

Numerische und algebraisch-graphentheoretische Algorithmen für korrelierte Quantensysteme

Von der Fakultät für Mathematik und Physik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades
Doktor der Naturwissenschaften
(Dr. rer. nat.)
genehmigte Dissertation
von

Diplom-Chemiker

Martin Paech

geboren am 4. September 1974
in Pasewalk

2015

Erklärung der Selbständigkeit

Hiermit versichere ich, Martin Paech,
daß ich die vorliegende Arbeit selbständig verfaßt
und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe,
alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen
übernommen wurden, als solche kenntlich gemacht sind
und die Arbeit in gleicher oder ähnlicher Form
noch keiner Prüfungsbehörde vorgelegen hat.

Lohra-Weipoltshausen,
im November 2014

Referent: Prof. Dr. Eric Jeckelmann
Koreferent: Priv.-Doz. Dr. Walter Apel
Tag der Promotion: 6. Februar 2015

Kurzzusammenfassung

Der hiermit vorgelegte Beitrag zu numerischen und algebraisch-graphentheoretischen Algorithmen für korrelierte Quantensysteme ist zweigeteilt. Bedingt ist dies durch die Methodik, nicht die Thematik: Die Wechselwirkung der Elektronen untereinander in Festkörpern zu behandeln, stellt eine zentrale Herausforderung für die Theoretische Physik dar, und entsprechend realitätsnähere Vielteilchenmodelle können jenseits weniger exakter Lösungen im Allgemeinen nur approximativ mittels Verfahren der Numerischen Mathematik untersucht werden. Als Maßstab für die Näherungsqualität kommt folglich Sonderfällen, welche beispielsweise noch in Störungsreihen prinzipiell – abgesehen vom Aufwand – bis zu beliebigen Ordnungen entwickelbar und bis zu gegebener Ordnung ebenfalls exakt gelöst sind, herausragende Bedeutung zu.

Wesentliches Problem numerischer Verfahren ist neben statistischen und systematischen Fehlern die vergleichsweise geringe Größe des abbildbaren quantenmechanischen Systems, sofern dessen Eigenschaften auf makroskopischen Skalen, also im Thermodynamischen Limes berechnet werden sollen. Im Falle der Dynamischen Dichtematrix-Renormierungsgruppe (DDMRG) und verwandter Verfahren, die frequenzabhängige Korrelationsfunktionen quasi-eindimensionaler endlicher Gittermodelle mit hoher Genauigkeit liefern, fließen diese *finite-size*-Effekte in die verfahrensspezifische verbreiternde Faltung der Ergebnisse mit einem somit nicht vollständig bekannten Kern ein. Um charakteristische Funktionsmerkmale, vor allem Singularitätstypen, in den gewonnenen Spektren besser physikalisch interpretieren zu können, ist folglich eine „blinde“ Entfaltung vonnöten, welche partiell die technisch bedingte Glättung vermindert, nicht aber das Problem der endlichen Gitterplatzanzahl wieder hervortreten läßt. Zwar wurden dafür bereits mehrere Prozeduren entwickelt, die oft aber an Modell und Eigenschaft angepaßt und somit nur eingeschränkt übertragbar sind. Im *ersten Teil* dieser Arbeit wird eine auf universelle Verwendbarkeit ausgelegte Methode vorgestellt, welche die Berechnung der positiv-semidefiniten und abschnittsweise glatten Spektralfunktion des unendlich ausgedehnten Systems aus den DDMRG-Daten erlaubt. Nach genauer Beschreibung der beteiligten Algorithmen – effektiv wird eine *least-square*-Optimierung unter nicht-linearen physikalisch motivierten Bedingungen und Zuhilfenahme eines Rauschdämpfungsverfahrens durchgeführt – erfolgen umfassende Probeanwendungen auf typische Funktionsverläufe, woran sich die Untersuchung verschiedener realer Spektren aus DDMRG-Rechnungen anschließt. Obschon die Ergebnisse noch Artefakte aufweisen, zeigt sich dabei bereits ein

erhebliches Leistungsvermögen dieses Ansatzes – im Besonderen am eindimensionalen halbgefüllten Hubbardmodell bei Bestimmung der Zustandsdichte der paramagnetischen Mott-Isolatorphase im Vergleich mit feld- und störungstheoretischen Resultaten.

Aus Sicht der rechnergestützten Vielteilchentheorie stehen der Numerik computeralgebraische Verfahren als Referenz gegenüber. Unter den auf dieser Grundlage noch exakt behandelbaren Modellen kommt jenem HUBBARDS durch die einfachst denkbare Berücksichtigung der Elektronenkorrelation hohe physikalische Relevanz zu. Bei Einschränkung auf starke Wechselwirkung (Mottisolator) und unendlicher Kooordinationszahl (Bethegitter) ist eine Kato-Takahashi-Störungsrechnung ausführbar, kraft welcher mit einem neuartigen kombinatorisch-graphenbasierten Zugang, getragen von Ganzzahlarithmetik, bisher bei halber Bandfüllung Grundzustandsenergie und kritische Parameter für den Mottischen Phasenübergang bis einschließlich elfter Ordnung in der Wechselwirkung bestimmt wurden. Im *zweiten Teil* dieser Arbeit folgt zunächst eine detaillierte Beschreibung dieser mehrstufigen Methode, die zuerst sämtliche Gitterausschnitte (Butcherbäume) aufbaut, nachfolgend den effektiven Hamiltonoperator nach der Wechselwirkung für die Bestimmung der Gewichte der Elektronentransferprozesse entwickelt, dann alle beitragenden Hüpfsequenzen auf den Butcherbäumen erzeugt und abschließend aus den Prozessen und Sequenzen den Beitrag zur Grundzustandsenergie berechnet. Dies erfolgt in einem für künftige Erweiterungen (beispielsweise zur Greenschen Funktion hin) offenen diskret-mathematischen Formalismus, auf dessen Grundlage das Verfahren für Hochleistungsrechner konsequent neu programmiert und parallelisiert wurde, wobei auch eine begleitende Zusammenarbeit mit der Theoretischen Informatik starke Optimierungen an den verwendeten komplexen Algorithmen aus Kombinatorik, Zahlen- und Graphentheorie unterstützt hat. Die hohe Effizienz der Neuimplementierung wird durch die technisch aufwendige Bestimmung der Reihenkoeffizienten bis einschließlich fünfzehnter Ordnung belegt; insbesondere ist mit einer davon ausgehenden Extrapolation auf die kritische Wechselwirkung in unendlicher Ordnung erstmals eine eindeutige Bewertung von Lösungen der Dynamischen *mean-field*-Gleichungen zugunsten von Quanten-Monte-Carlo als *impurity solver* möglich geworden.

Schlüsselworte in alphabetischer Reihenfolge: Bethegitter, Butcherbäume, Dynamische Dichtematrix-Renormierungsgruppe, (blinde) Entfaltung, Hochleistungsrechnen, (erweitertes) Hubbardmodell, Kato(-Takahashi)-Störungstheorie, Mottübergang, Rauschfilterung, Starkwechselwirkungslimes

Abstract

This thesis deals with numerical and algebraic graph-theoretical algorithms for correlated quantum systems. It is split into two parts, and both of them develop methods for treating the many-body problem: One of the greatest challenges in theoretical physics is the investigation of the electron-electron interaction in solids because the more “realistic” a many-particle model becomes, the less likely there is an exact solution. In general, only approximative results based on numerical methods are available; hence there is a need for criteria to judge the quality of the approximations. Consequently, special cases of “quasi-analytical” solutions are of great importance as benchmarks—in particular, series expansions: disregarding the computational costs, the series coefficients could be calculated to arbitrary order, and up to and including that order, the solution is exact.

The fundamental problem of numerical methods is—beside statistical and systematic errors—the comparatively small sizes of the treatable quantum systems, because in most cases, the thermodynamic limit is of interest. Frequency-dependent correlation functions of finite quasi one-dimensional lattice models are provided by the dynamical density-matrix renormalization group (DDMRG) and related techniques, but they are convolved by a method-specific kernel, which includes contributions from finite-size effects that are not fully known. For a better physical interpretation of characteristic features of the spectral functions, in particular to identify the types of singularities, a “blind” deconvolution is necessary. Such a post-processing should reduce the smoothing caused by the DDMRG procedure, but it should not reveal the effects of the finite system size. For this purpose, several procedures have been developed; they are often optimized for specific models and properties and thus they are not freely transferable between different models and properties. In the *first part* of this work, a universal method is presented that computes the positive semidefinite and piecewise smooth spectral function of the infinite quantum system from the DDMRG data. This procedure consists of a least-square optimization under non-linear constraints, which are provided by the physics of the model, and a supporting noise-reduction step. A detailed description of the involved algorithms is followed by comprehensive tests of typical spectral functions and an examination of several real spectra from DDMRG calculations. Although the results show some artifacts, this deconvolution method demonstrates already significant potential, especially in the case of the density of states of the paramagnetic Mott insulating phase of the one-dimensional half-filled Hubbard model, in comparison to field theory and perturbation theory.

From the viewpoint of computational many-particle physics, computer algebraic methods are a counterpart of and a reference for numerical methods, but only a few models are treatable with algebraic techniques. Among them the Hubbard model is of high interest, because it is the simplest possible model that takes electron correlation into account. Under the restriction of strong interaction (Mott insulator) and of infinite coordination number (Bethe lattice), one can utilize the Kato-Takahashi perturbation theory. This has already been studied at half-filling with a novel combinatorial and graph-theoretical algorithm, which is based on integer arithmetic, and the ground-state energy and the critical parameters for the Mott insulator are determined up to and including eleventh order in the interaction. In the *second part* of this work, at first a detailed description of this complex method is presented: initially the generation of all cut-outs of the lattice (Butcher trees), then the series expansion of the effective Hamiltonian in the inverse interaction strength and the computation of the weights of the electron-transfer processes, followed by the generation of all contributing hopping sequences on the Butcher trees, and finally the calculation of the contribution to the ground-state energy from the processes and the sequences. For future extensions of the method, for example for the series expansion of the Green's function, a formalism using discrete mathematics is introduced, and on this basis, the whole method is consistently implemented and parallelized for high-performance computers. This work was accompanied by a cooperation with theoretical informatics, which has supported the optimization of the complex algorithms from combinatorics, number theory and graph theory. The high efficiency of the new implementation is demonstrated by the challenging computation of the series coefficients for the ground-state energy up to and including fifteenth order. Based on this result, an extrapolation of the critical coupling to infinite order is performed, and for the first time this series expansion allows a clear comparison of the solutions of the dynamical mean-field theory in favor of quantum Monte-Carlo as impurity solver.

Keywords in alphabetical order: Bethe lattice, Butcher trees, (blind) deconvolution, dynamical density-matrix renormalization group, high-performance computing, (extended) Hubbard model, Kato(-Takahashi) perturbation theory, Mott transition, noise reduction, strong-coupling limit

3.1.1.	Notation	49
3.1.2.	Gemeinsame Parameter; δ -Funktion	49
3.1.3.	Gliederung der Abbildungen	50
3.1.4.	Rauschunterdrückung	50
3.2.	Lorentzverteilung	50
3.2.1.	Charakterisierung	50
3.2.2.	Ergebnisanalyse	51
3.3.	Halbelliptische Verteilung	52
3.3.1.	Charakterisierung	52
3.3.2.	Ergebnisanalyse	53
3.4.	Quadratwurzeldivergenz; Divergenzartefakt	54
3.4.1.	Charakterisierung	54
3.4.2.	Ergebnisanalyse	54
3.5.	Asymmetrische Quadratwurzeldivergenz	56
3.5.1.	Charakterisierung	56
3.5.2.	Ergebnisanalyse	56
3.6.	Stufenfunktion; Stufenartefakt	57
3.6.1.	Charakterisierung	57
3.6.2.	Ergebnisanalyse	58
3.7.	Deltafunktion nahe eines Kontinuums; Bandartefakt	59
3.7.1.	Charakterisierung	59
3.7.2.	Ergebnisanalyse	60
3.8.	Deltafunktion zwischen zwei Bändern	61
3.8.1.	Charakterisierung	61
3.8.2.	Ergebnisanalyse	62
3.9.	Untersuchung von Effekten endlicher Systemgröße	63
3.9.1.	Kette stark gebundener Elektronen, Mitte	63
3.9.2.	Kette stark gebundener Elektronen, Enden	64
3.10.	Abbildungen	64
3.10.1.	Analytische Prüffunktionen	64
3.10.2.	Effekte endlicher Systemgröße	65
4.	Anwendung auf Spektralfunktionen	82
4.1.	Vorbemerkungen	82
4.1.1.	Modelle	82
4.1.2.	Gliederung der Abbildungen	83
4.2.	Zustandsdichte des eindimensionalen Hubbardmodells	83
4.2.1.	Keine Wechselwirkung	83
4.2.2.	Endliche Wechselwirkung	85
4.3.	Zustandsdichte einer Luttingerflüssigkeit	87
4.3.1.	Photoemission	87
4.3.2.	Inverse Photoemission	87
4.4.	Optische Leitfähigkeit im Hubbard- und Peierls-Hubbard-Modell	89

4.4.1.	Optische Leitfähigkeit des Mott-Hubbard-Isolators	90
4.4.2.	Strom-Strom-Korrelationsfunktion im Hubbardmodell	91
4.4.3.	Optische Leitfähigkeit des Peierlsisolators	93
4.4.4.	Reduzierte optische Leitfähigkeit des Mott-Peierls-Isolators	94
4.5.	Abbildungen	95
5.	Zusammenfassung und Ausblick	112
5.1.	Derzeitiger Stand	112
5.2.	Aussichten	113
5.2.1.	Offene Probleme	113
5.2.2.	Physikalisch motivierte Erweiterungen	113
5.2.3.	Mathematisch motivierte Erweiterungen	114
5.2.4.	Datenverarbeitungstechnisch motivierte Erweiterungen	114
Anhang:	Ergänzungen zum Rechenverfahren	115
1.	Programmmodifikationen	115
1.1.	Rückfaltung mit einer Gaußverteilung	115
1.2.	Alternativer Ablauf des Gesamtverfahrens	116
2.	Probeanwendungen auf die Zustandsdichte	117
Literatur- und Quellenverzeichnis	119
Datenverarbeitungsressourcen	122
Teil II.	Graphentheoretische Störungsrechnung	125
1.	Einleitung und physikalische Grundlagen	128
1.1.	Modelle und deren Parameter; Gitter	128
1.1.1.	Hamiltonoperator des Hubbardmodells	128
1.1.2.	Hamiltonoperator des Falicov-Kimball-Modells	129
1.1.3.	Bethegitter	129
1.2.	Relevante Grundzustandseigenschaften	129
1.3.	Lösung des Eigenwertproblems; Ausgangszustände	130
1.4.	Reihenentwicklung nach der Wechselwirkung	131
1.4.1.	Allgemein	131
1.4.2.	Beispiel	132
1.5.	Prozesse, Bethegitterausschnitte, Sequenzen	133
1.5.1.	Hüpfprozesse	133
1.5.2.	Gitterparkettierung	134
1.5.3.	Spinbelegte Gitterausschnitte	137
1.5.4.	Ausgangszustände	138
1.5.5.	Hüpfsequenzen am Beispiel erster Störungsordnung	139
1.6.	Zusammenfassung	140
2.	Mathematische Grundlagen	141
2.1.	Begriffe und Notationen	141

2.1.1.	Graph, Teilgraph	I42
2.1.2.	Grad (Koordinationszahl)	I43
2.1.3.	Weg, Schleife	I43
2.1.4.	Zusammenhang	I43
2.1.5.	Baum; Wald	I43
2.1.6.	Wurzel, Blatt	I43
2.1.7.	Eckenklasse, Baumrotation	I44
2.1.8.	Verzweigungsgrad, Ast	I44
2.1.9.	Ausgerichtete und nicht ausgerichtete Bäume	I44
2.1.10.	Nicht ausgerichtete Bäume: Cayleybäume, Permutationsbäume	I44
2.1.11.	Ausgerichtete Bäume: Binärbäume	I45
2.1.12.	Isomorphismus	I46
2.1.13.	Topologische Äquivalenz, Topologiekategorie	I46
2.1.14.	Astordnung	I46
2.1.15.	Farbe, Färbung	I47
2.2.	Problemspezifische Objekte und Operationen	I48
2.2.1.	Bethegitter; Translationsinvarianz bzw. -periodizität	I48
2.2.2.	Ausgerichtete Bäume: Butcherbäume	I49
2.2.3.	Klammer- und Binärdarstellung	I53
2.2.4.	Baumgruppe	I54
2.2.5.	Baumprodukt	I55
2.2.6.	Baumexponent	I56
2.2.7.	Baumsumme	I58
2.3.	Zusammenfassung	I60
3.	Optimierter Gesamtalgorithmus	I60
3.1.	Erzeugung der ungefärbten Butcherbäume	I63
3.1.1.	Allgemeines Vorgehen	I63
3.1.2.	Beispiel	I67
3.2.	Erzeugung gefärbter Butcherbäume	I69
3.2.1.	Allgemeines Vorgehen	I69
3.2.2.	Beispiel	I72
3.3.	Reihenentwicklung des effektiven Hamiltonoperators	I76
3.3.1.	Vorbemerkungen	I76
3.3.2.	Grundsätzliches Vorgehen für halbe Bandfüllung	I83
3.3.3.	Beispiel	I88
3.3.4.	Optimierungen	I92
3.3.5.	Graphische Darstellung von Prozessen	I93
3.4.	Generierung der Hüpfsequenzen	I94
3.4.1.	Farbseparation	I95
3.4.2.	Ausgangs- und Zwischenzustände	I97
3.4.3.	Hüpfkantenmengen	I99
3.4.4.	Hüpfschritte	200

3.4.5.	Kantenklassen und Metagraphen	204
3.4.6.	Hüpfsequenzen, deren Eigenschaften und Notation	209
3.4.7.	Doppelbesetzungsverlauf und dessen graphische Darstellung	210
3.4.8.	Hüfspintupel	213
3.4.9.	Iterativer Rahmen	214
3.4.10.	Beispiel	218
3.4.11.	Rekursive Vorbereitungsstufe	222
3.4.12.	Beispiel	225
3.4.13.	Rekursiver Kern	227
3.4.14.	Beispiel	230
3.4.15.	Präemptive Hüpfspinwahl	240
3.5.	Berechnung des Sequenzbeitrages zum Reihenkoeffizienten	246
3.5.1.	Allgemein	246
3.5.2.	Beispiele	249
3.6.	Zusammenfassung	254
4.	Ergebnisse	255
4.1.	Grundzustandsenergie und Doppelbesetzungszahl	256
4.2.	Vergleich zur Numerik	256
4.3.	Kritische Parameter des Hubbardmodells	262
4.3.1.	Definitionen	262
4.3.2.	Systematische Extrapolationsanalyse	263
4.3.3.	Bewertung der DMFT-Befunde	270
5.	Parallelisierte Implementierung	271
5.1.	Technische Details	272
5.1.1.	Problemkomplexität	272
5.1.2.	Laufzeit- und Speicherbedarf	275
5.1.3.	Programm und Rechner	280
5.2.	Weitergehende Optimierungen	283
5.2.1.	Ganzzahlbehandlung der Energiebeiträge	284
5.2.2.	Feinere Auftragsgranularisierung; Systemlastprofile	289
6.	Zusammenfassung und Ausblick	298
6.1.	Physikalische Aspekte	298
6.1.1.	Derzeitiger Stand	298
6.1.2.	Allgemeine Aussichten	299
6.1.3.	Beliebige Gitterfüllungen; niedrigdimensionale Systeme	300
6.1.4.	Zustandsdichte und Greensche Funktion	301
6.1.5.	Bose-Einstein-Kondensate in optischen Gittern	303
6.1.6.	Konzeptionelle Übertragung auf andere Reihentwicklungen	303
6.2.	Mathematische Aspekte	303
6.2.1.	Derzeitiger Stand	303
6.2.2.	Allgemeine Aussichten	304
6.2.3.	Aufbau und Wagsuche auf Zustandsgraphen	306

6.3.	Datenverarbeitungstechnische Aspekte	309
6.3.1.	Derzeitiger Stand	309
6.3.2.	Allgemeine Aussichten	310
6.3.3.	Vorabberechnung der Energiebeiträge	314
6.3.4.	Skalierungsverhalten und Amdahlsches Gesetz	318
Anhang: Ergänzungen zum Rechenverfahren		323
1.	Höhere Algorithmen	323
1.1.	Erzeugung aller Doppelbesetzungsverlaufstypen	323
1.2.	Bestimmung eines Doppelbesetzungsverlaufstyps	329
1.3.	Berechnung des Reihenkoeffizienten	332
2.	Hilfsfunktionen	335
2.1.	Sortierverfahren	335
2.2.	Permutationen	336
2.3.	Partitionen	341
2.4.	Mangoldtzahlen	342
3.	Tafeln	344
3.1.	Zweigefärbte Butcherbäume	344
3.2.	Viergefärbte Butcherbäume	346
3.3.	Doppelbesetzungsverlaufstypen	349
Literatur- und Quellenverzeichnis		355
Datenverarbeitungsressourcen		361
Danksagung		365
Wissenschaftlicher Werdegang		367

Abbildungsverzeichnis

Teil 1. Numerische Entfaltungsverfahren		
	Schematische Darstellung der Renormierungsgruppenmethoden	22
1.1	Zustandsdichte D eines unendlich großen eindimensionalen Systems wechselwirkungsfreier Teilchen und deren Faltung \mathfrak{D}	26
1.2	Berechnung von D durch zwei Fouriertransformationen aus \mathfrak{D}	28
2.1	Programmablaufschemata der selbstkonsistenten Entfaltungsprozedur	35
2.2	Programmablaufschemata der Glättungsprozedur: Rahmenprogramm mit Extremumssuche	36

2.3	Programmablaufschemata der Glättungsprozedur: Lineare Interpolation des Spektrums zwischen benachbarten Extrema	37
2.4	Programmablaufschemata der Rückfaltungsprozedur	39
2.5	Programmablaufschemata der verbesserten selbstkonsistenten Entfaltungs- prozedur mit bidirektionalem Indextdurchlauf	40
2.6	Programmablaufschemata der verbesserten Glättungsprozedur mit Erhaltung des Spektralen Gewichts	43
2.7	Programmablaufschemata der Rediskretisierungs- und Regularisierungsprozedur	44
2.8	Vergleich zweier Regularisierungsverfahren	46
2.9	Programmablaufschemata des Gesamtverfahrens	47
3.1	Lorentzverteilung, $T^{\text{LD}}(\omega)$, und deren Faltung $\mathfrak{T}^{\text{LD}}(\omega)$	51
3.2	Entfaltung von $\mathfrak{T}^{\text{LD}}(\omega)$	51
3.3	Halbelliptische Verteilung, $T^{\text{SED}}(\omega)$, und deren Faltung $\mathfrak{T}^{\text{SED}}(\omega)$	52
3.4	Entfaltung von $\mathfrak{T}^{\text{SED}}(\omega)$	53
3.5	Quadratwurzeldivergenz, $T^{\text{SRD}}(\omega)$, und deren Faltung $\mathfrak{T}^{\text{SRD}}(\omega)$	54
3.6	Entfaltung von $\mathfrak{T}^{\text{SRD}}(\omega)$; Divergenzartefakt	55
3.7	Asymmetrische Quadratwurzeldivergenz, $T^{\text{ASRD}}(\omega)$, und deren Faltung $\mathfrak{T}^{\text{ASRD}}(\omega)$	56
3.8	Entfaltung von $\mathfrak{T}^{\text{ASRD}}(\omega)$	57
3.9	Stufenfunktion, $T^{\text{SF}}(\omega)$, und deren Faltung $\mathfrak{T}^{\text{SF}}(\omega)$	58
3.10	Entfaltung von $\mathfrak{T}^{\text{SF}}(\omega)$; Stufenartefakt	59
3.11	Deltafunktion nahe eines Kontinuums, $T^{\text{DPC}}(\omega)$, und deren Faltung $\mathfrak{T}^{\text{DPC}}(\omega)$	60
3.12	Entfaltung von $\mathfrak{T}^{\text{DPC}}(\omega)$; Bandartefakt	61
3.13	Deltafunktion zwischen zwei Bändern, $T^{\text{DPB}}(\omega)$, und deren Faltung $\mathfrak{T}^{\text{DPB}}(\omega)$	61
3.14	Entfaltung von $\mathfrak{T}^{\text{DPB}}(\omega)$	62
3.15	Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen aus 100 Gitterplätzen, Mitte, $\mathfrak{D}_\eta^{\text{TBCM100}}(\omega)$	63
3.16	Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen aus 50 Gitterplätzen, Enden, $\mathfrak{D}_\eta^{\text{TBC50}}(\omega)$	64
3.17	Entfaltung der künstlich verrauschten Lorentzverteilung, $\tilde{\mathfrak{T}}_\eta^{\text{LD}}(\omega)$	66
3.18	Entfaltung von $\tilde{\mathfrak{T}}_\eta^{\text{LD}}(\omega)$, Abbildung 3.17, mit unterschiedlich starker Rauschunterdrückung	67

3.19	Entfaltung der künstlich verrauschten Halbelliptischen Verteilung, $\tilde{\mathfrak{T}}_{\eta}^{\text{SED}}(\omega)$	68
3.20	Entfaltung von $\tilde{\mathfrak{T}}_{\eta}^{\text{SED}}(\omega)$, Abbildung 3.19, mit unterschiedlich starker Rauschunterdrückung	69
3.21	Entfaltung der künstlich verrauschten Quadratwurzeldivergenz, $\tilde{\mathfrak{T}}_{\eta}^{\text{SRD}}(\omega)$..	70
3.22	Entfaltung von $\tilde{\mathfrak{T}}_{\eta}^{\text{SRD}}(\omega)$, Abbildung 3.21, mit unterschiedlich starker Rauschunterdrückung	71
3.23	Entfaltung der künstlich verrauschten Asymmetrischen Quadratwurzel- divergenz, $\tilde{\mathfrak{T}}_{\eta}^{\text{ASRD}}(\omega)$	72
3.24	Entfaltung von $\tilde{\mathfrak{T}}_{\eta}^{\text{ASRD}}(\omega)$, Abbildung 3.23, mit unterschiedlich starker Rauschunterdrückung	73
3.25	Entfaltung der künstlich verrauschten Stufenfunktion, $\tilde{\mathfrak{T}}_{\eta}^{\text{SF}}(\omega)$	74
3.26	Entfaltung von $\tilde{\mathfrak{T}}_{\eta}^{\text{SF}}(\omega)$, Abbildung 3.25, mit unterschiedlich starker Rauschunterdrückung	75
3.27	Entfaltung der künstlich verrauschten Deltafunktion nahe eines Kontinuums, $\tilde{\mathfrak{T}}_{\eta}^{\text{DPC}}(\omega)$	76
3.28	Entfaltung von $\tilde{\mathfrak{T}}_{\eta}^{\text{DPC}}(\omega)$, Abbildung 3.27, mit unterschiedlich starker Rauschunterdrückung	77
3.29	Entfaltung der künstlich verrauschten Deltafunktion zwischen zwei Bändern, $\tilde{\mathfrak{T}}_{\eta}^{\text{DPB}}(\omega)$	78
3.30	Entfaltung von $\tilde{\mathfrak{T}}_{\eta}^{\text{DPB}}(\omega)$, Abbildung 3.29, mit unterschiedlich starker Rauschunterdrückung	79
3.31	Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen, Mitte, $\mathfrak{D}_{\eta}^{\text{TBCML}}(\omega)$	80
3.32	Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen, Enden, $\mathfrak{D}_{\eta}^{\text{TBCEL}}(\omega)$	81
4.1	Vergleich verschiedener Entfaltungsverfahren anhand der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathfrak{D}}_{0,08}^{U=0}(\omega)$	84
4.2	Vergleich verschiedener Entfaltungsverfahren anhand der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathfrak{D}}_{0,08}^{U=4}(\omega)$	86
4.3	Ausschnitt aus der entfalteten Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ im Bereich der Photoemission, $\tilde{\mathfrak{D}}_{0,1}^{\text{PE}}(\omega)$	88
4.4	Ausschnitt aus der entfalteten Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ im Bereich der inversen Photoemission, $\tilde{\mathfrak{D}}_{0,1}^{\text{IPE}}(\omega)$	89
4.5	Vergleich verschiedener Entfaltungsverfahren anhand der optischen Leitfähigkeit im eindimensionalen Hubbardmodell, $\tilde{\mathfrak{E}}_{0,1}^{\text{HMOC}}(\omega)$, bei $U = 3$..	91

