_ν die Kontinuitätsgleichung

$$0 \;=\; \partial_\nu\left(j^{\nu a} - igf^{abc}A^b_\mu\kappa(\sigma)F^{\mu\nu c}\right) \quad . \tag{3.10}$$

Physikalisch ist dies so zu deuten, dass das farbgeladene Gluonfeld auch zum Farbstrom beiträgt und nur der gesamte Strom erhalten ist.
Es gilt außerdem die "eichinvariante Kontinuitätsgleichung"

$$\begin{aligned}
[D_\nu, j^\nu] \;&=\; [D_\nu,[D_\mu,\kappa(\sigma)F^{\mu\nu}]] \\
&= D_\nu D_\mu\kappa(\sigma)F^{\mu\nu} + \kappa(\sigma)F^{\mu\nu}D_\mu D_\nu - D_\nu\kappa(\sigma)F^{\mu\nu}D_\mu - D_\mu\kappa(\sigma)F^{\mu\nu}D_\nu \\
&= \frac{1}{2}[D_\nu,D_\mu]\kappa(\sigma)F^{\mu\nu} + \frac{1}{2}\kappa(\sigma)F^{\mu\nu}[D_\mu,D_\nu] - D_\nu\kappa(\sigma)F^{\mu\nu}D_\mu - D_\nu\kappa(\sigma)F^{\nu\mu}D_\mu \\
&= \frac{1}{2}[D_\nu,D_\mu]\kappa(\sigma)F^{\mu\nu} + \frac{1}{2}\kappa(\sigma)F^{\mu\nu}[D_\mu,D_\nu]\underbrace{-D_\nu\kappa(\sigma)F^{\mu\nu}D_\mu + D_\nu\kappa(\sigma)F^{\mu\nu}D_\mu}_{=0} \\
&= -igF_{\nu\mu}\kappa(\sigma)F^{\mu\nu} - ig\kappa(\sigma)F^{\mu\nu}F_{\mu\nu} \\
&= 0 \quad ;
\end{aligned} \tag{3.11}$$

hierbei wurde mehrfach die Antisymmetrie des Feldtensors $F_{\mu\nu}$ ausgenutzt. Die "eichinvariante Kontinuitätsgleichung" (3.11) ist keine Kontinuitätsgleichung im eigentlichen Sinne, wird uns aber dennoch im nächsten Abschnitt nützlich sein. In adjungierter Darstellung lautet sie

$$\partial_\mu j^{\mu a} + igf^{abc}A^b_\mu j^{\mu c} = 0 \quad . \tag{3.12}$$

3.3 Klassische Approximation

Der bilineare Quarkterm in (3.1) ist in quantenmechanischer Form geschrieben, und eine Quantisierung des Quark- und des Farbfelds ist auch tatsächlich möglich. Die Quantisierung des σ-Feldes muss allerdings scheitern: wenn sich die Funktion $\kappa(\sigma)$ überhaupt in eine Potenzreihe entwickeln lässt, so bricht diese jedenfalls nicht nach dem σ^4-Term ab. Die höheren Terme führen zur Nichtrenormierbarkeit divergenter σ-glue-Diagramme.
In der vorliegenden Arbeit zielen wir jedoch auf eine Computersimulation der Zeitentwicklung in recht komplexen Szenarien ab; zur Erreichung dieses Ziels müssen wir uns auf eine klassische Version des Modells beschränken.

3.3.1 Naiver nichtabelscher Ansatz

Für den Entwurf einer solchen klassischen Version des Modells ist das Quarkfeld $q_i(x^\mu)$ mit Farbindex i und Spinorcharakter durch geeignete klassische Freiheitsgrade zu ersetzen.
Wir wählen dafür eine *Menge klassischer Teilchen* $\{P_1 \dots P_N\}$ (Quarks oder Antiquarks). Das Teilchen P_k mit der Masse m_k und Farbladung q_k^a hat dabei eine lokale Dichteverteilung, die am Punkt $\vec{x}_k(t)$ zentriert ist und die Form

$$\rho_N(\vec{x} - \vec{x}_k(t)) \;=\; \left(\frac{1}{2\pi r_0^2}\right)^{3/2} e^{-(\vec{x}-\vec{x}_k(t))^2/2r_0^2} \tag{3.13}$$

besitzt. Um die Ladungs- oder Massendichte dieses Teilchens P_k zu erhalten, wird $\rho_N(\vec{x} - \vec{x}_k(t))$ noch mit der Gesamtladung q_k^a oder Masse m_k skaliert. Der Einfachheit halber soll die Funktion ρ_N für jede Teilchensorte die gleiche starre Form besitzen[2].
Der Quarkterm der Lagrangedichte lautet nun

$$\mathcal{L}_p \;=\; -\sum_k m_k \sqrt{1 - \dot{\vec{x}}_k^{\,2}}\, \rho_N(\vec{x} - \vec{x}_k(t)) - j^{\mu a}(x) A_\mu^a(x) \tag{3.14}$$

mit dem klassischen Farbstrom

$$j^{\mu a}(x) \;=\; g \sum_k q_k^a u_k^\mu \rho_N(\vec{x} - \vec{x}_k(t)) \tag{3.15}$$

$$u_k^\mu \;=\; \begin{pmatrix} 1 \\ \dot{\vec{x}}_k \end{pmatrix} \;=\; \frac{p_k^{\mathrm{kin}\,\mu}}{E_k^{\mathrm{kin}}} \quad . \tag{3.16}$$

Die konkrete Wahl der Farbladungen q_k^a werden wir im Abschnitt 3.3.4 erläutern. Quarkspins werden in (3.14) vernachlässigt (bzw. ausgemittelt). In (3.15) sind der kinetische Impuls

$$\vec{p}_{k,\mathrm{kin}} \;=\; \frac{m\dot{\vec{x}}_k}{\sqrt{1 - \dot{\vec{x}}_k^{\,2}}} \tag{3.17}$$

und die kinetische Energie $E_k^{\mathrm{kin}} = \sqrt{\vec{p}_{k,\mathrm{kin}}^{\,2} + m_k^2}$ benutzt worden.

---

[2]Diese Verteilung hat die Normierung

$$\int d^3x\, \rho_N(\vec{x}) = 1$$

und Varianz

$$\langle \vec{x}^2 \rangle = \int d^3x\, \vec{x}^2 \rho_N(\vec{x}) = 3r_0^2.$$

3.3.2 Diskussion zur Verwendung ausgedehnter Teilchen

Im folgenden wollen wir den im letzten Abschnitt eingeführten Ansatz starrer, ausgedehnter Ladungsverteilungen für die "klassischen Quarks" kritisch hinterfragen und einige Konsequenzen untersuchen.

+ Zunächst ist positiv anzumerken, dass der Gaußsche Ansatz mit *festgehaltener* Breite der Ladungsverteilungen gewisse Probleme mit unphysikalischen Selbstwechselwirkungen umgeht: Wenn Form und Ausdehnung der Ladungsverteilungen nicht festgehalten sind, z. B. in einem transporttheoretischen Ansatz im Einteilchen-Phasenraum (Vlasovgleichung), bewirken die Farbkräfte eine unphysikalische Verformung (Expansion) der Ladungsverteilung. Dies ist von Stefan Loh ([35], S. 82ff) eingehend untersucht worden und soll hier nicht vertieft werden; die vorgestellte "molekulardynamische" Behandlung soll genau diese Probleme vermeiden.

- Leider ist durch die Annahme starrer Ladungsverteilungen die Lorentzinvarianz des Modells gebrochen; nun ist das spezielle Bezugssystem ausgezeichnet, in dem die Ladungsverteilungen sphärisch symmetrisch sind. Für einen kovarianten Ansatz müsste das Argument von ρ_N in (3.14) zunächst in das lokale Bezugssystem des Teilchens transformiert werden, was die Bewegungsgleichungen erheblich verkomplizieren würde, ohne die physikalische Glaubwürdigkeit des Modells wesentlich zu erhöhen. Denn selbst mit dieser Modifikation wäre das Modell nicht vollständig kovariant, da eine kovariante Wechselwirkung aus prinzipiellen Gründen nicht mit starren, ausgedehnten Objekte vereinbar ist. Wir sehen also von der erwähnten Änderung ab.

+ Nimmt man die Teilchen hingegen als punktförmige Ladungen, so kann auch das wechselwirkende Modell vollständig kovariant formuliert werden, dafür tauchen aber konzeptionelle Probleme mit unendlichen Feldenergien auf, ganz abgesehen von Problemen der numerischen Realisierung. Starre, ausgedehnte Ladungsverteilungen sind also tatsächlich ein naheliegender Kompromiss.

- Es gibt noch ein anderes, weitaus größeres konzeptionelles Problem des Quarkterms (3.14) der Lagrangedichte: Er ist nicht lokal $SU(3)_C$-eichinvariant, da im Ausdruck $q_k^a \rho_N(\vec{x} - \vec{x}_k(t))$ explizit Farben an verschiedenen Raumpunkten miteinander in Beziehung gesetzt werden. Nach einer lokalen Eichtransformation wird die veränderte Stromdichte i. a. nicht mehr durch einen solchen Ansatz $q_k^a \rho_N(\vec{x} - \vec{x}_k(t))$ beschreibbar sein.

Der letztgenannte Punkt führt zu einer Inkonsistenz der Bewegungsgleichungen: Wegen

$$\begin{aligned} \partial_\mu u_k^\mu \rho_N(\vec{x} - \vec{x}_k(t)) &= \partial_t \rho_N(\vec{x} - \vec{x}_k(t)) + \dot{\vec{x}}_k \cdot \vec{\nabla} \rho_N(\vec{x} - \vec{x}_k(t)) \\ &= 0 \end{aligned} \tag{3.18}$$

erfüllt der Farbstrom (3.15) einfach durch seine Konstruktion die Kontinuitätsgleichung

$$\partial_\mu j^{\mu a} = 0 \quad . \tag{3.19}$$

Durch Einsetzen dieser Kontinuitätsgleichung in die nichtabelsche Kontinuitätsgleichung (3.10), die aus den Yang-Mills-Gleichungen folgte, gelangt man zur folgenden Zwangsbedingung, die eine direkte Konsequenz unseres Ansatzes ist:

$$0 = \partial_\nu f^{abc} A_\mu^b \kappa(\sigma) F^{\mu\nu c} \quad . \tag{3.20}$$

In der fundamentalen Darstellung liest sich diese Gleichung so:

$$0 = \partial_\nu [A_\mu, \kappa(\sigma) F^{\mu\nu}] \quad . \tag{3.21}$$

Der physikalische Inhalt dieser Gleichung ist nicht leicht zu deuten, immerhin ist aber eine Bedingung erkennbar, die zur Erfüllung von (3.21) hinreicht: dass $A_\mu(x)$ vollständig in einen abelschen Unterraum der $SU(3)$ fällt. Dann befindet sich nämlich auch $F^{\mu\nu}$ sowie alle Ableitungen in diesem Unterraum, und der Kommutator in (3.21) verschwindet. Um diesen Gedanken etwas physikalischer zu formulieren: Der Quarkstrom allein kann nur erhalten sein, wenn die Gluonen im System farbneutral (abelsch) sind.

Wir kommen also zum überraschenden Schluss, dass ein klassisches, "molekulardynamisches" Modell mit Teilchen, die durch starre Farbladungsverteilungen repräsentiert werden, nur in einer abelschen Approximation der Farbdynamik überhaupt Sinn macht. Eine solche abelsche Approximation konstruieren wir im Abschnitt 3.3.4, nachdem im nächsten Abschnitt noch eine kurze Überlegung zur Farbdynamik mit punktförmigen klassischen Teilchen im vollen, nichtabelschen Modell vorgestellt wird.

3.3.3 Klassische Farbdynamik im nichtabelschen Fall

An dieser Stelle soll noch ein möglicher Einwand zur vorangegangenen Überlegung diskutiert werden: Sind die Farbladungen q_k^a der Teilchen nicht selbst als dynamische, also zeitlich veränderliche, Größen zu betrachten?

Die q_k^a wurden als Konstanten der Zeitentwicklung angenommen, da die Lagrangedichte (3.14) keinen kinetischen Term für eine Zeitentwicklung der q_k^a enthält. Allerdings folgt eine Bewegungsgleichung[3] für $q_k^a(t)$, wenn man die Definition des klassischen Farbstroms (3.15) in die "eichinvariante Kontinuitätsgleichung" (3.12) einsetzt. Es folgt

$$\begin{aligned}
&\left(\partial_\mu \delta^{ac} + igf^{abc}A_\mu^b\right) j^{\mu c} \\
&= \left(\partial_\mu \delta^{ac} + igf^{abc}A_\mu^b\right) \left(g \sum_k q_k^a(t) u_k^\mu \rho_N(\vec{x} - \vec{x}_k(t)) \right) \\
&\stackrel{(*)}{=} g \sum_k \left(\partial_t q_k^a(t) + gf^{abc}A_\mu^b(x) u_k^\mu q_k^c\right) \rho_N(\vec{x} - \vec{x}_k(t)) \\
&\stackrel{!}{=} 0 \quad \forall \vec{x} \in \mathbb{R}^3 .
\end{aligned} \tag{3.22}$$

Dabei wurde im Schritt $(*)$ die Gleichung (3.18) benutzt. Gleichung (3.22) ist überbestimmt, da sie als Feldgleichung für alle $\vec{x} \in \mathbb{R}^3$ gültig sein soll, aber nur endlich viele Freiheitsgrade ($q_k^a(t)$) besitzt, deren Dynamik sie bestimmen soll. Als überbestimmte Gleichung ist sie in der Regel nicht erfüllbar. Nur in dem Fall $\rho_N(\vec{x} - \vec{x}_k(t)) \propto \delta(\vec{x} - \vec{x}_k(t))$, also für punktförmige Teilchen, reduziert sich (3.22) auf das Gleichungssystem

$$\frac{\partial q_k^a}{\partial t} + gf^{abc}A_\mu^b(x) u^\mu q_k^c = 0 \quad . \tag{3.23}$$

Mit $dt/d\tau = E_k^{\mathrm{kin}}/m_k$ durchmultipliziert lautet (3.23) einfach

$$\frac{\partial q_k^a}{\partial \tau} + gf^{abc}A_\mu^b(x) \frac{p_k^{\mathrm{kin}\,\mu}}{m_k} q_k^c = 0 \quad , \tag{3.24}$$

die manifest kovariante Form der *Wong-Gleichung*. Da die f^{abc} vollständig antisymmetrisch sind, bewirkt (3.24) eine Präzessionsbewegung des Farbspins um die Achse $A_\mu^b(x) p_k^{\mathrm{kin}\,\mu}$ im Farbraum. Die Wong-Gleichung ist also eine direkte Folge eines Punktteilchenansatzes und der Yang-Mills-Gleichung. Die hier vorgeführte extrem kurze Herleitung entstand während des Zusammenschreibens der Dissertation als Überraschungsprodukt. Die Originalveröffentlichung von S. K. Wong [57] verwendet einen unsauberen semiklassischen Weg der Beweisführung, der zudem wesentlich länger ist als dieser Abschnitt.
Die im vorliegenden Modell diskutierten Teilchen sind aber nun nicht punktförmig, sondern ausgedehnt, aus konzeptionellen Gründen (endliche Feldenergie) und Gründen der numerischen Realisierbarkeit. Damit scheint kein Weg an der abelschen Approximation vorbeizuführen.
Startet man die durch (3.24) beschriebene Zeitentwicklung übrigens in einem "abelschen Szenario", also einem Anfangszustand, in dem A_μ^a, q_k^a

[3] Diese Bewegungsgleichung ist formal eigentlich ein "second-class constraint" nach Diracs Klassifikation, wie schon die eichinvariante Kontinuitätsgleichung (3.11). Sie folgt nicht als Euler-Lagrange-Gleichung aus der Lagrangedichte, sondern ist eine aus den Bewegungsgleichungen folgende Zwangsbedingung.

allesamt in einen abelschen Unterraum der $SU(3)_C$ fallen, so verschwindet der zweite Term in (3.24), und es folgt $q_k^a(t)$ =const, in Übereinstimmung mit unserer ursprünglichen Annahme. Die abelsche Approximation führt also zu einer trivialen Wong-Gleichung.

3.3.4 Funktionierender Ansatz in abelscher Approximation

Nach der ausführlichen Begründung, die im letzten Abschnitt gegeben wurde, wollen wir uns ab hier für den Rest der Arbeit auf eine abelsche Approximation des Modells beschränken. Im nächsten Abschnitt wird sich herausstellen, dass diese Approximation die Confinementeigenschaften des Modells nicht beschädigt.
Eine maximale abelsche Unteralgebra der $SU(3)$ wird zum Beispiel von den Erzeugern t^3 und t^8 aufgespannt. Alle Strukturkonstanten f^{38a} verschwinden. Damit ist ein einfacher Weg zu einer abelschen Approximation vorgezeichnet: wir beschränken uns in den Feldern $F^{\mu\nu a}$ auf die beiden Fälle $a \in \{3, 8\}$. Dann lautet die vollständige Lagrangedichte der klassischen Version des chromodielektrischen Modells

$$\mathcal{L} \;=\; \mathcal{L}_p + \mathcal{L}_c + \mathcal{L}_\sigma \quad . \tag{3.25}$$

Der Beitrag des σ-Feldes ist wie in (3.4)

$$\mathcal{L}_\sigma \;=\; \frac{1}{2}(\partial_\mu \sigma)(\partial^\mu \sigma) - U(\sigma) \quad . \tag{3.26}$$

Der Beitrag der Farbfelder liest sich formal wie in (3.3)

$$\mathcal{L}_c \;=\; -\frac{\kappa(\sigma)}{4} F^a_{\mu\nu} F^{\mu\nu a} \quad , \tag{3.27}$$

nur dass die Farbsummation nur noch über die Komponenten $a \in \{3, 8\}$ läuft, und der Feldtensor

$$F_{\mu\nu} \;=\; F^a_{\mu\nu} \frac{\lambda^a}{2} \;:=\; \frac{i}{g}[D_\mu, D_\nu]$$

die abelsche Form

$$F^a_{\mu\nu} \;=\; \partial_\mu A^a_\nu - \partial_\nu A^a_\mu \tag{3.28}$$

annimmt. Somit resultiert die abelsche Approximation darin, dass zwei entkoppelte Maxwellfelder die Farbdynamik beschreiben. Wir nennen diese einfach das 3- und das 8-Feld. Beide Felder kommunizieren nur indirekt, nämlich über das σ-Feld und die Farbladungen (siehe auch im Vorgriff auf den nächsten Abschnitt das Kopplungsschema Abb. 3.3).
Nun haben wir mit der erläuterten Näherung nicht nur die gluonische Selbstwechselwirkung vernachlässigt, sondern auch sechs Gluonfelder

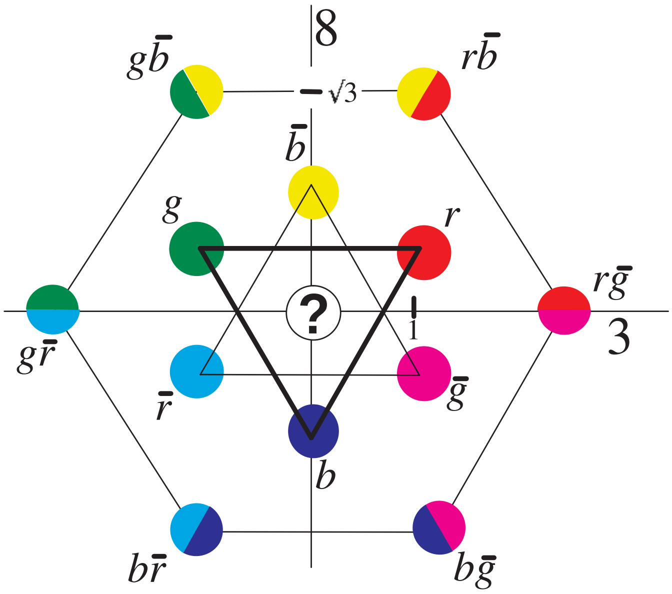


Abbildung 3.2: Quark- und Gluonenfarben in der abelschen Approximation. Das fettgedruckte gleichseitige Dreieck korrespondiert zu den Quarkfarben, das andere Dreieck zu den Antiquarkfarben. Das regelmäßige Sechseck stellt die Farben der sechs geladenen Gluonen dar. Die Farbwahl und die Bezeichnungen r, g, b, $\bar{r}$, $\bar{g}$, $\bar{b}$ sind beliebig, aber in der vorliegenden Arbeit einheitlich. Das Fragezeichen im Ursprung steht für die zwei fehlenden "abelschen" Gluonen des Oktetts, die wir als klassische, langreichweitige Felder behandeln.

$F^a_{\mu\nu}$ (mit $a \in \{1,2,4,5,6,7\}$). Diese sechs "geladenen" gluonischen Feldfreiheitsgrade können aber auf einfache Weise als Teilchen wieder ins Modell aufgenommen werden: Der Teilchenanteil der Lagrangedichte

$$\mathcal{L}_p = -\sum_k m_k \sqrt{1 - \dot{\vec{x}}_k^2}\, \rho_N(\vec{x} - \vec{x}_k(t)) - j^{\mu a}(x) A^a_\mu(x) \tag{3.29}$$

(Farbsummation wieder nur über $a \in \{3,8\}$) kann nicht nur farbgeladene Quarks beschreiben, sondern auch Gluonen. Dabei wird die Farbe eines Gluons aus einer Quark- und einer Antiquarkfarbe zusammengesetzt — dies ist durch den Wechselwirkungsterm der QCD-Lagrangedichte motiviert[4]. Gluonenspins werden dabei, wie schon die Quarkspins, vernachlässigt (bzw. ausgemittelt).

3.3.5 Wahl der Farbladungen

An dieser Stelle soll schließlich die konkrete Wahl der Ladungen q^a_k von Quarks und Antiquarks motiviert werden, aus denen sich dann die

[4]Ein Quark kann durch Absorption/Emission eines Gluons seine Farbe ändern.

Gluonladungen durch Zusammensetzung ergeben. Vergleichen wir den Ausdruck (3.8) für den quantenmechanischen Farbstrom der Quarks,

$$j^{\nu a} = g \sum_f \bar{q}^f \gamma^\nu \lambda^a q^f \quad , \tag{3.30}$$

mit dem klassischen Pendant (3.15)

$$j^{\mu a}(x) = g \sum_k q_k^a u_k^\mu \rho_N(\vec{x} - \vec{x}_k(t)) \quad , \tag{3.31}$$

und identifizieren die drei möglichen Quarkfarben mit den Farbvektoren

$$r = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} , \quad g = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \text{und} \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad ,$$

so ist die naheliegende Wahl für die drei Quarkfarben

$$\begin{aligned} q_{\mathrm{r}}^a &= r^\dagger \lambda^a r \\ q_{\mathrm{g}}^a &= g^\dagger \lambda^a g \\ q_{\mathrm{b}}^a &= b^\dagger \lambda^a b \end{aligned} \tag{3.32}$$

mit den Gell-Mann-Matrizen λ^a. Man beachte hierbei immer $a \in \{3, 8\}$[5]. Die resultierenden Farben der Quarks und Gluonen sind in Abb. 3.2 gezeigt und in Tab. 3.1 aufgeführt. Die Farben der sechs als Teilchen behandelten Gluonen folgen dabei im $\langle 3, 8\rangle$-Raum als vektorielle Summe je einer Quark- und einer Antiquarkladung, wie von der $SU(3)_C$-Struktur vorgegeben und am qqg-Vertex der pQCD abzulesen. Die Wahl der Farbnamen r, g, b, $\bar{r}$, $\bar{g}$, $\bar{b}$ ist natürlich beliebig, aber in der vorliegenden Arbeit einheitlich gehandhabt. Auf das Farbschema der Abb. 3.2 wird im weiteren Text noch oft Bezug genommen werden.

Im Ursprung in Abb. 3.2 sitzen eigentlich noch zwei Gluonen (symbolisiert als Fragezeichen), die zu λ^3 und λ^8 korrespondieren, diese sind aber genau die beiden "abelschen" Gluonen, die in der Simulation noch durch Felder repräsentiert werden. Würden wir diese auch als klassische Teilchen ins Modell aufnehmen, so wären sie in der abelschen Approximation farbneutral (deshalb sitzen sie ja im Ursprung) und würden daher auf die Modelldynamik überhaupt keinen Einfluss haben.

Nun zu den Bewegungsgleichungen: die klassischen, abelschen Bewegungsgleichungen des Modells gehen als Euler-Lagrange-Gleichungen aus der Lagrangedichte (3.25) hervor. Die Freiheitsgrade sind neben den beiden abelschen Feldern A_μ^a (immer $a \in \{3, 8\}$) und dem σ-Feld die Orte (Zentren) $\vec{x}_k(t)$ und kinetischen Impulse

$$\vec{p}_{k,\mathrm{kin}} = \frac{m\dot{\vec{x}}_k}{\sqrt{1 - \dot{\vec{x}}_k^{\,2}}} \tag{3.33}$$

[5] Die anderen Werte verschwinden sowieso, da in λ^a für $a \in \{1, 2, 4, 5, 6, 7\}$ nur nichtdiagonale Elemente besetzt sind.

Tabelle 3.1: *Quark- und Gluonenfarben in der abelschen Approximation.*

Teilchen	Farbname	Kürzel	q^3	q^8
q	rot	r	1	$1/\sqrt{3}$
q	grün	g	-1	$1/\sqrt{3}$
q	blau	b	0	$-2/\sqrt{3}$
$\bar{q}$	antirot(cyan)	$\bar{r}$	-1	$-1/\sqrt{3}$
$\bar{q}$	antigrün(magenta)	$\bar{g}$	1	$-1/\sqrt{3}$
$\bar{q}$	antiblau(gelb)	$\bar{b}$	0	$2/\sqrt{3}$
g	rot-antigrün	$r\bar{g}$	2	0
g	rot-antiblau	$r\bar{b}$	1	$\sqrt{3}$
g	grün-antiblau	$g\bar{b}$	-1	$\sqrt{3}$
g	grün-antirot	$g\bar{r}$	-2	0
g	blau-antirot	$b\bar{r}$	-1	$-\sqrt{3}$
g	blau-antigrün	$b\bar{g}$	1	$-\sqrt{3}$

der Partonen (Quarks und Gluonen). Die Bewegungsgleichungen lauten

$$
\begin{aligned}
\partial_\mu \left(\kappa(\sigma) F^{\mu\nu a}\right) &= j^{\nu a} \\
\partial_\mu \partial^\mu \sigma &= -U'(\sigma) - \frac{\kappa'(\sigma)}{4} F^a_{\mu\nu} F^{\mu\nu a} \\
\frac{d\vec{x}_k}{dt} &= \frac{\vec{p}_{k,\mathrm{kin}}}{\sqrt{\vec{p}^{\,2}_{k,\mathrm{kin}} + m^2}} \\
\frac{d\vec{p}_{k,\mathrm{kin}}}{dt} &= q^a_k \int d^3x \left(\vec{E}^a + \dot{\vec{x}}_k \times \vec{B}^a\right) \rho_N(\vec{x} - \vec{x}_k(t))
\end{aligned}
\tag{3.34}
$$

mit dem abelschen Feldtensor (3.28) und dem klassischen Farbstrom (3.15) der Quarks und sechs "farbigen" Gluonen (die wie die Quarks als starre ausgedehnte Teilchen behandelt werden). Dabei gilt für das farbelektrische und -magnetische Feld analog der Elektrodynamik

$$
\begin{aligned}
F^{0ia} &= E^a_i \\
F^{ija} &= \epsilon_{ijk} B^a_k \quad .
\end{aligned}
\tag{3.35}
$$

Daraus folgt für den Quellterm der σ-Feldgleichung

$$
-\frac{\kappa'(\sigma)}{4} F^a_{\mu\nu} F^{\mu\nu a} = +\frac{\kappa'(\sigma)}{2} \left(\vec{E}^{a2} - \vec{B}^{a2}\right) \quad . \tag{3.36}
$$

Die erste Gleichung von (3.34) ist eine Maxwellgleichung im Medium, die zweite eine Klein-Gordon-Gleichung mit Selbstwechselwirkung $U(\sigma)$, und in der letzten ist eine Kraft analog zur Lorentzkraft der Elektrodynamik enthalten. Die in (3.34) enthaltenen Kopplungen sind in Abb. 3.3 illustriert. Auffallend in dieser Abbildung ist, dass die beiden Kopplungsterme, die zu den Diagonalen des rautenförmigen Graphen

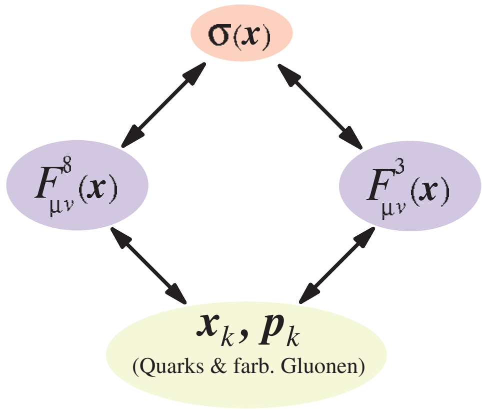


Abbildung 3.3: Kopplungsschema des chromodielektrischen Modells in der vorgestellten abelschen Version. Die zwei abelschen Vektorfelder wechselwirken nur indirekt miteinander, ebenso die Teilchen (Quarks und farbgeladene Gluonen) mit dem σ-Feld.

gehören würden, im vorliegenden Modell fehlen. Speziell der Quark-σ-Kopplungsterm, der zur senkrechten Diagonale des Graphen Abb. 3.3 korrespondiert, wird in Abschnitt 3.8 noch diskutiert werden.
Die vorliegende Arbeit beschäftigt sich mit der Untersuchung der vorgestellten Lagrangedichte und den entsprechenden Bewegungsgleichungen und ihrer Simulation in Hadronisierungsszenarien. Wenn wir im folgenden vom "chromodielektrischen Modell" oder auch einfach von "dem Modell" sprechen, so ist die in diesem Abschnitt vorgestellte Version gemeint, insbesondere auch die abelschen Näherung, mit der Vereinbarung, dass Farbindizes grundsätzlich nur die Werte $\{3, 8\}$ durchlaufen.

3.4 Farbconfinement

An der Lagrangedichte (3.25) lassen sich bereits viele Modelleigenschaften ablesen. Im folgenden sollen die Farbconfinementeigenschaften der Lagrangedichte gezeigt werden. Betrachten wir zunächst das Farbfeld A^a_μ, das nach (3.34) der Feldgleichung

$$\partial_\mu \left(\kappa(\sigma) F^{\mu\nu a}\right) = j^{\nu a} \tag{3.37}$$

genügt. In der üblichen Vektorschreibweise der Maxwellgleichungen liest sich (3.37) wie folgt:

$$\vec{\nabla} \cdot \vec{D}^a = \rho^a \tag{3.38}$$

$$\vec{\nabla} \cdot \vec{B}^a = 0 \tag{3.39}$$

$$\vec{\nabla} \times \vec{E}^a = -\frac{\partial \vec{B}^a}{\partial t} \tag{3.40}$$

$$\vec{\nabla} \times \vec{H}^a = \vec{j}^a + \frac{\partial \vec{D}}{\partial t} \quad . \tag{3.41}$$

mit der Definition (3.35) des farbelektrischen und -magnetischen Feldes, der bereits oft zitierten klassischen Stromdichte $j^{\mu a} = (\rho^a, \vec{j}^a)$ und

$$\begin{aligned} \vec{D}^a &= \kappa(\sigma)\vec{E}^a \\ \vec{H}^a &= \kappa(\sigma)\vec{B}^a \quad . \end{aligned} \tag{3.42}$$

Wir haben es also hierbei mit zwei "Farb-Maxwell-Feldern" zu tun ($a \in \{3, 8\}$), wobei $\kappa(\sigma)$ die Rolle der farbdielektrischen Zahl "ε" und gleichzeitig die der inversen farbmagnetischen Suszeptibilität "$1/\mu$" annimmt. Die "Farb-Lichtgeschwindigkeit $1/\sqrt{\varepsilon\mu}$" ist dadurch stets gleich 1, wie für ein kovariantes Modell zwingend notwendig.
Im Vergleich zu den bekannten elektrodynamischen Maxwellgleichungen ist an den obigen Gleichungen (3.38)–(3.41) ungewöhnlich, dass $0 \le \kappa(\sigma) \le 1$ gilt (siehe Abb. 3.1). In der Elektrodynamik gilt stets $\varepsilon \ge 1$. Das chromodielektrische Modell basiert ganz wesentlich auf dieser Eigenschaft der Dielektrizitätszahl κ. Um die Konsequenzen zu verstehen, betrachten wir die Energiedichte

$$\begin{aligned} \epsilon(x) &= \sum_k E_k^{\mathrm{kin}} \rho_N(\vec{x} - \vec{x}_k(t)) + \frac{\kappa(\sigma)}{2}\left(\vec{E}^{a2} + \vec{B}^{a2}\right) \\ &\quad + \frac{1}{2}\dot{\sigma}^2 + \frac{1}{2}(\vec{\nabla}\sigma)^2 + U(\sigma) \quad , \end{aligned} \tag{3.43}$$

die gemeinsam mit der Energieflussdichte

$$\vec{j}_\epsilon(x) = +\sum_k \vec{p}_k^{\mathrm{kin}} \rho_N(\vec{x} - \vec{x}_k(t)) - \kappa(\sigma)\vec{E}^a \times \vec{B}^a - \dot{\sigma}\vec{\nabla}\sigma \tag{3.44}$$

eine Kontinuitätsgleichung

$$\frac{\partial \epsilon}{\partial t} + \vec{\nabla} \cdot \vec{j}_\epsilon = 0 \tag{3.45}$$

erfüllt und daher im Integral eine Erhaltungsgröße darstellt. Wie man an (3.43) und Abb. 3.1 ablesen kann, nimmt das σ-Feld im Vakuum (Grundzustand) den Wert $\sigma(\vec{x}) \equiv \sigma_{\mathrm{vac}}$ an, also den Zustand minimaler Gesamtenergie $\int U(\sigma)\, d^3x$. Daraus folgt aber nach Abb. 3.1 zwingend $\kappa = 0$ — die funktionale Abhängigkeit $\kappa(\sigma)$ ist absichtlich so gewählt, dass sie diese Eigenschaft besitzt. Dies bewirkt, dass im physikalischen Vakuum sowohl $\vec{D}^a$ und $\vec{H}^a$ als auch der Farb-Poyntingvektor $\kappa\vec{E}^a \times \vec{B}^a$ verschwinden. Durch das physikalische Vakuum ist also kein Energietransport mit Farbwellen möglich.
Betrachten wir eine Ladungskonfiguration mit nur einem einzigen Quark (Abb. 3.4). Einzelne Quarks sind in der Natur bisher nicht beobachtet worden, vermutlich weil die Energie eines solchen Zustands sehr hoch

("unendlich") wäre. Das chromodielektrische Modell soll diese Eigenschaft der Natur widerspiegeln. Wir wollen dies verifizieren und betrachten nun die Feldenergie im Außenraum einer solchen Ladungskonfiguration. Es liegt nahe, die Feldkonfiguration als sphärisch symmetrisch anzunehmen[6], weil die Ladungsverteilung diese Symmetrie aufweist. Dann erzwingt das Gaußgesetz einen radialen farbelektrischen Fluss im Außenraum (Fernfeld) der Ladungskonfiguration von

$$D^a = \frac{Q^a}{4\pi r^2} \quad , \tag{3.46}$$

woraus für die Feldenergie des Fernfelds

$$\begin{aligned} E_f &= \int_{r_0}^{\infty} d^3r \left(\frac{\vec{D}^2}{2\kappa} + U(\sigma) \right) \\ &= \frac{Q^2}{4\pi} \int_{r_0}^{\infty} \frac{dr}{2r^2\kappa} + 4\pi \int_{r_0}^{\infty} dr\, r^2 U(\sigma) \end{aligned} \tag{3.47}$$

folgt. Die untere Integrationsgrenze r_0 ist hierbei beliebig gewählt, aber so, dass die Ladungskonfiguration vollständig in der Kugel um den Ursprung mit Radius r_0 eingeschlossen ist. Wäre die Einzelquarkkonfiguration im Modell zulässig, so müsste ihre Gesamtenergie, und damit insbesondere auch die Feldenergie des Fernfelds, endlich sein. Der erste Term in (3.47) kann nur konvergieren, wenn $\kappa(\sigma(r))$ für große r langsamer als $1/r$ gegen Null strebt. Dann wird jedoch $\sigma - \sigma_{\text{vac}}$ ebenfalls langsamer als $1/r$ gegen Null streben (siehe Abb. 3.1), und $U(\sigma)$, das in der Nähe des σ-Vakuumwerts quadratisch ($\propto (\sigma - \sigma_{\text{vac}})^2$) verläuft, wird folglich langsamer als $1/r^2$ gegen Null streben. Damit divergiert der zweite Term in (3.47), und die Einzelquarkkonfiguration ist im Modell nicht mit endlicher Energie zu realisieren (wie gewünscht).
Betrachten wir nun den farbelektrischen Flussschlauch zwischen einem Quark und einem Antiquark, die in der Summe farbneutral, "weiß", sind (Abb. 2.4). Das Gaußgesetz erzwingt hier kein farbelektrisches Fernfeld, also *kann* diese Konfiguration eine endliche Feldenergie besitzen. Die Lösung der statischen Feldgleichungen (3.34) erfüllt bekanntlich ein Variationsprinzip: sie ist die Feldkonfiguration minimaler Energie, wobei als Nebenbedingung das Gaußgesetz erfüllt sein muss:

$$\begin{aligned} \delta \int \left(\frac{\kappa(\sigma)}{2} \left(\vec{E}^{a2} + \vec{B}^{a2} \right) + \frac{1}{2}\dot{\sigma}^2 + \frac{1}{2}(\vec{\nabla}\sigma)^2 + U(\sigma) \right) &= 0 \\ \text{unter der Nebenbedingung } \vec{\nabla} \cdot \vec{D}^a &= \rho^a \end{aligned} \tag{3.48}$$

Vom Quark zum Antiquark muss also ein farbelektrischer Feldfluss $\vec{D}^a$ verlaufen. Im physikalischen Vakuum ($\sigma = \sigma_{\text{vac}}$, $\kappa = 0$) ist dies

[6]Dies muss nicht unbedingt der Fall sein! Wir vereinfachen nur unsere Überlegung damit.

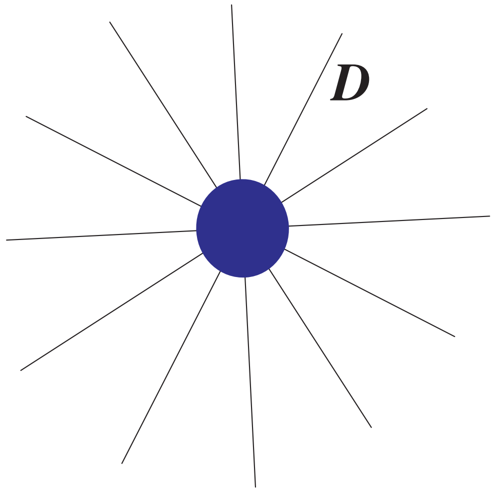


Abbildung 3.4: Einzelquarkkonfiguration.

unmöglich, also muss das σ-Feld vom Vakuumwert weggedrückt werden, was Arbeit gegen den Vakuumdruck $U(\sigma)$ kostet[7]. Es formiert sich ein Flussschlauch mit Radius R, an dessen "Wand" (praktischerweise etwa definiert durch $\sigma = 0.9\sigma_{\rm vac}$) ein Druckgleichgewicht herrscht: Von außen versucht der Vakuumdruck $U(\sigma)$, den Flussschlauch weiter zusammenzudrücken, und von innen bewirkt die Feldenergiedichte $\vec{D}^2/2\kappa \approx Q^2/2(4\pi R^2)^2\kappa$ einen Gegendruck bei kleinerwerdendem Schlauchradius R.

Im folgenden ist der erläuterte physikalische Inhalt der Lagrangedichte (3.25) noch einmal zusammengefasst:

- Durch das physikalische Vakuum ist kein Energietransport mit Farbwellen möglich, da dort der Poyntingvektor verschwindet.

- Einzelne Farbladungen sind Zustände unendlicher Energie, d. h. kommen im chromodielektrischen Modell nicht vor.

- Neutrale Systeme von Farbladungen bilden farbelektrische Flussschläuche aus.

Die letzten beiden Punkte werden wir im Resultatekapitel 5 mit Hilfe der Computersimulation verifizieren.

[7] Slang: *"Das Meson bohrt ein Loch ins physikalische Vakuum"*.

3.5 Irreduzible Weiße Cluster (IWCs) und Hadronenspektrum

Nachdem im letzten Abschnitt nachgewiesen wurde, dass die Lagrangedichte (3.25) tatsächlich Farbconfinement erzwingt, untersuchen wir in diesem Abschnitt die kleinstmöglichen farbneutralen (weißen) Objekte, die im Modell existieren. Diese Objekte, Cluster von Quarks und Gluonen, deren Farbladungen sich gegenseitig neutralisieren, können am ehesten mit "Hadronen" identifiziert werden; wir werden weiter unten diskutieren, inwiefern diese Identifikation gelingt.

Zunächst soll die Fragestellung präzisiert werden: wir suchen im vorliegenden Modell nach weißen Quark-Gluon-Clustern, die *irreduzibel* sind, d.h. nicht weiter zerlegbar in kleinere, ebenfalls weiße Cluster. Im Rahmen des vorgestellten Modells (das keine $q\bar{q}$-Produktion aus dem Gluonenfeld zulässt) sind solche irreduziblen weißen Cluster (IWC) stabil. Wir stellen zunächst die Frage nach der Farbstruktur eines IWC. Die möglichen Farben der Quarks und Gluonen wurden in Abb. 3.2 bereits grafisch dargestellt; nun suchen wir in diesem Schema nach IWCs. Dabei nehmen wir das Ergebnis vorweg, das im nächsten Abschnitt bewiesen wird: dass es nur endlich viele mögliche Farbkombinationen für IWCs gibt, und deren Farbstruktur sich nach Abb. 3.5 klassifizieren lässt.

Warum betrachten wir nur die *irreduziblen* weißen Cluster als die Hadronen des Modells? Ein reduzibles weißes Quark-Gluon-Cluster ist per definitionem in IWCs zerlegbar. Diese IWCs wechselwirken nur über Farbdipole und höhere Multipole; diese Wechselwirkung ist schwach und kurzreichweitig und damit *non-confining*. IWCs im Grundzustand (alle Konstituenten zueinander in Ruhe und am gleichen Punkt) wechselwirken sogar überhaupt nicht, da sie eine lokal verschwindende Farbladungsdichte besitzen und damit von allen Feldern vollständig entkoppeln. Ein reduzibles Cluster ist also nicht gebunden, sondern kann ohne Energieaufwand in eine Kombination von IWCs zerfallen.

Zu Abb. 3.5: In Klammern sind jeweils die Entartungsgrade der Kombinationen angegeben, also die Anzahl der IWCs, die durch Spiegelung oder Drehung aus der abgebildeten Kombination enstehen können. Falls die Quarks unterscheidbar sind, z. B. weil mehrere verschiedene Quarkflavors in einem IWC versammelt sind, erhöht sich der Entartungsgrad teilweise erheblich.

Auffallend ist, dass neben den Gluebällen nur baryonische (3 Quarks oder 3 Antiquarks) und mesonische (Quark+Antiquark) IWCs existieren, möglicherweise noch mit gluonischen Beimischungen — es existieren keine 4-Quark oder 6-Quark-IWCs. Dieses Resultat des einfachen chromodielektrischen Modells ist in überraschender Übereinstimmung mit der experimentellen Erfahrung der Hadronenphysik. Natürlich kann man weiße Cluster mit 4, 6 oder mehr Quarks kon-

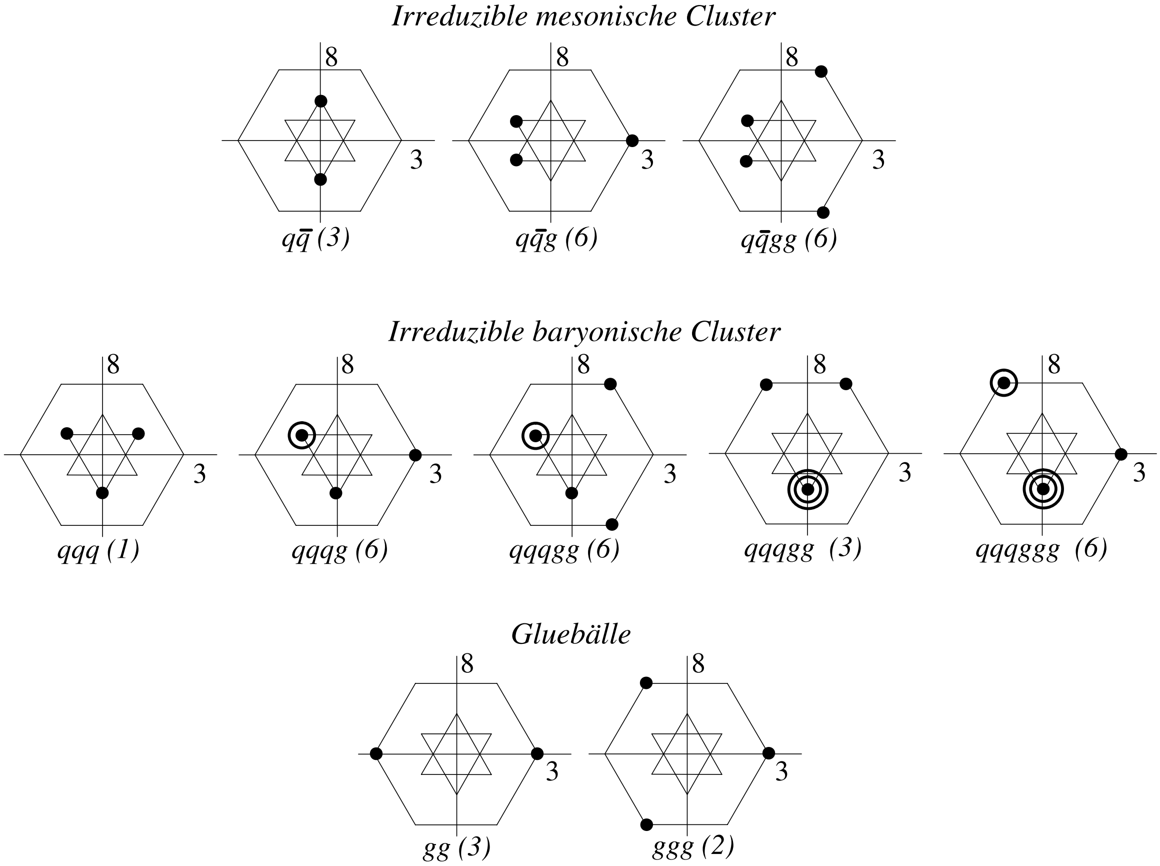


Abbildung 3.5: Klassifikation aller irreduziblen weißen Cluster (IWCs) des Modells. Die Zahlen in Klammern geben den Entartungsgrad an, wobei bei den Baryonen jeweils noch die gleiche Anzahl Antibaryonen dazuzuzählen ist. Falls mehrere Quarkflavors vorhanden sind, erhöht sich die Zahl der möglichen Kombinationen erheblich. Mehrfachbesetzungen sind durch Ringe dargestellt.

struieren, diese sind aber dann nicht mehr irreduzibel und damit keine "Hadronen" unseres Modells.
In Abb. 3.6 ist das resultierende Hadronenspektrum dargestellt, wobei wir neben den leichten up/down Quarks (q) auch strange (s) und charmed (c) Quarks betrachten. Die Wahl der Quarkmassen wird erst später (in Abschnitt 5.1) diskutiert und soll hier einfach hingenommen werden:

$$\begin{aligned} m_q &= 400MeV \\ m_s &= 550MeV \\ m_c &= 1500MeV \\ m_g &= 700MeV \end{aligned} \tag{3.49}$$

Jeder Strich in Abb. 3.6 stellt die minimale Masse eines entsprechenden Clusters dar, also diejenige Masse, die das Cluster im feldfreien Fall hat,

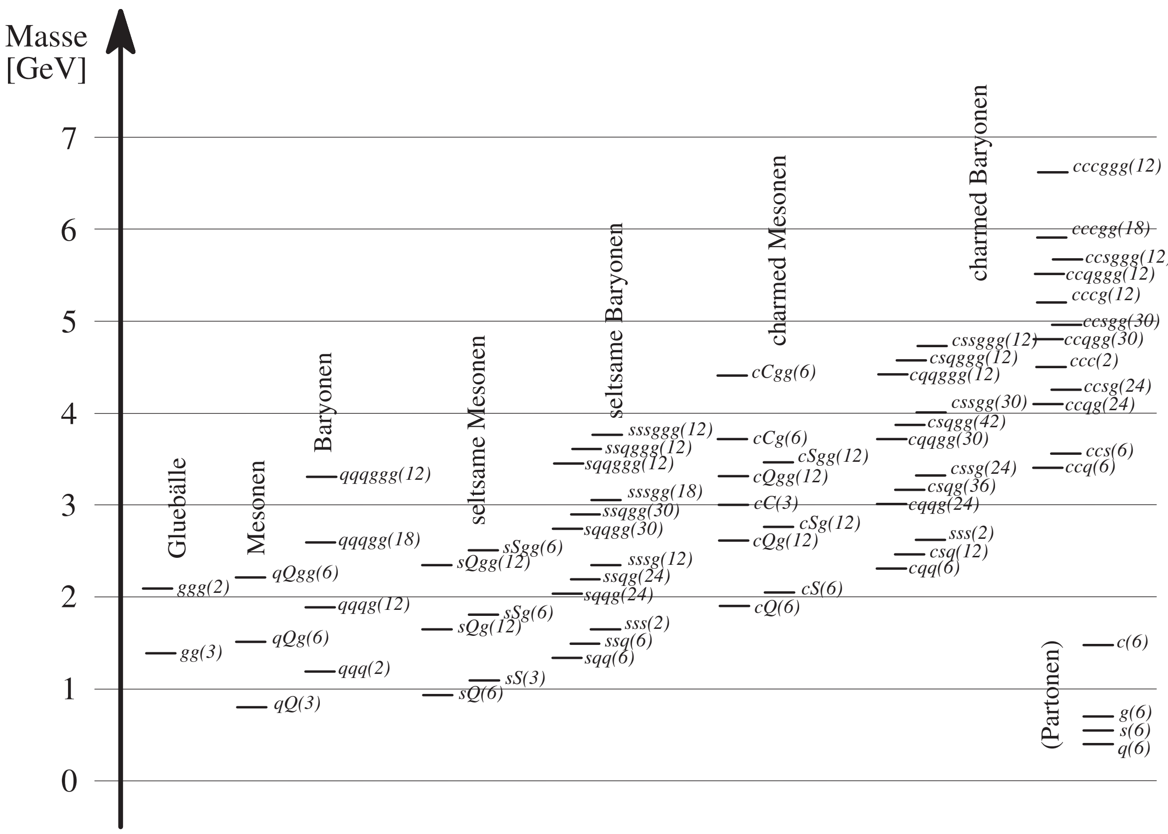


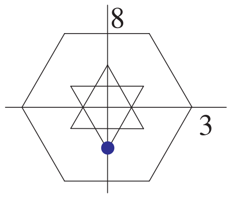
Abbildung 3.6: IWC-Massenspektrum. Jeder Strich markiert die Summe der Konstituentenmassen der jeweiligen Kombination, was der minimalen (feldfreien) Masse des Clusters entspricht. Daher steht jeder Strich für eine Schwelle (Threshold) im Spektrum. Die Bezeichnungen beziehen sich auf die IWCs nach Abb. 3.5. Dabei stehen q, s, c für Quarks, Großbuchstaben für die entsprechenden Antiquarks und g für ein Gluon. Die Zahlen in Klammern geben den Entartungsgrad an, wobei die jeweiligen Antiteilchen mitgezählt wurden.

wenn alle Ladungen an der gleichen Stelle zentriert sitzen und somit alle Felder überall ihren Vakuumwert annehmen. In diesem Fall ist die Masse des IWCs einfach gleich der Summe der Konstituentenmassen. Die Striche in Abb. 3.6 sind einfach die Summen der Konstituentenmassen für IWCs nach Abb. 3.5 mit beliebigen Flavorkombinationen.

Falls Felder vorhanden sind, also z. B. im Fall eines Jojo-Schwingungsmodus oder einer Rotationsbewegung nach Abb. 2.2, ist die Masse des Hadrons entsprechend höher. Also sind die Striche in Abb. 3.6 nicht wirklich diskrete Energieniveaus, sondern stattdessen Schwellenwerte (*Thresholds*) für den jeweiligen hadronischen Kanal.

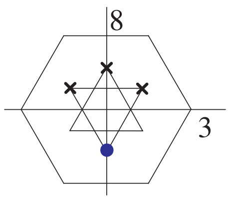
3.6 Vollständigkeitsbeweis der IWC-Klassifikation

In diesem Abschnitt wird die Vollständigkeit der Klassifikation nach Abb. 3.5 nachgewiesen, d. h. es wird gezeigt, dass es kein IWC geben kann, das nicht bereits in Abb. 3.5 erfasst ist.
Beschränken wir uns vorerst auf reine Quark-Antiquark-Cluster, und zeigen wir, dass ein solches Cluster stets ein IWC vom Typ $q\bar{q}$ oder ein Baryon vom Typ qqq (nach Abb. 3.5) enthält.
Ein beliebiges Quark-Antiquark-Cluster enthält mindestens ein Quark oder ein Antiquark, o. B. d. A. sei ein blaues Quark[8] im Cluster enthalten:

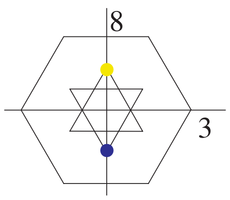


.

Um das Cluster farbneutral (weiß) zu machen, muss noch ein Teilchen oberhalb der 3-Achse hinzukommen. Wir beschränken uns momentan auf Quark-Antiquark-Cluster, also kommen nur die drei angekreuzten Positionen in Frage:

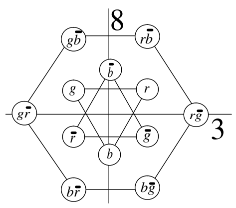


Die Wahl des oberen, antiblauen Antiquarks führt zur Mesonkonfiguration $q\bar{q}$



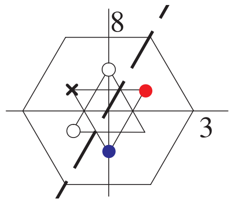
,

[8]Unser Farbschema aus Abb. 3.2 wird, wie versprochen, in dieser Arbeit konsequent verwendet. Zur Erinnerung der Farbnamen:



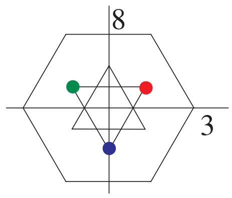
.

die in unserer Klassifikation auch vorkam. Die Wahl einer der anderen Positionen, o. B. d. A. des roten Quarks, führt zu



.

Betrachten wir die gestrichelte Linie. Um das Cluster weiß zu machen, fehlt noch mindestens ein Teilchen links von dieser Linie. Die beiden mit offenen Kreisen bezeichneten Positionen scheiden aus, da sie zu Mesonen als Subclustern führen würden; das resultierende Cluster wäre dann nicht mehr irreduzibel. Also muss das grüne Quark dazu, was zur Baryonkonfiguration qqq



.

führt, die auch in unserer Klassifikation auftaucht. Für reine Quark-Antiquark-Cluster ist damit der Beweis der Vollständigkeit abgeschlossen: die einzigen irreduziblen Cluster sind die beiden vom Typ $q\bar{q}$ bzw. qqq, die in Abb. 3.5 aufgeführt sind. Alle größeren weißen Cluster enthalten, wie bewiesen, stets mindestens einen der beiden IWC-Typen als Subcluster und sind damit reduzibel.
Die Beweisführung lässt sich direkt auf die reinen Gluebälle übertragen, die ja im Schema ebenfalls als regelmäßiges Sechseck angeordnet erscheinen. Es gibt also nur zwei verschiedene Gluebälle, einer mit zwei, der andere mit drei Gluonen, wie in Abb. 3.5 dargestellt.
Die noch verbleibende Aufgabe ist, den Beweis der Vollständigkeit für gemischte Quark-Antiquark-Gluon-Cluster zu führen. Ein Beweis im obigen Stil führt zu einem unästhetisch großen Baum von Fallunterscheidungen, den wir im Anhang A vollständig abarbeiten.

3.7 Analoge Betrachtung in der QCD

Um das wesentliche Resultat unseres Klassifikationstheorems noch einmal zu wiederholen: Neben den Gluebällen existieren nur baryonische (3 Quarks oder 3 Antiquarks) und mesonische (Quark+Antiquark) irreduzible weiße Cluster, wobei noch gluonische Beimischungen vorhanden sein können. Es existieren aber z. B. keine 6-Quark-IWCs, wie auch im Experiment niemals 6-Quark-Hadronen beobachtet wurden.

In der QCD kann man eine dazu analoge Beobachtung machen. Dazu eine Vorbemerkung: In einer Eichfeldtheorie, selbst einer klassischen, ist eine Quarkfarbe ("rot") nur lokal, also an einem Raumzeitpunkt, klar definiert. Um die Farbe eines Quarks mit der eines Antiquarks zu vergleichen, bspw. um die Singletteigenschaft eines Mesons zu verifizieren, muss die Quarkfarbe zunächst zum Antiquark "parallel transportiert" werden. Dabei treten Verbindungskoeffizienten in Form von Eichfeldern auf. Die damit verbundene Problematik der Singlettkonstruktion wollen wir hier ignorieren und nehmen daher an, wie auch in den obigen Modellbetrachtungen, dass alle Quarks, Antiquarks und Gluonen am gleichen Raumpunkt sitzen. Es geht uns also nur um die Farbstruktur. Dazu bedienen wir uns einer grafischen Notation ähnlich den Feynmandiagrammen, die im folgenden definiert wird.
Um ein Hadron "an einem Raumzeitpunkt" zu konstruieren, haben wir folgende Mittel zur Hand: Wir können ein Quark mit Farbe i ($i = 1 \ldots 3$) erzeugen

$$\bar{q}_i \left| 0 \right\rangle = i \!\rightarrow\!\!\!- \quad ,$$

ein Antiquark mit Farbe n ($n = 1 \ldots 3$)

$$q_n \left| 0 \right\rangle = n \!-\!\!\!\leftarrow \quad ,$$

oder ein Gluon mit Farbe a ($a = 1 \ldots 8$)

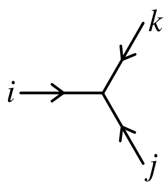
$$g^a \left| 0 \right\rangle = a \!=\!= \quad .$$

Diese Generatoren können nun zu Singletts wie folgt kombiniert werden:

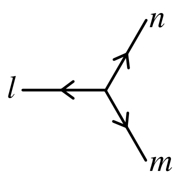
(i) durch Verwendung von Kroneckersymbolen

$$\begin{aligned} q_i \bar{q}_i \left| 0 \right\rangle &= q_i \delta_{il} \bar{q}_l \left| 0 \right\rangle = i \!\rightarrow\!\!\!-\, l \\ g^a g^a \left| 0 \right\rangle &= g^a \delta^{ab} g^b \left| 0 \right\rangle = a \!=\!=\, b \quad , \end{aligned}$$

(ii) durch Verwendung des vollständig antisymmetrischen Levi-Civita-Tensors

$$\epsilon_{ijk} \bar{q}_i \bar{q}_j \bar{q}_k \left| 0 \right\rangle =$$ 

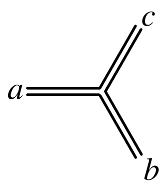
$$\epsilon_{lmn} q_l q_m q_n \left| 0 \right\rangle \;=$$



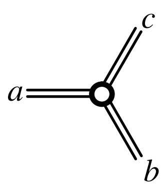
,

(iii) durch Verwendung der $SU(3)$-Strukturkonstanten f^{abc} oder des d^{abc}-Symbols

$$f^{abc} g^a g^b g^c \left| 0 \right\rangle \;=$$



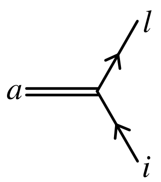
$$d^{abc} g^a g^b g^c \left| 0 \right\rangle \;=$$



,

(iv) durch Verwendung von Gell-Mann-Matrizen λ^a_{il}

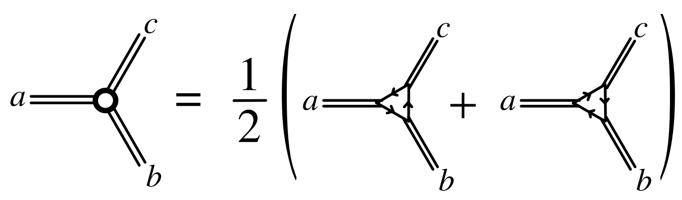
$$\lambda^a_{il} g^a q_i \bar{q}_l \left| 0 \right\rangle \;=$$



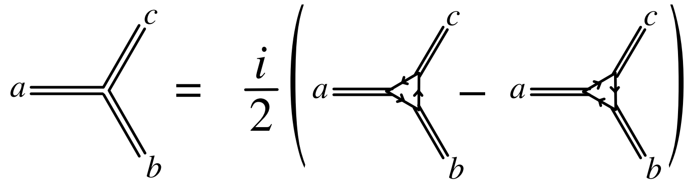
.