4.6	Vergleich des Entfaltungsergebnis' der Strom-Strom-Korrelationsfunktion im Hubbardmodell, $\tilde{\mathfrak{E}}_{0,01}^{\text{CCCF}}(\omega)$, mit der analytischen Lösung, $\mathfrak{F}_{0,01}^{\text{DPC}}(\omega)$	92
4.7	Vergleich des Entfaltungsergebnisses der optischen Leitfähigkeit eines Peierlsisolators, $\tilde{\mathfrak{E}}_{0,005}^{\text{PIOC}}(\omega)$, mit der analytischen Lösung, $\mathfrak{F}_{0,01}^{\text{ASRD}}(\omega)$	93
4.8	Vergleich des Entfaltungsergebnisses der red. optischen Leitfähigkeit eines Mott-Peierls-Isolators, $\tilde{\mathfrak{E}}_{0,01}^{\text{PIROC}}(\omega)$, mit der analytischen Lösung, $\mathfrak{F}_{0,01}^{\text{DPB}}(\omega)$..	94
4.9	Entfaltung der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathfrak{D}}_{0,08}^{U=0}(\omega)$	96
4.10	Entfaltung von $\tilde{\mathfrak{D}}_{0,08}^{U=0}(\omega)$, Abbildung 4.9, mit unterschiedlich starker Rauschunterdrückung	97
4.11	Entfaltung der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathfrak{D}}_{0,08}^{U=4}(\omega)$	98
4.12	Entfaltung von $\tilde{\mathfrak{D}}_{0,08}^{U=4}(\omega)$, Abbildung 4.11, mit unterschiedlich starker Rauschunterdrückung	99
4.13	Entfaltung der Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ im Bereich der Photoemission, $\tilde{\mathfrak{D}}_{0,1}^{\text{PE}}(\omega)$	100
4.14	Entfaltung von $\tilde{\mathfrak{D}}_{0,1}^{\text{PE}}(\omega)$, Abbildung 4.13, mit unterschiedlich starker Rauschunterdrückung	101
4.15	Entfaltung der Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ im Bereich der inversen Photoemission, $\tilde{\mathfrak{D}}_{0,1}^{\text{IPE}}(\omega)$	102
4.16	Entfaltung von $\tilde{\mathfrak{D}}_{0,1}^{\text{IPE}}(\omega)$, Abbildung 4.15, mit unterschiedlich starker Rauschunterdrückung	103
4.17	Entfaltung der optischen Leitfähigkeit im eindimensionalen Hubbardmodell, $\tilde{\mathfrak{E}}_{0,1}^{\text{HMOC}}(\omega)$, bei $U = 3$	104
4.18	Entfaltung von $\tilde{\mathfrak{E}}_{0,1}^{\text{HMOC}}(\omega)$, Abbildung 4.17, mit unterschiedlich starker Rauschunterdrückung	105
4.19	Entfaltung der Strom-Strom-Korrelationsfunktion im eindimensionalen Hubbardmodell, $\tilde{\mathfrak{E}}_{0,1}^{\text{CCCF}}(\omega')$	106
4.20	Entfaltung von $\tilde{\mathfrak{E}}_{0,1}^{\text{CCCF}}(\omega')$, Abbildung 4.19, mit unterschiedlich starker Rauschunterdrückung	107
4.21	Entfaltung der optischen Leitfähigkeit eines Peierlsisolators, $\tilde{\mathfrak{E}}_{0,05}^{\text{PIOC}}(\omega)$	108
4.22	Entfaltung von $\tilde{\mathfrak{E}}_{0,05}^{\text{PIOC}}(\omega)$, Abbildung 4.21, mit unterschiedlich starker Rauschunterdrückung	109

4.23	Entfaltung der reduzierten optischen Leitfähigkeit eines Mott-Peierls-Isolators, $\tilde{\mathcal{E}}_{0,1}^{\text{PIROC}}(\omega)$	110
4.24	Entfaltung von $\tilde{\mathcal{E}}_{0,1}^{\text{PIROC}}(\omega)$, Abbildung 4.23, mit unterschiedlich starker Rauschunterdrückung	111
Anhang zu Teil I		
1.1	Programmablaufschema des alternativen Gesamtverfahrens	116
2.1	Vergleich von Gauß- und Lorentzrückfaltung anhand der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathcal{D}}_{0,08}^{U=4}(\omega)$	118
Teil II. Graphentheoretische Störungsrechnung		
	Zustandsdichte $D(\omega)$ in der Mott-Hubbard-Isolatorphase	126
3.1	Ablaufplan des Hauptprogramms	161
3.2	Konstruktion aller ungefärbten Butcherbäume mit bis zu $k = 5$ Ecken	164
3.3	Ablaufplan eines Unterprogramms für die Erzeugung der ungefärbten Butcherbäume $\mathcal{A}_{k,l}$	165
3.4	Ablaufplan des Unterprogramms für die Erzeugung der c gefärbten Butcherbäume ${}^c\mathcal{B}_{k,l,m}$	170
3.5	Ablaufplan des Unterprogramms für die Erzeugung der Hüpfprozesse bei halber Bandfüllung	184
3.6	Ablaufplan des „iterativen Rahmens“ für die Erzeugung der Hüpfsequenzen	215
3.7	Beispiel zur Sequenzierung: Zum Ausgangszustand $\mathcal{B}_{4,2,65}$ gehöriger Permutationsbaum ${}^m\mathcal{P}_{5,1}$	222
3.8	Ablaufplan des rekursiven Dekodierers der Baumzeichenkette	223
3.9	Ablaufplan des „rekursiven Kernes“ für die Erzeugung der Hüpfsequenzen	228
3.10	Ablaufplan des Unterprogramms für die Berechnung des Hüpfsequenzbeitrages zum Reihenkoeffizienten	247
4.1	Grundzustandsenergie und Doppelbesetzungszahl: Vergleich der Störungsrechnung mit DMFT-Ergebnissen	258
4.2	Graphischer Vergleich der Extrapolationsergebnisse für Grundzustandsenergie $E(U)$ und Doppelbesetzungszahl $D(U)$ mit DMFT-Ergebnissen	262
4.3	Kritische Wechselwirkung: Graphische Darstellung der wesentlichen Ergebnisse	266
4.4	Kritische Wechselwirkung: Graphische Darstellung weiterer Ergebnisse	267
4.5	Graphischer Vergleich der Ergebnisse der verschiedenen Extrapolations-schemata für die kritische Wechselwirkung U_c	268

4.6	Graphischer Vergleich der Ergebnisse der verschiedenen Extrapolations- schemata für U_c im Grenzfall $n \rightarrow \infty$	270
5.1	Graphische Darstellung der Daten zur Wachstumsanalyse	274
5.2	Veranschaulichung der Komplexität des Störungsrechnungsproblems	278
5.3	Systemlastprofil für die Berechnung des Reihenkoeffizienten der Ordnung U^{-13} des Hubbardmodells mit 1020 Prozeßfäden	290
5.4	Systemlastprofil für die Berechnung des Reihenkoeffizienten der Ordnung U^{-13} des Hubbardmodells mit 72 Prozeßfäden	296
6.1	Zustandsdichte $D_U(\omega)$ des Hubbardmodells für verschiedene Wechselwirkungsstärken U	302
6.2	Leistungszuwachsfaktor bei der Sequenzierung in Abhängigkeit von der Prozessoranzahl; Vergleich mit dem Amdahlschen Gesetz	319
Anhang zu Teil II		
1.1	Ablaufplan eines Unterprogramms für die Generierung der Doppelbesetzungsverlaufstypen	325
1.2	Ablaufplan eines Unterprogramms für die Bestimmung des Doppelbesetzungsverlaufstyps der Hüpfsequenz ..	331
2.1	Ablaufplan eines Unterprogramms zur direkten Erzeugung der u ten lexikographischen Permutation des k' -Tupels ${}^m G$	338
2.2	Einfacher Algorithmus zur Berechnung der Folge der Mangoldtzahlen \mathcal{M} ..	342

Tabellenverzeichnis

Teil II. Graphentheoretische Störungsrechnung

1.1	Alle Möglichkeiten für die Bildung von Bethegitterausschnitten bis einschließlich Störungsordnung $n = 7$	135
1.2	Alle Möglichkeiten für die Bildung von mit Elektronen besetzten Bette- gitterausschnitten in den Störungsordnungen $n = 1$ und $n = 3$	137
1.3	Alle beitragenden Ausgangszustände für Elektronentransferprozesse bis einschließlich Störungsordnung $n = 5$	138
2.1	Anzahl λ_k der Butcherbäume k ter Ordnung und deren Anzahl γ_k ohne topologisch äquivalente	150
2.2	Anzahl $\eta_{k,l}$ der Eckenklassen im Butcherbaum k ter Ordnung und l ter Struktur	150

2.3	Gesamtzahl zwei- und vierfarbiger Butcherbäume k ter Ordnung	151
2.4	Anzahl $\mu_{k,l}$ zwei- und vierfarbiger Butcherbäume k ter Ordnung und l ter Struktur, Aufschlüsselung von Tabelle 2.3	151
2.5	Anzahl ν_p der Partitionen von $p \in \mathbb{N}$	152
3.1	Gesamtzahl vierfarbiger Butcherbäume in Fortsetzung von Tabelle 2.3	173
3.2	Hilfstabelle zur Erzeugung der \mathcal{T} - und \mathcal{D} -Produkte für $i = 0, 1, \dots, 11$..	178
3.3	Anzahl der Prozesse nach verschiedenen Programmschritten	185
3.4	Anzahl der Prozesse nach verschiedenen Optimierungsstufen in Fortsetzung von Tabelle 3.3	194
3.5	Graphische Veranschaulichung der Hüpfprozesse bis einschließlich Störungsordnung $n = 5$	194
3.6	Schema zur Spin- bzw. Farbebenenentrennung für Elektronen und beliebige Fermionenmischungen	197
3.7	Gesamtzahl der Ausgangszustände b'_k bezogen auf eine der beiden möglichen Wurzelfarben, vergleiche Tabelle 1.3	199
3.8	Anzahl der Hüpfsequenzen \mathbb{h}_n im Hubbard- und im Falicov-Kimball-Modell	211
3.9	Graphische Veranschaulichung der möglichen Doppelbesetzungsverläufe bis einschließlich Störungsordnung $n = 5$	213
3.10	Gesamtzahl d_n der möglichen Doppelbesetzungsverläufe aller Klassen bis einschließlich Störungsordnung $n = 15$	213
3.11	Beispielhafte Auswertung der den Hüpfsequenzen der Störungsordnung $n = 3$ zugrundeliegenden Permutationsbäume	246
4.1	Weitere Angaben zu den Reihenkoeffizienten des Hubbardmodells	257
4.2	Grundzustandsenergie und Doppelbesetzungszahl: Vergleich der Störungsrechnung mit DMFT-Ergebnissen	259
4.3	Regressionsanalyse und Extrapolation der Störungsrechnungsergebnisse für $E^{(n)}(U)$ und $D^{(n+1)}(U)$ bei $U = 4,8$ bis $n = 15$ und für $n \rightarrow \infty$	261
4.4	Hilfsdaten für die Berechnung der kritischen Parameter	263
4.5	Kritische Wechselwirkung und den kritischer Exponent: Ergebnisse für $s = n + 1$	264
4.6	Kritische Wechselwirkung und den kritischer Exponent: Ergebnisse für $s = n - 1$	265
4.7	Kritische Energie: Ergebnisse für $s = n + 1$ und $s = n - 1$	268
4.8	Kritische Wechselwirkung, kritischer Exponent, kritische Energie: Ergebnisse für weitere Extrapolationsschemata	269

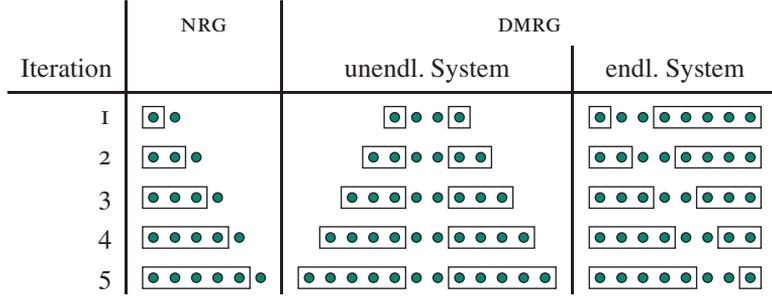
4.9	Kritische Wechselwirkung: Ergebnisse der verschiedenen Extrapolations- schemata im Grenzfall unendlicher Störungsrechnungsordnung	270
5.1	Daten zur Wachstumsanalyse der Anzahl verschiedener Objekte des graphentheoretischen Verfahrens	273
5.2	Daten zur Wachstumsanalyse der Hüpfsequenzanzahl im Vergleich	276
5.3	Datensammlung zur Komplexität des Störungsrechnungsproblems	277
5.4	Primfaktorzerlegter Hauptnenner z_n bis einschließlich Störungsordnung $n = 29$	285
5.5	Mangoldtzahlen M_i bis einschließlich $i = 17$	288
5.6	Näherung für den Hauptnenner \tilde{z}_n bis einschließlich Störungsordnung $n = 21$	289
5.7	Grobgranuläre Parallisierung: Für die Generierung sämtlicher Hüpf- sequenzen des Butcherbaumes $\mathcal{B}_{8,95,26215}$ benötigte Zeit	291
5.8	Feingranuläre Parallisierung: Für die Generierung sämtlicher Hüpf- sequenzen des Butcherbaumes $\mathcal{B}_{8,95,26215}$ benötigte Zeit	294
5.9	Auftragsanzahl und Laufzeitkomplexität für unterschiedliche Granularisierungsstufen der Sequenzierung	297
6.1	Berechnete Gesamtzahl d'_n der möglichen Doppelbesetzungsverläufe aller Hüpfsequenzklassen bis einschließlich Störungsordnung $n = 19$	316
6.2	Leistungszuwachsfaktor bei der Ausgangszustandssequenzierung in Abhängigkeit von der Prozeßanzahl	320
Anhang zu Teil II		
1.1	Zeitaufwand für die Erzeugung der möglichen Doppelbesetzungsverläufe der Graphenordnung $k = 5, 6, \dots, 11$	329
1.2	Doppelbesetzungsverlaufstypen λ und -typbeiträge \mathcal{W} bis einschließlich Störungsordnung $n = 5$	330
1.3	Anzahl der Hüpfsequenzen nach Doppelbesetzungsverlaufstyp, Modell und Vorzeichen bis einschließlich Störungsordnung $n = 5$	333
2.1	Zeitaufwand für die <i>brute-force</i> -Berechnung von Mangoldtzahlen	343

Numerische Verfahren zur Entfaltung von DDMRG-Spektren

Zum Verständnis der Eigenschaften quantenphysikalischer Vielteilchensysteme jenseits des exakt Lösbaren, insbesondere solcher mit Gitterstruktur und Wechselwirkung, hat sich – etwa neben den variantenreichen numerischen Methoden der Dichtefunktionaltheorie (DFT) oder des Quanten-Monte-Carlo-Algorithmus' (QMC) – die *Dichtematrix-Renormierungsgruppe*, kurz DMRG, als ein überaus vielseitig anwendbares und ungemein leistungsfähiges Rechenverfahren für niedrigdimensionale Probleme erwiesen [1, 2]. Damit vermag die Festkörpertheorie verbessert dem Experiment zuzuarbeiten, das vielfältige spektroskopische Untersuchungsergebnisse, beispielsweise von quasi-eindimensionalen Materialien, liefert. Wesentlich von STEVEN WHITE an der *University of California* in Irvine entwickelt und 1992 vorgestellt [3] und seither methodisch verallgemeinert und technisch stark verfeinert, beruht das Ansehen der DMRG vor allem auf der Kontrollierbarkeit der Näherung durch eine strikt systematische Verkleinerung des vom System aufgespannten Hilbertraumes [4].

Moderne DMRG-Algorithmen werden umfassend beispielsweise in Referenz [5] betrachtet; dem Ansatz nach wird die Wellenfunktion des Gesamtsystems (endlicher Größe) durch sukzessive Hinzunahme weiterer Gitterplätze zu einem exakt behandelbaren Teilsystem, „Block“ genannt, aus dessen Umgebung konstruiert, beginnend mit der Blockgröße eins. Dies geschieht durch die Projektion des Block-Gitterplatz-Hamiltonoperators auf seine wesentlichen Zustände und anschließende Zusammenfassung in einem einzelnen neuen Block-Hamiltonoperator, wodurch die Teilsystemgröße zunimmt und die Umgebung kleiner wird, siehe nachfolgende Abbildung. Effektiv werden die relevanten Zustände somit nicht aus den Eigenzuständen des Hamiltonoperators, sondern aus jenen der *reduzierten Dichtematrix* gewonnen. Im Gegensatz zur älteren Numerischen Renormierungsgruppe (NRG), deren Anwendungsbeschränkungen die DMRG-Entwicklung motiviert haben, besteht der effektive Hilbertraum aber nicht aus den Eigenzuständen niedrigster Energie, sondern aus jenen größter Wahrscheinlichkeit. Dieses Vorgehen ermöglicht die Untersuchung von bis zu einigen tausend Teilchen respektive Gitterplätzen mit der Qualität Exakter Diagonalisierung (ED), welche ihrerseits indes aufwandsbedingt auf einige zehn Teilchen beschränkt ist.

Besondere Eigenschaft der DMRG ist zudem, im Raum der Matrixproduktzustände (*matrix-product states*, MPS) eine variationale Methode zu sein [4, 6]. Leitender Gedanke



Schematische Darstellung der Renormierungsgruppenmethoden. Das Symbol \bullet kennzeichnet einen (modellspezifisch mit Teilchen besetzten) Gitterplatz; die Rahmen stehen für die im jeweiligen Algorithmus auftretenden Blöcke. Für die DMRG-Rechnung am endlichen System mit acht Plätzen entsprechen die Iterationen 6 bis 9 der umgekehrten Abfolge der Iterationen 1 bis 4. Details finden sich beispielsweise in Referenz [4].

bei diesen Ansatz ist, jeden Zustand $|\psi\rangle$ eines Quantensystems aus N Gitterplätzen mit dem Hamiltonoperator \hat{H} in dessen Hilbertraum \mathcal{H} aus den Vektoren $|\vec{s}\rangle$ einer vollständigen Basis \mathcal{B} des Systems mit entsprechenden Koeffizienten $p(\vec{s})$ zu entwickeln,

$$\forall |\psi\rangle \in \mathcal{H} : |\psi\rangle = \sum_{|\vec{s}\rangle \in \mathcal{B}} p(\vec{s}) |\vec{s}\rangle .$$

Dabei ist diese Basis wiederum Tensorprodukt

$$\mathcal{B} = \{ |\vec{s} = (s_1, s_2, \dots, s_N)\rangle = \bigotimes_{n=1}^N |s_n\rangle \mid s_n = 1, 2, \dots, d_n \}$$

der Gitterplatzeinzelbasen $|s_n\rangle$ mit der Konfigurationsraumdimension d_n .¹ Die besagten $p(\vec{s})$ lassen sich nun als Produkt aus N Matrices ansetzen:

$$\forall |\vec{s}\rangle \in \mathcal{B} : p_j(\vec{s}) = \prod_{i=1}^j A_i(s_i) \cdot C_j \cdot \prod_{i=j+1}^N B_i(s_i) .$$

Damit wird die Energie des Systems, $E = \langle \psi | \hat{H} | \psi \rangle / \langle \psi | \psi \rangle$, zu einer Funktion von $A_i(s_i)$, $B_i(s_i)$ und C_j , und diese ist unter Orthonormierungsbedingung zu minimieren.

¹Die Basis des n ten von N Gitterplätzen im später, insbesondere im zweiten Teil dieser Arbeit algebraisch-graphentheoretisch behandelten Hubbardmodell ist etwa durch

$$\mathcal{B}(n) = \{ |s_n\rangle \mid s_n = \square, \blacksquare, \blacklozenge, \blacktriangleright \}$$

gegeben, worin die vier Farben eindeutig den folgenden $d_n = 4$ Zuständen zugeordnet sind: $\square \mapsto \emptyset$ (Gitterplatz unbesetzt), $\blacksquare \mapsto \uparrow$ -Spin-Elektron, $\blacklozenge \mapsto \downarrow$ -Spin-Elektron und $\blacktriangleright \mapsto \uparrow \wedge \downarrow = \uparrow\downarrow$ (Platz doppelt besetzt). Mit $\mathcal{H} = \mathcal{B}(1) \otimes \mathcal{B}(2) \otimes \dots \otimes \mathcal{B}(N)$ ergibt sich $\dim(\mathcal{H}) = 4^N$.

Während die DMRG zunächst die „im Rahmen der Numerik exakte“ Berechnung von Grundzustandseigenschaften zum Ziel und darin vielbeachtete Erfolge hatte [7], ermöglicht eine ihrer bedeutsamsten Weiterentwicklungen, die *Dynamische Dichtematrix-Renormierungsgruppe*, DDMRG, wie auch die nahe verwandte Korrekturvektor-DMRG [8, 9] auf Grundlage des Variationprinzips die Berechnung dynamischer Korrelationsfunktionen und damit verbundener angeregter Zustände [5, 10–13] stark korrelierter niedrigdimensionaler Quantensysteme. Zu den besonderen Vorzügen dieses Verfahrens gehört physikalischerseits beispielsweise die Möglichkeit, Eigenschaften mit kleinem Spektralgewicht zu erfassen²; technischerseits ist im Gegensatz zu anderen DMRG-Varianten gleicher Zweckbestimmung [16–18] die Einfachheit des Algorithmus’ sowie dessen leichte Parallelisierbarkeit von Vorteil. Die in der Rechnung nötige Verschiebung der Korrelationsfunktionspole in die komplexe Ebene erfordert jedoch stets die *Faltung* des Spektrums der Observablen \hat{X} eines Quantensystems der Größe N , hier bezeichnet mit \mathfrak{S} in Abhängigkeit von der Frequenz ν , mit einer Lorentzverteilung endlicher Breite η ,

$$\mathfrak{S}_{N,\eta}(\nu) = \sum_m |\langle \psi_{N,m} | \hat{X} | \psi_{N,0} \rangle|^2 \frac{1}{\pi} \frac{\eta}{(\nu - E_{N,m})^2 + \eta^2},$$

worin $|\psi_{N,0}\rangle$ der Grundzustand, $|\psi_{N,m}\rangle$ die Anregungszustände und $E_{N,m}$ die zugehörigen Energien sind; Details sind umfassend in ERIC JECKELMANNs Erstbeschreibung [11] von 2002 dargelegt. Zwar existieren spezielle Ansätze zur Umgehung des Faltungsproblems [8, 19], aber im Allgemeinen ist für eine bessere Ergebnisanalyse durch eine substantielle Erhöhung der Aussagekraft der Spektren ein DDMRG-Rechnungen nachgeschaltetes *Entfaltungsverfahren* notwendig.

Im vorliegenden Teil der Arbeit wird nun eine solche Methode zur Rekonstruktion der ungefalteten *Spektralfunktionen* im *Thermodynamischen Limes*,

$$S(\nu) = \lim_{\eta \rightarrow 0} \lim_{N \rightarrow \infty} \mathfrak{S}_{N,\eta}(\nu),$$

beschrieben, welche im Vergleich zu vorausgegangenen Entwicklungen [5, 17, 20–25] in vielen Fällen qualitativ bessere Resultate durch den Erhalt physikalisch entscheidender Funktionsmerkmale erzielt und universeller nutzbar ist. Dabei wird der fundamentalen Schwierigkeit Rechnung getragen, daß Entfaltung ein schlecht-konditioniertes inverses Problem darstellt [26, 27], dessen direkte Lösung typischerweise zu mit starkem „Rauschen“ behafteten Spektren führt: Da sich der eigentliche Faltungskern aus der Lorentzfunktion und den Beiträgen Numerik-immanenter endlicher Systemgröße³ zusammensetzt und nicht exakt formulierbar ist und zudem numerische Effekte einfließen, wird erstmalig von einer sogenannten *blinden Entfaltung* [28] ausgegangen, die im folgenden

²Also solche, die zu feinen Strukturen in der Spektralfunktion führen, etwa die für das Verständnis von Luttingerflüssigkeiten [14, 15] bedeutsamen *power-law pseudogaps* mit $D(\omega) \sim |\omega|^\alpha$

³Das Entfaltungsproblem ist somit auch hochgradig unterbestimmt, da eine kontinuierliche Funktion aus endlich vielen Datenpunkten rückzugewinnen ist.

(Kapitel 1 und 2) systematisch aufgebaut wird. Umfassend studiert wird die Anwendbarkeit dieses Verfahrens an Lorentz-gefalteten und teilweise künstlich verrauschten analytischen Funktionen (Kapitel 3), die typenhaft für die später (Kapitel 4) untersuchten DMRG-Spektren stehen. Welche offenbleibenden Fragen und Verbesserungen im Rahmen künftiger Arbeiten zu behandeln sind, ist Gegenstand der abschließenden Diskussion (Kapitel 5).

I. Einleitung, Problemstellung und Lösung

Unser Interesse gilt der reellwertigen Spektralfunktion S der Frequenz $\omega \in \mathbb{R}$ eines Vielteilchensystems im Thermodynamischen Limes, also effektiv unendlicher, makroskopischer Größe. Numerische Rechenverfahren liefern uns jedoch selbst unter periodischen Randbedingungen immer Spektren endlicher Systeme – so auch bei Verwendung der DDMRG. Bestenfalls wäre das Ergebnis einer solchen Rechnung nur durch den verfahrensintrinsischen Cauchy-Lorentz-Kern

$$k_\eta(\zeta) = \frac{1}{\pi} \frac{\eta}{\zeta^2 + \eta^2} \quad (\text{I.1})$$

mit der Breite η , welche strenggenommen frequenzabhängig ist, *gefaltet* [11],

$$\mathfrak{S}_\eta(v) = \int_{-\infty}^{+\infty} d\omega S(\omega) k_\eta(v - \omega); \quad (\text{I.2})$$

tatsächlich jedoch haftet dem *Faltungskern* auch die besagte endliche Gitterplatzanzahl, die Diskretisierung des endlichen Frequenzintervalles und die endliche Rechengenauigkeit an.

Im Allgemeinen ist die lineare Antwort eines Quantensystems auf die Messung einer Observablen mit dem Operator \hat{X} durch die dynamische Korrelationsfunktion

$$G_{\hat{X}}(\omega + i\eta) = -\frac{1}{\pi} \left\langle \psi_0 \left| \hat{X}^* \frac{1}{E_0 + \omega + i\eta - \hat{H}} \hat{X} \right| \psi_0 \right\rangle \quad (\text{I.3})$$

gegeben, worin, neben dem hermitesch konjugierten Operator \hat{X}^* der Meßgröße, \hat{H} der zeitunabhängige Hamiltonoperator sowie E_0 und ψ_0 Grundzustandsenergie und Grundzustandswellenfunktion des Systems sind. Von tragender Bedeutung für die gesamte folgende Abhandlung ist dabei wiederum jener reelle Parameter $\eta > 0$ des Faltungskernes (I.1), welcher die Verschiebung der Korrelationsfunktionspole in die komplexe Ebene bewirkt, denn unsere Absicht ist die Berechnung des Imaginärteils von Gleichung (I.3) für $\eta \rightarrow 0$:

$$\begin{aligned} S_{\hat{X}}(\omega + i\eta) &= \text{Im} (G_{\hat{X}}(\omega + i\eta)) \\ &= \frac{1}{\pi} \left\langle \psi_0 \left| \hat{X}^* \frac{\eta}{(E_0 + \omega - \hat{H})^2 + \eta^2} \hat{X} \right| \psi_0 \right\rangle. \end{aligned} \quad (\text{I.4})$$

Um die Faltungsproblematik bei der Berechnung einer dynamischen Korrelationsfunktion besser hervorzuheben, betrachten wir die Aufgabe in einer variationalen Formulierung [12]: Für jedes $\eta \neq 0$ und eine feste Frequenz ω hat das Funktional mit dem unbekanntem Zustand $|\psi\rangle$

$$W_{\hat{X},\eta}(\omega, \psi) = \langle \psi | (E_0 + \omega - \hat{H})^2 + \eta^2 | \psi \rangle + \eta \langle \psi | \hat{X} \rangle + \eta \langle \hat{X} | \psi \rangle$$

ein wohldefiniertes und nicht-entartetes Minimum $|\psi_{\min}\rangle$. Dieses wiederum bestimmt durch

$$-\frac{1}{\pi\eta} W_{\hat{X},\eta}(\omega, \psi_{\min}) = S_{\hat{X}}(\omega + i\eta)$$

die gesuchte Spektralfunktion als Imaginärteil der dynamischen Korrelationsfunktion, siehe Gleichung (1.4). Damit ist die Berechnung von S auf ein Minimierungsproblem zurückgeführt, welches mit DDMRG gelöst werden kann – obschon unter Inkaufnahme eines endlichen Wertes von η .

1.1. Faltung. Der Faltungsbegriff gehört in das Gebiet der elementaren Analysis; eine allgemeine Einführung dieser Operation findet sich in entsprechenden Lehrwerken [29]. Vorbereitend auf die Umkehroperation – Gegenstand dieser Abhandlung – wird im Folgenden nur die Definition und ein kurzes, im Weiteren vertieftes Anwendungsbeispiel gegeben.

1.1.1. Definition. Stehe $f * g$ für die *Faltung* der Funktion f durch die Funktion g (Faltungskern), welche auf einem reellen Intervall B definiert seien. Im Allgemeinen gilt dann $f, g: B \subseteq \mathbb{R} \rightarrow \mathbb{C}$. Die Faltung selbst ist durch das Integral

$$(f * g)(y) = \int dx f(x) g(y - x) \quad (1.5)$$

definiert, dessen Integrationsbereich der gemeinsame Definitionsbereich beider Funktionen ist.