Zwischen den hier auftretenden Tensoren bestehen gewisse Beziehungen, die wir grafisch wie folgt notieren:

- Darstellung der Strukturkonstanten f^{abc} und d^{abc} durch die Gell-Mann-Matrizen λ^a_{ij}:

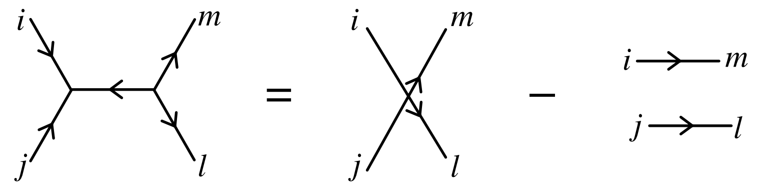


(3.50)



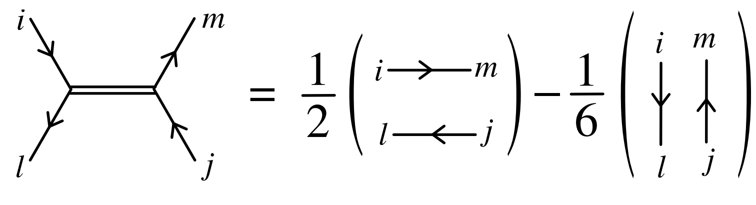
(3.51)

- Beziehung zwischen Levi-Civita-Tensor und Kroneckersymbol:



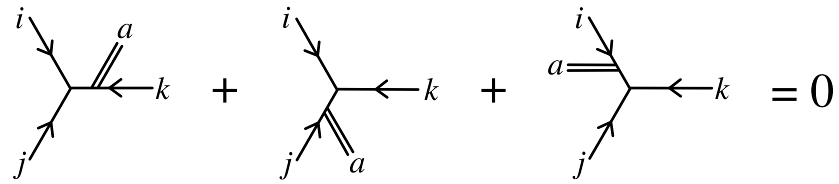
(3.52)

- Fierz-Relation zwischen Gell-Mann-Matrizen und Kroneckersymbol:



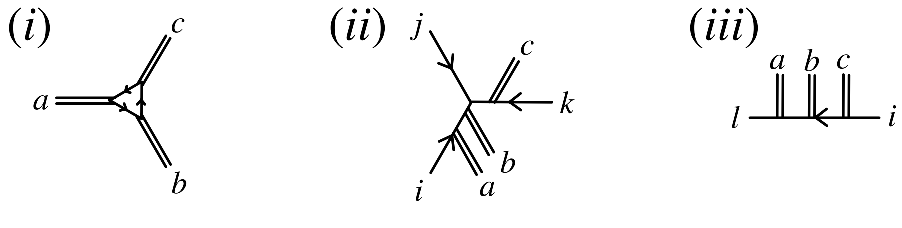
(3.53)

- Beziehung zwischen Levi-Civita-Tensor und Gell-Mann-Matrizen:



(3.54)

Mit Hilfe dieser Relationen kann man jeden komplizierteren Farbgraphen in eine Linearkombination von disjunkten Graphen zerlegen, wobei die resultierenden Graphen jeweils einem von drei möglichen Typen zuzuordnen sind: (i) reine Gluebälle, (ii) mesonische Cluster, (iii) baryonische Cluster, wobei noch Gluonen beigemischt sein können. Beispiele für die drei Fälle sind im folgenden gezeichnet:



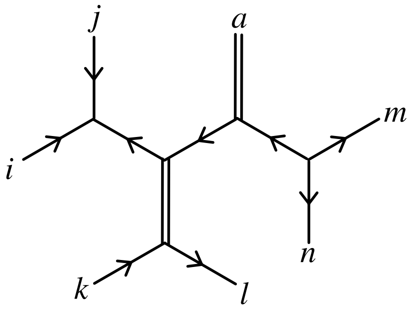
Die zu diesen Bildern korrespondierenden Zustände im Farbraum werden wie folgt konstruiert:

(i) $\lambda^a_{ij}\lambda^b_{jk}\lambda^c_{ki} g^a g^b g^c \,\big|\, 0\big\rangle$

(ii) $\lambda^a_{is}\lambda^b_{st}\lambda^c_{kn}\epsilon_{jtn} q^i q^j q^k g^a g^b g^c \,\big|\, 0\big\rangle$

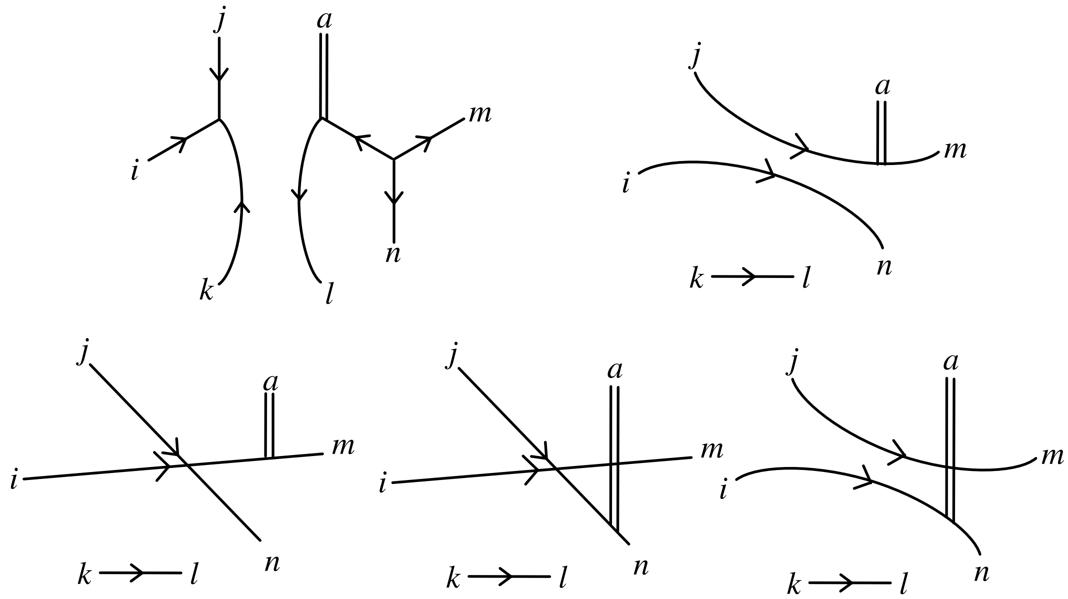
(iii) $\lambda^a_{lm}\lambda^b_{mn}\lambda^c_{ni} g^a g^b g^c q^l q^i \,\big|\, 0\big\rangle$

Dies sind also die "irreduziblen weißen Cluster" der $SU(3)$, zumindest in unserer vereinfachten Betrachtung ohne Eichfelder. Die Farbstruktur eines beliebigen weißen Zustands lässt sich stets als quantenmechanische Überlagerung disjunkter Graphen dieser Art verstehen. Die allgemeine Zerlegbarkeit von solchen Farbgraphen in irreduzible Diagramme der Formen $(i) - (iii)$ lässt sich durch Induktion über die Anzahl der Gluonlinien beweisen. Dieser Beweis soll hier nicht explizit ausgeführt werden.
Statt eines Beweises nehmen wir ein Beispiel: Der weiße Zustand



(3.55)

aus 3 Quarks, 3 Antiquarks und einem Gluon lässt sich mit Hilfe der Regeln (3.52)–(3.54) in eine Linearkombination der Terme



(3.56)

zerlegen, die alle nur mesonische und baryonische Graphen von den Typen (ii) und (iii) enthalten. Man kann dies einfach anhand der "Spielregeln" (3.52)–(3.54) nachvollziehen.
Was ist der physikalische Inhalt dieser Aussage in der QCD? Quantenmechanisch kann es natürlich sein, dass eine Linearkombination (kohärente Überlagerung) der Farbkopplungen nach (3.56) energetisch

günstiger ist als jeder einzelne der Terme in (3.56), d. h. ein gebundener Zustand von 3 Quarks, 3 Antiquarks und einem Gluon kann trotzdem existieren. In einem $SU(3)$-chromodielektrischen Modell (mit punktförmigen Farbladungen, wie in Abschnitt 3.3.3 diskutiert) macht die kohärente Überlagerung von Zuständen konzeptionell aber keinen Sinn, also können wir schließen, dass selbst im vollen $SU(3)$-chromodielektrischen Modell nur irreduzible Singlettcluster in den Konstruktionsmustern (i) – (iii) möglich sind: es gibt keine 6-Quark-Bags oder Vergleichbares. Wie in den vorigen Abschnitten bereits mehrfach festgestellt, scheinen wir durch die abelsche Näherung nicht viel, wenn überhaupt etwas, an physikalischem Inhalt zu verlieren.

3.8 Diskussion eines möglichen Quark-σ-Kopplungsterms

Am Ende dieses Kapitels soll noch ein möglicher Quark-σ-Wechselwirkungsterm diskutiert werden, vor allem deshalb, weil er in der Literatur des öfteren benutzt wurde, u. a. von den Erfindern des Modells, Friedberg und Lee [25, 26, 27]. Gemeint ist die Modifikation des Quarkterms zum Ausdruck

$$\mathcal{L}_p^{qm} = \sum_f \bar{q}^f \left(i\boldsymbol{D} - m_f - g_S\sigma\right) q^f \quad , \tag{3.57}$$

der einen direkten Kopplungsterm zwischen Quarkdichte und σ-Feld enthält. Diese Kopplung korrespondiert zu einem senkrechten Pfeil im Schema nach Abb. 3.3.
Die Konsequenzen dieser Modifikation auf die Modellphänomenologie sind vielfältig:

- Die Quark-σ-Wechselwirkung bewirkt einen Phasenübergang bei hoher Quarkdichte, selbst wenn die Quarkdichte räumlich konstant ist und daher überall die lokale Farbladung verschwindet [54, 55].

- Hadronen lassen sich als solitonische gebundene Zustände von Quarks auffassen, wobei die Farbladungsdichte überall lokal verschwindet [25, 32, 33, 37].

- Weiße Cluster bewirken über den neuen Kopplungsterm einen Abdruck ("Bag") im σ-Feld, selbst wenn alle Clusterpartonen am gleichen Punkt sitzen, also die lokale Farbladungsdichte verschwindet. Das hat zur Folge, dass solche "kalten" weißen Cluster noch miteinander wechselwirken. Die resultierende Wechselwirkung ist kurzreichweitig und attraktiv ohne einen repulsiven Kern (da die Verschmelzung zweier "Bags" einen Vorteil in der Volumenenergie bringt). Dies ist zur Beschreibung der NN-Wechselwirkung nicht ausreichend.

In der vorliegenden Arbeit verwenden wir keinen Kopplungsterm der Art (3.57). Die Gründe dafür sind:

- Zur Beschreibung des Farbconfinements wird der genannte Wechselwirkungsterm nicht benötigt.
- Die neue Wechselwirkung kommt mit einer neuen Kopplungskonstanten, die als zusätzlicher Fitparameter das Modell verkompliziert, ohne relevante neue Physik im Modell beschreiben zu können.
- Die effektive Quarkmasse $m_f + g_S\sigma$ in (3.57) kann lokal negativ werden, was ein echtes konzeptionelles Problem darstellt.
- Irreduzible weiße Cluster (IWC), die "Hadronen" unseres Modells, sollten wie in der Natur nur schwach miteinander wechselwirken (mit Potentialtiefen von maximal $50 - 70 MeV$ MeV). Die Beschreibung dieser im Vergleich zur Stringspannung ($1 GeV/fm$) schwachen Wechselwirkung mit der Quark-σ-Kopplung ist nicht möglich und auch nicht Ziel dieser Untersuchung.
- Thomas Vetter schreibt in seiner Dissertation: "Für ... Einschussenergien bis etwa $200 MeV$ kommt es zu einer Fusion der Nukleonen" ([33], S. 103). Die kurzreichweitige attraktive Wechselwirkung weißer Cluster über das σ-Feld würde also ausreichen, um bei Temperaturen $\leq 150 MeV$ auch reduzible Cluster zu binden. Damit wäre die Beschreibung der Hadronisierung, wie wir sie im Resultatkapitel 5 vorstellen werden, unmöglich. Dies gilt zumindest für eine skalare Kopplungskonstante $g_S \approx 8$, wie Vetter sie für seine Solitonenstudien verwendet hat[9].

Dies sollte genügen, die Nichteinbeziehung eines Kopplungsterms (3.57) in unsere Lagrangedichte zu begründen. Wir werden im Abschnitt 5.3 noch kurz die (geringfügigen) Auswirkungen dieser Kopplung auf den Feldverlauf im farbelektrischen Flussschlauch studieren, die direkte Quark-σ-Wechselwirkung aber ansonsten nicht weiter diskutieren.

[9]Eine kleinere Kopplungskonstante würde das genannte Problem zwar nicht verursachen, es ist aber nach Meinung des Autors fragwürdig, was das Modell an physikalischem Inhalt gewinnen würde (für den Preis eines zusätzlichen Parameters).

*“Möge dieses Instrument Ihnen helfen,
die Probleme dieser Welt zu beseitigen,
die wir Alten euch hinterlassen haben.”*
K. Zuse

*“Alles ist entweder Teil des Problems,
Teil der Lösung oder Teil der Landschaft.”*
R. de Niro als “Sam” im Film “Ronin”

4 Computersimulation des Modells

In diesem Kapitel sollen nach einer kurzen Wiederholung der Modellgleichungen aus Kapitel 3 die Approximationen erläutert werden, die zum Zweck der numerischen Simulation vorgenommen wurden. Danach werden die numerischen Verfahren vorgestellt und diskutiert, die für die Computersimulation CDM[1] im Rahmen dieser Arbeit eingesetzt wurden. Hier wird der Versuch unternommen, eine verständliche Einführung in die Finite-Elemente-Methode zu geben, die mittels adaptiver Simplexverfeinerung und Mehrgitter-Lösungsverfahren in der Lage ist, die Vektorfeldgleichungen schnell und genau zu lösen. Besondere Betonung findet dabei ein eleganter und sehr schneller Bisektionsalgorithmus zur adaptiven Verfeinerung von Simplexnetzen, der als Nebenprodukt dieser Arbeit entstanden ist [58].

Es werden an Schlüsselstellen konkrete Verweise auf Programmfiles, C++-Klassen und Klassenmethoden gegeben. Diese Hinweise sind für den Benutzer und Weiterentwickler des Simulationsprogramms gedacht, der sich in den etwa 37.000 C++-Programmzeilen, die in 107 Files und 125 Klassen organisiert sind, zurechtfinden muss. Ein Wegweiser durch das Programm findet sich im Anhang dieser Arbeit.

[1]Der Programmname ist einfach eine Abkürzung der Bezeichung “chromodielektrisches Modell”. Wichtige Teile von CDM mit eigenen Namen sind das Displaymodul DISP und der Multigrid-Differentialgleichungslöser MUESLEA (zu dieser Namensgebung siehe die Fußnote auf Seite 80).

4.1 Bewegungsgleichungen und Approximationen

4.1.1 Bewegungsgleichungen und Approximationen des Modells

Unsere Aufgabe ist die Simulation der aus der Lagrangedichte (3.25)

$$
\begin{aligned}
\mathcal{L} &= \mathcal{L}_p + \mathcal{L}_c + \mathcal{L}_\sigma \\
\mathcal{L}_p &= -\sum_k m_k \sqrt{1 - \dot{\vec{x}}_k^2}\, \rho_N(\vec{x} - \vec{x}_k(t)) - j^{\mu a}(x) A_\mu^a(x) \\
\mathcal{L}_c &= -\frac{\kappa(\sigma)}{4} F_{\mu\nu}^a F^{\mu\nu a} \\
\mathcal{L}_\sigma &= \frac{1}{2}(\partial_\mu \sigma)(\partial^\mu \sigma) - U(\sigma)
\end{aligned}
\tag{4.1}
$$

mit

$$
j^{\mu a}(x) = g \sum_k q_k^a u_k^\mu \rho_N(\vec{x} - \vec{x}_k(t))
$$

hervorgehenden Bewegungsgleichungen (3.34)

$$
\begin{aligned}
\frac{d\vec{x}_k}{dt} &= \frac{\vec{p}_k}{\sqrt{\vec{p}_k^2 + m^2}} \\
\frac{d\vec{p}_k}{dt} &= q_k^a \int d^3x \left(\vec{E}^a + \dot{\vec{x}}_k \times \vec{B}^a \right) \rho_N(\vec{x} - \vec{x}_k(t)) \\
\partial_\mu \left(\kappa(\sigma) F^{\mu\nu a} \right) &= j^{\nu a} \\
\partial_\mu \partial^\mu \sigma &= -U'(\sigma) - \frac{\kappa'(\sigma)}{4} F_{\mu\nu}^a F^{\mu\nu a} \quad .
\end{aligned}
\tag{4.2}
$$

Diese Gleichungen beschreiben eine klassische, molekulardynamische Approximation des chromodielektrischen Modells. Die Teilchen, Quarks und Gluonen, haben dabei Farbladungs- sowie Massenverteilungen proportional zur normierten Anzahldichte ρ_N aus (3.13),

$$
\rho_N(\vec{x} - \vec{x}_k(t)) = \left(\frac{1}{2\pi r_0^2} \right)^{3/2} e^{-(\vec{x} - \vec{x}_k(t))^2 / 2r_0^2} \quad . \tag{4.3}
$$

Die Wahl der starren Verteilungen ρ_N stellt eine nichtrelativistische Näherung dar (siehe Diskussion in Abschnitt 3.3.2), ebenso die Vernachlässigung jeglicher Spinstruktur der Teilchen.

4.1.2 Approximationen der numerischen Simulation

Eine bisher nicht erwähnte Approximation wird in der vorliegenden Computersimulation CDM gemacht, um den Rechenaufwand zu verringern: wir vernachlässigen das farbmagnetische Feld $\vec{B}^a$ in den Bewegungsgleichungen. Dadurch wird die Anzahl der zu berechnenden Feldfreiheitsgrade verringert, was die Problemstellung vereinfacht. Untersuchen wir die formalen Konsequenzen dieser Näherung: Setzen wir $\vec{B}^a = 0$, so wird das Induktionsgesetz zu

$$\vec{\nabla} \times \vec{E}^a \;=\; 0 \quad , \tag{4.4}$$

also ist das farbelektrische Feld wirbelfrei im gesamten Raum. Damit ist es als Gradient eines *farbelektrostatischen Potentials* darstellbar,

$$\vec{E}^a \;=\; -\vec{\nabla}\phi^a \quad . \tag{4.5}$$

(auch hier gilt wieder die Vereinbarung $a \in \{3, 8\}$ der abelschen Näherung, siehe Abschnitt 3.3.4). Durch Einsetzen in das Gaußgesetz (3.38) erhalten wir die Feldgleichung für dieses Potential

$$\vec{\nabla} \cdot \left(\kappa(\sigma)\vec{\nabla}\phi^a\right) \;=\; -\rho^a \quad , \tag{4.6}$$

die keine Zeitableitung enthält und daher ihrer Natur nach eine Zwangsbedingung ist. Diese Gleichung muss mit geeigneten räumlichen Randbedingungen in jedem Zeitschritt erneut gelöst werden[2]. In den Bewegungsgleichungen der Teilchen fällt durch die Vernachlässigung des Farbmagnetfelds die Lorentzkraft weg, so dass sie nun

$$\begin{aligned} \frac{d\vec{p}_k}{dt} \;&=\; -q_k^a \int d^3x \, \left(\vec{\nabla}\phi^a(\vec{x})\right) \rho_N(\vec{x} - \vec{x}_k(t)) \\ \frac{d\vec{x}_k}{dt} \;&=\; \frac{\vec{p}_k}{\sqrt{\vec{p}_k^{\,2} + m^2}} \end{aligned} \tag{4.7}$$

lauten. Der Quellterm der σ-Feldgleichung wird mittels

$$F^a_{\mu\nu}F^{\mu\nu a} \;=\; -2(\vec{\nabla}\phi^a)(\vec{\nabla}\phi^a) \tag{4.8}$$

auf den Feldfreiheitsgrad ϕ^a umgeschrieben; es resultiert die Feldgleichung

$$\partial_t^2\sigma \;=\; \nabla^2\sigma - U'(\sigma) + \frac{\kappa'(\sigma)}{2}(\vec{\nabla}\phi^a)(\vec{\nabla}\phi^a) \quad . \tag{4.9}$$

Nun zu den physikalischen Konsequenzen der Vernachlässigung des Farbmagnetfelds:

[2]Es sei denn, man fände ein Zeitschrittverfahren für die zeitabhängigen Größen, das die Zwangsbedingungen nicht verletzt. Die Konstruktion eines solchen Verfahrens ist aber nichttrivial, und es ist keine allgemeingültige Vorgehensweise bekannt.

- Die Näherung ist exakt für statische, stromlose Farbladungskonfigurationen. Die Näherung ist ebenfalls exakt für idealisierte jojoartige Anregungsmoden von Flussschläuchen in deren Ruhesystem, da sich in diesem Fall der Maxwellsche Verschiebungsstrom $\partial_t \vec{D}^a$ gerade mit der Stromdichte j^a aufhebt [59].

- Die Energie- und Energieflussdichte aus (3.43) und (3.44) verändern sich — insbesondere fällt der Poyntingterm weg:

$$
\begin{aligned}
\epsilon(x) &= \sum_k E_k^{\text{kin}} \rho_N(\vec{x} - \vec{x}_k(t)) + \frac{\kappa(\sigma)}{2}(\vec{\nabla}\phi^a)^2 \\
&\quad + \frac{1}{2}\dot{\sigma}^2 + \frac{1}{2}(\vec{\nabla}\sigma)^2 + U(\sigma) \\
\vec{j}_\epsilon(x) &= \sum_k \vec{p}_k \rho_N(\vec{x} - \vec{x}_k(t)) - \dot{\sigma}\vec{\nabla}\sigma \quad .
\end{aligned}
\tag{4.10}
$$

  Damit ist das Farbfeld nicht mehr in der Lage, durch Wellen Energie zu transportieren. Allerdings sind Wellen auch im vollen chromodielektrischen Modell nur in der Lage, *innerhalb* eines hadronischen Bags Energie zu transportieren — der Poyntingvektor $\kappa\vec{E}^a \times \vec{B}^a$ verschwindet im Außenraum (nonperturbatives Vakuum) sowieso.

- Der Mechanismus des Farbconfinements, wie er in Abschnitt 3.4 erläutert wurde, beruht ausschließlich auf dem Zusammenspiel von Gaußgesetz einerseits (erzwingt Feldlinien zwischen Farbladungen) und dielektrischer Kopplung ans σ-Feld und U-Potential andererseits (erzeugt Vakuumdruck, der Feldlinien zum String bündelt). Das $\vec{B}^a$-Feld ist nicht essentiell für diesen Mechanismus.

- Die Lorentzinvarianz des Wechselwirkungsterms zwischen Farbladungen und -feldern wurde bereits durch die Annahme starrer ausgedehnter Verteilungen $\rho_N(\vec{x} - \vec{x}_k)$ zerstört. Mit der Vernachlässigung des $\vec{B}^a$-Feldes ist auch das Farbfeld selbst nicht mehr kovariant. Nur die kinetischen Terme der Teilchen und des σ-Feldes bleiben formal relativistisch.

- Die Vernachlässigung des $\vec{B}^a$-Felds ist übrigens äquivalent zu einer Vernachlässigung des Vektorpotentials $\vec{A}^a$ in den Beziehungen

$$
\begin{aligned}
\vec{E}^a &= -\vec{\nabla}\phi^a - \frac{\partial \vec{A}^a}{\partial t} \\
\vec{B}^a &= \vec{\nabla} \times \vec{A}^a
\end{aligned}
$$

  zwischen Potentialen und Feldern.

- Leider ist es nicht einfach möglich, das vernachlässigte Farbmagnetfeld einmal probeweise auszurechnen; dazu müsste man die Maxwellgleichungen

$$\begin{aligned}\vec{\nabla}\cdot\vec{B} &= 0\\ \vec{\nabla}\times(\kappa\vec{B}) &= \vec{j}+\frac{\partial(\kappa\vec{E})}{\partial t}\end{aligned}$$

  mit geeigneten Randbedingungen lösen. Dies ist weitaus schwieriger zu bewerkstelligen als die Lösung der Poissongleichung. Wir können die Größenordnung des vernachlässigten $\vec{B}^a$-Feldes aber durch eine einfache Überlegung abschätzen: Betrachten wir das Innere eines σ-Feldbags mit $\kappa \approx 1$. Dort gelten nach Kapitel 3 für das Farbfeld die normalen Maxwellschen Gleichungen im Vakuum, mit der Energiedichte $\frac{1}{2}(\vec{E}^a)^2$ des farbelektrischen Feldes und der Energiedichte $\frac{1}{2}(\vec{B}^a)^2$ des farbmagnetischen Feldes. Betrachten wir ein leichtes thermalisiertes Quark bei $T = 150MeV$ (Abb. 4.1). In der Simulation hat dieses eine effektive Masse von $400MeV$, ist also noch in guter Näherung nichtrelativistisch. Die kinetische Energie beträgt $3/2T$, also etwa $225MeV$, so dass für die Rapidität η dieses Quarks gilt

$$\cosh\eta \;=\; \frac{E_{\mathrm{ges}}}{m} \;=\; \frac{625MeV}{400MeV} \;\approx\; 1.56$$

  Auf der Ebene, die das Quark enthält und senkrecht zur Bewegungsrichtung ist, folgt für das Verhältnis von farbmagnetischer und farbelektrischer Energiedichte

$$\frac{(\vec{B}^a)^2}{(\vec{E}^a)^2} = \tanh^2\eta \;=\; 1-\frac{1}{\cosh^2\eta} \;\approx\; 0.6 \quad .$$

  Die farbmagnetische Energiedichte beträgt also auf dieser Ebene etwa das 0.6-fache der farbelektrischen Energiedichte. Fern von der Transversalebene ist dieses Verhältnis sogar kleiner und damit günstiger.

Zusammenfassend lässt sich sagen, dass die Vernachlässigung des Farbmagnetfelds $\vec{B}^a$ eine Vernachlässigung innerer Freiheitsgrade hadronischer Bags bedeutet. Weder der Confinementmechanismus, also die absolute Stabilität der irreduziblen weißen Cluster (IWCs), noch die Wechselwirkung zwischen solchen Bags wird durch diese Approximation verändert: die Bags wechselwirken in keinem Falle miteinander. Für die in dieser Arbeit beabsichtigten Studien ist daher die Vernachlässigung des Farbmagnetfelds ein durchaus akzeptabler Schritt.

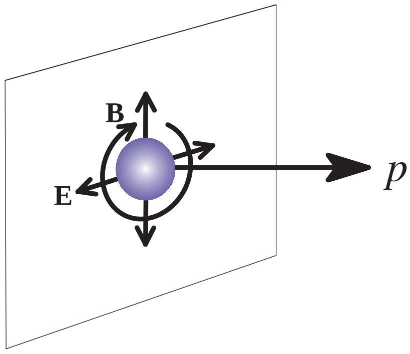


Abbildung 4.1: *$\vec{E}^a$- und $\vec{B}^a$-Feld auf der Transversalebene eines sich im perturbativen Vakuum ($\kappa \approx 1$) bewegenden Quarks. Es gilt $|\vec{B}^a|/|\vec{E}^a| = \tanh\eta$ überall auf dieser Ebene.*

Eine weitere Approximation, die für die numerische Umsetzung notwendig ist, ist die Aufweichung der Bedingung $\kappa(\sigma_{\text{vac}}) = 0$, die die zu lösende Feldgleichung (4.6) singulär werden lässt. Wir benutzen

$$\kappa(\sigma) = \frac{1}{\exp\left(\alpha\left(\frac{\sigma}{\sigma_{vac}} - \beta\right)\right) + 1}$$

für die Form der dielektrischen Funktion. Auf die genaue Wahl der Parameter α und β wird in Kapitel 5 eingegangen; hier soll genügen, dass $\alpha > 0$ und $\beta \in (0, 1)$. Die Simulationsresultate hängen von der konkreten Wahl dieser Funktion nicht stark ab. Insbesondere macht es praktisch keinen Unterschied, ob $\kappa(\sigma_{\text{vac}}) = 0.01$ oder 0.001 gewählt wird — der Energiebeitrag des elektrischen Feldes außerhalb des weißen hadronischen Bags ändert sich dadurch nur minimal.

Dies soll noch etwas näher erklärt werden: Nach 3.47 ist die Feldenergie der Einzelquarkkonfiguration im Fernfeld $\sim 1/\kappa_{\text{vac}}$, ist also sehr sensitiv auf die Wahl von κ_{vac}. Die Einzelquarkkonfiguration ist aber keine farbneutrale Ladungskonfiguration, und als solche erzwingt sie einen Farbfeldfluss ins physikalische Vakuum hinein. Die physikalisch interessanten und relevanten Situationen sind diejenigen, wo die Gesamtfarbe verschwindet, und wir Flusschläuche zwischen den farbigen Komponenten sehen können. Der Feldfluss in Flussschläuchen strömt durch einem Raumbereich mit $\kappa \gg \kappa_{\text{vac}}$ und nicht durch das physikalische Vakuum mit $\kappa = \kappa_{\text{vac}}$. Deshalb spielt für die Feldenergie solcher Konfigurationen der Unterschied zwischen $\kappa(\sigma_{\text{vac}}) = 0.01$ und 0.001 keine große Rolle mehr.

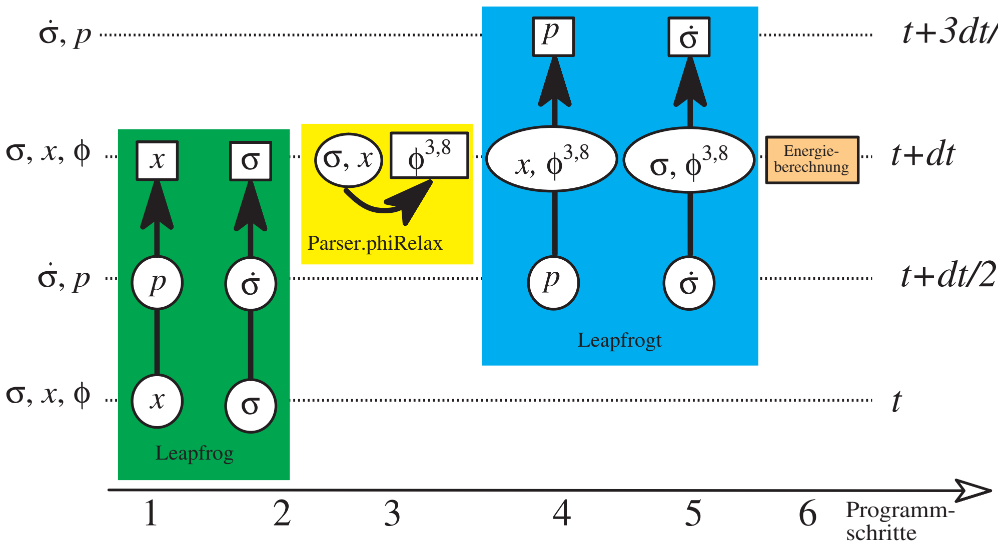


Abbildung 4.2: Berechnungsschema und -reihenfolge eines Simulationszeitschrittes. Die Namen "Leapfrog", "Leapfrogt" und "Parser.phiRelax" beziehen sich auf die Subroutinen im Programm, die die entsprechenden Berechnungen ausführen. Rechtecke markieren Schreibvorgänge, Ovale und Kreise stehen für Lesevorgänge. Man bemerke, dass die Farbfelder (ϕ^a) zwischen Zeitschritten nicht miteinander kommunizieren, weil das Gaußgesetz keine Zeitableitung enthält.

4.2 Behandlung des skalaren Feldes und der Teilchen

4.2.1 Explizite Methode für die Zeitschritte

Die Bewegungsgleichungen, die im Simulationsprogramm CDM numerisch gelöst werden, sind (4.6), (4.7) und (4.9). Die beiden Gleichungen (4.7) und (4.9) werden mit einer simplen *staggered leapfrog*-Methode gelöst. Die zugehörigen Update-"Moleküle" [60] sind in Abb. 4.2 gezeigt. Das σ-Feld, die Teilchenorte und die farbelektrostatischen Potentiale ϕ^a "leben" somit auf den Zeitschritten $\{t, t+\Delta t, t+2\Delta t, \ldots\}$, während das $\dot{\sigma}$-Feld und die Teilchenimpulse auf den halbzahligen Zeitschritten $\{t+\Delta t/2, t+3\Delta t/2, \ldots\}$ lokalisiert sind.
Zunächst werden in Programmschritt 1 und 2 Teilchen und σ-Feld lokal propagiert,

$$(\vec{x}_k)_{t+\Delta t} = (\vec{x}_k)_t + \Delta t \left(\frac{\vec{p}_k}{\sqrt{\vec{p}_k^2 + m^2}} \right)_{t+\Delta t/2} \tag{4.11}$$

$$(\sigma(\vec{x}))_{t+\Delta t} = (\sigma(\vec{x}))_t + \Delta t \, (\dot{\sigma}(\vec{x}))_{t+\Delta t/2} \quad . \tag{4.12}$$

Mit dem in Programmschritt 3 instantan zu lösenden Gaußgesetz (4.6) werden wir uns im nächsten Abschnitt befassen. In Programmschritt 4 und 5 werden die Teilchenimpulse und das $\dot{\sigma}$-Feld propagiert,

$$\begin{aligned}
(\vec{p}_k)_{t+3\Delta t/2} &= (\vec{p}_k)_{t+\Delta t/2} \\
&\quad +\Delta t \left(\sum_{a\in\{3,8\}} \left(-q_k^a \int d^3x \left(\vec{\nabla}\phi^a(\vec{x})\right) \rho_N(\vec{x}-\vec{x}_k)\right)\right)_{t+\Delta t} && (4.13) \\
(\dot{\sigma}(\vec{x}))_{t+3\Delta t/2} &= (\dot{\sigma}(\vec{x}))_{t+\Delta t/2} \\
&\quad +\Delta t \left(\nabla^2\sigma(\vec{x}) - U'(\sigma(\vec{x})) + \frac{\kappa'(\sigma(\vec{x}))}{2} \sum_{a\in\{3,8\}} \left(\vec{\nabla}\phi^a(\vec{x})\right)^2 \right)_{t+\Delta t} && (4.14)
\end{aligned}$$

Das σ-Feld ist auf einem einfachen kartesischen Gitter[3] gespeichert, so dass $\nabla^2\sigma(\vec{x})$ einfach mit Hilfe der Gitternachbarn approximiert werden kann. Wir benutzen die Formel

$$\nabla^2\sigma(x,y,z) = \frac{-\Sigma_2 + 16\Sigma_1 - 90\sigma(x,y,z)}{12h^2} + \mathcal{O}(h^4) \tag{4.15}$$

mit Gitterkonstante Δ und den Hilfsgrößen

$$\begin{aligned}
\Sigma_n &= \sigma(x+nh,y,z) + \sigma(x-nh,y,z) + \sigma(x,y+nh,z) \\
&\quad + \sigma(x,y-nh,z) + \sigma(x,y,z+nh) + \sigma(x,y,z-nh)
\end{aligned}$$

($n = 1,2$). Das zugehörige Gittermolekül ist in Abb. 4.3 abgebildet.
Das beschriebene Zeitschrittverfahren wurde im wesentlichen auch von Thomas Vetter und Stefan Loh [33, 35] verwendet, wobei in den zitierten Arbeiten zusätzliche Komplikationen dadurch entstehen, dass das σ-Feld mit der Teilchenmasse m_k zur "effektiven Masse" $m_k^* = m_k + g_S\sigma(\vec{x}_k)$ koppelt. Diese effektive Masse wird für die Teilchenpropagation $\vec{x}_k \to \vec{x}_k + \vec{p}_k/m_k^*$ benötigt, so dass das σ-Feld an jedem ganz- *und* halbzahligen Teilschritt in die Rechnung eingeht. Dies destabilisiert das Schema; zur Stabilisierung musste daher ein numerischer Reibungsterm [33, 60] eingesetzt werden, der in der vorliegenden Arbeit vollständig entfallen konnte. Trotz fehlenden Reibungsterms ist die Gesamtenergie des simulierten, isolierten Systems in ausgezeichneter Näherung konstant, wie wir in Kapitel 5 zeigen werden.

4.2.2 Berücksichtigung des Pauliprinzips

Zwei Quarks können sich nach dem Pauliprinzip nicht im gleichen Zustand befinden. In einer semiklassischen Übersetzung heißt das, sie können nicht die gleiche Farbe und den gleichen Spin besitzen und darüberhinaus noch die gleiche Zelle im Phasenraum belegen. Eine

[3] Klasse Grid verweist auf ein Feld mit Instanzen der Klasse Site.

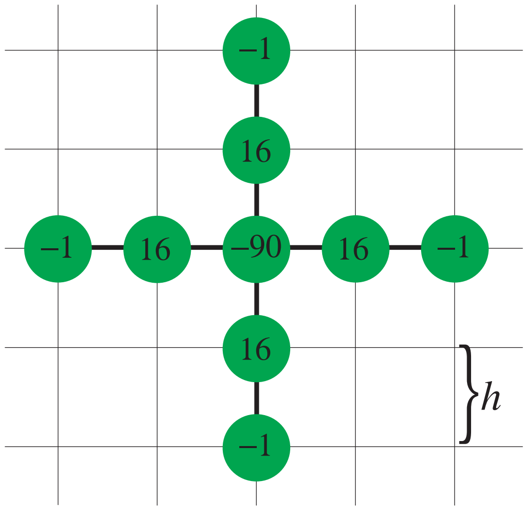


Abbildung 4.3: *Molekül zur numerischen Berechnung des Laplaceoperators* ∇^2 *mit Fehlerordnung* $\mathcal{O}(h^4)$ *auf einem Gitter mit Gitterabstand* h. *Die Gewichte müssen alle mit* $1/(12h^2)$ *skaliert werden.*

Phasenraumzelle hat (semiklassisch) die Größe[4] $h^3 = (2\pi\hbar)^3$. Da wir die Spin- und Isospinfreiheitsgrade nicht explizit in die Simulation einbeziehen, berücksichtigen wir sie für das Pauliprinzip über die Größe der Phasenraumzelle. Zum Beispiel passen vier leichte Quarks der gleichen Farbe ($u\uparrow$, $u\downarrow$, $d\uparrow$, $d\downarrow$) in eine Phasenraumzelle der Größe h^3, was wir einfach dadurch berücksichtigen, dass die Phasenraumzelle für ein leichtes Quark einer Farbe nur die Größe $h^3/4$ bekommt. In diesem kleineren Volumen kann sich dann nur ein einziges leichtes Quark einer Farbe aufhalten. Die konkrete Form dieses Phasenraumvolumens in der numerischen Simulation CDM kann willkürlich gewählt werden; die semiklassische Näherung beruht nicht auf einer bestimmten Wahl.
Programmtechnisch wird das Pauliprinzip wie folgt umgesetzt: In jedem Zeitschritt wird der "Phasenraumabstand"

$$\delta^2 = \left(\frac{\delta^2\vec{x}}{x_0}\right)^2 + \left(\frac{\delta^2\vec{p}}{p_0}\right)^2 \tag{4.16}$$

aller Quarkpaare mit gleicher Farb/Flavorkombination gemessen (konkrete Wahl von x_0 und p_0 s. u.). Ist dieser Abstand kleiner als 1, und bewegen sich die Teilchen außerdem noch aufeinander zu, d. h. würde

[4]Dies ist die übliche semiklassische Annahme; man kann sie theoretisch aus dem Wignerformalismus herleiten [61], wobei die Einstein-de-Broglie-Beziehung

$$\begin{pmatrix} E \\ \vec{p} \end{pmatrix} = \hbar \begin{pmatrix} \omega \\ \vec{k} \end{pmatrix}$$

eingeht, oder auch experimentell als Definition des Planckschen Wirkunsquantums h auffassen.

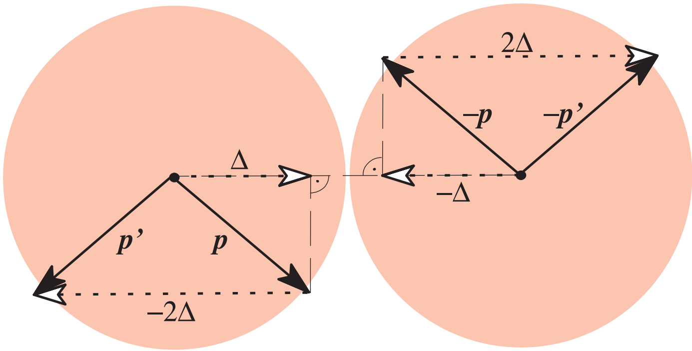


Abbildung 4.4: *Elastische, harte Streuung zweier gleichfarbiger Quarks im Schwerpunktsystem. Impulse vor dem Stoß* $\pm\vec{p}$, *nachher* $\pm\vec{p}'$, *Impulsübertrag* 2Δ *entlang der Verbindungslinie. Kinetische Energie und Gesamtimpuls sind erhalten.*

sich der Abstand bei freier Propagation noch weiter verkleinern, so werden die Quarks wie Billardkugeln elastisch aneinander gestreut. Dabei tauschen sie im Schwerpunktsystem einen Impuls aus, der genau doppelt so groß ist wie die Projektion eines Impulses auf die Verbindungslinie zwischen den Teilchen (siehe Abb. 4.4). Der gesamte Viererimpuls bleibt dabei erhalten. Nach der elastischen Streuung entfernen sich die Teilchen im Phasenraum wieder voneinander.

Eine einzige Streuung kann die Erfüllung des Pauliprinzips aber nicht garantieren: es ist immer noch möglich, dass bei der Paulistreuung zweier Quarks (gleicher Farbe und Sorte) eines von beiden in die unmittelbare Nachbarschaft eines dritten Quarks (ebenfalls gleicher Farbe und Sorte) hineingestreut wird, wodurch das Ausschließungsprinzip wieder verletzt wäre. Um dies zu verhindern, wird eine *instantane Paulikaskade* simuliert: Wir streuen *noch im gleichen Zeitschritt* der Simulation erneut. Wird also z. B. A an B gestreut und gerät dadurch in die Phasenraumnähe von C, so wird *noch im gleichen Zeitschritt* erneut A an C gestreut. So führt eine einzige anfängliche Verletzung des Pauliprinzips zu einer (instantanen) Kaskade von Streuvorgängen. In der Praxis unserer Simulationsläufe geschieht aber selbst eine einzige Paulistreuung bereits äußerst selten (weniger als zehnmal während eines Hadronisierungslaufs), da es bei hohen Temperaturen die Quarks im Impulsraum meist weit voneinander entfernt sind. Mehrstufige Kaskaden kamen in unseren Simulationsläufen bisher überhaupt nicht vor. Trotzdem taugt unser Ansatz der Paulikaskade prinzipiell dazu, das Pauliprinzip im Modell zu forcieren (wenn auch auf eine nichtkovariante Weise).

Nun zur Wahl von x_0 und p_0 in (4.16). Mit dem beschriebenen Streumechanismus wird für jedes Quark garantiert, dass die ellipsoidförmige Phasenraumzelle $\delta^2 < 1$ frei von Quarks derselben Color-/Flavorkombination ist (abgesehen von dem erwähnten Problem, dass in die Nachbarschaft eines gleichfarbigen Quarks hineingestreut werden könnte). Dieser 6-dimensionale Ellipsoid hat das Volumen

$$V_p = \frac{\pi^3}{6}(x_0 p_0)^3 \quad . \tag{4.17}$$

Setzen wir nun x_0 willkürlich gleich dem zweifachen RMS-Radius

$$\begin{aligned} x_0 &= 2\sqrt{\langle \vec{r}^2 \rangle_{\rho_N}} \\ &= 2\sqrt{3r_0^2} \end{aligned} \tag{4.18}$$

der zum Teilchen gehörenden gaußförmigen Anzahldichte (4.3), so streuen die Teilchen wie Billardkugeln mit Radien $\sqrt{\langle \vec{r}^2 \rangle}$. Durch die Bedingung

$$V_p = (2\pi\hbar)^3/K$$

(mit Spin/Flavor-Entartungsfaktor K) bestimmen wir mit (4.17) und (4.18) den Impulsradius der Phasenraumzelle zu

$$\begin{aligned} p_0 &= \frac{2 \cdot 6^{1/3}\hbar}{x_0 K} \\ &= \frac{6^{1/3}\hbar}{3^{1/2} r_0 K} \quad . \end{aligned} \tag{4.19}$$

Die Größen x_0 und p_0 aus (4.18) und (4.19) werden zur Bestimmung des Phasenraumabstands (4.16) benutzt.
Eine Nachbemerkung hierzu: Die numerische Behandlung des Pauliprinzips über eine effektive klassische Wechselwirkung ist keine neue Idee; sie wurde u. a. bereits von Wilets eingesetzt, allerdings mit Hilfe einer stetigen Wechselwirkung. Deren konkrete Form ist allerdings genauso willkürlich wie der hier gewählte "relativistische Billardkugel"-Ansatz.

4.3 Behandlung des Farbfeldes

4.3.1 Problemstellung

Nach den im Abschnitt 4.1.2 beschriebenen Approximationen verbleibt die Aufgabe, das Gaußgesetz (4.6),

$$\vec{\nabla} \cdot \left(\kappa(\sigma) \vec{\nabla} \phi^a \right) = -\rho^a \quad , \tag{4.20}$$

mit geeigneten Randbedingungen ($\phi^a = 0$ im Unendlichen) zu jedem Zeitschritt erneut zu lösen. Die rechte Seite dieser Gleichung ist die von den Quarks, Antiquarks und "farbigen" Gluonen erzeugte Farbladungsdichte. Das σ-Feld folgt selbst einer Wellengleichung, deren numerische

Behandlung in Abschnitt 4.2 bereits beschrieben wurde. Somit ist die dielektrische Funktion $\kappa(\sigma(\vec{x}))$ räumlich und zeitlich veränderlich. Der lineare Differentialoperator in (4.20) ist also nicht translationsinvariant, was die schnelle Methode der Fouriertransformation für die Lösung von (4.20) ausschließt [60]. Hinzu kommt, dass κ im physikalischen Vakuum nahezu verschwindet, was (4.20) beinahe singulär macht — falls κ in einer Raumregion exakt verschwinden würde, wäre ϕ^a durch (4.20) gar nicht bestimmt. Diese Beinahe-Singularität der Gleichung schließt auch Operator-Splitting-Methoden zur Lösung aus. Successive-Overrelaxation-Schemata (SOR), die Standardmethode der 70er Jahre zur Lösung von Randwertproblemen, sind zwar anwendbar, aber sehr schwer zu optimieren und besitzen im besten Fall eine Zeitkomplexität von $\mathcal{O}(N^4)$ auf einem numerischen Gitter mit N^3 Gitterpunkten [60], was für unsere Zwecke viel zu langsam ist.

Für die vorliegende Arbeit wurde daher zur Lösung der partiellen Differentialgleichung (4.20) das hierarchische Mehrgitterverfahren auf Basis der Finite-Elemente-Methode mit adaptiver Verfeinerung gewählt [62]. Dieses Verfahren ist *nicht state-of-the-art* (dies wären parallele Mehrgitterverfahren [63]), aber seit etwa 15 Jahren existent, etabliert und im Vergleich zu neueren Methoden noch relativ leicht zu verstehen. Stefan Loh verwendete in seiner Dissertation für die Lösung des gleichen Randwertproblems eine Subroutine von Mitchell [64], die ebenfalls auf dem hierarchischen Mehrgitterverfahren beruhte, allerdings auf Randwertprobleme in zwei Dimensionen beschränkt war.

4.3.2 Diskretisierung mit der Methode der finiten Elemente

4.3.2.1 Variationsverfahren

Die Methode der finiten Elemente wird oft benutzt, um partielle Differentialgleichungen zu diskretisieren. Obwohl die Methode wohlbekannt ist, soll sie hier kurz dargestellt werden, wobei wir auf die konkrete Problemstellung (Lösung von (4.20)) Bezug nehmen. Basis der Methode ist das Ritz-Galerkinsche Variationsverfahren: Zunächst schreibt man die partielle Differentialgleichung in ihre *schwache Form* um: statt die Erfüllung von (4.20) (bzw. gewisse Randbedingungen) an jedem Raumpunkt zu fordern, fordern wir stattdessen, dass die Projektion

$$\int \eta(\vec{x}) \left(\vec{\nabla} \cdot \left(\kappa(\sigma(\vec{x}))\vec{\nabla}\phi^a(\vec{x})\right) + \rho^a(\vec{x})\right) d^3x \;=\; 0 \tag{4.21}$$

des Residuums auf eine beliebige *Testfunktion* $\eta(\vec{x})$ verschwindet. Wir fordern sozusagen die Erfüllung der Differentialgleichung im Distributionssinn. Wählen wir nun eine endliche Basis von Funktionen $\{\eta_i(\vec{x})\}$

für das Variationsverfahren und entwickeln die Lösungsfunktion näherungsweise in dieser Basis,

$$\phi^a(\vec{x}) = \sum_i \phi_i^a \eta_i(\vec{x}) \quad , \tag{4.22}$$

so wird die Lösung von (4.21) i. a. nicht im Raum der durch $\{\eta_i(\vec{x})\}$ aufgespannten Funktionen liegen. Wir müssen daher die Bedingung (4.21) zu einem endlichen Gleichungssystem abschwächen. Dies geschieht, indem wir als Testfunktionen auch nur genau diese $\{\eta_i(\vec{x})\}$ betrachten. Damit wird (4.21) zu

$$\int \eta_i(\vec{x}) \left(\vec{\nabla} \cdot \left(\kappa(\sigma(\vec{x})) \vec{\nabla} \sum_j \phi_j^a \eta_j(\vec{x}) \right) + \rho^a(\vec{x}) \right) d^3x = 0 \quad . \tag{4.23}$$

Durch partielle Integration wird hieraus das Gleichungssystem

$$\sum_j D_{ij} \phi_j^a = r_i^a \tag{4.24}$$

mit

$$D_{ij} = \int \kappa(\sigma(\vec{x})) \left(\vec{\nabla} \eta_i(\vec{x}) \right) \cdot \left(\vec{\nabla} \eta_j(\vec{x}) \right) d^3x \tag{4.25}$$

$$r_i^a = \int \eta_i(\vec{x}) \rho^a(\vec{x}) d^3x \quad . \tag{4.26}$$

Damit ist das ursprüngliche Problem der Lösung einer Differentialgleichung reduziert auf die Lösung des linearen Gleichungssystems (4.24). Randbedingungen generieren zusätzliche Gleichungen (siehe Abschnitt 4.3.4.9).
Man bemerke, dass in (4.25) nur erste Ableitungen von Basisfunktionen auftreten, obwohl das zugrundeliegende Randwertproblem von zweiter Ordnung ist. Es genügt also, stetige und stückweise differenzierbare Ansatzfunktionen $\eta_i(\vec{x})$ zu benutzen. Genau dies werden wir im folgenden tun.

4.3.2.2 Finite Elemente als Basisfunktionen

Soweit zum Variationsverfahren. Die Methode der finiten Elemente besteht in einer geschickten Wahl der Basis $\{\eta_i(\vec{x})\}$. Hier gibt es eine Vielfalt an praktikablen Möglichkeiten. Sie alle haben aber folgendes gemeinsam: Das Gebiet Ω, das dem Randwertproblem zugrundeliegt, wird zerlegt (*partitioniert*), meist in *Simplices*, also Dreiecke, Tetraeder usw. je nach Dimensionalität des Gebiets. Jeder Simplex besitzt an seinen Ecken und Kanten *Knoten* (Stützstellen), die er mit den umliegenden Simplices teilt. Jede der Basisfunktionen $\{\eta_i(\vec{x})\}$ lässt sich eindeutig einem Knoten zuordnen. Dieser Knoten heißt dann auch *Trägerknoten*

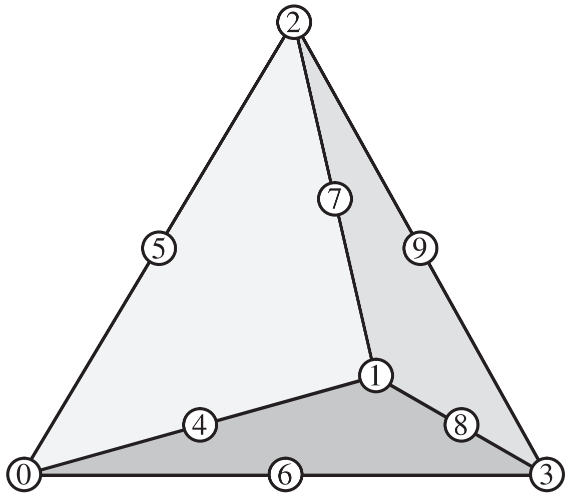


Abbildung 4.5: Ein Simplex mit seinen Knoten. Das Nummerierungsschema wird konsequent eingehalten.

(*supporting node*) der Basisfunktion. Meist wählt man die Basisfunktionen so, dass sie an ihrem Trägerknoten den Wert 1 annehmen, über den Raumbereich der anliegenden Simplices stetig abfallen und an allen anderen Knoten den Wert 0 annehmen. Im Raumbereich außerhalb der am Trägerknoten anliegenden Simplices verschwindet die Basisfunktion völlig, mit anderen Worten: Der Träger (*support*) einer Basisfunktion, also das Gebiet, in dem sie nicht Null ist, besteht genau aus den Simplices, die den Trägerknoten enthalten.

Aus dieser Eigenschaft folgt direkt, dass die Matrix D_{ij} *dünn besetzt* (*sparse*) ist: Matrixelemente D_{ij} können nur dann von Null verschieden sein, wenn es mindestens einen Simplex gibt, der die Trägerknoten der Basisfunktionen η_i und η_j beide enthält. Wegen des endlichen Trägers (*support*) heißen die Basisfunktionen auch *finite Elemente*. Es gibt gleich viele Basisfunktionen wie Knoten, und damit genauso viele Unbekannte und Gleichungen in (4.24).

Ein Simplex[5] ist in unserem Falle einfach eine Dreieckspyramide (asymmetrischer Tetraeder). Wir bringen Knoten an jeder Ecke und in der Mitte jeder Kante des Simplexes an. Dabei wird die Nummerierungskonvention nach Abb. 4.5 stets eingehalten: Jeder Simplex hat genau zehn Knoten, nämlich die Eckknoten, nummeriert von 0 bis 3, und die Kantenknoten, nummeriert von 4 bis 9. Dabei ist stets der Knoten 6 in der Mitte zwischen Knoten 0 und 3 usw. Es genügt also, die Eckknoten zu zeichnen, um die Lage der Kantenknoten daraus schließen zu können. Die Parität ist allerdings nicht festgelegt; es ist also nicht immer so, dass vom Knoten 0 aus die Achsen in Richtung der Knoten 1, 2 und 3 ein rechtshändiges System bilden.

[5] Klasse Simplex

Wir arbeiten mit einem würfelförmigen Gebiet[6], das anfänglich aus sechs kongruenten Simplices (sog. *Kuhnsimplices*) zusammengesetzt ist (Abb. 4.6).

4.3.2.3 Simplexkoordinaten und lineare Basis

Eine mathematische Notation für ein Simplex ist ein *geordnetes Quadrupel* $S = (\vec{n}_0, \vec{n}_1, \vec{n}_2, \vec{n}_3)$ von Eckpunkten oder Eckknoten (*corner nodes*). Die Position $i = 0 \dots 3$ eines Knotens innerhalb dieses Quadrupels nennen wir *Knotenindex*. Das Simplexvolumen $\mathcal{V}_S$ ist die konvexe Hülle der Eckpunkte,

$$\mathcal{V}_S \;:=\; \left\{ \vec{x} \in \mathbb{R}^3 \;\middle|\; \vec{x} = \sum_{i=0}^{3} \alpha_i \vec{n}_i \text{ mit } \alpha_i > 0,\; \sum_{i=0}^{3} \alpha_i = 1 \right\} \tag{4.27}$$

Diese Darstellung der Simplexpunkte als Linearkombination der Ecken mit positiven Koeffizienten α_i, die sich zu 1 summieren, ist sogar eindeutig. Wir können daher die *Simplexkoordinaten* für den Simplex S wie folgt definieren: Sei, für $i \in \{0 \dots 3\}$,

$$\alpha_i(\vec{x}) : \; \mathbb{R}^3 \mapsto \mathbb{R} \tag{4.28}$$

die eindeutige lineare Funktion mit

$$\alpha_i(\vec{n}_j) \;=\; \delta_{ij} \quad (i, j \in \{0 \dots 3\}). \tag{4.29}$$

Dann gilt $\sum_i \alpha_i \equiv 1$ überall, und wir können jeden beliebigen Punkt $\vec{x} \in \mathcal{V}_S$ eindeutig durch ein Quadrupel[7] $(\alpha_0 \dots \alpha_3) \in [0,1]^4$ mit $\sum_i \alpha_i = 1$ darstellen (zu dieser Standardmethode siehe auch [65], S. 212ff). Beispiele: Den vier Eckknoten werden die Quadrupel $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ und $(0,0,0,1)$ zugeordnet, der Kantenknoten $\vec{n}_6$ (in der Mitte zwischen $\vec{n}_0$ und $\vec{n}_3$ siehe Schema Abb. 4.5) bekommt das Quadrupel $(0.5, 0, 0, 0.5)$ zugeordnet, und der Schwerpunkt des Simplex hat die Koordinaten $(0.25, 0.25, 0.25, 0.25)$. Diese sog. *Simplexkoordinaten* oder *lokalen Koordinaten* sind uns im folgenden überaus nützlich. Die Hin- und Rückkonvertierung zwischen Quadrupeln und kartesischen Koordinaten ist mit nur minimalem Rechenaufwand verbunden[8]. Übrigens lässt sich mit Hilfe der α_i leicht testen, ob ein beliebiger Punkt $\vec{x}$ im Simplexvolumen $\mathcal{V}_S$ enthalten ist: man rechnet $\vec{x}$ in Simplexkoordinaten um und überprüft, ob alle vier Zahlen des Quadrupels positiv sind.

---

[6] Von der Klasse List⟨Simplex⟩ abgeleitet ist die Klasse Domain. Die spezielle Klasse Cube, die das würfelförmige Gebiet aus Abb. 4.6 beschreibt, erbt nicht nur von Domain, sondern auch von Grid und ParticleList und bildet somit die Grundlage der Simulation.

[7] Klasse quadrupel, deklariert in localmap.h.

[8] Optimierte Konvertierungsroutinen, die auch noch einige Tricks mehr können, finden sich (und sind kommentiert) in localmap.cpp in der Klasse LocalMap gekapselt.

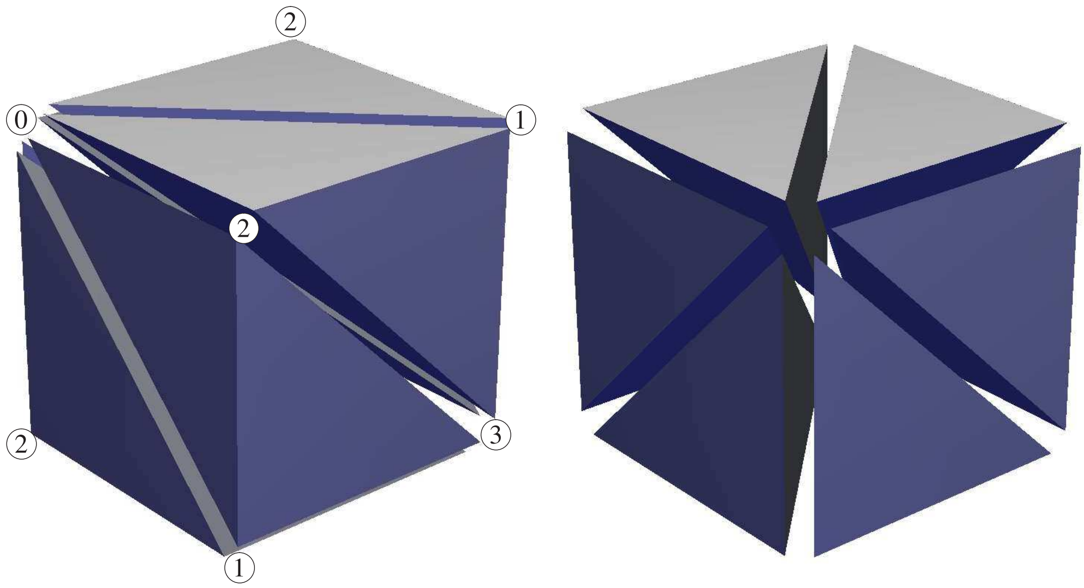


Abbildung 4.6: *Partition eines würfelformigen Gebiets in sechs kongruente Simplices ("Kuhnsimplex"). Die Eckknoten haben in jedem der Simplices den gleichen Knotenindex, was nach dem Schema aus Abb. 4.5 auch für die Kantenknoten gilt.*

Für einen Variationsansatz mit *stückweise linearer Finite-Elemente-Basis* sind die α_i hilfreich: In diesem Fall befinden sich Trägerknoten nur an jeder Simplexecke[9]. Die zum Trägerknoten $\vec{n}$ gehörende stückweise lineare Basisfunktion wird dann wie folgt definiert:

$$\eta^L_{\vec{n}}(\vec{x}) = \begin{cases} \alpha_i(\vec{x}) & \text{falls } \vec{x} \text{ im Simplex } S \text{ liegt} \\ & \text{und } \vec{n} \text{ der Knoten } \vec{n}_i \text{ in } S \text{ ist.} \\ 0 & \text{sonst.} \end{cases} \tag{4.30}$$

Die Simplexkoordinaten α_i beziehen sich dabei auf den Simplex S, der $\vec{x}$ enthält.
Ein Beispiel für eine solche Basisfunktion η_L ist für den zweidimensionalen Fall in Abb. 4.7 abgebildet. Man bemerke, dass die zeltförmige Basisfunktion an allen Knoten verschwindet, mit der einzigen Ausnahme des Trägerknotens, wo sie den Wert 1 annimmt. Die Wahl einer Basis in dieser Weise ist aus zwei Gründen besonders bequem: Erstens, wie bereits erwähnt, sind Matrixelemente lokaler Differentialoperatoren zwischen solchen Basisfunktionen oft trivialerweise gleich Null, nämlich immer dann, wenn die Träger der beteiligten Basisfunktionen disjunkt sind, d. h. wenn die Trägerknoten nicht an mindestens einem gemeinsamen Simplex anliegen, verschwindet das Matrixelement. Zweitens kann man

[9] Im Falle quadratischer Ansatzfunktionen befinden sich Trägerknoten an den Eckpunkten und Kantenmitten des Simplexes, siehe Abb. 4.5.

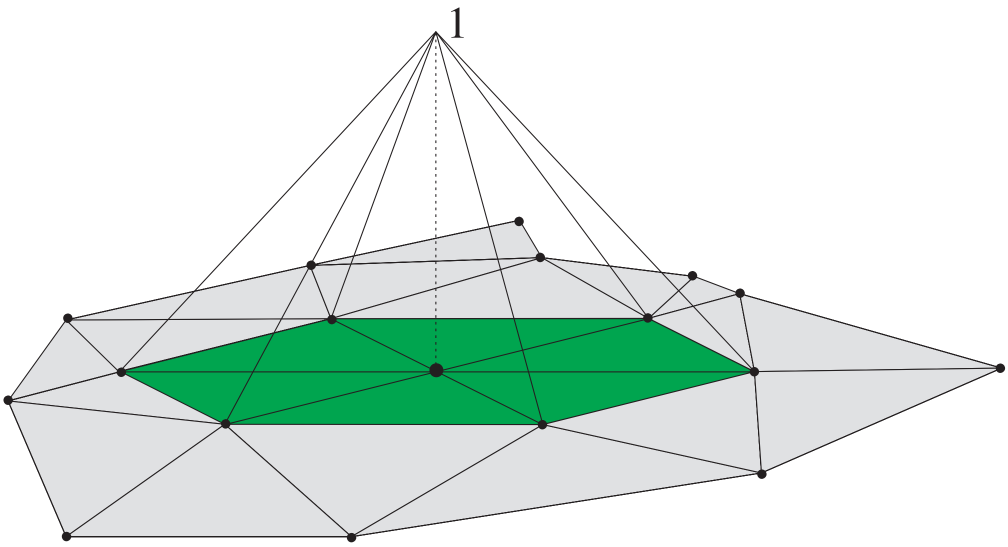


Abbildung 4.7: Stückweise lineare Finite-Elemente-Basisfunktion. Die markierten Punkt sind die Knotenpunkte des zweidimensionalen Gebietes; in der senkrechten Richtung sind die Funktionswerte aufgetragen. Die "zeltförmige" Basisfunktion verschwindet überall bis auf die sechseckige Fläche in der Mitte (Träger), die auch den Trägerknoten (großer Punkt) enthält. Am Trägerknoten hat die Funktion den Wert 1, *an allen anderen Knoten den Wert* 0.

eine beliebige, stückweise lineare Funktion $f : \Omega \mapsto \mathbb{R}$ einfach in der Basis entwickeln:

$$f(\vec{x}) \;=\; \sum_{\text{Knoten } \vec{n}\in\Omega} f(\vec{n})\eta^L_{\vec{n}}(\vec{x}) \quad . \tag{4.31}$$

(Diese Beziehung ist als Approximation immer nützlich, aber nur exakt, wenn f stückweise (auf jedem Simplex) linear ist.)

4.3.2.4 Quadratische Ansatzfunktionen

Zur numerischen Lösung von (4.6) werden in MUESLEA[10] stückweise quadratische Ansatzfunktionen benutzt, die im Vergleich zu linearen glatte Funktionen mit weniger Knoten bereits gut approximieren können. Knoten sitzen hier nicht nur an den Eckpunkten der Simplices, sondern auch im Mittelpunkt jeder Simplexkante (siehe Abb. 4.5). Die Basisfunktionen lassen sich mit Hilfe der Simplexkoordinaten α_i aus dem letzten Abschnitt leicht definieren:

$$\eta_{\vec{n}}(\vec{x}) \;=\; \begin{cases} \alpha_i(\vec{x})(2\alpha_i(\vec{x})-1) & \text{falls } \vec{x} \text{ im Simplex } S \text{ liegt} \\ & \text{und } \vec{n} = \vec{n}^S_i. \\ 4\alpha_i(\vec{x})\alpha_j(\vec{x}) & \text{falls } \vec{x} \text{ im Simplex } S \text{ liegt} \\ & \text{und } \vec{n} = (\vec{n}^S_i + \vec{n}^S_j)/2 \text{ (Kantenknoten).} \\ 0 & \text{sonst.} \end{cases} \tag{4.32}$$

Die Simplexkoordinaten α_i beziehen sich dabei wieder auf den Simplex S, der $\vec{x}$ enthält. Mit $\vec{n}^S_i$ ist der i-te Knoten des Simplexes S gemeint[11]. Im zweiten Fall ist also $\vec{n}$ der Kantenknoten, der in der Mitte zwischen den Eckknoten $\vec{n}^S_i$ und $\vec{n}^S_j$ liegt (also zum Beispiel Knoten $\vec{n}_6$ in der Mitte zwischen $\vec{n}_0$ und $\vec{n}_3$, siehe Abb. 4.5). In einer zulässigen (*kompatiblen*) Gebietspartition stößt jeder Simplex mit jeder seiner vier Seitenflächen an (höchstens) einen Nachbarsimplex an. Deshalb ist ein Knoten, der in einem Simplex Kantenknoten ist, im Nachbarsimplex garantiert auch ein Kantenknoten, und entsprechend gilt dies auch für Eckknoten. Damit wird durch (4.32) eine stetige (aber nicht differenzierbare!) und stückweise glatte Basisfunktion definiert. Graphische Darstellungen finden

[10] MUESLEA, *MUltigrid partial differential Equation SoLvEr with Adaptive refinement*, ist der adaptive Finite-Elemente-Mehrgitter-Differentialgleichungslöser, der für das Simulationsprogramm CDM entwickelt wurde. MUESLEA ist damit ein Teil von CDM. Der Name ist übrigens absichtlich an die bekannte Frühstücksformel angelehnt: genau wie man sich aussuchen kann, welches Obst man in sein Müsli mischt, sucht sich MUESLEA aus, welche Simplices und Ansatzfunktionen es für die Lösung einer Differentialgleichung hinzumischt. Der Name steht also für Adaptivität.

[11] Man erkennt: Das Problem mit der Finite-Elemente-Methode ist das *book-keeping* aller beteiligten Simplices, Knoten, Funktionen und Matrizen. Dementsprechend aufwendig ist die Notation.