Sei einschränkend $S: B_0 \rightarrow \mathbb{R}$ eine *Spektralfunktion*, $g: B \rightarrow \mathbb{R}$ ein Faltungskern, und gelte $B_0 \subseteq B \subseteq \mathbb{R}$. Dann ist die Faltung als

$$\begin{aligned} (S * g)(y) &= \int_{B_0} dx S(x) g(y - x) \quad \text{so daß } y - x \in B \\ &= \int_B dx S(x) g(y - x) h(x) \quad \text{mit} \\ h(x) &= \begin{cases} 1 & \text{falls } x \in B_0 \text{ bzw. } S(x) \in \mathbb{R} \\ 0 & \text{sonst, also } x \notin B_0 \end{cases} \end{aligned} \quad (1.6)$$

zu definieren, da B_0 alle Elemente aus \mathbb{R} enthält, für welche $S(x \in B_0) \in \mathbb{R}$ gilt. Diese Erweiterung des Integrationsintervalls beziehungsweise des Definitionsbereichs von S auf B wird als (*aperiodische*) *Fortsetzung mit Null* bezeichnet.

1.1.2. *Beispiel.* Die Zustandsdichte $D(\omega)$ (density of states, DOS) eines eindimensionalen Systems wechselwirkungsfreier Teilchen [30, 31] läßt sich im Rahmen des *tight-binding*-Modells⁴ analytisch angeben:

$$D(\omega) = \frac{1}{\pi} \frac{1}{\sqrt{4 - \omega^2}} \quad \text{mit } |\omega| < 2. \quad (1.7)$$

Wie eingangs dargelegt ist das numerische Ergebnis einer entsprechenden DDMRG-Rechnung zumindest durch die Lorentzfunktion (1.1) mit dem bekannten Parameter η gefaltet. Man erhält also analog zu Gleichung (1.2), aber unter Berücksichtigung der vorausgegangenen Betrachtung über den Definitionsbereich, Gleichung (1.6),

$$\mathfrak{D}_\eta(v) = \int_{\mathbb{R}} d\omega D(\omega) k_\eta(v - \omega) \Omega(\omega)$$

mit der Funktion

$$\Omega(\omega) = \begin{cases} 1 & \text{falls } |\omega| \leq 2 \\ 0 & \text{sonst} \end{cases}$$

zur „Ausblendung“ eines negativen Radikanden in der Zustandsdichte. Folglich vereinfacht sich das Faltungsintegral zu

$$\mathfrak{D}_\eta(v) = \frac{1}{\pi^2} \int_{-2}^{+2} d\omega \frac{1}{\sqrt{4 - \omega^2}} \frac{\eta}{(v - \omega)^2 + \eta^2}. \quad (1.8)$$

Abbildung 1.1 zeigt die Graphen der Funktionen (1.7) und (1.8), letztere mit drei unterschiedlichen Werten des verbreiternden Parameters η .

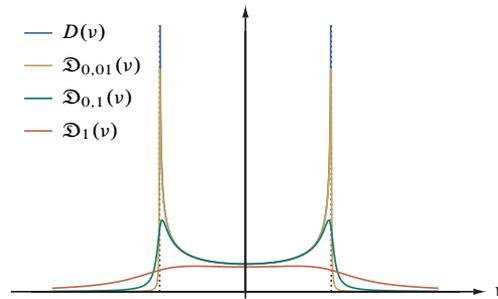


Abbildung 1.1. Zustandsdichte D eines unendlich großen eindimensionalen Systems wechselwirkungsfreier Teilchen und deren Faltung \mathfrak{D} .

⁴Diesem Modell, daß Grenzfall des eindimensionalen Hubbardmodelles, siehe Abschnitt 4.1.1 und insbesondere 4.2.1, bei fehlender Elektronkorrelation ist, kommt im Rahmen der analytischen Prüffunktionen in Kapitel 3 besondere Bedeutung zu.

1.2. Entfaltung. Die *Entfaltung*, insbesondere die „blinde“, also jene mit einem nicht oder nur unvollständig bekannten Faltungskern, gehört zu den schwierigen *inversen Problemen*, welche in vielen natur- und technikwissenschaftlichen Gebieten auftreten und eine große Herausforderung für das numerische Rechnen darstellen. Messungen oder Simulationen liefern nämlich oft nur Näherungswerte der gemessenen oder simulierten Größen; die Abweichungen vom exakten Ergebnis durch die endliche Genauigkeit und Auflösung des Meß- oder Simulationsverfahrens können aber häufig als Faltung mit einer Glättungsfunktion unter Einbeziehung von Rauschen behandelt werden. Ein typisches Beispiel für blinde Entfaltung stellt in diesem Sinne die Rekonstruktion eines reinen Signals aus einem verfälschten Abbild dar, wobei nur unzulängliche Informationen über den Verfälschungsprozeß vorliegen. Eng damit verwandt ist, wie in diesem Abschnitt ausführlich behandelt wird, das Entfaltungsproblem realer DDMRG-Spektren.

Eine grundlegende Behandlung der Entfaltung findet sich in Übersichten zur Angewandten und Numerischen Mathematik [32, 33]; angesichts der hohen Bedeutung der vielfältigen Ansätze gibt es darüber hinaus eine Fülle tiefergehender Betrachtungen in der Spezialliteratur für numerische Rechenmethoden [26, 27] und beispielsweise im Bereich Signal- und Bildverarbeitung [28]

1.2.1. Entfaltung „idealer“ DDMRG-Spektren. Für die Umkehroperation der Faltung im Allgemeinen ist eine Fülle an Methoden entwickelt worden. Beispielsweise [23, 24] kann man sich für den anfangs erwähnten hypothetischen Idealfall der Lorentzkern-Faltung zunutze machen, daß zwischen $\mathfrak{S}_\eta(v)$ und $S(\omega)$ nach Übergang in die Zeitdomäne folgender Zusammenhang besteht:

$$\mathfrak{s}_\eta(t) = e^{-\eta|t|} s(t) .$$

Die Rekonstruktion des ursprünglichen Spektrums $S(\omega)$ aus den DDMRG-Daten, also deren Entfaltung, bedarf bei bekanntem η folglich nur der Ausführung zweier Fouriertransformationen, (1.9-1) und (1.9-3), sowie zwischengeschobener Anwendung des Faltungssatzes (1.9-2) [26, 32]:

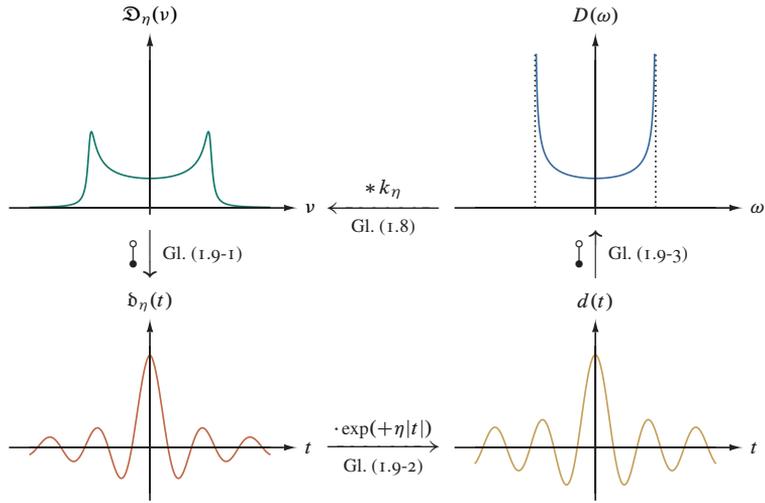
$$\mathfrak{s}_\eta(t) = \mathcal{F}^{-1}\{\mathfrak{S}_\eta(v)\} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} dv e^{+ivt} \mathfrak{S}_\eta(v) , \quad (1.9-1)$$

$$s(t) = e^{+\eta|t|} \mathfrak{s}_\eta(t) , \quad (1.9-2)$$

$$S(\omega) = \mathcal{F}\{s(t)\} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} dt e^{-i\omega t} s(t) . \quad (1.9-3)$$

Eine Veranschaulichung findet sich in Abbildung 1.2.

1.2.2. Reale Vorbedingungen für die Entfaltung. Praktisch stehen der Anwendung des zuvor dargelegten Verfahrens – wie auch allen anderen bisher gewählten Ansätzen [5, 17, 20–25], welche den Faltungskern als vollständig durch Gleichung (1.1) beschrieben ansehen – drei schwerwiegende Gründe zuwider:

Abbildung 1.2. Berechnung von D durch zwei Fouriertransformationen aus \mathfrak{D} .

- (1) Wir kennen \mathfrak{S}_η nur für einen diskreten Satz an Anregungsenergien, also *endlich viele* (gegebenenfalls äquidistante) Frequenzen ν_k ($k = 1, 2, \dots, M$), müssen daher von kontinuierlichen zu *diskreten* Fouriertransformationen übergehen, welche immanent mit Fehlern behaftet sind. Insbesondere gibt es einerseits einen minimalen Abstand zweier Frequenzen $\Delta\nu = \nu_{k+1} - \nu_k$ beziehungsweise eine maximale Zeitskala

$$T = \frac{2\pi}{\Delta\nu}$$

und andererseits durch den endlichen Frequenzbereich $F = \nu_M - \nu_1$ eine minimale Zeitskala

$$\tau = \frac{2\pi}{F} = \frac{T}{M}.$$

Prinzipiell kann die DDMRG-Rechnung für beliebig feine Frequenzraaster durchgeführt werden, wobei jedoch für $\Delta\nu \lesssim \eta$ kein Informationsgewinn eintritt.

- (2) DDMRG-Daten weisen neben signifikanten systematischen, etwa durch die eingangs erwähnte variationale Natur der Vorgehensweise [11], wie alle Numerik-Verfahren auch *statistische* Ungenauigkeiten $\mathfrak{E}(\nu)$ auf, und wir kennen also statt $\mathfrak{S}_\eta(\nu)$ nur

$$\tilde{\mathfrak{S}}_\eta(\nu) = \mathfrak{S}_\eta(\nu) + \mathfrak{E}(\nu) \quad (1.10)$$

respektive die entsprechende Fouriertransformierte

$$\tilde{\mathfrak{s}}_\eta(t) = \mathfrak{s}_\eta(t) + \xi(t),$$

somit allenfalls die Näherung

$$\tilde{s}(t) = e^{+\eta|t|} \tilde{s}_\eta(t) = e^{+\eta|t|} s_\eta(t) + e^{+\eta|t|} \xi(t) .$$

Der erste Summand weist eine sich „wohlverhaltende“ Transformierte $S(\omega)$ auf, der zweite hingegen ist für $|t| \rightarrow \infty$ „eigenwillig“ und verursacht nach Fouriertransformation starke Oszillationen; das Entfaltungsproblem ist durch (numerisches) Rauschen *schlecht konditioniert*.

- (3) DDMRG-Rechnungen sind nur für *endliche Systemgrößen* möglich; somit steht nicht einmal das Spektrum $\tilde{\mathfrak{S}}_\eta(v)$ aus Gleichung (I.10), sondern, unter expliziter Einbeziehung der Faltungformel (I.2), nur der Ausdruck

$$\tilde{\mathfrak{S}}_{N,\eta}(v) = \left[\int_{\mathbb{R}} d\omega S(\omega) k_{N,\eta}(v, \omega) \right] + \mathcal{E}(v) \quad (\text{I.11})$$

für N Gitterplätze mit dem neuen, nicht mehr vollständig bekannten Faltungskern $k_{N,\eta}(v, \omega)$ zur Verfügung⁵, und auch dadurch ist das Entfaltungsproblem schlecht konditioniert. Können wir gemäß zuvor beschriebenem Verfahren exakt entfalten, so erhalten wir letztlich bloß das *diskrete Spektrum* des der (D)DMRG-Näherung geschuldeten endlichen Systems⁶, die Lehmannsumme

$$\tilde{S}_N(\omega) = \left[\sum_{\alpha=1}^d |c_\alpha|^2 \delta(\omega - \omega_\alpha) \right] + \mathcal{E}_{\mathcal{F}}(\omega) , \quad \mathcal{E}_{\mathcal{F}}(\omega) = \mathcal{F} \{ e^{+\eta|t|} \xi(t) \}$$

mit der Hilbertraum-Dimension d und den Amplituden c_α der Eigenzustände, nicht jedoch das kontinuierliche⁷, allenfalls noch mit Rauschen behaftete Spektrum⁸

$$\tilde{S}(\omega) = \lim_{\eta \rightarrow 0} \lim_{N \rightarrow \infty} \tilde{\mathfrak{S}}_{N,\eta}(\omega)$$

⁵Neben dem „Lorentz-Anteil“ aus Gleichung (I.1) haften entsprechend die Effekte der endlichen Systemgröße dem Faltungskern an. Bekannt ist demzufolge

$$\lim_{N \rightarrow \infty} k_{N,\eta}(v, \omega) = k_\eta(v - \omega) .$$

⁶Genaugenommen handelt es sich dabei nur um den ersten Summanden, da der Rauschanteil ein numerischer Effekt ist, siehe Punkt (2). Das Spektrum

$$\lim_{\eta \rightarrow 0} \mathfrak{S}_{N,\eta}(\omega) = S_N(\omega) = \sum_{\alpha=1}^d |c_\alpha|^2 \delta(\omega - \omega_\alpha) = \sum_m S_{N,m} \delta(\omega - E_{N,m})$$

wird auch als Dirac-Kamm bezeichnet, da es sich um eine endliche Summe aus Dirac-Distributionen handelt. Diese Summe läuft über alle zum Spektrum beitragenden Eigenzustände $|m\rangle$ des Hamiltonoperators, welche ein von Null verschiedenes Spektralgewicht $S_{N,m}$ besitzen; $E_{N,m}$ steht für die zugehörige Anregungsenergie.

⁷Im Falle einfacher, tatsächlich diskreter Spektren liegt damit allerdings bereits die Lösung vor.

⁸Man beachte, daß die beiden Grenzübergänge im Allgemeinen nicht vertauscht werden können.

im Thermodynamischen Limes.⁹

Folglich ist ein Entfaltungsverfahren zu entwickeln, das nicht die in Punkt (3) dargelegte Sonderform von *finite-size*-Effekten rekonstruiert, siehe Abschnitt 3.9, sondern uns im Idealfall unter Erhaltung der „Physik-seitig erlaubten“ Funktionscharakteristika das (abschnittsweise) glatte¹⁰ und positiv-semidefinite Spektrum des unendlichen Quantensystems aus dem gefalteten DDMRG-Spektrum des endlichen Systems liefert. Während dabei die Frequenzauflösung aus Punkt (1) durch die DDMRG-Rechnung respektive eine Rediskretisierung, siehe Abschnitt 2.4, beeinflussbar ist, wird dem in Punkt (2) erwähnten Rauschen technischerseits mit einer Filterung, siehe Abschnitt 2.2.2 und 2.3.2 begegnet.

Der Bedeutung der Aufgabe entsprechend, sind in der Vergangenheit mehrere Methoden beschriebener Zielsetzung ausgearbeitet worden. Dazu gehören die direkte Inversion bei niedriger Auflösung [20], lineare Regularisierungsverfahren [5, 21], Fouriertransformationen mit Tiefpaßfilterung [23, 24], nicht-lineare Regularisierungsverfahren wie die Methode maximaler Entropie [24], abschnittsweise Parametrisierung mit Polynomen [17, 24] und ein Entfaltungsansatz für die Selbstenergie [25]. Allen diesen Herangehensweisen ist aber gemeinsam, das Problem nicht als blinde Entfaltung zu behandeln, sondern den Kern als wohlbekannt vorauszusetzen; die Notwendigkeit von Regularisierungs- und Filtertechniken kann daher als Folge des schlecht-konditionierten und unterbestimmten Gleichungssystems (1.11) mit der Lorentzfunktion (1.1) angesehen werden.

Während die genannten Verfahren bei jeweils speziellen Spektralfunktionstypen, beispielsweise im Rahmen der Untersuchung von eindimensionalen Modellen und Störstellenproblemen, mit einigem Nutzen angewendet werden können¹¹, stand eine allgemeine Lösung bislang aus. Die zentrale Schwierigkeit mit allen Ansätzen besteht nämlich darin, charakteristische Funktionsmerkmale wie Stufen, „Spitzen“ oder Potenzgesetz-Singularitäten (*power-law singularities*) [11, 12, 15] innerhalb eines Kontinuums oder an dessen Kanten entweder durch die Regularisierung zu glätten oder durch Artefakte (scharfe Strukturen, starke Oszillationen oder negatives Spektralgewicht) zu verfälschen.

⁹Zumindest $\tilde{S}(\omega)$ kann grundsätzlich und naheliegenderweise auch durch eine Extrapolationsanalyse [11] aus den Ergebnissen der Systemgröße N und den Breiteparameter η variierender DDMRG-Simulationen gewonnen werden. Zumeist ist sogar eine simultane Extrapolation von $N \rightarrow \infty$ und $\eta \rightarrow 0$ möglich, indem man ηN konstant hält. Während auf diesem Wege vorzugsweise isolierte „Auffälligkeiten“ in der Spektralfunktion untersucht werden können, verbietet sich aber dieser Ansatz im Allgemeinen aufgrund der Gesamtkosten für Rechenzeit und -ressourcen für vollständige Spektren; besonders für kleinere Werte von η wächst der datenverarbeitungstechnische Aufwand sehr stark an.

¹⁰Die Forderung eines glatten Funktionsverlaufes durch Rauschdämpfung steht technischerseits offensichtlich in Konkurrenz zum Erhalt „echter“ Extrema, die Aufschluß über das Quantensystem geben. Diesem Problem wird durch Einführung eines minimalen Abstandes zwischen zwei benachbarten Extremwerten im (teil)entfalteten Spektrum in Abschnitt 2.2.2 nachgegangen.

¹¹Erstaunlicherweise ermöglichen es einige dieser Methoden, das abschnittsweise glatte Spektrum des unendlichen Quantensystems direkt aus dem verbreiterten DDMRG-Ergebnis für das endliche System ohne Skalierungsanalyse zu gewinnen.

Zwar können die Ergebnisse verbessert werden, wenn die Eigenschaften der ungefalteten Spektralfunktion a priori bekannt sind [16, 26]; dieser Fall tritt bei der Untersuchung aktueller Fragestellungen indes selten auf. Dementsprechend wird in der vorliegenden Arbeit Entwurf, Implementierung und umfassender Test einer generell gültigen Methode vorgenommen.

1.3. Zusammenfassung. Das Entfaltungsproblem¹² läßt sich nun wie folgt formulieren: Uns liegen die DDMRG-Ergebnisse $\tilde{\mathfrak{S}}_{N,\eta} = \mathfrak{S}_{N,\eta} + \mathcal{E}$ für M Frequenzen ν_k vor, und wir suchen eine physikalisch sinnvolle und „einigermaßen“¹³ glatte Funktion $S(\omega)$, welche

$$\begin{aligned} \tilde{\mathfrak{S}}_{N,\eta}(\nu_k) &= \left[\int_{\mathbb{R}} d\omega S(\omega) k_{N,\eta}(\nu_k, \omega) \right] + \mathcal{E}(\nu_k) \quad \text{mit } k = 1, 2, \dots, M \\ &\approx \mathfrak{S}_{N,\eta}(\nu_k) = \int_{\mathbb{R}} d\omega S(\omega) k_{N,\eta}(\nu_k, \omega) \quad (1.12) \end{aligned}$$

erfüllt [24]. Der kleinste informationstragende Abstand zweier Frequenzen des Rasters ist dabei $|\nu_k - \nu_{k'}| \geq \eta$; kleinere Abstände durch Vergrößerung von M erhöhen nur das Rauschen \mathcal{E} .

Für das Gleichungssystem (1.12) ist festzuhalten, daß wir es durch die $\tilde{\mathfrak{S}}_{N,\eta}$ anhaftenden schwer abzuschätzenden stochastischen Fehler \mathcal{E} sowie die *finite-size*-Effekte nur mit einer Näherung zutun haben; zudem ist das System (1.12) *unterbestimmt*, da durch die (quasi)kontinuierliche Funktion $S(\omega)$ mehr Variablen unbekannt sind, als diese bestimmende Datenpunkte und somit Gleichungen vorliegen.¹⁴ Infolgedessen kann die Entfaltung nur näherungsweise erfolgen, beispielsweise mit einem *Minimierungsverfahren* für das häufig als „*cost function*“ bezeichnete Funktional

$$R[S(\omega)] = \sum_{k=1}^M \left(\tilde{\mathfrak{S}}_{N,\eta}(\nu_k) - \frac{1}{\pi} \int_{\mathbb{R}} d\omega S(\omega) \frac{\eta}{(\nu_k - \omega)^2 + \eta^2} \right)^2, \quad (1.13)$$

welches mit physikalisch begründeten Forderungen an $S(\omega)$ – glatt und positiv-semidefinit – und unter der Annahme $\eta \neq \eta(\omega)$ zu formulieren wäre. Der dabei erfolgte Übergang

$$k_{N,\eta}(\nu_k, \omega) \longrightarrow k_{\eta}(\nu_k - \omega) = \frac{\eta}{(\nu_k - \omega)^2 + \eta^2} \quad (1.14)$$

ist statthaft und auch erforderlich, wenn $S(\omega)$ den Thermodynamische Limes erfüllen, also das unendlich ausgedehnte System beschreiben soll.

¹²Strenggenommen handelt es sich bei der Ausgangsgleichung (1.11) nicht um eine Faltung; da aber im Thermodynamischen Limes der Faltungskern in eine exakt definierte Cauchy-Lorentz-Funktion übergeht, vergleiche Fußnote 5, scheint der Terminus grundsätzlich gerechtfertigt.

¹³Die Wortwahl ist durch den prinzipiellen Näherungscharakter numerischen Rechnens bedingt; eine tiefergehendere Beschreibung der „Glattheitsforderung“ findet sich in Abschnitt 2.2.2.

¹⁴Erschwerend kommt hinzu, daß die Lorentzfaltung als Glättungsoperation ein extrem schlecht-konditioniertes inverses Problem verursacht: Die Lösung reagiert stark auf kleine Änderungen oder Fehler in den Ausgangsdaten.

Nach Diskretisierung von $S(\omega)$ durch L Frequenzen μ_j zu $S_j = S(\mu_j)$ mit $j = 1, 2, \dots, L$ ist Gleichungssystem (1.13), der „Ansatz kleinster Fehlerquadrate“ (*least-square fit*), also die Summe der Quadrate der Differenzen zwischen den DDMRG-Daten und den Werten des approximierten tatsächlichen Spektrums, wie folgt zu schreiben:

$$R = \sum_{k=1}^M \left(\tilde{\mathfrak{S}}_{N,\eta}(v_k) - \frac{1}{\pi} \sum_{j=1}^L \Delta\mu_j S_j \frac{\eta}{(v_k - \mu_j)^2 + \eta^2} \right)^2$$

mit $\Delta\mu_j = \frac{\mu_{j+1} - \mu_{j-1}}{2}$. (I.15)

Die Funktionswerte $S_j = S(\mu_j)$ für $L = M$ Frequenzen μ_i sind im Prinzip, das heißt ohne die obengenannten physikalischen Forderungen an $S(\omega)$, durch die notwendige Bedingung an den Gradienten

$$\frac{\partial}{\partial S_i} R = \frac{2}{\pi} \sum_{k=1}^M \Delta\mu_i \frac{\eta}{(v_k - \mu_i)^2 + \eta^2} \cdot \left(\frac{1}{\pi} \sum_{j=1}^L \Delta\mu_j S_j \frac{\eta}{(v_k - \mu_j)^2 + \eta^2} - \tilde{\mathfrak{S}}_{N,\eta}(v_k) \right) \stackrel{!}{=} 0 \quad (\text{I.16})$$

gegeben.¹⁵ Bei der numerischen Bearbeitung dieses für $L > M$ unterbestimmten¹⁶ linearen Gleichungssystems ist, wie erwähnt, zu berücksichtigen, daß es erheblich von Details wie Anzahl und Verteilung der v_k und μ_j ($\Delta\mu_j < \eta$) sowie besagten DDMRG wesenseigenen Fehlern auf $\tilde{\mathfrak{S}}_{N,\eta}(v_k)$ abhängt.

2. Algorithmische Umsetzung

2.1. Nebenbedingungen. Abweichend von vorausgegangenen Arbeiten [21, 22] behandeln wir im Interesse einer *stabilen* Lösung aufgrund zuvor dargelegter Probleme nicht

¹⁵Physikalisch gesehen ist im Falle verschwindender Fehler \mathfrak{E} das absolute Minimum von R null, und die entsprechenden „Parameter“ S_j sind eindeutig durch ein lineares System aus M Gleichungen,

$$\tilde{\mathfrak{S}}_{N,\eta}(v_k) - \mathfrak{E}(v_k) = \mathfrak{S}_{N,\eta}(v_k) = \frac{1}{\pi} \sum_{j=1}^L \Delta\mu_j S_j \frac{\eta}{(v_k - \mu_j)^2 + \eta^2}$$

bestimmt. Aus mathematischer Sicht hingegen kann obiges Gleichungssystem ebenso keine oder unendlich viele Lösungen haben; überdies kann jeder Fehler $\mathfrak{E}_k = \mathfrak{E}(v_k)$ grundverschiedene und zumeist unphysikalische Lösungen herbeiführen.

¹⁶In der praktischen Anwendung des im Folgenden ausformulierten Verfahrens sind, wie ab Kapitel 3 umfassend behandelt wird, $M = 10^2 \dots 10^3$ und $L = 10^3 \dots 10^4$ typische Werte. Es gilt somit $\Delta\mu_j < \Delta v_k$ sowie allgemein $\Delta\mu_j \sim 1/L$, $\Delta v_k \sim 1/M$ und $[\mu_1, \mu_L] \subseteq [v_1, v_M]$. Für $L \rightarrow \infty$ und folglich $\mu_1 \rightarrow -\infty$ und $\mu_L \rightarrow +\infty$ geht das Gleichungssystem wieder in das inverse Problem (1.12) mit verschwindenden Fehlern über, wenn $k_\eta(v_k - \omega) \rightarrow k_{N,\eta}(v_k, \omega)$ resubstituiert wird.

das Gleichungssystem direkt, sondern minimieren R unter mehreren nicht-linearen Nebenbedingungen. Da die Entfaltung *iterativ-approximierend* erfolgt, bezeichne im weiteren S sowohl die gefaltete wie auch die (teilweise) entfaltete Spektralfunktion.

2.1.1. *Physikalisch begründet.* Aus unserem eingangs dargelegten Wissen um die „Physik“ hinter der Funktion $S(\omega)$ leiten sich folgenden Bedingungen her:¹⁷

- (1) Das Spektrum sei endlich. Bezugnehmend auf die Betrachtungen zum Definitionsbereich von S in Abschnitt 1.1.1, muß das Randverhalten der Spektralfunktion folgender Forderung genügen:

$$S(\omega < \omega_a) \stackrel{!}{=} S(\omega > \omega_b) \stackrel{!}{=} 0$$

mit $\omega_a = \mu_1$ und $\omega_b = \mu_L$ für L diskrete Frequenzwerte μ_i .

- (2) Das Spektrum sei positiv-semidefinit. Es gelte $S(\omega) \geq 0$, das gleichbedeutend mit $S_i = S(\mu_i) \geq 0$ für $i = 1, 2, \dots, L$ ist.
- (3) Das *Spektrale Gewicht*,

$$Z = \int_{\omega_a}^{\omega_b} d\omega S(\omega) \approx \sum_{i=2}^{L-1} \Delta\mu_i S(\mu_i) \quad \text{mit} \quad \Delta\mu_i = \frac{\mu_{i+1} - \mu_{i-1}}{2} \quad (2.1)$$

sowie $S(\mu_1) \stackrel{!}{\approx} 0$ und $S(\mu_L) \stackrel{!}{\approx} 0$, bleibe im Verlauf der Prozedur konstant.

- (4) $S(\omega)$ sei *glatt*

Während sich die Punkte (1) und (2) sehr einfach in den Verfahrensparametern berücksichtigen lassen, ziehen die (teilweise, wie später behandelt wird, zusammenhängenden) Punkte (3) und (4) einigen rechentechnische Aufwand nach sich, auf welchen die weitere Darstellung detailliert eingeht. Insbesondere muß das Konzept der abschnittsweise glatten Spektralfunktion konkreter formuliert werden, was zunächst im Abschnitt 2.2.2 erfolgt.

2.1.2. *Numerikbedingt.* Das iterative Minimierungsverfahren erfordert ein Konvergenzbeziehungswise Abbruchkriterium; dazu kann ausgehend von Gleichung (1.15) beispielsweise die normierte *cost function* \bar{R} für den Vergleich mit einem Schwellenwert (*threshold*) ϑ dienen:

$$\frac{1}{Z} \sqrt{\frac{R}{L}} =: \bar{R} < \vartheta . \quad (2.2)$$

¹⁷Man beachte, daß im Allgemeinen eine Lösung des in Abschnitt 1.3 aus (1.12) entwickelten Gleichungssystems (1.16) unter den hier aufgeführten Bedingungen nicht einmal einem lokalen Minimum von R entsprechen muß. Tatsächlich führt die Lösung des inversen Problems bei endlichen Fehlern auf

$$R = \sum_{k=1}^M \mathcal{E}(v_k)^2 ,$$

wenn wir $S_j = S(\mu_j)$ als geltend annehmen; sonst lassen sich auch andere Parametersätze für das Gleichungssystem (1.16) finden, die zu einem kleineren Wert von R führen, aber „unphysikalisch“ sind. Darin liegt der wesentliche Unterschied zu den vorausgegangenen Ansätzen, welche das Rauschen sowie die Diskretisierungseffekte als vernachlässigbare Störung angesehen haben.

Ebenso kann eine Beendigung des Verfahrens erfolgen, wenn für einen Funktionswert S_i des aktuellen und einen Wert S'_i des vorausgegangenen Entfaltungszyklus

$$\frac{|S'_i - S_i|}{Z} =: \bar{S}_i < \vartheta' \quad (2.3)$$

mit einem entsprechend gewählten Wert ϑ' gilt.¹⁸

2.2. Erster Ansatz. Die in diesem Abschnitt informell vorgestellte erste Umsetzung stellt eine geradlinige Behandlung des in Abschnitt 1.3 beschriebenen Variationsproblems dar. Ein jeder Iterationsschritt besteht dabei aus zwei Teilen. Im ersten Teilschritt, siehe Abschnitt 2.2.1, wird die *cost function* R sukzessiv für eine jede Variable $S_i \geq 0$ minimiert; die zuvor formulierten Zwangsbedingungen sind dabei unproblematisch, weil R ein Polynom zweiten Grades in seinen Variablen ist. Im zweiten Teilschritt, siehe Abschnitt 2.2.2, kommt ein Rauschunterdrückungsverfahren zur Anwendung, das seinerseits bis zur Eliminierung aller Störungen iteriert wird.¹⁹ Man beachte, daß dabei der Wert von R im Allgemeinen wieder zunimmt.²⁰ Das Gesamtverfahren wird bis zur Konvergenz²¹, die entscheidend von der Qualität der Rohdaten und der Komplexität des Spektrums abhängt, wiederholt.

Falls die DDMRG-Ausgangsdaten im Intervall $[v_1, v_M]$ nicht gleichmäßig verteilt sind oder viele engbenachbarte Extrema aufweisen, können vor der Entfaltung eine Regularisierungsstufe, siehe Abschnitt 2.4 und gegebenenfalls ein Glättungszyklus vorgeschaltet werden (“*pre-processing*”); abschließend (“*post-processing*”) kann S mit einer schmalen Lorentzverteilung $\eta' \ll \eta$ „weichzeichnend“ rückgefaltet werden, um eine bessere Näherung an $S(\omega)$ zu erzielen²², siehe Abschnitt 2.2.3.

2.2.1. Minimierung der cost function. Unter Berücksichtigung der zuvor erläuterten Randbedingungen läßt sich der Entfaltungsalgorithmus in seiner einfachsten Form als

¹⁸Diverse Kontrollrechnungen mit den in den folgenden Abschnitten beschriebenen Implementierungen haben gezeigt, daß diese Kriterien für einen automatisierten Abbruch nicht von der behandelten Spektralfunktion unabhängig sind, also ϑ oder ϑ' empirisch gewählt werden muß, damit keine Beendigung vor der Konvergenz eintritt. Dennoch bleibt der Verfahrensschritt als solcher – einer besseren Wahl von \bar{R} oder \bar{S} vorbehaltlich – in allen folgenden Darstellungen des Algorithmus’ berücksichtigt; in den Anwendungen, beginnend mit Kapitel 3, werden ohnedies zum Studium des Konvergenzverhaltens stets 1000 Entfaltungszyklen, die etwa doppelte bis fünffache Anzahl des typischerweise Notwendigen, durchgeführt. In der hier vorgestellten Fassung des Entfaltungsprogrammes wird abweichend von Referenz [44] orientierungshalber ein normierter Wert von R nach jeder Iteration protokolliert, wobei sich nach einem anfänglichen rapiden monotonen Abfall Sättigung oder ein mehr oder minder ausgeprägt „sägezahnartiges“ Verhalten einstellen.

¹⁹Da beispielsweise die Konstanzhaltung des Spektralen Gewichts Z bei Glättung numerikbedingter Artefakte Verfeinerungen erfordert, ist der sich anschließende Abschnitt 2.3 im Wesentlichen durch die sich jeweils zuvor ergebenden Schwierigkeiten motiviert.

²⁰Wird auf diesen Glättungsschritt verzichtet, erhält man ein gänzlich unbrauchbares Ergebnis, da sich durch die stark unterbestimmte Entfaltung ($L \gg M$) ohne einen Regularisierungsmechanismus die Fehler von Iteration zu Iteration ungedämpft akkumulieren.

²¹Siehe Fußnote 18

²²Sofern diese abschließende Glättung mit ausreichender Breite erfolgt, wird damit das Problem der durch viele lokale Minima nicht eindeutigen Lösung von R umgangen. Mit einigem Vorteil kann dazu auch eine Gaußverteilung mit dem Breiteparameter $\sigma = \eta'$ verwendet werden [17]; eine umfassendere Betrachtung zu dieser Option findet sich in Kapitel 5.

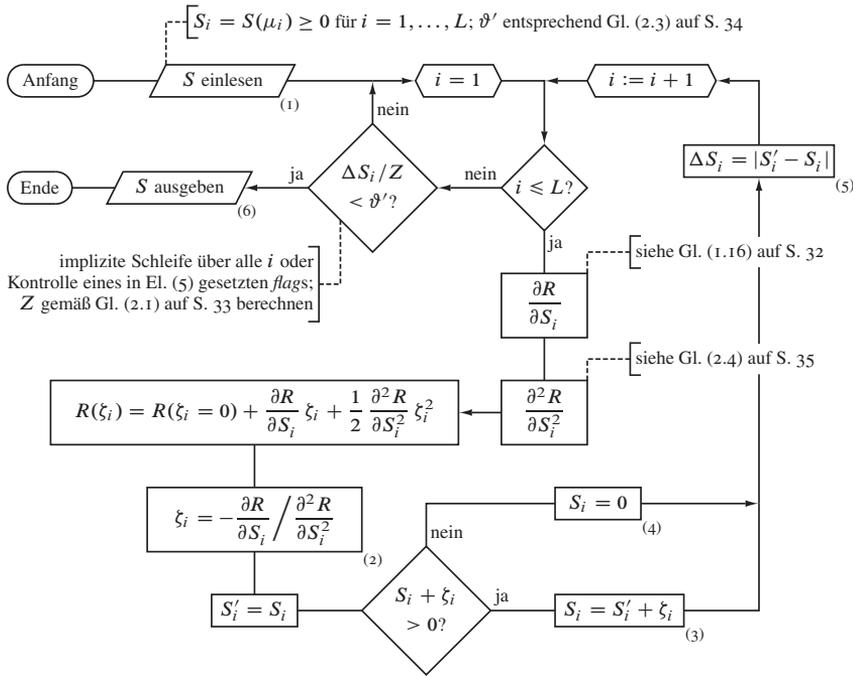


Abbildung 2.1. Programmablaufschemata der selbstkonsistenten Entfaltungsprozedur.

mehrfache eindimensionale Minimierung in orthogonalen Richtungen wie folgt skizzieren:

- 1 Ausgangskonfiguration $S_i = S(\mu_i)$ für $i = 1, 2, \dots, L$ einlesen, siehe Flußdiagrammelement (1) in Abbildung 2.1²³
- 2 R bezüglich S_i für alle $i = 1, 2, \dots, L$ minimieren:
 - 2.1 $R(\zeta_i) = R(\zeta_i = 0) + \frac{\partial R}{\partial S_i} \zeta_i + \frac{1}{2} \frac{\partial^2 R}{\partial S_i^2} \zeta_i^2$ behelfs

$$\frac{\partial^2}{\partial S_i^2} R = \frac{2}{\pi^2} \sum_{k=1}^M \Delta \mu_i^2 \frac{\eta^2}{((v_k - \mu_i)^2 + \eta^2)^2} \quad (2.4)$$

sowie des Gradienten von R , Gleichung (1.16), berechnen

$$2.2 \quad \zeta_i = - \frac{\partial R}{\partial S_i} / \frac{\partial^2 R}{\partial S_i^2} \text{ berechnen, Element (2)}$$

- 2.3 S_i durch $S_i + \zeta_i$ substituieren, wenn $S_i + \zeta_i > 0$, anderenfalls $S_i = 0$ setzen, Elemente (3) und (4)

²³Aufbauprinzip und Sinnbilder aller Flußdiagramme in dieser Arbeit folgen strikt der DIN 66 001 in der Fassung vom September 1966. Soweit nicht durch Pfeilspitzen an den Flußlinien anders gekennzeichnet, sind die Vorzugsrichtungen des Datenflusses von links nach rechts sowie von oben nach unten.

- 3 Verfahren beenden und Endkonfiguration $S(\mu) = S$ ausgeben, Element (6), falls das gewählte Konvergenzkriterium, hier gemäß Gleichung (2.3), erfüllt ist; sonst zu Schritt 2 zurückkehren

Mit $S(\mu)$ liegt nun das vollständig entfaltete, aber mit verstärktem *numerischen Rauschen* behaftete Spektrum vor.

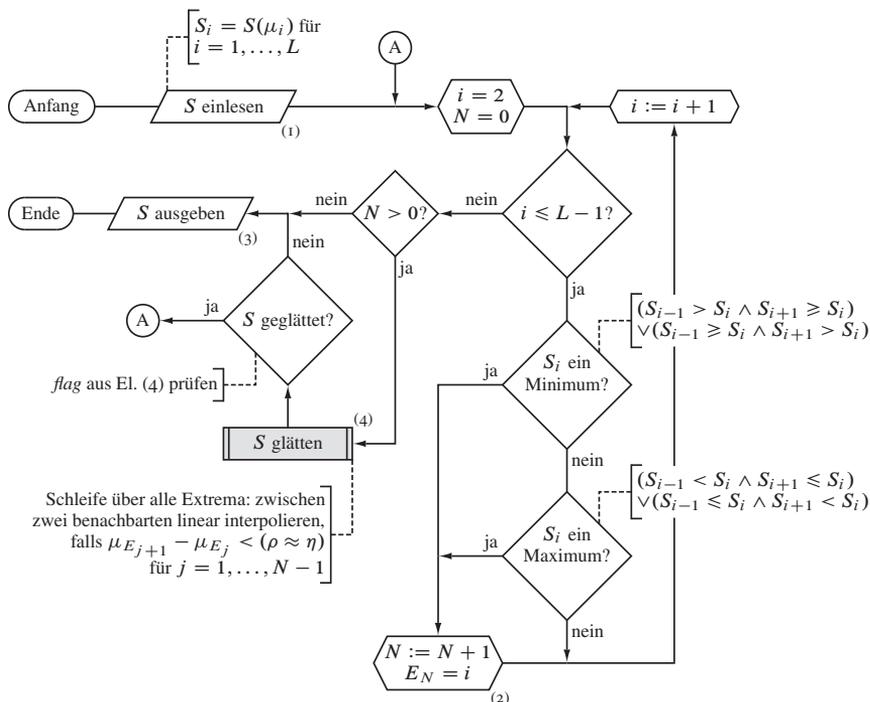


Abbildung 2.2. Programmablaufschemata der Glättungsprozedur: Rahmenprogramm mit Extremumssuche. Das Unterprogramm, Element (4), ist explizit in Abbildung 2.3 dargestellt.

2.2.2. *Glättung des Spektrums.* Dieses Verfahren wird in Erweiterung des Entfaltungsalgorithmus' zur Rauschdämpfung²⁴ unter Erhaltung der Struktur des Spektrums eingangs auf den Rohdatensatz sowie nach jeder Iteration der Minimierungsprozedur (Abschnitt 2.2.1) angewendet; eine eigenständige Verwendung ist jedoch auch möglich. Zur möglichststen Erhaltung von Nutzinformationen ist dabei Schlüsselidee, als ‚Rauschen‘

²⁴Rauschminderung oder -unterdrückung sind, zusammen mit der Entfaltung, ein ubiquitäres Problem der Signalverarbeitung in sämtlichen Natur- und Technikwissenschaften, und die Fachliteratur ist entsprechend umfangreich. Einige Bemerkungen dazu finden sich beispielsweise in den Referenzen [26] und [32].

nur solche Extrema anzusehen, deren Frequenzabstand kleiner als ein vorgegebener Verfahrensparameter $\rho \approx \eta$ ist.

- 1 Datensatz $S_i = S(\mu_i)$ für $i = 1, 2, \dots, L$ und Parameter ρ für Flußdiagrammelement (4) einlesen oder vom aufrufenden Programm übernehmen, siehe Element (1) in Abbildung 2.2
- 2 Alle N Extrema in S_i für $i = 2, 3, \dots, L - 1$ finden und deren Indices in der Liste E speichern, Element (2):
 - 2.1 S_i ist ein Minimum, falls $(S_{i-1} > S_i \wedge S_{i+1} \geq S_i) \vee (S_{i-1} \geq S_i \wedge S_{i+1} > S_i)$.
 - 2.2 S_i ist ein Maximum, falls $(S_{i-1} < S_i \wedge S_{i+1} \leq S_i) \vee (S_{i-1} \leq S_i \wedge S_{i+1} < S_i)$.
- 3 Wenn $N = 0$ ist (oder in Schritt 4 kein Extremum geglättet wurde) Verfahren beenden und $S(\mu) = S$ aus- beziehungsweise an das übergeordnete Programm zurückgeben, Element (3)

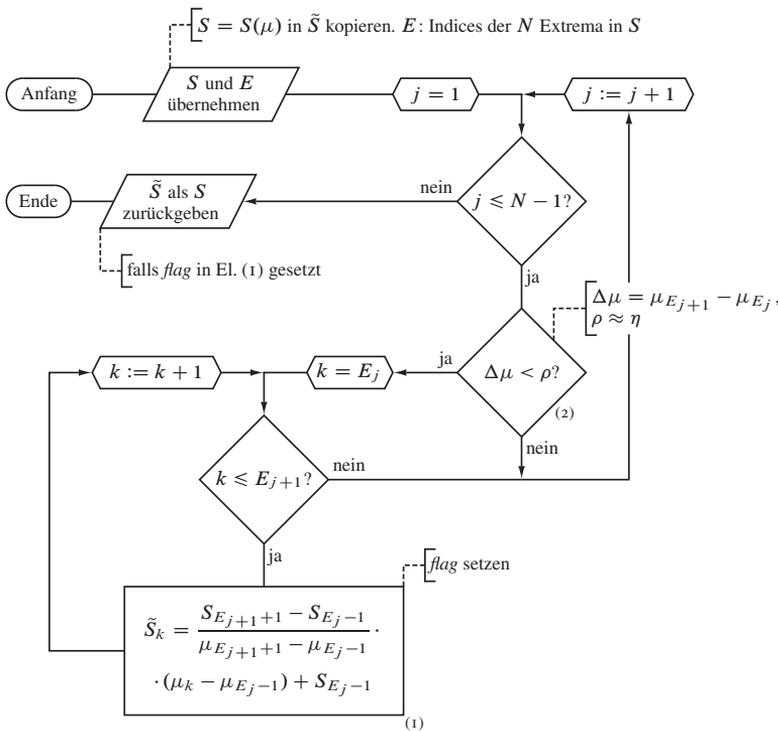


Abbildung 2.3. Programmablaufscha der Glättungsprozedur: Unterprogramm zur linearen Interpolation des Spektrums zwischen benachbarten Extrema, Flußdiagrammelement (4) in Abbildung 2.2.

- 4 Flußdiagrammelement (4) in Abbildung 2.2 sowie Abbildung 2.3: \tilde{S} als Kopie von S anlegen, ρ übernehmen, Schleife über $j = 1, 2, \dots, N - 1$ ausführen: Falls die Bedingung $\mu_{E_{j+1}} - \mu_{E_j} < \rho$ mit μ_{E_j} als Frequenz des j ten Extremums erfüllt ist, Flußdiagrammelement (2), alle Werte \tilde{S}_k zwischen den benachbarten Extrema j und $j + 1$ mit $k = E_j, E_j + 1, \dots, E_{j+1}$ linear interpolieren, Element (1):

$$\tilde{S}_k = S_{E_{j-1}} + \frac{S_{E_{j+1}+1} - S_{E_{j-1}}}{\mu_{E_{j+1}+1} - \mu_{E_{j-1}}} (\mu_k - \mu_{E_{j-1}})$$

- 5 Wird mindestens ein Extremum dem Abstandskriterium gemäß geglättet, S mit \tilde{S} überschreiben und zu Schritt 2 zurückkehren; sonst zu Schritt 3 übergehen

Mit diesem Glättungsansatz wird insbesondere Punkt (4) aus Abschnitt 2.1.1 Rechnung getragen: Grundsätzlich erwarten wir von $S(\omega)$, abschnittsweise zusammenhängend zu sein; dabei sollen die Abstände zwischen „Diskontinuitäten“ größer als die minimale Frequenz- beziehungsweise Energiedifferenz ρ sein. Da uns die Spektralfunktion nur in einer diskretisierten Repräsentation vorliegt, müssen wir die „Kontinuitätsbedingung“ folglich für den endlichen Satz an Frequenzwerten definieren; entsprechend verlangen wir, daß der Abstand zwischen zwei signifikanten, also physikalisch relevanten Extrema größer als der Methodenparameter $\rho > 0$ ist.

Der minimale Extremwertabstand muß vorsichtig für das behandelte Spektrum gewählt werden, nämlich schmaler als der Abstand tatsächlicher Singularitäten, indes nicht zu schmal, um nicht zu viele künstliche Spitzen im Entfaltungsergebnis zuzulassen. In der Praxis hat sich erwiesen, daß $\rho \gtrsim \eta$ zu vernünftigen, abschnittsweise glatten Resultaten führt.

2.2.3. (Rück)faltung. Zwecks Kontrolle der Zwischenergebnisse aus der (iterativen) Entfaltungprozedur mit den Ausgangsdaten von S sowie aus später dargelegten Gründen, welche die Interpretation des Ergebnisses betreffen, wird nachfolgendes Verfahren zur erneuten Faltung benötigt:

- 1 Datensatz $S_i = S(\mu_i)$ für $i = 1, 2, \dots, L$ und Breitenparameter η' einlesen oder vom aufrufenden Programm übernehmen, siehe Flußdiagrammelement (1) in Abbildung 2.4
- 2 Innerhalb der äußeren Schleife, Index i , über alle Frequenzen μ_i und Funktionswerte S_i des ungefalteten Spektrums in einer Schleife folgende „Faltungssumme“, Index j , mit η' ausführen:

$$\mathfrak{S}_i = \frac{1}{\pi} \sum_{j=1}^L \Delta\mu_j S_j \frac{\eta'}{(\mu_i - \mu_j)^2 + \eta'^2} \quad (2.5)$$

- 3 Das gefaltete Spektrum $\mathfrak{S}_{\eta'}(\mu) = \mathfrak{S}$ aus- beziehungsweise an das übergeordnete Programm zurückgeben, Element (2)

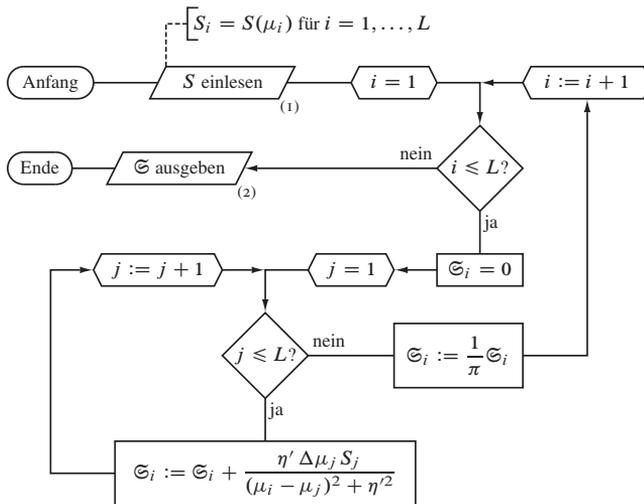


Abbildung 2.4. Programmablaufschaema der Ruckfaltungsprozedur.

2.3. Verfeinerungen. Das in den Anwendungsabschnitten 3 und 4 genutzte gesamte Rechenverfahren unterscheidet sich von dem im vorausgegangenen Abschnitt 2.2 dargelegten durch eine nützliche (Laufzeit des Entfaltungsalgorithmus') und zwei notwendige (Erhaltung von Symmetrieeigenschaften der Spektralfunktionen im Verlauf der Entfaltung, Wirkungsweise des Glättungsalgorithmus') Verbesserungen, wie im Folgenden behandelt wird.

2.3.1. *Minimierung der cost function.* Der in Abschnitt 2.2.1 beschriebene Entwurf der Minimierungsprozedur oder, nun nach deren Wirkungsweise benannt, des *Entfaltungsverfahrens* zeigt in der Anwendung einen bei der späteren Ergebnisanalyse erheblich störenden Effekt, nämlich die Beeinflussung möglicher Symmetrien (siehe beispielsweise Abbildung 1.1) in den Spektralfunktionsdaten. Bereits ohne die nötige tieferegreifende Betrachtung hat sich empirisch gezeigt, daß sich dieses Problem durch eine abwechselnd „links- und rechtseitige“ Minimierung, also zunächst bezüglich der Indexnummer aufsteigendem und danach absteigendem Lauf durch die Datenpunkte beheben läßt.

Überdies wird eine wesentliche technische Schwäche in der überarbeiteten Version behoben: Schritt 2 in der ersten Fassung, eine Schleife über alle L Datenpunkte der (teil)entfalteten Zielfunktion $S(\mu)$, beinhaltet nämlich die Berechnung des Gradienten von R gemäß Gleichung (1.16) in Form zweier verschachtelter Summen über $k = 1, 2, \dots, M$ und $j = 1, 2, \dots, L$. Daraus läßt sich unter dem Zusammenhang $L \sim M$ die Laufzeit zu $\mathcal{O}(L^3)$ abschätzen. Durch Ausgliederung der zweiten Summe im Gradienten in die Berechnung einer Hilfsgröße $Q(v)$ ist jedoch eine Reduktion auf

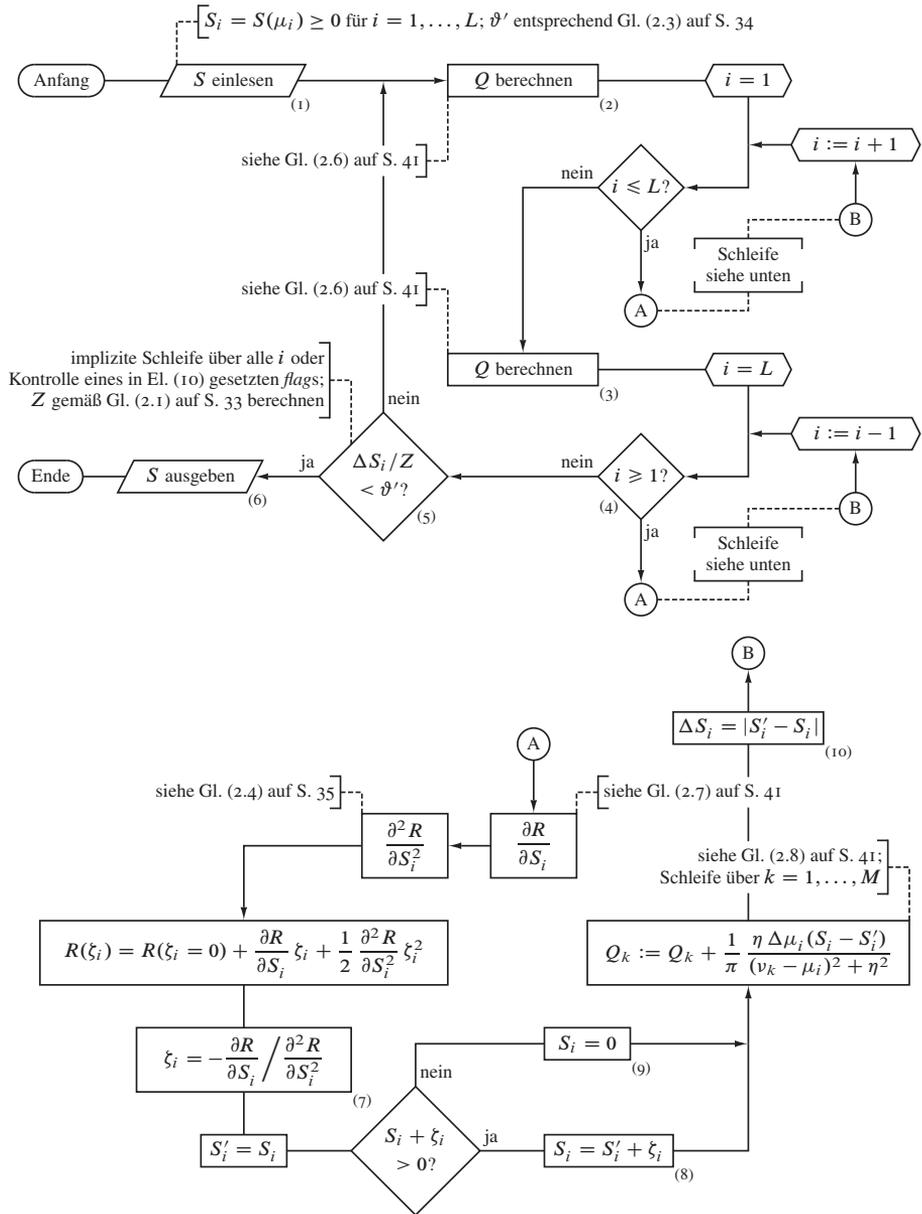


Abbildung 2.5. Programmablaufschemata der laufzeitoptimierten selbstkonsistenten Entfaltungsprozedur mit abwechselnd „links- und rechtseitigem“ (bidirektionalem) Indextdurchlauf.

in L quadratischen Zeitbedarf möglich, und der Algorithmus kann zusammenfassend folgendermaßen geschrieben werden:

- 1 Ausgangskonfiguration $S_i = S(\mu_i)$ für $i = 1, 2, \dots, L$ einlesen, siehe Flußdiagrammelement (1) in Abbildung 2.5, oder vom Rahmenprogramm übernehmen
- 2 $Q_k = Q(v_k)$ für alle $k = 1, 2, \dots, M$ berechnen, Element (2):

$$Q_k = \frac{1}{\pi} \sum_{i=1}^L \Delta\mu_i S_i \frac{\eta}{(v_k - \mu_i)^2 + \eta^2} \quad (2.6)$$

- 3 R bezüglich S_i „von links nach rechts“ für alle $i = 1, 2, \dots, L$ minimieren:

$$3.1 \quad R(\zeta_i) = R(\zeta_i = 0) + \frac{\partial R}{\partial S_i} \zeta_i + \frac{1}{2} \frac{\partial^2 R}{\partial S_i^2} \zeta_i^2 \text{ behelfs}$$

$$\frac{\partial R}{\partial S_i} = \frac{2}{\pi} \sum_{k=1}^M \Delta\mu_i \frac{\eta}{(v_k - \mu_i)^2 + \eta^2} (Q_k - \tilde{\mathcal{E}}_{N,\eta}(v_k)) \quad (2.7)$$

sowie der Divergenz von R , Gleichung (2.4), berechnen

$$3.2 \quad \zeta_i = -\frac{\partial R}{\partial S_i} / \frac{\partial^2 R}{\partial S_i^2} \text{ berechnen, Element (7)}$$

- 3.3 S_i als S' temporär sichern

- 3.4 S_i durch $S_i + \zeta_i$ substituieren, falls $S_i + \zeta_i > 0$, sonst $S_i = 0$ setzen, Elemente (8) und (9)

- 3.5 Q_k für alle $k = 1, 2, \dots, M$ „korrigieren“:

$$Q' = Q_k + \frac{1}{\pi} \frac{\eta}{(v_k - \mu_i)^2 + \eta^2} \Delta\mu_i (S_i - S'); \quad Q_k = Q' \quad (2.8)$$

- 4 $Q_k = Q(v_k)$ für alle $k = 1, 2, \dots, M$ berechnen, Element (3), analog zu Schritt 2
- 5 R bezüglich S_i „von rechts nach links“ für alle $i = L, L-1, \dots, 1$ minimieren, analog zu Schritt 3
- 6 Verfahren beenden und Endkonfiguration $S(\mu) = S$ ausgeben, Element (6), wenn das hier beispielsweise nach Gleichung (2.3) gewählte Konvergenzkriterium erfüllt ist; anderenfalls zu Schritt 2 springen

Sofern dieses Verfahren nicht eigenständig, sondern als Unterprogramm verwendet und das Abbruchkriterium innerhalb einer übergeordneten Selbstkonsistenzschleife geprüft wird, entfällt das Flußdiagrammelement (5) unter direktem Übergang von (4) zu (6), siehe Abschnitt 2.5.

2.3.2. *Glättung des Spektrums.* Die erste Version der Rauschdämpfungsprozedur, Abschnitt 2.2.2, weist durch die einfache lineare Interpolation zwischen zwei Artefaktextrrema eine schwerwiegende Schwäche auf, nämlich eine starke Beeinflussung des in Gleichung (2.1) definierten Spektralgewichtes, dessen Konstanz wir ja verlangen. Um nun den Flächenerhalt in einer ersten Stufe zu verbessern, bedarf es zunächst (beispielsweise) der „Trapezintegration“ der Spektralfunktionsdaten $S(\mu)$ im Intervall der zu glättenden Extrempunkte (Hilfsgröße \tilde{Z}) und anschließender Verschiebung der interpolierten S bis zur näherungsweise Gleichheit der Flächen.