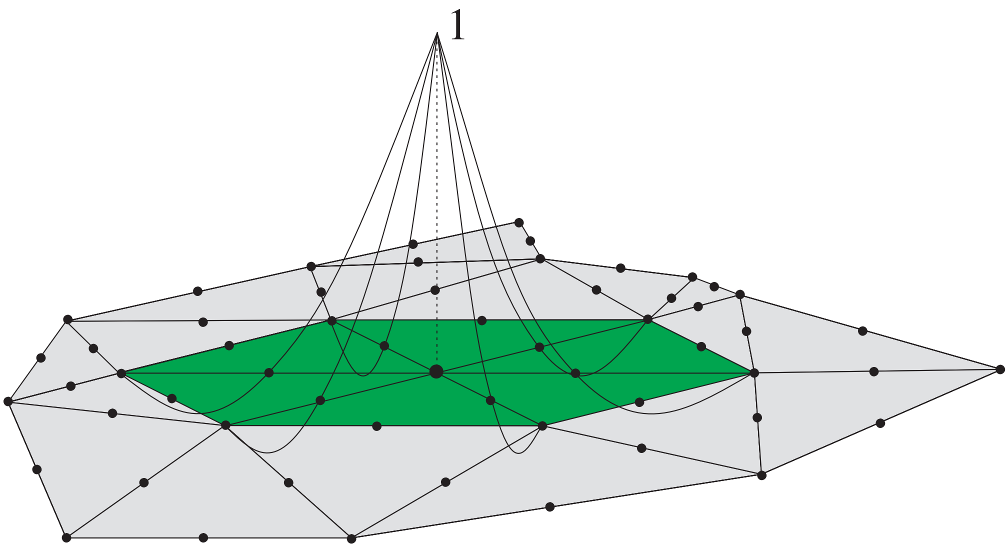


Abbildung 4.8: Stückweise quadratische Finite-Elemente-Basisfunktion. Die markierten Punkt sind die Knotenpunkte des zweidimensionalen Gebietes; in der senkrechten Richtung sind die Funktionswerte aufgetragen. Die sechseckige Fläche in der Mitte ist wieder der Träger und enthält den Trägerknoten (großer Punkt), einen Eckknoten. Am Trägerknoten hat die Funktion den Wert 1, an allen anderen Knoten den Wert 0. Man bemerke, dass die Ansatzfunktion zu negativen Funktionswerten hin "überschwingt".

sich für den zweidimensionalen Fall in Abb. 4.8 und Abb. 4.9 für Eck- und Kantenknoten als Trägerknoten. Die Entwicklungsformel

$$f(\vec{x}) \;=\; \sum_{\text{Knoten } \vec{n}\in\Omega} f(\vec{n})\eta_{\vec{n}}(\vec{x}) \quad . \tag{4.33}$$

ist mit der Definition (4.32) exakt für beliebige stückweise (auf jedem Simplex) quadratische Funktionen $f : \Omega \mapsto \mathbb{R}$.
Übrigens ist die Wahl der Ansatzfunktionen nach (4.32) eindeutig, wenn die Entwicklungsformel (4.33) gelten soll: (4.33) gilt nämlich nur, wenn die Funktionen an ihrem Trägerknoten den Wert 1 annehmen und an allen anderen Knoten verschwinden. Betrachten wir einen einzelnen Simplex, so legt diese Bedingung ein Gleichungssystem mit 10 Gleichungen fest (4 für die Eckknoten und 6 für die Kantenknoten). Auf einem einzelnen Simplex soll die Ansatzfunktion aber quadratisch sein. Da der Raum der quadratischen Polynome in drei Variablen

$$\langle 1, x, y, z, x^2, y^2, z^2, xy, yz, xz \rangle \tag{4.34}$$

aber genau 10-dimensional ist, und die gesuchten Ansatzfunktionen mit

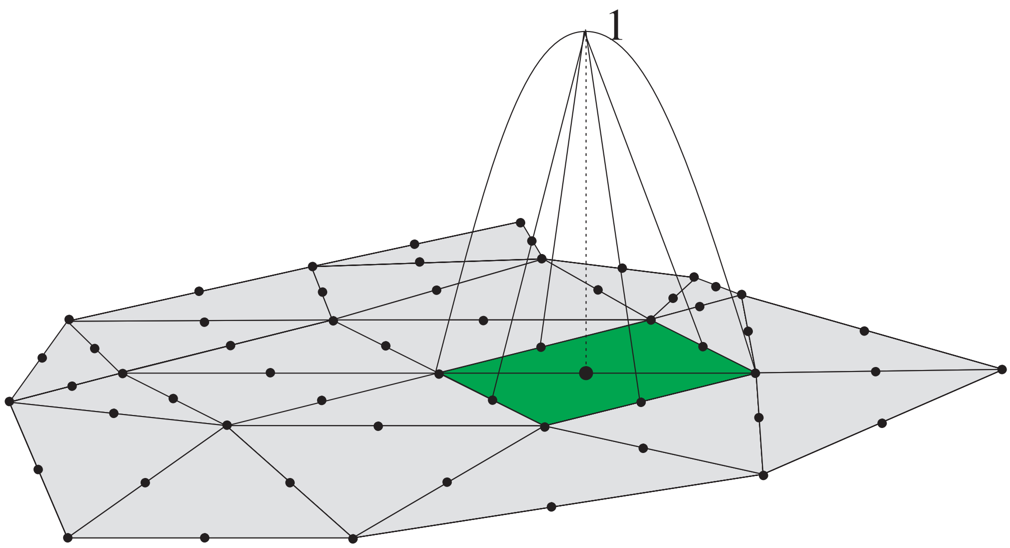


Abbildung 4.9: Stückweise quadratische Finite-Elemente-Basisfunktion für einen Kantenknoten als Trägerknoten. Der Träger besteht hier nur aus zwei Simplices.

Sicherheit linear unabhängig sind, gibt es stets genau eine Lösung für das o. g. Gleichungssystem.
Man bemerke, dass die Ansatzfunktion für den Fall eines Eck-Trägerknotens auf einem Teilgebiet sogar negativ wird (siehe Abb. 4.8). Dies kann dazu führen, dass eine positiv definite, aber nicht stückweise quadratische Funktion, die nur auf den Knoten bekannt ist, durch die Formel (4.33) derart schlecht approximiert wird, dass ihre Approximation nicht mehr positiv definit ist. Dieser Umstand wird uns später noch Probleme bereiten.
Die Wahl der Basisfunktionen als stückweise quadratisch ist nicht die einfachste, hat aber große Vorteile, wenn man bedenkt, dass für lineare Funktionen achtmal mehr Simplices erforderlich wären, um die gleiche Zahl der Knoten zu erreichen; selbst dann wäre die Qualität der Approximation einer glatten Funktion aber nicht so gut wie die mit dem quadratischen Ansatz erreichte.

4.3.2.5 Diskussion

Die Diskretisierungsmethode der finiten Elemente ist von der Methode der finiten Differenzen, die unter Physikern sehr beliebt ist, sehr verschieden: dort wird die Erfüllung einer diskretisierten ("interpolierten") Version der Differentialgleichung an jedem Gitterpunkt gefordert. Bei finiten Elementen arbeiten wir in einem bekannten linearen Unterraum

des Raums der stetigen Funktionen und fordern eine abgeschwächte ("projizierte") Form der Differentialgleichung in diesem Unterraum. Wir haben es dabei stets mit stückweise glatten und überall stetigen Ansatzfunktionen zu tun. An die Stelle der Interpolation tritt hier die Projektion. Erwähnenswert und interessant ist die Beobachtung, dass bei gewissen speziellen Gebietspartitionen die resultierenden Gleichungen beider Methoden identisch sind [60].
Im folgenden sollen die relativen Vor- und Nachteile der beiden Methoden, finite Elemente (FEM) und finite Differenzen (FDM), diskutiert werden. Da es von beiden numerischen Methoden mittlerweile unzählige Varianten gibt, beschränken wir uns zum Vergleich der Diskretisierungsmethoden FEM und FDM auf Relaxationsverfahren zur Lösung elliptischer Randwertprobleme (z. B. Lösung der Poissongleichung). Die Wahl eines Lösungsverfahrens ist eine von der Wahl der Diskretisierungsmethode zunächst getrennte Entscheidung, wenn auch die Relaxationsverfahren sicherlich die populärsten sind.

- Die FEM erlaubt uns die einfache Spezifikation von Randbedingungen auf unregelmäßigen Gebietsrändern. Dies ist im Falle der FDM mit i. d. R. kartesischen Gittern nicht einfach zu bewerkstelligen; in der Nähe eines unregelmäßigen Gebietsrandes tauchen durch die ungeeignete kartesische Anordnung der Knotenstellen relativ große numerische Fehler auf. In der Tat hat genau dieser Defizit der FDM ursprünglich zum Erfolgszug der FEM geführt, auch wenn wir in CDM diese vorteilhafte Eigenschaft der FEM gar nicht benötigen.

- Die FEM eröffnet zudem einen konzeptionell leicht verständlichen Weg, Differentialgleichungen auf unregelmäßigen Netzen zu diskretisieren. Dies ermöglicht eine lokale, adaptive Verfeinerung (wird im nächsten Abschnitt behandelt), die letztlich zur effizienteren Nutzung der Rechenzeit beiträgt. Die FDM erlaubt für Randwertprobleme keine adaptive Verfeinerung; zumindest ist dem Autor kein allgemeingültiges Verfahren dafür bekannt, wie der Übergang zwischen Gittern verschiedener Feinheit mit Differenzenverfahren zu behandeln ist.

- Die FEM arbeitet ohne eine Interpolation der Lösungsfunktionen zwischen den Knoten. Stattdessen wird die Lösungsfunktion stets durch eine im ganzen Gebiet definierte und stückweise glatte Näherungsfunktion beschrieben. Eine direkte Konsequenz dessen ist, dass eine solide Theorie der Fehlerabschätzung für die FEM existiert. Im Fall der FDM ist man meist darauf angewiesen, Kontrollrechnungen mit unterschiedlichen Gittergrößen zu machen und die Resultate miteinander zu vergleichen.

- All die Vorteile der FEM haben ihren Preis: die Methode ist kompliziert, es ist viel buchhalterischer Aufwand mit der Verwaltung der Simplices, Knoten, Matrizen usw. verbunden, und nicht für alle Problemstellungen kann die FEM ihre Stärken wirklich ausspielen.

- Zum Schluss ein Wort der Warnung: Natürlich konvergieren beide Verfahren für ein gegebenes Problem zur gleichen Lösung, falls sie überhaupt konvergieren. Macht man das Netz hinreichend fein, so lässt sich auch in jedem Fall die gewünschte Genauigkeit erreichen. Dieses Denken führt den Numeriker aber nicht weiter, sobald er sich vor ein Randwertproblem gestellt sieht, dessen Gebiet eine Dimensionalität von mehr als zwei besitzt. In drei Raumdimensionen benötigt ein feines Netz schnell sehr viel Speicherplatz: ein 200^3-Netz braucht schon $64MB$, um nur ein einziges skalares Feld mit seiner Zeitableitung zu speichern. Wobei 200^3 nicht gerade besonders fein ist, gemessen an den Zahlen, die man gerne in zwei Raumdimensionen nimmt. Da selbst für gutoptimierte Relaxationsverfahren (nicht Mehrgitterverfahren) üblicherweise die Anzahl der nötigen Iterationen mit $1/ds$ skaliert (ds die Gitterkonstante) [60], sind einfache Relaxationsverfahren in drei Raumdimensionen in der Regel unbrauchbar. Diese Warnung bezieht sich gleichermaßen auf FEM und FDM. Mehrgitterlösungsverfahren gibt es für beide Diskretisierungsmethoden.

- Letztlich möchte man natürlich die Summe von Programmierzeit und Rechenzeit minimieren und wird deshalb i. a. der FDM den Zuschlag geben. Im Fall der CDM-Simulation jedoch stellte sich das Bildladungsproblem wegen der im Modell enthaltenen langreichweitigen Wechselwirkungen als so gravierend heraus, dass m. E. nicht auf adaptive Verfeinerung verzichtet werden kann: wir vermindern den Einfluss von Spiegelladungen auf die Simulation, indem wir den Rand des Gebiets, auf dem das Vektorfeld berechnet wird, weiter hinausschieben (siehe Abschnitt 4.4). Dadurch wird das Gebiet etwa um einen Faktor 5 vergrößert, was aber durch die adaptive Methode nicht dementsprechend mehr Rechenzeit beansprucht. Dies ist ein schlagendes Argument, um in unserem Fall die FEM, in der es einen natürlichen Weg der adaptiven Verfeinerung gibt, der FDM vorzuziehen[12].

[12]Natürlich spielten bei dieser Entscheidung auch "institutshistorische" Gründe eine Rolle: nachdem Stefan Loh mit einem zweidimensionalen adaptiven Finite-Elemente-Mehrgitterverfahren erfolgreich war, lag der Gedanke nahe, die Methode prinzipiell beizubehalten, allerdings auf drei Dimensionen zu erweitern.

4.3.3 Adaptive Verfeinerung

4.3.3.1 Anforderungen an ein Verfeinerungsschema

Mit der beschriebenen Methode der finiten Elemente steht ein Werkzeug zur Verfügung, das in recht allgemeinen Fällen, insbesondere auch unregelmäßigen Gebietspartitionen, die lineare Differentialgleichung in ein lineares Gleichungssystem überführen kann, das darüberhinaus dünn besetzt (*sparse*) ist[13].
Die Kunst besteht in der geschickten Wahl der Simplexform, der Gebietspartition und der Basisfunktionen. Unsere Wahl der Simplexform als (unregelmäßige) Tetraeder ist dabei nur eine Möglichkeit, wenn auch sicher die am häufigsten verwendete. Unregelmäßige Gebietspartitionen werden in der Praxis häufig eingesetzt, um Randbedingungen zu behandeln, die auf einem kompliziert geformten Gebietsrand gefordert werden. Dies ist hier nicht der Fall; wir verlangen für die Differentialgleichung (4.6) lediglich die Dirichletsche Randbedingung im Unendlichen

$$\phi^a(\vec{x}) \to 0 \quad (|\vec{x}| \to \infty) \quad . \tag{4.35}$$

Die Vorteile der Finite-Elemente-Technik werden in unserem Fall deutlich, wenn wir die Gebietspartition unregelmäßig *machen*, um die Lösung der Differentialgleichung besser beschreiben zu können. Dies ist die Idee der *adaptiven Verfeinerung*: Verfeinere (zerteile) die Simplices dort, wo die Lösungsfunktion stärker schwankt, und lasse die Simplices grob, wo die Lösungsfunktion recht gleichmäßig ("langweilig") aussieht.
In diesem Abschnitt befassen wir uns mit dem Algorithmus, der die eigentliche Verfeinerung (Zerteilung) von Simplices vornimmt. Das benutzte Bisektionsverfahren ist während dieser Arbeit erst entstanden; seine Funktionstüchtigkeit ist in [58] für den allgemeinen d-dimensionalen Fall bewiesen. Wir beschränken uns hier auf eine reine Beschreibung ohne Beweise. Für die weiterführende Lektüre, insbesondere auch zu alternativen Verfeinerungsverfahren, sei der Leser auf [66] verwiesen.
Zunächst einmal stellt man an ein Schema, das zur Verfeinerung eines simplizialen Finite-Elemente-Gebietspartition eingesetzt werden soll, meist folgende Bedingungen aus Gründen der Praktikabilität und der numerischen Stabilität der resultierenden Gleichungen:

- Die Verfeinerung hat *verschachtelt* zu geschehen, d. h. Simplices, die nicht verfeinert werden, bleiben unverändert, und verfeinerte

[13] Große Matrizen — wir sprechen hier über Matrixgrößen von 10.000×10.000 bis 100.000×100.000 — sind in der Praxis nur dann überhaupt speicherbar und handhabbar, wenn sie dünn besetzt sind. Numerische Verfahren für dünn besetzte Matrizen bilden einen aktiven und umfangreichen Forschungszweig der numerischen Mathematik.

Simplices entstehen ausschließlich durch Zerschneiden vorhandener Simplices. Mit anderen Worten: man bewegt Knoten nicht, man fügt nur neue Knoten hinzu.

- Gebietspartitionen müssen *kompatibel* sein, d. h. zwei beliebige Simplices sind entweder disjunkt oder haben eine (vollständige) Seitenfläche, Kante oder Ecke gemeinsam. Dies impliziert beim Zerschneiden eines Simplexes, dass für alle Seitenflächen oder Kanten, die dabei verletzt (durchgeschnitten) werden, die betroffenen Nachbarsimplices auch geteilt werden müssen.

- Der kleinste innere Winkel zwischen Flächen muss gleichmäßig von unten beschränkt sein. Mit anderen Worten, während des Verfeinerungsprozesses dürfen Simplexecken nicht beliebig spitz und Kanten nicht beliebig scharf werden. Diese Forderung erwächst aus Stabilitätsuntersuchungen des resultierenden Finite-Elemente-Gleichungssystems und heißt daher *Stabilitätsbedingung*.

- Wirklich nützlich wird ein Verfeinerungsschema dann, wenn es *lokal (selektiv, adaptiv)* arbeiten kann: Wenn ein Simplex verfeinert wird, z. B. weil die Lösngsfunktion der Differentialgleichung in dieser Region stärker schwankt, so sollte diese Verfeinerung auf eine kleine Gruppe von Nachbarsimplices beschränkt bleiben: Die Zerteilung eines Simplexes darf nicht eine globale Verfeinerung aller Simplices notwendig machen. Diese Lokalitätsbedingung ist nur dann mit der obigen Kompatibilitätsbedingung vereinbar, wenn das Verfeinerungsschema nicht stets alle Seitenflächen eines Simplexes bei der Teilung verletzt. Solche Schnitte, die nicht alle Seitenflächen verletzen, dienen dazu, das feinere Simplexnetz vom äußeren, groben “abzuschirmen”.

Zwei Beispiele für Schemata, die alle o. g. Anforderungen erfüllen, sind in Abb. 4.10 grafisch dargestellt: das reguläre Schema in der oberen Zeile, das Bisektionsschema in der unteren Zeile. In jedem Verfeinerungsschritt wird jeweils der Simplex, der den markierten Punkt enthält, nach dem jeweiligen Schema zerteilt (im regulären Schema in vier Teile, im Bisektionsschema in zwei Teile), sowie in den Nachbarsimplices weitere Schnitte vorgenommen, die die Kompatibilität der Partition wiederherstellen. Der markierte Punkt steht hier symbolisch für eine Stelle, an der die Lösungsfunktion stark schwankt und dadurch eine Netzverfeinerung notwendig macht.
Man bemerke beim Vergleich: das reguläre Schema gelangt in weniger Schritten zu recht kleinen Simplexgrößen und wirkt damit “lokaler” als das Bisektionsschema. Auf der anderen Seite aber ist das reguläre Schema auch komplizierter, da sog. “Abschlussschnitte” (*closure cuts*, das sind die nichtregulär zweigeteilten Simplices, z. B. unten links in den bei-

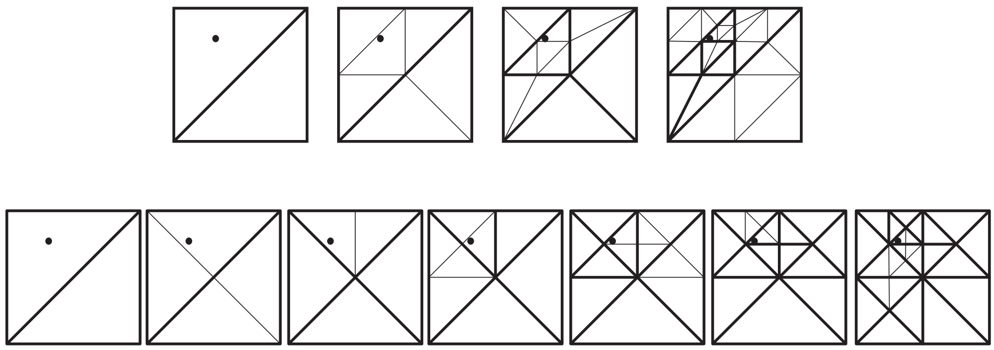


Abbildung 4.10: Zwei mögliche Verfeinerungsschemata für zweidimensionale Gebietspartitionen: oben das reguläre, unten das Bisektionsschema. Verfeinert wird in jedem Schritt lokal derjenige Simplex, der den markierten Punkt enthält, sowie diejenigen Nachbarsimplices, die nach dem jeweiligen Schema zerteilt werden müssen, um die Kompatibilität wiederherzustellen.

den mittleren Bildern des regulären Schemas zu sehen) bei der weiteren Verfeinerung erst wieder rückgängig gemacht werden müssen, um die Simplices dann regulär zu teilen. Im Gegensatz dazu kommt beim Bisektionsschema nur eine einzige Art von Teilung zum Einsatz: eben die Bisektion. Verfeinerungsschnitte und Abschlussschnitte sind formal identisch. Dies ist möglich, da bei der Zweiteilung eines Simplexes einige Kanten unverletzt bleiben.
Beide Schemata sind erfolgreich auf d Dimensionen verallgemeinert worden. Zum regulären Schema, das wir hier nicht verwenden, ist ein entsprechender Algorithmus im Artikel von Bey [66] zu finden. Der Bisektionsalgorithmus soll hier vorgestellt werden, wobei der Beweis seiner Funktionstüchtigkeit und der Erfüllung der obigen Anforderungen in [58] zu finden ist.

4.3.3.2 Bisektionsschema in d Dimensionen

Ein Simplex in d Dimensionen wird als geordnetes $(d+1)$-Tupel seiner Eckknoten (*nodes*) dargestellt,

$$S \;=\; (\vec{n}_0, \vec{n}_1, \ldots, \vec{n}_d)_g \quad . \tag{4.36}$$

Die Ordnung der Knoten ist dabei essentiell, da sie das Teilungsschema entscheidend beeinflusst. Die Position $0 \ldots d$ eines Knotens innerhalb des $(d+1)$-Tupels heißt *Knotenindex*. In der obigen Notation wurde noch ein *Generationsindex* g untergebracht. Wir starten mit Simplices der Generation 0, und wann immer ein Simplex der Generation g zerschnitten wird, sind die beiden Teilstücke dann Simplices der Generation $g+1$. Ein adaptiv verfeinertes Netz besteht dann aus Simplices verschiedenster Generationen. Das Generationskonzept lässt sich übrigens auf Kno-

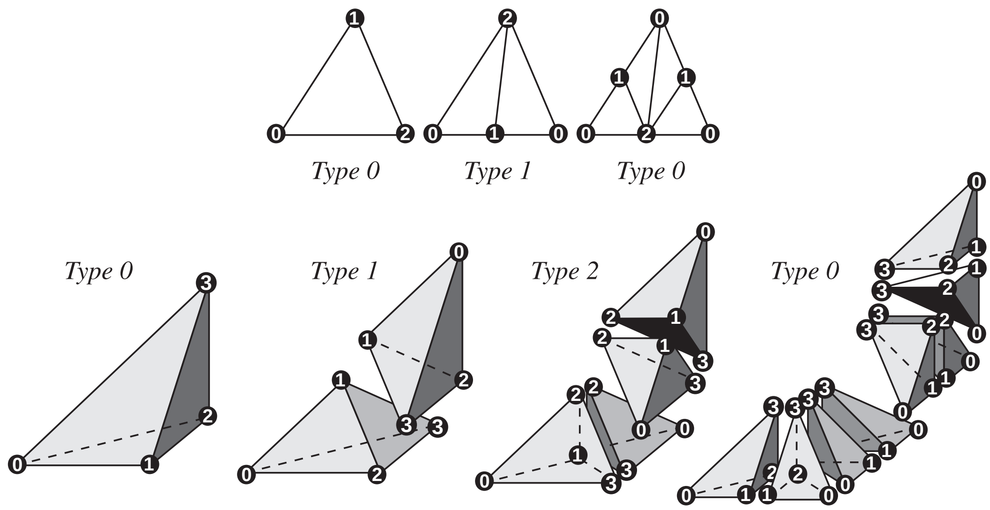


Abbildung 4.11: Bisektionsschema für die Fälle $d = 2$ (oben) und $d = 3$ (unten). Die Zahlen in schwarzen Kreisen zeigen die Knotenordnung der Simplices an: alle Simplices sind von der Form $(0, 1, 2)_g$ bzw. $(0, 1, 2, 3)_g$. Der Simplextyp ist definiert als Generation modulo d.

ten und Basisfunktionen erweitern: wenn Simplices der Generation g geteilt werden, entstehen neue Simplices mit neuen Knoten und auch neue Finite-Elemente-Basisfunktionen, die wir alle mit "Generation $g+1$" bezeichnen wollen.

Das Bisektionsschema wird durch folgende, extrem einfache Regel formuliert: Der oben eingeführte Simplex S der Generation g wird bei der Verfeinerung entlang der Hyperfläche $\{(\vec{n}_0 + \vec{n}_d)/2, \vec{n}_1 \ldots \vec{n}_{d-1}\}$ in zwei Teile zerschnitten. Diese Hyperfläche enthält alle Simplexkanten bis auf die Kante $\overline{\vec{n}_0\vec{n}_d}$, die folglich die einzige ist, die verletzt wird: sie wird in der Mitte durchgeschnitten, und der neue Knoten $\vec{N} = (\vec{n}_0 + \vec{n}_d)/2$ entsteht. Die geschnittene Kante nennen wir auch die *Simplexachse* (*axis, cut edge*). Es entstehen bei diesem Schnitt zwei *Kindsimplices* (*children simplices*) aus dem ursprünglichen *Elternsimplex* (*parent simplex*), der eine behält den Eckknoten $\vec{n}_0$, der andere $\vec{n}_d$. Alle anderen Knoten $\vec{n}_1 \ldots \vec{n}_{d-1}$ werden von beiden Kindsimplices als Eckknoten geerbt. Wir nennen die beiden "Kinder" daher C_0 und C_d. Führen wir nun noch den *Simplextyp* $\gamma := (g \bmod d) \in \{0 \ldots d-1\}$ ein, so können wir die beiden entstehenden "Kindsimplices" wie folgt notieren:

$$C_0 \;:=\; (\vec{n}_0, \vec{N}, \underbrace{\vec{n}_1 \ldots \vec{n}_\gamma}_{\longrightarrow}, \underbrace{\vec{n}_{\gamma+1} \ldots \vec{n}_{d-1}}_{\longrightarrow})_{g+1} \tag{4.37}$$

$$C_d \;:=\; (\vec{n}_d, \vec{N}, \underbrace{\vec{n}_1 \ldots \vec{n}_\gamma}_{\longrightarrow}, \underbrace{\vec{n}_{d-1} \ldots \vec{n}_{\gamma+1}}_{\longleftarrow})_{g+1} \quad . \tag{4.38}$$

Die Pfeile deuten die Richtung wachsender Indices an. Dabei ist die obige Zeile im Fall $\gamma = 0$ als

$$\begin{aligned} C_0 &:= (\vec{n}_0, \vec{N}, \vec{n}_1 \dots \vec{n}_{d-1})_{g+1} \\ C_d &:= (\vec{n}_d, \vec{N}, \vec{n}_{d-1} \dots \vec{n}_1)_{g+1} \quad , \end{aligned}$$

zu lesen, und im Falle $\gamma = n - 1$ als

$$\begin{aligned} C_0 &:= (\vec{n}_0, \vec{N}, \vec{n}_1 \dots \vec{n}_{d-1})_{g+1} \\ C_d &:= (\vec{n}_d, \vec{N}, \vec{n}_1 \dots \vec{n}_{d-1})_{g+1}. \end{aligned}$$

zu lesen. Die Knotenordnung ist der Schlüssel dieses Schemas, weil sie die zukünftigen Verfeinerungsschritte beeinflusst. Durch die beschriebene Knotenordnung ist garantiert, dass Simplices bei fortgesetzter Verfeinerung nie beliebig spitz werden (die Stabilitätsbedingung von Seite 86); diese Bedingung ist weitaus schwieriger zu erfüllen, als es auf den ersten Blick scheint. Damit ist das präsentierte Schema das eleganteste, weil einfachste, Verfeinerungsschema für d Dimensionen, das gegenwärtig auf dem Markt ist. Der beschriebene Algorithmus ist ein völlig neuer Zugang zur Bisektionsverfeinerung, der als Nebenprodukt meiner Arbeit entstanden ist und in [58] publiziert wurde; dort findet sich auch der strenge Beweis, dass das beschriebene Schema die Stabilitätsbedingung erfüllt. Der Beweis ist jedoch sehr technisch und passt nicht in den Rahmen dieser Arbeit.
Eine Illustration des Verfeinerungsschemas für die Fälle $d = 2$ und $d = 3$ ist in Abb. 4.11 gezeigt. Dort ist die Wirkung der invertierten Knotenordnung in (4.38) deutlich erkennbar: man achte im dreidimensionalen Fall (untere Zeile) auf den oberen Simplex vom Typ 1.

4.3.3.3 Gleichmäßige Partitionsverfeinerung

Ist der Simplex S Teil einer Gebietspartition, d. h. eingebettet in ein System von benachbarten Simplices, so müssen zusätzliche Schnitte an den Nachbarsimplices durchgeführt werden, damit die Kompatibilitätsbedingung nach der Bisektion des Simplex S wieder gilt.
Wir wollen zunächst den einfachen Fall der *gleichmäßigen Verfeinerung* betrachten: alle Simplices einer Generation werden gemeinsam verfeinert, so dass die verfeinerte Partition wieder aus Simplices der gleichen Generation besteht. In diesem Fall garantiert das Bisektionsschema die Kompatibilität der verfeinerten Partitionen bis in beliebig hohe Generationen, wenn nur die folgende Bedingung von der Ausgangspartition erfüllt wird: Die unverfeinerte Ausgangspartition (die Menge der Simplices von Generation 0) muss eine *reflektierte Gebietspartition* (*reflected domain partition*) sein. Was das bedeutet, wird nachfolgend erläutert.
Zunächst müssen wir drei verschiedene Begriffe eines "Nachbarsimplexes" einführen (siehe Abb. 4.12): Sei S ein beliebiger Simplex und T ein Nachbarsimplex, der mit S alle d Knoten außer $\vec{n}_k^S$ gemeinsam habe. T

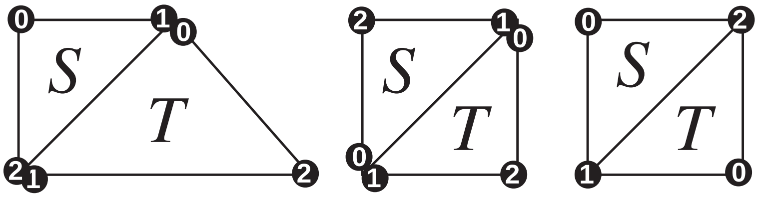


Abbildung 4.12: *Verschiedene Nachbarschaftsbegriffe. Im linken Bild ist T ein 0-Nachbar von S und S ein 2-Nachbar von T. Im mittleren Bild sind S und T einander zugewandte (facing) 2-Nachbarn. Im rechten Bild sind S und T reflektierte 2-Nachbarn. Die Ecknummern bezeichnen die Indices (Positionen), an denen die Knoten in den Simplices auftauchen, d. h., jeder Simplex ist von der Form $(0, 1, 2)_g$.*

liegt gewissermaßen gegenüber dem Eckknoten $\vec{n}_k^S$ an S an. Dann heiße T einfach der k-Nachbar von S. Simplices, die weniger als d Knoten gemeinsam haben, betrachten wir in diesem Kontext nicht als Nachbarn. Die Relation "ist k-Nachbar von" ist natürlich nicht reflexiv, d. h. S ist nicht notwendigerweise auch der k-Nachbar von T. Sind aber doch S und T jeweilige k-Nachbarn voneinander, so nennen wir die beiden *einander zugewandte k-Nachbarn* (*facing k-neighbours*). Im speziellen Fall, in dem zugewandte Nachbarn für sämtliche Knoten, die sie gemeinsam berühren, die gleichen Knotenindizes verwenden, heißen die beiden *reflektierte k-Nachbarn* (*reflected k-neighbours*). Abb. 4.12 zeigt Beispiele für die drei Nachbarschaftsbegriffe. Von links nach rechts werden die Relationen immer "stärker"; insbesondere sind reflektierte Nachbarn auch immer einander zugewandte Nachbarn.

Eine reflektierte Gebietspartition, wie sie als Ausgangspartition von unserem Schema gefordert wird, ist eine Partition, in der sämtliche Nachbarsimplices auch reflektierte Nachbarsimplices sind. Das bedeutet, jeder Knoten muss in jedem der Simplices, zu denen er gehört, den gleichen Index tragen (die gleiche Position im $(n + 1)$-Tupel besetzen). Das Schema stellt dann automatisch sicher, dass dies auch nach einer gleichmäßigen Verfeinerung für die Partitionen der Generationen d, $2d$, $3d$ wieder gilt. Beispielsweise sind alle Partitionen vom Typ 0 in Abb. 4.11 reflektierte Partitionen, nicht aber die dreidimensionale Typ-1-Partition. In [58] findet man eine vollständige Untersuchung für den allgemeinen d-dimensionalen Fall, welche Bedingung eine Gebietspartition erfüllen muss, damit man die Knoten einer Partition so ordnen kann, dass eine reflektierte Partition entsteht. Dort steht auch, wie die Partition ggf. präpariert werden muss, und natürlich werden alle unsere Aussagen auch dort bewiesen.

4.3.3.4 Adaptive (lokale) Partitionsverfeinerung

Die Vorteile der Partitionsverfeinerung kommen erst dann wirklich zum Tragen, wenn sie *adaptiv*, also lokal, erfolgt. Wir wollen nicht immer alle Simplices einer Partition gleichzeitig verfeinern, sondern nur einen ganz bestimmten Simplex S teilen. Dabei muss aber eine gewisse Menge der Nachbarsimplices auch geteilt werden, um die Kompatibilität der resultierenden Partition sicherzustellen.
Die Entscheidung, wann, wo und wie sehr lokal verfeinert wird, wird von der Präzisionssteuerung des Mehrgitterverfahrens getroffen; diese *high-level*-Routine wird im Abschnitt 4.3.4.5 erläutert. Hier soll nur die praktische Durchführung der lokalen Verfeinerung erklärt werden.
Die folgende Vorgehensweise garantiert dabei, dass nur die minimal notwendige Anzahl zusätzlicher Schnitte durchgeführt wird:

- Bevor der erwähnte Simplex S (von der Generation g) durch einen Bisektionsschnitt verfeinert werden kann, müssen alle Simplices, die mit ihm seine Achse A_S (die zu schneidende Kante $\overline{\vec{n}_0\vec{n}_d}$) gemeinsam haben, von der gleichen Generation g wie Simplex S sein. Es kann vorkommen, dass einzelne dieser Simplices von niedrigerer Generation sind. Dann müssen diese erst durch Verfeinerungsschritte (rekursiv!) soweit zerteilt werden, bis alle an A_S anliegenden Simplices der Generation g angehören.

- Wenn nun alle an A_S anliegenden Simplices von Generation g sind, so ist A_S auch gemeinsame Achse aller dieser Simplices. Dies wird durch das Bisektionsschema garantiert. Dann können mit der Bisektion von S gleichzeitig alle diese Simplices geteilt werden, und die resultierende Partition ist wieder kompatibel. Die Menge aller dieser in einem Schritt geteilten Simplices nennen wir eine *Verfeinerungszelle* (*refinement cell*).

Es ist unmöglich, dass an der Simplexachse von S anliegende Nachbarsimplices von höherer Generation sind. Das Bisektionsschema lässt diesen Fall gar nicht zu.
Das Bisektionsschema besitzt eine weitere nützliche Eigenschaft: In *allen* durch das Bisektionsschema gewonnenen verfeinerten Partitionen sind Nachbarsimplices stets einander zugewandt, aber nicht reflektiert [14]. Diesen Umstand nennen wir die *Strukturbedingung* (Beweis siehe wieder [58]). Sie ist oft nützlich, um im Programm bestimmte Nachbarsimplizes direkt zu lokalisieren, ohne einen Suchlauf über die vier Nachbarn starten zu müssen. Das Bisektionsschema nutzt die Strukturbedingung auch selbst vielfach aus. Der Leser mag sich an sämtlichen Partitionen der Abb. 4.11 davon überzeugen, dass die Strukturbedingung vom Bisektionsschema erfüllt wird.

[14] zu diesem Begriff siehe die Definitionen im letzten Abschnitt

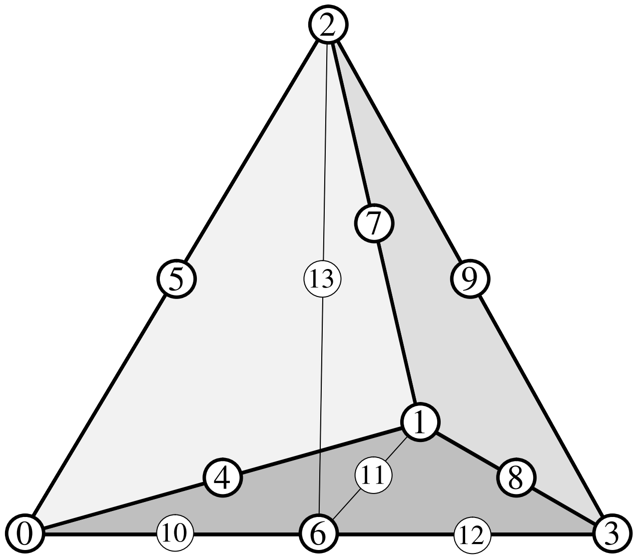


Abbildung 4.13: *Ein Simplex mit seinen Knoten und den neu hinzukommenden Knoten der beiden Kindsimplices. Wieder wird das Nummerierungsschema konsequent eingehalten.*

Damit endet der Überblick über das Bisektionsschema in d Dimensionen. Für Details sowie Beweise aller Aussagen sei der Leser nochmals auf [58] hingewiesen.

4.3.3.5 Simplexverfeinerung in MUESLEA

In MUESLEA werden die beiden Subsimplices C_0 und C_d (Nomenklatur aus Abschnitt 4.3.3.2) mit den Simplex-Klassenkomponenten positive und negative referenziert, weil sie auf der "positiven" bzw. "negativen" Seite der Schnittebene cut liegen. Die Subsimplices beziehen sich auf den ursprünglichen Simplex, der auch nach dem Schnitt weiterhin gespeichert bleibt, mit dem Zeiger parent zurück.
Das Nummerierungsschema der Simplexknoten (siehe Abb. 4.5) wird in MUESLEA aus praktischen Gründen auf die Knoten der nächsten Generation ausgedehnt. Statt also Ausdrücke wie simplex→negative→node(7) schreiben zu müssen, um den Kantenknoten 7 des Kindsimplexes negative zu referenzieren, kann einfach simplex→xnode(11) geschrieben werden. Die Methode Simplex::xnode steht hierbei für *extended node* und dient einfach als Abkürzung, um die Knoten der Kindsimplices einfach erreichen zu können. Das hierbei verwendete Schema ist in Abb. 4.13 abgebildet. Die Methode Simplex::xnode sowie viele verwandte Methoden sind in simplex.h zu finden.
In Abb. 4.14 ist eine Verfeinerungszelle in MUESLEA dargestellt. Der schattierte Simplex unten rechts ist der Ausgangssimplex S. Seine Achse, die Kante $\overline{\vec{n}_0\vec{n}_3}$, die beim Verfeinerungsschnitt verletzt wird, teilt er

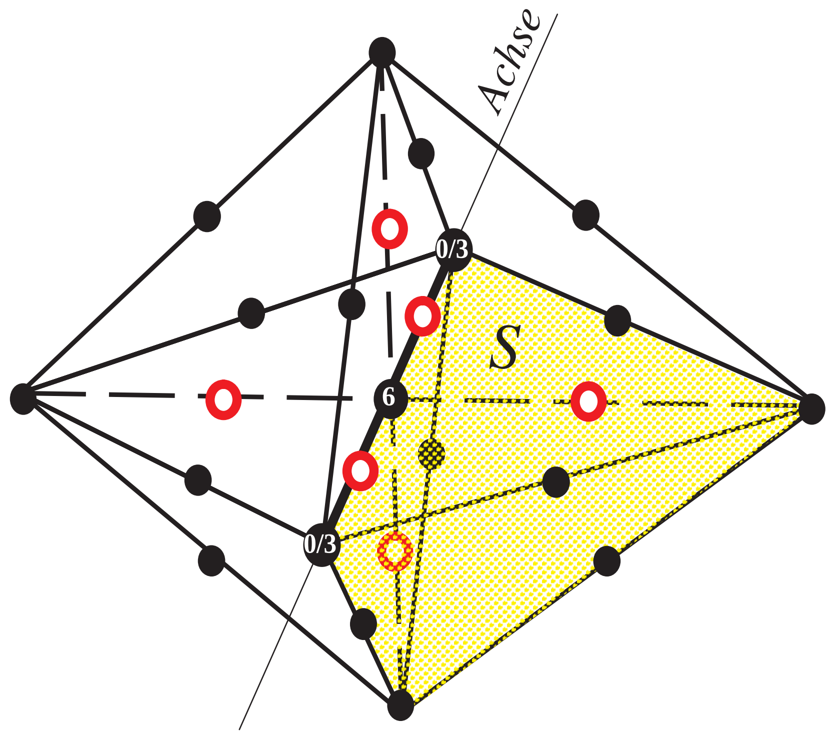


Abbildung 4.14: Dreidimensionale Verfeinerungszelle.

sich mit drei weiteren Simplices der gleichen Generation. Zusammen bilden die vier Simplices eine sog. Verfeinerungszelle, die alle die gleiche Achse haben, wenn auch die Bezeichnung der Knoten $\vec{n}_0$ und $\vec{n}_3$ in manchen Nachbarsimplices anders sein kann. Den Knoten $\vec{n}_6$ haben sie alle gemeinsam, und er hat auch in allen Simplices der Zelle den Knotenindex 6. Der Verfeinerungsschnitt erfolgt entlang der gestrichelten Linien. Die nach dem Schnitt neu hinzukommenden Kantenknoten sind als offene Kreise markiert.

Die Routinen Refine, Shred und TweakNeigh (Methoden der Klasse Simplex und definiert in simplex.cpp) übernehmen in einer verschränkten Rekursion die adaptive Simplexverfeinerung mit den notwendigen zusätzlichen Schnitten an Nachbarsimplices. Die Routine Simplex::Coarsify macht eine Verfeinerung rückgängig, wiederum unter Erhaltung der Partitionskompatibilität. Mit Domain::Chop steht eine Routine zur gleichmäßigen Verfeinerung zur Verfügung, und Domain::SetupDomain korrigiert unter anderem die anfängliche Knotennummerierung in den Simplices einer Partition so, dass die Partition zur reflektierten Partition wird. Alle diese Methoden sind definiert im File simplex.cpp.

4.3.3.6 Zäune im Simplexwald

Dieser Abschnitt geht auf ein eher technisches Detail in MUESLEA ein. Da bei einem Bisektionsschritt zwei Kindsimplices aus einem Elternsimplex entstehen, nimmt der Graph aller Subsimplices, die aus einem Ursimplex der Generation 0 entstehen, die Form eines *binären Baumes* an[15].

Wir starten aber mit der Verfeinerung nicht von einem einzigen Simplex, sondern von einer Partition[16] des würfelförmigen Gebiets Ω, die aus sechs Kuhnsimplices besteht (siehe Abb. 4.6). Daher entsteht nach einigen Verfeinerungsschritten ein *Simplexwald* aus genau sechs binären Simplexbäumen. Die adaptiven Simplexwälder, die bei den in Kapitel 5 vorgestellten Rechnungen entstehen, besitzen bis zu etwa 10^5 Simplices, organisiert in bis zu etwa 25 Verfeinerungsstufen.

Ein solcher Simplexwald aus vier Generationen ist in Abb. 4.15 schematisch gezeichnet. Der Kind-Eltern-Graph (Zeiger positive, negative, parent) ist dargestellt, nicht gezeigt ist der Nachbarschaftsgraph (Zeigerfeld Simplex::neigh[]). Nicht alle Simplices sind verfeinert (Hinweis “no kids”). Die Bedeutung der Schattierung in der Abbildung ist das Thema dieses Abschnittes; mehr dazu im nächsten Absatz.

In unserem Fall müssen zwei partielle Differentialgleichungen gelöst werden, da in (4.6) eigentlich zwei skalare Gleichungen für die beiden Felder $\phi^{3,8}$ kompakt notiert sind. Zur numerischen Lösung müssten also zwei separate Simplexwälder aufgebaut werden, was einigen rechnerischen Aufwand und Speicherplatz kostet. Um diesen Aufwand zu umgehen, werden *fences*, Zäune, im Simplexwald eingeführt. Diese Zäune sind ein einfacher Trick, um mit einem einzigen Simplexwald mehrere unabhängige virtuelle Simplexwälder zu “simulieren”. Dazu wird an jedem Simplex für jedes Feld ein Flag angebracht, das wir Fence (Zaun) nennen. Die Namen dieser Flags sind in MUESLEA an die zugehörigen Feldnamen angelehnt, z. B. PHI3FENCE, PHI8FENCE[17]. Ein Fence f ist dann gesetzt (true), wenn der Simplex, betrachtet als Teil des zum Fence f gehörigen virtuellen Simplexwaldes, nicht weiter verfeinert ist. Beispielsweise wird in einem Simplex der Fence PHI8FENCE gesetzt, wenn er nicht “in Beziehung auf das Feld ϕ^8” geteilt wurde (also im virtuellen Simplexbaum dieses Feldes atomar ist). Sollte dieser Simplex trotzdem auf Kindsimplices verweisen, z. B. weil diese zur Speicherung anderer Felder benötigt werden, so werden die Kindsimplices bei der Bearbeitung des ϕ^8-Feldes ignoriert. In Abb. 4.15 ist dies durch die Schattierung verdeutlicht: falls die Fence-Flags der grau unterlegten Simplices gesetzt sind und die Fences der Elternsimplices nicht gesetzt sind, so wird

[15] Die Elternsimplices werden dabei nicht gelöscht.

[16] in MUESLEA organisiert durch die Klassenhierarchie List⟨Simplex⟩ < Domain < Cube.

[17] Definitionen siehe simplex.h

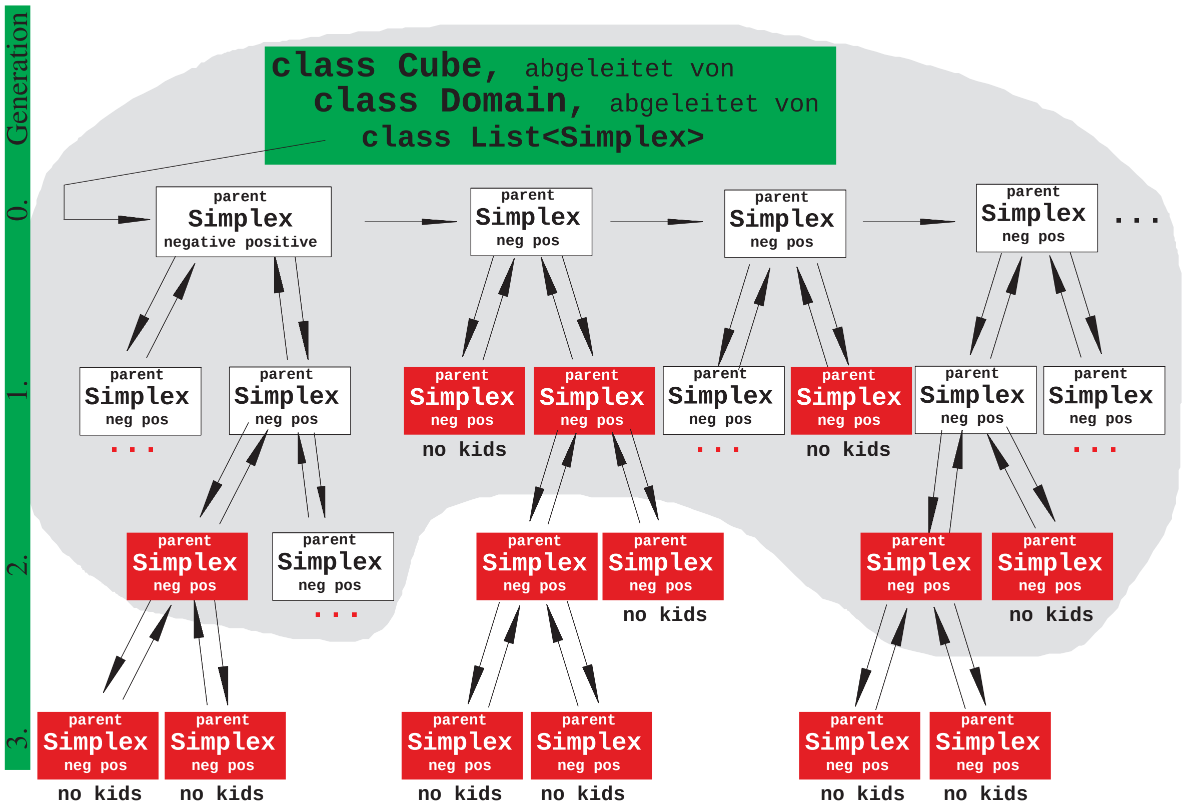


Abbildung 4.15: *Der Simplexwald. Der grau schattierte Bereich deutet eine Sub-Wald-Struktur an, die mit Zäunen (Fences) erzeugt wird. Dabei sind in den rot (dunkel) schattierten Simplices alle Fences gesetzt (Erläuterungen dazu siehe Text). Der Nachbarschaftsgraph sowie die von den Simplices referenzierten Knoten, die auch Rückreferenzen auf ihre erzeugenden Simplices enthalten, sind hier aus Gründen der Übersichtlichkeit nicht dargestellt; diese Strukturen sind dem Graphen noch überlagert.*

der schattierte Teilwald als virtueller Simplexwald benutzt. Die Fences der Subsimplices darunter sind irrelevant, in MUESLEA aber stets auch gesetzt (Konvention).
Für den praktischen Umgang mit Fences und dem Simplexwald im allgemeinen ist folgendes wichtig zu wissen:

- Zu jedem auf Simplexknoten gespeicherten Feld gehört genau ein Fence. Zum Beispiel gehört zu ϕ^3 eben PHI3FENCE und zu ϕ^8 entsprechend PHI8FENCE.

- Sowie eine Methode aufgerufen wird, die Simplices verfeinert, Verfeinerungen zurücknimmt, das Vorhandensein von Kindsimplices testet oder sonst irgendwie im Simplexwald umherwandert, so muss dieser Methode ein Fence übergeben werden, der regelt, auf welchen virtuellen Simplexwald sich die Operation bezieht.

- Die MUESLEA[18] -Routinen sorgen dafür, dass Simplices physikalisch erzeugt werden, sobald sie von einem der virtuellen Wälder benötigt werden. Ebenso werden sie wieder gelöscht, sobald sie überflüssig geworden sind. Es existiert eine Routine (Simplex::CheckStructure, definiert in tools.cpp), die die Konsistenz der Gebietspartition ausführlich testet und im Programm immer wieder aufgerufen wird. Zur Aufdeckung von Speicherlecks dient das Watchdog-Klassentemplate (counters.h), das die Erzeugung und Vernichtung u. a. von Simplex-Instanzen überwacht (Instanzen mitzählt).

- Knoten werden ebenfalls von den Routinen automatisch erzeugt und vernichtet. Eine *garbage collection* zur Vernichtung nicht mehr benötigter Knoten wird übrigens trotz des komplizierten Verfeinerungsgraphen nicht eingesetzt: das Klassentemplate ReferenceCounter (counters.h) erzeugt automatisch die entsprechenden Destruktoraufrufe, sobald ein Knoten von keinem Simplex mehr referenziert wird.

- "Von Hand", also unter Umgehung der vorgesehenen Routinen, sollten niemals Fences gesetzt oder gelöscht oder gar Simplices oder Knoten erzeugt oder vernichtet werden. Einzige Ausnahme dieser Regel ist die Konstruktion der anfänglichen, groben Partition Ω, die aus einer Liste von Generation-0-Simplices besteht, in denen alle Fences gesetzt sind. Falls man mit einer anderen als der in Klasse Cube vorgesehenen würfelförmigen Partition (Abb. 4.6) arbeiten möchte, so kann man die korrekte Erzeugung einer solchen Partition im Cube-Konstruktor (cube.cpp) nachvollziehen.

[18]Zur Namensgebung siehe Fußnote auf Seite 80.

4.3.4 Hierarchisches Mehrgitterverfahren

4.3.4.1 Problem und Lösungsidee

In Abschnitt 4.3.2 wurde die Diskretisierungmethode der finiten Elemente für eine beliebige simpliziale Gebietspartition vorgestellt. Der darauffolgende Abschnitt 4.3.3 hatte ein Schema zur adaptiven Verfeinerung der Partition zum Thema. Es fehlt nun noch ein Baustein, um aus diesen Ideen einen Differentialgleichungslöser zu machen:

- Wir haben bisher nicht erklärt, wie das dünn besetzte lineare Gleichungssystem (4.24) eigentlich zu lösen ist.

- Auch ist noch nicht bekannt, wie wir damit umgehen, wenn während der Lösung klar wird, dass die Partition verfeinert werden muss (Präzisionssteuerung siehe Abschnitt 4.3.4.5); als Konsequenz der Verfeinerung werden ja Knoten und Basisfunktionen hinzugefügt, was das Gleichungssystem (4.24) eigentlich völlig verändern sollte.

Die Antwort auf *beide* Punkte gibt das *hierarchische Mehrgitterverfahren*[19] (*hierarchical multigrid algorithm*) auf eine äußerst elegante Weise. In den folgenden Abschnitten soll eine verständliche Einführung in das Verfahren gegeben werden, wobei wir uns auf einen pragmatischen Blickwinkel beschränken wollen, um den "roten Faden" der Dissertation (Simulation eines physikalischen Modells) nicht aus den Augen zu verlieren. Insbesondere wird die mathematische Konvergenztheorie der Mehrgitterverfahren nicht erläutert; hierzu sei auf die Referenzen [62, 63, 64, 67] und die darin gegebenen Literaturhinweise verwiesen.

Die Mehrgitterverfahren bilden mittlerweile einen eigenen Forschungszweig der numerischen Mathematik. Es gibt Mehrgitterverfahren auch ohne zugrundeliegendes Finite-Elemente-Schema und ohne adaptive Verfeinerung. Das Buch *Numerical Recipes* [60] enthält auf Seite 871ff eine leicht verständliche Einführung in Mehrgitterverfahren für Finite-Differenzen-Schemata ohne adaptive Verfeinerung. Die anschließenden Literaturhinweise können als Startpunkt zur Vertiefung dienen.

Nach diesen Vorbemerkungen nun zum hierarchischen Mehrgitterverfahren. Wir betrachten zur Einführung das eindimensionale Dirichlet-

[19] Um mit der Literatur konform zu bleiben, verwenden wir bei der Erläuterung des Mehrgitterverfahrens das Wort "Gitter" synonym mit "Netz" und "Partition". Mit "Gitter" ist also keinesfalls ein regelmäßiges, kartesisches Gitter gemeint, sondern das bekannte adaptive Simplexnetz. In MUESLEA verwenden wir die Klassen Domain, Simplex und Node, um das adaptive Simplexnetz zu beschreiben, auf der anderen Seite beschreiben die Klassen Grid und Site das starre kartesische Gitter des σ-Feldes.

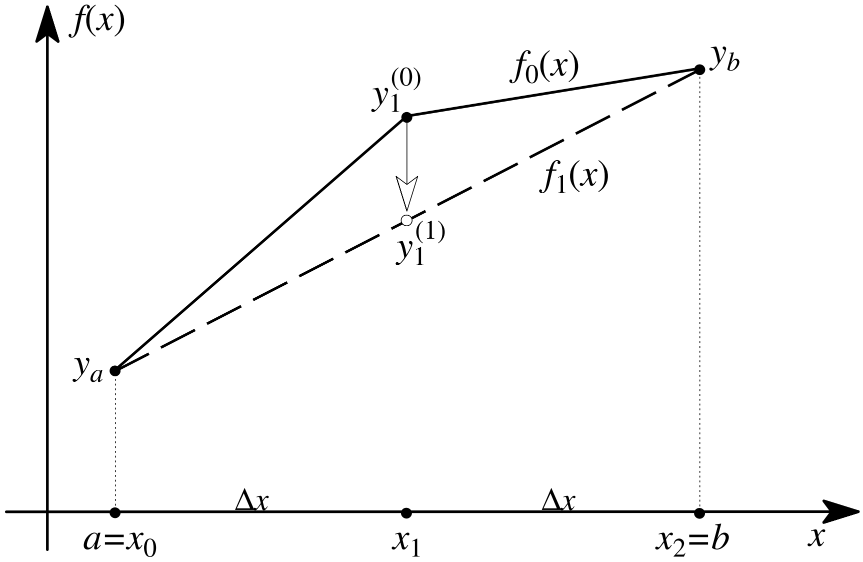


Abbildung 4.16: *Grobe Diskretisierung des Randwertproblems* (4.39). *Die Punkte auf der* x*-Achse markieren die Knoten.*

sche Randwertproblem

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= 0 \\ f(a) &= y_a \\ f(b) &= y_b \quad , \end{aligned} \tag{4.39}$$

das die triviale Lösung

$$f(x) = \frac{(x-a)y_b + (b-x)y_a}{b-a} \tag{4.40}$$

besitzt. Obwohl wir die Lösung (eine gerade Linie) bereits kennen, versuchen wir sie nun mit *Gauß-Seidel-Iterationen* aus einer Näherungslösung zu finden. Nehmen wir zunächst an, das Problem sei auf einem regelmäßigen, groben Gitter, das aus nur drei Punkten $\{a = x_0, x_1, x_2 = b\}$ besteht, diskretisiert (siehe Abb. 4.16). Die Funktionswerte an den Randknoten sind durch die Randbedingungen fixiert. Es gibt nur einen inneren Knoten. Der diskretisierte Differentialoperator lautet

$$\frac{\partial^2 f(x_k)}{\partial x^2} = \frac{f(x_{k+1}) - 2f(x_k) + f(x_{k-1})}{\Delta x^2} + \mathcal{O}(\Delta x^2) \quad . \tag{4.41}$$

Das diskretisierte Randwertproblem lautet folglich

$$\frac{f(x_{k+1}) - 2f(x_k) + f(x_{k-1})}{\Delta x^2} = 0 \tag{4.42}$$

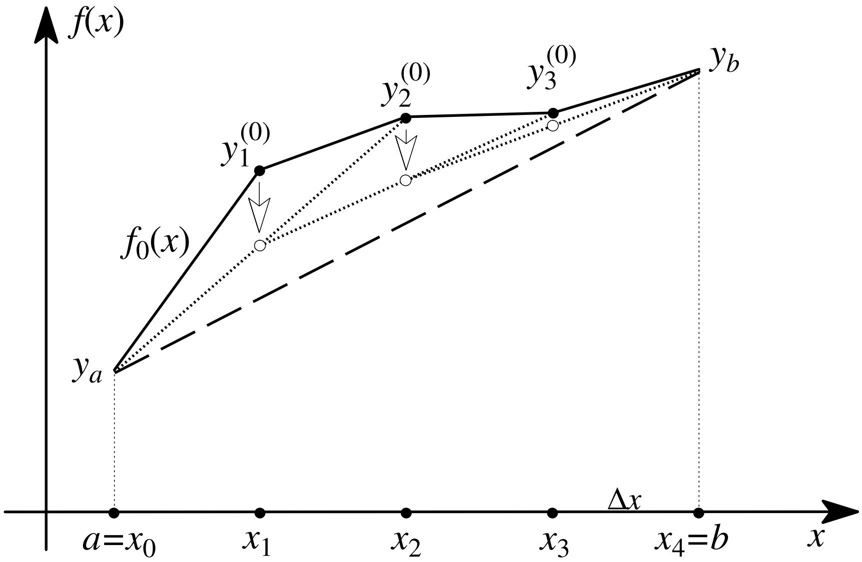


Abbildung 4.17: Feinere Diskretisierung des Randwertproblems (4.39).

mit den bekannten Randwertbedingungen. Sei $f_0(x)$ eine Näherungslösung (Abb. 4.16), die mit $f_0(x_1) = y_1^{(0)}$ nicht der exakten Lösung (gerade Linie) entspricht. Dann ergibt eine einzige Gauß-Seidel-Iteration, die einfach (4.42) invertiert

$$f_{n+1}(x_k) = \frac{1}{2}\left(f_n(x_{k+1}) + f_n(x_{k-1})\right) \quad , \tag{4.43}$$

bereits die exakte Lösung (betrachte $n = 0$ und $k = 1$).

Nun ist die Lösung von (4.39) eine lineare Funktion und daher auf einem beliebig groben Gitter exakt darzustellen. Würden wir eine Quellfunktion auf der rechten Seite der Differentialgleichung (4.39) hinzufügen, so könnte die Lösungsfunktion völlig anders aussehen; ein grobes Gitter würde zur Darstellung nicht mehr ausreichen. Betrachten wir daher aus rein akademischer Motivation das gleiche Problem, wobei wir ein feineres Gitter zu Hilfe nehmen. In Abb. 4.17 ist die anfängliche Näherungslösung wieder als durchgezogener Graph gezeichnet. In jedem Gauß-Seidel-Iterationsschritt wird (4.43) nacheinander für jeden Knoten durchgeführt, wobei immer die jeweils neuesten Knotenwerte eingehen[20]. Wir bewegen also im ersten Schritt den Wert $y_1^{(0)}$, im zweiten Schritt $y_2^{(0)}$ usw. In Abb. 4.17 sind drei Schritte gezeigt. Man erkennt aus dem Bild bereits:

[20]Falls die Knotenwerte *gleichzeitig* (also nicht nacheinander unter Verwendung der jeweils neuesten Knotenwerte) per (4.43) bewegt werden, heißen die Schritte *Jacobi-Iterationen*. Dieses Verfahren ist aber sehr langsam und wirklich nur von akademischen Interesse [60].

- Nach drei Gauß-Seidel-Schritten ist die Näherungslösung auf dem feinen Gitter immer noch weit von der exakten Lösung (gestrichelte Linie) entfernt.
- Die Abweichung von der exakten Lösung ist eine *langwellige Störung*, zu deren Korrektur das grobe Gitter ideal wäre.
- Die anfängliche Abweichung vom exakten Ergebnis war eine Überlagerung aus kurz- und langwelliger Störung. Die *kurzwellige Störung* wurde durch die Gauß-Seidel-Iterationen auf dem feinen Gitter schnell weggedämpft. Das grobe Gitter wäre nicht in der Lage, die kurzwellige Störung überhaupt darzustellen, geschweige denn abzudämpfen.

Wir formulieren diese Beobachtung etwas allgemeiner:

- Langwellige Störungen werden schnell auf einem groben Gitter abgedämpft. Noch dazu ist der Rechenaufwand hier wegen der geringen Knotenzahl sehr niedrig.
- Kurzwellige Störungen werden schnell auf einem feinen Gitter abgedämpft. Das feine Gitter ist überdies in der Lage, die Lösungsfunktion präziser zu approximieren.

Ein Mehrgitterverfahren kombiniert die Vorteile grober und feiner Gitter. Dazu werden mehrere Gitter verschiedenster Feinheit eingesetzt. Das feinste Gitter soll schließlich die Lösungsfunktion darstellen, während die gröberen Gitter zum schnellen Abdämpfen langwelliger Störungen dienen. Um die Rechenergebnisse der groben Gitter auf eine für das feine Gitter akzeptable Genauigkeit zu bringen, werden Korrekturterme für die gröberen Gitter auf dem feinsten Gitter berechnet. Im folgenden wird diese Idee konkretisiert.

4.3.4.2 Lösung der Gleichungen auf dem groben Gitter

Das grobe Gitter[21] besteht in unserem Fall aus der würfelförmigen Partition mit sechs Simplices (Abb. 4.6). Da nach dem Knotenschema Abb. 4.5 an jeder Kantenmitte und jeder Ecke der Simplices Knoten angebracht sind, befinden sich die Knoten in der groben Partition nach Abb. 4.6 in einem $3 \times 3 \times 3$-Gitter. Es sind also 27 Knoten $\{\vec{n}_k\}$ im groben Gitter enthalten. Die Gleichungen (4.24),

$$\sum_j D_{ij}\phi_j^a \;=\; r_i^a \tag{4.44}$$

[21] Es soll nochmals darauf hingewiesen werden, dass wir bei den Erläuterungen zum Mehrgitterverfahren das Wort “Gitter” synonym zu “Netz” und “Partition” verwenden. Mit “Gitter” ist also keinesfalls ein regelmäßiges, kartesisches Gitter gemeint (MUESLEA-Klassen Grid, Site), sondern eine unregelmäßige, adaptive Struktur (MUESLEA-Klassen Domain, Simplex, Node), die im folgenden erklärt werden wird.

mit den Definitionen (4.45) und (4.46),

$$D_{ij} = \int \kappa(\sigma(\vec{x})) \left(\vec{\nabla}\eta_i(\vec{x})\right) \cdot \left(\vec{\nabla}\eta_j(\vec{x})\right) d^3x \tag{4.45}$$

$$r_i^a = \eta_i(\vec{x}) \int \rho^a(\vec{x}) d^3x \quad , \tag{4.46}$$

werden auf diesem groben Gitter praktikabel und numerisch exakt mit Hilfe einer *LR-Zerlegung* gelöst [60]. Wegen der geringen Knotenzahl ist es hier noch möglich, die Struktur der D_{ij}-Matrix zu ignorieren und ein so allgemeines Lösungsverfahren zu verwenden.
Bei der Diskretisierung werden die von (4.32) her bekannten quadratischen Ansatzfunktionen $\{\eta_i\}$ der groben Partition benutzt. Die Funktion $\kappa(\sigma(\vec{x}))$ geht hier ebenfalls nur auf den 27 Grobgitterknoten $\{\vec{n}_k\}$ ein und wird dazwischen durch finite Elemente interpoliert. Statt $\kappa(\sigma(\vec{x}))$ wird also

$$\kappa^L := \sum_i \kappa(\sigma(\vec{x}_i))\eta_i^L(\vec{x}) \tag{4.47}$$

benutzt. Als Ansatzfunktionen $\eta_i^L(\vec{x})$ könnten die von (4.32) her bekannten stückweise quadratischen Funktionen mit $\eta_i(\vec{n}_j) = \delta_{ij}$ verwendet werden; wir werden aber im nächsten Abschnitt sehen, dass dies zu Problemen führt. An dieser Stelle soll der Hinweis genügen, dass die $\eta_i^L(\vec{x})$ stückweise *linear* sind, aber die linearen Bereiche sogar kleiner als Simplices sind, so dass wie für die quadratischen Ansatzfunktionen aus (4.32) gilt

$$\eta_i^L(\vec{n}_j) = \delta_{ij} \quad , \tag{4.48}$$

aber die "Überschwinger" in den Bereich negativer Funktionswerte (siehe Abb. 4.8) eliminiert sind.