Zudem führen wir zur verbesserten Rauscherkennung neben dem durch $\rho \approx \eta$ vorgegebenen maximalen „horizontalen“ Abstand zweier benachbarter Extremumsfrequenzen,

$$\mu_{E_{j+1}} - \mu_{E_j} < \rho, \quad (2.9)$$

einen zweiten Verfahrensparameter ein: $h \geq 0$ skaliert den minimalen „vertikalen“ Abstand der Spektralfunktionswerte zweier benachbarter Extremumsfrequenzen, und diese werden als signifikant angesehen, wenn für deren relative Höhendifferenz

$$2 \frac{|S_{E_{j+1}} - S_{E_j}|}{(S_{E_{j+1}} + S_{E_j})} > h \quad (2.10)$$

gilt. Der optimale Wert von h muß wie jener von ρ auf Grundlage der betrachteten Funktion durch Probeanwendungen ermittelt werden.²⁵

Der Algorithmus ist modifiziert wie folgt zu schreiben:

- 1 Datensatz $S_i = S(\mu_i)$ für $i = 1, 2, \dots, L$ und Parameter ρ für Flußdiagrammelement (4) einlesen oder vom aufrufenden Programm übernehmen, siehe Element (1) in Abbildung 2.2
- 2 Alle N Extrema in S_i für $i = 2, 3, \dots, L - 1$ finden und deren Indices in der Liste E speichern, Element (2):
 - 2.1 S_i ist ein Minimum, falls $(S_{i-1} > S_i \wedge S_{i+1} \geq S_i) \vee (S_{i-1} \geq S_i \wedge S_{i+1} > S_i)$.
 - 2.2 S_i ist ein Maximum, falls $(S_{i-1} < S_i \wedge S_{i+1} \leq S_i) \vee (S_{i-1} \leq S_i \wedge S_{i+1} < S_i)$.
- 3 Wenn $N = 0$ ist (oder in Schritt 4 kein Extremum geglättet wurde) Verfahren beenden und $S(\mu) = S$ aus- beziehungsweise an das übergeordnete Programm zurückgeben, Element (3)
- 4 Flußdiagrammelement (4) in Abbildung 2.2 sowie Abbildung 2.6: \tilde{S} als Kopie von S anlegen, ρ übernehmen, Schleife über $j = 1, 2, \dots, N - 1$ ausführen: Falls die Bedingungen (2.9) und (2.10) erfüllt sind, im Flußdiagramm Element (2), wie folgt vorgehen:
 - 4.1 \tilde{Z} berechnen:

$$\tilde{Z} = \frac{1}{E_{j+1} - E_j + 3} \left(\frac{1}{2} S_{E_{j-1}} + \sum_{l=E_j}^{E_{j+1}} S_l + \frac{1}{2} S_{E_{j+1}+1} \right)$$

- 4.2 Alle Werte \tilde{S}_k zwischen den benachbarten Extrema j und $j + 1$ auf folgende Weise linear interpolieren, Element (1), $k = E_j, E_j + 1, \dots, E_{j+1}$:

$$\tilde{S}_k = \tilde{Z} + \frac{S_{E_{j+1}+1} - S_{E_{j-1}}}{\mu_{E_{j+1}+1} - \mu_{E_{j-1}}} \left(\mu_k - \frac{\mu_{E_{j+1}+1} + \mu_{E_{j-1}}}{2} \right)$$

²⁵Sämtliche Untersuchungen im Rahmen dieser Arbeit behandeln alle lokalen Extrema der jeweiligen Spektren als relevant; zur Genauigkeit der Fließkomma-Arithmetik passend wurde daher $h \approx 10^{-16}$ gesetzt. Auf die Wahl der Parameter wird nochmals in Abschnitt 3.1.4 eingegangen.

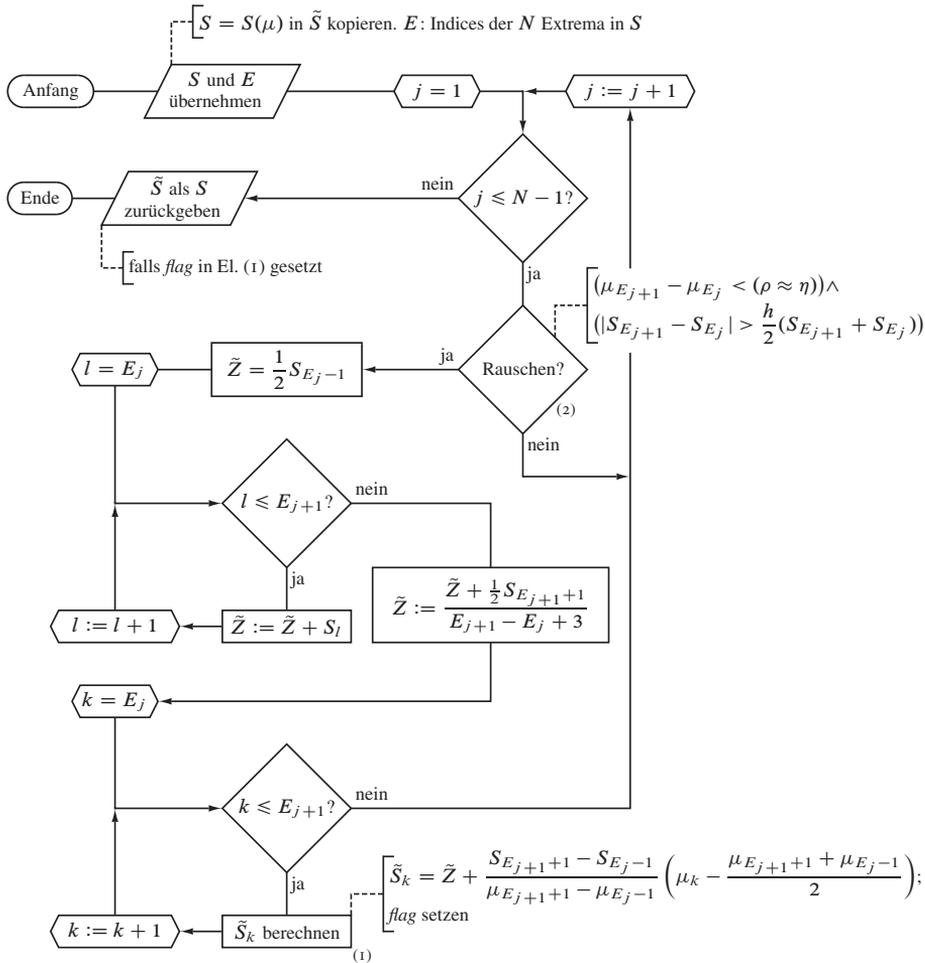


Abbildung 2.6. Programmablaufscha der Glättungsprozedur: Lineare Interpolation des Spektrums zwischen zwei benachbarten Extrema unter Erhaltung des Spektralen Gewichts, Flußdiagrammelement (4) in Abbildung 2.2.

- 5 Wird mindestens ein Extremum gemäß den Abstandskriterien geglättet, S mit \tilde{S} überschreiben und zu Schritt 2 zurückkehren; anderenfalls zu Schritt 3 übergehen

2.4. Rediskretisierung und Regularisierung der DDMRG-Rohdaten. Vereinfachend sind wir bei der Formulierung der numerischen Nebenbedingungen, Abschnitt 2.1.2, und somit der Konstruktion des Verfahrens bislang davon ausgegangen, daß das Ergebnis der zugrundeliegenden DDMRG-Rechnungen zusätzlich folgende Voraussetzungen erfüllt:

(1) Für den Abstand zweier beliebiger benachbarter Punkte des Frequenzrasters gelte

$$\mu_{i+1} - \mu_i \approx \eta' \quad \text{für } i = 1, 2, \dots, L-1,$$

da sonst der Rückfaltungskern in Abschnitt 2.2.3 „ δ -funktionshaften Charakter“ annimmt.

(2) Der Frequenzabstand sei auf dem gesamten Intervall gleich, formal

$$\mu_{i+1} - \mu_i = \frac{\mu_L - \mu_1}{L-1} \quad \text{für } i = 1, 2, \dots, L-1.$$

Da im Besonderen die erste Anforderung bei gängigen Ausgangsdatensätzen aufgrund des stark anwachsenden Aufwands der DDMRG-Rechnung nicht erfüllt ist, wird eine der gesamten Entfaltungsprozedur vorgeschaltete Verarbeitungsstufe notwendig, welche die Frequenzrasterweite vereinheitlicht und verkleinert. Dazu wählen wir einen die Anzahl der Datenpunkte erhöhenden Faktor c derart, daß

$$\eta' \geq \frac{2}{c} \eta$$

erfüllt ist, woraus bei einem typischen Verhältnis von $\eta/\eta' = 10$ eine Vergrößerung um $c \geq 20$ resultiert.

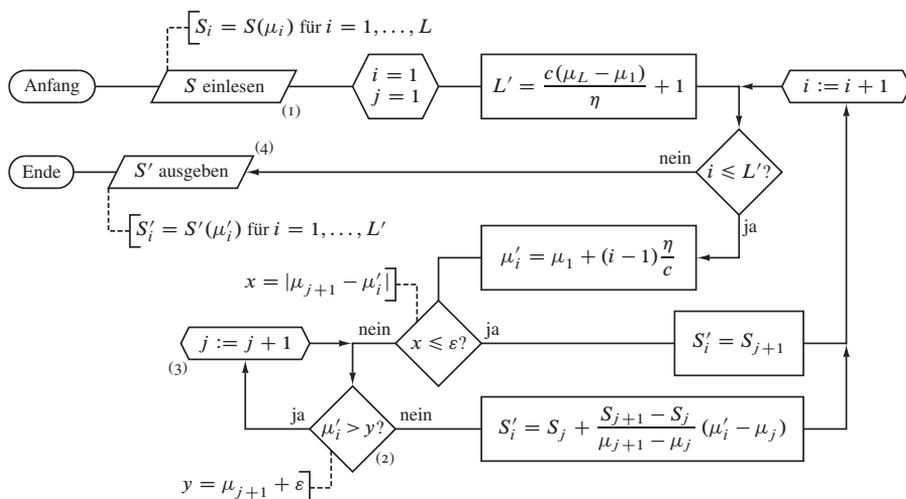


Abbildung 2.7. Programmablaufschemata der Rediskretisierungs- und Regularisierungsprozedur.

Die folgende einfache programmtechnische Umsetzung auf Grundlage einer linearen Interpolation ist derart angelegt, daß auch ein die Zielauflösung übertreffender Datensatz, $(\mu_L - \mu_1)/(L-1) < \eta'/2$, korrekt verarbeitet werden kann:

1 Rohdatensatz $S_i = S(\mu_i)$ für $i = 1, 2, \dots, L$, Faltungsbreiteparameter η und Rasterskalierungswert c einlesen oder vom aufrufenden Programm übernehmen, siehe Flußdiagrammelement (1) in Abbildung 2.7; Schwellenwert ε in der Größenordnung der Maschinengenauigkeit wählen

2 Rohdatenpunktzähler $j = 1$ initialisieren, neue Datenpunkanzahl für $\mu_L > \mu_1$ berechnen:

$$L' = \frac{c(\mu_L - \mu_1)}{\eta} + 1$$

3 Neue Datenpunkte für $i = 1, 2, \dots, L'$ berechnen:

3.1 Frequenz berechnen:

$$\mu'_i = \mu_1 + (i - 1) \frac{\eta}{c}$$

3.2 Zugehörigen Spektralfunktionswert berechnen. Falls $|\mu_{j+1} - \mu'_i| \leq \varepsilon$, dann:

$$S'_i = S_{j+1}$$

Sonst j um eins erhöhen, so lange $\mu'_i > \mu_{j+1} + \varepsilon$, Flußdiagrammelement (2) und (3), dann:

$$S'_i = S_j + \frac{S_{j+1} - S_j}{\mu_{j+1} - \mu_j} (\mu'_i - \mu_j)$$

4 Rediskretisiertes und regularisiertes Spektrum $S'_i = S'(\mu'_i)$ für $i = 1, 2, \dots, L'$ aus- beziehungsweise (die Datenfelder und -variablen S , μ und L ersetzend) an das übergeordnete Programm zurückgeben, Element (4)

Eine beispielhafte Anwendung des zuvor beschriebenen Rediskretisierungs- und Regularisierungsverfahrens zeigt Abbildung 2.8; dieser liegt eine an sich hochaufgelöste, hier jedoch von 4000 (gleichmäßig verteilte, $\Delta\omega = 5 \cdot 10^{-3}$) auf 25 (ungleichmäßig verteilte, $\Delta\omega \geq 5 \cdot 10^{-1}$) Datenpunkte reduzierte DDMRG-Rechnung zugrunde, die ausführlich in Abschnitt 4.3.2 behandelt wird. In Abbildungsteil (a) sind die Rohdaten, das Ergebnis des hier betrachteten Verfahrens und das Resultat einer Interpolation mit dem Computeralgebrasystem *Mathematica* [45] der noch gefalteten und rauschbehafteten Spektralfunktion $\mathfrak{D}(\omega)$ dargestellt. Deutlich ist zu erkennen, daß die hiesige einfache Methode durch Formveränderungen nur einen vergleichsweise geringen Informationsverlust bewirkt, während eine direkte²⁶, unangepaßte Anwendung von *Mathematica* zu einer deutlichen Verfälschung führt, siehe insbesondere im Intervall $3 < \omega < 5$.

²⁶Die *Mathematica*-Funktion *Interpolation* stellt mehrere Ansätze zur Wahl; Vorgabe ist eine Hermiteinterpolation, welche der generierten *InterpolatingFunction* gewisse analytische Eigenschaften verleiht und eine „beliebig genaue“ numerische Auswertung erlaubt. Eine noch ungeprüfte Erhöhung der Interpolationsordnung oder die alternative Verwendung von Polygonzügen (*splines*) [34] führt möglicherweise zu einer bedeutend besseren Anpassung an die Ausgangsdaten, allerdings unter Vergrößerung des Rechenaufwands. Da sämtliche Algorithmen offenliegen [26, 32, 33], ist eine spätere Verbesserung innerhalb des hier betrachteten Entfaltungsverfahrens bedenkenenswert.

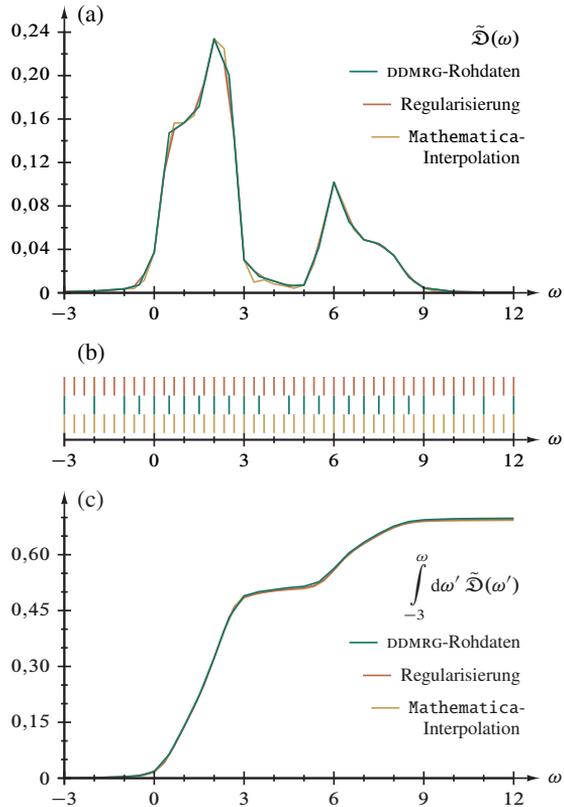


Abbildung 2.8. (a) Vergleich eines (hier stark verkleinerten) irregulären Rohdatensatzes mit dem Ergebnis zweier verschiedener Regularisierungsverfahren; (b) zu (a) gehöriges Frequenzraster in gleicher Farbkodierung; (c) Flächenintegral von (a).

Abbildungsteil (b) veranschaulicht das jeweilige Frequenzgitter und Teil (c) das Spektrale Gewicht; bei letzterem ist auf die trotz der extrem geringen Auflösung bereits sehr kleine Gesamtabweichung bei $\omega = 12$ zwischen den Ausgangs- und Enddaten hinzuweisen: eine Verringerung um 3,98 ‰ für die hier vorgestellte Methode und um 7,73 ‰ für Mathematica.

2.5. Gesamtverfahren. Die Zusammensetzung der drei vorgenannten Prozeduren *Minimierung/Entfaltung*, *Glättung* und *Rückfaltung* zum Gesamtverfahren führt unter der Voraussetzung regulärer und nach den Erfordernissen aufgelöster Ausgangsdaten zu diesem Programmablauf:

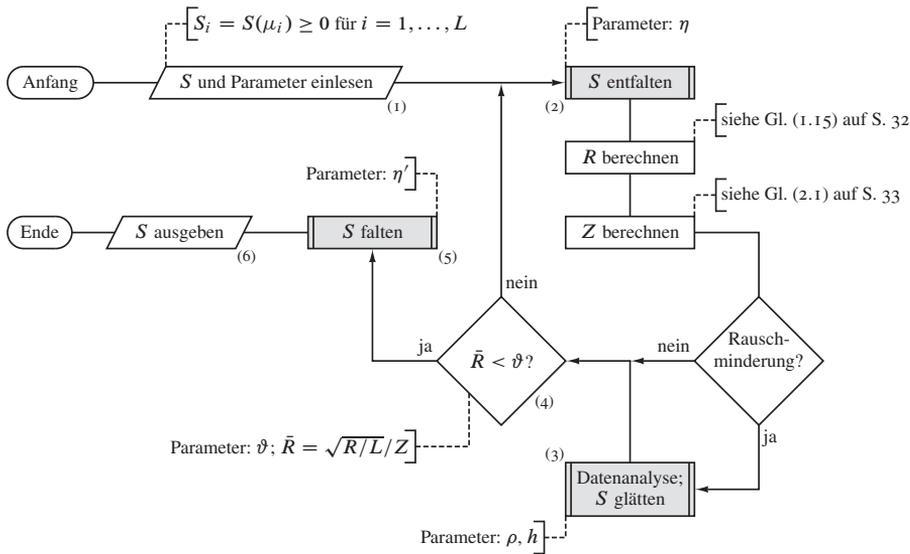


Abbildung 2.9. Programmablaufscha des Gesamtverfahrens. Der zentrale Programmteil, Flußdiagrammelement (2), ist in Abbildung 2.5 (Abschn. 2.3.1 ab S. 39) detailliert gezeichnet; ebenso ist der Ablauf der Unterprogramme Element (3) in den Abbildungen 2.2 und 2.6 (Abschn. 2.2.2 ab S. 36 u. 2.3.2 ab S. 41) und Element (5) in Abbildung 2.4 (Abschn. 2.2.3 ab S. 38) ausführlich dargestellt.

- 1 Ausgangskonfiguration $S_i = S(\mu_i)$ für $i = 1, 2, \dots, L$, siehe Flußdiagrammelement (1) in Abbildung 2.9, sowie Entfaltungparameter η , Rückfaltungparameter η' sowie optional ρ und h einlesen
- 2 Entfaltungsunterprogramm, Element (2), Abschnitt 2.3.1 und Abbildung 2.5, aufrufen: Ausgangsdaten beziehungsweise Zwischenergebnis und η übergeben, entfaltenen Datensatz übernehmen
- 3 R berechnen, siehe Gleichung (1.15)
- 4 Z berechnen, siehe Gleichung (2.1)
- 5 Optional Rauscherkennungsroutine, Element (3), siehe Abschnitt 2.2.2 und darin Abbildung 2.2, und darin das optimierte Glättungsunterprogramm, vergleiche Abschnitt 2.3.2 und dort Abbildung 2.6, aufrufen: Zwischenergebnis sowie Parameter ρ und h der Rauschminderung übergeben, geglättete Daten übernehmen
- 6 Verfahren beenden²⁷, wenn beispielsweise gemäß Gleichung (2.2) $\bar{R} < \vartheta$; sonst zu Schritt 2 zurückkehren

²⁷Vergleiche Fußnote 18.

- 7 Rückfaltungsunterprogramm, Element (5), Abschnitt 2.2.3 und Abbildung 2.4, aufrufen: (geglättetes) Endergebnis und η' übernehmen, erneut gefalteten Datensatz übergeben
- 8 Endkonfiguration $S(\mu) = S$ ausgeben, Element (6)

Da in dieser Arbeit durch Programmschritt 7 effektiv immer nur eine Verminderung der Faltungsbreite erfolgt, im Allgemeinen auf $\eta' = \eta/10$, niemals indes „vollständig“ entfaltet wird, steht für das Gesamtverfahren synonym oft der Begriff *Faltungsminderung*.

2.6. Technische Details. Die in den vorausgegangenen Abschnitten beschriebenen Algorithmen wurden zunächst mit den numerischen Fähigkeiten von *Mathematica* und, auf verschiedene Prüfrechnungen folgend, weitestgehend ANSI-konform in der Programmiersprache C implementiert. Um größtmögliche Portabilität zu erreichen, beschränkt sich das Gesamtprogramm auf die Verwendung der Standardbibliotheken. Der Quelltextumfang in kompakter Formatierung ohne Kommentare und strukturierende Leerzeilen beläuft sich auf etwa 350 Zeilen, welche sowohl sämtliche Funktionen zum Einlesen der Laufzeitparameter und Rohdaten (Standardeingabe) als auch zur Ausgabe der Ergebnisse (Datei) und Kontrollinformationen (Fehlerkanal) beinhalten.

Zwecks graphischer Darstellung der Zwischenergebnisse und der in den folgenden Abschnitten erläuterten Kenngrößen des iterativen Entfaltungsprozesses wird in einer erweiterten Fassung auf das Visualisierungsprogramm *Grace* [46] zurückgegriffen. Dabei erfolgt die Datenausgabe unmittelbar über die davon bereitgestellte Schnittstelle auf die Graphikanzeige des Endgerätes; zudem werden sämtliche Informationen über den Verfahrensablauf von *Grace* zwischengespeichert, abschließend in dessen Dateiformat abgelegt und optional auf einem PostScript-Seitendrucker ausgegeben. Diese Funktionalität vergrößert die Länge des Entfaltungsprogrammes besonders durch die aufwendigen Anweisungen zur Formatierung der Graphikausgabe auf etwa 500 Quelltextzeilen.

Auf der Entwicklungsumgebung, einer *RISC-workstation* vom Typ HP A9638A [47], beläuft sich Rechenzeit des höchstmöglich für den Befehlssatz des Mikroprozessors optimierten Maschinenprogramms für 1000 Entfaltungszyklen mit Rauschdämpfung bei 4000 Datenpunkten auf etwa 54 Minuten; damit ist im Allgemeinen der zeitliche Aufwand für die Entfaltung gegenüber jenem für die Gewinnung der DDMRG-Ergebnisse vernachlässigbar. Aus diesem Grunde ist zunächst nicht in Leistungsverbesserungen investiert worden, die über die Implementierung des Entfaltungsverfahrens mit $\sim L^2$ Operationen (statt $\sim L^3$ bei geradliniger Umsetzung) hinausgehen, siehe Abschnitt 2.3.1.

3. Anwendung auf analytische Prüffunktionen charakteristischer Form

Im Rahmen einer ersten Anwendung des im Abschnitt 2.5 vorgestellten Gesamtverfahrens betrachten wir analytische Funktionen mit wohlbekannter Faltung mit einem Lorentzkern,

$$k_\eta(\zeta) = \frac{\eta}{\zeta^2 + \eta^2}, \quad (3.1)$$

welche entweder exakt berechenbaren Operatorspektren entsprechen oder charakteristische Merkmale wie *Unstetigkeiten*, *Asymptoten* und insbesondere *Polstellen* der physikalisch relevanten Spektren aufweisen, sofern für diese keine analytischen Lösungen bekannt sind. Die zugehörigen Datensätze wurden mit *Mathematica* generiert, das auch die Erzeugung künstlichen Rauschens, siehe Abschnitt 3.1.4, durch die Verfügbarkeit gleichverteilter Pseudozufallszahlen hoher Qualität vereinheitlicht und vereinfacht hat.

Das Vorgehen bei der Entfaltung der Testfunktionen erfolgt geradlinig nach dem in Abschnitt 2.5 vorgestellten Gesamtverfahren unter optional durchgeführter Glättung. Im Sinne einer besseren Qualitätsbewertung der Konvergenz werden indes stets 1000 vollständige Zyklen durchlaufen; es wird also nicht nach Unterschreiten eines der in Abschnitt 2.1.2 behandelten Schwellenwerte automatisiert abgebrochen.

3.1. Vorbemerkungen.

3.1.1. *Notation.* Gleichwohl wir im Weiteren Funktionswerte einer diskretisierten Variablen betrachten, notieren wir diese wie kontinuierliche Funktionen und bezeichnen diese sämtlich mit dem lateinischen Buchstaben T und – sofern es sich um die Faltung handelt – konsequent mit dessen Frakturvariante \mathfrak{T} (respektive $\tilde{\mathfrak{T}}$ im Falle „verrauschter“ Daten), ergänzt durch eine hochgestellte eindeutige abkürzende Bezeichnung des Typs, Gleichung (1.2) entsprechend. Abweichend verwenden wir jedoch für das Testfunktionsargument, die Frequenz, vereinheitlicht das Zeichen ω , selbst wenn von der Faltung die Rede ist. Der Breitensparameter des Faltungskernes wird analog zu Gleichung (3.1) ebenfalls tiefgestellt unterhalb des Funktionstyps vermerkt; dem Argument der Funktion folgend findet sich gegebenenfalls (hochgestellt) die Angabe des schon in Abschnitt 2.2.2 und später noch eingehend diskutierten Rauschdämpfungsparameters ρ und (tiefgestellt) jene der Entfaltungsschrittnummer i :

$$\tilde{\mathfrak{T}}_{\eta}^{\text{Typ}}(\omega)_i^{\rho}. \quad (3.2)$$

Ähnliches gilt auch für die Darstellung anderer Kenngrößen des Verfahrens; R_0 steht beispielsweise für den Wert des Minimierungsparameters vor dem ersten, R_i für jenen nach dem i ten Schritt.

3.1.2. *Gemeinsame Parameter; δ -Funktion.* Die allen Funktionen gemeinsamen Parameter legen wir folgendermaßen fest und orientieren uns dabei bereits an den Erfordernissen der im Kapitel 4 folgenden Anwendung auf DDMRG-Ergebnisse: *Skalierender Faktor* $t = 1$, *Abstand* zweier benachbarter Frequenzen im diskretisierten Spektrum („Auflösung“) $d = t/100$, *Faltungsbreite* $\eta = 10d$ und *Bandlücke* $\Delta = 6t/10$. Überdies wird eine der Frequenzrasterweite angepaßte *Rechteckfunktion* analog einer δ -Distribution benötigt,

$$\Lambda(\omega) = \begin{cases} 1/d & \text{falls } |2\omega| \leq d \\ 0 & \text{sonst} \end{cases} \quad \text{mit} \quad \int_{-\infty}^{+\infty} d\omega \Lambda(\omega) = 1. \quad (3.3)$$

3.1.3. *Gliederung der Abbildungen.* Übersichtsgraphen der Testfunktionen sowie der zunächst ge- und anschließend wieder entfalteten rauschfreien Datensätze finden sich im jeweiligen Abschnitt; die umfangreicheren Abbildungen zu den Entfaltungsergebnissen mit Dämpfung des Rauschens sind mit weitergehenden Erläuterungen im Abbildungsabschnitt 3.10, beginnend auf Seite 64, gesammelt. Die Doppelseiten enthalten links eine Darstellung sämtlicher im Verfahren unter Anwendung des Glättungsparameters $\rho = 0,1$ (synonym: Rauschdämpfungsparameter) auftretender Größen und rechts einen Vergleich verschiedener Werte von ρ einschließlich „abgeschaltetem“ Rauschminderungsverfahren. Die Darstellung der Studienergebnisse von *finite-size*-Effekten auf dem Seiten 80 und 81 weicht in deren Gliederung ab; Anmerkungen diesbezüglich finden sich im Unterabschnitt 3.10.2. Die Resultate der gesamten Untersuchung sind ausnahmslos *reproduzierbar*.

3.1.4. *Rauschunterdrückung.* Zu einer vollständigen und systematischen Untersuchung der Rauschunterdrückung bedürfte es der Behandlung einer Vielzahl an Ausgangsdatsätzen, welche sich allein aus der unabhängigen Größenvariation der stochastischen Störung der Frequenzen $\mu_j + \epsilon_j$, beispielsweise mit $m = 1, 2, 3$ in $\epsilon_j \in [-10^{-m}, +10^{-m}]$, und der Funktionswerte $S_j + \mathcal{E}_j$, ebenfalls etwa mit $m = 1, 2, 3$ für zufällige $\mathcal{E}_j \in [-10^{-m}, +10^{-m}]$, ergäbe. Den Studienumfang weiter vergrößernd müßten zudem in Abhängigkeit von der betrachteten Prüffunktion empirisch verschiedene Werte der Rauschdämpfungsparameter ρ und h verwendet werden (siehe Abschnitte 2.2.2 und 2.3.2), zum Beispiel für ρ ganzzahlige Vielfache von η , welches aufgrund dessen Bindung an das Frequenzraster naheliegt.