4.3.4.3 Einschub: Positivität des diskretisierten Differentialoperators

Fast alle Lösungsmethoden dünn besetzter linearer Gleichungssysteme

$$Df = r \tag{4.49}$$

verlangen die Positivität der zu invertierenden Matrix D. Dies gilt auch für die Gauß-Seidel-Iterationen, die wir im Mehrgitterverfahren nutzen werden: damit sie konvergieren, müssen die zu invertierenden Matrizen alle positiv definit sein[22].

[22]Diese Bedingung hängt damit zusammen, dass die Gauß-Seidel-Iterationen das Funktional

$$S[f] := \langle f \,|\, D \,|\, f \rangle - \langle f \,|\, r \rangle$$

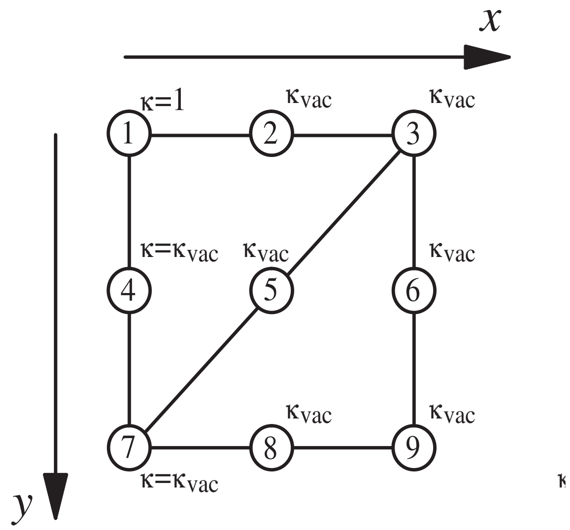


Abbildung 4.18: Das Simplexnetz $\mathcal{T}$ mit zwei Simplices bildet eine Partition des Einheitsquadrats. Die Knoten $\{\vec{n}_1 \ldots \vec{n}_9\}$ sind durch Kreise gekennzeichnet.

Verwendet man für den diskretisierten Differentialoperator (4.45) die stückweise quadratischen Ansatzfunktionen η_k aus (4.32) sowohl zur Entwicklung der Lösungsfunktion ϕ (in (4.45) bereits eingesetzt) als auch für die dielektrische Funktion κ,

$$\kappa(\vec{x}) \;=\; \sum_k \kappa_k \eta_k(\vec{x}) \quad , \tag{4.50}$$

so kann der Fall eintreten, dass bereits die Grobgittermatrix

$$\begin{aligned} D_{ij} &= \int \kappa(\sigma(\vec{x})) \left(\vec{\nabla}\eta_i(\vec{x})\right) \cdot \left(\vec{\nabla}\eta_j(\vec{x})\right) d^3x \\ &= \sum_k \kappa_k \int \eta_k(\vec{x}) \left(\vec{\nabla}\eta_i(\vec{x})\right) \cdot \left(\vec{\nabla}\eta_j(\vec{x})\right) d^3x \end{aligned} \tag{4.51}$$

nicht positiv definit ist. Betrachten wir zum Beispiel das zweidimensionale Simplexnetz $\mathcal{T}$ aus Abb. 4.18. Die Funktion $\kappa(\vec{x})$ nimmt den Wert 1 am Knoten $\vec{n}_1$ an, und den Wert κ_{vac} an allen anderen Knoten. Approximieren wir $\kappa(\vec{x})$ von diesen gegebenen Knotenwerten, so erhalten wir nach (4.50)

$$\begin{aligned} \kappa^A(\vec{x}) &:= \sum_k \kappa_k \eta_k(\vec{x}) \\ &= \kappa_{\mathrm{vac}} + (1-\kappa_{\mathrm{vac}})\eta_1(\vec{x}) \quad . \end{aligned} \tag{4.52}$$

zu minimieren versuchen. Die Lösung f erfüllt dann auch (4.49). Sollte D nicht positiv definit sein, so hat $S[f]$ aber kein Minimum, sondern einen Sattelpunkt, wodurch das Iterationsverfahren divergiert. Zur Vertiefung siehe [60] und die darin enthaltenen Referenzen.

Die Dirichletschen Randbedingungen legen die Lösung auf dem Gebietsrand unmittelbar fest, so dass die Randknoten nicht als Unbekannte in das Gleichungssystem eingehen. Hier sind alle Knoten außer $\vec{n}_5$ auf dem Gebietsrand lokalisiert, so dass der diskretisierte Operator (4.51) eine triviale "1×1-Matrix" bildet. Wir erhalten den einzigen Eintrag dieser Matrix als

$$\begin{aligned} D_{55} &= \int \kappa^A(\vec{x}) \left(\vec{\nabla}\eta_5(\vec{x})\right) \cdot \left(\vec{\nabla}\eta_5(\vec{x})\right) d^3x \\ &= \sum_k \kappa_k \int \eta_k(\vec{x}) \left(\vec{\nabla}\eta_5(\vec{x})\right)^2 d^3x \\ &= \int (\kappa_{\text{vac}} + (1-\kappa_{\text{vac}})\eta_1(\vec{x})) \left(\vec{\nabla}\eta_5(\vec{x})\right)^2 d^3x \end{aligned} \tag{4.53}$$

Das Integral lässt sich schnell auswerten. Es genügt, über den oberen linken Simplex zu integrieren, wo $\eta_1(\vec{x}) = (1-x-y)(2(1-x-y)-1)$ und $\eta_5(\vec{x}) = 4xy$, und das Resultat lautet

$$\begin{aligned} &= \int (\kappa_{\text{vac}} + (1-\kappa_{\text{vac}})(1-x-y)(2(1-x-y)-1))16(y^2+x^2)dx \\ &= \frac{8}{45}(16\kappa_{\text{vac}} - 1) \quad . \end{aligned}$$

Dieser lineare Operator ist nicht positiv definit, sobald $\kappa_{\text{vac}} < 1/16$. Man bemerke jedoch in der ersten Gleichung in (4.53), dass $D_{55} > 0$ gälte, wenn die quadratisch approximierte Funktion $\kappa^A(\vec{x})$ durch die exakte $\kappa(\vec{x})$ ersetzt würde, da diese ja positiv definit ist. Also bewirkt erst die Approximation (4.52) erst das Problem. Die Ursache sind die "Überschwinger" der Basisfunktionen (siehe Abb. 4.8) ins Negative.

Natürlich wird der diskretisierte Operator positiv definit sein, sobald das *approximierte* κ^A es ist. Dies ist sicherlich der Fall, wenn eine der folgenden Bedingungen erfüllt ist:

(i) Stückweise lineare Basisfunktionen werden zur Bildung der Approximation $\kappa^A(\vec{x})$ benutzt. Die lineare Approximation lässt interpolierte Werte grundsätzlich *zwischen* den Knotenwerten liegen. Da die exakte Funktion $\kappa(\vec{x})$ überall positiv definit ist, ist es dann auch die Approximation $\kappa^A(\vec{x})$.

(ii) Das Simplexnetz ist fein genug, um die Approximation $\kappa^A(\vec{x})$ so gut werden zu lassen, dass sie positiv definit ist wie $\kappa(\vec{x})$. Da $\kappa(\vec{x})$ stetig ist, konvergiert die angenäherte Funktion $\kappa^A(\vec{x})$ mit feiner werdendem Simplexnetz gleichmäßig zu $\kappa(\vec{x})$ hin.

(iii) $\kappa(\vec{x})$ wird immer — auch auf dem groben Gitter — exakt behandelt.

Bei Mehrgittercodes ist *(ii)* in der Regel nicht erfüllt, da es in der Natur dieser Methoden liegt, auch sehr grobe Gitter einzusetzen (zur Konvergenzbeschleunigung auf den feinen Gittern). Obwohl *(iii)* manchmal ein

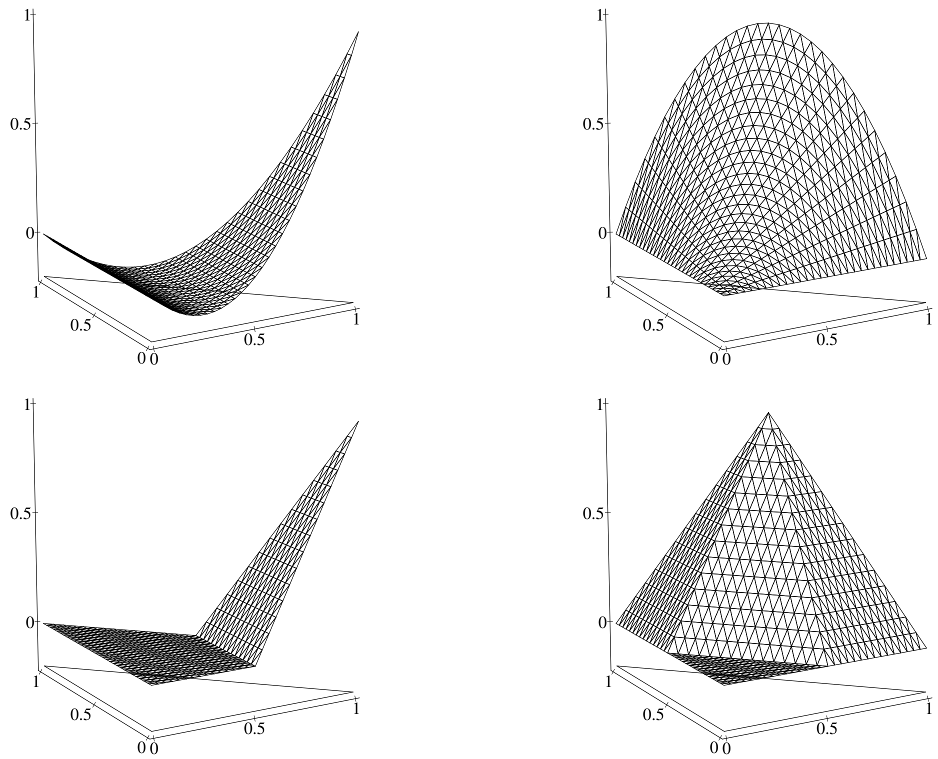
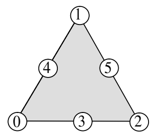
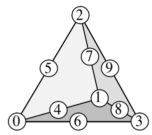


Abbildung 4.19: *Die üblichen quadratischen Basisfunktionen η_k für Eck- und Kantenträgerknoten (oben), verglichen mit den stückweise linearen η_k^L (unten) im zweidimensionalen Simplex. Die horizontalen Achsen markieren die Simplexkoordination α_0 and α_1. Man bemerke, dass in jeder Spalte die beiden Basisfunktionen an allen sechs Knoten (Ecken und Kantenmitten des Simplexes) übereinstimmen. Im dreidimensionalen Fall mitteln wir über die stückweise linearen Funktionen, die aus drei verschiedenen Möglichkeiten der regulären Simplexteilung resultieren.*

Tabelle 4.1: *Matrixelemente, die die üblichen linearen und quadratischen Basisfunktionen sowie die neuen “linearisierten” Basisfunktionen involvieren. Alle Integralwerte sind für ein Simplexvolumen* 1 *notiert und müssen für andere Volumina entsprechend skaliert werden. Alle nicht aufgeführten Matrixelemente lassen sich durch Symmetriebetrachtungen erschließen. Die Benennungsschemata der Simplexknoten sind in der Kopfzeile der Tabelle skizziert. Das dreidimensionale Schema ist bereits von Abb. 4.5 her bekannt; das zweidimensionale Schema ist so gewählt, dass die Tabelle möglichst einfach wird.*

						
Knoten	Eckknoten	Eckknoten	2-dim Fall		3-dim Fall	
$\vec{n}_k$	$\vec{n}_l$	$\vec{n}_m$	$\int \eta_k \alpha_l \alpha_m$	$\int \eta_k^L \alpha_l \alpha_m$	$\int \eta_k \alpha_l \alpha_m$	$\int \eta_k^L \alpha_l \alpha_m$
0	0	0	$\frac{1}{30}$	$\frac{23}{480}$	$\frac{1}{140}$	$\frac{1}{64}$
0	0	1	0	$\frac{7}{960}$	$-\frac{1}{420}$	$\frac{1}{480}$
0	1	1	$-\frac{1}{90}$	$\frac{1}{480}$	$-\frac{1}{140}$	$\frac{1}{1920}$
0	1	2	$-\frac{1}{180}$	$\frac{1}{960}$	$-\frac{1}{280}$	$\frac{1}{3840}$
4	0	0	$\frac{1}{15}$	$\frac{5}{96}$	$\frac{1}{35}$	$\frac{7}{320}$
4	0	1	$\frac{2}{45}$	$\frac{7}{192}$	$\frac{2}{105}$	$\frac{17}{1152}$
4	0	2	$\frac{1}{45}$	$\frac{1}{64}$	$\frac{1}{105}$	$\frac{77}{11520}$
4	2	2	$\frac{1}{45}$	$\frac{1}{96}$	$\frac{1}{105}$	$\frac{11}{1920}$
4	2	3	entfällt	entfällt	$\frac{1}{210}$	$\frac{11}{2880}$
Knoten n_k	Knoten n_l		$\int \eta_k \eta_l$	$\int \eta_k^L \eta_l$	$\int \eta_k \eta_l$	$\int \eta_k^L \eta_l$
0	0		$\frac{1}{30}$	$\frac{1}{30}$	$\frac{1}{70}$	$\frac{3}{320}$
0	1		$-\frac{1}{180}$	$-\frac{1}{160}$	$\frac{1}{420}$	$-\frac{1}{480}$
0	4		0	$\frac{7}{240}$	$-\frac{1}{105}$	$\frac{1}{120}$
0	5		$-\frac{1}{45}$	$\frac{1}{240}$	$-\frac{1}{105}$	$\frac{1}{120}$
0	9		entfällt	entfällt	$-\frac{1}{70}$	$\frac{1}{960}$
4	0		0	0	$-\frac{1}{105}$	$-\frac{1}{160}$
4	2		$-\frac{1}{45}$	$-\frac{1}{48}$	$-\frac{1}{70}$	$-\frac{11}{960}$
4	4		$\frac{8}{45}$	$\frac{7}{48}$	$\frac{8}{105}$	$\frac{17}{288}$
4	5		$\frac{4}{45}$	$\frac{1}{16}$	$\frac{4}{105}$	$\frac{77}{2880}$
4	9		entfällt	entfällt	$\frac{2}{105}$	$\frac{11}{720}$

gangbarer Weg sein mag, können wir dies im Differentialgleichungslöser MUESLEA nicht tun, da die Berechnung der Grobgitter-Matrixelemente dadurch zu aufwendig würde.

Die meisten Mehrgittercodes erfüllen (i) und vermeiden dadurch alle Schwierigkeiten (siehe z. B. [68]). Mitchell [64], der die von Stefan Loh benutzte FORTRAN-Routine MGGHAT entwickelt hat, benutzt quadratische Basisfunktionen und sogar Basisfunktionen höherer Ordnung, testet seinen Code aber nur an Problemen wie der Laplacegleichung (wo (iii) trivial erfüllt ist) und solchen Problemen mit moderaten Oszillationen der Koeffizientenfunktionen, wo (ii) bereits auf dem groben Gitter erfüllt ist und damit die Stabilität garantiert ist.

Trotz der Schwierigkeiten möchten wir quadratische Basisfunktionen als finite Elemente einsetzen, da sie signifikante Vorteile gegenüber linearen bieten: bessere Speichereffizienz[23], bessere Qualität der Approximation und damit höhere Rechengeschwindigkeit. Der Schritt von linearen zu quadratischen Basisfunktionen bietet eine wesentliche Verbesserung, wie Mitchell [64] feststellt.

Die naheliegende Idee zur Lösung des Dilemmas ist, stückweise lineare Basisfunktionen $\eta_k^L(\vec{x})$ zur Entwicklung der Koeffizientenfunktion $\kappa(\vec{x})$ zu verwenden. Als Finite-Elemente-Basis zur Entwicklung der Lösungsfunktion ϕ^a nehmen wir aber weiterhin die stückweise quadratischen Funktionen. Abb. 4.19 zeigt Graphen der Ansatzfunktionen im zweidimensionalen Fall. Die neuen, stückweise linearen Ansatzfunktionen entstehen wie folgt: zunächst führen wir einen regulären (symmetrischen) Verfeinerungsschritt durch, d. h. wir teilen den Simplex in vier *Minisimplices* halber Größe. Dann werden einfach die bekannten, auf jedem Minisimplex linearen Basisfunktionen verwendet (Abb. 4.19 unten). Im dreidimensionalen Fall ist diese Vorschrift nicht eindeutig, da es drei verschiedene Möglichkeiten der regulären Verfeinerung gibt: hier wird ein Simplex in acht Teile geschnitten — zunächst werden die vier Eck-Minisimplices abgeschnitten, und dann wird der übriggebliebene zentrale Oktaeder in einer von drei möglichen Weisen in weitere vier Simplices zerlegt. Die natürlichste Vorgehensweise in unserem Fall ist es, über die drei möglichen Arten zu mitteln, was in sehr komplizierten, aber immer noch stückweise linearen Funktionen resultiert.

In Tab. 4.1 sind einige Integrale aufgeführt, die zur Berechnung des diskretisierten Differentialoperators (4.45) benötigt werden. Symmetriebetrachtungen ermöglichen es, die Tabellen knapp und trotzdem

[23] Im Falle quadratischer Ansatzfunktionen in drei Dimensionen hat ein Simplex zehn Knoten; im linearen Fall nur vier. Um in einem Simplexnetz die gleiche Anzahl Knoten zu erreichen, benötigt man daher quadratischen Fall weniger Simplices als im linearen Fall, erreicht aber gleichzeitig eine bessere (weil quadratische) Approximation der Lösungsfunktion. Simplices sind reine "Verwaltungsobjekte", die nur zum Aufstellen der Gleichungen und der Knotenorganisation dienen — die Feldwerte sind in den Knoten gespeichert. Deshalb ist es günstig, ihre Anzahl eher gering zu halten.

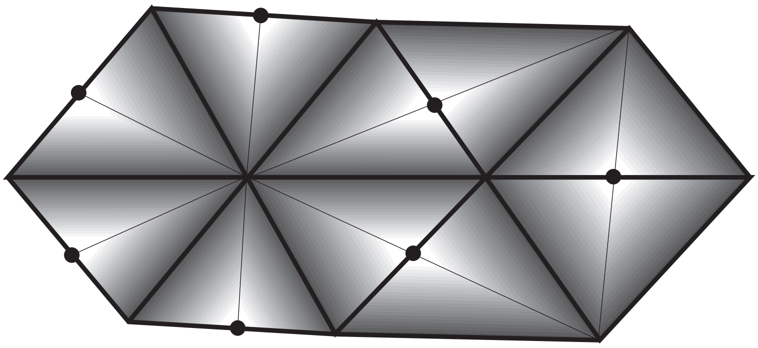


Abbildung 4.20: *Wenn eine Gebietspartition (fette Dreiecke) durch Bisektion verfeinert wird, entstehen neue, "rote" Knoten (dicke Punkte), die als Trägerknoten für neue lineare Basisfunktionen dienen. Die neuen Basisfunktionen sind hier durch Schattierung angedeutet (Funktionswerte: weiß= 1, schwarz= 0) und paarweise orthogonal, da ihre Träger paarweise disjunkt sind.*

vollständig zu halten[24].

4.3.4.4 Grobe und feine Relaxation auf zwei Gitterstufen

Unser Vorgehen auf dem groben Gitter mit nur 27 Knoten ist nun ausführlich beschrieben worden. Nun geht es darum, das Zusammenspiel der groben und feinen Gitter zu beschreiben. Es ist dabei ein großes Problem, eine übersichtliche Notation einzuführen, da wir in einem Raum von finiten Elementen verschiedene Basen betrachten müssen. In Anbetracht des Umfangs der existierenden Literatur zu diesem Thema [62, 64, 67] soll hier versucht werden, die Methode mit möglichst wenig Algebra und vor allem anhand von Grafiken zu erläutern.
Für jede Simplexgeneration g bezeichnen wir diejenigen Knoten des Netzes, die in der Simplexgeneration $g-1$ noch nicht im Netz enthalten waren, als *rote Knoten*. Diese Knoten sind dann per definitionem (Seite 87) auch von Generation g. Die Knoten der Generationen $0 \ldots g-1$ heißen *schwarze Knoten*. In diesem Abschnitt betrachten wir nur eine einzige Verfeinerungsstufe[25]. Nehmen wir an, das grobe Gitter sei um eine Stufe verfeinert worden, d. h. manche der groben Generation-0-Simplices sind zu Kindsimplices der Generation 1 geteilt worden, während andere intakt geblieben sind. Die (offenen) Verfeinerungszellen (Abb. 4.14) sind als Resultat unseres Bisektionsschemas disjunkt. Im Falle linearer Ansatzfunktionen entsteht ein neuer, roter Knoten (und damit eine neue Basisfunktion) pro Verfeinerungszelle. Die neu hinzukommenden

[24] In CDM kommen viele solcher Tabellen zum Einsatz, die alle im File `tables.cpp` untergebracht sind

[25] Später wird es von der betrachteten Verfeinerungsstufe (Generation) abhängen, welche Knoten gerade die "roten" sind und welche die "schwarzen".

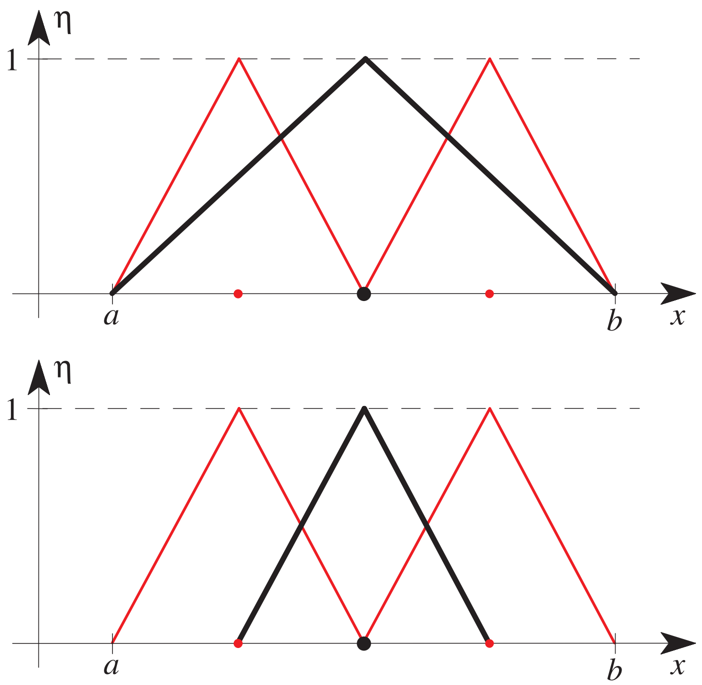


Abbildung 4.21: Hierarchische Basis (oben) und nodale Basis (unten) im Falle eindimensionaler finiter Elemente. Der fette schwarze Punkt in der Mitte gehört zum groben Gitter, die kleineren (roten) Punkte links und rechts sind die roten Knoten des feinen Gitters. Lässt man bei der Verfeinerung die alten, groben Ansatzfunktionen unverändert, entsteht die hierarchische Basis (oben). Die nodale Basis (unten) entsteht, wenn man die groben Ansatzfunktionen so verändert, dass die Entwicklungsformel (4.33) *wieder gilt, d. h. die Entwicklungskoeffizienten einfach gleich den nodalen Feldwerten sind. Man bemerke, dass beide Basen den gleichen Funktionenraum aufspannen.*

Basisfunktionen einer Generation sind wegen der disjunkten Träger alle zueinander orthogonal (Abb. 4.20). Im Falle quadratischer Ansatzfunktionen entstehen in jeder Verfeinerungszelle *mehrere* neue, rote Knoten; die neu hinzukommenden Basisfunktionen einer Zelle C spannen dann einen Unterraum U_C auf, den wir kurz *Zellraum* nennen. Die Zellräume einer Generation sind zueinander orthogonal,

$$U_C \perp U_{C'} \quad \forall\, C \neq C' \quad ,$$

da die Träger ihrer erzeugenden Basisfunktionen — nämlich die (offenen) Verfeinerungszellen selbst — disjunkt sind,

$$C \cap C' = \emptyset \quad \forall\, C \neq C' \quad .$$

Der 27-dimensionale Funktionenraum, der durch unsere Ansatzfunktionen auf dem groben Gitter (*schwarze Trägerknoten*) aufgespannt wurde, wird also durch die Verfeinerung um einen Satz neuer Ansatzfunktionen mit *roten Trägerknoten* ergänzt. Dabei werden die ursprünglichen groben Ansatzfunktionen zunächst nicht verändert, es werden nur die neuen hinzugefügt. Wir arbeiten also zunächst mit einer *hierarchischen Basis* (Abb. 4.21 oben), die der hierarchischen Mehrgittermethode auch ihren Namen gegeben hat. Wir bezeichnen die hierarchische Basis mit $\{\eta_k^h(\vec{x})\}$, die nodale mit $\{\eta_k^n(\vec{x})\}$. Man macht sich leicht folgende Fakten über die beiden Basen klar:

(i) Die hierarchische und nodale Basis spannen beide den gleichen Funktionenraum auf,

$$\langle \eta_k^h \rangle \;=\; \langle \eta_k^n \rangle \qquad . \tag{4.54}$$

(ii) Für alle roten Trägerknoten $\vec{n}_k$ sind die Basisfunktionen identisch,

$$\eta_k^h \;=\; \eta_k^n \quad \forall \text{ rote Trägerknoten } \vec{n}_k. \tag{4.55}$$

(iii) Für alle schwarzen Trägerknoten $\vec{n}_k$ sind die Entwicklungskoeffizienten einer Funktion identisch in beiden Basen und identisch mit den Feldwerten an den Knoten,

$$\begin{aligned} f(\vec{x}) \;&=\; \sum_{\text{Knoten } \vec{n}_k} a_k \eta_k^h(\vec{x}) \;=\; \sum_{\text{Knoten } \vec{n}_k} b_k \eta_k^n(\vec{x}) \\ &\Longrightarrow \; a_k = b_k = f(\vec{n}_k) \quad \forall \text{ schwarze Trägerknoten } \vec{n}_k. \end{aligned} \tag{4.56}$$

(iv) Im Falle der nodalen Basis sind die Entwicklungskoeffizienten stets gleich den Funktionswerten an den Knoten,

$$b_k = f(\vec{n}_k) \quad \forall\, k. \tag{4.57}$$

(v) Die hierarchischen Basisfunktionen lassen sich durch die nodale Basis leicht ausdrücken:

$$\eta_k^h(\vec{x}) \;=\; \sum_{\text{Knoten } \vec{n}_l} \eta_k^h(\vec{n}_l)\eta_l^n(\vec{x}). \tag{4.58}$$

Dabei gilt $\eta_k^h(\vec{n}_l) = \delta_{kl}$, falls $\vec{n}_k$ ein roter (feiner) Knoten ist oder $\vec{n}_l$ ein schwarzer (grober) Knoten. Nur falls $\vec{n}_k$ schwarz und $\vec{n}_l$ rot ist, hat $\eta_k^h(\vec{n}_l)$ überhaupt einen nichttrivialen Wert. Da die Matrix $\eta_k^h(\vec{n}_l)$ folglich Dreiecksform besitzt, ist die Inversion von (4.58) einfach zu bewerkstelligen. Der Übergang von nodalen zu hierarchischen Entwicklungen ist also kein Problem. Der ständige Wechsel zwischen nodaler und hierarchischer Basis ist aber wesentlich für das Verständnis der hierarchischen Mehrgittermethode.

Der vergrößerte Funktionenraum ist die direkte (nicht orthogonale) Summe eines *groben* und eines *feinen Sektors*. Der grobe Sektor

$$S_c \;:=\; \langle\{\eta_k^h|\vec{n}_k \; schwarz\}\rangle \tag{4.59}$$

(Index 'c' steht für *coarse*, grob) ist identisch mit dem unverfeinerte Finite-Elemente-Raum vor der Verfeinerung. Der feine Sektor

$$S_f \;:=\; \langle\{\eta_k^n|\vec{n}_k \; rot\}\rangle \;=\; \bigoplus_{\text{Zellraum } C} U_C \tag{4.60}$$

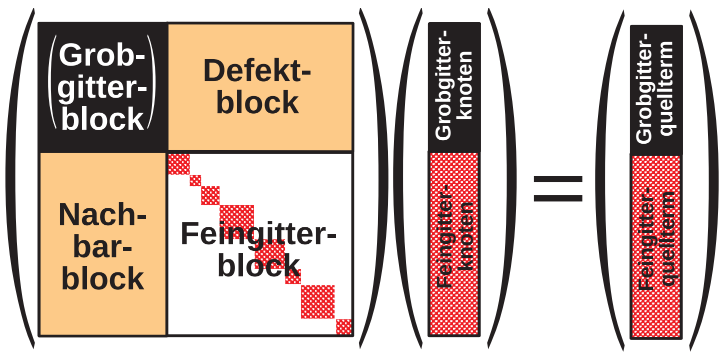
(Index 'f' steht für *fine*) wiederum ist die orthogonale Summe der Zellräume aller Verfeinerungszellen dieser Generation.
Durch die disjunkten Verfeinerungszellen gilt die Orthogonalität der Zellräume noch viel allgemeiner: Alle Matrixelemente beliebiger Differentialoperatoren verschwinden zwischen Ansatzfunktionen verschiedener Zellräume:

$$\int \phi D \phi' \;=\; 0 \quad \forall\, \phi,\, \phi' \text{ mit } \phi \in U_C \neq U_{C'} \ni \phi' \tag{4.61}$$

gilt für einen beliebigen Differentialoperator D. Es sei nochmals betont, dass dieser ästhetische Zerfall des feinen Sektors alleine ein Resultat unseres Bisektionsschemas ist. Im regulären Teilungsschema sind die Verhältnisse komplizierter.
Schauen wir auf die Struktur der diskretisierten Differentialgleichung (4.24),

$$\sum_l D_{kl}\phi_l \;=\; r_k \quad , \tag{4.62}$$

in der hierarchischen Basis. Da die Zellräume die Bedingung (4.61) erfüllen, nimmt der Differentialoperator D_{ij} im feinen Sektor eine diagonale Blockmatrixform an[26]. Im folgenden ist die Gleichung (4.62) grafisch und algebraisch notiert:



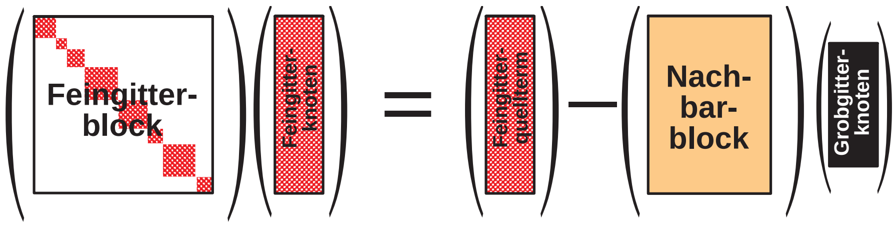
$$\sum_l \langle \eta_k^h \,|\, D \,|\, \eta_l^n \rangle \, \phi_l \;=\; \langle \eta_k^h \,|\, \rho \rangle \tag{4.63}$$

[26]Wir gehen hier nicht näher auf die Behandlung der Dirichletschen Randbedingungen ein. Es würde die Erläuterungen verkomplizieren, ohne einen besseren Einblick in die Mehrgittermethode zu bieten. Siehe aber Abschnitt 4.3.4.9 für Hinweise hierzu.

Zur Projektion haben wir die hierarchische Basis $\left\langle \eta_k^h \right|$ verwendet. Im Lösungsvektor (ϕ_l) steht für jeden Knoten $\vec{n}_l$ einfach der Feldwert an diesem Knoten $\phi_l = \phi(\vec{n}_l)$. Dieser Feldwert entspricht dem Entwicklungskoeffizienten von ϕ in der nodalen Basis (Eigenschaft (iv) oben), daher steht rechts vom Differentialoperator die nodale Entwicklung $\sum_l \phi_l \left| \eta_l^n \right\rangle$. Der Lösungsvektor (ϕ_l) zerfällt in Feldwerte an schwarzen (groben) und roten (feinen) Knoten, die in der grafischen Gleichung einfach mit "Grobgitterknoten" und "Feingitterknoten" bezeichnet sind. Der Quellterm (rechte Seite) zerfällt analog in einen groben und einen feinen Teil. Zur Operatormatrix in (4.63): Die Verfeinerung vergrößert die Operatormatrix um drei Blöcke. Dabei bildet der *Feingitterblock* eine Blockdiagonalmatrix, deren Blöcke separat auf je einen Zellraum wirken (da die Zellräume (4.61) erfüllen). Der *Nachbarblock* und der *Defektblock* beschreiben die Wechselwirkung der feinen mit den groben Knoten. Den *Grobgitterblock* (oben links in (4.63)) diskutieren wir in Kürze bei der Grobgitterrelaxation.

Das Wesen der hierarchischen Mehrgittermethode ist es, abwechselnd die feinen (roten) und die groben (schwarzen) Gitterknoten per Gauß-Seidel-Iteration zu *relaxieren*. Dabei wird auf dem groben Gitter der (bereits vor der Verfeinerung bekannte) grobe Differentialoperator invertiert, wodurch auch weiter entfernte Knoten miteinander kommunizieren und langwellige Störungen abgedämpft werden. Auf dem feinen Gitter wird der präzisere, fein diskretisierte Differentialoperator benutzt, der das nötige Detail in die Lösungsfunktion bringt.

Zur roten Relaxation: Die roten Knoten werden relaxiert, indem die untere Hälfte des Gleichungssystems (4.63)



(4.64)

nach den Feldwerten der roten Knoten (Vektor links) gelöst wird. Dieses Gleichungssystem zerfällt (dank des Bisektionsschemas) in Blöcke, in denen jeweils nur die Knoten einer Verfeinerungszelle (Abb. 4.14) miteinander kommunizieren. Die Unbekannten sind dabei die roten Knoten, während die schwarzen als feste "Randbedingungen" in diesen Schritt eingehen. Das Zerfallen des Gleichungssystems in Zellräume ermöglicht die exakte Lösung von (4.64), nachdem alle Blöcke der Diagonalblockmatrix LR-zerlegt sind[27].

[27] Die rote Relaxation wird in MUESLEA blockweise durch die Klasse RedRelax durchgeführt: es existiert für jede Verfeinerungszelle genau eine RedRelax-Instanz, die alle

Nun die Grobgitterrelaxation (*coarse grid relaxation*): Hier werden die Feldwerte an den (schwarzen) Grobgitterknoten relaxiert. Dabei *denken* wir auf dem groben Gitter und wollen den grob diskretisierten Differentialoperator (den vor dem Verfeinerungsschritt bereits bekannten) invertieren, damit weiter entfernte Knoten miteinander kommunizieren können und so langwellige Störungen abgedämpft werden. Vorher berechnete Korrekturterme des feinen Gitters sollen aber dafür sorgen, dass die Präzision dieser Grobgitterrelaxation den Ansprüchen des feinen Gitters genügt. Nun ist der *Grobgitterblock*

$$\langle \eta_k^h \,|\, D \,|\, \eta_l^n \rangle \quad (\vec{n}_k, \vec{n}_l \text{ schwarze (grobe) Knoten}) \tag{4.65}$$

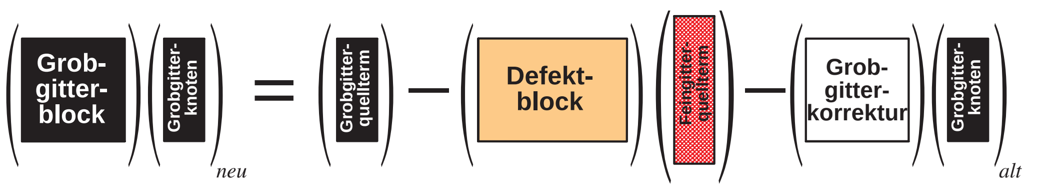
oben links in (4.63) eingeklammert gezeichnet, da er *nicht* identisch ist mit der unverfeinerten Operatormatrix

$$\langle \eta_k^h \,|\, D \,|\, \eta_l^h \rangle \quad (\vec{n}_k, \vec{n}_l \text{ schwarze (grobe) Knoten}) \tag{4.66}$$

die bereits auf dem groben Gitter bekannt war. Deshalb transformieren wir die Grobgittergleichungen (oberer Teil der grafischen Matrixgleichung (4.63)) so, dass sie die Grobgittermatrix (4.66) enthält, die wir auf dem groben Gitter schnell invertieren können. Dazu müssen wir den Übergang von der nodalen Entwicklung in (4.63) zur hierarchischen Entwicklung vollführen, d. h. wir wollen den Grobgitterteil von (4.63) in

$$\sum_l \langle \eta_k^h \,|\, D \,|\, \eta_l^h \rangle \, \phi_l \;=\; \langle \eta_k^h \,|\, \rho \rangle \qquad \forall \text{ schwarzen (groben) Knoten } \vec{n}_k \tag{4.67}$$

umschreiben und nach den groben Koeffizienten des Lösungsvektors auflösen. Nun kommen uns die Eigenschaften $(i) - (v)$ der hierarchischen und nodalen Basis entgegen, die wir oben notiert hatten. Der grobe Teil des Lösungsvektors enthält Entwicklungskoeffizienten der Lösungsfunktion für Ansatzfunktionen an den schwarzen (groben) Knoten, und diese sind in beiden Basen (nodal und hierarchisch) identisch. Der obere (grobe) Teil des Lösungsvektors muss daher gar nicht transformiert werden. Die zu invertierende Operatormatrix ist auf dem groben Gitter näherungsweise bereits bekannt.
Die Grobgitterrelaxation geschieht dann durch Lösung der Gleichung



(4.68)

nötigen Matrizen — auch den relevanten Teil der LR-zerlegten Feingittermatrix und den relevanten Teil des Nachbarblocks — enthält.

Dabei wird zunächst die rechte Seite mit den vorhandenen Knotenwerten auf dem feinen Gitter präpariert. Im nächsten Schritt wird das bekannte Grobgitter-Lösungsverfahren benutzt, um die Werte an den groben (schwarzen) Knoten zu relaxieren. Dabei werden langwellige Störungen schnell abgedämpft.
Zum letzten Term in (4.68): Es kann im Falle ortsabhängiger Operatoren (man denke an $\vec{\nabla}\cdot\kappa(\vec{x})\vec{\nabla}$) sein, dass die Operatordiskretisierung auf dem groben Gitter nicht perfekt ist, weil dort die Koeffizientenfunktion $\kappa(\vec{x})$ nicht hinreichend präzise diskretisiert war. Dies führt zu dem letzten Korrekturterm "Grobgitterkorrektur" auf der rechten Seite von (4.68). Der für diese zusätzlichen Grobgittermatrizen benötigte Speicherplatz ist etwas unangenehm groß: diese Matrixeinträge alleine verschlingen pro Verfeinerungszelle so viel Speicherplatz wie alle anderen Matrizen zusammen.
Die Berechnung der rechten Seite von (4.68) lässt sich abkürzen, wenn der Operator auf dem groben Gitter bereits exakt diskretisiert war (Wegfall des letzten Grobgitterkorrekturterms) oder der Operator symmetrisch ist (dann lässt sich der Defektblock aus dem Nachbarblock mittels eines Basiswechsels errechnen). In beiden Fällen wird auch Speicherplatz für entsprechende Matrizen gespart. Auf Details soll hier aber nicht näher eingegangen werden. In MUESLEA sind beide Fälle implementiert und lassen sich bei der Spezifikation der Differentialgleichung durch Setzen der Flags COARSE_EXACT und/oder SYMMETRIC in der PDEquation-Instanz aktivieren. Diese Flags beeinflussen nur die Matrixberechnung (in red.cpp) und die Berechnung der Grobgitterkorrektur (Methode PDESolver::ComputeCoarseGridCorrection in solve.cpp). Sicherer, aber langsamer und erheblich speicherplatzintensiver ist es, diese Flags im Zweifelsfall ungesetzt zu lassen.
Eine vollständige Iteration im Stil des hierarchischen Mehrgitterverfahrens läuft folgendermaßen ab:

- Rote Relaxation
- Berechnung der Grobgitterkorrekturterme (rechte Seite von (4.68))
- Grobgitterrelaxation
- Rote Relaxation

Zur schwarzen Relaxation: Wir gehen in MUESLEA über die reine Anwendung des hierarchischen Mehrgitterverfahrens hinaus. Insbesondere führen wir (ähnlich Mitchell [64]) noch zusätzlich *schwarze Relaxationen* durch, also Relaxationen der schwarzen Knoten *auf dem feinen Gitter*. Dazu wird der obere Grobgitterteil (4.67) nochmals in der nodalen Basis relaxiert,

$$\sum_l \left\langle \eta_k^n \,\middle|\, D \,\middle|\, \eta_l^n \right\rangle \phi_l \;=\; \left\langle \eta_k^n \,\middle|\, \rho \right\rangle \quad \forall \text{ schwarzen (groben) Knoten } \vec{n}_k. \tag{4.69}$$

Dies geschieht bequem und ohne viel rechnerischen Aufwand vor dem roten Relaxationschritt. Dort wird ein einziger Gauß-Seidel-Iterationsschritt durchgeführt. Die nötigen Werte für die rechte Seite liegen zu diesem Zeitpunkt bereits in den Knoten gespeichert vor, und die Berechnung der Matrixelemente des Operators in (4.69) erfolgt sowieso als Nebenprodukt der anderen zu berechnenden Matrizen. Versuche haben gezeigt, dass das Konvergenzverhalten des Mehrgitterverfahrens sich mit der zusätzlichen schwarzen Relaxation nochmals deutlich verbessert. In der schwarzen Relaxation werden nur solche schwarzen Knoten relaxiert, die zu roten Knoten benachbart sind. Bei stark lokaler Verfeinerung sind das bei weitem nicht alle schwarzen Knoten.

Eine vollständige Iteration im MUESLEA-Stil läuft also folgendermaßen ab:

- Schwarze Relaxation
- Rote Relaxation
- Berechnung der Grobgitterkorrekturterme
- Grobgitterrelaxation
- Rote Relaxation

Diese Schritte werden wiederholt, bis die gewünschte Genauigkeit erreicht ist.

4.3.4.5 Volles Mehrgitterverfahren

Der komplizierteste Abschnitt liegt nun hinter uns. Die Erweiterung auf viele Generationen von Verfeinerungszellen ist nun fast trivial; der Leser mag es bereits erraten haben. Die Relaxation der Grobgittergleichung im letzten Abschnitt bezog sich auf die gröbste Partition mit nur 27 Knoten, auf der wir die Gleichungen stets numerisch exakt lösen können. Das ist eigentlich keine Notwendigkeit; das "grobe Gitter" kann seinerseits wieder analog behandelt werden. So entsteht eine verschachtelte Struktur; der grobe Sektor ist seinerseits wieder die direkte Summe eines feinen und eines noch gröberen Sektors. Die Matrixgleichung

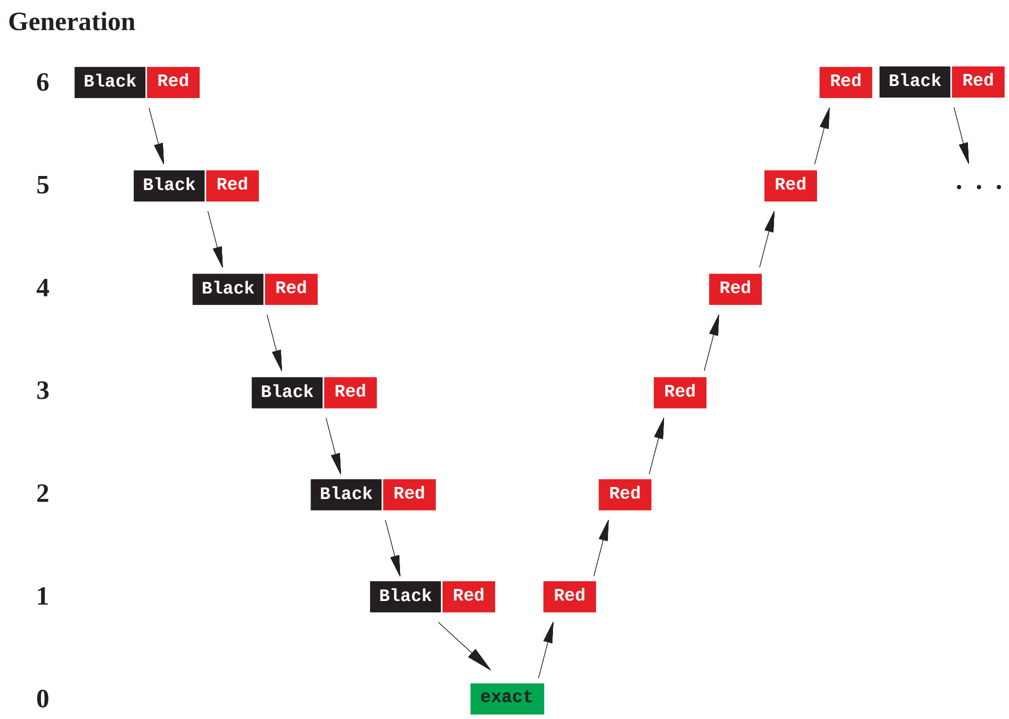
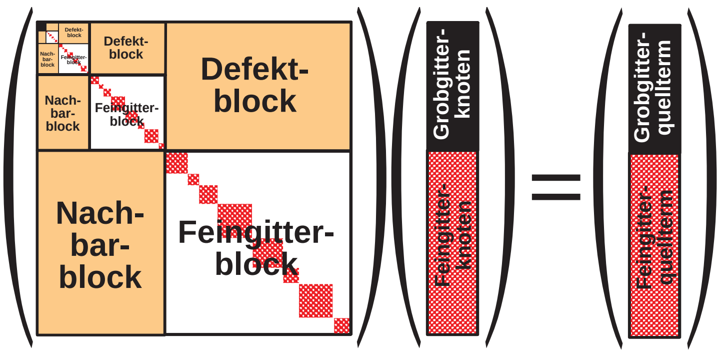


Abbildung 4.22: V-Zyklen. Auf jeder Verfeinerungsstufe wird zur Relaxation das nächstgröbere Gitter konsultiert, bis auf dem gröbsten Gitter (Generation 0) die Gleichungen exakt gelöst werden können. Der ganze Zyklus wird bis zur gewünschten Präzision iteriert.

lautet dann formal



, (4.70)

und die Relaxationsiterationen nach dem oben skizzierten Algorithmus laufen dann im Stil eines *V-Zyklus* (*V-Cycle*): Jede Gitterstufe konsultiert während ihrer Relaxation das nächstgröbere Gitter. Das Schema ist in Abb. 4.22 grafisch dargestellt.

4.3.4.6 Grobstruktur und Präzisionssteuerung

Um das Prinzip der Präzisions- und Verfeinerungssteuerung zu verstehen, muss zunächst der Begriff des *Residuums* eingeführt werden. Unter dem Residuum versteht man die betragliche Differenz zwischen linker und rechter Seite der Differentialgleichung in schwacher Form,

$$\mathrm{Res}_i \;:=\; \left| \int \eta_i(\vec{x}) \left(\vec{\nabla} \cdot \left(\kappa(\sigma(\vec{x})) \vec{\nabla} \sum_j \phi_j^a \eta_j(\vec{x}) \right) + \rho^a(\vec{x}) \right) d^3x \right| \quad . \tag{4.71}$$

Verschwindet das Residuum für jede Basisfunktion $\eta_i(\vec{x})$, so bedeutet dies die Erfüllung der Ritz-Galerkin-Gleichungen (4.23). Wir stellen als Forderung an die Qualität der vom Mehrgitterverfahren gelieferten Differentialgleichungslösung, dass der relative Fehler einen bestimmten Wert (meist 1%) nicht überschreitet, d. h.

$$r \;:=\; \frac{\sum_i \mathrm{Res}_i}{V \cdot \max\limits_V |\phi^a|} \;<\; 1\% \quad . \tag{4.72}$$

Hier ist V das gesamte Gebietsvolumen, und i läuft über den vollständigen Satz Ansatzfunktionen. Die Größe r nennen wir leicht verfälschend den *relativen Fehler* der Lösung.
Das Residuum wird während jedes V-Zyklus *en passant* ausgewertet. Die V-Zyklen werden wiederholt, bis das Residuum die gewünschte Schranke unterschritten hat, oder offenbar keine Konvergenz eintritt (im letzteren Fall kann man nur hoffen, dass sich die Situation nach einigen Verfeinerungsschritten wieder bessert). Wenn eine Serie (etwa 10-50) V-Zyklen beendet ist, folgt ein sog. Verfeinerungslauf: Hier werden in

jeder Generation für jede Verfeinerungszelle die jeweils nächste Verfeinerungsstufe probeweise ausgeführt und das Residuum für die neu hinzukommenden Ansatzfunktionen berechnet. Abhängig von dieser Größe wird dann entschieden, ob die Verfeinerung wirklich durchgeführt wird und die relevanten Matrizen in die nächste Serie von V-Zyklen aufgenommen werden. Die Verfeinerungsläufe brechen ab, wenn entweder die probeweise ausgerechneten Residuen immer kleiner werden oder die Anzahl der neu hinzugekommenen Verfeinerungszellen eine bestimmte Zahl überschreitet.
Verfeinerungsläufe und V-Zyklus-Serien wechseln sich ab, bis Konvergenz auftritt oder Speicherplatzprobleme (wegen zu vieler Matrizen) auftreten. Im letzteren Fall ist die Lösung aber oft trotzdem brauchbar.
Ohne einen Lehrtext zum Thema Mehrgitterverfahren verfassen zu wollen, wurde in diesem Abschnitt das hierarchische Mehrgitterverfahren aus Sicht des Autors zu schildern versucht. Der CDM-Benutzer sei ermutigt, vor allem die Referenz [62] zur Vertiefung heranzuziehen.

4.3.4.7 Berechnung der Matrizen und Relaxation

In diesem und den folgenden zwei Abschnitten soll der für das beschriebene Mehrgitterverfahren relevante Teil der MUESLEA-Klassenhierarchie und das Zusammenwirken seiner Teile demonstriert werden:
Zunächst das Aufstellen der Grobgittergleichungen: Die Knoten der groben Partition werden vom RedRelax-Konstruktor (erstes Aufrufformat "Constructor for a Coarse Domain Partition") in red.cpp gesammelt. Dabei wird in der RedRelax-Instanz das Flag COARSEST gesetzt. Später im Konstruktor wird die Methode ReComputeMatrices aufgerufen, die für alle beteiligten Simplices der groben Partition wiederum AddFineContribution (alle Routinen in red.cpp) aufruft. Dort wird PDE.GetOperator (pdequation.cpp) benutzt, um die Matrix (4.45) für einen Simplex zu erhalten.
Der Umweg über die Klasse PDEquation (pdequation.h) dient dazu, den Differentialgleichungslöser allgemeiner zu halten: in PDEquation sind alle zur Definition des Randwertproblems nötigen Daten (Gebiet Ω, Randbedingungen, Operator, Quellterm) versammelt. Nun gehört eine RedRelax-Instanz zu einem PDESolver (solve.h) (Baumstruktur Abb. 4.23); die RedRelax::PDE-Referenz verweist dabei stets auf die eine PDESolver::PDE-Instanz. Diese wird bei der Konstruktion der PDESolver-Instanz (bspw. in CommandStreamParser::phiRelax, siehe parser.h) von einer Vorlage kopiert, beispielsweise von CommandStreamParser::InMediumPDE8. Der PDEquation::Operator-Eintrag enthält dann einen Zeiger auf eine matrixgenerierende Funktion, beispielsweise InMediumMinusLaplaceFunc (operator.cpp).
Die Funktion InMediumMinusLaplaceFunc (endlich eine *low-level-Routine*) berechnet die 10×10-Matrix (4.45) für ein Simplexvolumen wie folgt:

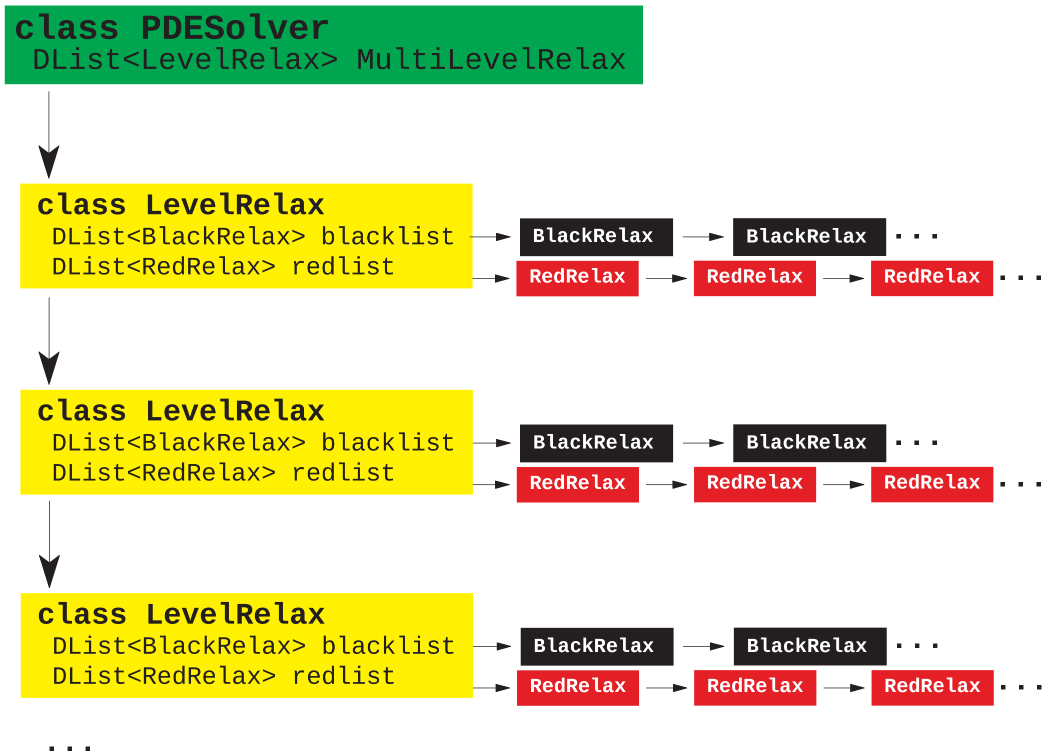


Abbildung 4.23: Struktur verketteter Listen (Baumstruktur) ausgehend von einer PDESolver*-Instanz. Die* LevelRelax*-Instanzen sind nach absteigender Generation sortiert; ein V-Zyklus läuft also in dieser Liste einmal hinunter und wieder hinauf. Die* BlackRelax*- und* RedRelax*-Instanzen enthalten alle Matrizen sowie Zeiger auf die benötigten Knoten, um einen Relaxationsschritt durchführen zu können.*

Mit der Entwicklung (4.47) kann die Matrix D_{ij} aus (4.45) wie folgt geschrieben werden

$$
\begin{aligned}
D_{ij} &= \int \kappa(\sigma(\vec{x})) \left(\vec{\nabla}\eta_i(\vec{x})\right) \cdot \left(\vec{\nabla}\eta_j(\vec{x})\right) d^3x \\
&= \sum_{\text{Knoten } k} \kappa(\vec{n}_k) \int \eta_k^L(\vec{x}) \left(\vec{\nabla}\eta_i(\vec{x})\right) \cdot \left(\vec{\nabla}\eta_j(\vec{x})\right) d^3x \\
&= \sum_k \kappa(\vec{n}_k) \int \eta_k^L(\vec{x}) \sum_{l=1}^{3} \frac{\partial\eta_i(\vec{x})}{\partial x_l}\frac{\partial\eta_j(\vec{x})}{\partial x_l} d^3x \\
&= \sum_k \kappa(\vec{n}_k) \sum_{\text{Simplices } S} \int_{\mathcal{V}_S} \eta_k^L(\vec{x}) \sum_{l=1}^{3} \frac{\partial\eta_i(\vec{x})}{\partial x_l}\frac{\partial\eta_j(\vec{x})}{\partial x_l} d^3x \\
&= \sum_S \sum_{m,n=1}^{3} \left(\sum_{l=1}^{3} \frac{\partial\alpha_m}{\partial x_l}\frac{\partial\alpha_n}{\partial x_l}\right) \left(\sum_k \kappa(\vec{n}_k) \int_{\mathcal{V}_S} \eta_k^L(\vec{x}) \frac{\partial\eta_i(\vec{x})}{\partial\alpha_m}\frac{\partial\eta_j(\vec{x})}{\partial\alpha_n} d^3x\right)
\end{aligned}
$$

$$
= \sum_S \sum_{m,n=1}^{3} \left(\sum_{l=1}^{3} \frac{\partial\alpha_m}{\partial x_l}\frac{\partial\alpha_n}{\partial x_l}\right) \left(\frac{\partial^2\eta_i(\vec{x})}{\partial\alpha_p\partial\alpha_m}\right) \left(\frac{\partial^2\eta_j(\vec{x})}{\partial\alpha_q\partial\alpha_n}\right) \left(\sum_k \kappa(\vec{n}_k) \int_{\mathcal{V}_S} \eta_k^L(\vec{x})\alpha_p\alpha_q\right) \tag{4.73}
$$

Die $\alpha_{m,n}$ sind dabei Simplexkoordinaten des aktuellen Simplexes in der Summation über Simplices S, und die partiellen Ableitungen nach $\alpha_{m,n}$ verstehen sich als Ableitung nach $\{\alpha_1, \alpha_2, \alpha_3\}$ als unabhängigen Variablen bei abhängigem α_0.
Bei der letzten Umformung in (4.73) haben wir uns zunutze gemacht, dass die η_i bekannte quadratische Funktionen der α_n sind (siehe (4.32)). Die Ableitungen

$$
\frac{\partial\eta_i(\vec{x})}{\partial\alpha_m} = \sum_{k=1}^{3} \frac{\partial^2\eta_i}{\partial\alpha_k\partial\alpha_m}\alpha_k
$$

sind damit feste, bekannte, lineare Funktionen der α_m, wobei die konstanten Koeffizienten

$$
\mathsf{BD}_{kmi} := \frac{\partial^2\eta_i}{\partial\alpha_k\partial\alpha_m}
$$

in der Tabelle Simplex::BasisDerivatives (siehe tables.cpp) gespeichert sind.
Die von Tab. 4.1 her bekannte $10 \times 4 \times 4$-Matrix

$$
\frac{1}{\mathcal{V}_S} \int_{\mathcal{V}_S} \eta_k^L \alpha_m \alpha_n d^3x \tag{4.74}
$$

ist ebenfalls gänzlich unabhängig von der Simplexform. In CDM findet man die Einträge tabelliert in Simplex::LinearizedWeightedSmallInnerProductMatrix (tables.cpp).

Schließlich zur "Metrik" in (4.73),

$$g_{mn} := \left(\sum_{l=1}^{3} \frac{\partial \alpha_m}{\partial x_l} \frac{\partial \alpha_n}{\partial x_l} \right) \quad . \tag{4.75}$$

Zur Erinnerung: Die $\alpha_{m,n}$ sind hier die Simplexkoordinaten von $\vec{x}$ im aktuellen Simplex S (über den in (4.73) summiert wird). Damit sind die $\alpha_{m,n}$ Funktionen von $\vec{x}$, wobei jeweils drei von ihnen linear unabhängig sind. Die partiellen Ableitungen nach $\alpha_{m,n}$ in (4.73) verstehen sich als Ableitung nach $(\alpha_1, \alpha_2, \alpha_3)$ als unabhängigen Variablen bei abhängigem α_0. Die Metrik (4.75) ist für einen beliebigen Simplex schnell zu berechnen — die Klasse MetricMap in localmap.h erledigt dies in ihrem Konstruktor.
Die vielen Summationen, die zur Berechnung von (4.73) auszuführen sind (Knoten k, Simplex S, kartesische Koordinaten l, Simplexkoordinaten $\{m, n, p, q\}$, Simplexknoten i und j), lassen sich geschickt so ordnen, dass pro Simplex nur etwa 1800 Multiplikationen zur Berechnung der kompletten D_{ij}-Matrix benötigt werden — eine extrem geringe Anzahl, wenn man die Zahl der Terme pro Simplex bedenkt, die (4.73) in ausmultiplizierter Form besitzt:

$$3^8 \cdot 10^2 \approx 656.000.$$

Die Summation wird in InMediumMinusLaplaceFunction (operator.cpp) ausgeführt.
Nach dieser Berechnung von D_{ij} für einen einzelnen Simplex geht es wieder zurück in der Hierarchie. Die nun gewonnene 10×10-Matrix wird in AddFineContribution auf die Knotenordnung in RedRelax::SparseMatrix umsortiert und aufaddiert (schließlich haben wir im Integral (4.45) über alle Simplices zu summieren). Die Methode ReComputeMatrices ruft dann noch LUDecomposition auf, um die Grobgittermatrix zu zerlegen. Damit ist das *Aufstellen der Matrixgleichungen* abgeschlossen; es bleibt die Lösung (*Relaxation*) der zugehörigen Gleichungen.
Während jedes *Multigrid-V-Cycles* (PDESolver::VCycle, siehe solve.cpp) wird auf der gröbsten Gitterstufe die Routine LevelRelax::PerformRedRelaxation (siehe level.h) aufgerufen, die ihrerseits per RedRelax::GaussSeidelRelax (red.cpp) die Grobgittergleichungen löst.
Auf den Verfeinerungsstufen höherer Generationen ruft die Routine PDESolver::VCylce zunächst die schwarze Relaxation (LevelRelax::PerformBlackRelaxation) auf, dann die rote Relaxation (LevelRelax::PerformRedRelaxation). Danach berechnet VCylce via ComputeCoarseGridCorrection (solve.cpp) die benötigten Korrekturterme für das grobe Gitter. Schließlich ruft VCylce sich selbst in rekursiver Art für das gröbere Gitter auf. Nach Beendigung der Grobgitterrelaxation erfolgt noch ein roter Relaxationsschritt.

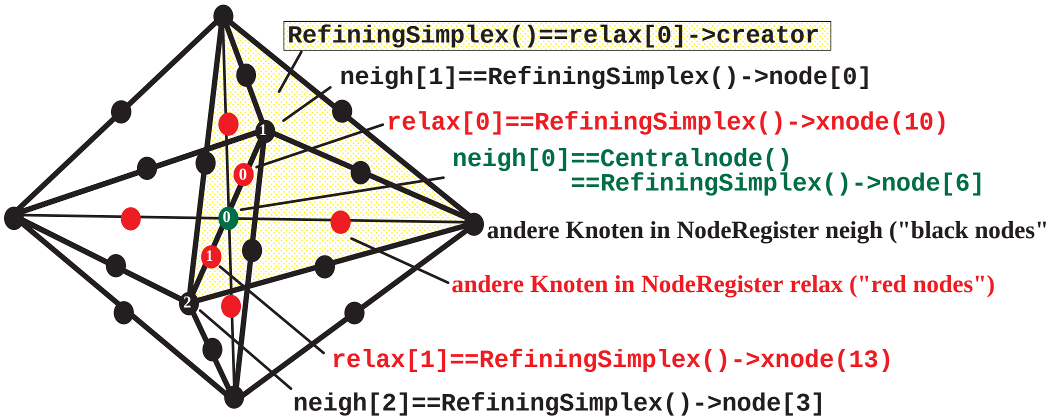


Abbildung 4.24: Knotenschema für eine Verfeinerungszelle. Die Abbildung beschreibt die Lage der Knoten und Simplices aus Sicht der zugehörigen RedRelax*-Instanz. Der zentrale Knoten* relax[0] *wird benutzt, um die Verfeinerungszelle eindeutig zu bezeichnen; sein erzeugender Simplex* relax[0] →creator *ist der Simplex, dessen Verfeinerung ursprünglich zur Erzeugung der Verfeinerungszelle geführt hat. Ein Zeiger auf diesen Simplex kann mit der Methode* RedRelax::RefiningSimplex() *abgerufen werden. Die Knoteneinträge* relax[0], relax[1] *und* neigh[0 . . . 2] *in* RedRelax *folgen dann stets der dargestellten Konvention.*

4.3.4.8 Relax-Datenstrukturen

Betrachten wir kurz die Datenstruktur der Klasse RedSparseMatrix, die in RedRelax benutzt wird. Wie RedRelax bezieht sich RedSparseMatrix dabei auf eine Verfeinerungszelle und enthält daher nur Matrixeinträge, die die beteiligten Zellknoten miteinander verknüpfen. RedSparseMatrix enthält die Matrizen relaxblock, LUrelaxblock und neighblock sowie (optional) coarseblock und defectblock. Diese entsprechen den Matrizen “Feingitterblock”, LR-zerlegter Feingitterblock, “Nachbarblock”, “Grobgitterkorrektur” und “Defektblock” (Bezeichnungen aus (4.67)), bzw. den für die aktuelle Verfeinerungszelle relevanten Einträgen dieser Matrizen.