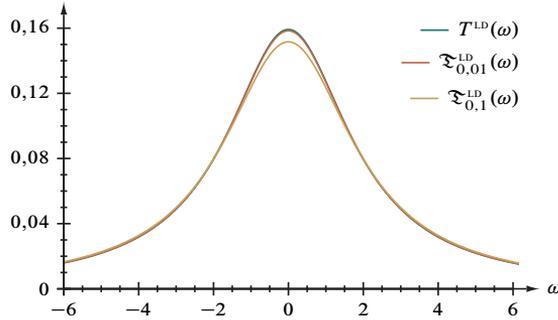
Einerseits hat sich jedoch in Voruntersuchungen gezeigt, daß für die Glättung in der Praxis dem „horizontalen“ Parameter ρ , dem Frequenzabstand zweier Extrema, die entscheidende Bedeutung zukommt und die Wahl des „vertikalen“ Abstandes h , quasi des Höhenunterschiedes, keine nennenswerte Auswirkung hat.²⁸ Da sich zudem andererseits die statistischen Fehler auf den uns letztlich interessierenden DDMRG-Spektralfunktionen in vielen Fällen auf die Größenordnung 10^{-2} beschränken und wir die L diskreten Frequenzen im betrachteten endlichen Bereich $B \subset \mathbb{R}$, unabhängig von einem vollständig regulären ($\mu_{j+1} - \mu_j = \mu_{j+2} - \mu_{j+1}$ für $j = 1, 2, \dots, L - 2$) oder einem aus mehreren jeweils regulären Intervallen unterschiedlicher Rasterweite bestehenden Gitter, als im Rahmen der Numerik „genügend exakt“ (Fehler sehr viel kleiner als 10^{-3}) ansehen, wird im Folgenden einzig die Variation von $\rho = n\eta$ mit $n = 1, 2, 3$ genauer untersucht.

3.2. Lorentzverteilung.

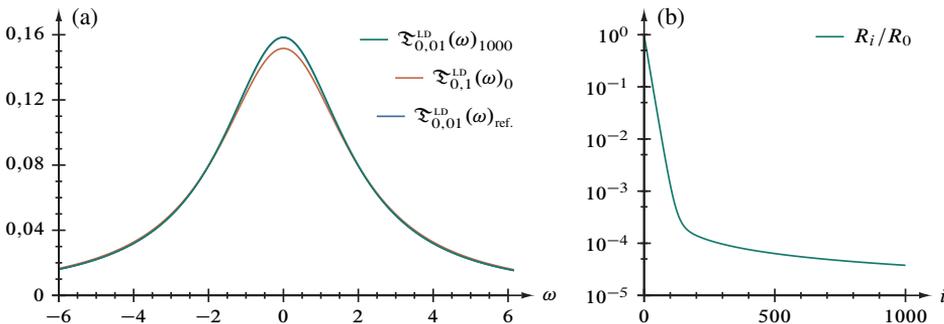
3.2.1. *Charakterisierung.* Als Beispiel für eine *glatte* und *unendlich breite*, also auf ganz \mathbb{R} endliche Funktion betrachten wir die Cauchy-Lorentz-Verteilung (kurz LD für *Lorentz distribution*) selbst,

$$T^{\text{LD}}(\omega) = \frac{1}{\pi} \frac{2t}{(2t)^2 + (\omega - \alpha)^2} . \quad (3.4)$$

²⁸Siehe Fußnote 25

Abbildung 3.1. Lorentzverteilung, $T^{\text{LD}}(\omega)$, und deren Faltung $\mathfrak{F}^{\text{LD}}(\omega)$.

welche 1-normiert und deren Parameter $\alpha = 0$ ist. Die Lorentzverteilung stellt insofern unter allen Tests eine Ausnahme dar, als daß diese nicht prototypisch für eine hier behandelte Spektralfunktion steht²⁹. Dennoch ist, wie im Weiteren gezeigt, der glatte Funktionsverlauf eine besondere Herausforderung für die Rauschdämpfung.

Abbildung 3.2. (a) Entfaltung der Lorentzverteilung $\mathfrak{F}^{\text{LD}}(\omega)$ in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration.

3.2.2. Ergebnisanalyse. Die Faltungsminderung der rauschfreien Testfunktion von $\eta = 0,1$ auf $\eta' = 0,01$ ist, wie aus Abbildung 3.2 und darin insbesondere Teil (b) zu ersehen, unproblematisch; der probalber nur mit $\eta = 0,01$ gefaltete Referenzdatensatz wird vollkommen reproduziert, und Konvergenz tritt bereits nach circa 150 Zyklen ein.

Abbildung 3.17 auf Seite 66 zeigt Verlauf und Ergebnis der Anwendung des Entfaltungsverfahrens auf die rauschbehafteten Daten ($\mathcal{E}_j \in [-10^{-2}, +10^{-2}]$, $\rho = 0,1$); im Besonderen sind in Abbildungsteil (a) die gefaltete, $\mathfrak{F}_{0,1}^{\text{LD}}(\omega)_0$ (Ausgangsdaten), und die ent- sowie rückgefaltete Lorentzverteilung, $\mathfrak{F}_{0,01}^{\text{LD}}(\omega)_{1000}$, dargestellt. Letztgenannte wird zum Vergleich durch die (rauschfreie) Faltung $\mathfrak{F}_{0,01}^{\text{LD}}(\omega)$ ergänzt.

²⁹Diese Funktion tritt in der klassischen Festkörpertheorie im Kontext des Drude-Lorentz- beziehungsweise Lorentzoscillator-Modells auf [30].

Während alle Gütekriterien des Entfaltungsverfahrens, Abbildungsteile (b), (d) und (f), Konvergenz bereits nach etwa 100 Iterationen aufweisen, ist anhand des Funktionsgraphen von $\tilde{\mathfrak{T}}_{0,01}^{\text{LD}}(\omega)_{1000}$ sowie dessen Gradienten in Teil (c), $\partial_{\omega} \tilde{\mathfrak{T}}_{0,01}^{\text{LD}}(\omega)_{1000}$, deutlich nur unvollkommene Rauschminderung zu erkennen. Auch die in Abbildung 3.18 auf Seite 67 dargestellte sukzessive Erhöhung des Verfahrensparameters ρ von 0,1 auf 0,3 verbessert das Ergebnis nicht wesentlich; es scheint, als veränderte das nach jedem Entfaltungs- und Glättungsschritt verbleibende Rauschen seine Struktur hin zur (aus Sicht des Algorithmus) „vermeintlichen“ Nutzinformation, welche sich weiterer Glättung entzieht. Beachtenswert ist jedoch, daß $\rho \geq 0,2$ ein Konvergieren verhindert, wie aus den Teilen (b) und (d) ersichtlich. Dennoch beweist ein Vergleich dieser Resultate mit dem ebenfalls einzeichneten Ergebnis der nicht glättenden Entfaltungsprozedur den Wert einer (im Einzelfall gegebenenfalls nur) unvollkommenen Rauschdämpfung eindrucksvoll.

3.3. Halbelliptische Verteilung.

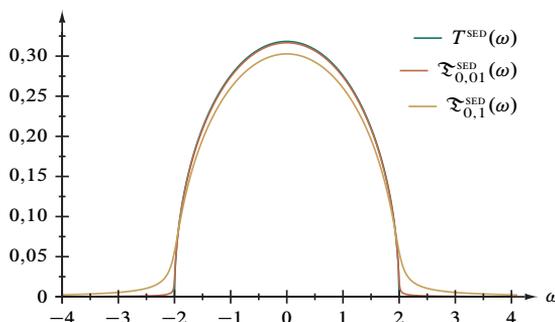


Abbildung 3.3. Halbelliptische Verteilung, $T^{\text{SED}}(\omega)$, und deren Faltung $\mathfrak{F}^{\text{SED}}(\omega)$.

3.3.1. *Charakterisierung.* Beispielhaft für eine ebenfalls *glatte*, jedoch nur innerhalb eines *endlichen Intervalls* von \mathbb{R} endliche Funktion ist die (auf 1 normierte) Halbelliptische Verteilung (*semi-elliptic distribution*, SED),

$$T^{\text{SED}}(\omega) = \begin{cases} \frac{1}{2\pi t^2} \sqrt{(2t)^2 - \omega^2} & \text{falls } |\omega| \leq 2t \\ 0 & \text{sonst} \end{cases}. \quad (3.5)$$

Diese Funktion, physikalisch betrachtet das „Gegenstück“ zu jener im folgenden Abschnitt 3.4 behandelten, tritt in der Festkörpertheorie als analytische Lösung der Zustandsdichte des Hubbardmodells im Limes *unendlich starker Wechselwirkung*³⁰ in Erscheinung [35] – ein Ergebnis, das zwar im Zusammenhang mit DDMRG-Rechnungen nicht weitergehend vertieft wird, indes (gleich der Lorentzverteilung) ein Test für den

³⁰Dieser Grenzfall ist zentraler Gegenstand des zweiten Teiles dieser Arbeit.

Rauschdämpfungsalgorithmus und in Abschnitt 3.9 als Grenzfall $l \rightarrow \infty$ von Gleichung (3.12) für das Verhalten des Entfaltungsalgorithmus gegenüber *finite-size*-Effekten ist.

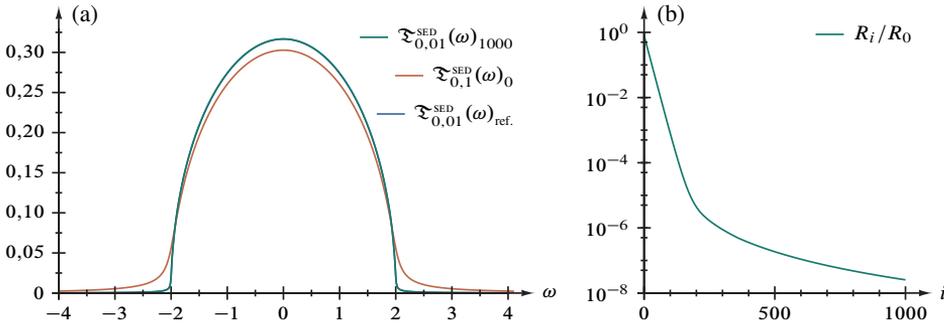


Abbildung 3.4. (a) Entfaltung der Halbelliptischen Verteilung in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration.

3.3.2. *Ergebnisanalyse.* Abbildung 3.4 belegt, daß die Entfaltung dieser Prüffunktion mit $\eta = 0,1$ unter abschließender Rückfaltung mit $\eta' = \eta/10$ zu einem Ergebnis führt, das mit der direkten Faltung mit $\eta = 0,01$ vollständig übereinstimmt. Ein Konvergieren ist bereits nach etwa 200 Entfaltungsschritten zu erkennen, vergleiche Teil (b), wobei der Endwert R_{1000}/R_0 bemerkenswerterweise um nahezu drei Größenordnungen unter jenem der Lorenzverteilung, Abschnitt 3.2, liegt.

Die Betrachtung der rauschmindernden Entfaltung mit $\rho = 0,1$ der gestörten Ausgangsdaten in Abbildung 3.19 auf Seite 68 (Erläuterungen finden sich dazu in der Einführung zum Abbildungsabschnitt 3.10, Seite 64) erweist in den Abbildungsteilen (b), (d) und (f) schnelle Konvergenz bei relativ guter Annäherung an die ungestörten Referenzdaten, auch wenn der Wert des Minimierungsparameters R_{1000}/R_0 fünf Größenordnungen über der rauschfreien Probeentfaltung in Abbildung 3.4 liegt. Auffällig ist der in (f) markierte reproduzierbare Sprung im Integral der Korrekturquadrate, welcher sich notwendigerweise auch bei Betrachtung einer Ausschnittsvergrößerung von Teil (d) findet; ein Vergleich des Entfaltungsergebnisses nach 288 Zyklen, also „vor dem Sprunge“ und vollen 1000 Entfaltungsschritten zeigt insbesondere in den Teilen (a) und (c) nur noch marginale Verbesserungen, wie auch dem eindeutigen Konvergenzverhalten nach zu erwarten ist.

Aus Abbildung 3.20 auf Seite 69 ist einerseits die generelle Wirksamkeit der Rauschdämpfung ersichtlich und andererseits zu erkennen, daß eine Vergrößerung des Enttauschparameters $\rho > 0,1$ keine nennenswerte Verbesserung der Glättung bewirkt, sondern nur in Teil (d) den Verlauf von $\int_{-4}^{+4} d\omega \zeta^2(\omega)_i^\rho$ für $i \gtrsim 200$ sägezahnartig verändert – entsprechend einer Erhöhung der Korrekturstärke, welche sich aus der sprungbedingten Verschlechterung der Entfaltung ergibt.

3.4. Quadratwurzeldivergenz; Divergenzartefakt.

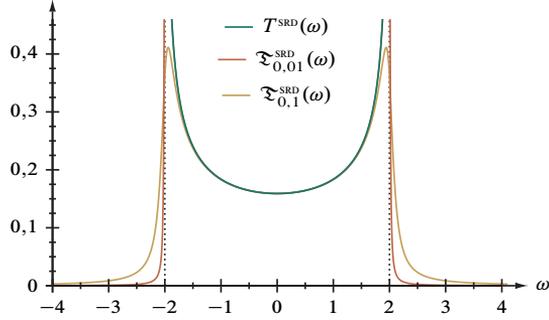


Abbildung 3.5. Quadratwurzeldivergenz, $T^{\text{SRD}}(\omega)$, und deren Faltung $\mathfrak{Z}^{\text{SRD}}(\omega)$.

3.4.1. Charakterisierung. Als Beispiel für eine *symmetrische* Funktion mit zwei *Divergenzen*, welche wie die Halbelliptische Verteilung eine *endliche Bandbreite* aufweist, läßt sich auf die bereits in Abschnitt 1.1.2 vorgestellte (1-normierte) Zustandsdichte $D(\omega)$, Gleichung (1.7), zurückgreifen, die *square-root divergence* (SRD)

$$T^{\text{SRD}}(\omega) = \begin{cases} \frac{1}{\pi} \frac{1}{\sqrt{(2t)^2 - \omega^2}} & \text{falls } |\omega| \leq 2t \\ 0 & \text{sonst} \end{cases}. \quad (3.6)$$

Neben den sich anschließenden Prüffunktionen kommt diesem Testfall besondere Bedeutung für das weitere Studium der Algorithmen zu, da es sich um die analytische Lösung eines mit DDMRG untersuchten Systemes handelt, nämlich – im Gegensatz zu Abschnitt 3.3 – um das Hubbardmodell im Grenzfalle *fehlender* Wechselwirkung [31]. Gegenstand eingehender Betrachtung wird diese Funktion im Abschnitt 4.2.1 (vergleiche dazu auch Abbildungen 4.9 und 4.10) sowie bei der Behandlung von Effekten endlicher Systemgröße im Abschnitt 3.9 als Grenzfalle $l \rightarrow \infty$ von Gleichung (3.11). Die besondere Herausforderung besteht aus technischer Sicht in der „Rekonstruktion“ der Polstellen bei $\omega = \pm 2$ aus den Maxima der gefalteten Rohdaten, welche – wie unten gezeigt – vom Rauschdämpfungsverfahren durch sorgfältige Parameterwahl nicht als Störung erfaßt werden dürfen.

3.4.2. Ergebnisanalyse. Wie aus Abbildung 3.6 unmittelbar ersichtlich, reproduziert die Faltungsminderung von $\eta = 0,1$ auf $\eta' = 0,01$ den Referenzdatensatz mit beachtenswerter Qualität unter zuverlässiger Konvergenz, aber nicht vollkommen: Die Maxima liegen zum einen bei $\mathfrak{Z}_{0,01}^{\text{SRD}}(\omega)_{1000} \approx 0,946$ für $\omega = \pm 1,99$, zum anderen bei $\mathfrak{Z}_{0,01}^{\text{SRD}}(\omega)_{\text{ref.}} \approx 1,239$ für $\omega = \pm 1,98$. Bedeutungsschwerer ist ein im Bereich $|\omega| \leq 1$ symmetrisch rechts des Pols bei $\omega = -2$ und links des Pols bei $\omega = +2$ auftretendes Artefakt im Ergebnis, welches beispielhaft die Ausschnittsvergrößerung

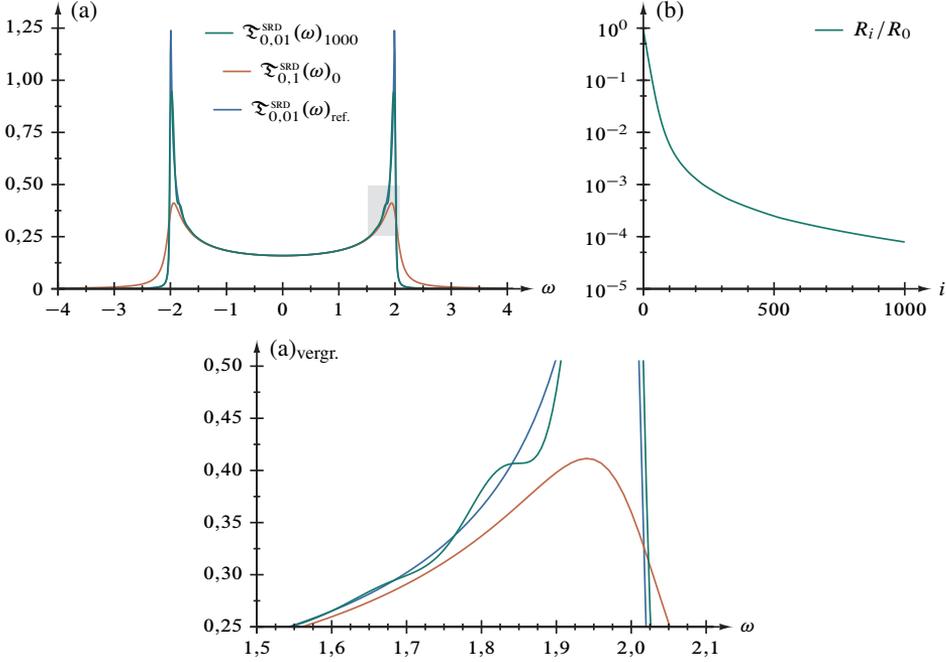


Abbildung 3.6. (a) Entfaltung der Quadratwurzeldivergenz in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration; (a)_{vergr.} Vergrößerung des in (a) grau hinterlegten Bereichs.

von (a) verdeutlicht; es scheint, als bliebe das Maximum der Rohdaten als Sattelpunkt im Entfaltungsergebnis leicht frequenzversetzt „erhalten“. Dieses stets nur in Intervallen mit $T(\omega) > 0$ auftretende und für die künftige Anwendung der Algorithmen auf DDMRG-Ergebnisse bedeutsame Phänomen bezeichnen wir im Weiteren als *Divergenzartefakt*.

Auch die Entfaltung des probenhalber künstlich verrauschten Datensatzes $\tilde{\mathfrak{T}}_{0,1}^{\text{SRD}}(\omega)_0$ in Abbildung 3.21 auf Seite 70 bestätigt die Wirksamkeit der Verfahren; wie im vorausgehenden Falle der Halbelliptischen Verteilung sind jedoch Sprünge in der Auftragung des Minimierungsparameters R_i gegen die Entfaltungsschrittnummer i in Abbildungsteil (d) und – deutlicher noch – in (f), der darauf „antwortenden“ Gesamtkorrektur $\int_{-5}^{+5} d\omega \zeta^2(\omega)_i$, zu erkennen. Gleichwohl auch nach dem markanten Sprung im Konvergenzbereich (wir betrachten genauer den Schritt zuvor, $i = 441$) noch Korrekturen am Spektrum vorgenommen werden, siehe Teil (e), unterschieden sich die Funktionsgraphen $\tilde{\mathfrak{T}}_{0,01}^{\text{SRD}}(\omega)_{441}$ und $\tilde{\mathfrak{T}}_{0,01}^{\text{SRD}}(\omega)_{1000}$ in Teil (a) beziehungsweise deren Gradienten in (c) nur noch geringfügig (Maxima unter Aufhebung der Symmetrie bei $\tilde{\mathfrak{T}}_{0,01}^{\text{SRD}}(\omega_1)_{441} \approx 0,864$ und $\tilde{\mathfrak{T}}_{0,01}^{\text{SRD}}(\omega_2)_{441} \approx 0,897$ für $\omega_1 = -1,980$ und $\omega_2 = +1,970$ gegenüber $\tilde{\mathfrak{T}}_{0,01}^{\text{SRD}}(\omega_1)_{1000} \approx 0,933$ und $\tilde{\mathfrak{T}}_{0,01}^{\text{SRD}}(\omega_2)_{1000} \approx 0,960$ für $\omega_1 = -1,985$ und $\omega_2 = +1,975$).

Daß Sprünge im Verhalten des Minimierungsparameters als eine „Reaktion“ des Entfaltungsalgorithmus' auf die Stärke der Glättung zu interpretieren sind, zeigt eindrücklich Abbildung 3.22 auf Seite 71. Bereits eine Erhöhung des Rauschfilterwertes ρ von 0,1 auf 0,2 bewirkt eine drastische Störung im Konvergenzverhalten, vergleiche Abbildungsteile (b) und (d). Überdies zeigt sich auch in den rückgefalteten Datensätzen, dargestellt in (a), eine zunehmende Asymmetrie an den Polstellen; das Glättungsverfahren „greift“ sozusagen zusehends die Nutzdaten „an“.

3.5. Asymmetrische Quadratwurzeldivergenz.

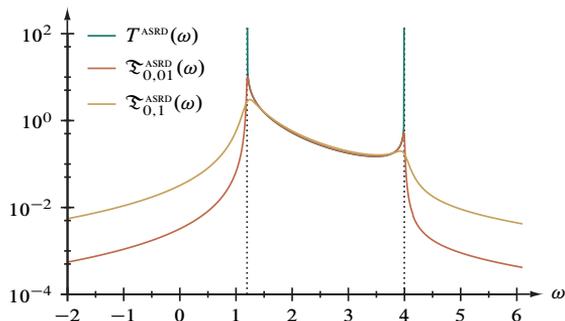


Abbildung 3.7. Asymmetrische Quadratwurzeldivergenz $T^{\text{ASRD}}(\omega)$ und deren Faltung $\mathfrak{T}^{\text{ASRD}}(\omega)$.

3.5.1. *Charakterisierung.* Als Beispiel für eine *unsymmetrische* und nur einseitig *divergierende* Funktion, gleichfalls mit *endlicher Bandbreite*, verwenden wir eine asymmetrische Quadratwurzeldivergenz (*asymmetric square-root divergence*, ASRD):

$$T^{\text{ASRD}}(\omega) = \begin{cases} \frac{32\Delta^2 t^2}{\omega^2} \frac{1}{\sqrt{(\omega^2 - 4\Delta^2)(16t^2 - \omega^2)}} & \text{falls } 2\Delta < \omega < 4t \\ 0 & \text{sonst} \end{cases} \quad (3.7)$$

Bei numerischer Auswertung der auftretenden elliptischen Integrale ergibt sich mit $\Delta = 0,6$ eine Normierung von $\int_{\mathbb{R}} d\omega T^{\text{ASRD}}(\omega) \approx 2,19296$. Wie die vorausgegangene *symmetrische* Quadratwurzeldivergenz (SRD) findet sich die nun betrachtete asymmetrische Funktion in den DDMRG-Ergebnissen, und zwar jenen zur optischen Leitfähigkeit eines Peierlsisolators [11], vergleiche Abschnitt 4.4.3 sowie Abbildungen 4.21 und 4.22. Die technischen Schwierigkeiten entsprechen aufgrund der ähnlichen Eigenschaften, zwei Pole bei $\omega_1 = 6/5$ und $\omega_2 = 4$, jenen des zuvor diskutierten symmetrischen Falles.

3.5.2. *Ergebnisanalyse.* Die „Entfaltungsleistung“ des Algorithmus' ohne Rauschminderung an der ASRD ist jener an der verwandten SRD vergleichbar: Weitgehende Rekonstruktion des Kontrolldatensatzes, Abbildung 3.6, unter zweifacher Ausbildung des zuvor beschriebenen Divergenzartefaktes an den Rändern des Abschnitts $6/5 < \omega < 4$ und leichter Verschiebung der Maxima, $\mathfrak{T}_{0,01}^{\text{ASRD}}(\omega_1)_{1000} \approx 7,715$ für $\omega_1 = 1,220$ und

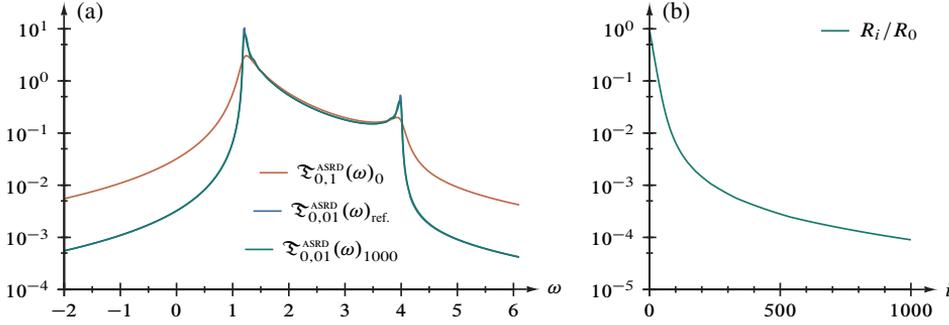


Abbildung 3.8. (a) Entfaltung der Asymmetrischen Quadratwurzeldivergenz in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i -ten Iteration.

$\mathfrak{T}_{0,01}^{\text{ASRD}}(\omega_2)_{1000} \approx 0,405$ für $\omega_2 = 3,975$ gegenüber $\mathfrak{T}_{0,01}^{\text{ASRD}}(\omega_1)_{\text{ref.}} \approx 10,277$ für $\omega_1 = 1,21$ und $\mathfrak{T}_{0,01}^{\text{ASRD}}(\omega_2)_{\text{ref.}} \approx 0,525$ für $\omega_2 = 3,99$.

Mit zugeschaltetem Glättungsverfahren, Abbildung 3.23 auf Seite 72, ergibt sich am rauschbehafteten Testdatensatz ein dem Ergebnis der symmetrischen Quadratwurzeldivergenz ähnliches Bild; die akkumulierten Korrekturen an den rückgefalteten Daten, Teil (e), nach dem auf Entfaltungszyklus 813 folgenden Sprung, siehe Teile (d), (f) und sogar (b), sind entsprechend beträchtlich. Innerhalb der verbleibenden 185 Schritte erfolgt daher nur noch eine geringfügige Verbesserung der Ergebnisse in Teil (a) durch eine kleine „Verschärfung“ an den Polstellen ($\tilde{\mathfrak{T}}_{0,01}^{\text{ASRD}}(\omega_1)_{813} \approx 7,321$ und $\tilde{\mathfrak{T}}_{0,01}^{\text{ASRD}}(\omega_2)_{813} \approx 0,379$ gegenüber $\tilde{\mathfrak{T}}_{0,01}^{\text{ASRD}}(\omega_1)_{1000} \approx 7,604$ und $\tilde{\mathfrak{T}}_{0,01}^{\text{ASRD}}(\omega_2)_{1000} \approx 0,382$ für $\omega_1 = 1,215$ und $\omega_2 = 3,970$).

Ebenso ähnelt das Entfaltungsergebnis unter Variation des Rauschfilterwertes über $\rho = \eta = 0,1$ hinaus, Abbildung 3.24 auf Seite 73, jenem der symmetrischen Quadratwurzeldivergenz. Wie aus den Abbildungsteilen (b) und (d) eindrücklich zu ersehen, wird das Gesamtverfahren schon im Konvergenzbereich, also den ersten 500 Schritten empfindlich gestört; in Teil (a) liegen die Maxima für $\rho \geq 0,2$ durch die zunehmende Glättung von „Nutzdaten“ deutlich und unter jenem im Ergebnis für $\rho = 0,1$.

3.6. Stufenfunktion; Stufenartefakt.

3.6.1. *Charakterisierung.* Als Beispiel für eine endlich breite Funktion mit einem endlichen Sprung und einer Divergenz dient folgende (auf 1 normierte) *step function*, kurz SF:

$$T^{\text{SF}}(\omega) = \begin{cases} \frac{2}{\pi} \frac{1}{\sqrt{(2t)^2 - \omega^2}} & \text{falls } 0 < \omega < 2t \\ 0 & \text{sonst} \end{cases} \quad (3.8)$$

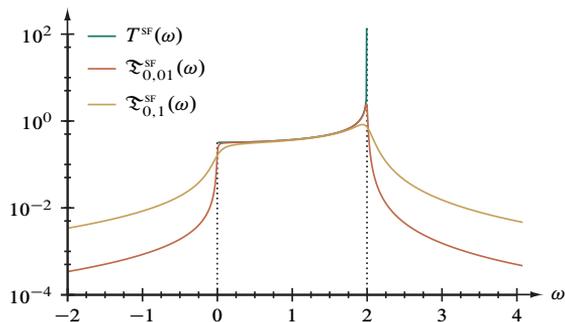


Abbildung 3.9. Stufenfunktion, $T^{\text{SF}}(\omega)$, und deren Faltung $\mathfrak{Z}^{\text{SF}}(\omega)$.

Auch die Stufenfunktion ist für die Analyse von DDMRG-Daten von großer Bedeutung und tritt beispielsweise im Kontext der Zustandsdichte des Hubbardmodells bei *endlicher* Wechselwirkung [12] auf, vergleiche Abschnitte 4.2.2 und 4.3.1 sowie Abbildungen 4.11 bis 4.16. Aus Sicht der Algorithmusprüfung stellt die Wiederherstellung des Sprunges aus dem gefalteten, „weichgezeichneten“ Datensatz eine bislang in dieser Studie noch nicht aufgetretene Herausforderung dar.