Der Bezug zwischen Matrixzeilen und -spalten zu Knoten ist in den Komponenten RedRelax::relax und Relax::neigh gespeichert. Diese beiden Einträge sind Instanzen der Klasse NodeRegister und erlauben eine bijektive Verknüpfung von Matrixzeile/-spalte mit einem Zeiger auf den jeweiligen Knoten (Node). Dabei folgen die Einträge in RedRelax::relax und Relax::neigh gewissen Regeln (Abb. 4.24), die Suchalgorithmen in MUESLEA vereinfachen helfen.

Instanzen der Klasse BlackRelax (black.h) relaxieren immer nur einen einzigen Knoten, der durch den Zeiger relax referenziert wird. Die zu lösende Matrixzeile ist im Vektor neighline und dem Skalar relaxcoefficient gespeichert.

4.3.4.9 Behandlung der Randbedingungen

Auch wenn momentan nur *Dirichletsche Randbedingungen* zum Einsatz kommen, ist MUESLEA darauf vorbereitet, auch mit *von Neumannschen Randbedingungen* zu arbeiten. Die Behandlung der Randknoten wird dabei auf elegante Weise mit der Behandlung der inneren Knoten vereinheitlicht: die eigentliche Relaxationsmethode RedRelax::GaussSeidelRelaxation (red.cpp) macht keinen Unterschied zwischen inneren Knoten und Randknoten. Der Unterschied wird nur bei der Berechnung der Matrizen (RedRelax::AddFineContribution, RedRelax::AddCoarseContribution, beide im File red.cpp, und BlackRelax::ReComputeMatrices, in black.cpp) deutlich: hier wird für jede Zeile, deren Diagonalelement einen Randknoten referenziert, der BoundaryOperator für die Matrixeinträge herangezogen, während die inneren Knoten den eigentlichen Differentialoperator bekommen — im Programm heißt dieser einfach Operator. Der BoundaryOperator bezieht sich im Falle der CDM-Simulation immer auf die Funktion DirichletBoundaryFunc (operator.cpp), die einfach eine Diagonalmatrix zurückliefert. Ein diagonaler Eintrag für einen bestimmten Randknoten der Generation g entspricht dabei der gesamten Randfläche der anliegenden Simplices gleicher Generation. Dies ist aus Stabilitätsgründen im Mehrgitterverfahren notwendig — die Matrixeinträge müssen mit steigender Generation kleiner werden. Die Multiplikation der Matrixeinträge mit den Randflächen schlägt sich auch in der Berechnung der rechten Seite der Gleichungen nieder: Am Anfang jedes V-Zyklus, bei der Initialisierung des Defektvektors auf dem feinsten Gitter (PDEquation::AddRHS in pdequation.cpp) werden die Randwerte mit den entsprechenden Flächen multipliziert, bevor sie als Defekt in den Knoten aufaddiert werden.

4.3.5 Der letzte Schliff: Optimierung

Man kann die einfache Beobachtung machen, dass die Feldwerte an den Knoten während aufeinanderfolgender V-Zyklen "linear" konvergieren, d. h. die Abweichung eines beliebigen Knotenwertes ϕ von der Lösung s wird mit jedem V-Zyklus um einen bestimmten Faktor q reduziert:

$$(\phi^{(n+1)} - s) \;= q(\phi^{(n)} - s) + \mathcal{O}(q^2) \quad (|q| \ll 1). \tag{4.76}$$

Die Lösung dieser Rekursion ist der funktionale Zusammenhang

$$\phi^{(n)} \;\approx\; s + aq^n + \mathcal{O}(q^{2n}) \tag{4.77}$$

Die Beobachtung verschiedener Knoten während des Iterationsverfahrens bestätigt diesen Zusammenhang (Abb. 4.25). Auch ohne q und a

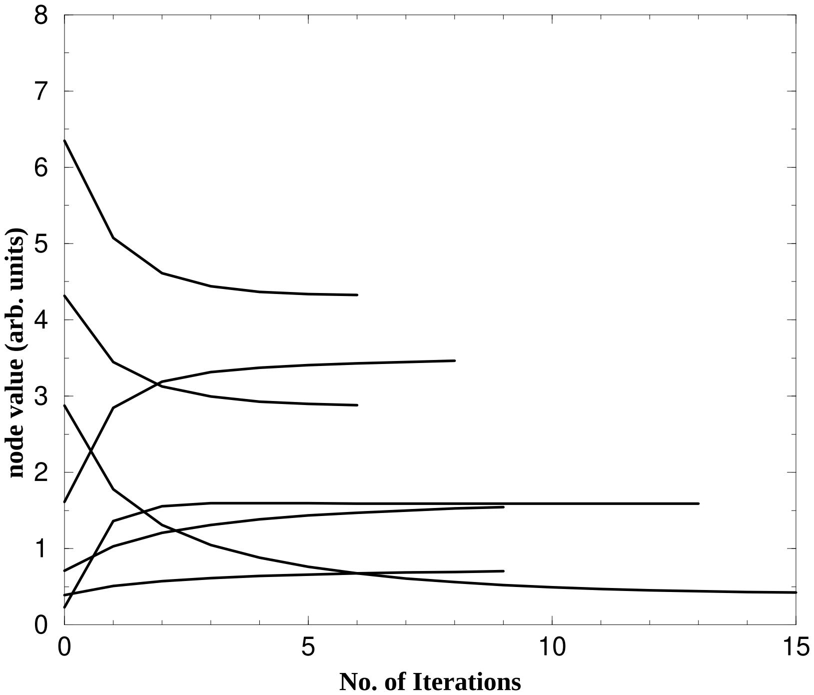


Abbildung 4.25: Konvergenz des ϕ^a-Wertes während der V-Zyklus-Iteration an verschiedenen Knoten.

für jeden Knoten $\vec{n}$ zu kennen, können wir mit dem einfachen Trick der *Aitkentransformation* (bekannt vor allem durch Rombergs Quadraturverfahren) den führenden Abweichungsterm entfernen. Dazu betrachten wir den Feldwert ϕ in drei aufeinanderfolgenden Iterationsschritten. Offenbar gilt

$$q = \frac{\phi^{(n+2)} - \phi^{(n+1)}}{\phi^{(n+1)} - \phi^{(n)}} \tag{4.78}$$

$$s = \frac{\phi^{(n+2)} - q\phi^{(n+1)}}{1-q} \quad . \tag{4.79}$$

Das praktische Vorgehen liegt nun auf der Hand: Wir speichern alle Feldwerte in einen Puffer aitken0 ab. Nach einem V-Zyklus speichern wir die Feldwerte erneut ab, diesmal in den Puffer aitken1. Nach einem weiteren V-Zyklus haben wir nun für jeden Knoten drei Feldwerte, die in äquidistanten Abständen in der Folge $\phi^{(n)}$ liegen. Mit (4.78) rechnen wir zu jedem Knoten das zugehörige q aus. Falls wirklich $|q| \ll 1$ gilt[28], führen wir mittels (4.79) eine Aitkentransformation durch und beginnen von vorne mit V-Zyklen. Es zeigt sich, dass dieses Vorgehen die Konvergenz nochmals erheblich beschleunigt (Abb. 4.26).
Die Relaxationen sind aber auch ohne Konvergenzbeschleunigung bereits sehr schnell. Wesentlich mehr Rechenzeit verschlingt die Berechnung der Matrizen und Sammlung der verschiedenen Einträge in den jeweiligen Teilmatrizen. Durch *Caching* (Zwischenspeicherung) der Matrizen[29] werden unnötige mehrfache Berechnungen der gleichen Matrix vermieden.

4.3.6 Rechenzeit und Speicherplatzbedarf

In Abb. 4.27 ist die Abhängigkeit des Rechenzeit- und Speicherplatzbedarfs der Mehrgitterroutine von der gewünschten Präzision ("relativer Fehler") der Lösung gezeigt. Der Knick in den drei Kurven, der bei kleinen Werten des Präzisionsparameters auftritt, ist damit zu erklären, dass hier der Abbruch der Berechnung erfolgte, weil die zuvor spezifizierte maximale Anzahl von Matrizen erreicht wurde. In diesem Fall befanden sich zum Zeitpunkt des Abbruchs insgesamt 230.000 Simplices, 150,000 Knoten, 20,000 RedRelax-Instanzen, 75.000 BlackRelax-Instanzen im System, zuzüglich unzähliger Matrizen, die zum schnelleren Zugriff in den Simplices zwischengespeichert waren. Der Gesamtspeicherbedarf des Mehrgitterverfahrens betrug zu diesem Zeitpunkt etwa $210MB$. Der Abbruch der Rechnung hatte natürlich zur Folge, dass die Rechenzeit und der Speicherbedarf nicht die Werte erreichten, die

[28] In der Praxis verlangen wir $q < 0.85$; dieser Wert hat sich empirisch bewährt.

[29] Zwischengespeicherte Matrizen sind über einen Zeiger matrixcache an die Klasse Simplex angeheftet.

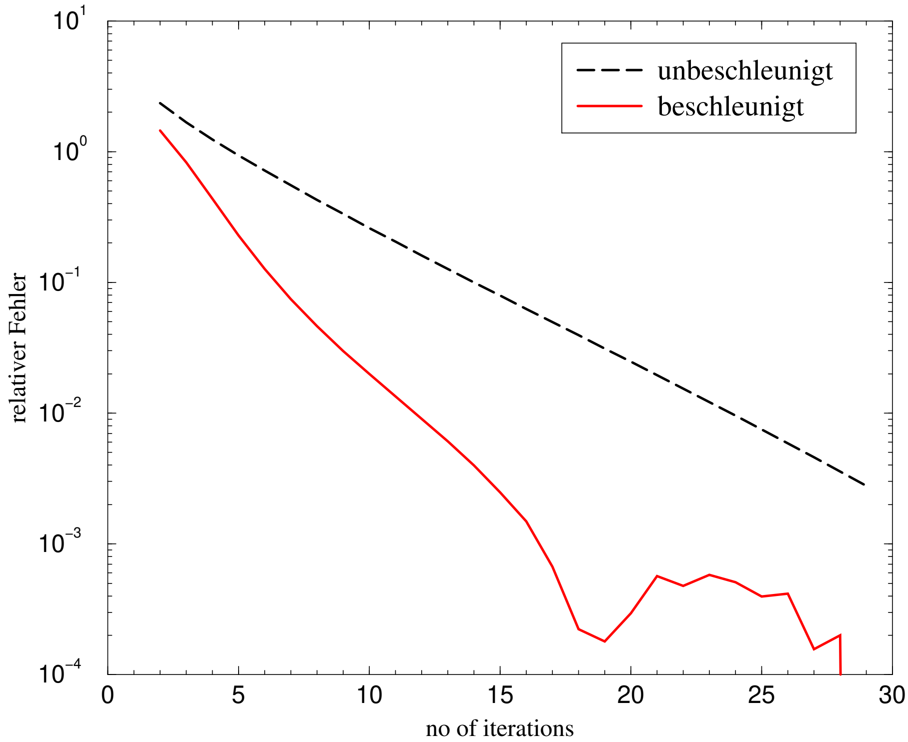


Abbildung 4.26: Aitkentransformation beschleunigt die Konvergenz erheblich.

sie zur Berechnung der Lösung mit der gewünschten Genauigkeit gebraucht hätten.

Um die Qualität der Lösungen für die beiden Extremwerte in Abb. 4.27 (rel. Fehler 0.01 und 0.0001) vergleichen zu können, wurden die Lösungsfunktionen in Abb. 4.28 auf mehrere verschiedene Weisen geplottet. In der linken Spalte sieht man die genaueste Lösung (rel. Fehler 0.0001), in der rechten die ungenaueste (rel. Fehler 0.01). Der Vergleich der ϕ^a-Funktionen (obere Zeile) lässt deutliche Unterschiede bemerken; die rechte Seite mag zwar noch qualitativ richtig sein, ist aber deutlich als schlechte Approximation auszumachen. Physikalisch wesentlich aber sind die Unterschiede im farbelektrischen Fluss $\kappa\vec{\nabla}\phi^a$, und die sind erstaunlich gering, wie die untere Zeile von Abb. 4.28 erkennen lässt. Der Farbfeldfluss hat auch ähnlichen Betrag in beiden Bildern, wie wir beispielhaft an einem Punkt des Graphen überprüfen: Die Feldenergiedichte $\kappa\vec{E}^2$ im Zentrum zwischen den beiden unteren Quarks beträgt im linken ($r = 0.001$) Bild $0.17GeV/fm^3$, im rechten ($r = 0.01$) Bild $0.2GeV/fm^3$. Die Abweichung von immerhin über 10% (über 5% in der Feldstärke) verdeutlicht, dass der Präzisionsparameter r, der ein ungefähres Maß für die relative Genauigkeit der ϕ^a-Berechnung darstellt, nicht direkt als Maß für die Genauigkeit des $\vec{D}^a$-Feldes gelten kann.

Für die praktische Optimierung unserer Simulation wurde stets die ge-

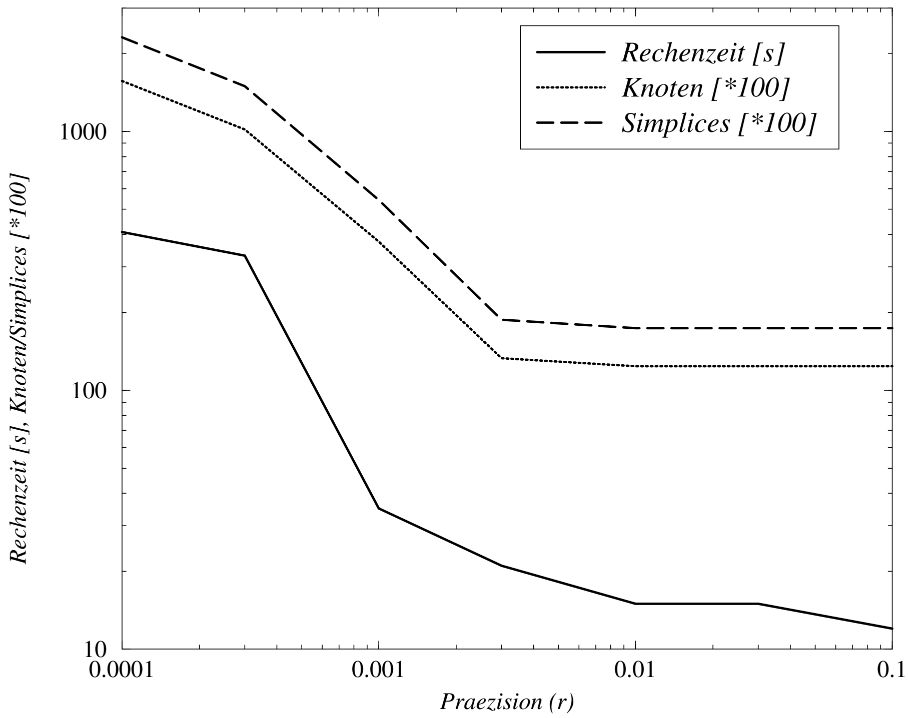


Abbildung 4.27: Rechenzeit und Speicheraufwand des Mehrgitterverfahrens gegen die erreichte Präzision. Als Beispielproblem wurde eine Ladungskonfiguration antiparalleler $q\bar{q}$-Strings genommen, mit bekanntem Verlauf des σ-Felds.

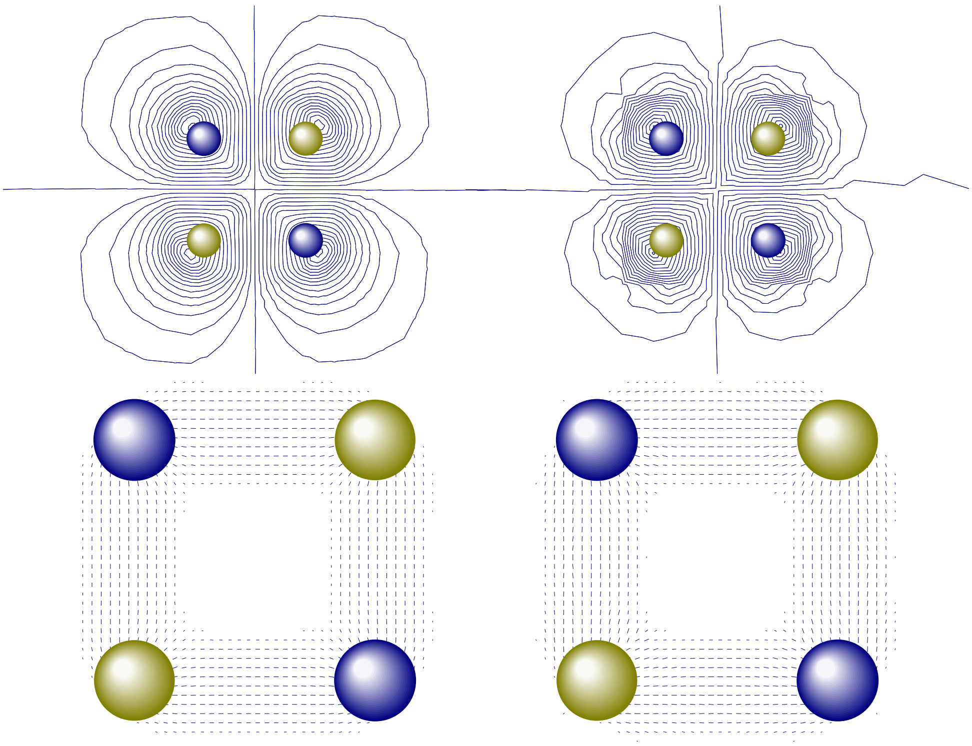


Abbildung 4.28: Qualität der Lösung des Testproblems (String-String-Konfiguration mit Stringlänge $4fm$ und Stringabstand $4fm$). In der linken Spalte ist die Lösungsfunktion $\phi^8(\vec{x})$ für den relativen Fehler $r = 0.0001$ der Lösung gezeigt, in der rechten Spalte das Gleiche für den relativen Fehler $r = 0.01$. Auch die Lösung auf der rechten Seite ist noch qualitativ richtig und recht brauchbar.

ringstmögliche Genauigkeitsgrenze gewählt, die nicht zu numerischen Instabilitäten führte. Dies kann wie folgt geschehen: da die Farbfelder zwischen Zeitschritten nicht miteinander kommunizieren (die Gaußgleichung wird in jedem Zeitschritt instantan gelöst), können die numerischen Fehler im Farbfeld von Zeitschritt zu Zeitschritt stark oszillieren. Dies führt zu einem numerischen Rauschen im Quellterm der σ-Feldgleichung, das umso stärker ist, je größer der Präzisionsparameter r der Farbfeldlösung ist.
Die für unsere Simulationsläufe meist benutzte Größe $r = 1\%$ wurde durch "trial and error" als guter Kompromiss zwischen Genauigkeit und Geschwindigkeit ermittelt. Es ist natürlich die Frage, ob bei so geringen Genauigkeitsansprüchen (1%) nicht doch ein simplerer Algorithmus gereicht hätte. Doch solch eine Frage lässt sich nach Meinung des Autors erst rückblickend beantworten. Folgendes soll aber schon hier zusammengefasst werden, teilweise im Vorgriff auf spätere Resultate:

- Das in MUESLEA implementierte Mehrgitterverfahren ist extrem schnell und trotzdem recht allgemein: die Spezifikation der Differentialgleichung und der Randbedingungen geschieht an einer einzigen Stelle *außerhalb* der MUESLEA-Bibliothek (nämlich in einer Instanz der Klasse PDEquation, die dann an den Lösealgorithmus übergeben wird).

- Die hohe Qualität der Lösung wird erkauft durch exzeptionell hohen Speicherplatzbedarf. Dieser ließe sich aber auf Kosten des Tempos verringern (z. B. durch Ausschalten des Matrixcachings).

- Die Simulation wird in den Hadronisierungsszenarios (siehe Kapitel 5) dadurch ausgebremst, dass zu jedem Zeitschritt ein Randwertproblem gelöst wird. Die Länge der Zeitschritte wiederum ist durch die Leapfrog-Methode der σ-Feldpropagation beschränkt.

- Die Suche nach der richtigen numerischen Methode hat einen Großteil der Arbeitszeit gekostet, die in diese Dissertation geflossen ist. Rückblickend ist jedoch einige Skepsis angebracht, ob die letztendlich verwendeten Verfahren wirklich das Optimum darstellen. Siehe hierzu auch den letzten Punkt der "Brainstorming"-Liste im Anhang 6.2.2.

4.4 Diskussion

Das σ-Feld unseres Simulationsprogramms "lebt" auf einem regelmäßigen Gitter, während das farbelektrostatische Potential ϕ^a in den Knoten eines adaptiven Simplexnetzes gespeichert ist. Für diese methodische Trennung gibt es folgende Argumente:

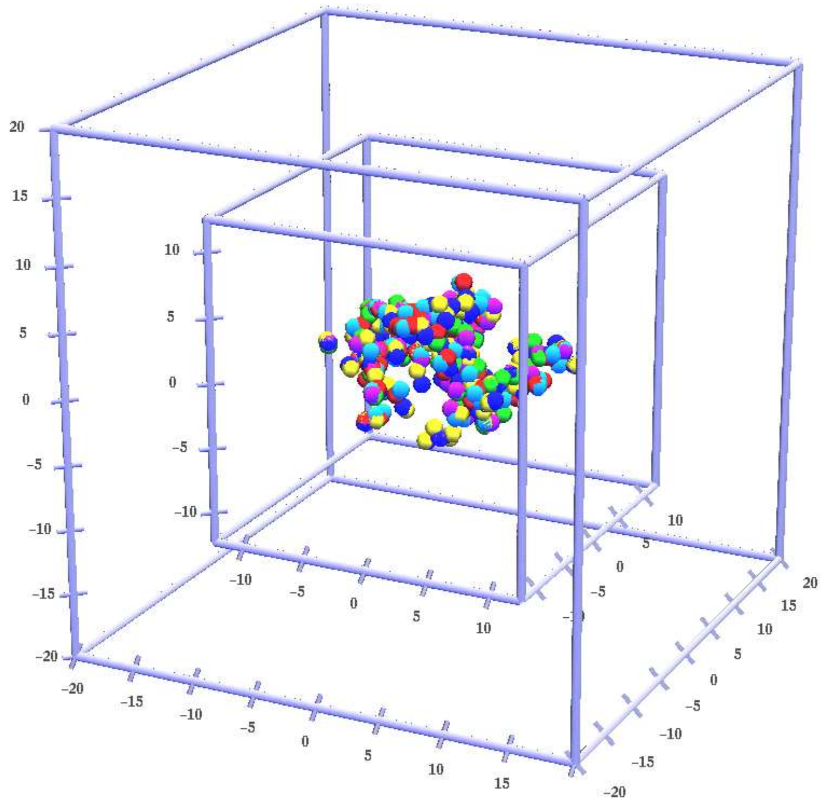


Abbildung 4.29: Unterschiedliche Gebietsgrößen für das Gitter des σ-Felds (kleiner Würfel) und das adaptive Knoten-/Simplexnetz des ϕ^a-Felds (großer Würfel). Da sich Farbladungen nur im σ-Feldgitter aufhalten können, bleiben sie immer in sicherem Abstand vom Rand des großen Würfels, jenseits dessen sich unphysikalische Spiegelladungen befinden.

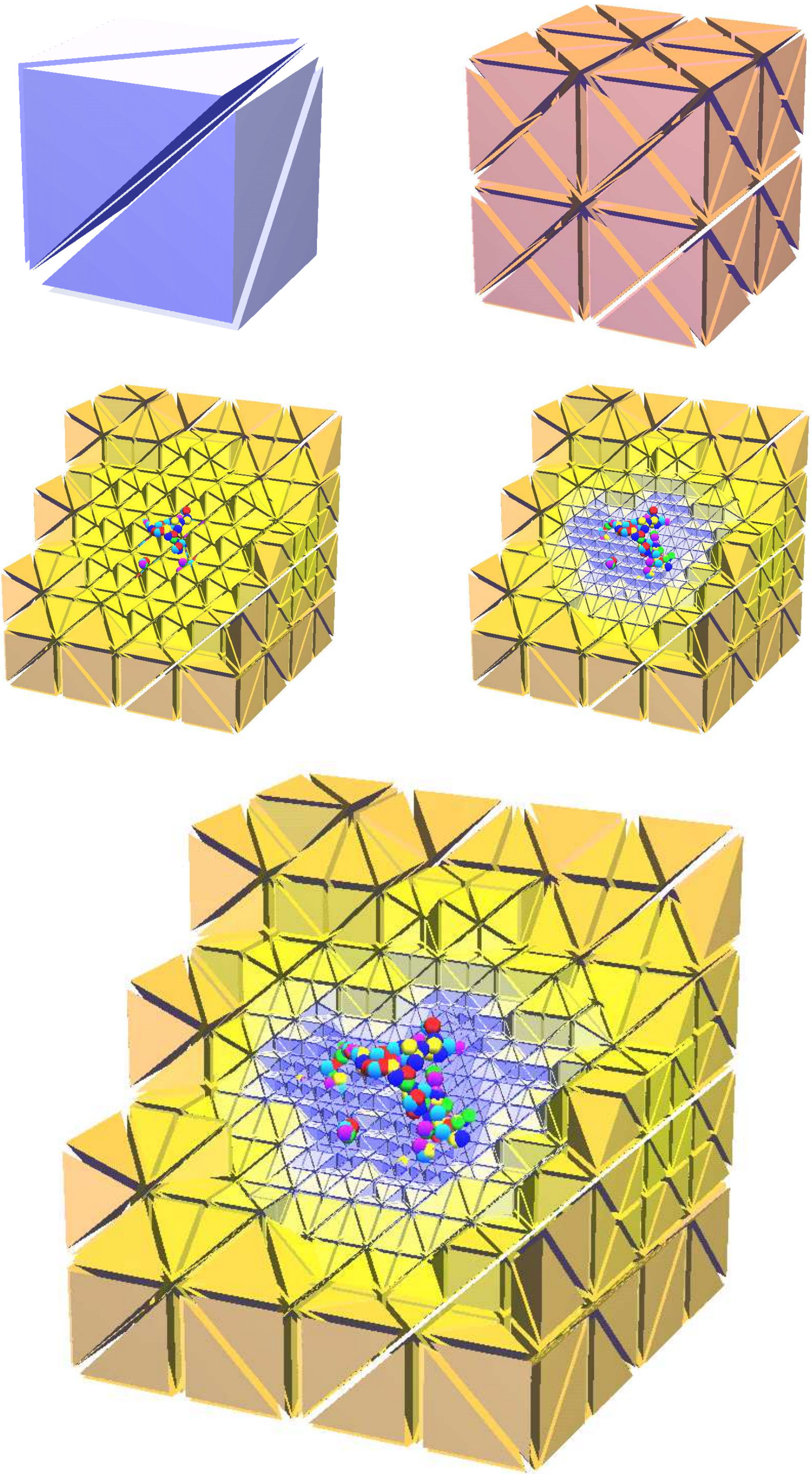


Abbildung 4.30: Simplexpartition des großen Würfels nach vielen Verfeinerungsschritten.

- Das σ-Feld dient physikalisch dazu, mittels $U(\sigma)$ einen Vakuumdruck zu simulieren, der dann über $\kappa(\sigma)$ an das farbelektrische Feld angekoppelt wird. Dies ist ein *lokaler* Effekt; Fernwechselwirkungen sind mit dem σ-Feld nicht zu vermitteln, da es so massiv ist. Die Leapfrogschritte auf einem kartesischen Gitter stellen eine robuste und kontrollierbare Methode zur dynamischen Simulation der σ-Wellengleichungen dar. Die Energieerhaltung von σ-Wellen ist (ohne Reibungsterm!) ausgezeichnet.

- Das Vektorfeld hingegen ist langreichweitig. Damit das Farbconfinement funktioniert, müssen sich auch weit entfernte Quark-Gluon-Cluster, die nur schwach farbgeladen sind, mit der korrekten Stringspannung anziehen. Das bedeutet, die Gaußgleichung muss sowohl lokal als auch auf größeren Längenskalen numerisch genau gelöst werden. Dafür ist ein Mehrgitterverfahren gut geeignet.

- *Spiegelladungen* können einige Sorgen bereiten: eine Farbladung sieht jenseits des Gebietsrands (mit Dirichletschen Randbedingungen $\phi^a \equiv 0$) eine virtuelle Bildladung (Spiegelladung). Falls sie dem Rand zu nahe kommt, kann sich zwischen der echten Farbladung und ihrer Bildladung ein String (Flussschlauch) ausbilden. Natürlich ist dies unphysikalisch, da in der Natur der Gebietsrand im Unendlichen liegt. Wir können das Bildladungsproblem aber elegant umgehen, indem wir verschiedene Gebietsgrößen zur Berechnung des σ-Feldes und der Farbfelder zugrundelegen. Von den Farbladungen verlangen[30] wir, dass sie stets im Inneren des kleineren σ-Feldgitters $\mathcal{G}$ bleiben. Dadurch halten sie immer einen gewissen Abstand vom Rand des Gebiets $\mathcal{H}$, in dem die Farbfelder berechnet werden. Typische Größen sind dabei $25fm$ Kantenlänge für $\mathcal{G}$ und $40fm$ Kantenlänge für $\mathcal{H}$, wodurch der Abstand einer Farbladung zu ihrer Spiegelladung stets $\geq 15fm$ bleibt. Abb. 4.29 zeigt die beiden Gebiete, wie sie ineinandergeschachtelt aussehen. Die adaptive Verfeinerung (Abb. 4.30) bewirkt dabei, dass der zusätzlich eingeführte Außenraum nicht viel Rechenzeit kostet; er wird praktisch nicht verfeinert (nur als Folge von Kompatibilitätsschnitten).

- Da das σ-Feld kurzreichweitig ist, gibt es keine Probleme mit unphysikalischen Bildladungen in der Nähe des σ-Gitterrandes. Die Behandlung des σ-Feldes auf dem kleineren Gitter ist damit berechtigt.

---

[30] Bei der Untersuchung von Gleichgewichtsszenarios erreichen wir dies, indem wir die Farbladungen vom Rand des σ-Gitters reflektieren lassen. Im Fall von Hadronisierungsszenarios hoffen wir, dass die Farbladungen sich nicht aus dem σ-Gitter entfernen, bevor sie in IWCs zu farblosen Hadronen konvertiert werden. Sollte doch einmal eine Farbladung gegen den σ-Gebietsrand stoßen, bleibt uns nichts anderes übrig, als sie zurück ins Innere zu reflektieren.

Leider benötigt CDM dennoch sehr lange Rechenzeiten für Hadronisierungsszenarien (siehe Kapitel 5). Die Ursache liegt in einem unglücklichen Zusammenwirken der Numerik: aus Stabilitätsgründen müssen wir den σ-Leapfrogzeitschritt sehr kurz (typischerweise $0.04fm$) machen. Das σ-Feld alleine wäre auch mit diesen kleinen Zeitschritten schnell zu propagieren, da die Zeitschritte nicht viel Rechenzeit verschlingen. Nun müssen wir aber zu jedem Zeitschritt das Gaußgesetz lösen. Dazu haben wir zwar eine extrem gut optimierte und spezialisierte Numerik (eben das adaptive Mehrgitterverfahren), dennoch dauert eine Lösung der ϕ^a-Gleichungen einige Minuten Rechenzeit. Einen möglichen Ausweg aus diesem Dilemma werden wir in Kapitel 6 diskutieren.

4.5 Clustererkennung und Hadronisierung in CDM

Der Phasenübergang von hadronischer Materie zum Quark-Gluon-Plasma führt zu einer viel größeren spezifischen Wärme, da im Plasma wesentlich mehr Freiheitsgrade involviert sind. Das ist in Gitterrechnungen klar erkennbar (Abb. 2.7). Die spezifische Wärme ist als Funktion der Temperatur am Phasenübergangspunkt sogar unstetig[31]. Dies ist ein von anderen Flüssig/Gas-Phasenübergängen altbekanntes Phänomen. Es liegt in der Natur des umgekehrten Phasenübergangs (Hadronisierung), dass diese zusätzlichen Freiheitsgrade wieder einfrieren.
Für unsere Simulation heißt das: um Quark-Gluon-Cluster zu Hadronen zu machen, müssen wir an irgendeinem Punkt des Programmlaufs innere Freiheitsgrade der Cluster vernachlässigen, . Die dabei vernichtete Information ist die gesamte innere Struktur der Cluster. Nur die nach außen wirksamen Größen (weiße Farbe, Baryonenzahl, Ladung, Masse, Impuls) bleiben erhalten. Nun ist bereits in Abschnitt 3.5 eine Klassifikation irreduzibler weißer Cluster (IWCs) von Quarks und Gluonen vorgestellt worden (Abb. 3.5). Die Klassifikation enthält zudem nur "wünschenswerte" Hadronen, nämlich baryonische, mesonische und rein gluonische Cluster, wie sie auch in der Natur vorkommen. Man bemerke, dass diese IWC-Klassifikation ein Resultat des Modells in abelscher Näherung ist, d. h. zum "Output" und nicht zum "Input" des Modells zählt.
Damit liegt die folgende Vorgehensweise für ein Hadronisierungsszenario auf der Hand: Man initialisiere ein Quark-Gluon-Plasma (Gas) und propagiere einfach die Modellgleichungen, ohne das Modelluniversum in irgendeiner Form zu "stören". Löst sich ein irreduzibles weißes Cluster

[31]Im Falle eines Phasenübergangs erster Ordnung. Gitterrechnungen lassen eine Stufe erkennen, doch ist diese stets ein wenig "ausgeschmiert". Dabei ist nicht klar, ob die scheinbare Stetigkeit auf die endliche Systemgröße (Gitter) oder auf einen Phasenübergang zweiter Ordnung zurückzuführen ist.

(IWC) vom Rest des Plasmas, so betrachte man dieses als Hadron, klassifiziere es nach Abb. 3.5, berechne die enthaltene Energie (auch Feldenergie) und den Impuls (auch Feldimpuls) und die Masse $M^2 = E^2 - \vec{P}^2$. Danach lösche man die Quarks und Gluonen dieses IWCs, beseitige die Spuren, die das Cluster in den diversen Feldern besaß (den *field imprint* des Clusters) und setze an die Stelle des IWCs ein "Hadron". Die Eigenschaften des Hadrons sind dabei in allen wesentlichen Erhaltungsgrößen (weiße Farbe, Baryonenzahl, Ladung, Masse, Impuls) identisch mit denen des vorherigen IWCs (unter Berücksichtigung der Feldenergie und des Feldimpulses). Andere Eigenschaften, wie innere Freiheitsgrade, höhere Farbmultipole etc. sind vernachlässigt worden.
Natürlich kann das entstandene Hadron nicht mehr wechselwirken, da es weiß ist und eventuelle höhere Farbmultipole bei der Hadronisierung vernichtet wurden. Unser Plan vernachlässigt also eventuelle Endzustandswechselwirkungen; wir erwarten aber sowieso nicht, dass diese durch ein einfaches klassisches Modell auf Quarkebene gut beschrieben werden können.
In CDM funktioniert die Clustererkennung im wesentlichen so: In regelmäßigen Abständen wird das gesamte Modelluniversum von der Funktion Cube::PlasmaClusters (cluster.cpp) nach "σ-Bags" durchsucht. Jeder Gitterpunkt[32] gehört danach zum Innenraum eines Bags, wenn dort $\sigma < \sigma_{\text{boundary}}$ gilt, ansonsten zum Außenraum[33]. Für jeden Bag wird eine Instanz der Klasse Cluster (cluster.h) erzeugt. Die Gitterpunkte werden von Cluster::MarkSites je nach ihrer Bagzugehörigkeit mit Nummern versehen (Außenraum, Bag 1, Bag 2, etc.). Alle Teilchen, die zu einem Bag gehören, werden in der Liste Cluster::particles vermerkt. Sind nach der "Clusterisierung" noch farbige Teilchen im Außenraum übrig, so kann dies nur zwei Ursachen haben: Entweder handelt es sich um Teilchen, die knapp außerhalb eines Bags liegen (der Bagrand bei $\sigma = \sigma_{\text{boundary}}$ ist ja etwas beliebig gewählt), oder es handelt sich um ein farbneutrales Cluster, dessen Partonen so dicht beieinander liegen, dass das Cluster praktisch keinen Feldeindruck hinterlässt. Die Routine Cube::PlasmaClusters erkennt auch diese Fälle und unternimmt sinnvolle Schritte, diese Teilchen Clustern zuzuordnen. Cube::PlasmaClusters gibt als Resultat eine Liste von Cluster-Instanzen zurück.
Die zurückgegebene Liste kann, falls gewünscht, von Cube::RemoveAllIrreducibleClusters (analyze.cpp) nach IWCs durchsucht werden. Diese Routine nimmt alle oben beschriebenen Schritte vor, um die gefundenen IWCs zu hadronisieren. Sie gibt eine Liste von HadronizedCluster-Instanzen zurück, die die Hadronen repräsentieren.

[32] Gemeint ist das Gitter, auf dem das σ-Feld gespeichert ist.

[33] Die Clusterroutine ließe sich auch ohne weiteres für Perkolationsstudien einsetzen; die verwendeten Algorithmen sind für diesen Fall vorbereitet.

KUHN: *“Was [the discovery of the positron] a great vindication?”*
DIRAC: *“Yes.”*
KUHN: *“Does that sort of event generate great immediate excitement and satisfaction?”*
DIRAC: *“I don’t think it generated so much satisfaction as getting the equations to fit.”*
nach Kragh [69]

5 Resultate

In diesem Kapitel sollen die Forschungsergebnisse vorgestellt werden, die als Teil dieser Dissertation in [36] publiziert wurden. Zunächst muss aber das Modell vervollständigt werden: es wurde bisher wenig über unsere speziellen Wahl aller CDM-Modellparameter gesagt, was im folgenden nachgeholt werden soll.

5.1 Wahl der Modellparameter

Wir wählen für all die verschiedenen Teilchensorten, sowohl Quarks als auch Gluonen, die gleiche, feste Ladungsverteilung nach (3.13),

$$\rho_N(\vec{x} - \vec{x}_k(t)) \;=\; \left(\frac{1}{2\pi r_0^2}\right)^{3/2} e^{-(\vec{x}-\vec{x}_k(t))^2/2r_0^2} \tag{5.1}$$

mit dem (willkürlichen) Radius $\sqrt{\langle \vec{x}^2 \rangle} = \sqrt{3}\, r_0 = 0.7fm$, der an den Mittelwert der Radien des Nukleons ($0.83fm$) und des Pions ($0.66fm$) angelehnt ist. Das chromodielektrische Modell besitzt in diesem Fall ein besonders einfaches “Hadronenspektrum”. Kalte (nicht angeregte) Hadronen sind weiße (farbneutrale) Kombinationen farbiger Teilchen, die am gleichen Raumpunkt zentriert sind und sich relativ zueinander nicht bewegen. Da alle farbigen Teilchen exakt die gleiche Form (5.1) der Farbladungsverteilung besitzen, wobei die Summe der Gesamtladungen verschwindet, ist für unser kaltes Hadron die lokale Farbladungsdichte überall gleich Null. Daher verschwinden für diese Konfiguration alle Farbfelder im gesamten Raum, und da unsere Lagrangedichte keine direkte Kopplung der Teilchen an das σ-Feld enthält, nimmt auch das σ-Feld überall seinen Vakuumwert an. Damit ist die Feldenergie gleich Null, und die Masse des kalten Hadrons einfach gleich der Summe der Massen seiner Konstituenten. Es handelt sich bei unserer Version des CDM also um ein *Konstituentenquarkmodell* der Hadronen. Solche Konstituentenquarkmodelle können das beobachtete Hadronenspektrum nicht besonders gut wiedergeben. Erstens sind die Spektren eines klassischen Modells niemals diskret (auf unser kaltes Hadron baut ein kontinuierliches Spektrum klassischer Anregungen auf). Zweitens ist es in einem Modell ohne jede Grundzustandswechselwirkung unmöglich, die Konstituentenquarkmassen so zu wählen, dass gleichzeitig die extrem geringe Masse des Pions und die Masse der Nukleonen gut getroffen werden. Da wir Spin- und Isospinfreiheitsgrade (und -kopplung) vernachlässigen, ist es uns im Rahmen dieses Modells auch nicht möglich, den π-ρ- oder N-Δ-Massenunterschied zu reproduzieren. Wir haben an dieser Stelle zwei "philosophische" Optionen:

- Wir geben deshalb von vorneherein den Versuch auf, Pion- oder Nukleonmassen beschreiben zu können, und konzentrieren uns *in jedem Flavorkanal auf die jeweils leichteste Resonanz über dem Grundzustand*.

- Wir vernachlässigen das Pion, dessen geringe Masse durch seine Rolle als Goldstoneboson der gebrochenen chiralen Symmetrie erklärt werden kann, die chirale Symmetrie aber nicht im Modell enthalten ist. Bei den Baryonen hingegen mitteln wir Spin- und Isospinkanäle aus. Im Falle des Nukleons hat die erste Resonanz, das Δ, durch seinen Spin $3/2$ und Isospin $3/2$ ein viermal größeres statistisches Gewicht, also muss korrekt $\frac{1}{5}(N + 4\Delta)$ gemittelt werden.

Praktisch gehen wir wie folgt vor: Um im Rahmen des Möglichen gute hadronische Massenskalen zu erhalten, wählen wir eine Masse von $400MeV$ für das leichte Quark (*up* und *down*), $550MeV$ für das *strange*-Quark, $1500MeV$ für das *charm*-Quark und $700MeV$ für das Gluon. Das liefert uns eine Masse von $800MeV$ für das ρ-Meson und $1200MeV$ für das Nukleon bzw Δ. Weitere Massen sind in Tab. 5.1 aufgeführt. Man erkennt eine recht gute Übereinstimmung zwischen Modell- und Messwerten. Dies stellt in diesem Kontext eine interessante Beobachtung dar, die wir aber nicht weiter verfolgen wollen. Die Beschreibung des unteren hadronischen Spektrums kann nicht ernsthaft unser Ziel sein; wir befassen uns in dieser Arbeit mit der Hadronisierung, also der Plasma- und Hadronenphysik nahe des Confinementphasenübergangs, und haben ein Modell gewählt, das vor allem zur Beschreibung des Farbconfinements geeignet ist.
Die fünf Parameter des Potentials $U(\sigma)$, das als Polynom vierten Grades angesetzt wird, und die dielektrische Funktion $\kappa(\sigma)$ sind schwieriger

Tabelle 5.1: *Effektive Massen von Quarks und Gluonen und korrespondierende Hadronenmassen [4]. Alle Massen sind in* MeV *angegeben. Das Symbol* q *bezeichnet ein up- oder down-Quark,* s *ein strange-Quark,* g *ein Gluon.*

Teilchen	Modellmasse	Exp. Masse	Fehler
q	400	-	-
s	550	-	-
c	1500	-	-
g	700	-	-
$\frac{1}{2}(N+\Delta)$ (qqq)	1200	1086	+10.5%
$\frac{1}{5}(N+4\Delta)$ (qqq)	1200	1174	+2.2%
Δ (qqq)	1200	1232	-2.5%
Y (qqs)	1350	1385	-2.5%
Ξ (qss)	1500	1530	-2.0%
Ω (sss)	1650	1672	-1.3%
ρ $(q\bar{q})$	800	770	+3.9%
ω $(q\bar{q})$	800	782	+2.3%
K^* $(q\bar{s})$	950	892	+6.5%
Φ $(s\bar{s})$	1100	1020	+7.8%
gg	1400	1400 ?	-

festzulegen, da kein bekanntes physikalisches Teilchen mit dem σ-Feld assoziiert ist. Interpretiert man σ-Wellen als reine Luebälle mit einer Masse von etwa $1400MeV$, so fixiert dies

$$U''(\sigma_{vac}) \;=\; (1400MeV)^2. \tag{5.2}$$

Die Bagkonstante (phys. Vakuumdruck)

$$U(0) \;=\; B \;=\; (150MeV)^4 \tag{5.3}$$

wird vergleichbar den typischen Werten im MIT-Bagmodell gewählt, und

$$\sigma_{vac} \;=\; 0.31 fm^{-1} \tag{5.4}$$

ist eine willkürliche Festlegung, die dazu führt, dass die Oberflächenspannung hadronischer Bags relativ niedrig ist — das ist numerisch stabiler und damit leichter handhabbar; "Messwerte" für die Oberflächenspannung hadronischer Bags existieren sowieso nicht, so dass wir aus dem Experiment keinen diesbezüglichen Anhaltspunkt für die Wahl von U haben. Die genannten Gleichungen, gemeinsam mit den Bedingungen

$$U'(0) \;=\; 0 \tag{5.5}$$

$$U(\sigma_{\rm vac}) \;=\; U'(\sigma_{\rm vac}) \;=\; 0 \tag{5.6}$$

fixieren die Form des $U(\sigma)$-Potentials vollständig. Wie in Abb. 5.1 zu sehen, besitzt U ein Intervall negativer Krümmung, eine Eigenschaft, die

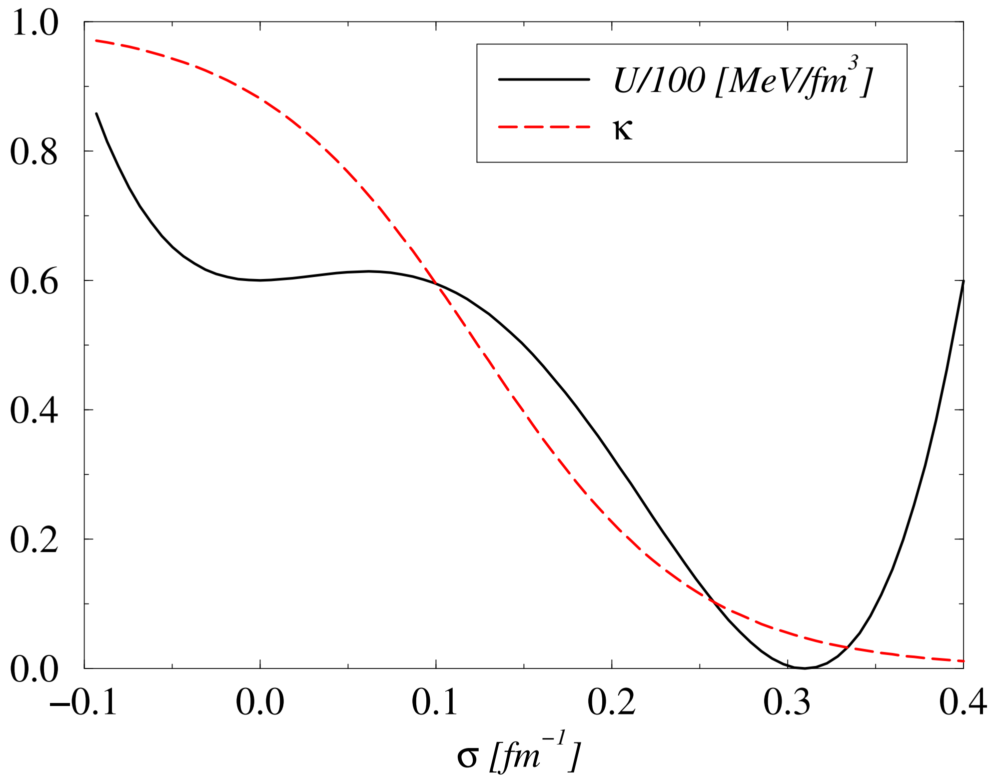


Abbildung 5.1: *Die Funktionen $U(\sigma)$ und $\kappa(\sigma)$, wie sie in unsere Simulation eingehen.*

relevant für die Existenz eines Phasenübergangs ist[1]. Unsere konkrete Wahl von U ist auch motiviert durch die früheren Gießener Studien der solitonischen Zustände im Modell [32, 33]. Wir benutzen letztlich

$$U(\sigma) \;=\; B + a\sigma^2 + b\sigma^3 + c\sigma^4 \tag{5.7}$$

mit $a = 6.184 fm^{-2}$, $b = -80.72 fm^{-1}$, $c = 163.1$, $B = (150 MeV)^4$.
Die konkrete Form der dielektrischen Funktion $\kappa(\sigma)$ scheint nicht besonders wichtig für unsere Simulationsresultate zu sein. Wie im Modellkapitel 3 bereits beschrieben, ist der einzige Zweck dieser Funktion die Kopplung des Vakuumdrucks $U(\sigma)$ an den Farbfluss eines Flussschlauchs. Um dies zu erreichen, muss die κ-Funktion einen Wert $\ll 1$ nahe $\sigma \approx \sigma_{\rm vac}$ annehmen und von dort schnell zu $\kappa(0) \approx 1$ ansteigen, um unsere Interpretation von $U(0) = B$ als Vakuumdruck zu unterstützen. Wir benutzen

[1]Bei Vorhandensein einer direkten Quark-σ-Kopplung bewirkt die Nichtkonvexität des U-Potentials eine Nichtkonvexität der Helmholtzschen freien Energie $F(\rho_S)$, was zu einem Phasenübergang erster Ordnung bei endlicher skalarer Quarkdichte ρ_S führt. In der vorliegenden Arbeit ist die Relevanz dieser Nichtkonvexität für den Phasenübergang unklar, auch wenn der Autor vermutet, dass nur mit einer solchen U-Form ein Deconfinement-Phasenübergang erster Ordnung im Modell enthalten ist.

die allgemeine Form

$$\kappa(\sigma) = \frac{1}{\exp\left(\alpha\left(\frac{\sigma}{\sigma_{vac}} - \beta\right)\right) + 1} \tag{5.8}$$

mit $\alpha = 7$ und $\beta = 0.4$. Es ist für die Stabilität der Numerik essentiell, dass nicht wirklich $\kappa(\sigma_{\text{vac}}) = 0$ gilt, sondern dass diese Bedingung "aufgeweicht" wird — für unseren Parametersatz gilt $\kappa(\sigma_{\text{vac}}) \approx 0.015$. Für die Phänomenologie hingegen ist dies nebensächlich; wir werden in Abschnitt 5.4 in Bildern von Flussschläuchen sehen, dass der farbelektrische Fluss auch so sehr schön zum Schlauch gebündelt wird. Und nur dies ist wesentlich: damit die Vektorfeldenergie außerhalb von σ-Bags vernachlässigbar klein bleibt, muss nur $\kappa_{\text{vac}} \ll 1$ gelten.
Die Funktionen U und κ sind in Abb. 5.1 gemeinsam dargestellt.
Die starke Kopplungskonstante $\alpha_S = g_v^2/4\pi$ wird so gewählt, dass die $q\bar{q}$-Stringspannung etwa $1 GeV/fm$ beträgt, der Wert, der in Kapitel 2 aus der Quarkoniumspektroskopie abgeleitet wurde (siehe auch [41]). Wir wollen das Resultat aus Abschnitt 5.3 vorwegnehmen und sagen, dass ein Wert $\alpha_S = 2$ in einem befriedigenden Wert für die numerische Stringspannung resultiert.
Modellparameter sind übrigens im Programmcode CDM allesamt in der Klasse ModelParameters (siehe den Klassenkonstruktor im File model.cpp) untergebracht. Mathematische Konstanten und Naturkonstanten findet man in trans.h.

5.2 Ein einzelnes Quark

Bevor Hadronisierungsszenarien oder andere komplizierte Konfigurationen betrachtet werden, wollen wir nachweisen, dass die in Kapitel 3 erläuterten Confinementeigenschaften des Modells auch von die Numerik reproduziert werden. Wir untersuchen zu diesem Zweck eine Konfiguration mit einer *einzelnen Farbladung*. Eine solche Ladungskonfiguration kann, wie wir in Abschnitt 3.4 (auf Seite 50) nachgewiesen haben, im Modelluniversum eigentlich nicht mit finiter Feldenergie existieren. Aus zwei Gründen ist es im Simulationsprogramm CDM dennoch möglich, finite Feldenergien für zu berechnen. Erstens ist mit (5.8) die Bedingung $\kappa(\sigma_{\text{vac}}) = 0$ für die numerische Behandlung "aufgeweicht" worden; es gilt für unseren Standardparametersatz ungefähr $\kappa(\sigma_{\text{vac}}) = 0.015$. Zweitens besteht der simulierte Raum nur aus einem endlichen Würfel; der Beweis unendlicher Feldenergie in Abschnitt 3.4 beruht aber wesentlich auf einem Grenzübergang ins Unendliche.
Der Leser sollte diesen Abschnitt daher als Test der Numerik auffassen: *ein Einzelquarkszenario ist in der QCD und im chromodielektrischen Modell unphysikalisch*. Die folgende Untersuchung kann daher *per se* keinen physikalischen Inhalt besitzen.

Nach dieser Vorrede betrachten wir die Einzelquarkkonfiguration im Rahmen der imperfekten Numerik und erwarten eine endliche, wenn auch große, Feldenergie dafür: Eine grobe Abschätzung nach oben erhält man, wenn man für die Felder im Außenraum ($r > r_0 = 0.7 fm$) der Ladungsverteilung

$$\begin{aligned}\sigma &\equiv \sigma_{\text{vac}}\\ D^a &= \frac{Q^a}{4\pi r^2}\end{aligned}$$

annimmt. Vernachlässigt man die Feldenergie innerhalb der Ladungsverteilung vernachlässigen, so folgt aus (3.47) für die gesamte Feldenergie

$$\begin{aligned}E_{\text{Feld}} &= \frac{Q^2}{4\pi}\int_{r_0}^{\infty}\frac{dr}{2r^2\kappa} + \underbrace{4\pi\int_{r_0}^{\infty} dr\, r^2 U(\sigma)}_{=0 \text{ da } \sigma\equiv\sigma_{\text{vac}}}\\ &= \frac{Q^2}{2\cdot 4\pi\kappa_{\text{vac}}}\int_{r_0}^{\infty}\frac{dr}{r^2}\\ &= \frac{Q^2}{2\cdot 4\pi\kappa_{\text{vac}} r_0}\\ &= \frac{4}{3}\frac{\alpha_S}{2\kappa_{\text{vac}} r_0}\\ &\approx 127 fm^{-1}\\ &\approx 25 GeV. \end{aligned} \tag{5.9}$$

Dies stellt eine *obere Abschätzung* dar, weil $\sigma \equiv \sigma_{\text{vac}}$ sicher nicht eine Feldkonfiguration minimaler Energie darstellt; nach Relaxation wird die Energie der Einzelquarkkonfiguration niedriger sein. Allerdings kann (5.9) als Anhaltspunkt für die späteren Resultate dienen.

Von "großer" Feldenergie kann man nur sprechen, wenn man eine Vergleichsgröße zur Hand hat. Wir betrachten daher statt unserer dielektrischen Funktion $\kappa(\sigma)$ aus (5.8) allgemeiner eine Klasse von dielektrischen Funktionen der Form

$$\kappa(\sigma) = \kappa_0 + (1-\kappa_0)\frac{1}{\exp\left(\alpha\left(\frac{\sigma}{\sigma_{vac}} - \beta\right)\right)+1}. \tag{5.10}$$

Der Grenzfall $\kappa_0 = 0$ reproduziert das gewohnte (5.8) mit $\kappa(\sigma_{\text{vac}}) = 0.015$, das wir in späteren Simulationsruns stets verwenden werden. Im Grenzfall $\kappa_0 = 1$ gilt hingegen $\kappa(\sigma) \equiv 1$, was die Farbfeld-σ-Kopplung völlig abschaltet. Für einige Zwischenwerte wurde nun die Einzelquarkkonfiguration initialisiert und die Felder relaxiert (unter Abdämpfung der Wellen propagiert, bis sie ihren Grundzustand unter Beachtung des Gaußgesetzes erreicht haben.) In Abb. 5.2 ist die resultierende Feldenergie als

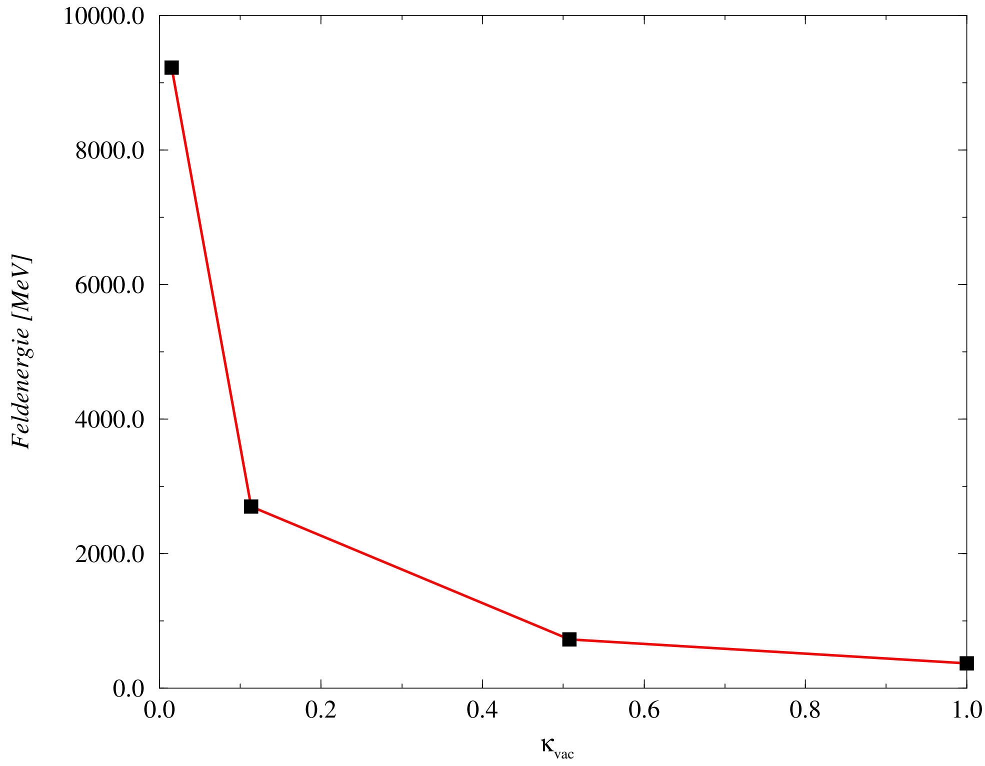


Abbildung 5.2: Feldenergie eines einzelnen Quarks im Simulationsuniversum.

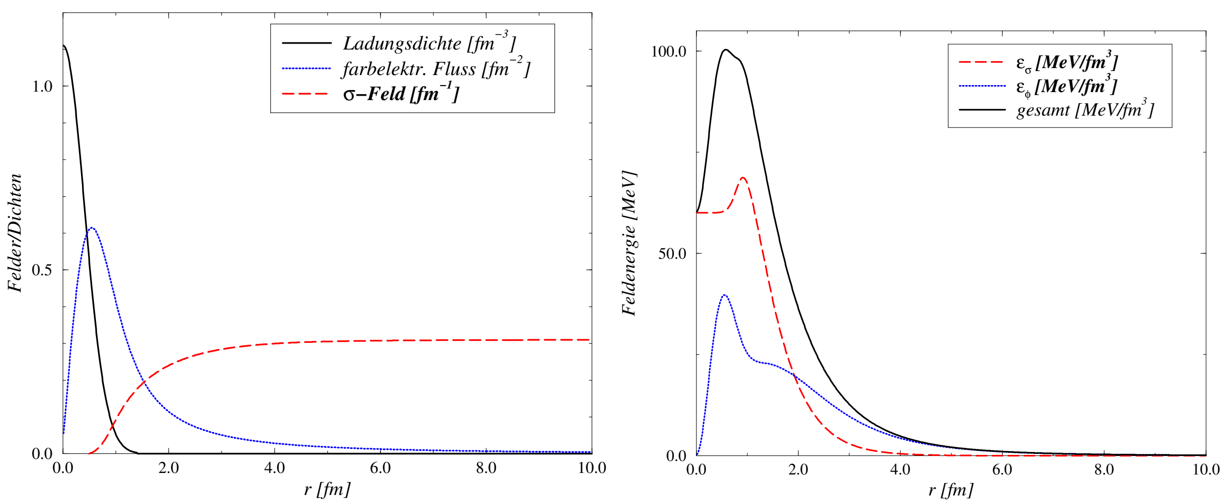


Abbildung 5.3: Feldstärken und dazugehörige Energiedichten der Einzelquarkkonfiguration für $\kappa_{\rm vac} = 0.015$.

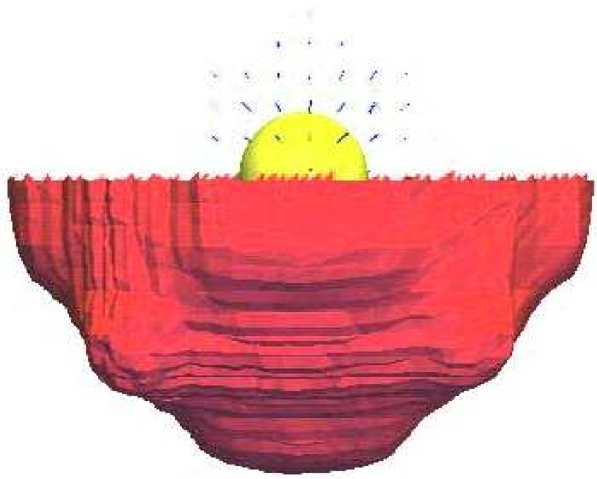


Abbildung 5.4: Feldkonfiguration in der dreidimensionalen Simulation. Die leichte Abweichung vonder sphärisch symmetrischen Form ist auf den numerischen Fehler (Wechselwirkung mit unphysikalischen Bildladungen) zurückzuführen. Die Äquipotentialfläche markiert $\sigma = 0.95\sigma_{\text{vac}}$, und die Kugel in der Mitte (Radius $0.7fm$) veranschaulicht den Ladungsradius des Quarks.

Funktion von κ_0 geplottet. Man erkennt, wie die Feldenergie zu $\kappa_0 = 0$ hin steil ansteigt. Die Energie von etwa $9.5GeV$ für das einzelne Quark genügt für die Simulation von Hadronisierungsvorgängen (bei Temperaturen $\ll 1GeV$) vollauf, so dass wir mit unserer Parameterwahl soweit zufrieden sein können. Der Wert von etwa $9.5GeV$ für die Feldenergie liegt in derselben Größenordnung und ist verträglich mit der errechneten oberen Abschätzung $25GeV$ aus (5.9). Abb. 5.3 zeigt schließlich die Feldstärken und Energiedichten der (sphärisch symmetrischen) Einzelquarkkonfiguration als Funktion von r, und in Abb. 5.4 ist das Ergebnis der dreidimensionalen Rechnung nochmals veranschaulicht. Es soll aber betont werden, dass eine Ladungskonfiguration mit nichtverschwindender Gesamtfarbladung unphysikalisch ist. Der Leser nehme also Abb. 5.2 nicht zu ernst.

5.3 Stringspannung

In Abb. 5.5 sind die Feldkonfigurationen der CDM-Flussschläuche dargestellt. Teilbild (a) zeigt den farbelektrischen Fluss $\vec{D}^8 = -\kappa(\sigma)\nabla\phi^8$ zwischen einem blauen Quark und einem antiblauen Antiquark (in der bekannten Nomenklatur nach Abb. 3.2). Das Vektorfeld ist in einer Schnittebene gezeigt, die die Zentren beider Quarks enthält. Die Kugeln an beiden Enden des Flussschlauchs repräsentieren die Quarks; ihr Radius gleicht dem mittleren Radius (RMS) der Farbladungsverteilungen $(0.7fm)$. Deutlich zu erkennen ist der parallele Verlauf der Farbfeldli-

nien im Flussschlauch, der bei einer Stringkonfiguration auch erwartet wird. Nach der Klassifikation irreduzibler weißer Cluster (IWCs, siehe Abb. 3.5) stellt dieser Flussschlauch ein angeregtes Meson des Typs $q\bar{q}$ dar.
Teilbild (b) der Abb. 5.5 zeigt das σ-Feld für die gleiche Farbladungskonfiguration und die gleiche Schnittebene. Äquipotentiallinien sind bei σ-Werten von $0fm^{-1}$, $0.1fm^{-1}$, $0.2fm^{-1}$ und $0.3fm^{-1}$ (von innen nach außen) gezeichnet; das ganz außen liegende physikalische Vakuum hat den σ-Wert $\sigma_{vac} \approx 0.31fm^{-1}$. Der Leser sei in diesem Kontext daran erinnert, dass der einzige Quellterm des σ-Felds durch $\kappa'(\sigma)F^2/4$ gegeben ist. Das Vektorfeld aus Teilbild (a) hinterlässt einen Eindruck im σ-Feld, der sich nicht ganz über die Quarks erstreckt. Man kann in den Teilbildern (a) und (b) gut erkennen, wie das äußere nichtperturbative Vakuum $(\sigma = \sigma_{\text{vac}})$ ins perturbative Vakuum $(\sigma \approx 0)$ übergeht. Dieser Übergang muss geschehen, um den vom Gaussgesetz geforderten farbelektrischen Fluss $\vec{D}^a = \kappa\vec{E}^a$ vom Quark zum Antiquark zu ermöglichen.
Fügen wir der Lagrangedichte einen direkten Quark-σ-Kopplungsterm hinzu (diskutiert in Abschnitt 3.8), wobei wir als Kopplungskonstante den Wert $g_S = 8$ von [32, 34] übernehmen, und approximieren wir $\langle\bar{q}q\rangle$ durch die Quarkzahldichte (3.13), so erhalten wir einen σ-Feldeindruck des Flussschlauchs, wie er in Teilbild (c) dargestellt ist. Dieser Graph weicht nur wenig von (b) ab, weil die Quark-σ-Kopplung überall bis auf die Enden des Flussschlauchs verschwindet.
Teilbild (d) schließlich zeigt das σ-Feld für einen gluonischen Flussschlauch. Die Gesamtfarbladung (definiert durch $\sqrt{(q^3)^2 + (q^8)^2}$) eines Gluons ist in unserem Modell um einen Faktor $\sqrt{3}$ größer als die Farbladung eines Quarks (siehe Abb. 3.2). Dadurch ist der farbelektrische Fluss um den gleichen Faktor vergrößert, was zu einem größeren transversalen Innendruck des Strings führt und damit zu einem größeren Durchmesser des Flussschlauchs. In diesem Bild haben wir Gluonen mit den Ladungen $r\bar{g}$ und $g\bar{r}$ benutzt. Nach der IWC-Klassifikation stellt dieser Flussschlauch einen angeregten Glueball des Typs gg dar.
Man sollte aber hinter diesen Abb. 5.5 nicht zu viel Physik vermuten; QCD-Flussschläuche dieser Länge $(4fm)$ würden sehr schnell durch $q\bar{q}$-Paarbildung zerfallen.
Die *Stringspannung* (Flussschlauchspannung) ist, wie in Kapitel 2 erläutert, für die grobe Form des hadronischen Spektrums in mittleren Energiebereichen verantwortlich. Auch ist die Stringspannung eine zentrale Größe, wenn es um das Studium der Hadronisierung geht — immerhin gibt es sogar Hadronisierungsmodelle [13], die auf Strings als elementaren Bausteinen beruhen, wobei die Spannung fast das einzige physikalische Merkmal dieser Strings ist (neben Ladung und Flavors).
Die CDM-Modellparameter sind so gewählt, dass der phänomenologische Wert der Quark-Antiquark-Stringspannung von $1GeV/fm$ im Modell reproduziert wird. Das ist in Abb. 5.6 demonstriert. Der gluonische

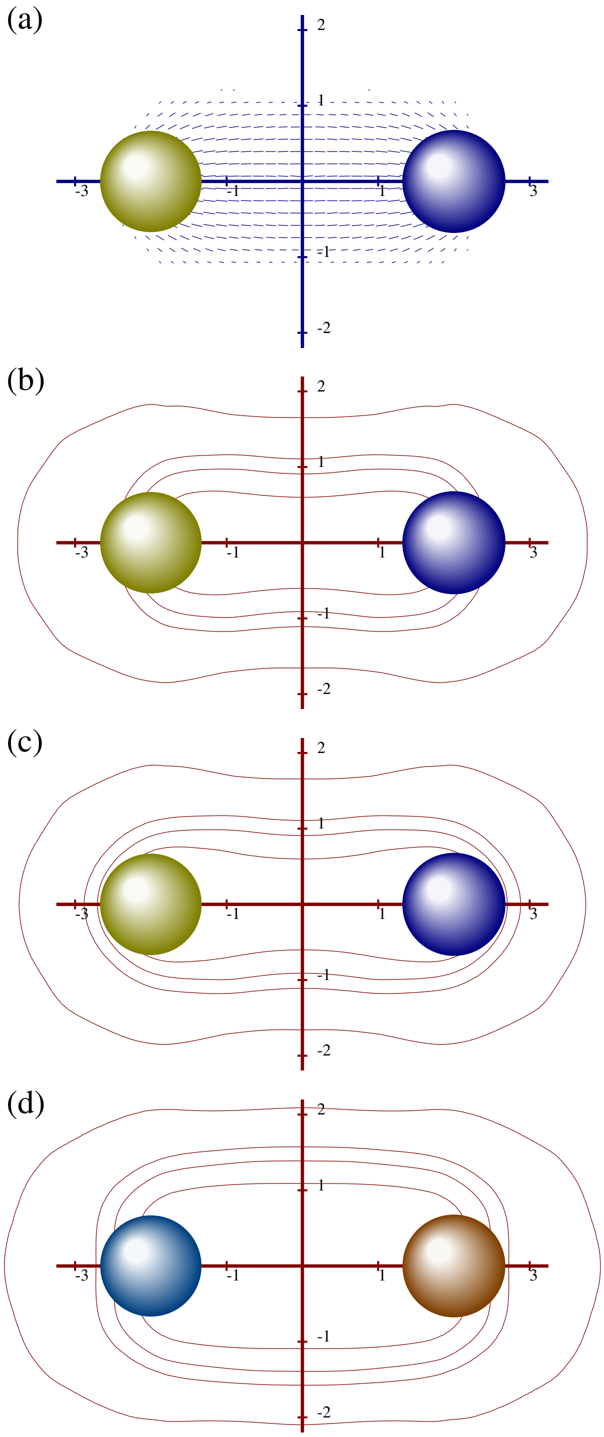


Abbildung 5.5: (*a*) *Vektorfeld eines Quark-Antiquark-Flussschlauchs der Länge* $4fm$. *Die Energiedichte des gezeigten Vektorfeldes beträgt etwa* $0.3GeV/fm^3$. *Die konkrete Farbwahl (blau-antiblau) spielt für dieses Resultat keine Rolle.* (*b*) σ-*Feld des gleichen Flussschlauchs.* (*c*) *das gleiche Bild, modifiziert durch einen zusätzlichen skalaren Kopplungsterm* (3.57) *mit* $g_S = 8$. (*d*) σ-*Feld eines gluonischen Strings. Äquipotentiallinien sind durch* $\sigma = 0fm^{-1}$, $0.1fm^{-1}$, $0.2fm^{-1}$, *und* $0.3fm^{-1}$ *definiert* ($\sigma_{vac} \approx 0.31fm^{-1}$). *Die Striche auf den Achsen sind in Abständen von* $1fm$ *gezeichnet.*

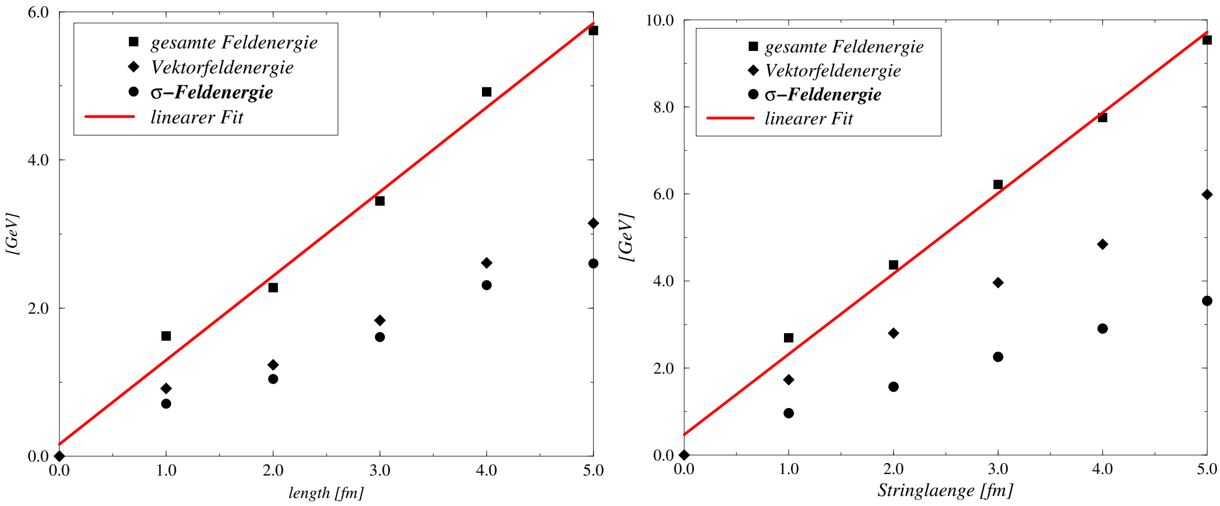


Abbildung 5.6: Die Flussschlauchenergie (Beiträge des σ- und Farbfelds) gegen den Partonenabstand. Der linke Plot gilt für einen Quark-Antiquark-String, der rechte für einen Gluon-Gluon-String. Die Steigung der Regressionsgeraden beträgt im $q\bar{q}$-Fall (links) $1.1GeV/fm$, was in etwa der bekannten hadronischen Stringkonstanten entspricht. Im gg-Fall (rechts) finden wir $1.8GeV/fm$. Diese höhere mechanische Spannung des gluonischen Strings im Vergleich zum Quarkstring ist letztlich auf die um einen Faktor $\sqrt{3}$ erhöhte Gesamtfarbladung $\sqrt{(q^3)^2 + (q^8)^2}$ der Gluonen zurückzuführen.

String hat mit $1.8GeV/fm$ eine ungleich höhere mechanische Spannung, was durch die höhere Gesamtfarbladung der Gluonen zurückzuführen ist.

5.4 String-String-Wechselwirkung

Das Simulationsprogramm CDM kann auch benutzt werden, um eine mögliche *Wechselwirkung zwischen Strings* zu untersuchen [34]. Das σ-Feld paralleler und antiparalleler Konfigurationen zweier Flussschläuche ist in Abb. 5.7 geplottet. Äquipotentiallinien sind auch hier wieder durch $\sigma = 0fm^{-1}$, $0.1fm^{-1}$, $0.2fm^{-1}$, und $0.3fm^{-1}$ definiert ($\sigma_{vac} \approx 0.31fm^{-1}$). Im letzten Bild der linken Spalte von Abb. 5.7 ist zu bemerken, dass die vollständig fusionierten, parallelen Flussschläuche einen im Vergleich zum einzelnen String wesentlich breiteren Abdruck im σ-Feld besitzen; dies liegt am stärkeren transversalen Druck der gebündelten Feldlinien zweier $q\bar{q}$-Paare und ist analog zum Bild des gluonischen Strings zu verstehen (Abb. 5.5 (d)). In der rechten Spalte fällt der charakteristischen "string flip" auf: sobald die Entfernung der antiparallelen Strings geringer wird als die Stringlänge ($4fm$), ist ein Umklappen der Strings in die orthogonale Richtung energetisch von Vorteil. Die Konfiguration niedrigster Energie (die die Lösung der statischen Feldgleichungen darstellt) ist die, bei der nur die benachbarten Quarks durch Flussschläuche verbunden sind.
Abb. 5.8 zeigt die resultierenden String-String-Wechselwirkungspotentiale. Im parallelen Fall (Abb. 5.8 links) weicht das Potential wenig von einem konstanten Wert ab, wobei der leicht attraktive Teil durch σ-Bagfusion und der repulsive Kern durch Coulombabstoßung der ausgedehnten Ladungen erklärt werden kann. Das Potential der antiparallelen Konfiguration (Abb. 5.8 rechts) zeigt das erwartete lineare (stringartige) Verhalten, sobald die Flussschläuche bei geringer Entfernung in die orthogonale Richtung übergesprungen sind (bei Entfernungen kleiner $4fm$).

5.5 Hadronisierung

Das beschriebene Simulationswerkzeug erlaubt uns, der mikroskopischen Dynamik eines Hadronisierungsszenarios im Rahmen des chromodielektrischen Modells (CDM) zu folgen. Indem wir uns auf wenige Observablen beschränken, erhoffen wir einen qualitativen Einblick. Man darf einem einfachen klassischen Modell nicht zu viele Observable abverlangen, sonst endet man leicht bei der physikalischen Interpretation von Modellartefakten. Im folgenden stellen wir zunächst die initialisierten Szenarien vor. Danach führen wir einige Observablen ein

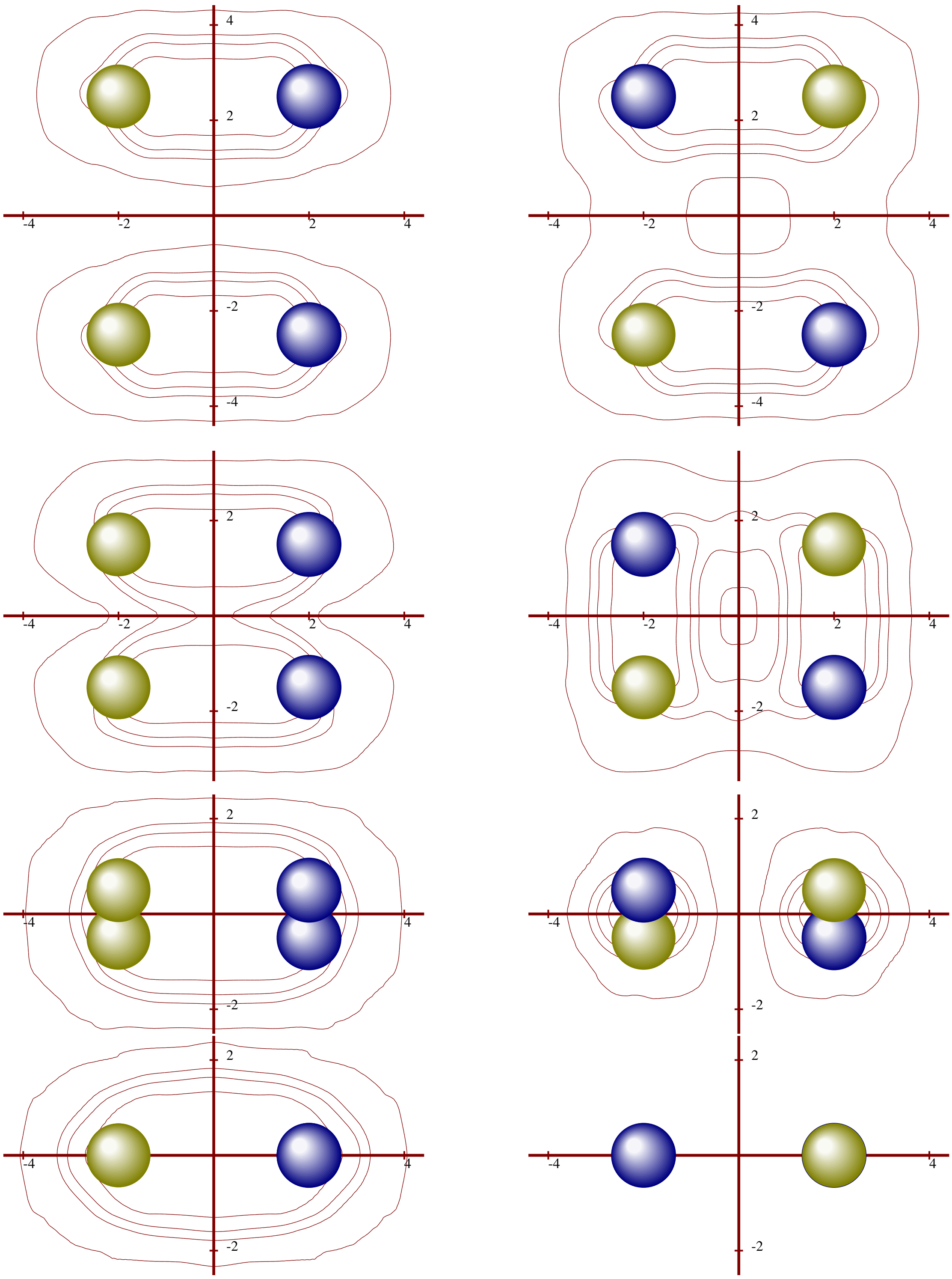


Abbildung 5.7: Fusion paralleler Flussschläuche (linke Spalte) bzw. antiparalleler Flussschläuche (rechte Spalte). Erläuterung siehe Text.

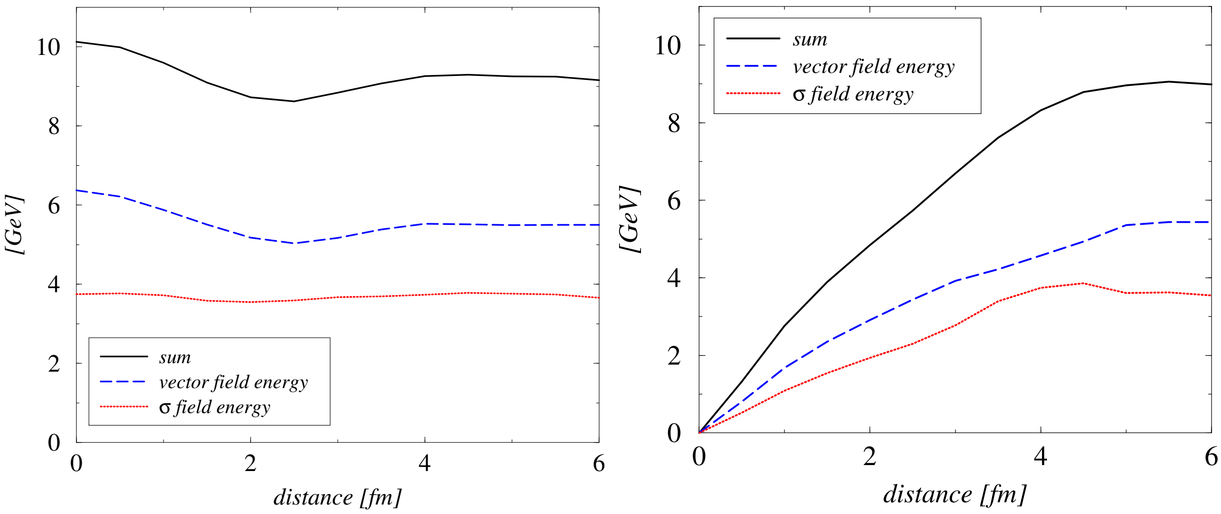


Abbildung 5.8: String-string-Potential für die parallele Konfiguration (links) und antiparallele Konfiguration (rechts). Im Falle der antiparallelen Konfiguration sieht man für kleine Distanzen wieder einen stringtypischen linearen Anstieg, hier allerdings mit der doppelten Steigung von $2GeV/fm$, da hier zwei Flussschläuche aufaddiert werden. Die Feldenergie beider Konfigurationen konvergiert für große Stringentfernung zum Wert $9GeV$, was etwa dem Doppelten der Feldenergie eines $q\bar{q}$-Strings der Länge $4fm$ aus Abb. 5.6 entspricht.

und erklären, wie sie aus Simulationsdaten gewonnen werden können. Am Ende diskutieren wir unsere Resultate für Massenverteilungen und die Zeitentwicklung der Quark-Gluon-Hadron-Materie und analysieren spektrale Eigenschaften.

5.5.1 Initialisierung eines Anfangszustands

Um eine molekulardynamische Simulation mit explizit propagierten Farbladungen und -feldern durchführen zu können, müssen wir einige Kompromisse eingehen. Mit heutigen Computern können wir trotz der ausgefeilten Numerik nur Systeme mit einigen hundert Teilchen in einem Volumen von $(25fm)^3$ simulieren. Ein Programmlauf braucht zur vollständigen Hadronisierung etwa $200fm$ Modellzeit; die korrespondierende Rechenzeit beträgt einige Wochen auf einem heutigen PC[2].
Wegen dieser Beschränkungen wurde die Entscheidung getroffen, nur einen zentralen Teil des Kollisionsbereichs der Schwerionenreaktion zu simulieren, in dem sich etwa 400 Quarks und Gluonen auf etwa $100fm^3$ dicht gedrängt versammeln. In realen Pb-Pb-Kollisionen am CERN SPS befinden sich vermutlich mehrere tausend Partonen in der QGP-Phase. Wir vergleichen zwei unterschiedliche Szenarien, um einen qualita-

[2] Pentium Pro 200 MHz, 256 MB, Betriebssystem Linux.

tiven und ersten Einblick in die Hadronisierung zu erhalten: (i) ein vollständig abgestopptes QGP mit einer Anfangstemperatur von $T_0 = 160 MeV$, und (ii) ein heißes QGP, initialisiert mit einem Bjorken-Flow (lokal lorentzinvariant), der sich über den Rapiditätsbereich $[-y_0, +y_0] = [-0.5, +0.5]$ erstreckt.
Das Reaktionsvolumen wurde berechnet, indem wir einen Radius von $R_0 = 4fm$ für die anfängliche QGP-Kugel bzw den QGP-Zylinder annehmen. Dazu kommt eine anfängliche Energiedichte von $\varepsilon_0 = 2.5 GeV/fm^3$, die aus der SPS-Kollisionsenergie resultiert. Die Quark- und Gluonzahlen der verschiedenen Flavors werden nach der nichtrelativistischen Maxwell-Boltzmann-Statistik gewählt, wobei die effektiven Massen und die Anfangstemperatur eingehen. Für jede Sorte (Flavor) i ergibt sich dabei die Gesamtzahl der Partonen dieser Sorte als proportional zu

$$N_i(0) \propto e^{-m_i/T_0} d_i \left(\frac{m_i T_0}{2\pi} \right)^{3/2}, \tag{5.11}$$

wobei d_i die Degenerationszahl (Spin und Farbe) für die Partonsorte i ist. Quarks haben 2 Spin- und 3 Farbzustände; bei leichten Quarks kommt noch der Faktor 2 für die Isospinentartung hinzu, da wir in der Simulation nicht zwischen *up*- und *down*-Quarks unterscheiden. Gluonen haben 2 Spin und 6 Farbzustände. Die Zahl 6 statt 8 kommt daher, dass die beiden abelschen Gluonen nicht als Teilchen in die Simulation eingehen, sondern als klassische Felder.
Die resultierenden Teilchenzahlen für die hier vorgestellten Events sind: 133 leichte Quarks, 40 seltsame Quarks, die gleichen Zahlen der jeweiligen Antiquarks, und 44 Gluonen für das Bjorken-Event. 196 leichte Quarks, 52 seltsame Quarks, wieder die gleiche Zahl Antiquarks, und 57 Gluonen für das Full-Stopping-Event. Die gaußverteilten Impulse (im Bjorkenfall auf den initialisierte Flow aufgesetzt) erfüllen

$$\left\langle \frac{p_{i,\mu}^2}{m_i} \right\rangle = T_0 \tag{5.12}$$

für jede Partonsorte i und Impulskomponente $\mu = x$, y oder z. Wir betrachten nur verschwindendes quarkchemisches Potential in dieser Arbeit. Außerdem enthalten die initialisierten Plasmen in der Summe keine Seltsamkeit, und natürlich besitzen die Plasmatropfen verschwindende Gesamtfarbe.
An dieser Stelle ist eine Bemerkung notwendig: Die Initialisierung von Anfangszuständen, sei es nun für die Berechnung der Stringspannung, der Stringfusion oder der Hadronisierungsszenarien, gestaltet sich i. a. recht einfach: Wir initialisieren die Partonen (stationär an bekannten Orten, thermisch nach einer vorgegebenen Verteilung, mit Flow...) und lösen danach das Farbfeld mit Hilfe des Gaussgesetzes. Einziges (und kritisches) Problem dabei ist die Initialisierung des σ-Feldes. Damit die

Lösung des Farbfelds nicht etwa bereits am Anfang zum Programmabsturz führt, muss das σ-Feld so initialisiert werden, dass die Lösung des Gaussgesetzes erleichtert wird, m. a. W.: es darf nicht etwa $\sigma \equiv \sigma_{\text{vac}}$ initialisiert werden, denn das würde zu extrem starken $\vec{E}^a$-Feldern führen. Wir gehen deshalb praktisch wie folgt vor: um jedes Parton herum wird in einer Kugel mit Radius $\sim 2fm$ das σ-Feld auf den Wert 0 "heruntergedrückt". Damit das σ-Feld möglichst glatt bleibt, geschieht dies mit der Form einer umgedrehten Gaußglocke, die im Zentrum den Wert $\sigma = 0$ annimmt, außen den Wert $\sigma = \sigma_{\text{vac}}$. Wenn die Partonen dicht genug aneinander liegen (und das tun sie in unseren Fällen immer), so genügt dies für eine "sanfte" Lösung der Vektorfeldgleichung im ersten Zeitschritt. Nach wenigen fm Propagationszeit, in denen das σ-Feld sich einschwingen kann (siehe Abb. 5.11 rechts) und die überschüssige Bagenergie in σ-Wellen umgewandelt wurde, kann man den auf diese Weise ziemlich künstlich erzeugten Anfangszustand als physikalisch sinnvoll betrachten. Die Einschwingvorgänge hingegen sind natürlich "numerisches Rauschen".

5.5.2 Analyse des Endzustands

Grundlage der Identifikation aller Observablen in unseren Simulationsläufen ist die IWC-Klassifikation nach Abb. 3.5 und der Hadronisierungsmechanismus, der in Abschnitt 3.5 beschrieben wurde.
Der Gesamtimpuls und die Gesamtenergie eines IWCs kann aus den einzelnen Viererimpulsen der enthaltenen Partonen *und der Beiträge der Felder* errechnet werden. Die Masse eines IWCs ist dann definiert durch

$$M_{\text{IWC}}^2 = \left(E_{\text{Feld}} + \sum_i E_i\right)^2 - \left(\vec{p}_{\text{Feld}} + \sum_i \vec{p}_i\right)^2 . \tag{5.13}$$

Hierbei gehen Impuls und Energie ein:

$$\begin{aligned} E_{\text{Feld}} &= \int d^3x \left(\frac{1}{2}\dot{\sigma}^2 + \frac{1}{2}(\vec{\nabla}\sigma)^2 + U(\sigma) + \frac{1}{2}\kappa(\sigma)\vec{E}^a\vec{E}^a\right) \\ \vec{p}_{\text{Feld}} &= -\int d^3x\, \dot{\sigma}\vec{\nabla}\sigma \quad .^3 \end{aligned} \tag{5.14}$$

Die *Massenverteilung* dN/dM repräsentiert die Zahl der IWCs mit einer Masse zwischen M und $M + dM$. Als Auflösung dM des Massenbinnings wurde $dM = 100MeV$ gewählt. Wenn dM zu groß ist, kann man verschiedene Hadronsorten in der Verteilung nicht mehr unterscheiden; ist

[3] Aufgrund unserer Vernachlässigung des $\vec{B}^a$-Felds verschwindet der Farb-Poyntingvektor, und das Farbfeld leistet keinen Beitrag zum Feldimpuls.

dM hingegen zu klein, wird die Statistik zu schlecht, und die Verteilung verliert an Aussagekraft. Der Wert $100MeV$ hat sich als Kompromiss bewährt.
Weitere Observablen sind die *Rapiditätsverteilung* dN/dy mit

$$y = \frac{1}{2} \ln \frac{E + p_z}{E - p_z}, \tag{5.15}$$

und die *transversale Impulsverteilung* $dN/p_T dp_T$ mit

$$p_T = \sqrt{p_x^2 + p_y^2}\,. \tag{5.16}$$

Der kollektive Flow und die lokale Temperatur beeinflussen diese Spektren: ein starker longitudinaler Flow verursacht ein Plateau im dN/dy-Spektrum, während die Steigung der transversalen Impulsspektren — falls sie denn exponentielle Form besitzen — sowohl die thermische (Brownsche) Bewegung als auch radialen Flow miteinbeziehen.
Schließlich sind noch einige globale Größen von Interesse, um aus der molekulardynamischen Simulation einen Einblick in die ungefähre Zustandsgleichung zu gewinnen. Die *Temperatur* (bzw. eine *temperaturartige Größe*) kann berechnet werden, indem wir die Impulsverteilung an eine nichtrelativistische Maxwell-Boltzmann-Verteilung anfitten. Dadurch erhalten wir die "Temperatur" als

$$T_{i,\mu} = \langle p_{i,\mu}^2 \rangle / m_i \tag{5.17}$$

für die Clustersorte i mit Masse m_i und Impulsrichtungen $\mu = x, y$ oder z. Falls diese "Temperaturen" für alle Raumrichtungen etwa gleich sind, so spiegelt dies Druckisotropie des Plasmas wieder. Sind die $T_{i,\mu}$ hingegen für verschiedene Clustersorten i ungefähr gleich, so bedeutet dies chemisches Gleichgewicht. Beide Bedingungen gemeinsam zeigen thermisches Gleichgewicht an.
Die Analyse der Clustergrößenverteilungen oder höherer Momente der Spektren muss auf spätere Projekte verschoben werden, wenn höhere Statistik verfügbar ist. H. Satz [70] wies darauf hin, dass solche Untersuchungen Information über die Ordnung des QGP-Phasenübergangs liefern könnten.

5.5.3 Schnappschüsse der Simulation

Abb. 5.9 und Abb. 5.10 zeigen sechs Momentaufnahmen der Zeitentwicklung für das Bjorken- und Full-Stopping-Szenario. Die Simulationszeit jedes Schnappschusses ist in den jeweiligen Rahmen in fm markiert; die Rahmendiagonale beträgt ungefähr $45fm$. Quarks, Gluonen und Hadronen sind als Kugeln gezeichnet. Um die Szenen besser überblicken zu können, wurden die Radien aller farbigen Objekte auf ein Drittel des

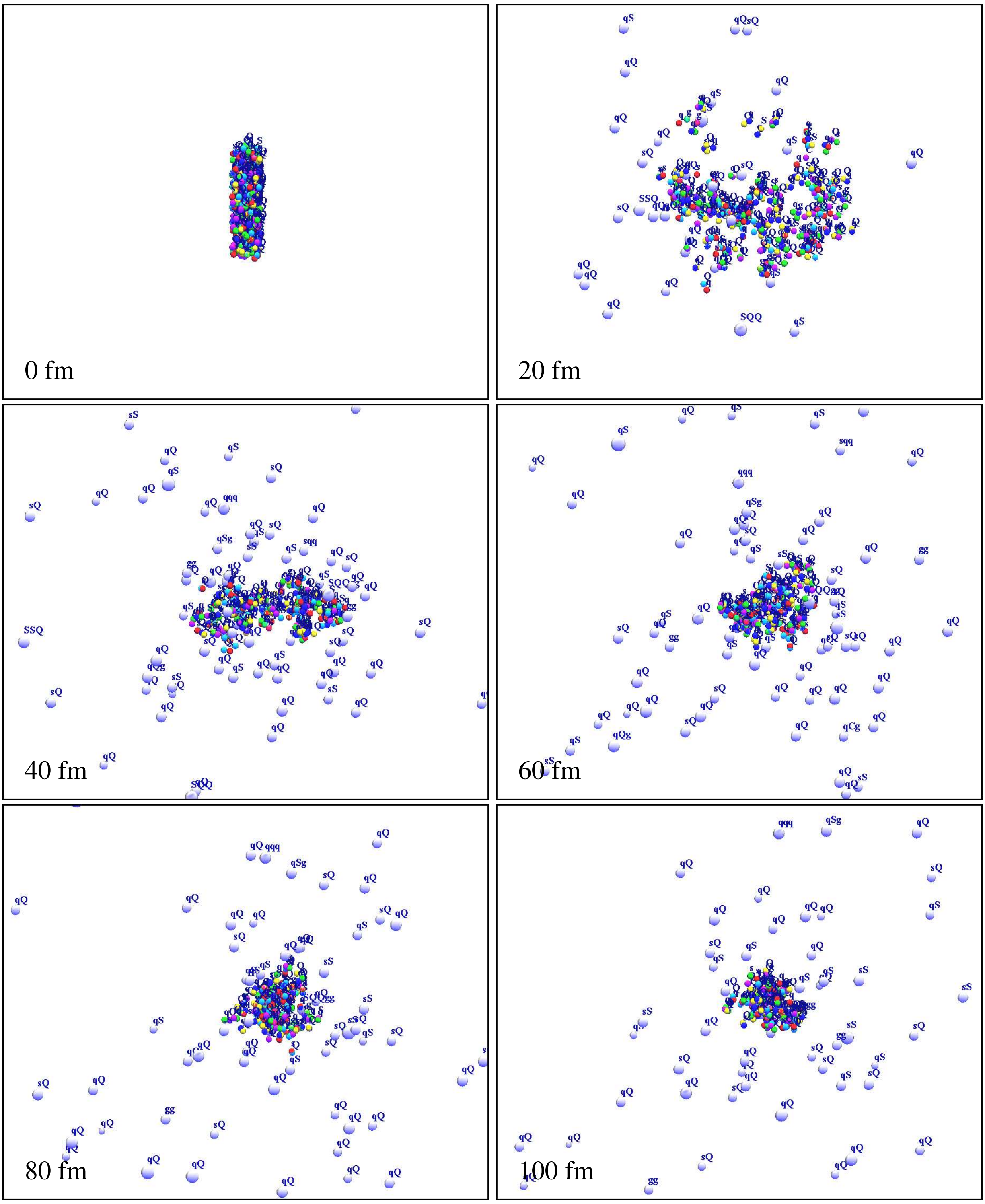


Abbildung 5.9: Schnappschüsse (Momentaufnahmen) der Computersimulation nach der Initialisierung eines zentralen Rapiditätsintervalls eines QGP-Zylinders mit lorentzinvariantem Flow ("Bjorkenszenario"). Die Rahmendiagonale beträgt etwa $45\,fm$. Die kleinen Etiketten an den Partonen und Hadronen zeigen die Sorten (Flavors) der Partonen und Bestandteile der IWCs an. Dabei bedeutet der Buchstabe q ein leichtes Quark (up oder down); die anderen Buchstaben haben offensichtliche Bedeutung, wobei Großbuchstaben Antiquarks bezeichnen.

tatsächlichen RMS-Radius der Ladungen ($0.7fm$) verkleinert. Die Radien der Hadronen (IWCs) sind so gewählt, dass sie das Volumen des σ-Bags vor der Hadronisierung widerspiegeln, auch diese Radien sind zur graphischen Darstellung nochmals auf die Hälfte herunterskaliert worden. Für IWCs ohne σ-Bag ("kalte" Hadronen haben in CDM keinen Bag, siehe Abschnitt 4.5) zeichnen wir eine Kugel mit Radius $0.35fm$. Sowohl σ- als auch Farbfeld sind nicht dargestellt. Das Farbfeld wäre in dieser Skalierung nicht gut zu erkennen, und das σ-Feld, selbst wenn wir es auf einer Schnittebene darstellen, lässt vor allem Wellenbewegungen erkennen, denen man wenig Struktur ansieht. Im jeweils letzten Schnappschuss haben viele Hadronen bereits das Bild verlassen. Das ist auch der Grund, weshalb wir nicht die gesamte Hadronisierung (bis hinauf zu $200fm$ Simulationszeit) zeigen: es wären viele fast leere Bilder dabei.

Wie man an den Bildern erkennen kann, expandiert der initialisierte QGP-Feuerball und hadronisiert. Zu jedem Zeitschritt werden die Bewegungsgleichungen der Ladungen und des σ-Feldes sowie das Gaußgesetz gelöst. Sobald ein IWC detektiert wird (durch den in Abschnitt 4.5 beschriebenen Clustererkennungsmechanismus), wird es nach Abb. 3.5 als Meson, Baryon oder Glueball klassifiziert. Die gesamte Energie, der Impuls und die Masse des IWCs werden berechnet, und die Partonen und der Feldeindruck des IWCs wird aus der Simulation entfernt und durch ein nicht weiter wechselwirkendes "Hadron" ersetzt, das in den folgenden Momentaufnahmen als graue Kugel erscheint.

Dies geschieht nicht nur, um im weiteren Verlauf der Simulation Rechenzeit zu sparen. Darüberhinaus ist es eine prinzipielle Notwendigkeit in einer Hadronisierungssimulation, dass an irgendeiner Stelle einmal "Hadronen" erkannt und definiert werden und die inneren Freiheitsgrade der zugehörigen Quarkcluster vernichtet werden. Von diesem Schritt an hat das neu erzeugte Hadron nur noch Gesamtenergie und -impuls, aber keine inneren Freiheitsgrade mehr. Die Anregungsenergie wird dabei der Masse zugeschlagen. Eine nachgeschaltete hadronische Kaskade lässt die Hadronen natürlich alle innerhalb kürzester Zeit zu Protonen, Neutronen, Pionen, Kaonen etc. zerfallen lassen, wobei die Anregungsenergie hauptsächlich in Form von Pionen abgestrahlt wird. Diese nachgeschaltete Kaskade ist jedoch in unserer Simulation nicht enthalten; für uns ist das (angeregte) Hadron der Endzustand der Simulation.

Es soll an dieser Stelle betont werden, dass die Produktion von hadronischen Zuständen nicht künstlich vom CDM-Programm erzwungen wird. Vielmehr produziert das Modell durch seine Bewegungsgleichungen selbst IWCs im Endzustand, und es verbleibt nur die Aufgabe, sie zu erkennen, zu klassifizieren und aus dem Simulationsuniversum zu entfernen. Die Detektorroutine interferiert dabei in keiner Weise mit dem modelleigenen Mechanismus der Hadronenformation; sie dient nur dazu, die Resultate zu interpretieren. Auch wenn die Momentaufnahmen bei $100fm$ immer noch Quarks und Gluonen zeigen, ist dies nicht der Endzustand der Simulationsläufe. Im Endzustand ist die Hadronisierung vollständig: kein partonischer Freiheitsgrad bleibt übrig.

5.5.4 Energieerhaltung, Hadronisierungszeit und Temperatur

Abb. 5.11 zeigt den *Verlauf der Gesamtenergie* und der einzelnen Beiträge über das simulierte Zeitintervall. Dem Graphen liegt das gleiche Bjorkenereignis zugrunde, für das die Momentaufnahmen im letzten Abschnitt gezeigt wurden (Abb. 5.9). Abb. 5.12 zeigt für das gleiche Event und Zeitintervall die Temperaturentwicklung im Feuerball und die produzierten Hadronen. Die Temperaturen sind mit Hilfe der Beziehung

$$\frac{3}{2}T \ := \ \left\langle \frac{\vec{p}^{\,2}}{2m} \right\rangle \tag{5.18}$$

berechnet worden.
Anfänglich heizen die Partonen sich auf über $500MeV$ Temperatur auf, weil sie durch relativ lange ($> 1fm$) Flussschläuche beschleunigt werden. Diese Flussschläuche sind einfach deswegen vorhanden, weil die anfängliche räumliche Zufallsverteilung der Farbladungen unphysikalisch ist. Sehr bald jedoch (nach $0.5fm$) kühlen die heißen Partonen sich auf etwa $350MeV$ ab. In den folgenden $10fm$ kühlt das Plasma schnell ab, wobei es viel Energie zum (anfänglich kalten) σ-Feld überträgt. Im gleichen Zeitraum nimmt die Vektorfeldenergie stark ab und ist nach $25fm$ fast verschwunden. Dies ist ein Indikator für die starke elektrische Abschirmung des Plasmas. Das Bild des Plasmas als Menge von Flussschläuchen bricht also bereits ganz am Anfang zusammen. Dies erfolgt einfach durch die Thermalisierung; in der Realität würden Flussschläuche auch durch $q\bar{q}$-Produktion zerrissen, ein Mechanismus, der in unserer Simulation nicht enthalten ist.
Flussschläuche treten im folgenden nur noch auf der Oberfläche des Feuerballs auf, wenn sie farbige Objekte am Entkommen hindern. Eine interessante Beobachtung ist, dass das Farbfeld so wenig Energie beiträgt, obwohl es als wesentlicher Bestandteil des Confinementmechanismus die Dynamik diktiert.
Die hadronische Energie ist eine monoton steigende Stufenfunktion: sie macht immer dann einen Sprung nach oben, wenn wir ein neues IWC erkennen, die zugehörigen Felder und Partonen aus dem System entfernen und das IWC als Hadron deklarieren. Da die Hadronen in unserem Modell nicht wechselwirken, ist die hadronische Energie zwischen den Stufen konstant. Ungefähr die Hälfte der Endzustandsenergie ist in σ-Wellen deponiert. Diese Energie kommt ursprünglich aus

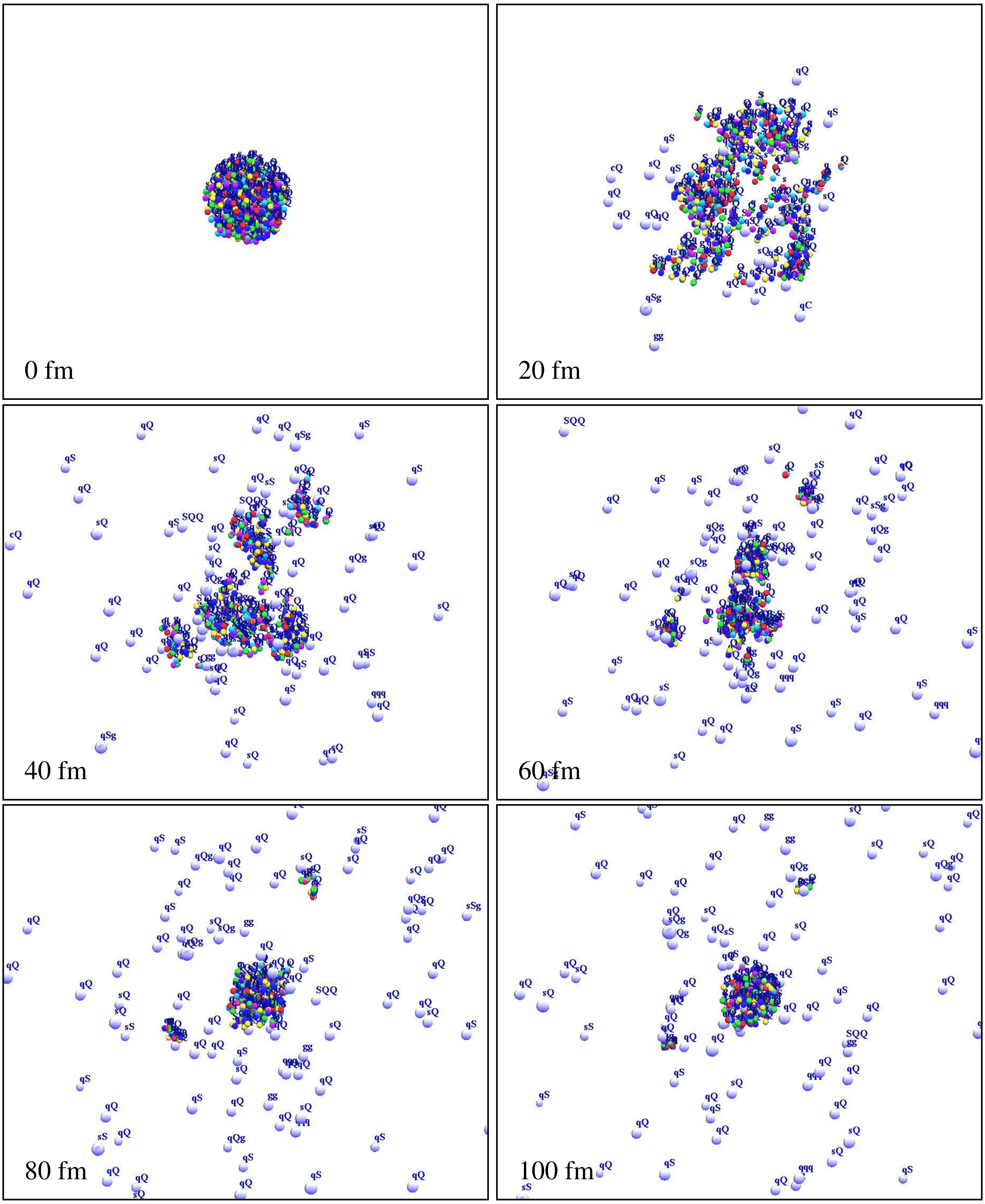


Abbildung 5.10: Schnappschüsse (Momentaufnahmen) der Simulation nach Initialisierung eines thermalisierten QGP-Feuerballs ("Full-Stopping-Szenario").

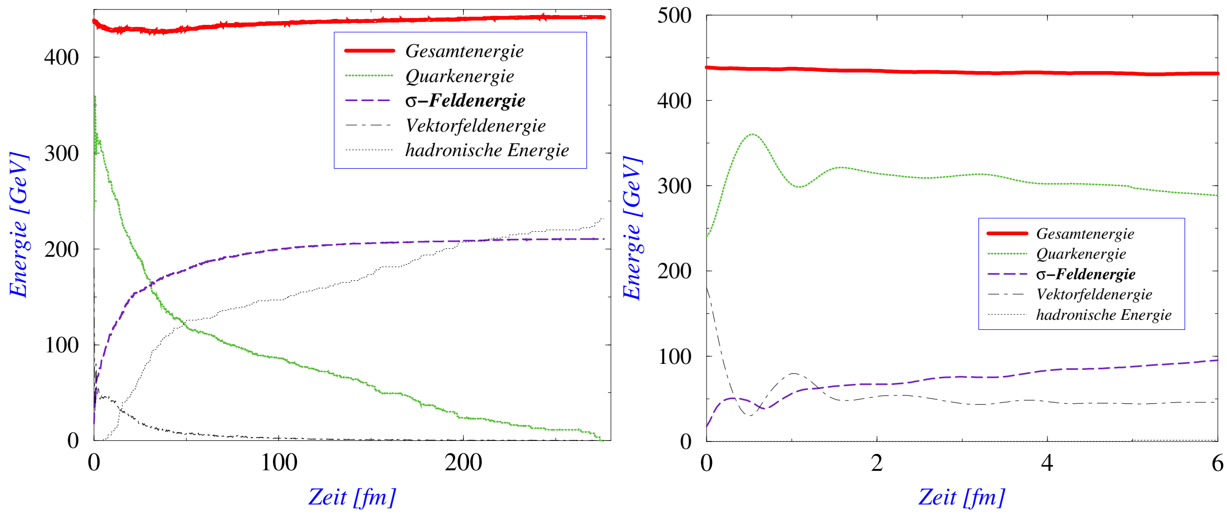


Abbildung 5.11: *Systemenergie während der Zeitentwicklung im Bjorkenevent. Rechts eine Ausschnittvergößerung (bea. andere Achsenbeschriftung).*

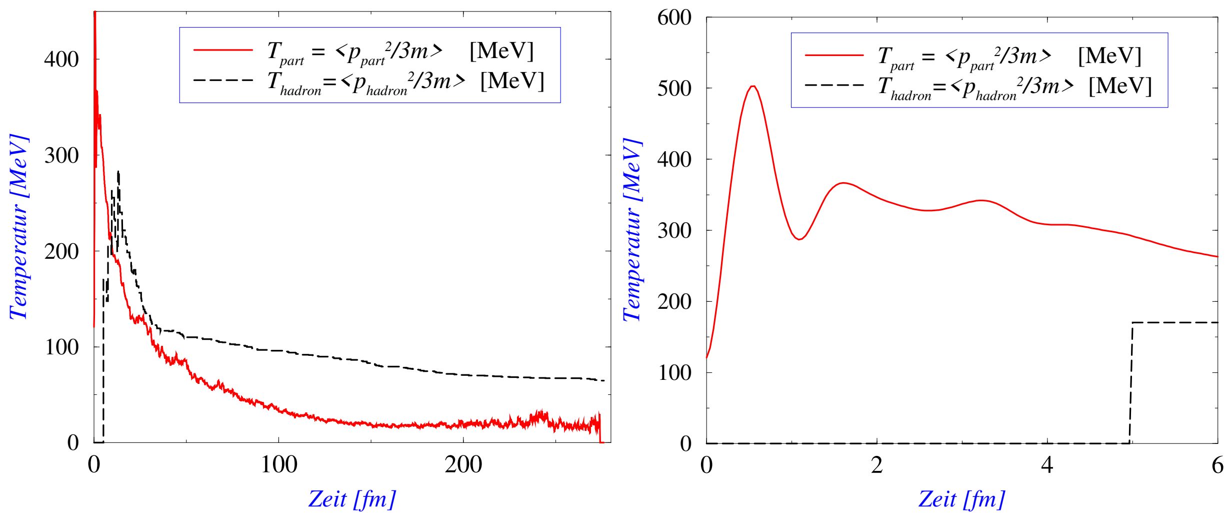


Abbildung 5.12: *Parton- und Hadrontemperaturen im Laufe der Zeitentwicklung im Bjorkenereignis. Rechts wieder eine Ausschnittvergrößerung (bea. andere Achsenbeschriftung).*

der initialisierten thermischen Energie der Farbladungen. Diese formen Flussschläuche, und über den Kopplungsterm $\kappa(\sigma)F^2/4$ kühlen die rotierenden und oszillierenden Flussschläuche ab, indem sie σ-Wellen emittieren. Da das σ-Feld völlig kalt initialisiert wird ($\dot{\sigma} \equiv 0$) und als klassisches Feld viel mehr Freiheitsgrade besitzt als die 400 Partonen, verwundert es nicht, dass die Partonen sich am σ-Feld als "Kältebad" abkühlen. Das σ-Feld beschreibt langreichweitige kollektive Effekte der QCD. Sein großer Energieinhalt im Endzustand zeigt, dass das nichtperturbative QCD-Vakuum hochangeregt ist; in der Realität würden diese Anregungen größtenteils in Pionen zerfallen, die bisher nicht im Simulationsprogramm beschrieben werden.

Man bemerke, dass der größte Teil der hadronischen Energie im Zeitintervall $[10fm, 50fm]$ produziert wird. Diese $40fm$ können daher als die Hadronisierungszeit in unserem Modell interpretiert werden.

Es gibt keine *Endzustandswechselwirkung* (*final-state interaction*) zwischen den Hadronen oder zwischen Hadronen und Partonen in unserem einfachen Modell. Außerdem gibt es keine Rückreaktion von Hadronen zu Partonen. Deshalb kann man zwischen Hadronen und Partonen kein thermisches Gleichgewicht erwarten; das hadronische Teilsystem ist nicht in Kontakt mit dem Plasmafeuerball. Dies ist in etwa realistisch: die hadronische Temperatur spiegelt die Temperatur des Partongases *zum Zeitpunkt der Hadronformation* wider (*freeze-out point*). Da die meisten Hadronen im Zeitintervall $[10fm, 50fm]$ produziert werden, ist die hadronische Temperatur des Endzustands etwa gleich der Temperatur des Feuerballs in diesem Zeitraum. Der Feuerball kühlt weiter ab, was zum unrealistischen Bild führt, dass nach etwa $30fm$ der Feuerball wesentlich kälter als die Hadronen ist.

In der Realität würde ein derart kalter Feuerball ($T = 50MeV$) wahrscheinlich unmittelbar hadronisieren. In unserer Simulation aber sind Hadronen nur dann überhaupt wohldefiniert, wenn sie klar den Feuerball verlassen haben und als separates IWC detektiert werden können. Nach spätestens etwa $30fm$ ist die Hadronisierung im CDM-Programmlauf einfach ein zufälliger Abdampfprozess: im Inneren des Plasmafeuerballs sind Farbfelder stark abgeschirmt, was stabile weiße Cluster in der Formation behindert. Deshalb können IWCs ("Hadronen") sich nur an der Oberfläche des Plasmatropfens bilden, nämlich sobald die Bestandteile eines IWCs, die im Feuerball nahezu frei umherfliegen, zufällig mit koaleszenten Impulsen an der gleichen Stelle der Plasmaoberfläche erscheinen. Solch eine Partonkombination verlässt das Plasma dann ungehindert als IWC und wird alsbald zum Hadron konvertiert. Die vollständige Hadronisierung des Plasmas dauert somit längere Zeit, besonders wenn der Feuerball schon stark abgekühlt ist.

Wir schreiben diese lange Hadronisierungszeit teilweise der fehlenden repulsiven Wechselwirkung zwischen IWCs in unserer Simulation zu [71]. Außerdem ist von Kernfragmentationstudien her längst bekannt,

dass ein Mean-Field dazu tendiert, die Bildung kleine Cluster zu verhindern [72]. Fluktuationen der Farbfelder, d.h. Abweichung von der Mean-Field-Approximation, könnten in der Endphase des Hadronisierungsprozesses daher wichtig werden.
Es ist unser Empfinden, dass weitere Abstimmung der Modellparameter dieses Resultat nicht drastisch verändern wird. Daraus ergibt sich die Frage, welche kritische Eigenschaft es wohl ist, die unserem Modell fehlt und die die Natur dazu bringt, so viel schneller zu hadronisieren. Hadronische Resonanzen mit starken Zerfallskanälen haben typische Breiten von $100 - 200 MeV$, was zu "Lebensdauern" von $1 - 2 fm$ korrespondiert. Abschätzungen der Hadronisierungszeit mit Hilfe chemischer Ratengleichungen liegen bei etwa $5 - 20 fm$. Unser Resultat, etwa $200 fm$ für die vollständige Hadronisierung eines Plasmatropfens, fällt auf diesem Hintergrund betrachtet weitaus zu groß aus.
Im gegenwärtigen Verständnis des Modells können wir nicht sagen, was unserem Modell zur schnellen Hadronisierung fehlt. Wir können aber eine falsche Antwort ausschließen: Ein Teilchenproduktionsterm, wie er in der QCD existiert und im Modell fehlt, wird die Hadronisierungszeit nicht verkürzen. Auf der einen Seite kann die Produktion zweier entgegengesetzter Farbladungen einen Flussschlauch zerreißen. Folglich kann ein einzelnes Parton, sagen wir ein Quark, mit hoher Energie den Feuerball verlassen, wenn der Flussschlauch, der es zurückzuhalten versucht, durch $q\bar{q}$-Produktion zerrissen wird. Das Antiquark bildet dann mit dem entkommenen Quark ein Meson. Dieser Vorgang lässt aber einen Plasmatropfen zurück, der nicht weniger Quarks als zu Beginn enthält. Ein schnelles Quark kann also schnell hadronisieren, aber die Gesamtzahl der Farbladungen im Plasma nimmt dadurch nicht ab. Wir ziehen aus dieser Überlegung den Schluss, dass ein Teilchenproduktionsterm zwar die Abkühlung des Plasmatropfens beschleunigt, nicht aber dessen Hadronisierung.

5.5.5 Massenverteilungen

Die hadronische Massenverteilung im Endzustand der Hadronisierung ist in Abb. 5.13 für beide betrachteten Hadronisierungsereignisse gezeigt. Die Spektren beginnen bei $800 MeV$, der Masse des leichtesten Hadrons $q\bar{q}$ im Modell. Leichtere Teilchen wären farbige Partonen, und Farbladungen sind im Endzustand nicht mehr vorhanden.
Auch wenn die Spektren auf den ersten Blick sehr verschieden aussehen, sollte man bei der Betrachtung bedenken, dass es sich um einzelne Events handelt. Die Unterschiede können also den statistischen Fluktuationen zugeschrieben werden und sind nicht notwendigerweise typisch für das jeweilige Szenario. Man bemerke auch, dass die *Form* des Massenspektrums im Bjorkenszenario exakt die gleiche bliebe, wenn das untersuchte Rapiditätsintervall $\Delta y = [-0.5, 0.5]$ auf eine realisti-

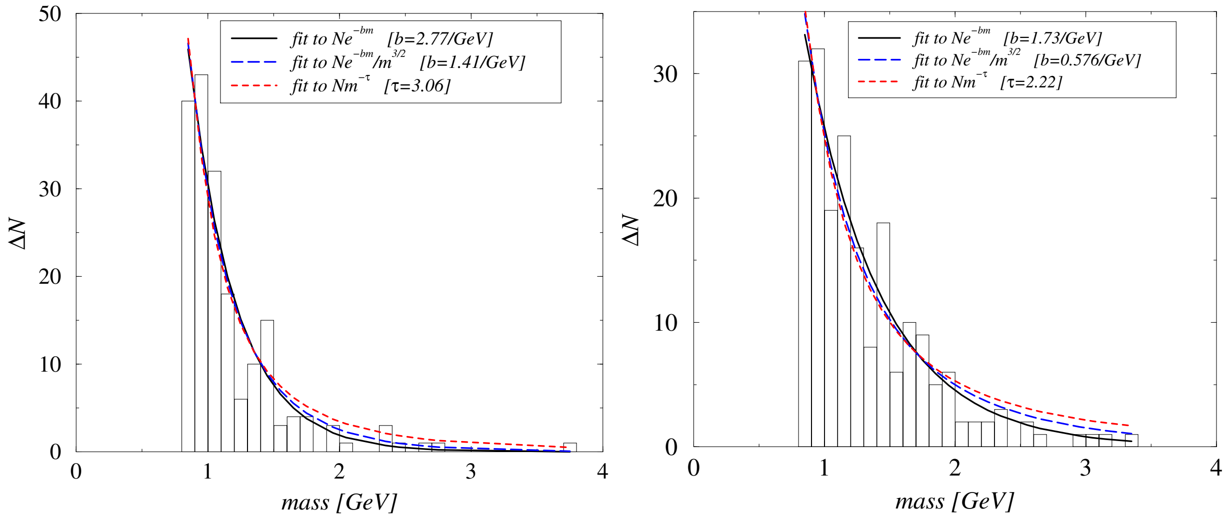


Abbildung 5.13: *Massenverteilungen; der linke Graph gehört zum Bjorkenereignis, der rechte zum Full-Stopping-Ereignis.*

sche Ausdehnung erweitert würde, einfach weil Massen lorentzinvariant sind.
Die nun folgende Diskussion ist ein Versuch, unsere Resultate mit denen anderer zu vergleichen. Da die Vergleichbarkeit nicht unbedingt gewährleistet ist, sei darum gebeten, diese Absätze nicht allzu ernst zu nehmen[4].
Wenn wir eine *Hagedornsche hadronische Zustandsdichte* annehmen [73],

$$\rho(m) \;\propto\; m^a e^{m/T_H}, \tag{5.19}$$

so erwarten wir als Konsequenz im nichtrelativistischen Grenzfall eine gemessene Massenverteilung der Form

$$\begin{aligned} \frac{dN}{dm} \;&\propto\; \rho(m) \int d^3p\, e^{-(m+p^2/2m)/T} \\ &\propto\; m^{a+\frac{3}{2}} e^{-m\left(\frac{1}{T}-\frac{1}{T_H}\right)}. \end{aligned} \tag{5.20}$$

Der nichtrelativistische Limes ist gerechtfertigt, weil die Konstituentenmassen der Partonen unseres Modells so hoch sind. In (5.19) und (5.20) bedeutet T_H die sog. *Hagedorntemperatur*, die im *statistischen Bootstrapmodell*[5] die Grenztemperatur hadronischer Materie darstellt. Jenseits

[4]Diese defensive Einstellung ist nach Meinung des Autors einfach professionell und wird i. d. R. auch von den Vertretern der anderen Modelle eingenommen werden.

[5]Das statistische Bootstrapmodell ist ein Modell aus den sechziger Jahren [40], in dem die Hadronen als “aus Hadronen bestehend” erklärt werden. Das Hadronenspektrum wird in diesem Sinne aus sich selbst generiert, was zu (5.19) als Massenspektrum führt.

dieser Temperatur muss das Flussschlauchbild durch ein echtes Plasma (mit freien Farbladungen) ersetzt werden. Wir nehmen $T_H = 160 MeV$ an (wie Hagedorn selbst ursprünglich schätzte) und setzen $a = -3$ nach Gorenstein et al. [73]. Abb. 5.13 zeigt, dass ein befriedigender Fit der Form (5.20) möglich ist. Unser Fitparameter b ist dabei zur Temperatur T über die Beziehung

$$b = \frac{1}{T} - \frac{1}{T_H} \tag{5.21}$$

verwandt. Die aus dem Fit resultierende Hadrontemperatur beträgt $T = 130 MeV$ für das Bjorkenszenario ($T = 146 MeV$ für das Full-Stopping-Szenario), was etwas höher als die kinetische Temperatur liegt, die im letzten Abschnitt gemessen wurde. Diese Werte hängen aber von der Wahl des Parameters a ab; nehmen wir $a = -3/2$, so sehen wir eine simple exponentielle Verteilung der Massen, die uns zu anderen T führt, nämlich $T = 110 MeV$ für das Bjorkenereignis ($125 MeV$ Full-Stopping-Ereignis). In dieser Einzel-Event-Analyse mit kleiner Statistik ist es nicht wirklich möglich zu entscheiden, welches der beste Wert für a ist.
Eine weitere Erwartung an die Massenverteilung könnte die eines *Perkolationsprozesses* nahe dem kritischen Punkt sein [74]. Hier rechnet man mit einer skalierenden Lösung

$$\frac{dN}{dm} \propto m^{-\tau}. \tag{5.22}$$

Versucht man, diese Relation an unsere Daten anzupassen (siehe Abb. 5.13), so findet man eine befriedigende Übereinstimmung für $\tau = 3.06$ (Bjorken) bzw. $\tau = 2.22$ (full stopping). Bemerkenswerterweise widersprechen unsere Daten nicht den Erwartungen aus der Perkolationstheorie.

5.5.6 Rapiditäts- und Impulsspektren

Abb. 5.14 zeigt die *hadronischen Rapiditätsspektren* beider untersuchter Szenarien. Die beiden Bilder sehen nicht wirklich verschieden aus, auch wenn im Falle des Bjorkenszenarios die Verteilung wegen des longitudinalen Flows etwas breiter sein mag. Vor allem ist keine Plateaustruktur im Bjorkenfall erkennbar, weil wir vom zylindrischen, lorentzinvarianten QGP nur ein schmales zentrales Rapiditätsintervall ausgeschnitten und simuliert haben. Man könnte Abb. 5.14 leicht zum vollen Bjorkenszenario extrapolieren, indem man die gezeigte Rapiditätsverteilung mit der eines Bjorken-QGP-Zylinders mit realistischen Ausmaßen faltet. Diese Faltung würde in einer Plateaustruktur resultieren.
Schließlich zeigt Abb. 5.15 noch die *transversale Impulsverteilung* $dN/p_T dp_T$ der nichtseltsamen IWCs im Bjorkenszenario, beobachtet in Massenbins von $0.6 GeV$ Breite. Hierbei ist der transversale Impuls definiert als

$$p_T = \sqrt{p_x^2 + p_y^2} \quad , \tag{5.23}$$

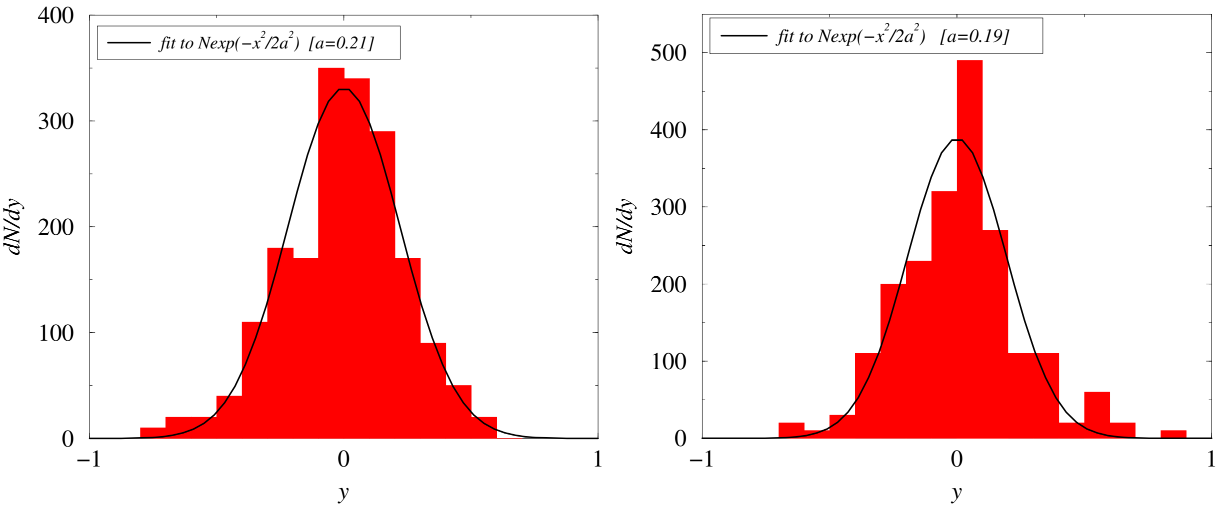


Abbildung 5.14: Rapiditätsspektren; links wieder das Ereignis im Bjorkenszenario, rechts das Full-Stopping-Szenario.

so dass die Verteilung $dN/p_T dp_T$ lorentzinvariant bezüglich der Strahlrichtung ist. Auch wenn die Datenpunkte nicht wirklich auf exponentiellen Kurven liegen, ist es interessant, die Steigungswerte gegen die Massen zu plotten, um mögliche Information über den Plasmaflow und die Temperatur am Freeze-Out-Punkt zu gewinnen. Die inversen Steigungen, wir bezeichnen sie hier wieder einmal als “Temperaturen” $T(m)$, sind in Abb. 5.16 gegen die IWC-Masse m geplottet. Der Fit an die lineare Beziehung

$$T(m) = T_0 + m\langle v_T^2\rangle \tag{5.24}$$

gelingt erstaunlich gut (trotzdem sind die Ergebnisse wegen der großen Fehlerbalken der Datenpunkte nicht allzu ernst zu nehmen). Wir lesen vom Plot eine Temperatur von $T_0 = 77MeV$ und einen radialen Flow von

$$\sqrt{\langle v_T^2\rangle} = 0.32$$

ab. Die direkte Berechnung des radialen Flows aus dem Endzustand der Simulation liefert

$$\sqrt{\langle v_T^2\rangle} = 0.30 \quad ,$$

was dem Fitwert wirklich *bemerkenswert* nahe kommt. Die analoge Untersuchung des Full-Stopping-Events macht uns da wieder bescheidener: die gleiche $T(m)$-Analyse liefert $T_0 = 126MeV$ und $\sqrt{\langle v_T^2\rangle} = 0.18$, während der wahre radiale Flow $\sqrt{\langle v_T^2\rangle} = 0.27$ beträgt — eine weniger gute Übereinstimmung, die angesichts der geringen Statistik aber nicht enttäuschen kann.