3.6.2. Ergebnisanalyse. Abbildung 3.10 zeigt bei ungestörter Konvergenz eine im Ganzen zufriedenstellende Faltungsminderung der zunächst unverrauschten Faltung der Testfunktion von $\eta = 0,1$ auf $\eta' = 0,01$. Neben dem bereits aus den Quadratwurzeldivergenzen (Abschnitte 3.4 und 3.5) bekannten Divergenzartefakt unterhalb von $\omega = 2$ und der Verschiebung des Maximums an der Polstelle, $\mathfrak{Z}_{0,01}^{\text{SF}}(\omega)_{1000} \approx 1,891$ für $\omega = 1,98$ einerseits, $\mathfrak{Z}_{0,01}^{\text{SF}}(\omega)_{\text{ref.}} \approx 2,477$ für $\omega = 1,99$ andererseits, offenbart die Ausschnittsvergrößerung von (a) eine deutliche Versetzung der Sprungkante von $\omega = 0$ ($T^{\text{SF}}(\omega) = 1/\pi \approx 0,31831$) auf $\omega = 0,075$ bei $\partial_{\omega} \mathfrak{Z}_{0,01}^{\text{SF}}(\omega)_{1000} \approx 0$ ($\mathfrak{Z}_{0,01}^{\text{SF}}(\omega)_{1000} \approx 0,315$) durch Oszillationen des Endergebnisses der Faltungsminderung im Frequenzintervall $0 < \omega \lesssim 0,4$. Dieses für Stufen typische Entfaltungsergebnis nennen wir fortan *Stufenartefakt*.

Entsprechendes findet sich in den Entfaltungsergebnissen der künstlich verrauschten Prüfdaten, siehe Abbildung 3.25 auf Seite 74: $\mathfrak{Z}_{0,01}^{\text{SF}}(\omega)_{1000}$ „schwingt“ im gesamten Frequenzbereich zwischen Sprung- und Polstelle um die Referenzdaten $\mathfrak{Z}_{0,01}^{\text{SF}}(\omega)$; unterhalb des Stufe sowie oberhalb der Divergenz treten Artefakte in Form winziger Nebenmaxima auf, zum Beispiel in der Größenordnung $3,5 \cdot 10^{-4}$ über den Vergleichsfunktionswerten bei $\omega = 3,055$, gut zu erkennen bei logarithmisch skaliertem Ordinate in Abbildungsteil (a)_{log.}. Besonders letztgenannte werden vom Glättungsalgorithmus nicht mehr als Störung erfasst und eliminiert, da sich neben einem dieser lokalen Maxima kein weiteres Extremum innerhalb des durch den Filterparameter $\rho = 0,1$ vorgegebenen Abstandes findet. Beachtlich ist auch die Konvergenz des kombinierten Entfaltungs- und

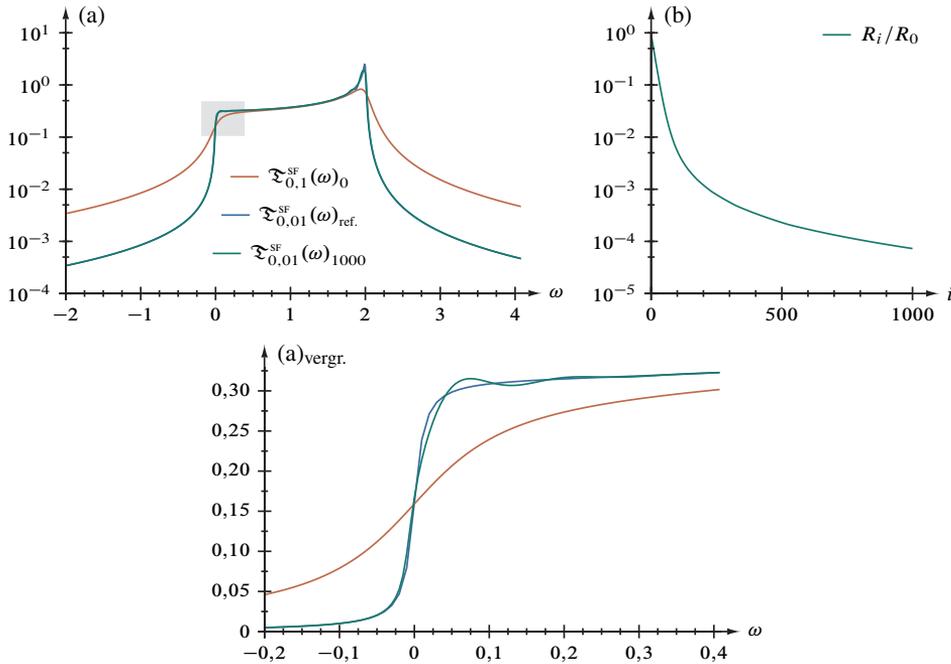


Abbildung 3.10. (a) Entfaltung der Stufenfunktion in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration; (a)_{vergr.} Vergrößerung des in (a) grau hinterlegten Bereichs (abweichend mit linearer Ordinate).

Glättungsverfahren, Teile (d) und (f), das jenem bei Berechnung der rauschfreien Daten, Abbildung 3.10 Teil (b), qualitativ nicht nachsteht.

Eine Vergrößerung des Parameters ρ verändert das Bild leicht, vergleiche zugehörige Abbildung 3.26 auf Seite 75: Während ein Filterwert von 0,2 zwar bereits die Glättung der Oszillationen oberhalb der Sprungkante bewirkt, jedoch noch keine Wirkung auf die Artefaktmaxima, etwa bei $\omega \approx -0,815$ und $-0,520$, hervorruft, verschwinden diese bei $\rho = 0,3$ – ein Wert, welcher den Abstand dieser beiden Extrema übersteigt. Paradoxerweise findet sich in den Abbildungsteilen (b) und (d) eine deutliche Erhöhung der konvergenzanzeigenden Größen bei $i = 136$, wird der Rauschdämpfungsparameter auf 0,2 gesetzt, bei Verwendung von $\rho = 0,3$ tritt indes wieder ein „weicherer“ Entfaltungsverlauf ein.

3.7. Deltafunktion nahe eines Kontinuums; Bandartefakt.

3.7.1. *Charakterisierung.* Eine für die spätere detaillierte Betrachtung der DDMRG-Befunde zur Strom-Strom-Korrelationsfunktion im Hubbardmodell [36], Abschnitt 4.4.2

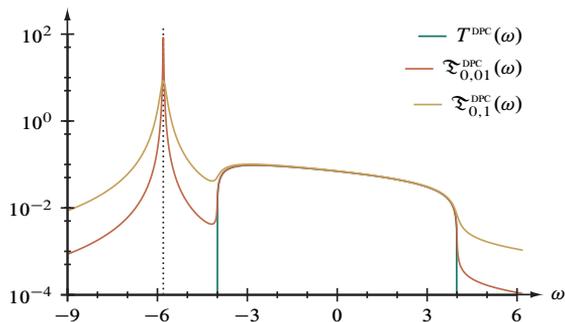


Abbildung 3.11. Deltafunktion nahe eines Kontinuums, $T^{\text{DPC}}(\omega)$, und deren Faltung $\mathfrak{F}^{\text{DPC}}(\omega)$.

sowie Abbildungen 4.19 und 4.20, wichtige charakteristische Form weißt folgende Testfunktion, ein „delta-peak close to a continuum“ (DPC), auf:

$$T^{\text{DPC}}(\omega) = \begin{cases} \frac{t^2}{2v} \frac{\sqrt{16t^2 - \omega^2}}{\omega + v + 4t^2/v} & \text{falls } |\omega| \leq 4t \\ \pi t^2 (1 - (2t/v)^2) \Lambda(\omega + v + 4t^2/v) & \text{sonst} \end{cases} \quad (3.9)$$

Im Folgenden setzen wir $v = 5t$; das Spektrale Gewicht ergibt sich zu $\int_{\mathbb{R}} d\omega T^{\text{DPC}}(\omega) = \pi$. Erstmals in der hiesigen Studie werden die beteiligten Algorithmen mit der Rekonstruktion der eingangs in Gleichung (3.3) vorgestellten „ δ -ähnlichen Rechteckfunktion“ $\Lambda(\omega)$ konfrontiert, welche nur die Breite d zweier benachbarter Frequenzen auf dem diskretisierten Gitter und die Höhe $1/d$ aufweist.

3.7.2. *Ergebnisanalyse.* Wenn auch im Bereich der „numerischen δ -Funktion“ bei $\omega = -29/5$ mit $\mathfrak{F}_{0,01}^{\text{DPC}}(\omega)_{1000} = 39,824$ nicht annähernd das Maximum $\mathfrak{F}_{0,01}^{\text{DPC}}(\omega)_{\text{ref.}} = 82,313$ erreicht wird, ist das Entfaltungsergebnis am rauschfreien Datensatz, siehe Abbildung 3.12, bei unproblematischem Konvergenzverhalten, Teil (b), qualitativ beachtlich. Eine Einschränkung ist jedoch im Intervall $|\omega| \leq 4$, also im Bereich des Bandes zu machen, und zwar oberhalb dessen unterer Kante: $\mathfrak{F}_{0,01}^{\text{DPC}}(\omega)_{1000}$ „schwimmt“ um die Referenzdaten, wie aus Abbildungsteil (a)_{vergr.} zu ersehen. Diese Erscheinung, welche auch die Prüffunktion im sich anschließenden Abschnitt 3.8 betrifft, nennen wir *Bandartefakt*.

Die Betrachtung des Faltungsminderungsverhaltens der rauschbehafteten Testdaten, Abbildung 3.27 auf Seite 76, läßt erstaunlich gute Übereinstimmung mit jenem der unverrauschten Probedaten erkennen. Obwohl eine stärkere Glättung kaum auf das Bandartefakt einwirkt, erweist sich das Gesamtverfahren selbst bei Erhöhung des Dämpfungsparameters auf Werte über $\rho = \eta$, vergleiche Abbildung 3.28 auf Seite 77, als ausgesprochen robust; insbesondere die Teile (b) und (d) zeigen ein sozusagen unbeirrtes Konvergieren.

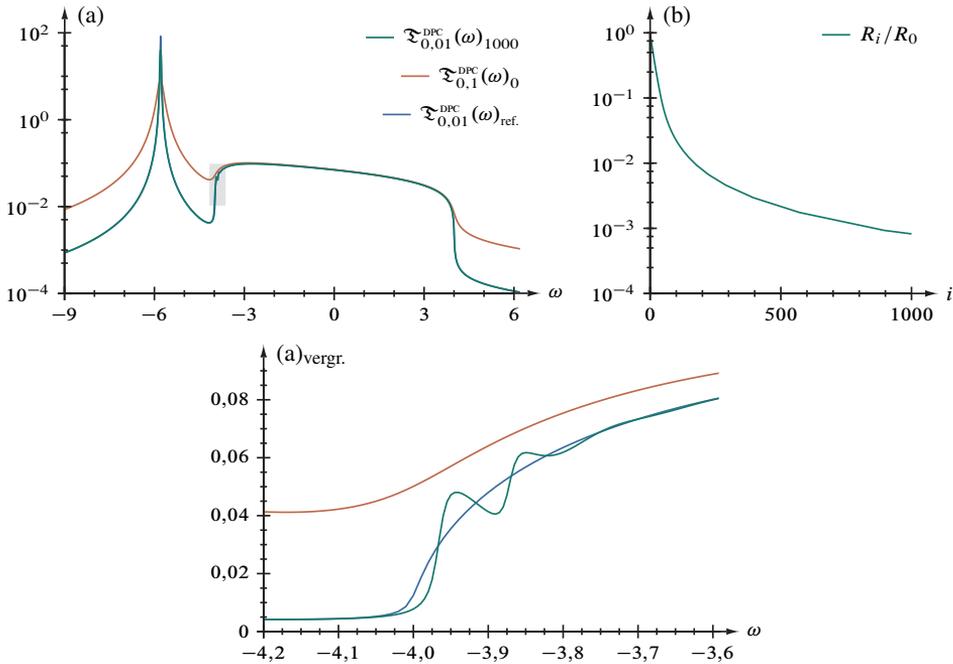


Abbildung 3.12. (a) Entfaltung einer Deltafunktion nahe eines Kontinuums in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration; (a)vergr. lineare Vergrößerung des in (a) grau hinterlegten Bereichs.

3.8. Deltafunktion zwischen zwei Bändern.

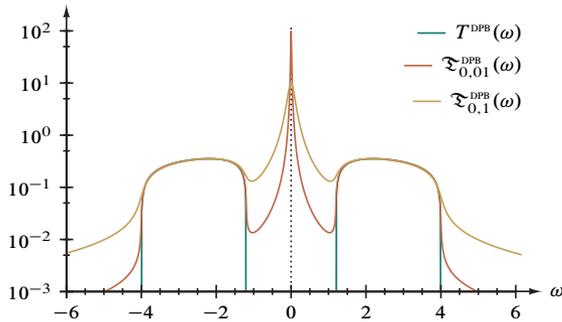


Abbildung 3.13. Deltafunktion zwischen zwei Bändern, $T^{\text{DPB}}(\omega)$, und deren Faltung $\mathfrak{F}^{\text{DPB}}(\omega)$.

3.8.1. *Charakterisierung.* Ebenso wie die im vorausgegangenen Abschnitt 3.7 behandelte ist hiesige Prüffunktion zusammengesetzt, und zwar aus einer Deltafunktion bei

$\omega = 0$, welche symmetrisch von zwei Bandbereichen umgeben ist, woraus sich die Bezeichnung *delta-peak between two bands* (DPB) herleitet. Bedeutung kommt diesem abschließenden Testfall hinsichtlich der reduzierten optischen Leitfähigkeit des Mott-Peierls-Isolators [11] zu, siehe Abschnitt 4.4.3 sowie Abbildungen 4.23 und 4.24.

$$T^{\text{DPB}}(\omega) = \begin{cases} \pi t^2 \Lambda(\omega) & \text{falls } |\omega| < 2\Delta \\ 0 & \text{sonst} \end{cases} + \begin{cases} \frac{\sqrt{(\omega^2 - 4\Delta^2)(16t^2 - \omega^2)}}{8|\omega|} & \text{falls } 2\Delta \leq |\omega| \leq 4t \\ 0 & \text{sonst} \end{cases} \quad (3.10)$$

Numerische Berechnung ergibt eine Normierung von $\int_{\mathbb{R}} d\omega T^{\text{DPB}}(\omega) \approx 4,68097$.

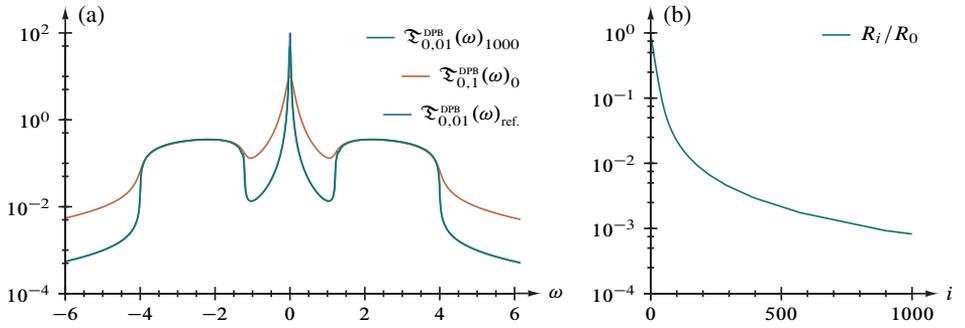


Abbildung 3.14. (a) Entfaltung einer Deltafunktion zwischen zwei Bändern in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration.

3.8.2. Ergebnisanalyse. Die Probe an den unverrauschten Daten, siehe Abbildung 3.14, besteht das Entfaltungsverfahren mit abschließender Rückfaltung bei ungestörter Konvergenz gut, wie auch im Falle der Deltafunktion nahe *eines* Kontinuums; das Maximum vergrößert sich von $\mathfrak{F}_{0,1}^{\text{DPB}}(\omega)_0 = 10,008$ auf $\mathfrak{F}_{0,01}^{\text{DPB}}(\omega)_{1000} = 47,408$, bleibt aber deutlich unter $\mathfrak{F}_{0,01}^{\text{DPB}}(\omega)_{\text{ref.}} = 97,992$. Während jedoch die äußeren Kanten der beiden Kontinuumsbereiche bei $\omega = \pm 4$ tadellos rekonstruiert werden, treten an den inneren, der Deltafunktion zugewandten Kanten bei $\omega = \pm 6/5$ zuvor benannte Bandartefakte auf.

Analog zur DPC-Prüffunktion wird auch der Test am künstlich mit Rauschen versehenen DPB-Datensatz erfolgreich absolviert, vergleiche Abbildung 3.29 auf Seite 78, und so steht besonders das Konvergenzverhalten, Teile (d) und (f), jenem bei Entfaltung des

unverrauschten Datensatzes nicht nach. Auch die Variation des Rauschminderungsparameters, Abbildung 3.30 auf Seite 79, beeinflusst die Konvergenz nicht; gegenteilig werden sogar bei $\rho = 0,3$ die Bandartefakte merklich vermindert.

3.9. Untersuchung von Effekten endlicher Systemgröße. Innerhalb der umfangreichen Verfahrenstests kommt der folgenden Faltungsminderung der Lokalen Zustandsdichte (LDOS) einer linearen Kette stark gebundener Elektronen (*tight-binding chain*, TBC) der Länge l besondere Bedeutung zu, da sich an diesem eindimensionalen Gitter die Effekte endlicher Systemgröße am einfachst möglichen Falle untersuchen lassen³¹. Zu unterscheiden sind dabei Kettenmitte und Kettenenden.

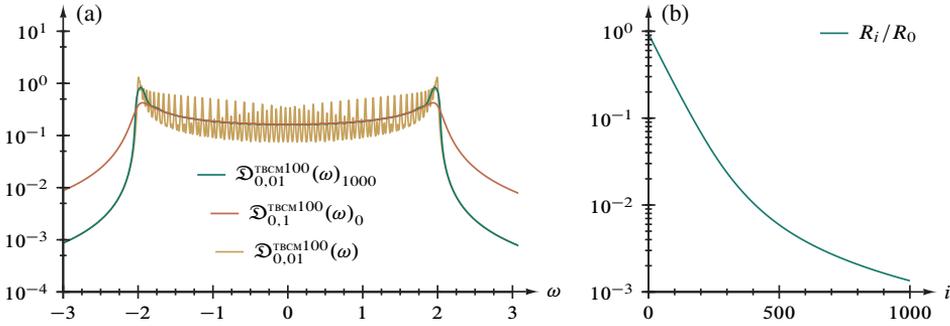


Abbildung 3.15. (a) Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen aus 100 Gitterplätzen (Mitte) in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration.

3.9.1. Kette stark gebundener Elektronen, Mitte. Die lokale Zustandsdichte in der Kettenmitte (TBCM),

$$D^{\text{TBCM}l}(\omega) = \frac{2}{l+1} \sum_{i=1}^l \left(\sin\left(\frac{l}{2l+1}\pi i\right) \right)^2 \Lambda\left(\omega + 2t \cos\left(\frac{\pi}{l+1}i\right)\right) \quad (3.11)$$

geht im Grenzfalle unendlicher Länge in die bereits zuvor behandelte Quadratwurzeldivergenz, siehe Abschnitt 3.4 und insbesondere Gleichung (3.6) über: $\lim_{l \rightarrow \infty} D^{\text{TBCM}l}(\omega) = T^{\text{SRD}}(\omega)$. Abbildung 3.31 auf Seite 80 läßt für die größtmögliche hier betrachtete Länge mit $l = 100$ Gitterplätzen diesen Übergang in Form einer gedachten „ausgleichenden“ Funktion über der schwachen Faltung $\mathfrak{D}_{0,01}^{\text{TBCM}l}(\omega)$ bereits deutlich erkennbar werden.

Die vergleichende Betrachtung der von $\eta = 0,1$ auf $\eta' = 0,01$ faltungsgemindernten Daten $\mathfrak{D}_{0,01}^{\text{TBCM}l}(\omega)_{1000}$ mit jenen der Referenzfaltung mit $\eta = 0,01$ zeigt nun, daß die „Spitzen“ der δ -Funktion (beziehungsweise der numerischen Näherungsfunktion Λ

³¹Da der Schwerpunkt dieses Abschnittes auf den *finite-size*-Effekten liegt, wurde die Betrachtung nicht auf die Auswirkung von Rauschen und dessen Dämpfung auf die Verfahrenskonvergenz ausgeweitet, was den Umfang erheblich erweiterte und zu einer eigenständigen Studie führte.

in Gleichung (3.11), siehe Gleichung (3.3)) umso weniger wieder hervortreten, je größer die Gitterplatzanzahl ist. Wesentliche Erkenntnis daraus ist, daß die Entfaltungsprozedur nicht nur die *finite-size*-Effekte nicht rekonstruiert, sondern diese zentrale Forderung der Verfahrensentwicklung mit $l \geq 100$, siehe auch Abbildung 3.15, bereits bei aus Sicht von DDMRG-Rechnungen moderaten Werten für die Systemgröße erfüllt.

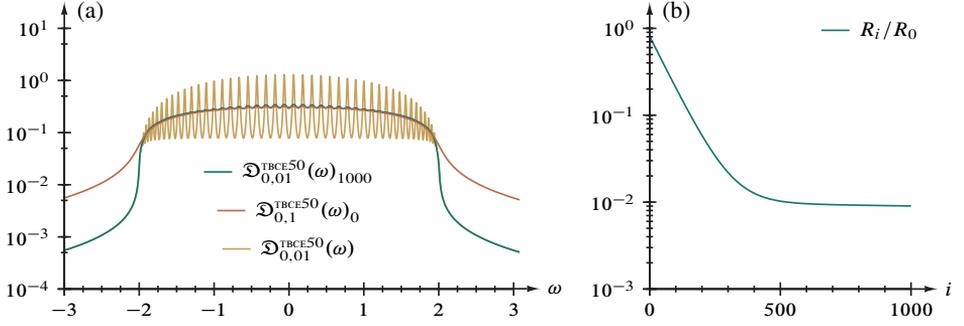


Abbildung 3.16. (a) Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen aus 50 Gitterplätzen (Enden) in 1000 Schritten; (b) Normierter Minimierungsparameter R nach der i ten Iteration.

3.9.2. *Kette stark gebundener Elektronen, Enden.* Auch die lokale Zustandsdichte in der Kettenmitte (TBCM),

$$D^{\text{TBC}l}(\omega) = \frac{2}{l+1} \sum_{i=1}^l \left(\sin\left(\frac{\pi}{l+1}i\right) \right)^2 \Lambda\left(\omega + 2t \cos\left(\frac{\pi}{l+1}i\right)\right) \quad (3.12)$$

besitzt einen bereits behandelten Grenzfall unendlicher Länge, nämlich die Halbelliptische Verteilung, siehe Abschnitt 3.3 und darin Gleichung (3.5): $\lim_{l \rightarrow \infty} D^{\text{TBC}l}(\omega) = T^{\text{SED}}(\omega)$. Analog zur im vorigen Abschnitt behandelten Kettenmitte und zugehörigen Abbildungen zeigt sich dieser Übergang in Abbildung 3.32 auf Seite 81. Ebenso stimmt der Schluß aus den Ergebnissen mit jenem aus Abschnitt 3.9.1 überein: Schon bei $l \geq 50$, vergleiche dazu Abbildung 3.16, wird die „diskret-linienhafte“ Struktur nicht mehr aufgelöst.

3.10. Abbildungen.

3.10.1. *Analytische Prüffunktionen.* Sämtlichen in diesem Abschnitt zusammengefaßten Funktions- und Datenauftragungen ist folgende Struktur gemeinsam:

Teil (a): Mit Gleichung (3.1) gefaltete ($\eta = 0,1$) und künstlich verrauschte ($\mathcal{E} = \pm 10^{-2}$) Prüffunktion \mathfrak{T}_η äquidistanter Frequenzen ω , in $i = 1000$ Schritten mit Rauschdämpfung ($\rho = \eta$) entfaltet und mit $\eta' = \eta/10$ rückgefaltet. Ergänzend ist der rauschfreie gefaltete Datensatz $\mathfrak{T}_{\eta'}(\omega)$ eingezeichnet;

Teil (b): Flächenintegral $\int_{B' \subset \mathbb{R}} d\omega \mathfrak{T}_\eta(\omega)_i$ von (a) nach der i ten Iteration;

Teil (c): Gradienten $\partial_\omega \tilde{\mathfrak{T}}_\eta(\omega)$ von (a);

Teil (d): Normierter Minimierungsparameter R_i/R_0 nach der i ten Iteration;

Teil (e): Summe aller Korrekturen $\sum_{i=1}^{1000} \zeta(\omega)_i$ an $\tilde{\mathfrak{T}}_\eta(\omega)$ nach 1000 Schritten;

Teil (f): Gesamtkorrektur $\int_{B' \subset \mathbb{R}} d\omega \zeta^2(\omega)_i$ am entfalteten Spektrum nach der i ten Iteration.

Spezifische Angaben zur Darstellung der einzelnen Testfunktionen sind im jeweiligen Abbildungstitel vermerkt. Die jeder Hauptabbildung folgende Ergänzung ist wie folgt aufgebaut:

Teil (a): Unter Variation des Rauschdämpfungsparameters ρ in $i = 1000$ Schritten störungsgemindert entfalteter und mit $\eta' = \eta/10$ rückgefalteter Datensatz der jeweiligen Prüffunktion;

Teil (b): Normierter Minimierungsparameter R_i/R_0 nach der i ten Iteration;

Teil (c): Summe aller Korrekturen $\sum_{i=1}^{1000} \zeta(\omega)_i$ am Ausgangsdatsatz nach dem letzten Entfaltungsschritt;

Teil (d): Gesamtkorrektur $\int_{B' \subset \mathbb{R}} d\omega \zeta^2(\omega)_i$ am entfalteten Spektrum nach der i ten Iteration.

3.10.2. Effekte endlicher Systemgröße. Dieser Abschnitt zeigt die ebenfalls mit Gleichung (3.1) gefalteten ($\eta = 0,1$) und mit dem im Abschnitt 2.5 erläuterten Gesamtverfahren in $i = 1000$ Schritten ent- und rückgefalteten ($\eta' = \eta/10$) Zustandsdichten einer Kette stark gebundener Elektronen, $\mathfrak{D}_\eta^{\text{TBCM}l}(\omega)$ (Kettenmitte) und $\mathfrak{D}_\eta^{\text{TBC}El}(\omega)$ (Kettenenden), bestehend aus l Gitterplätzen. Zur Qualitätsbewertung sind jeweils auch die direkten Faltungen der Gleichungen (3.11) und (3.12) mit $\eta' = 0,01$ eingezeichnet.

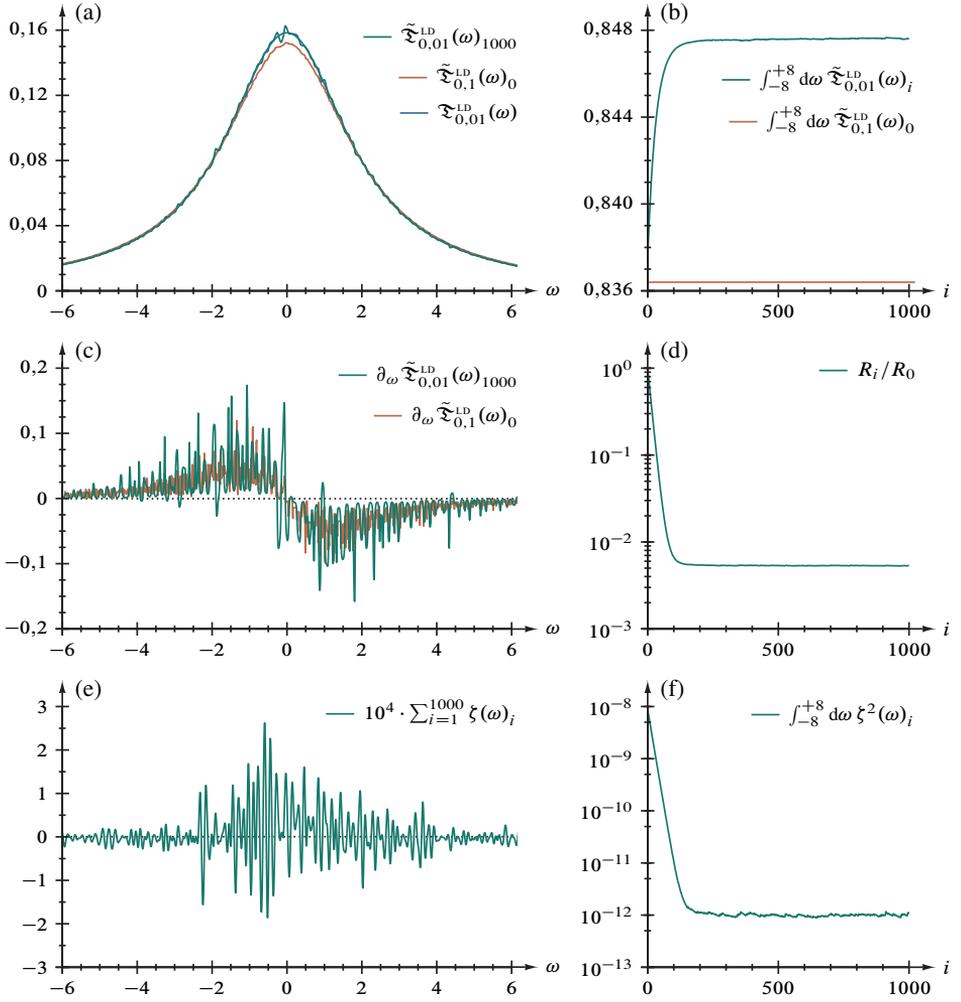
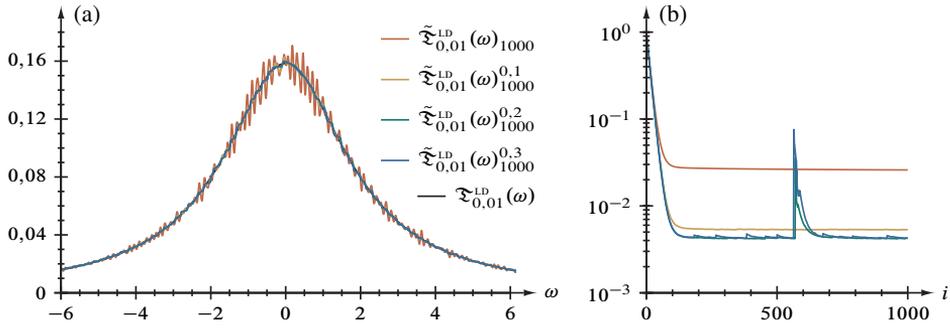


Abbildung 3.17. Entfaltung der künstlich verrauschten Lorentzverteilung, $\tilde{\mathfrak{T}}_{\eta}^{\text{LD}}(\omega)$, Gleichung (3.4), 1600 Gitterpunkte im Intervall $\omega \in [-8, +8]$.