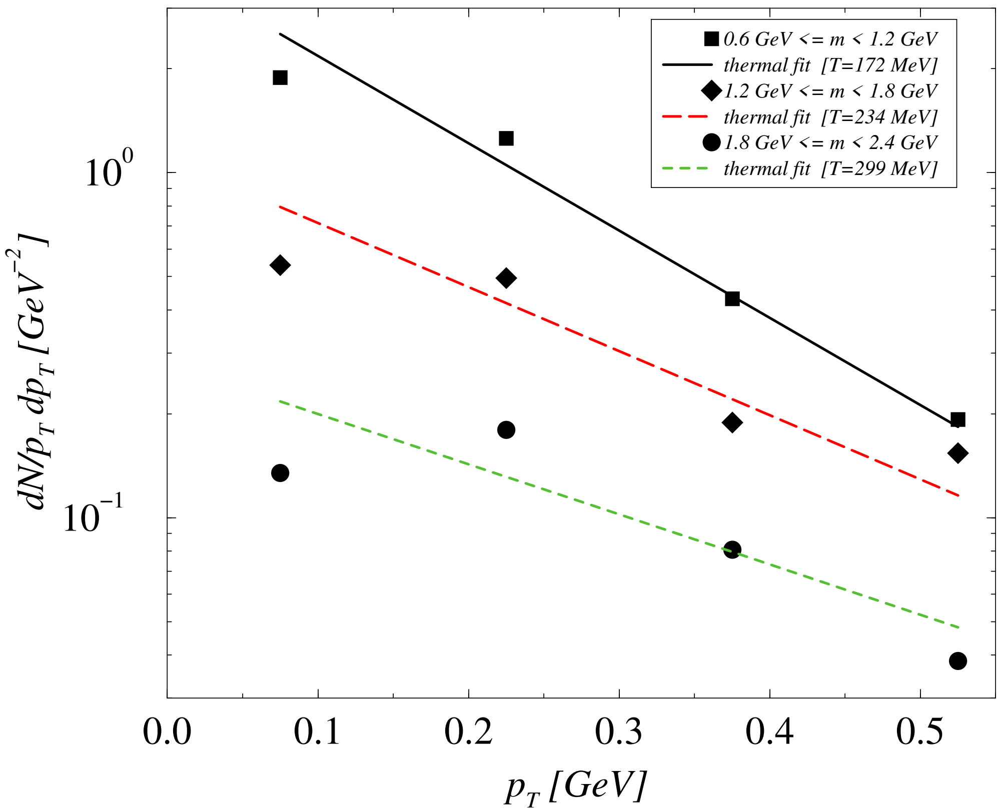


Abbildung 5.15: Transversales Impulsspektrum im Endzustand des Bjorken-Events, getrennt nach drei Massenbereichen geplottet. Die Regressionsgeraden sind unter Berücksichtigung der jeweiligen statistischen Gewichte berechnet worden [60].

Verglichen mit experimentellen Daten in SPS-Schwerionenkollisionen am CERN [75]

$$\begin{aligned} \langle v_T \rangle &= 0.4 \\ v_{\max} &= 0.6 \\ T &= 140\; MeV \end{aligned}$$

zeigt unser Resultat für das Bjorkenereignis ein etwas geringeren Flow und einen kälteren Feuerball. Es soll daran erinnert werden, dass diese Analyse auf einem einzelnen Event beruht und die Statistik angesichts des kleinen untersuchten Plasma-"Ausschnittes" viel zu gering ist, um ernsthaft Aussagen machen zu können. Außerdem haben wir Spektren von relativ schweren IWCs betrachtet. Die Zerfallsprodukte (z. B. Pionen) können wesentlich höhere p_T-Steigungen besitzen, was insgesamt zu einer höheren Schätzung von T_0 führt. Aus diesem Blickwinkel sind die gefundenen Übereinstimmungen äußerst befriedigend.

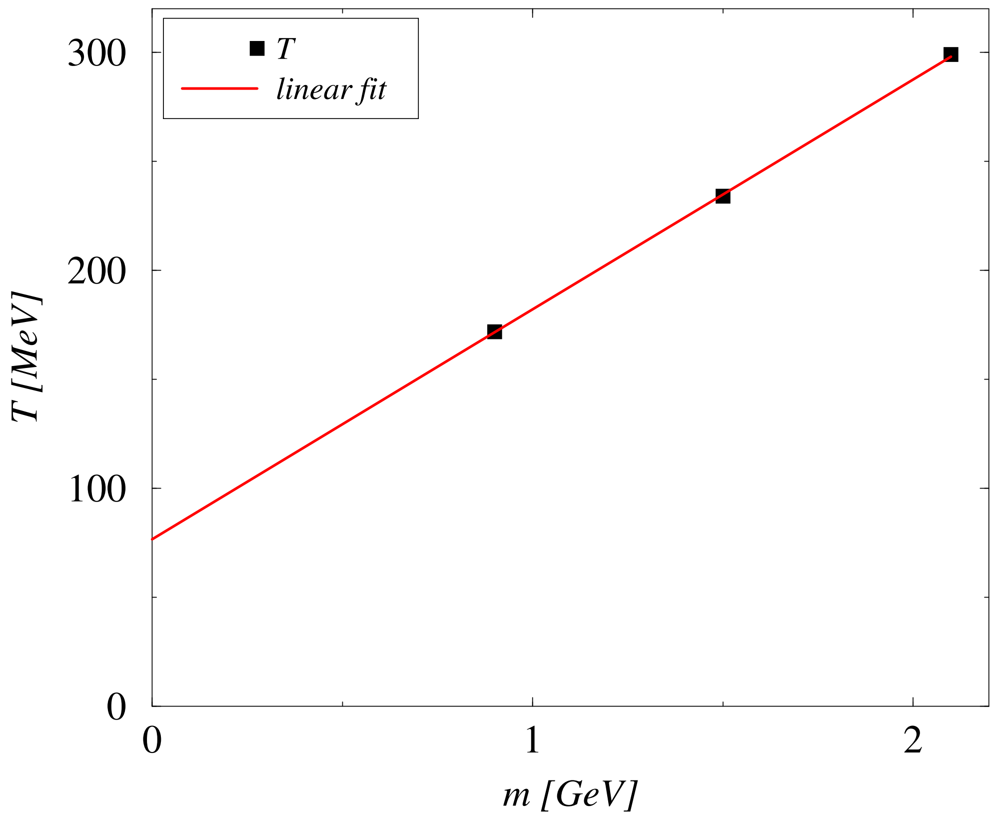


Abbildung 5.16: Temperaturen, die aus den massendifferentiellen p_T-Spektren extrahiert wurden, gegen die Hadronenmasse geplottet. Diese Art des Plots wurde von [75] übernommen und erlaubt im Experiment einen Schluss auf den Partonflow in der Plasmaphase kurz vor dem Ausfrieren.

"Now doctor will you give me in a few words the low-down on all your investigations?"
"No,", says he.
"Good,", says I.
— Interview mit P. A. M. DIRAC, *aus [69].*

6 Zusammenfassung und Ausblick

6.1 Innovationen und Resultate

Grundlage dieser Arbeit ist das chromodielektrische Modell (CDM), das in Kapitel 3 vorgestellt und theoretisch analysiert wurde. Ein überraschendes Ergebnis war dabei, dass die klassische Approximation und der Ansatz starrer Farbladungsverteilungen bereits die abelsche Approximation erzwingen (siehe Abschnitt 3.3.2). Klassische Punktteilchen in CDM erhalten die Wonggleichung als Bewegungsgleichung, wobei diese interessanterweise als Zwangsbedingung zweiter Klasse aus der Kontinuitätsgleichung folgt (siehe Abschnitt 3.3.3).
Zur praktischen Durchführung von dynamischen Modellrechnungen wurde ein Differentialgleichungslöser von so hoher Qualität entwickelt (Kapitel 4), dass auch kleine Farbladungsunterschiede bereits die Dynamik größerer Partoncluster entscheidend beeinflussen können. Hierzu wurde erstmals ein adaptives Mehrgitterverfahren in drei Dimensionen mit quadratischen Ansatzfunktionen verwendet. Die Lösung wird dadurch erschwert, dass die Feldgleichungen fast singulär ($\kappa_{\text{vac}} \approx 0$) sind[1]. Die numerische Simulation macht auch weitere Approximationen notwendig, insbesondere müssen wir uns auf ein kleines Volumen als "Simulationsuniversum" beschränken. Unphysikalische Spiegelladungen, die in der Nähe des Gebietsrandes unerwünschte Effekte bewirken, werden durch einen Trick entfernt: wir verwenden ein kleines Gebiet für das kurzreichweitige σ-Feld und ein wesentlich größeres für das langreichweitige Vektorfeld (Abb. 4.29). Irreduzible weiße Cluster (IWCs) werden vollautomatisch erkannt, nach Abb. 3.5 klassifiziert und aus der Plasmasimulation beseitigt ("zu Hadronen kondensiert").

[1] Stefan Loh [35] umging dieses Problem durch geeignete Wahl des Gebietsrandes.

Im Kapitel 5 schließlich wurde der Hadronisierungsprozess eines Quark-Gluon-Plasmas untersucht. Die Modellparameter (Quark- und Gluonmassen, Form der Funktionen $U(\sigma)$ und $\kappa(\sigma)$, starke Kopplungskonstante g) wurden so angepasst, dass bestimmte Hadronenmassen (die jeweils erste Resonanz über dem Grundzustand in jedem Flavorkanal), die Stringspannung und der Vakuumdruck nach dem MIT-Bagmodell gut getroffen werden.

Als Beispielszenarien wurden zwei unterschiedliche Anfangszustände untersucht, einer mit voller Abstoppung (*full stopping*), der andere mit Bjorken-Transparenz und dem dazugehörigen invarianten *Flow*. Beide Szenarien verhalten sich sehr ähnlich, wohl hauptsächlich wegen der vergleichbaren anfänglichen Energiedichte. Obwohl die Simulationszeit bis zur vollständigen Identifikation aller irreduziblen weißen Cluster (IWCs) relativ lange dauert, entwickelt die Massenverteilung bereits relativ am Anfang, nach etwa $10 fm$ Simulationszeit, einen hadronischen Teil. Der größte Teil des Plasmas ist hadronisiert, nachdem weitere etwa $25 fm$ verstrichen sind. Ein neuerer RQMD-Artikel [76] berichtet über ähnliche Hadronisierungszeiten.

Die Hadronisierung ist mit Sicherheit ein wesentlich schnellerer Prozess als unsere Simulation suggeriert. Dies mag damit zusammenhängen, dass CDM ein IWC erst eindeutig identifizieren kann, wenn es den Feuerball bereits verlassen hat. Deshalb ähnelt die Hadronisierung in dieser Simulation einem langsamen Abdampfprozess. In der Realität können sich Prä-Hadronen natürlich bereits im Plasma formieren. Solche Teilchen bleiben aus prinzipiellen Gründen durch unser Programm undetektiert. Eine effektive, quantenmechanische, abstoßende Kraft zwischen weißen Clustern, wie sie von Koepf et al. [71] berechnet wurde, könnte dieses Bild ändern, wenn nur ein geeigneter Weg gefunden würde, solch eine Kraft in unsere Simulation einzufügen.

Im vorgestellten Ansatz wird die Hadronisierung allein durch die Kräfte des Farbconfinements ausgelöst, im Gegensatz zum NJL-Modell, wo die chirale Symmetrie die Hadronisierungsdynamik bestimmt [23]. Diese beiden Ansätze sind komplementär: CDM enthält perfektes Farbconfinement; die hohen effektiven Quarkmassen des Modells müssen jedoch indirekt durch die chirale Symmetriebrechung erklärt werden, die im Modell nicht enthalten ist. Auf der anderen Seite enthält das NJL-Modell einen expliziten Mechanismus zur Brechung der chiralen Symmetrie, aber kein langreichweitiges Confinementpotential. Die Abwesenheit freier Farbladungen im Endzustand der Hadronisierung wird daher nur vom CDM garantiert.

Zusammenfassend beobachten wir keine wesentliche Abweichung vom thermischen Gleichgewicht während der gesamten Länge unserer mikroskopischen Hadronisierungssimulation. Wenig erstaunlich ist, dass das chemische Gleichgewicht hingegen unvollkommen ist, wie man an den unterschiedlichen Steigungen der Maxwell-Boltzmann-Impulsver-

teilungen für die verschiedenen Clusterarten sehen kann, sowie an den unterschiedlichen Temperaturen der farbigen und weißen Phase. Diese Beobachtung unterstützt die Vermutung, dass während der gesamten Hadronisierungszeit kein Phasengleichgewicht zwischen Quark-Gluon- und Hadronphase erreicht wird; die Gibbs-Kriterien sind nicht anwendbar. Das zur Beschreibung geeignete Bild ist vielmehr ein Gemisch aus hadronischem Gas und Plasma mit additivem statt gleichem Druck.
Unsere Überlegungen in Kapitel 3 zum klassischen Confinement, der Klassifikation der irreduziblen weißen Cluster (IWCs) und die Erzwingung der Hadronisierung durch Entfernung des Quark$-\sigma-$Kopplungsterms wurden durch die Simulationsresultate bestätigt. Auch wenn erkennbar ist, dass ein reiner Abdampfprozess zu langsam ist, erhalten wir im Endzustand nur IWCs — weder größere Plasmacluster noch einzelne Farbladungen bleiben übrig. Damit ist dies das erste funktionierende Hadronisierungsmodell mit expliziter Behandlung des Farbfreiheitsgrades.
Die extrahierten Observablen (Massenspektren, Temperaturen, Flow) sind in guter Übereinstimmung mit anderen einfachen Modellen (Hagedorns Bootstrapmodell, Perkolationsmodell) bzw. experimentellen Daten. Ein erstaunliches Resultat ist dabei, dass aufgrund der Farbabschirmung im Plasma nur wenig Energie im Farbfeld deponiert wird. Das Farbfeld, das sicher die Dynamik des Systems wesentlich mitbestimmt (so verhindert es z. B. das Entkommen farbiger Objekte aus dem Plasmatropfen), scheint nur einen unwesentlichen Energiebeitrag zu leisten.

6.2 Ausblick

6.2.1 Miteinbeziehung von Pionen

Wie bereits in Abschnitt 5.5.4 diskutiert, ist etwa die Hälfte der Gesamtenergie des Endzustandes in σ-Wellen deponiert. Die damit verbundene Erklärungsnot weist auf die Notwendigkeit hin, Pionen in einer späteren, verbesserten Version des Modells miteinzubeziehen. Wegen ihrer höheren Masse würden die σ-Fluktuationen dann in pionische Anregungen umgewandelt. Die Miteinbeziehung der Pionen könnte dadurch geschehen, dass alle σ^2-Terme der CDM-Lagrangedichte zu $\sigma^2 + \vec{\pi}^2$ erweitert werden. Dieser Ansatz würde elementare Pionenfelder statt leichter $q\bar{q}$-Cluster zur Beschreibung der Pionen verwenden. Dies ähnelt konzeptionell dem NJL-Modell und ist auch genauso unbefriedigend. Außerdem ist die übliche chirale Erweiterung des σ-Felds zu einem $(\sigma, \vec{\pi})$-Feld unverträglich mit unserer Interpretation der σ-Wellen als reine Gluonwellen mit einer Glueballmasse von $1400 MeV$. Auf der anderen Seite würde uns ein explizites Pionenfeld ermöglichen, eine langreichweitige Hadron-Hadron-Wechselwirkung zu beschreiben, die momentan im Mo-

dell fehlt.
Eher vielversprechend erscheint es daher, leichte, relativistische Quarks in CDM zu behandeln. Leichte Mesonen, vor allem das Pion, passen nicht in die Stringphänomenologie, deshalb können sie auch nicht als klassische hadronische Cluster beschrieben werden. Falls leichte Quarks ins Modell aufgenommen werden sollen, muss auch ein Paarerzeugungsterm nach dem Schwingermechanismus betrachtet werden (momentan ebenfalls nicht im Modell enthalten). Wir erwarten jedoch nicht, dass diese Erweiterung die Hadronisierungszeit des Modells verkürzt; sie wird höchstens die Resultate des Modells leichter interpretierbar und glaubwürdiger machen.

6.2.2 Studium des Phasenübergangs und andere Projekte

Dieser letzte Abschnitt ist der "Science Fiction" gewidmet: mögliche zukünftige Projekte sollen erörtert werden, die mit dem vorliegenden Simulationsprogramm CDM oder einer weiterentwickelten Version durchgeführt werden könnten.

- Das nach Meinung des Autors interessanteste Projekt ist das Studium des Phasenübergangs. Durch Simulation eines thermalisierten Plasmas und Beobachtung der lokalen Teilchendichte sollte sich in bestimmten Temperaturbereichen eine Phasentrennung in Plasma- und Hadronenphase beobachten lassen. Dieses Projekt könnte allerdings an den numerischen Grenzen scheitern (zu hohe Partonzahlen erforderlich).

- Nachdem von CDM bereits σ-Bags erkannt werden können, ist es auch möglich, Perkolationsstudien zu betreiben. Die Clustererkennungsroutinen sind darauf vorbereitet. In Kombination mit dem ersten Punkt ließe sich auch die interessante Frage untersuchen, ob der Perkolationsphasenübergang bei der gleichen Temperatur/Dichte auftritt wie der QGP-Phasenübergang, bzw. wie man die Definition der Perkolation modifizieren müsste, um beide Phasenübergänge auf den gleichen Temperaturpunkt zu bringen.

- Die Untersuchung der *strangeness distillation* ist von Carsten Greiner vorgeschlagen worden: Aus rein kombinatorischen Gründen wird ein Plasmafeuerball, der eine anfängliche Baryonenasymmetrie aufweist (im Experiment ist dies der Fall!), während der Hadronisierung eine nichtverschwindende Seltsamkeit entwickeln.

- Ein technischer Vorschlag wäre, die Simulationsnumerik nochmals zu überarbeiten. Das vorliegende Programm benötigt viel Rechenzeit, um zu jedem Zeitschritt das Gaußgesetz für die Farbfelder erneut “from scratch” zu lösen. Nun existieren neben dem Gaussgesetz auch die zeitabhängigen Maxwellgleichungen:

$$\begin{aligned} \partial_t \vec{D}^a &= \vec{\nabla} \times (\kappa(\vec{x})\vec{B}^a) - \vec{j} \\ \partial_t \vec{B}^a &= -\vec{\nabla} \times (\vec{D}^a/\kappa(\vec{x})) \quad . \end{aligned} \tag{6.1}$$

Hat man einmal eine genaue Lösung des Gaussgesetzes für eine Ladungskonfiguration gefunden, und ist man in der Lage, ein Updateschema für Partonen und Felder zu formulieren, das die Kontinuitätsgleichung und (6.1) (in einer diskretisierten Fassung) numerisch exakt erfüllt, so würde dieses Schema auch das Gaussgesetz erhalten. Dann bräuchte man nur am Anfang der Simulation einmal das Gaussgesetz lösen und könnte danach Partonen und Felder mit einem schnellen expliziten Zeitschrittverfahren propagieren. Dies könnte das Programm enorm beschleunigen, und es würde auch einen Weg eröffnen, das Farbmagnetfeld in die Simulation aufzunehmen. Damit wäre dies auch ein großer Schritt in Richtung einer echt kovarianten Simulation.

Dazu sind aber zunächst einige schwierige (unlösbare?) Probleme zu lösen: Erstens muss die anfängliche Lösung des Gaußgesetzes noch viel genauer erfolgen, als dies jetzt der Fall ist — dies wird bei unseren Hadronisierungsszenarien zu einem extrem hohen temporären Speicherplatzbedarf (über $1GB$) des Mehrgitteralgorithmus führen. Zweitens müssen die Maxwell- und σ-Feldgleichungen so diskretisiert werden, dass das Gaußgesetz bei der Propagation numerisch exakt erhalten bleibt. Die Wechselwirkung der Teilchen mit den Feldern könnte sich dabei als wesentliches Problem erweisen. Auch das Problem der Spiegelladungen ist erneut zu lösen.

Dieser letzte Vorschlag ist als Anregung zum “Brainstorming” gemeint; es ist dem Autor keineswegs klar, ob und wie die genannten Probleme überwunden werden können.

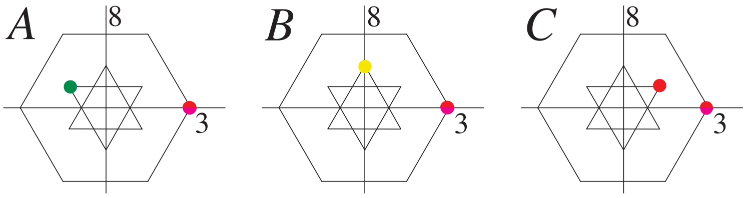
"Acceptance without proof is the fundamental characteristic of Western religion, Rejection without proof is the fundamental characteristic of Western science."
G. ZUKAV, *"The Dancing Wu Li Masters", gefunden in der* UNIX-fortune-*Datenbank.*

A Ergänzung zum Beweis der IWC-Klassifikation

In Abschnitt 3.6 wurde die Vollständigkeit der Klassifikation der irreduziblen weißen Cluster (IWCs) nach Abb. 3.5 bewiesen. Dabei haben wir eine Fallunterscheidung vorgenommen und den Fall gemischter Quark-Antiquark-Gluon-Cluster noch offen gelassen. Diese Lücke soll hier geschlossen werden.

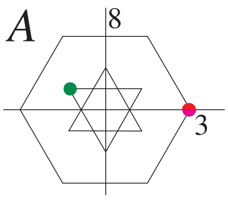
Der folgende Vollständigkeitsbeweis wird indirekt geführt, d. h. wir versuchen, ein irreduzibles weißes Cluster (IWC) zu konstruieren, das in der Klassifikation (Abb. 3.5) nicht erfasst ist. Dabei können wir die Konstruktion immer dann abbrechen, wenn ein von Abb. 3.5 her bekanntes IWC als Subcluster auftaucht.

Ein Beweis im obigen Stil führt zu einem unästhetisch großen Baum von Fallunterscheidungen, den wir im Anhang A vollständig abarbeiten. Ein gemischtes Quark-Gluon-Cluster besteht aus mindestens einem Quark und mindestens einem Gluon. Als Startcluster eines konstruktiven Vorgangs können wir also drei wesentlich verschiedene Fälle unterscheiden[1]:



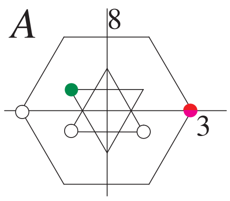
.

Fall A: Wir nehmen als Ausgangssituation das Cluster



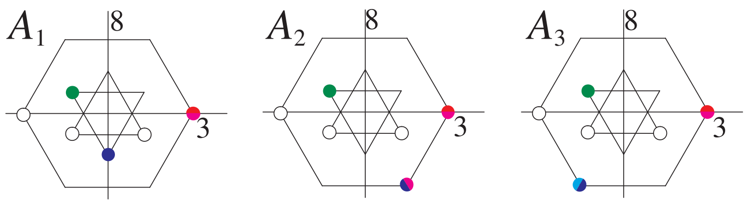
.

Die im folgenden mit offenen Kreisen bezeichneten Positionen würden sofort zu irreduziblen Subclustern führen, die bereits in unserer Klassifikation (Abb. 3.5) berücksichtigt sind:



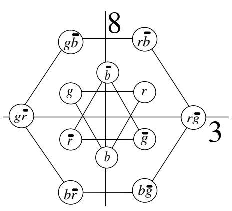
.

Unterhalb der 3-Achse muss aber noch mindestens ein Parton ergänzt werden, damit das Cluster weiß werden kann. Es bleiben also drei Positionen übrig, die wir im nächsten Schritt besetzen können:



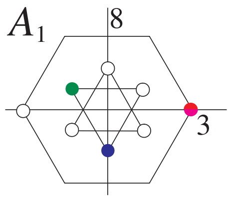
.

[1]Es sei nochmals an das Farbschema nach Abb. 3.2 erinnert, das in dieser Arbeit konsequent verwendet wird:



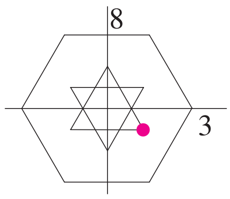
.

Fall A_1: wir besetzen das blaue (untere) Quark. Wieder sind im folgenden alle Positionen, deren Besetzung unmittelbar zu bekannten IWC-Subclustern führen würden, mit offenen Kreisen markiert:



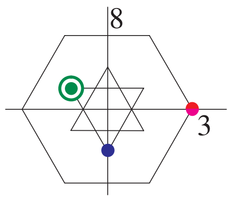
.

Die Gesamtfarbe dieses Clusters ist äquivalent zu einem antigrünen Antiquark,



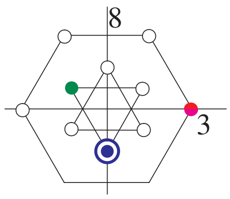
,

also muss zu A_1 noch ein Quark hinzugefügt werden, denn aus Gluonen allein kann man nicht die Farbe grün zusammensetzen. Ergänzen wir ein grünes Quark, so gelangen wir aber zu einem bekannten IWC, nämlich dem $qqqg$-Cluster



.

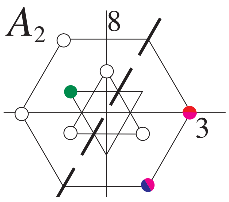
Also keine neue Entdeckung in diesem Fall. Die verbleibende Möglichkeit im Fall A_1, die erneute Besetzung des blauen Quarks, führt zu folgender Situation:



.

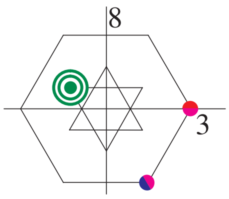
Es muss noch mindestens ein Parton oberhalb der 3-Achse ergänzt werden, wenn das Cluster einmal weiß werden soll. Allerdings kann keine einzige Position oberhalb der 3-Achse noch besetzt werden, ohne dass ein nach Abb. 3.5 bekanntes IWC als Subcluster entstehen würde (Das grün-antiblaue Gluon oben links beispielsweise ist ausgeschlossen, da es zu einer der beiden bekannten und klassifizierten $qqqgg$-Cluster führen

würde). Also finden wir im Fall A_1 kein IWC, das nicht bereits in Abb. 3.5 enthalten gewesen wäre.
Fall A_2: wir besetzen im ersten Schritt nicht das blaue (untere) Quark, sondern das blau-antigrüne Gluon (unten rechts):



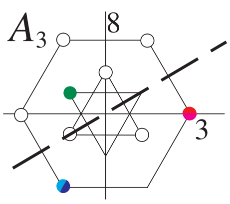
.

Hier sind wieder alle Positionen mit offenen Kreisen markiert, deren Besetzung im nächsten Schritt unmittelbar auf bekannte IWCs führen würde. Links der gestrichelten Linie müssen noch Partonen hinzugefügt werden, um das Cluster weiß zu machen (momentane Farbe ist zweifach antigrün). Es bleibt uns also nichts anderes übrig, als das grüne Quark erneut zu besetzen. Dieses Argument gilt dann gleich noch einmal, also müssen wir das grüne Quark ein drittes Mal besetzen. Dies führt auf das bekannte $qqqgg$-Cluster

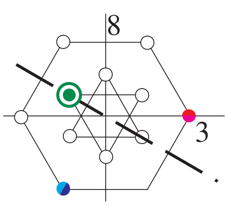


.

Also nichts Neues im Fall A_2.
Fall A_3: Mittlerweile ist die Technik geläufig. Wir besetzen im ersten Schritt das blau-antirote Gluon (unten links), was zu

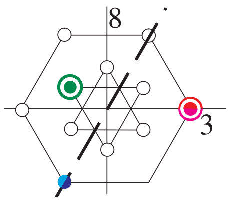


führt. Oberhalb der gestrichelten Linie müssen noch Partonen besetzt werden, was uns zur erneuten Besetzung des grünen Quarks zwingt:

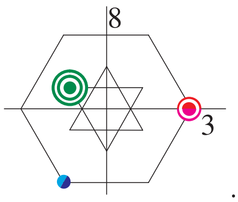


.

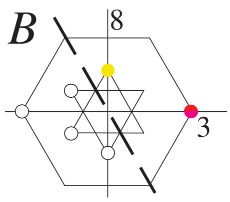
Wieder muss oberhalb der gestrichelten Linie Farbe ergänzt werden, was uns zur Doppelbesetzung des rot-antigrünen Gluons (rechts) zwingt.



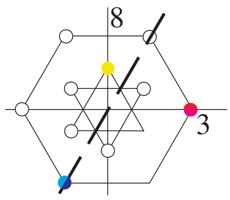
Jetzt fehlt links der gestrichelten Linie Farbe, und wir dürfen nur noch das grüne Quark besetzen, das bereits doppelt besetzt ist.



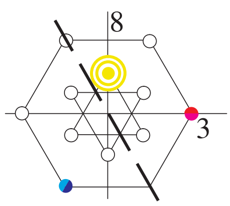
Dies ist ein bekanntes IWC, nämlich das baryonische $qqqggg$-Cluster. Somit führt auch Fall A_3 nicht zu neuen IWCs. Damit ist Fall A abgehandelt.
Nun zu Fall B:



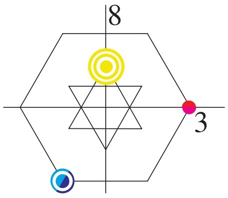
Hier sind sowohl diejenigen Positionen mit offenen Kreisen markiert, deren Besetzung unmittelbar zu IWC-Subclustern führen würde, als auch solche, deren Besetzung bereits in Fall A abgehandelt wurde. Da links der gestrichelten Linie noch Farbe ergänzt werden muss, bleibt uns nichts übrig, als ein Gluon hinzuzufügen:



Links der gestrichelten Line muss noch ein Feld besetzt werden, was uns zur erneuten Besetzung des Quarkfelds zwingt. Das gleiche Argument gilt sogar ein zweites Mal, was uns zum Cluster

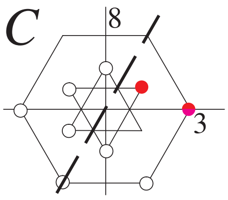


führt. Da links der Linie noch Farbe fehlt, gelangen wir zwangsläufig zum Cluster



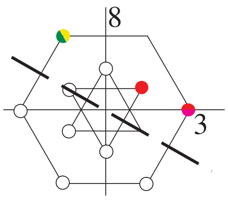
,

das bekannte $\bar{q}\bar{q}\bar{q}ggg$-Cluster. Also ist Fall B auch abgehakt.
Schließlich Fall C:



.

Auch hier sind wieder offene Kreise dort gesetzt, wo wir IWC-Cluster erhalten würden oder wo eine Besetzung zu einem in A oder B abgehandelten Fall führen würde. Links der Linie muss Farbe ergänzt werden. Zwangsläufig besetzen wir das grün-antiblaue (obere) Gluon:



und sind fertig, da unterhalb der gezeichneten Linie noch Farbe fehlt, aber keine Position mehr zur Verfügung steht. Damit ist Fall C erledigt und der Vollständigkeitsbeweis der Klassifikation nach Abb. 3.5 abgeschlossen.

B Wegweiser durch das Programm

B.1 CDM, MUESLEA, DISP und GUI

B.1.1 Programmaufruf und Kommunikationswege

Abb. B.1 zeigt ein “Blockschaltbild” des CDM-Projekts. Wir gehen im folgenden auf die verschiedenen Kommunikationswege ein.
Der Benutzer kommuniziert per Maus und Tastatur mit der grafischen Oberfläche cdm.gui (ganz links), die in Perl/Tk programmiert ist. Einige Menüs der Oberfläche (“Panel”) sind in Abb. B.2 gezeigt. Links sieht man die Menüs der oberen Menüleiste.

- Unter *File* kann man Files verschiedener Formate speichern, MDF-Files (proprietäres CDM-Format) lesen oder eine umfangreiche Auswertung veranlassen.

- *Proj* ist das leicht erweiterbare Projektmenü, das Berechnungen startet. Jeder Eintrag korrespondiert zu einer Projektroutine, die den jeweiligen Simulationslauf startet.

- Im *Sim*-Menü findet man einige globale Flags, die das Verhalten der Simulation wesentlich beeinflussen. So veranlasst *ClusterCondensation*, falls gesetzt, die Hadronisierung von IWCs. Falls dieses Flag nicht gesetzt ist, werden immer nur Partonen propagiert und niemals hadronisiert. *ContinueOldRun* ist ein Flag, das entsprechend programmierte Projektroutinen anweist, nach Ausgabefiles, die bei einem evtl. unterbrochenen früheren Lauf erzeugt wurden, zu schauen und den früheren Lauf fortzusetzen.

- Im *Disp*-Menü findet man globale Flags, die das Display steuern. Es gibt acht verschiedene Darstellungsarten des Simulationsszenarios, die alle auch simulation angewählt sein können (dann dauert die Berechnung eines Bildes natürlich entsprechend länger):

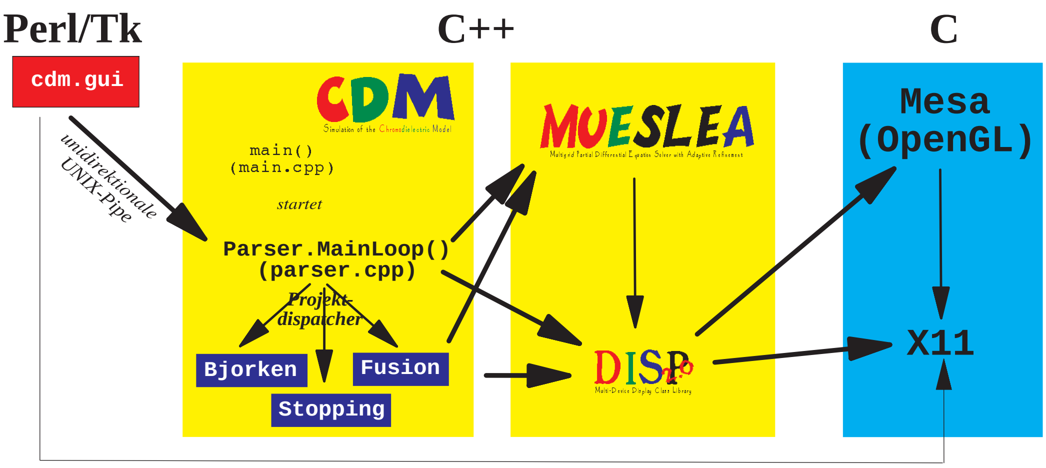


Abbildung B.1: Kommunikationswege in CDM.

Abbildung B.2: Einige Menüs des Bedienungspanels cdm.gui.

- **X1**, *Cross Section View 1*, zeigt ein Schnittbild des Szenarios mit Äquipotentiallinien, Gradienten oder dem Feldwert auf der senkrechten Achse.
- **X2**, *Cross Section View 2*, ist ein paralleles Display der gleichen Art. So kann man zwei Felder gleichzeitig sehen (z. B. ϕ^3 und ϕ^8).
- **EqS**, *Equipotential Surfaces*, zeigt Äquipotentialflächen von Feldern. Angewandt auf das σ-Feld kann man so Bilder von Bags erzeugen.
- **V1**, *Full Volume Gradient 1*, zeigt einen Feldgradienten als Vektorfeld im gesamten Volumen.
- **V2**, *Full Volume Gradient 2*, ist ein paralleles Display der gleichen Art.
- **P**, *Particles & Hadrons*, zeigt die Teilchen und Hadronen der Simulation als farbige Kugeln.
- **Str**, *Structures*, zeigt Programmstrukturen wie das verfeinerte Simplexnetz oder die Matrizen.
- **Db**, *Debugging*, ist in parser.cpp zu definieren und stellt eine einfache Möglichkeit dar, "mal schnell" ein eigenes Display zu definieren.

Zu jeder Displayart gibt es ein Blatt im oberen Notebook des Panels, in dem weitere Optionen einstellbar sind. Abb. B.2 zeigt rechts das **X1**-Blatt mit den Menüs *Field* und *Options*. Im *Field*-Menü wird das Feld ausgesucht, auf das das Display zugreifen soll, wobei die Felder *sigma*, *sigmat*, *kappa*, *phi3* und *phi8* die offensichtlichen Bedeutungen haben. *F2* ist $F^a_{\mu\nu}F^{\mu\nu a}$, was in unseren Approximationen einfach $(\vec{E}^a)^2$ gleicht; *kappaF2* ist dementsprechend gleich $\kappa(\vec{E}^a)^2$. Die *epsilon*-Einträge sind Energiedichten. Während eines Simulationslaufes werden noch einige weitere Einträge nützlich: So ist unter *aux* das Feld $\nabla^2\sigma$ abzurufen, *res* und *res0* verraten etwas über das Residuum (Fehler) der Mehrgitterlösung. Das *Options*-Menü regelt, was gezeigt wird, wobei der Eintrag *Gradient* am wichtigsten ist: Wenn er gesetzt ist, wird nicht das Feld, sondern der Feldgradient gezeigt (die einzig sinnvolle Wahl für ϕ^a). Gleichzeitig sollte aber dann der Punkt *ShowDnotE* im *Display*-Menü angewählt sein, damit zum Gradienten auch noch κ multipliziert wird. Mit diesen Einstellungen sieht man dann das $\vec{D}^a$-Feld.

Das Panel kommuniziert über eine UNIX-Pipe mit dem in C++ geschriebenen Hauptprogramm, allerdings nur unidirektional. Dazu schreibt es für jeden Mausklick und jede Schieberbewegung eine Zeile nach STDOUT, die beschreibt, was der Benutzer gerade wünscht. Für manche wenigen

Buttons werden auch mehrere Zeilen geschrieben (auch zur Initialisierung). Das Executable des Hauptprogramms heißt standardmäßig go, so dass ein vollständiger Programmaufruf unter UNIX einfach lautet:

```
cdm.gui | go - >run.log
```

Der STDOUT der grafischen Oberfläche wird in den STDIN von go "gepipet". Man kann auch in zwei Schritten vorgehen: erst mit

```
cdm.gui >chore
```

ein Scriptfile chore erzeugen, das man im zweiten Schritt übergibt:

```
go chore >run.log
```

oder

```
go - <chore >run.log   .
```

Auch die Kommunikation über eine *named pipe* ist möglich: Man erzeugt eine *named pipe* unter Linux mittels

```
mkfifo .cdmcmd   .
```

und kann dann zwei getrennte Prozesse für grafische Oberfläche und das Hauptprogramm starten:

```
cdm.gui >>.cdmcmd \& go - <.cdmcmd >run.log
```

In der Tat liest go, wenn es ohne jeden Parameter aufgerufen wird, seine Steuerkommandos von /.cdmcmd.
Das Executable go schreibt ein Logfile nach STDOUT, das mittels ANSI-Steuerzeichen sogar farbig ist. Mit dem Perl-Kommando

```
perl -pe 's/\\033[^m]*m//g;' <run.log >runbw.log
```

kann man die Farben übrigens leicht entfernen. In diesem Logfile ist u. a. auch ein Echo der von cdm.gui gegebenen Kommandos enthalten. Das Tool makechore kann nun die Kommandos der letzten Sitzung aus einem run.log-File extrahieren und daraus ein Inputfile chore produzieren, mit dem man go erneut starten kann. Dies erweist sich oft als praktisch.
Der Sourcecode des C++-Programms CDM, der zum Executable go kompiliert wird, residiert normalerweise im Verzeichnis /cdm. Die Unterverzeichnisse mueslea und disp dieses Verzeichnisses sind für die Libraries MUESLEA und DISP bestimmt, die von go zur Laufzeit hinzugelinkt werden. MUESLEA ist dabei der Mehrgitter-Differentialgleichungslöser, DISP eine universelle Displayklasse.
Der Hauptteil CDM, der kompiliert also das Executable go ergibt, ist die eigentliche Simulation des chromodielektrischen Modells. Die Klasse

CommandStreamParser parser.h liest den Output der grafischen Oberfläche und veranlasst die Ausführung der Kommandos. Dazu existieren für jede Statusvariable des Panels eine Spiegelvariable in der Klasse CommandStreamParser. Der Output des Panels wird gelesen und die Variablen entsprechend geschrieben oder Simulationsläufe gestartet. Die Methode CommandStreamParser::MainLoop wird von main() kurz nach dem Start von go aufgerufen und stellt die Hauptschleife des Programms dar, die Zeile für Zeile abarbeitet, was vom chore-Skript oder vom laufenden cdm.gui gelesen wird.
Es gibt dabei spezielle Kommandos, sog. "Projekte", die längere Rechnungen auslösen. Momentan sind die folgenden Projekte definiert:

- MeasureStringTension
- StringStability
- MovingSolitonStability
- WaveEquationStability
- YoyoModeStability
- BjorkenQGPTubeHadronization
- StoppedFireballHadronization
- TransparentFireballHadronization
- StrangenessDistillation
- Distiller_AGS
- Distiller_SPS
- PhaseSeparation
- SingleQuark
- StringFusion

Zu jedem Projekt gehören folgende Merkmale (zu beachten, wenn man ein neues Projekt anfügen möchte!):

- Ein Eintrag in der @projects-Tabelle in cdm.gui. Dies erzeugt einen entsprechenden Punkt im Pull-Down-Menü der grafischen Oberfläche.

- Ein Eintrag in der projects()-Tabelle in main.cpp. Für diesen Eintrag ist es nötig, ein Outputverzeichnis zu definieren (sinnvollerweise für jedes Projekt ein anderes, man schaue in main.cpp für Beispiele), das beim Start des Projekts bereits vorhanden und beschreibbar sein muss. Dateipfade in CDM sind grundsätzlich vom Homeverzeichnis des Users aus anzugeben. Ebenfalls essentiell für einen Projekteintrag ist die eigentliche Projektfunktion zu definieren (siehe projects.h). Man betrachte als Vorlage die kurzen Projekte, die direkt in main.cpp definiert sind.

Die Projektroutine rechnet dann und benutzt dabei Klassen von CDM, allen voran natürlich die übergebene CommandStreamParser-Instanz, die "als Spielplatz" das Cube-Gebiet Ω enthält. Dieses Gebiet enthält ein Grid für das σ-Feld, ein Domain für das Farbfeld sowie eine ParticleList und eine HadronizedClusterList für Partonen bzw. Hadronen. Die Projektroutine benutzt, meist über CommandStreamParser-Methoden, die Libraries MUESLEA und DISP. DISP beruht auf der X11 bzw. Mesa/OpenGL-Library und kann sowohl am Bildschirm Grafiken anzeigen als auch EPS- oder PPM/JPEG/GIF-Bilder und sogar MPEG-Filme erzeugen.
Man orientiere sich bei der Programmierung neuer Projekte an den bereits vorhandenen.

B.1.2 Hinweise zur Benutzung und Weiterarbeit

Es gibt einige Voraussetzungen, die erfüllt sein *müssen*, damit man CDM erfolgreich übersetzen und benutzen kann. Andere Voraussetzungen *sollten* erfüllt sein, damit man alle Features benutzen kann.

- Ein lauffähiges UNIX-System mit grafischer Oberfläche X11. Der Autor bevorzugt GNU/Linux, da fast alle weiteren hier erwähnten Tools in jeder Linux-Distribution kostenlos mitgeliefert sind und mühelos, d. h. ohne Kompilieren, installiert werden können[1].

- Ein C++-Compiler. Da C++ zwar bereits standardisiert ist, die momentan existierenden Compiler aber noch nicht wirklich dem Standard folgen, kann leider nicht dafür garantiert werden, dass das Programm mit jedem C++-Compiler zusammenarbeitet. Der GNU g++, Version 2.7.2.1, der *Free Software Foundation* kompiliert das Programm ohne eine einzige Warnung; dies mag aber bereits bei Version 2.8.2 anders sein. (Von der Verwendung der Zwischenversionen wird abgeraten.). Der fantastische GNU-Debugger gdb in Verbindung mit der grafischen Oberfläche DDD (von Dorothea

[1] Die Linux-Distribution der Firma S.u.S.E., Fürth, enthält in der Version 6.0 alle benötigten Tools (und noch viel mehr) bereits vorkompiliert auf CD.

Lütkehaus und Andreas Zeller aus Braunschweig) sei auch empfohlen. GNU-Tools findet man unter www.fsf.org bzw. einem der vielen Mirrors.

- Eine installierte Mesa oder OpenGL-Library. Getestet wurde das Programm mit Mesa 2.4; höhere Versionen sollten auch funktionieren, und theoretisch sollte das Programm auch mit SGI's Original-OpenGL zusammenarbeiten. Mesa wurde von Brian Paul programmiert und unterstützt auch Hardwarebeschleunigung, (www..mesa3d.org).

- Larry Walls und Nick Ing-Simmons Perl/Tk. Dabei sollte man Perl sowieso installiert haben (unentbehrlich für die Datenaufbereitung, Systemadministration und den "quick hack" zwischendurch); den Zusatz Tk erhält man vom Autor Nick Ing-Simmons (man suche nach diesem Autor im CPAN unter www.perl.org/CPAN). Die Tk-Version 400.202 unter Perl 5.004 funktioniert gut. Auch das Modul Tk::FileDialog von Brent Powers wird benötigt; hier kommt gegenwärtig die Version 1.3 zum Einsatz.

- Jef Poskanzers Grafikkonvertierungspaket netpbm, das unzählige Konvertierer von ppmtogif bis hin zu pnmtops enthält sowie Nachbearbeitungsprogramme für die Kommandozeile ppmbrighten, pnmcrop, etc.

- Die JPEG-Codierungssoftware der Independent JPEG Group, insbesondere die Programme cjpeg und djpeg.

- Die MPEG-Codierungssoftware von Andy Hung (havefun.stanford..edu:pub/mpeg/MPEGv1.2.tar.Z).

- ACE/gr, auch bekannt als xmgr, ein 2D-Grafikprogramm zum Erstellen von Diagrammen. CDM ist in der Lage, Tabellen so in Datenfiles zu schreiben, dass der einfache Aufruf

  ```
  xmgr -nxy -source stdin <xyz.dat
  ```

  bereits zu einer fertigen grafischen Darstellung mit Legende und Überschrift in Publikationsqualität führt.

- Das PSUtils-Paket von Angus Duggan et al.

- Ein CVS-Server ist zwar nicht notwendig, aber sehr nützlich, um die weitere Entwicklung des CDM-Projektes zu betreiben.

Zur Kompilation von CDM sind folgende Schritte zu unternehmen: Zunächst muss das Perl-Skript gmake im aktuellen Suchpfad enthalten sein. Vorsicht: Mit gmake ist ein CDM-Tool gemeint, nicht etwa das gleichnamige GNU-Tool! Nun kann im Verzeichnis /cdm das configure-Skript aufgerufen werden:

./configure

oder (fürs Debuggen)

./configure debug

Das configure-Skript erkennt momentan leider nur am Hostnamen den Typ der Maschine, auf dem das Programm übersetzt werden soll (eleganter wäre die Verwendung von uname statt hostname). Deshalb sind Anpassungen an configure notwendig, um es auf einer neuen Maschine lauffähig zu machen. Auch die Skripte mueslea/configure und disp/configure müssen entsprechend angepasst werden. Diese Skripte sind aber recht kurz und einfach zu ändern.
Nach der Konfiguration existeren die Files makefile, mueslea/makefile und disp/makefile.

make go

erzeugt nun das CDM-Executable. Man kann die Makefiles und alle Objektcodes mittels

make distclean

wieder löschen und erneut configure aufrufen.

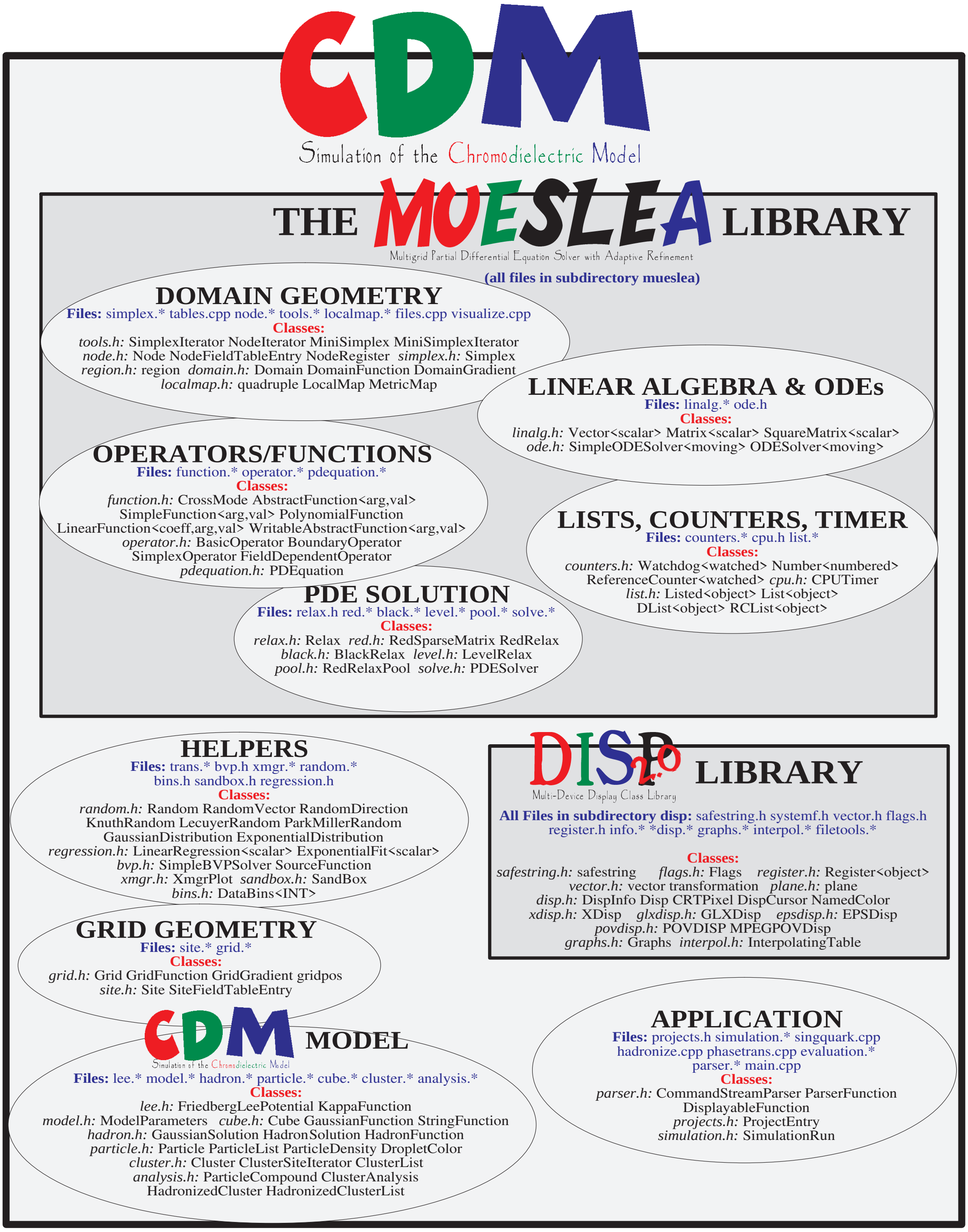


Abbildung B.3: In diesem Bild sind alle C++-Klassen des Programms und alle Programmfiles aufgeführt. Erkennbar ist die CDM-Gliederung in Hauptteile (CDM, MUESLEA, DISP) und Aufgabenbereiche (ovale Bereiche).

B.2 Klassenstruktur des Programms CDM

Abb. B.3 zeigt einen groben Überblick über die Klassenstruktur des Programms. Der objektorientierte Programmierstil zeichnet sich durch die gemeinsame Kapselung von Algorithmen und zugehörigen Datenstrukturen aus. Dadurch werden die meisten Algorithmen "lokal trivial", d. h., die Komplexität des Programms zeigt sich erst im Zusammenwirken der vielen Klassen, wobei jede einzelne Klasse für sich relativ überschaubar ist. Die Gliederung des Programms ist dadurch stark horizontal; die Klassen stehen einfach nebeneinander, wenn man von der Vererbungshierarchie einmal absieht. Um dem Benutzer den Einstieg ins CDM-Programm zu erleichtern, folgt eine vollständige, alphabetisch sortierte Referenzliste aller Klassen und ihrer wichtigsten Klassenkomponenten/-methoden. Die Angaben der Komponenten und Methoden sind aus Platzgründen unvollständig; nur die wirklich zentralen Routinen werden hier erwähnt.

template AbstractFunction<arg,val> (mueslea/function.h).

• *Basisklasse für* DomainGradient *(S. 188)*, GaussianFunction *(S. 190)*, GridGradient *(S. 191)*, HadronFunction *(S. 191)*, KappaFunction *(S. 192)*, LinearFunction<coeff,arg,val> *(S. 192)*, ParserFunction<arg,val> *(S. 194)*, ParticleDensity *(S. 195)*, PolynomialFunction *(S. 195)*, SimpleFunction<arg,val> *(S. 198)*, WritableAbstractFunction<arg,val> *(S. 200)*.

Abstraktes Basisklassentemplate zur Kapselung beliebiger Funktionen mit Argument der Klasse arg *und Wert der Klasse* val. *Die Methode* clone *ist ein virtueller Konstruktor [77].*

class BasicOperator (mueslea/operator.h).

• *Basisklasse für* BoundaryOperator *(S. 186)*, FieldDependentOperator *(S. 189)*, SimplexOperator *(S. 198)*.

Abstrakte Basisklasse zur Kapselung beliebiger Operatoren (Funktionen, die auf ein Simplex eine 10×10*-Matrix von Operatormatrixelementen zurückgeben).*

class BlackRelax (mueslea/black.h).

• *Abgeleitet von* Relax *(S. 197)*, Watchdog<BlackRelax> *(S. 200)*.

Verantwortlich für die schwarze Relaxation (S. 113) eines einzelnen Knotens relax. *Der Koeffizient dieses Knotens ist in* relaxcoefficient *gespeichert, der Koeffizientenvektor der Nachbarknoten im Vektor* neighline.

class BoundaryOperator (mueslea/operator.h).

• *Abgeleitet von* BasicOperator *(S. 186)*.

Ein Operator (Funktion, die auf ein Simplex eine 10×10*-Matrix von Operatormatrixelementen zurückgibt) für Randelemente. Arbeitet mit finiten Elementen auf den Randflächen (Simplexflächen, bei denen in allen drei Knoten das Flag* boundary *gesetzt ist).*

class Cluster (cluster.h).
• *Abgeleitet von* Flags *(S. 189).*

Cube::PlasmaClusters *liefert eine* ClusterList *zurück. Cluster sind dabei entweder auf dem* Grid *durch* Site$\rightarrow$ $*$field¡criterion *definiert (Flag* LESSTHAN*), oder einfach durch eine Liste von Teilchen (Flag* PARTICLES*) definiert. Im ersten Fall zeigt* bagsite *auf eine Position innerhalb des Clusters. Jedes Cluster hat eine eindeutige Nummer.*

class ClusterAnalysis (analysis.h).

Die Routine Analyze nimmt eine Instanz von ParticleCompound *entgegen (das ist ein Schema nach Abb. 3.2 mit beliebigen Besetzungszahlen) und zerlegt sie auf die einfachste mögliche Weise in irreduzible weiße Cluster. Ist das übergebene Schema ein irreduzibles weißes Cluster, so wird dessen Nummer (nach dem Code der Klasse* ParticleCompound*) zurückgegeben, ansonsten 0.*

class ClusterList (cluster.h).
• *Abgeleitet von* List<Cluster> *(S. 192).*

Liste von Cluster*-Instanzen.*

class ClusterSiteIterator (cluster.h).

Iterator über alle Gitterpositionen, die zu einem Cluster *gehören. Verwendung dieses Iterators erfolgt über Makros wie z. B.* forAllClusterSites. *Es ist auch möglich, über die Oberfläche eines Clusters zu iterieren.*

class CommandStreamParser (parser.h).

Parser der Kommandosprache, die von cdm.gui *ausgegeben wird. Enthält die Hauptschleife des Programms. Siehe Abschnitt B.1.1.*

class CPUTimer (mueslea/cpu.h).

Klasse zur Messung verbrauchter CPU-Zeit.

class CrossMode (mueslea/function.h).

Sollte besser ein namespace *denn eine Klasse sein. Enthält nur eine Liste von Konstanten zur Spezifikation von Darstellungstypen von Schnittbildern.*

class CRTPixel (disp/disp.h).

Klasse repräsentiert ein "Pixel" auf dem Bildschirm (CRT=Cathode Ray Tube), also ein Paar ganzzahliger Koordinaten.

class Cube (cube.h).
• *Abgeleitet von* Domain *(S. 188),* Grid *(S. 191).*

Unser Spielplatz. Enthält alle Felder und Teilchenlisten. In CDM *existiert stets genau eine* Cube*-Instanz (obwohl das keine Beschränkung darstellt), deren Name immer* Ω *ist.*

template DataBins<INT> (bins.h).

Klassentemplate zum Binning von statistischen Daten. Passt die Feldgröße selbständig dem Datenintervall an.

class DifferentialSpectrumEntry (evaluation.cpp).

Einfache Hilfsklasse zum Aufbau einer Tabelle, nur lokal in evaluation.cpp *von Bedeutung.*

class Disp (disp/disp.h).

- *Abgeleitet von* DispInfo *(S. 188).*
- *Basisklasse für* EPSDisp *(S. 189),* POVDisp *(S. 196),* XDisp *(S. 200).*

*Definiert eine Programmierschnittstelle zum Zeichnen von 2D- und 3D-Grafiken auf Bildschirm (*X11 *und* OpenGL*), Postscriptfiles und POV-Ray-Szenenfiles (fotorealistische Raytracingbilder). Wichtigste Grafikprimitive sind unten aufgeführt.*

class DispCursor (disp/disp.h).

2D- und 3D-Cursor der Displayklasse.

class DispInfo (disp/disp.h).

- *Abgeleitet von* Flags *(S. 189).*
- *Basisklasse für* Disp *(S. 188).*

Klasse, die die abstrakte Spezifikation eines Displays (Format und Projektion/ Perspektive) speichert. Dient hauptsächlich dazu, aus dem Display einer Sorte eines einer anderen Sorte zu konstruieren. Enthält auch alle Routinen zur perspektivischen Projektion.

class DisplayableFunction (parser.h).

Dient als Rückgabewert der Funktion CommandStreamParser::GetDisplayableFunction. *Fasst eine abstrakte Funktion mit einer Displayfarbe zusammen.*

template DList<object> (mueslea/list.h).

- *Abgeleitet von* List<object> *(S. 192).*

Listentemplate. Die Elemente werden bei Aufruf des Listendestruktors gelöscht (deshalb das D *in* DList*).*

class Domain (mueslea/domain.h).

- *Abgeleitet von* Flags *(S. 189),* List<Simplex> *(S. 192),* Number<Domain> *(S. 194),* Watchdog<Domain> *(S. 200).*
- *Basisklasse für* Cube *(S. 187).*

Eine Liste von Simplices („Simplexwald") und damit die Basisstruktur des Mehrgitterlösers MUESLEA. *Felder sind in* Node*-Instanzen gespeichert, auf die die Simplices verweisen. Feldwerte außerhalb des abgedeckten Gebiets werden als identisch mit den Vakuumwerten angenommen, die in der einen* Node*-Instanz* vacuumnode *gespeichert sind.*

class DomainFunction (mueslea/domain.h).

- *Abgeleitet von* WritableAbstractFunction<vector,float> *(S. 200).*

Funktion, die auf den Knoten einer Domain*-Instanz gespeichert ist.*

class DomainGradient (mueslea/domain.h).

- *Abgeleitet von* AbstractFunction<vector,vector> *(S. 186).*

Vektorwertige Funktion, die als Gradient einer DomainFunction *definiert ist.* DomainFunction *besitzt sogar eine Methode, die den Gradient als* DomainGradient-*Instanz zurückgibt.*

template DRegister<object> (mueslea/register.h).

• *Abgeleitet von* Register<object> *(S. 197).*

Registertemplate (Array mit anpassbarer Größe). Die Elemente werden bei Aufruf des Registerdestruktors gelöscht (deshalb das D *in* DRegister*).*

class DropletColor (particle.h).

Speichert die Gesamtfarbe eines Partonenclusters und kann (zwecks Initialisierung eines farbneutralen QGP-Feuerballs) eine "Empfehlung" geben, welche Farbe man hinzufügen sollte, um den Plasmatropfen weiß zu machen.

class EPSDisp (disp/epsdisp.h).

• *Abgeleitet von* Disp *(S. 188).*

Displayklasse zur Generierung von Postscriptfiles. Diese Postscriptfiles enthalten die gesamte 3D-Information und sind sogar wieder einlesbar (z. B. zur Änderung der Perspektive).

template ExponentialDistribution<random> (random.h).

• *Abgeleitet von* random.

Klassentemplate zur Erzeugung von Zufallszahlen mit exponentieller Verteilung. Die Basisklasse random *hat die Aufgabe, gleichmäßig verteilte Zufallszahlen zu erzeugen.*

template ExponentialFit<scalar> (regression.h).

• *Abgeleitet von* LinearRegression<scalar> *(S. 192).*

Exponentielle Regression (Fit). Berücksichtigt korrekt die statistischen Gewichte; dies ist nicht einfach eine lineare Regression an den logarithmierten Datenpunkten, sondern besser.

class FieldDependentOperator (mueslea/operator.h).

• *Abgeleitet von* BasicOperator *(S. 186).*

Operator (Funktion Simplex $\to 10\times 10$*-Matrix), in dessen Berechnung noch eine Funktion eingeht. Bestes Beispiel im Kontext dieser Arbeit ist der Operator* $\vec{\nabla}\cdot(\kappa(\vec{x})\vec{\nabla}\ldots)$.

class Flags (disp/flags.h).

• *Basisklasse für* Cluster *(S. 187),* DispInfo *(S. 188),* Domain *(S. 188),* HadronizedCluster *(S. 191),* Node *(S. 194),* PDEquation *(S. 195),* Relax *(S. 197),* Simplex *(S. 198),* Site *(S. 198),* XmgrPlot *(S. 200).*

Sehr zentrale und gleichzeitig extrem simple Klasse. Verwaltet eine Menge von max. 32 (Bitlänge eines Integers in heute üblichen C++*-Compilern) Flags. Funktionen zur Abfrage und Setzen/Löschen von Flags werden bereitgestellt.*

class FontTable (disp/xdisp.h).

Krücke zum Fonthandling unter X11. *Liefert zu gegebener Fontgröße den Namen eines* X11-*Fonts zurück. Die Tabelle muss natürlich entsprechend initialisiert werden.*

class FontTableEntry (disp/xdisp.h).

Eintrag in FontTable.

class FriedbergLeePotential (lee.h).

- *Abgeleitet von* PolynomialFunction *(S. 195).*

Das Friedberg-Lee-Potential

$$\begin{aligned} U(\sigma) &= B + a\sigma^2 + b\sigma^3 + c\sigma^4 \\ &= B\left(\frac{\sigma}{\sigma_{vac}} - 1\right)^2 \left[1 + 2\left(\frac{\sigma}{\sigma_{vac}}\right) + \zeta\left(\frac{\sigma}{\sigma_{vac}}\right)^2\right] \end{aligned}$$

erfüllt die Gleichungen

$$\begin{aligned} U(0) &= B \\ U'(0) &= 0 \\ U'(\sigma_{vac}) &= 0 \\ U''(\sigma_{vac}) &= m^2_{glue} \quad . \end{aligned}$$

template GaussianDistribution<random> (random.h).

- *Abgeleitet von* random.

Klassentemplate zur Erzeugung von Zufallszahlen mit Gaußverteilung. Die Basisklasse random *hat die Aufgabe, gleichmäßig verteilte Zufallszahlen zu erzeugen.*

class GaussianFunction (cube.h).

- *Abgeleitet von* AbstractFunction<vector,float> *(S. 186).*
- *Basisklasse für* StringFunction *(S. 199).*

Eine dreidimensionale normierte Gaußverteilung.

class GaussianSolution (hadron.h).

- *Abgeleitet von* HadronSolution *(S. 191).*

Löst die σ-Feldgleichung für gegebene gaußsche skalare Dichte $\rho_S(\vec{x})$. Benutzt die Klasse SimpleBVPSolver *zur Lösung des sphärisch symmetrischen Randwertproblems. Ist bei fehlender Quark-sigma-Kopplung überflüssig.*

class GaussianSourceFunction (hadron.cpp).

- *Abgeleitet von* SourceFunction *(S. 198).*

Quellterm für GaussianSolution.

class GLXDisp (disp/glxdisp.h).

- *Abgeleitet von* XDisp *(S. 200).*

Disp-*Derivat zur grafischen Ausgabe mit* OpenGL-*Qualität (empfohlen ist die Verwendung des freien* OpenGL-*Clones* Mesa*).*

class Graphs (disp/graphs.h).

Klasse diente ursprünglich zum Debugging. Plottet mehrere Graphen in ein X11*-Fenster.*

class Grid (grid.h).

- *Basisklasse für* Cube *(S. 187).*

Starres kartesisches Gitter. Dient in CDM *zur Speicherung und Propagation des σ-Feldes. Felder sind in* Site*-Instanzen gespeichert.*

class GridFunction (grid.h).

- *Abgeleitet von* WritableAbstractFunction<vector,float> *(S. 200).*

*Funktion, die auf den Gitterpositionen (*Site*-Instanzen) einer* Grid*-Instanz gespeichert ist.*

class GridGradient (grid.h).

- *Abgeleitet von* AbstractFunction<vector,vector> *(S. 186).*

Vektorwertige Funktion, die als Gradient einer GridFunction *definiert ist.* GridFunction *besitzt sogar eine Methode, die den Gradient als* GridGradient*-Instanz zurückgibt.*

class gridpos (grid.h).

Instanzen dieser Klasse benennen eine Site*-Instanz innerhalb eines Gitters (*Grid*-Instanz). Im Prinzip einfach ein dreikomponentiger ganzzahliger Vektor.*

class HadronFunction (hadron.h).

- *Abgeleitet von* AbstractFunction<vector,float> *(S. 186).*

Skalares σ-Feld in der Umgebung eines Hadrons. Überflüssig in Abwesenheit der direkten Quark-σ-Kopplung.

class HadronizedCluster (analysis.h).

- *Abgeleitet von* Flags *(S. 189),* ParticleCompound *(S. 195).*

Repräsentiert ein hadronisiertes irreduzibles weißes Cluster (IWC) der Simulation. HadronizedCluster*-Instanzen sind damit die Hadronen der Simulation.*

class HadronizedClusterList (analysis.h).

- *Abgeleitet von* List<HadronizedCluster> *(S. 192).*

Liste von HadronizedCluster*-Instanzen mit Methode zur Propagation.*

class HadronSolution (hadron.h).

- *Basisklasse für* GaussianSolution *(S. 190).*

Abstrakte Basisklasse für Lösungen der $sigma$-Feldgleichung in der Umgebung eines Hadrons. Mit Fehlen des direkten Quark-σ-Kopplungsterms obsolet.

class InfoStream (disp/info.h).

Diese Klasse wird hauptsächlich in Zusammenhang mit den info(...)*-Makros benutzt. Dient dazu, ein übersichtliches Logfile der Programmaktivitäten zu erzeugen, wobei das Detail der ausgegebenen Nachrichten (Info-Level) wählbar ist.*

template InterpolatingTable<Rvector> (disp/interpol.h).
- *Basisklasse für* SimpleBVPSolver *(S. 197).*

Interpoliert linear, quadratisch oder kubisch zwischen gegebenen Stützstellen. Die Werteklasse Rvector *muss dabei ein Vektorraum über* float *sein, kann aber ansonsten beliebig gewählt werden.*

class KappaFunction (lee.h).
- *Abgeleitet von* AbstractFunction<float,float> *(S. 186).*

Die dielektrische Funktion

$$\kappa(\sigma) \;=\; \kappa_0 + (1-\kappa_0)\frac{1}{\exp\left(\alpha\left(\frac{\sigma}{\sigma_{vac}} - \beta\right)\right)+1}.$$

class KnuthRandom (random.h).
- *Abgeleitet von* Random *(S. 196).*

Zufallszahlengenerator nach Knuth; kann als Argument für die verschiedenen Zufallszahlen-Templates (z. B. GaussianDistribution<random>*) dienen.*

class LecuyerRandom (random.h).
- *Abgeleitet von* Random *(S. 196).*

Zufallszahlengenerator nach Lecuyer; kann als Argument für die verschiedenen Zufallszahlen-Templates (z. B. GaussianDistribution<random>*) dienen.*

class LevelRelax (mueslea/level.h).

Im wesentlichen eine Liste von BlackRelax*-Instanzen und eine Liste von* RedRelax*-Instanzen, alle von einer gemeinsamen Generation. Fasst alle Dinge zusammen, die man für eine Stufe des V-Zyklus nach Abb. 4.22 braucht.*

template LinearFunction<coeff,arg,val> (mueslea/function.h).
- *Abgeleitet von* AbstractFunction<arg,val> *(S. 186).*

Lineare Funktion der Form $y = ax$ *mit wählbaren Klassen für alle drei Größen.*

template LinearRegression<scalar> (regression.h).
- *Basisklasse für* ExponentialFit<scalar> *(S. 189).*

Berechnung linearer Fits nach der altbekannten Schulmethode.

template List<object> (mueslea/list.h).
- *Abgeleitet von* Watchdog< List<object> > *(S. 200).*
- *Basisklasse für* ClusterList *(S. 187),* DList<object> *(S. 188),* Domain *(S. 188),* HadronizedClusterList *(S. 191),* ParticleList *(S. 195),* RCList<object> *(S. 196),* SimplexIterator *(S. 198).*

Allgemeines Listentemplate. Die eigentliche Liste besteht aus doppelt verketteten Listed<object>*-Instanzen, die* List<object>*-Instanz bildet nur den Listenkopf. Bei Vernichtung der Liste werden die* Listed<object>*-Instanzen zwar mitbeseitigt, nicht aber die Objekte selbst. (vgl.* DList<object>*-Template)*

template Listed<object> (mueslea/list.h).
- *Abgeleitet von* Watchdog< Listed<object> > *(S. 200).*

Eintrag in einer Liste. Der Zeiger guy *zeigt auf das eigentliche Objekt, um das es geht. Reine Hilfsklasse; Destruktor wird von der steuernden* List<object>*- bzw* Dlist<object>*-Instanz aufgerufen*

class LocalMap (mueslea/localmap.h).
- *Abgeleitet von* Watchdog<LocalMap> *(S. 200).*
- *Basisklasse für* MetricMap *(S. 193).*

Abbildung von Vektoren auf lokale Simplexkoordinaten ("Quadrupel" von Gewichten der vier Eckknoten, die sich zu 1 *summieren, siehe Abschnitt 4.3.2.3) und umgekehrt. Stark optimiert. Caching im Simplex selbst ist vorgesehen, momentan aber nur für die gröbsten Simplices realisiert.*

template Matrix<scalar> (mueslea/linalg.h).
- *Abgeleitet von* Watchdog< Matrix<scalar> > *(S. 200).*

Einfache Matrixklasse. Zugriff auf die Elemente erfolgt über den []*-Operator.*

class MetricMap (mueslea/localmap.h).
- *Abgeleitet von* LocalMap *(S. 193).*

Eine LocalMap*-Klasse mit einer Matrix: der Metrik* (4.75)*, die bei Konstruktion ausgerechnet wird.*

class MiniSimplex (mueslea/tools.h).
- *Basisklasse für* MiniSimplexIterator *(S. 193).*

Ein Minisimplex ist ein Achtel eines normalen Simplices und hat deshalb nur vier Eckknoten. Wird momentan nur zum Zeichnen von Äquipotentialflächen in visualize.cpp *benutzt.*

class MiniSimplexIterator (mueslea/tools.h).
- *Abgeleitet von* MiniSimplex *(S. 193).*

Iterator, der indirekt über das Makro forAllMiniSimplices *benutzt wird. Wird momentan nur zum Zeichnen von Äquipotentialflächen in* visualize.cpp *eingesetzt.*

class ModelParameters (model.h).

Aus dieser Klasse gibt es genau eine Instanz im CDM*-Programm, die stets den Namen* Model *trägt. Sie kapselt alle Modellparameter inklusive der Funktionen $U(\sigma)$ und $\kappa(\sigma)$ sowie die Kopplungskonstanten, Quarkmassen und Simulationsparameter.*

class MPEGPOVDisp (disp/povdisp.h).
- *Abgeleitet von* POVDisp *(S. 196).*

Klasse zum Erzeugen von MPEG*-Movies, wobei jedes Bild erst von* POV-Ray *erzeugt wird. Sehr langsam, liefert aber qualitativ hochwertige Filme. Obsolet.*

class NamedColor (disp/disp.h).

Zuordnung von Farbnamen zu RGB-Werten.

class Node (mueslea/node.h).
- *Abgeleitet von* Flags *(S. 189)*, Number<Node> *(S. 194)*, ReferenceCounter<Node> *(S. 197)*, Watchdog<Node> *(S. 200).*

Simplices verweisen auf Node*-Instanzen, in denen Feldwerte gespeichert sind. Die Speicherverwaltung wird durch* ReferenceCounter<Node> *erledigt; diese Basisklasse zählt alle Verweise auf den Knoten mit und löscht den Knoten, wenn er nicht mehr gebraucht wird.*

class NodeFieldTableEntry (mueslea/node.h).

Tabelleneintrag, der die Zuordnung von Feldern, Feldnamen, Fences und Farben vornimmt.

class NodeIterator (mueslea/tools.h).

Iterator über alle Knoten einer Domain*-Instanz. Dieser Iterator trifft Knoten der gröbsten Generation evtl. mehrmals.*

class NodeList (mueslea/node.h).
- *Abgeleitet von* RCList<Node> *(S. 196).*

Knotenliste mit korrektem Handling der Referenzzählung an den Knoten.

class NodeRegister (mueslea/node.h).
- *Abgeleitet von* RCRegister<Node> *(S. 196).*

Knotenregister (adaptives Array) mit korrektem Handling der Referenzzählung an den Knoten.

template Number<numbered> (mueslea/counters.h).
- *Basisklasse für* Domain *(S. 188)*, Node *(S. 194)*, Particle *(S. 195)*, Simplex *(S. 198).*

Einfache Klasse, die jeder erzeugten Instanz eine neue, fortlaufende Nummer zuteilt. Dies dient hauptsächlich der Fehlersuche.

template ODESolver<moving> (mueslea/ode.h).
- *Abgeleitet von* SimpleODESolver<moving> *(S. 198).*

Differentialgleichungslöser gewöhnlicher DGLs nach einem adaptiven Runge-Kutta-ähnlichen Verfahren (nach [60] mit adaptiver Schrittweitensteuerung). Die Argumentklasse moving *enthält alle dynamischen Größen, die zu propagieren sind.*

class oscillator (mueslea/ode.h).

Beispielanwendung for das ODESolver<moving>*-Klassentemplate.*

class ParkMillerRandom (random.h).
- *Abgeleitet von* Random *(S. 196).*

Zufallszahlengenerator nach Park und Miller; kann als Argument für die verschiedenen Zufallszahlen-Templates (z. B. GaussianDistribution<random>*) dienen.*

template ParserFunction<arg,val> (parser.h).
- *Abgeleitet von* AbstractFunction<arg,val> *(S. 186).*

Funktion, die auf den Einstellungen und Parametern des Parsers beruht. Da der Parser auch das Gebiet Omega *sowie die Modellparameter enthält, kann eine solche Funktion zur Auswertung praktisch auf alle Variablen des Programms zugreifen.*

class Particle (particle.h)**.**
• *Abgeleitet von* Number<Particle> *(S. 194),* ReferenceCounter<Particle> *(S. 197).*

Repräsentiert ein Parton (Quark oder eines der sechs farbigen Gluonen) in der Simulation.

class ParticleCompound (analysis.h)**.**
• *Basisklasse für* HadronizedCluster *(S. 191).*

Speichert ein mit Partonen besetztes Schema der Art von Abb. 3.2.

class ParticleDensity (particle.h)**.**
• *Abgeleitet von* AbstractFunction<vector,float> *(S. 186).*

AbstractFunction*-Derivat, das den Quellterm der Farbladungsdichte oder anderer Dichten einer ganzen Liste von Teilchen berechnet. Leider nicht besonders schnell in der Auswertung (hier gibt es Verbesserungspotential!).*

class ParticleList (particle.h)**.**
• *Abgeleitet von* List<Particle> *(S. 192).*

Liste von Partonen. Interessant sind die Methoden zum Aufheizen und Abkühlen der Partonen, die Propagation und die Stöße zur klassischen Modellierung des Pauliprinzips.

class PDEquation (mueslea/pdequation.h)**.**
• *Abgeleitet von* Flags *(S. 189).*

Spezifikation eines Randwertproblems für den Mehrgitterlöser. Dazu gehört ein Operator, ein Quellterm ("rechte Seite"), ein Operator am Gebietsrand und ein Quellterm am Gebietsrand. Randflächen sind dabei Simplexflächen, bei denen in allen drei Knoten das Flag boundary *gesetzt ist.*

class PDESolver (mueslea/solve.h)**.**

Die zentrale Klasse des Mehrgitterlösers partieller Differentialgleichungen. Die Spezifikation der Differentialgleichung geschieht in einer PDEquation*-Instanz, das Lösen durch Aufruf der Methode* Relax.

class plane (disp/plane.h)**.**

Eine Instanz dieser Klasse repräsentiert eine Ebene, definiert durch $\{x \in \mathbb{R}^3 \mid x \cdot n = \xi\}$.

class PolynomialFunction (mueslea/function.h)**.**
• *Abgeleitet von* AbstractFunction<float,float> *(S. 186),* Vector<float> *(S. 199).*
• *Basisklasse für* FriedbergLeePotential *(S. 190).*

Ein reelles Polynom. Die Koeffizienten sind als Vektor gespeichert. Neben dem Funktionswert lassen sich auch die Werte der ersten beiden Ableitungen abfragen.

class POVDisp (disp/povdisp.h).
- *Abgeleitet von* Disp *(S. 188).*
- *Basisklasse für* MPEGPOVDisp *(S. 193).*

Disp-*Derivat zur Erzeugung von* POV-Ray-*Szenenfiles.*

class ProjectEntry (projects.h).

Eintrag in der Projekttabelle. Projekte werden mittels dieser Tabelle vom CommandStreamParser *aufgerufen.*

template quadruple<object> (mueslea/localmap.h).

Quadrupel (Simplexkoordinaten) sind in Abschnitt 4.3.2.3 erläutert. Diese Klasse wirkt im Zusammenhang mit LocalMap.

class Random (random.h).
- *Basisklasse für* KnuthRandom *(S. 192),* LecuyerRandom *(S. 192),* ParkMillerRandom *(S. 194).*

Abstrakte Basisklasse für Zufallszahlengeneratoren.

template RandomDirection<random> (random.h).
- *Abgeleitet von* RandomVector<random> *(S. 196).*

Zufallszahlentemplate zur Erzeugung von Einheitsvektoren mit zufälliger Richtung (korrektes Raumwinkelmaß). Die Basisklasse random *hat die Aufgabe, gleichmäßig verteilte Zufallszahlen zu erzeugen.*

template RandomVector<random> (random.h).
- *Basisklasse für* RandomDirection<random> *(S. 196).*

Zufallszahlentemplate zur Erzeugung von zufälligen Vektoren in einem dreidimensionalen Intervall. Die Basisklasse random *hat die Aufgabe, gleichmäßig verteilte Zufallszahlen zu erzeugen.*

template RCList<object> (mueslea/list.h).
- *Abgeleitet von* List<object> *(S. 192).*
- *Basisklasse für* NodeList *(S. 194).*

Liste mit korrektem Handling der Referenzzählung. Die Argumentklasse muss vom ReferenceCounter-*Template abgeleitet sein.*

template RCRegister<object> (mueslea/register.h).
- *Abgeleitet von* Register<object> *(S. 197).*
- *Basisklasse für* NodeRegister *(S. 194).*

Register *(Array) mit korrektem Handling der Referenzzählung. Die Argumentklasse muss vom* ReferenceCounter-*Template abgeleitet sein.*

class RedRelax (mueslea/red.h).
- *Abgeleitet von* Relax *(S. 197),* Watchdog<RedRelax> *(S. 200).*

Eine RedRelax-*Instanz speichert und relaxiert (löst) die lineare Gleichung, die zu den roten Knoten einer Verfeinerungszelle gehört (S. 111).*

class RedRelaxPool (mueslea/pool.h).

Ein "Pool" von RedRelax*-Instanzen, die momentan im Mehrgitterverfahren nicht gebraucht werden. Dient einerseits als Cache, damit nicht ständig alle Matrizen neu berechnet werden müssen. Andererseits werden hier die* RedRelax*es abgelegt, die eine Verfeinerungssstufe unterhalb den feinsten* RedRelax*es der Mehrgitterstruktur liegen, und die damit in Frage kommen, bei Verfeinerungsschritten in den Mehrgitteralgorithmus involviert zu werden.*

class RedSparseMatrix (mueslea/red.h).
- *Abgeleitet von* Watchdog<RedSparseMatrix> *(S. 200).*

Kapselt alle Matrizen, die zu einer RedRelax*-Instanz gehören, inklusive des LR-zerlegten Relaxblocks.*

template ReferenceCounter<watched> (mueslea/counters.h).
- *Basisklasse für* Node *(S. 194),* Particle *(S. 195).*

Abstraktes Basisklassentemplate dient zum Mitzählen von Referenzen und Aufruf des Destruktors, falls das Objekt nicht mehr referenziert wird. Wird in CDM *für* Node*-Instanzen benutzt, die vernichtet werden müssen, wenn kein* Simplex *sie mehr referenziert.*

class region (mueslea/region.h).
- *Basisklasse für* Sandbox *(S. 197).*

Quaderförmiger Bereich im $\mathbb{R}^3$.

template Register<object> (mueslea/register.h).
- *Basisklasse für* DRegister<object> *(S. 189),* RCRegister<object> *(S. 196).*

Array variabler Größe.

class Relax (mueslea/relax.h).
- *Abgeleitet von* Flags *(S. 189).*
- *Basisklasse für* BlackRelax *(S. 186),* RedRelax *(S. 196).*

Abstrakte Basisklasse für RedRelax *und* BlackRelax.

class safestring (disp/safestr.h).

Primitive Stringklasse.

class Sandbox (sandbox.h).
- *Abgeleitet von* region *(S. 197).*

Dient zur Berechnung von Teilchendichtestatistiken.

class sigmaSourceFunction (singquark.cpp).
- *Abgeleitet von* SourceFunction *(S. 198).*

Dient als Quellfunktion für die Lösung der sphärisch symmetrischen σ*-Feldgleichung mit der* SimpleBVPSolver*-Klasse (S. 197). Obsolet.*

class SimpleBVPSolver (bvp.h).
- *Abgeleitet von* InterpolatingTable<double> *(S. 192).*

Klasse zur Lösung eindimensionaler Randwertprobleme mittels Überrelaxation. Die Lösung lässt sich dann über die InterpolatingTable<double> *(Basisklasse) abfragen und auswerten.*

template SimpleFunction<arg,val> (mueslea/function.h).
- *Abgeleitet von* AbstractFunction<arg,val> *(S. 186).*

Einfache Funktion, deren Auswertung auf den Aufruf einer beliebigen C++*-Funktion hinausläuft.*

template SimpleODESolver<moving> (mueslea/ode.h).
- *Basisklasse für* ODESolver<moving> *(S. 194).*

Einfache Klasse zur Lösung eindimensionaler Anfangswertprobleme mit verschiedenen Algorithmen ohne adaptive Schrittweitensteuerung. Die Argumentklasse moving *enthält alle dynamischen Größen, die zu propagieren sind.*

class Simplex (mueslea/simplex.h).
- *Abgeleitet von* Flags *(S. 189),* Number<Simplex> *(S. 194),* Watchdog<Simplex> *(S. 200).*

Ein Simplex mit Zeigern auf seine zehn Knoten, vier Nachbarsimplices, zwei Kindsimplices und einen Elternsimplex.

class SimplexIterator (mueslea/tools.h).
- *Abgeleitet von* List<Simplex> *(S. 192).*

Iterator über alle Simplices einer List<Simplex>*-Instanz. Wird über Makros wie* forAllSimplices *(*tools.h*) benutzt. Ermöglicht die iterative Bearbeitung der Simplices eines Waldes und macht damit rekursive Routinen zur Bearbeitung des Simplexwaldes weitgehend unnötig.*

class SimplexOperator (mueslea/operator.h).
- *Abgeleitet von* BasicOperator *(S. 186).*

Einfacher Operator (Funktion Simplex $\rightarrow 10 \times 10$*-Matrix), dessen Berechnung einfach auf den Aufruf einer gegebenen* C++*-Funktion hinausläuft.*

class SimulationRun (simulation.h).

Klasse, die die Zeitschrittschleife (Methode TimeLoop*) der dynamischen Simulation des chromodielektrischen Modells kapselt. Wird meist von Projektroutinen nach der Initialisierung eines Anfangsszenarios aufgerufen. Die Klasse ist darauf vorbereitet, angefangene Simulationsläufe fortzusetzen. Viele Auswertefunktionen.*

class Site (site.h).
- *Abgeleitet von* Flags *(S. 189).*

Was die Node *für den* Domain*, ist die* Site *für das* Grid*: elementare Klasse, die einen Knoten (hier: Gitterposition) repräsentiert und die Feldwerte speichert.*

class SiteFieldTableEntry (site.h).

Tabelleneintrag, der die Zuordnung von Feldern, Feldnamen, Fences und Farben vornimmt.

class SourceFunction (bvp.h).
- *Basisklasse für* GaussianSourceFunction *(S. 190),* sigmaSourceFunction *(S. 197).*

Abstrakte Basisklasse für die Quellfunktion eines eindimensionalen Randwertproblems. Wirkt in Zusammenarbeit mit SimpleBVPSolver.

template SpatialDistribution<random> (random.h).
- *Basisklasse für* UniformRegionDistribution<random> *(S. 199)*, UniformSphericalDistribution<random> *(S. 199)*, UniformTubularDistribution<random> *(S. 199)*.

Abstrakte Basisklasse für vektorwertige Zufallszahlengeneratoren.

template SquareMatrix<scalar> (mueslea/linalg.h).
- *Abgeleitet von* Watchdog< SquareMatrix<scalar> > *(S. 200)*.

Klasse zur Darstellung quadratischer Matrizen. Es existieren u. a. Routinen zur LR-Zerlegung und zum Lösen linearer Gleichungssysteme mit diesen Matrizen.

class StringFunction (cube.h).
- *Abgeleitet von* GaussianFunction *(S. 190)*.

Eine Funktion $\mathbb{R}^3 \mapsto \mathbb{R}$, deren Wert gleich dem einer gaußschen Glockenkurve ist. Als Argument dient aber nicht die Entfernung von einem punktförmigen Zentrum der Glockenkurve, sondern die Entfernung von einem endlichen geraden Strich (String). Dient zur Initialisierung des σ-Feldes eines Flussschlauchs.

class transformation (disp/vector.h).

Dreidimensionale Transformation (Translation und Rotation).

template UniformInteger<random> (random.h).

Zufallszahlengenerator mit ganzzahligem Wertebereich.

template UniformRegionDistribution<random> (random.h).
- *Abgeleitet von* SpatialDistribution<random> *(S. 199)*.

*Homogene Zufallsverteilung in einem Quader (*region*).*

template UniformSphericalDistribution<random> (random.h).
- *Abgeleitet von* SpatialDistribution<random> *(S. 199)*.

Homogene Zufallsverteilung in einer Kugel.

template UniformTubularDistribution<random> (random.h).
- *Abgeleitet von* SpatialDistribution<random> *(S. 199)*.

Homogene Zufallsverteilung in einem Zylinder.

template Vector<scalar> (mueslea/linalg.h).
- *Abgeleitet von* Watchdog< Vector<scalar> > *(S. 200)*.
- *Basisklasse für* PolynomialFunction *(S. 195)*.

Vektorklassentemplate, zur Verwendung mit Klassen wie Matrix<scalar> *oder* SquareMatrix<scalar>. *Es existieren Funktionen z. B. zur Matrix-Vektor-Multiplikation.*

class vector (disp/vector.h).

Vielseitige dreidimensionale Vektorklasse.

template Watchdog<watched> (mueslea/counters.h).

- *Basisklasse für* BlackRelax *(S. 186)*, Domain *(S. 188)*, List<object> *(S. 192)*, Listed<object> *(S. 193)*, LocalMap *(S. 193)*, Matrix<scalar> *(S. 193)*, Node *(S. 194)*, RedRelax *(S. 196)*, RedSparseMatrix *(S. 197)*, Simplex *(S. 198)*, SquareMatrix<scalar> *(S. 199)*, Vector<scalar> *(S. 199)*.

Dieses Klassentemplate dient einfach dazu, die Erzeugung und Vernichtung von watched-*Instanzen zu überwachen, sprich mitzuzählen, und entsprechende Meldungen auszudrucken, wenn es kein Objekt dieser Klasse mehr gibt. Für Debuggingzwecke (Aufspüren von Speicherlecks).*

template WritableAbstractFunction<arg,val> (mueslea/function.h).

- *Abgeleitet von* AbstractFunction<arg,val> *(S. 186)*.
- *Basisklasse für* DomainFunction *(S. 188)*, GridFunction *(S. 191)*.

Funktion, die zum Beispiel auf einem Domain *oder* Grid *definiert ist und damit auch beschreibbar wird.*

class XDisp (disp/xdisp.h).

- *Abgeleitet von* Disp *(S. 188)*.
- *Basisklasse für* GLXDisp *(S. 190)*.

Disp-*Derivat zur grafischen Ausgabe auf einem* X11-*Display. Speicherfunktionen in verschiedenen Grafikformaten.*

class XmgrPlot (xmgr.h).

- *Abgeleitet von* Flags *(S. 189)*, ofstream.

Klasse zur Erzeugung von Datenfiles. Im wesentlichen öffnet diese Klasse einen ofstream *(Outputstream), in den danach die Daten geschrieben werden. Am Kopf dieses Files werden Kommandos geschrieben, so dass das Plotprogramm* xmgr *(auch bekannt als* ACE/gr*) u. a. eine benannte Legende erzeugt. Mit dem einfachen Kommando* xmgr -source stdin -nxy <out.dat *kann so ein Plot in Publikationsqualität erzeugt werden.*

“Gegenwärtig hat er’s mit der Chemie. Er liest riesig schwierige Bücher über die Elektronentheorie und über die kinetische Gastheorie und über die Quantentheorie und solche Sachen. Er will Gelehrter werden und herauskriegen, was in den Atomen drin ist.”
E. KÄSTNER *[1]*

Literaturverzeichnis

[1] E. Kästner, “Das fliegende Klassenzimmer”, Stuttgart 1933.

[2] H. Satz, “RHIC and LHC: physics perspectives”, in Proc. of the Conference *Quark Matter ’91*, Gatlinburg, TN, USA, Nov 11-15, 1991, *Nucl. Phys.* **A** 544 (1992) 371.

[3] J. I. Friedman, H. W. Kendall, “Deep inelastic electron scattering”, *Ann. Rev. Nucl. Science* 22 (1972) 203.

[4] Particle Data Group, “Particle Physics Booklet”, Auszug aus C. Caso et al., “Review of Particle Physics”, *Europ. Phys. J.* **C** 3 (1998) 1

[5] F. Halzen und A. D. Martin, “Quarks and Leptons: An Introductory Course in Modern Particle Physics”, Wiley, New York 1984

[6] M. Creutz, “Quarks, Gluons, and Lattices”, Cambridge 1983

[7] B. Petersson, “Lattice Results on Finite-Temperature QCD: A Status Report”, *Nucl. Phys.* **A** 525 (1991) 237c.

[8] C. DeTar, in “Quark-Gluon Plasma 2”, ed. R. C. Hwa, World Scientific, Singapore 1995

[9] H. Satz, “RHIC and LHC: Physics Perspectives”, *Nucl. Phys.* **A** 544 (1992) 371c

[10] E. Braaten, R. D. Pisarski, *Phys. Rev.* **D** 42 (1990) 2156.

ein neuerer Übersichtsartikel ist: M. H. Thoma, in “Quark-Gluon Plasma 2”, ed. R. C. Hwa, World Scientific, Singapore, 1995, S. 51.

[11] L. V. Keldysh, Sov. Phys. JETP 20 (1965) 1018.

[12] *Proc. of the Workshop on QCD at Finite Baryon Density*, Bielefeld, April 27-30, 1998.

W. Bietholz, U.-J. Wiese, "Perfect Actions with Chemical Potential", hep-lat/9801022.

S. D. Hsu, M. Schwetz, "On the QCD Phase Transition at Finite Baryon density", hep-ph/9803386.

R. Aloisio, V. Azcoiti, G. DiCarlo, A. Galante, A. F. Grillo, "Finite Density QCD in the Chiral Limit", hep-lat/9709118, hep-lat/9804020.

I. M. Barbour, S. E. Morrison, E. G. Klapfish, J. B. Kogut, M.-P. Lombardo, "Results on Finite Density QCD", hep-lat/9705042.

W. Bietenholz, "Improved Lattice Action with Chemical Potential", in Proc. of the Workshop on *QCD at Finite Baryon Density*, Bielefeld, April 27-30, 1998, hep-lat/9805014.

[13] B. Andersson, G. Gustafsson, G. Ingelman, T. Sjöstrad, *Phys. Rep.* 97 (1983) 33.

B. Andersson, G. Gustafsson, B. Nielssen-Almquist, *Nucl. Phys.* **B** 281 (1987) 289.

[14] H. Fukuda, C. Iso, *Prog. Theor. Phys.* 60 (1978) 1439.

S. A. Azimov, G. M. Chernov, K. G. Gularnov, V. Sh. Navotnyi, N. S. Scripnik, *Phys. Lett.* **B** 73 (1978) 500.

D. H. Boal, *Phys. Rev.* **C** 33 (1986) 2206.

[15] R. J. Glauber, "Theory of high-energy hadron-nucleus collisions", in Proc. of Methods in Subnuclear Physics, Vol. V., part 1, pp 1-52, Herceg-novi 1969, New York 1977.

R. J. Glauber, "High-energy collision theory", in Geometrical pictures in hadronic collisions, S. 83, ed. S. Y. Lo, 1987.

[16] R. P. Feynman, R. D. Field, *Phys. Rev.* **D**15 (1977) 2590, *Nucl. Phys.* **B** 136 (1978) 1.

[17] K. Geiger, B. Müller, *Nucl. Phys.* **B** 369 (1992) 600.

K. Geiger, *Phys. Rep.* 258 (1995) 238.

J. Ellis, K. Geiger, *Phys. Rev.* **D** 54 (1996) 1755.

[18] K. Werner, *Z. Phys.* **C** 42 (1989) 85, *Phys. Rep.* 232, 87, 1993.

[19] H. Sorge, H. Stöcker, W. Greiner, *Ann. Phys.* 192 (1989) 266.

S. A. Bass et. al., "Equation of state, spectra and composition of hot and dense hadronic matter in a microscopic transport model", nucl-th/9804058.

[20] T. S. Biró, P. Lévai, J. Zimányi, *Phys. Lett.* **B** 347, 6, 1995, J. Phys. G 23 (1997) 1941.

[21] H. W. Barz, B. L. Friman, J. Knoll, H. Schulz, *Nucl. Phys.* **A** 484, 661, 1988.

[22] T. S. Biro, J. Zimányi, "Hadronization Transchemistry", hep-ph/9807303.

[23] P. Rehberg, S. P. Klevansky, J. Hüfner, *Phys. Rev.* **C** 53 (1996) 410.

P. Rehberg, S. P. Klevansky, J. Hüfner, *Nucl. Phys.* **A** 608 (1996) 356.

S. P. Klevansky, P. Rehberg, A. Ogura, J. Hüfner, "Developing transport theory to systematically include mesons and hadronization", in Proc. of Workshop on QCD Phase Transitions, Hirschegg 1997, p 397, Ed. H. Feldmeier, J. Knoll, hep-ph/9701355.

[24] S. P. Klevansky, *Rev. Mod. Phys.* 64 (1992) 649.

[25] R. Friedberg, T. D. Lee, *Phys. Rev.* **D** 15 (1977) 1694,

[26] R. Friedberg, T. D. Lee, *Phys. Rev.* **D** 16 (1977) 1096.

[27] R. Friedberg, T. D. Lee, *Phys. Rev.* **D** 18 (1978) 2623.

[28] R. Goldflam, L. Wilets, *Phys. Rev.* **D** 25 (1982) 1951.

G. Fái, R. J. Perry, L. Wilets, *Phys. Lett.* **B** 208 (1988) 1.

R. Horm, R. Goldflam, L. Wilets, *Comp. Phys. Comm.* 42 (1986) 105

L. Wilets, M. C. Birse, G. Lübeck, E. M. Henley, *Nucl. Phys.* **A** 434 (1985) 129

G. Lübeck, M. C. Birse, E. M. Henley, L. Wilets, *Phys. Rev.* **D** 33 (1986) 234

G. Lübeck, E. M. Henley, L. Wilets, *Phys. Rev.* **D** 35 (1987) 2809

M. Bickeböller, M. C. Birse, H. Marschall, L. Wilets, *Phys. Rev.* **D** 31 (1985) 2892

P. Tang, L. Wilets, *J. Math. Phys.* 31 (1990) 1661

M. Bickeböller, M.C. Birse, L. Wilets, *Z. Phys.* **A** 326 (1988) 89

[29] S. Kahana, G. Ripka, *Nucl. Phys.* **A** 429 (1984) 462.

[30] U. Mosel, “Fields, Symmetries and Quarks”, McGraw-Hill, London 1989.

[31] U. Kalmbach, T. Vetter, T. S. Biró, U. Mosel, *Nucl. Phys.* **A** 563 (1993) 584.

[32] T. Vetter, T. S. Biró, U. Mosel, *Nucl. Phys.* **A** 581 (1995) 598.

[33] T. Vetter, Doktorarbeit, Giessen 1994.

[34] S. Loh, T. S. Biró, U. Mosel, M. H. Thoma, *Phys. Lett.* **B** 387 (1996) 685.

[35] S. Loh, Doktorarbeit, Giessen 1997.

[36] C. T. Traxler, T. S. Biro, U. Mosel, “Hadronization of a Quark-Gluon-Plasma in the Chromodielectric Model”, im Druck bei *Phys. Rev.* **C**(erscheint voraussichtlich im März 1999).

Vorträge über diese Arbeit sind vom Autor in Budapest, Heidelberg und Nantes und von T. S. Biro in Matrahaza gehalten worden.

[37] L. Wilets, “Nontopological Solitons”, World Scientific, Singapore 1989

[38] M. Gell-Mann, “The Quark and the Jaguar”, Freeman, New York 1994

[39] S. C. Frautschi, “Regge Poles and S-Matrix Theory”, Benjamin, New York 1963

[40] S. C. Frautschi, “Statistical Bootstrap Model Of Hadrons”, *Phys. Rev.* **D** 3 (1971) 2821

[41] D. H. Perkins, “Introduction to High Energy Physics”, Addison-Wesley 1987.

[42] P. Pennanen, A. M. Green und C. Michael, “Four-quark flux distribution and binding in lattice QCD”, *Phys. Rev.* **D** 59 (1999) 4504, hep-lat/9804003

[43] G.S. Bali, K. Schilling und C. Schlichter, “Observing Long Color Flux Tubes in $SU(2)$ Lattice Gauge Theory”, *Phys. Rev.* **D** 51 (1995) 5165

[44] J. E. Mandula, *Phys. Rev.* **D** 14 (1976) 3497

[45] M. Magg, *Phys. Lett.* **B** 74 (1978) 246

[46] K. Cahill, *Phys. Rev. Lett.* 41 (1978) 599

[47] H. Arodz, *Phys. Lett.* **B** 78 (1978) 129

[48] R. A. Freedman. L. Wilets, S. D. Ellis, E. M. Henley, *Phys. Rev.* **D** 22 (1980) 3128

[49] S. Mandelstam, *Phys. Rev.* **D** 19 (1979) 2391.

[50] M. Baker, J. S. Ball, F. Zachariasen, *Phys. Rev.* **D** 34 (1986) 3894.

[51] H. B. Nielsen, A. Patkos, *Nucl. Phys.* **B** 195 (1982) 137.

[52] G. Chanfray, O. Nachtmann, H. J. Pirner, *Phys. Lett.* **B** 147 (1984) 249.

O. Nachtmann, H. J. Pirner, *Z. Phys.* **C** 21 (1984) 277.

[53] H. J. Pirner, "The Color Dielectric Model of QCD"", Prog. Part. Nucl. Phys. 29 (1992) 33.

[54] E. Naar, M. C. Birse, *J. Phys.* **G** 19 (1993) 555.

[55] J. A. McGovern, M. C. Birse, D. Spanos, *J. Phys.* **G** 16 (1990) 1561.

[56] L. M. Krauss, "The Physics of Star Trek", BasicBooks, New York 1995

[57] S. K. Wong, "Field and Particle Equations for the Classical Yang-Mills Field and Particles with Isotopic Spin", *Nuovo Cim.* **A**65 (1970) 689

[58] C. T. Traxler, "An Algorithm for Adaptive Mesh Refinement in n Dimensions", *Computing* 59 (1997) 115.

[59] L. Wilets, R. D. Puff, "Collapse of Flux Tubes", *Phys. Rev.* **C** 51 (1995) 339.

[60] W. H. Press, S. A. Teukolsky, W. T. Vetterling und B. P. Flannery, "Numerical Recipes in C", 2nd Edition, Cambridge 1992.

[61] R. Balian, "From Microphysics to Macrophysics", Vol.I und II, Springer 1991

[62] R. E. Bank, T. F. Dupont, and H. Yserentant, "The Hierarchical Multigrid Method", *Num. Math.* 52 (1988) 427.

[63] J. H. Bramble, J. E. Pasciak, and J. Xu, "Parallel Multilevel Preconditioners", *Math. Comp.* Vol. 55 No. 191 (1990) 1.

[64] W. Mitchell, "Optimal Multilevel Iterative Methods for Adaptive Grids", *SIAM J. Sci. Stat. Comp.* 13 (1992) 146.

[65] W. F. Ames, "Numerical Methods for Partial Diff. Equations", 3rd edition, Academic Press 1992

[66] H. Freudenthal, "Simplizialzerlegungen von beschränkter Flachheit", *Ann. Math.* 43 (1942) 580.

R. E. Bank, A. H. Sherman, H. Weiser, "Refinement Algorithms and Data Structures for Regular Local Mesh Refinement", *Scientific Computing*, R. Stepleman et al. (eds.), Amsterdam, IMACS North-Holland 1983

M. C. Rivara, "Design and Data Structures of fully adaptive, multigrid, finite-element software", *ACM Trans. Math. Soft.* 10 (1984) 242.

E. Bänsch, "Local Mesh Refinement in 2 and 3 Dimensions", *IMPACT Comput. Scr. Eng.* 3 (1991) 181.

D. Moore, "Subdividing Simplices", Kapitel des Buchs "Graphics Gems III", hrsg. von Kirk, Academic Press 1992

J. Bey, "Tetrahedral Grid Refinement", *Computing* 55 (1995) 355.

J. M. Maubach, "Local Bisection Refinement for N-Simplicial Grids Generated by Reflections", *SIAM J. Sci. Comput.* 16 (1995) 210.

[67] H. Yserentant, "On the Multi-Level Splitting of Finite Element Spaces", *Num. Math.* 49 (1986) 379.

[68] F. Bornemann, B. Erdmann, and R. Kornhuber, "Adaptive Multilevel Methods in Three Space Dimensions", *Int. J. Num. Meth. Eng.* 36 (1993) 3187

[69] H. Kragh, "Dirac: A Scientific Biography", Cambridge 1990

[70] H.Satz, "Deconfinement and Percolation", hep-ph/9805418.

[71] W. Koepf, L. Wilets, S. Pepin, F. Stancu, *Phys. Rev.* **C** 50 (1994) 614.

W. Koepf, L. Wilets, *Phys. Rev.* **C** 51 (1995) 3445.

[72] C. Jung, W. Cassing, U. Mosel, *Nucl. Phys.* **A** 477 (1988) 256.

[73] R. Hagedorn, "Statistical Thermodynamics Of Strong Interactions At High Energies", *Nuovo Cim. Suppl.* 3 (1965) 147.

M. I. Gorenstein, V. A. Miranskii, V. P. Shelest, G. M. Zinovev, H. Satz, "The Physical Content Of The Statistical Bootstrap", *Nucl. Phys.* **B** 76 (1974) 453.

G. Veneziano, *Nuovo Cim.* **A** 57, (1968) 190.

[74] D. Stauffer, *Phys. Rep.* 54 (1978) 1.

D. Stauffer and A. Aharony, “Introduction to Percolation Theory”, London 1992

[75] N. Xu (NA44 Collaboration), “Hadron distributions - recent results from the CERN experiment NA44”, *Nucl. Phys.* **A** 610 (1996) 175c.

[76] H. van Hecke, H. Sorge, N. Xu, “Evidence of early multi-starnge hadron freeze out in high energy nuclear collisions”, nucl-th/9804035.

[77] S. Meyers, “Effective C++”, 2nd Edition, Addison-Wesley 1998

[78] E. G. Nikonov, A. A. Shanenko, V. D. Toneev, “A mixed phase model and the ’softest point’ effect”, nucl-th/9802018.

[79] T. Elze, U. Heinz, *Phys. Rep.* 183 (1989) 81.

[80] M. Belkacem et al., “Equation of State, Spectra and Composition of Hot and Dense Infinte Hadronic Matter in a Microscopic Transport Model”, nucl-th/9804058.

[81] Ape Collaboration (P. Bacilieri et al.), “Scaling In Lattice QCD, Glueball Masses And String Tension”, *Phys. Lett.* **B** 205 (1988) 535

[82] N. A. Campbell, C. Michael, P. E. L. Rakow, “The String Tension From Lattice QCD”, *Phys. Lett.* **B** 139 (1984) 288

[83] UKQCD Collaboration (C. R. Allton et al.), “The Light Hadron Spectrum And Decay Constants In Quenched Lattice QCD”, *Phys. Rev.* **D** 49 (1994) 474

[84] G. J. H. Burgers, C. Fuglesang, R. Hagedorn, V. Kuvshinov, “Multiplicity Distributions In Hadron Interactions Derived From The Statistical Bootstrap Model”, *Z. Phys.* **C** 46 (1990) 465

[85] G. J. H. Burgers, R. Hagedorn, V. Kuvshinov, “Multiplicity Distributions In High-energy Collisions Derived From The Statistical Bootstrap Model”, *Phys. Lett.* **B** 195 (1987) 507

[86] R. Fiore, R. Hagedorn, F. d’Isep, “Statistical Bootstrap Model And Phase Transition From Hadron Matter To Quark Gluon Plasma”, *Nuovo Cim.* **A** 88 (1985) 301

[87] S. Kogitz, R. K. Logan, S. Tanaka, “Statistical Bootstrap Duality”, *Phys. Rev.* **D** 14 (1976) 1420

[88] R. K. Logan, M. Bates, S. Kogitz, “Statistical Bootstrap Duality II”, *Phys. Rev.* **D** 25 (1982) 2441

Index

abelschen, 13, 24, 35, 42–48, 65, 149
Ableitung, kovariante, 26
Ableitungsoperator, kovariante, 24, 26
Abschätzung, obere, 140, 142
Abschlussschnitte, 86, 87
abzuschirmen, 86
adaptiv, 2, 14, 63, 74, 80, 83–87, 91, 93, 94, 97, 100, 128, 129, 131, 132, 165, 194, 198, 205, 206
AddFineContribution, 117, 120
aitken0, 124
Aitkentransformation, 124, 125
ALCOR-Modell, 11, 12
ANSI, 180
anti-screening, 32
apmd, 211
ar, 210
aufgeweicht, 139
Aufstellen der Matrixgleichungen, 120
ausgeschmiert, 132
Ausschließungsprinzip, 22, 72
Ausschnittes, 162

Baryonen, 7, 12, 17, 21, 22, 53, 136
Basis, hierarchischen, 109
Baumes, binären, 94
BD, 119
Billardkugel, relativistische, 73
billiger, 9
BjorkenQGPTubeHadronization, 181
Bjorkenszenario, 152, 158, 160, 161
BlackRelax, 118, 121
Blockschaltbild, 177
book-keeping, 80
Bootstrapmodell, statistischen, 159
BoundaryOperator, 122
Brainstorming, 128, 169

Caching, 124, 193
CDM, 2, 3, 10, 14, 15, 63, 65, 69, 71, 80, 83, 84, 107, 117, 119, 122, 132, 133, 139, 146, 153, 157, 166, 168, 177, 180, 182–186, 210
CERN, 5, 6, 9, 24, 148, 162, 207
children simplices, 88
chore, 180, 181
cjpeg, 183
closure cuts, 86
Cluster, 133
Cluster, irreduziblen weißen, 52, 60, 67, 166, 167, 171
Cluster, irreduziblen weißen, 53
Clusterisierung, 133
coarse grid relaxation, 112
coarse, 110, 117
COARSE_EXACT, 113
coarseblock, 121
COARSEST, 117
CommandStreamParser, 181, 182
ComputeCoarseGridCorrection, 120

Computersimulation des Modells, 1, 14, 63
configure, 184
Confinement, 1, 6, 7, 26, 27, 29–32, 34, 167
Confinement-Phasenübergangs, 9
Copyleft, 210
corner nodes, 77
Cube, 77, 96, 182
cut edge, 88
cut, 92
CVS, 183

Darstellung, adjungierte, 39
Darstellung, adjungierten, 25, 26
Darstellung, fundamentale, 24, 26, 42
Das chromodielektrische Modell, 1, 14, 37, 49, 50, 135, 165
DDD, 182
Deconfinement, 7, 206
Deconfinement-Phasenübergang, 30, 138
Defektblock, 111, 113, 121
Dirichletsche Randbedingungen, 103, 110, 122, 131
disp, 3, 63, 177, 180, 182, 185
Distiller_AGS, 181
Distiller_SPS, 181
Domain, 77, 94, 97, 100, 182
Drehimpulsalgebra, quantenmechanischen, 18

Eichgruppe, 1, 6, 24–26, 29, 37, 41, 57
Elternsimplex, 88, 94
Empfehlung, 189
Endzustandswechselwirkung, 133, 157
EPS, 182
erklärt, 7, 11, 14, 21, 23, 27, 68, 91, 97, 100, 136, 146, 159, 166
event-by-event, 13
Events, 9, 149, 158, 161, 162
exakte, 10, 99, 100, 103, 111

Faden, roten, 97
Farb-Maxwell-Feldern, 49
Farb-Poyntingvektor, 49, 150
Farbconfinement, 1, 7, 11, 12, 14, 29, 31, 34, 37, 48, 52, 62, 66, 131, 136, 166
farbigen, 47, 68, 73, 135, 151, 167, 195
Farbladung, einzelnen, 139
Fehler, relativen, 116, 127
Fehler, relativer, 124
Feingitterblock, 111, 121
Feingitterknoten, 111
Fences, 94–96, 194, 198
festgehaltener, 41
Feynman, 11, 23, 27, 57, 202
Feynman-Dagger, 23
field imprint, 133
final-state interaction, 157
Finite Elemente, 2, 74–76, 82, 83, 85, 97, 101, 106, 107, 186
Flavors, 21, 143, 149, 152
Form, schwache, 74, 116
FORTRAN, 106
fortune, 171
Free Software Foundation, 182, 210
freeze-out point, 157
Freiheitsgrade, relevanten, 7
from scratch, 169
full stopping, 160, 166
Full-Stopping-Szenario, 155, 160

garbage collection, 96
gcc, 210
gdb, 182, 210
Gebietspartition, reflektierte, 89, 90
geladenen, 6, 13, 45
Gell-Mann, 6, 17, 24, 26, 46, 58, 59, 204
gepipet, 180
Ghostscript, 211
GIMP, 211
Gitter, 28, 31, 70, 71, 74, 83, 97, 99, 100, 103, 107, 111–113, 115,

120, 122, 128, 129, 131–133, 191
Gitter, grobe, 2, 98–101, 103, 106–108, 111–114
Gleichgewichtshadronisierung, 11
Gluebälle, 7, 52, 56, 59, 137
Gluonen, 7, 8, 11–13, 24, 27, 30, 34, 42, 45–48, 52, 56, 57, 59, 64, 73, 132, 133, 135, 137, 143, 145, 146, 148, 149, 151, 154, 173, 195
gmake, 183
GNU Emacs, 210
GNU, 182–184, 210
go, 180, 181
Grid, 70, 77, 97, 100, 182
groben, 86, 96, 100, 101, 107–113, 117, 186
Grobgitterblock, 111, 112
Grobgitterknoten, 101, 111, 112
Grobgitterkorrektur, 113, 121
Grobgitterrelaxation, 112–114, 120
großer, 14, 140, 157, 169
großer, 79, 81, 129
GUI, 3, 177

Hadron, 8, 11, 12, 24, 54, 57, 133, 135, 136, 153, 154, 157, 158, 167, 179, 202, 204, 207
Hadronbag, 23
Hadronen bestehend, aus, 159
Hadronen kondensiert, zu, 165
Hadronen, 1, 6, 7, 9, 12–14, 17, 18, 21–23, 27, 32, 52, 53, 56, 61, 62, 131–133, 135, 151–154, 157, 159, 166, 179, 191
Hadronenspektrum, 1, 37, 52, 53, 135, 136, 159
Hadronisierung, evolutionäre, 11
Hadronisierung, instantane, 11
Hadronisierungsquerschnitte, 12
hadronization cross sections, 12
HadronizedCluster, 133
HadronizedClusterList, 182
Hagedornsche hadronische Zustandsdichte, 159
Hagedorntemperatur, 159
Herleitung, 11, 23, 33, 35, 43
heruntergedrückt, 150
hierarchical multigrid algorithm, 97

Impulsverteilung, transversale, 151, 160
InMediumMinusLaplaceFunc, 117
InMediumMinusLaplaceFunction, 120
institutshistorische, 84
interpolierten, 82
irreduzibel, 52, 53

Jacobi-Iterationen, 99

kalten, 61, 136, 154
Kern, harten, 209
Kernkraft, 6
Kinder, 88
Kindsimplices, 88, 92, 94, 96, 107, 198
Knoten, 75–82, 84, 86–90, 92, 93, 95–97, 99, 100, 102–107, 109, 111–114, 117–122, 124, 128, 129, 186, 188, 194, 195, 198
Knoten, rote, 107, 108, 111, 114, 196
Knoten, schwarze, 107, 113, 114
Knotenindex, 77, 78, 87, 93
kompatibel, 86
kompatiblen, 80
Konstituentenquarkmodell, 22, 136
Kontinuitätsgleichung, eichinvariante, 39, 43
Koordinaten, lokalen, 77
Kopplung, minimale, 24
Kuhnsimplex, 78
Kuhnsimplices, 77, 94

Lamb shift, 23
langweilig, 85
last but not least, 210

leapfrog, 69, 128
Leapfrogt, 69
Lebensdauern, 23, 158
LevelRelax, 118
Liegruppe, 24
linear, 2, 20, 27–29, 74, 75, 77–80, 82, 85, 97, 99, 101, 103–107, 119, 120, 122, 146, 148, 161, 189, 192, 196, 199
linearisierten, 105
Linux, 182
LocalMap, 77
lokal trivial, 186
lokaler, 34, 78, 86, 114, 131
LUDecomposition, 120
LUrelaxblock, 121

machen, 10, 13, 55, 56, 83, 85, 97, 122, 132, 162, 174
makechore, 180
makefile, 184
mal schnell, 179
Masse, effektiven, 11, 70, 149
Massenverteilung, 3, 64, 148, 150, 158–160, 166
matrixcache, 124
Maxwell-, 29, 44, 47, 66, 67, 149, 151, 169
Maxwellgleichungen, 27, 31, 48, 49, 67, 169
MDF, 177
MeasureStringTension, 181
Mechanismus, 7, 11, 29, 32, 66, 153, 154, 166
Mehrgitterverfahren, hierarchische, 2, 74, 97, 113, 117
Menge klassischer Teilchen, 40
Mesa, 182, 183, 210
Mesonen, 7, 12, 17, 21, 22, 56, 168
Messwerte, 137
Metrik, 120, 193
MGGHAT, 106
Minisimplices, 106
mixed phase model, 11, 207
Modell, 1, 10–14, 23, 30–35, 37, 39–46, 48–50, 52, 61, 62, 64, 72, 84, 97, 132, 133, 135, 136, 138, 139, 143, 146, 153, 154, 157–159, 166–168
Modell, chromodielektrische, 10, 12, 37
Modell, chromodielektrischen, 14, 33, 35, 37, 44, 48, 51, 52, 61, 66, 139, 146, 198
Modell, chromodielektrisches, 35, 63
Modell, dem, 13, 48
ModelParameters, 139
molekulardynamische, 13, 41, 64, 148
molekulardynamischen, 12, 151
molekulardynamisches, 42
Moleküle, 69
MovingSolitonStability, 181
MPEG, 182
MUESLEA, 2, 3, 63, 80, 92, 94, 96, 97, 100, 106, 111, 113, 114, 117, 121, 122, 128, 177, 180, 182, 185
Multigrid-V-Cycles, 120
Multipletts, 21

Nachbarblock, 111–113, 121
Nachbarsimplexes, 89
Nambu-Jona-Lasinio-Modells, 12
named pipe, 180
Nebenprodukt, 14, 63, 89, 114
negative, 92, 94
negativen, 81, 92
neighblock, 121
neighline, 121
netpbm, 183, 211
Neumannschen Randbedingungen, von, 122
Neutronen, 6, 8, 153
Neutronenstern, 8
no kids, 94
Node, 97, 100, 121
NodeRegister, 121

Numerical Recipes, 97, 205

Oktave, 210
OpenGL, 182, 183, 210
Output, 132, 181

parallel transportiert, 57
parent simplex, 88
parent, 92, 94
ParticleList, 77, 182
Partition, 78, 86, 89–91, 93, 94, 96, 97, 100–102, 114, 117
partitioniert, 75
Paulikaskade, instantane, 72
PDEquation, 113, 117, 128
PDESolver, 117, 118
Perkolationsprozesses, 160
Perl, 180, 183, 210
Phasenraumabstand, 71, 73
PhaseSeparation, 181
PHI3FENCE, 94, 96
PHI8FENCE, 94
philosophische, 136
Photoshop, 211
pnmcrop, 183
pnmtops, 183
positive, 92, 94
positiven, 77, 92
Potential, farbelektrostatisches, 13, 69, 128
Potentials, farbelektrostatischen, 65
ppmbrighten, 183
ppmtogif, 183
Projekte, 3, 151, 168, 181–183, 196
projizierte, 83
Protonen, 6, 8, 153
PSUtils, 183
Pulsare, 8

QCD-Gitterrechnungen, 7, 13, 29, 31
Quadrupel, 77, 193, 196
Quadrupel, geordnetes, 77
Quantenchromodynamik, 1, 6, 7, 10, 11, 13, 17, 23, 24, 27–34, 37, 38, 45, 56, 57, 60, 139, 143, 157, 158, 201–205, 207
Quantenelektrodynamik, 6, 23, 24
Quark-Gluon-, 52, 131, 132, 148, 167, 171
Quark-Gluon-Plasma, 5, 7–10, 30, 132, 166, 204
Quarkclustermodell, 11
Quarkconfinement ohne Farbe, 34
Quarkconfinement, 29, 34
Quarkmodell, statischen, 21, 23, 24
Quarkmodell, statisches, 21, 23
Quarks, 1, 6–8, 11–13, 17, 21–25, 27, 30, 31, 33, 39, 40, 45–49, 52–54, 56, 57, 60, 61, 64, 67, 70–73, 125, 132, 133, 135, 137, 141–143, 146, 148, 149, 151, 154, 158, 168, 173, 174, 201, 204
Quarks, klassischen, 41
quenched approximation, 31
quick hack, 183

Randbedingungen, 2, 23, 31, 65, 67, 73–75, 83, 85, 98, 111, 117, 122, 128
Rapiditätsspektren, hadronischen, 160
Rapiditätsverteilung, 151, 160
Ratengleichungen, chemischen, 12
Raumzeitpunkt, an einem, 57
Rauschen, numerisches, 150
ReComputeMatrices, 117, 120
RedRelax, 111, 117, 118, 121
RedSparseMatrix, 121
Refine, 93
refinement cell, 91
reflected domain partition, 89
Regge, 204
Regge-Poltheorie, 17
Regge-Trajektorien, 17
rein hadronisch, 9
Relax, 2, 121
Relaxation, 2, 107, 111, 113–117, 120, 124, 140, 186

Relaxation, roten, 111
Relaxation, schwarzen, 113, 114
Relaxationen, schwarze, 113
relaxblock, 121
relaxieren, 111, 121
Residuums, 74, 116
Resultate, 2, 3, 10, 14, 30, 31, 83, 128, 135, 140, 148, 154, 159, 165, 168

Saiten, 18
sanfte, 150
Schrödingergleichung, 6
schwarzen, 88, 107, 111–114
Science Fiction, 168
screening, 32
second-class constraint, 43
Seite, rechte, 73, 99, 111, 113, 114, 116, 122, 125, 127, 195
Sektors, feinen, 110
seltsamen, 17
Shred, 93
Signatur, 9
Simplex, 76, 92, 93, 96, 97, 100, 124
Simplexachse, 88, 91
Simplexkoordinaten, 2, 77, 78, 80, 119, 120, 193, 196
Simplextyp, 88
Simplexwald, 2, 94, 96, 188, 198
Simplices, 75–78, 80, 82, 84–91, 93–96, 100–102, 106, 107, 117, 119–122, 124, 188, 193, 194, 198, 206
Simulationsuniversum, 153, 165
simulieren, 94, 131, 148
SingleQuark, 181
Site, 97, 100
sparse, 76, 85
Spiegelladungen, 84, 129, 131, 165, 169
Spielplatz, als, 182
Spielregeln, 60
Stabilitätsbedingung, 86, 89
staggered leapfrog, 69
stärker, 85, 86, 90, 128, 146
STDIN, 180
STDOUT, 179, 180
StoppedFireballHadronization, 181
stören, 132
strange, 9, 22, 23, 53, 136, 137
strangeness distillation, 168
StrangenessDistillation, 181
string flip, 146
String-String-Wechselwirkung, 2, 146
StringFusion, 181
Stringkonstante, 145
Stringmodell, 1, 17, 18, 27
Strings, 1, 13, 17, 18, 32, 126, 143, 145, 146, 148
Stringspannung, 2, 18, 20, 27, 62, 131, 139, 142, 143, 149, 166
StringStability, 181
Stringtheorie, 21
Strukturbedingung, 91
support, 76
supporting node, 76
SYMMETRIC, 113
Szenario, abelschen, 43

Temperatur, 3, 5, 7, 29, 30, 132, 151, 154, 157, 160, 161, 168, 201
Temperaturen, 5, 8, 9, 31, 62, 72, 142, 151, 154, 161, 163, 167
Tennisbälle, 13
Testfunktion, 74, 75
The Dancing Wu Li Masters, 171
Theorie, 6, 7, 10, 21, 23, 24, 30, 31, 33, 83
Thresholds, 54
Tk, 183, 210
Trägerknoten, 75, 76, 78, 79, 81, 82, 107, 109
Trägerknoten, roten, 109
Trägerknoten, schwarze, 108, 109
TransparentFireballHadronization, 181
Transporttheorie, semiklassische, 12, 35

trial and error, 128
TweakNeigh, 93

überschwingt, 81
uname, 184
unendlich, 31, 32, 34, 41, 50, 51, 73, 85, 131, 139
Unichor, 209
UNIX, 171, 179, 180, 182

V-Zyklus, 116–118, 122–124, 192
VCylce, 120
Verfeinerung, adaptiven, 63, 84, 85, 97
Verfeinerungszelle, 91–93, 107, 108, 110, 111, 113, 114, 117, 121, 196
Verlauf der Gesamtenergie, 154
verschachtelt, 85, 114
Verwaltungsobjekte, 106
Von Hand, 96

Watchdog, 96
WaveEquationStability, 181
Wechselwirkung zwischen Strings, 146
Wechselwirkung, hadronische, 6
Wechselwirkung, starke, 27
Weißen Zwerg, 8
Wong, 43, 44, 165, 205
Wong-Gleichung, 43

X11, 182
xmgr, 183, 211
xv, 211

Yang-Mills, 39, 43, 205
Yang-Mills-Gleichungen, 27, 29, 31, 42
YoyoModeStability, 181

Zahl, dielektrischen, 37
Zellraum, 108, 110, 111
zeltförmige, 78, 79
Zusammenfassung und Ausblick, 3, 14, 165
Zustandsgleichung, nichtideale, 11

"Sie sprachen kein Wort. Doch ehe sie über den Zaun kletterten, gaben sie einander die Hand. Es war, als gäben sie sich ein stummes Versprechen. Ein Versprechen, das sich mit Worten gar nicht ausdrücken lässt."
E. KÄSTNER *[1]*

Danksagung

Herrn Prof. Dr. Ulrich Mosel danke ich für die freundliche Aufnahme in das Institut für Theoretische Physik der Justus-Liebig-Universität Gießen, seine Themenstellung und Förderung dieser Arbeit. Auch danke ich ihm für seine Unterstützung des DFG-Projekts DFG-MTA 101/1998, das mir drei Reisen nach Budapest ermöglichte.

Für die angenehme Zusammenarbeit, die in Form einer gemeinsamen Veröffentlichung Früchte trug, möchte ich mich herzlich bei Dr. Tamás Biró vom kernphysikalischen Institut der ungarischen Akademie der Wissenschaften (KFKI) bedanken. Er hat mich in vielen Gesprächen und manchen Disputen physikalisch und persönlich beraten und weitergebracht.

Dr. Carsten Greiner und Dr. Stefan Leupold zeigten großes Interesse an meiner Arbeit, machten mich oft auf interessante Literatur aufmerksam und berieten mich bei physikalischen und sonstigen Problemen. Für ihre konstante Unterstützung bin ich beiden sehr zu Dank verpflichtet.

Dr. Stefan Loh und Dr. Markus Thoma danke ich für viele fruchtbare Gespräche über physikalische und private Themen sowie für ihr fortgesetztes Interesse an dieser Arbeit.

Der Gemeinschaft der Doktoranden und Diplomanden des Instituts danke ich für die angenehme, von gegenseitiger Anteilnahme und feinsinnigem Humor gekennzeichnete Atmosphäre. Von Diskussionen mit Dr. Jörn Häuser habe ich besonders profitiert; ihm sei nochmals für einen gewissen Telefonanruf gedankt, der mir sehr geholfen hat.

Elke Jung, Christiane Pausch, Dr. Jochen Geiß, Klaus Schertler und Frank Hofmann danke ich für die (system-)administrative Hilfe.

Beim "harten Kern" des Collegium Musicum Vocale ("Unichor"), bei gewissen dem Unichor nahestehenden Personen sowie bei allen meinen

anderen Freunden möchte ich mich für die persönliche, psychologische und musikalische Beratung und Kritik bedanken, die mir u. a. in zahl- und endlosen Sitzungen in der "Oktave" zuteil wurde und in jeder Hinsicht prägend wirkte. Auf die namentliche Nennung einzelner soll verzichtet werden (so kann ich auch niemanden dabei versehentlich auslassen); wer sich angesprochen fühlt, ist auch bestimmt gemeint.

Danken möchte ich, *last but not least*, meinen Eltern für die Unterstützung meines über neunjährigen Studiums und die Bereitstellung eines Kellerbiotops.

Etwas unüblich vielleicht, aber keineswegs unpassend: an dieser Stelle möchte ich den vielen Autoren qualitativ hochwertiger, freier[2] Software danken. Ohne deren Bereitschaft, ihre Werke der Öffentlichkeit zur freien Nutzung und Weiterentwicklung zur Verfügung zu stellen, hätte die vorliegende Arbeit (und sicher Tausende anderer, ähnlicher Arbeiten) nicht in dieser Form bewältigt werden können. Viele Menschen benutzen die im folgenden erwähnte Software sogar, ohne über die selbstlose Freigabe von seitens der Autoren überhaupt informiert zu sein.

Im einzelnen danken möchte ich:

- Richard Stallman, stellvertretend für die Free Software Foundation und das GNU-Projekt, u. a. für den GNU Emacs, den phantastischen GNU C++ Compiler gcc und Debugger gdb, und die vielen, vielen GNU-Systemtools von ar bis yes.
- Dorothea Lütkehaus und Andreas Zeller für den Data Display Debugger DDD.
- Linus Torvalds, stellvertretend für die Internet-Gemeinde der Linux-Kernel-Hacker, für den stabilen und leistungsfähigen Betriebssystemkern Linux.
- Dem XFree86 Project, Inc., für den freien X-Server XFree86.
- Brian Paul für Mesa, den berühmten OpenGL-Clone. Alle 3D-Grafiken dieser Arbeit wurden mit Mesa erstellt.
- Larry Wall für Perl, Jon Ousterhout für Tk und Nick Ing-Simmons für Perl/Tk. Die graphische Oberfläche des CDM-Programms wurde unter Perl/Tk programmiert. Ein Backupskript (geschrieben in Perl) rettete mich bei einem Plattencrash vor einem Daten-GAU, und viele Nachbearbeitungen von Programm-Outputfiles wurden durch Perl sehr erleichtert oder überhaupt erst ermöglicht.

[2]Frei wird hier im Sinne der GNU General Public License (GPL), der Library General Public License (LGPL), der BSD License oder der Artistic License verwendet. In diesem Sinn ist freie Software keineswegs Freeware, sie ist vielmehr durch ein "Copyleft" (Richard Stallman) geschützt.

- Paul Turner und Evgeny Stambulchik für das Grafikprogramm ACE/gr, auch bekannt als xmgr, mit dem viele der zweidimensionalen Diagramme dieser Arbeit erstellt wurden.

- Jef Poskanzer (stellvertretend) für das umfangreiche Konvertierungspaket netpbm für Grafikformate, Andy Hung für die MPEG-Codierungssoftware, der Independent JPEG Group für die JPEG-Codierungssoftware, Angus Duggan (stellvertretend) für die nützlichen Postscript-Utilities psutils.

- John Bradley für das Grafikprogramm xv[3].

- Rik Faith für seine umfangreiche Sammlung von Systemutilities (z. B. apmd).

- dem GIMP-Entwicklungsteam für den GIMP (neben dem der Photoshop wie ein Clone aussieht).

- Aladdin Software für Ghostscript[4] und Tim Theisen für Ghostview.

- schließlich Donald Knuth, Leslie Lamport und Thomas Esser für TEX, LATEX und TeTEX.

Auch wenn ich fast sicher bin, immer noch wesentliche Beiträge vergessen zu haben, war jedes einzelne der genannten freien Programme enorm hilfreich bei meiner Arbeit.

[3] frei für den nichtkommerziellen Gebrauch
[4] jeweils vorletzte Version ist frei