Legende zu (b):

- R_i/R_0
- $R_i^{0,1}/R_0^{0,1}$
- $R_i^{0,2}/R_0^{0,2}$
- $R_i^{0,3}/R_0^{0,3}$

Legende zu (c):

- $10^4 \cdot \sum_{i=1}^{1000} \zeta(\omega)_i$
- $10^4 \cdot \sum_{i=1}^{1000} \zeta(\omega)_i^{0,1}$
- $10^4 \cdot \sum_{i=1}^{1000} \zeta(\omega)_i^{0,2}$
- $10^4 \cdot \sum_{i=1}^{1000} \zeta(\omega)_i^{0,3}$

Legende zu (d):

- $\int_{-8}^{+8} d\omega \zeta^2(\omega)_i$
- $\int_{-8}^{+8} d\omega \zeta^2(\omega)_i^{0,1}$
- $\int_{-8}^{+8} d\omega \zeta^2(\omega)_i^{0,2}$
- $\int_{-8}^{+8} d\omega \zeta^2(\omega)_i^{0,3}$

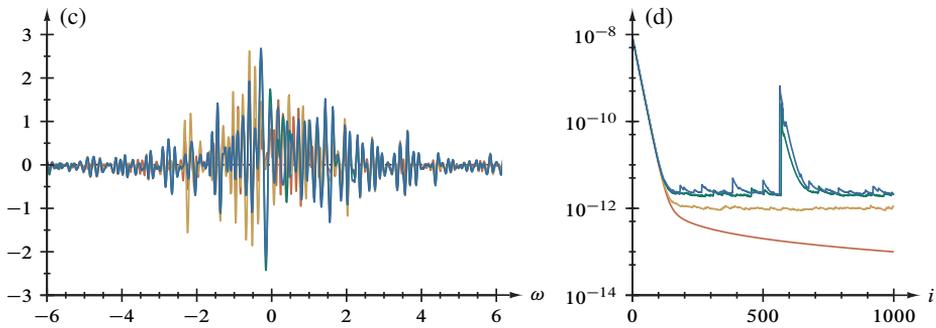


Abbildung 3.18. Entfaltung der Lorentzverteilung, Abbildung 3.17, mit unterschiedlich starker Rauschunterdrückung.

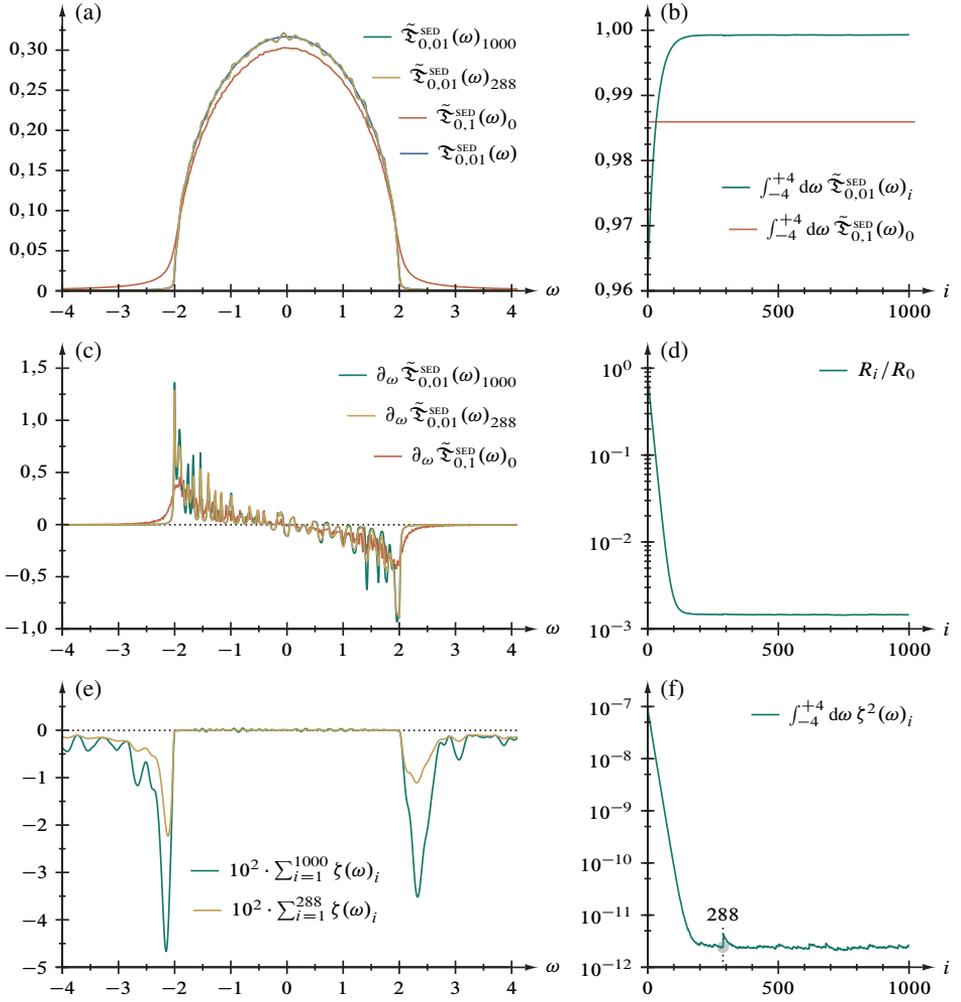


Abbildung 3.19. Entfaltung der künstlich verrauschten Halbelliptischen Verteilung, $\tilde{\mathfrak{T}}_\eta^{\text{SED}}(\omega)$, Gleichung (3.5), 1000 Gitterpunkte im Intervall $\omega \in [-5, +5]$. Zusätzlich ist das Zwischenergebnis nach 288 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

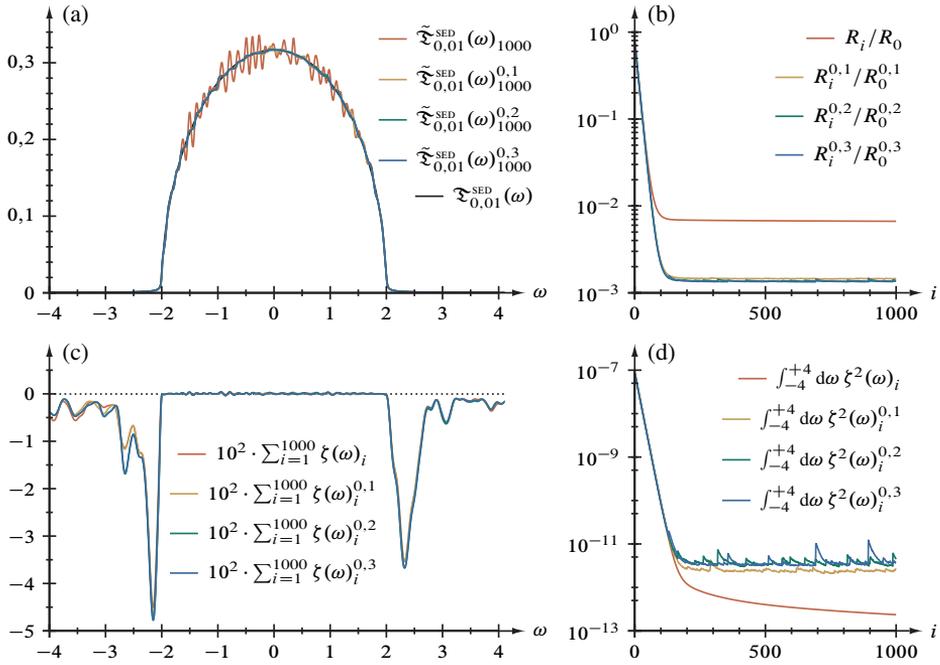


Abbildung 3.20. Entfaltung der Halbelliptischen Verteilung, Abbildung 3.19, mit unterschiedlich starker Rauschunterdrückung.

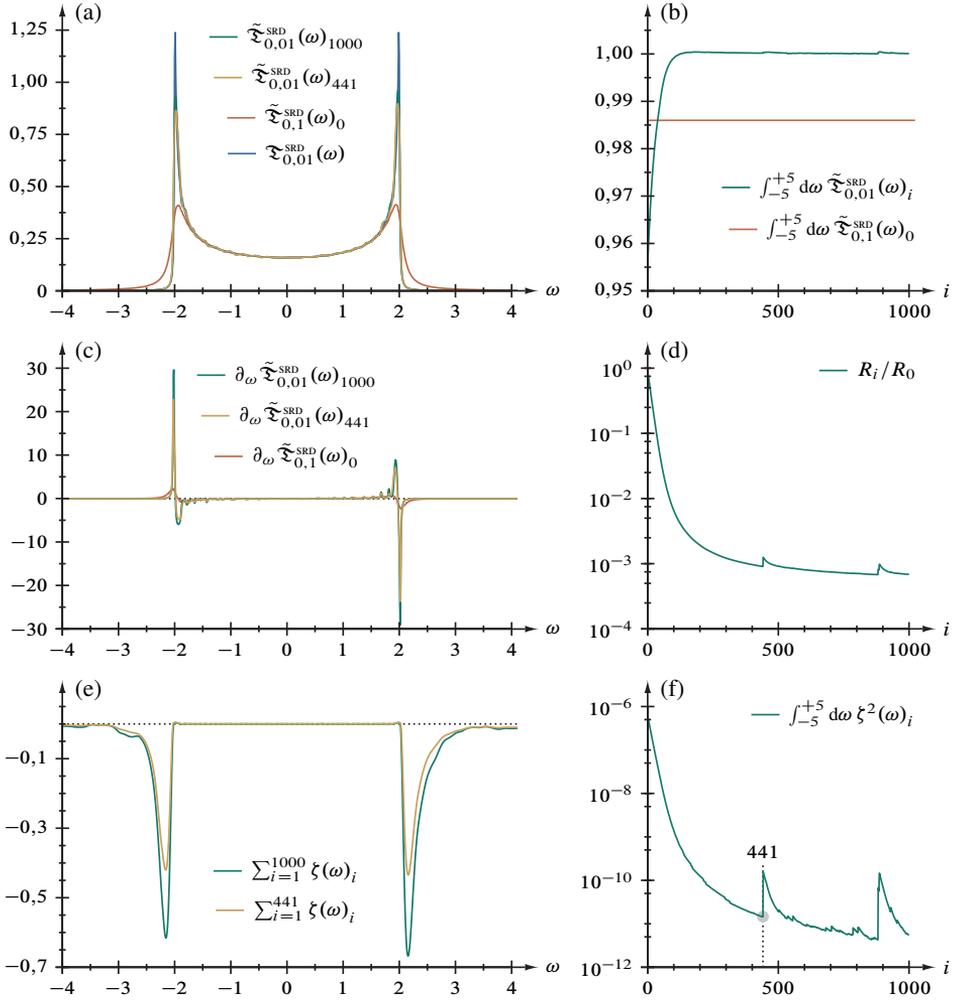
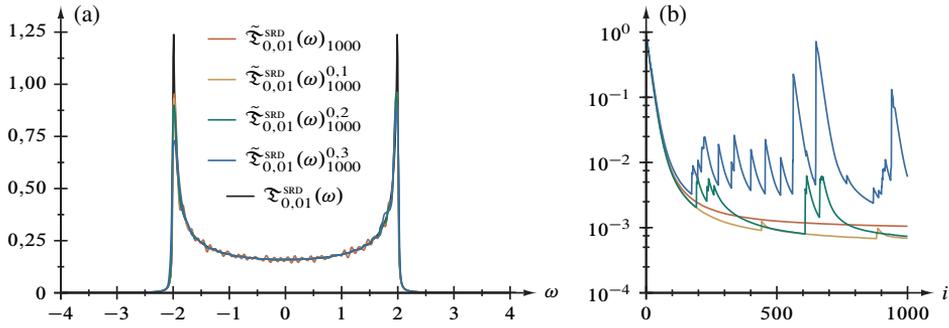


Abbildung 3.21. Entfaltung der künstlich verrauschten (symmetrischen) Quadratwurzeldivergenz, $\tilde{\mathfrak{T}}_\eta^{\text{SRD}}(\omega)$, Gleichung (3.6), 1000 Gitterpunkte im Intervall $\omega \in [-5, +5]$. Zusätzlich ist das Zwischenergebnis nach 441 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).



Legende zu (b):

- R_i/R_0
- $R_i^{0,1}/R_0^{0,1}$
- $R_i^{0,2}/R_0^{0,2}$
- $R_i^{0,3}/R_0^{0,3}$

Legende zu (d):

- $\int_{-5}^{+5} d\omega \zeta^2(\omega)_i$
- $\int_{-5}^{+5} d\omega \zeta^2(\omega)_i^{0,1}$
- $\int_{-5}^{+5} d\omega \zeta^2(\omega)_i^{0,2}$
- $\int_{-5}^{+5} d\omega \zeta^2(\omega)_i^{0,3}$

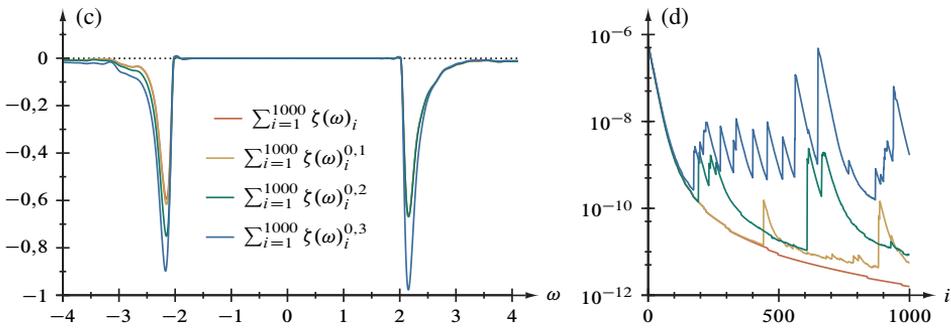


Abbildung 3.22. Entfaltung der Quadratwurzeldivergenz, Abbildung 3.21, mit unterschiedlich starker Rauschunterdrückung.

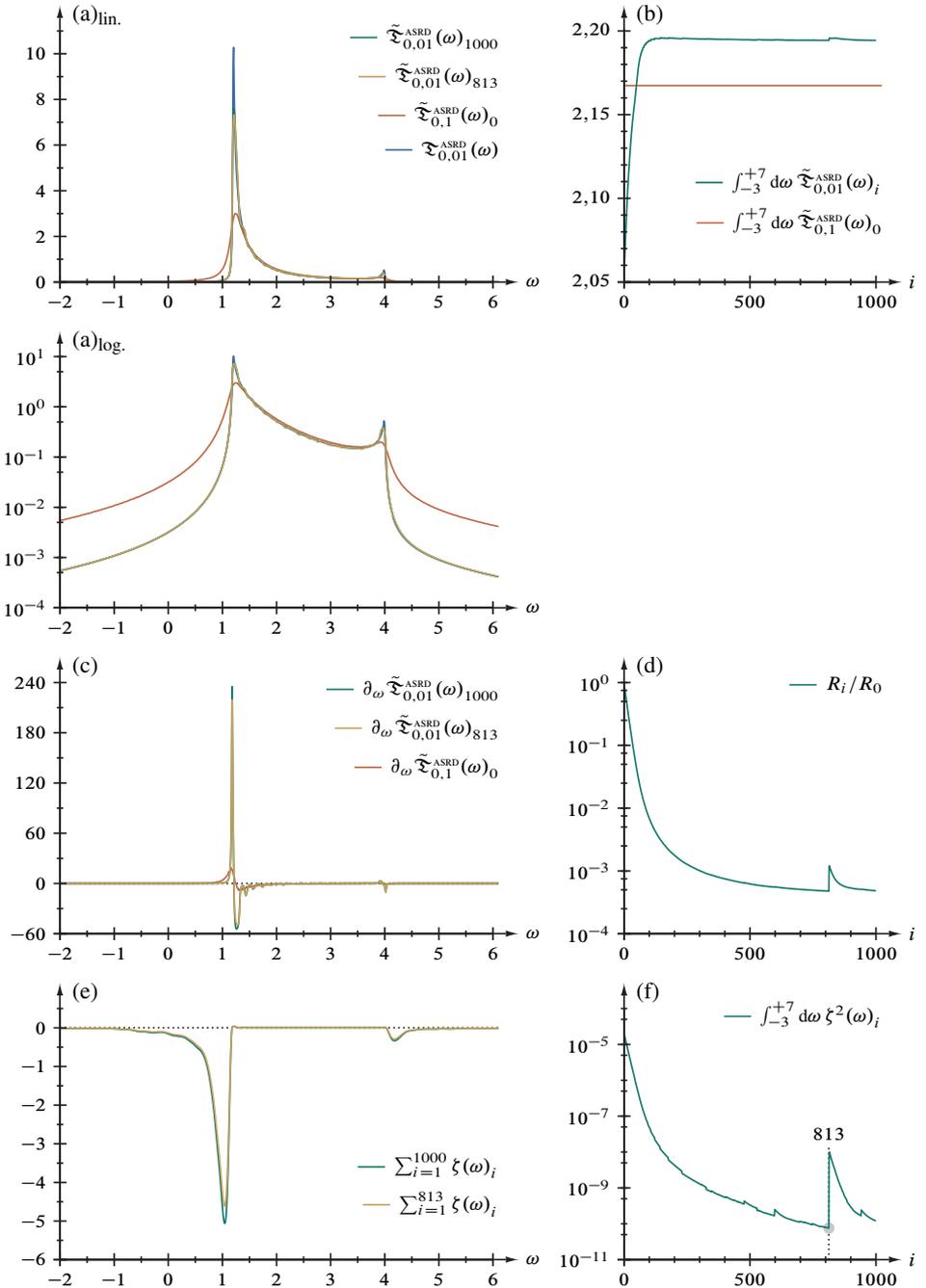


Abbildung 3.23. Entfaltung der künstlich verursachten Asymmetrischen Quadratwurzeldivergenz, $\tilde{\mathfrak{F}}_{\eta}^{\text{ASRD}}(\omega)$, Gleichung (3.7), 1100 Gitterpunkte im Intervall $\omega \in [-3, +8]$. Zusätzlich ist das Zwischenergebnis nach 813 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

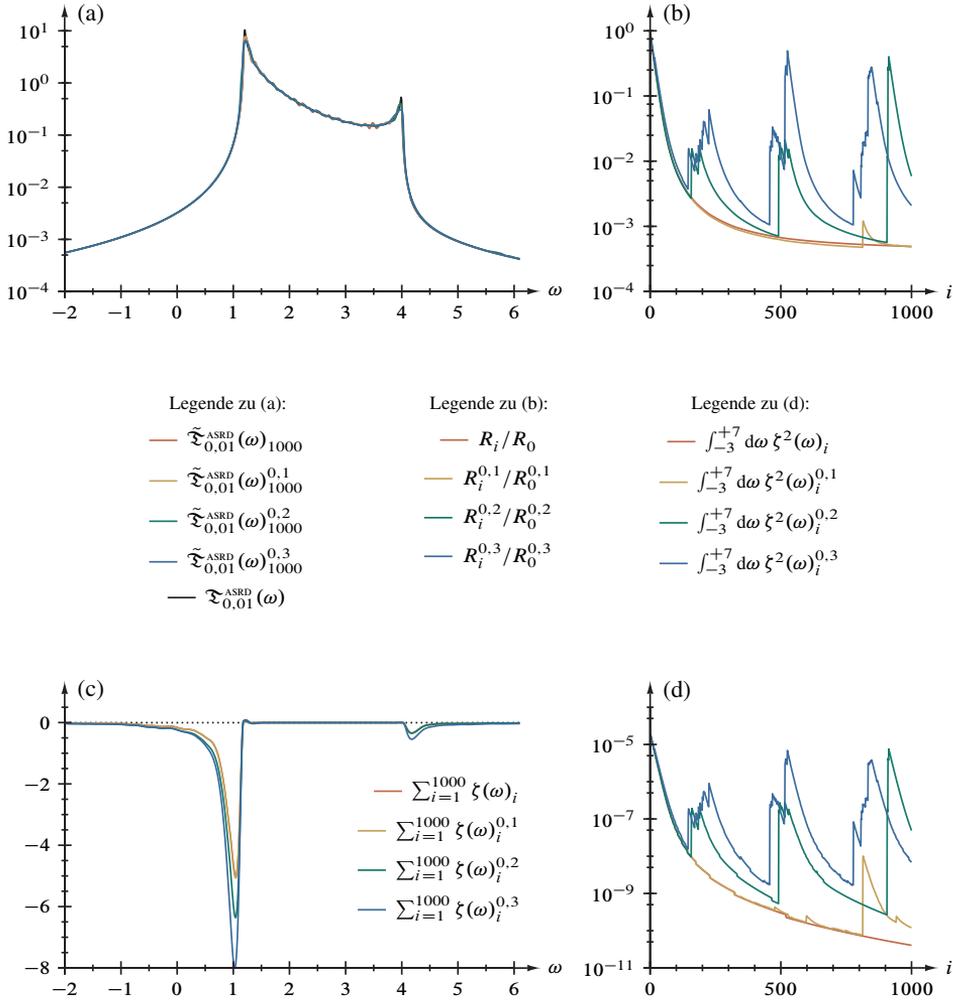


Abbildung 3.24. Entfaltung der Asymmetrischen Quadratwurzeldivergenz, Abbildung 3.23, mit unterschiedlich starker Rauschunterdrückung.

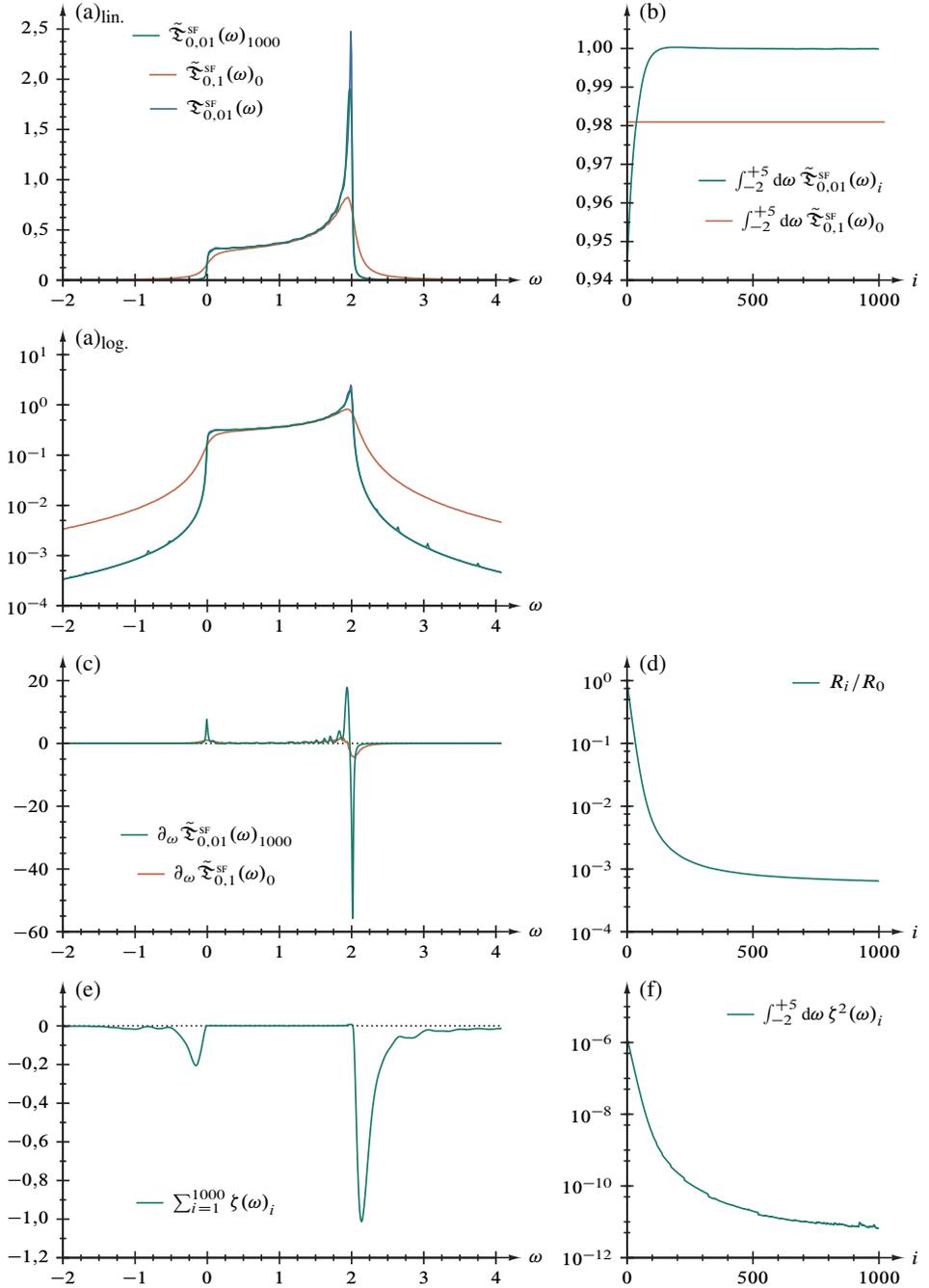


Abbildung 3.25. Entfaltung der künstl. verrauschten Stufenfunktion, $\tilde{\mathfrak{T}}_{\eta}^{\text{SF}}(\omega)$, Gleichung (3.8), 700 Gitterpunkte im Intervall $\omega \in [-2, +5]$.

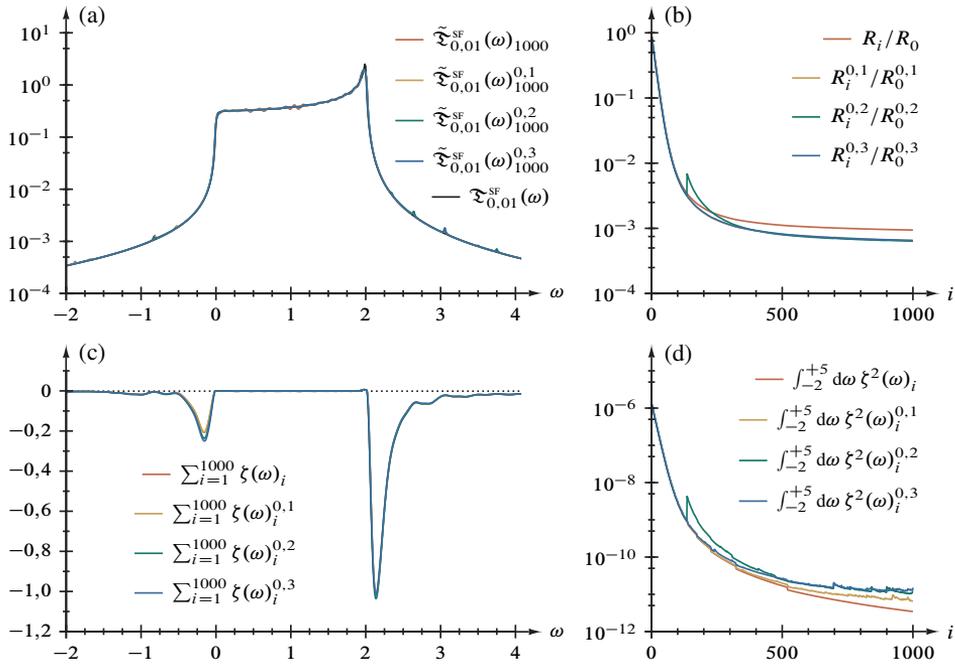


Abbildung 3.26. Entfaltung der Stufenfunktion, Abbildung 3.25, mit unterschiedlich starker Rauschunterdrückung.

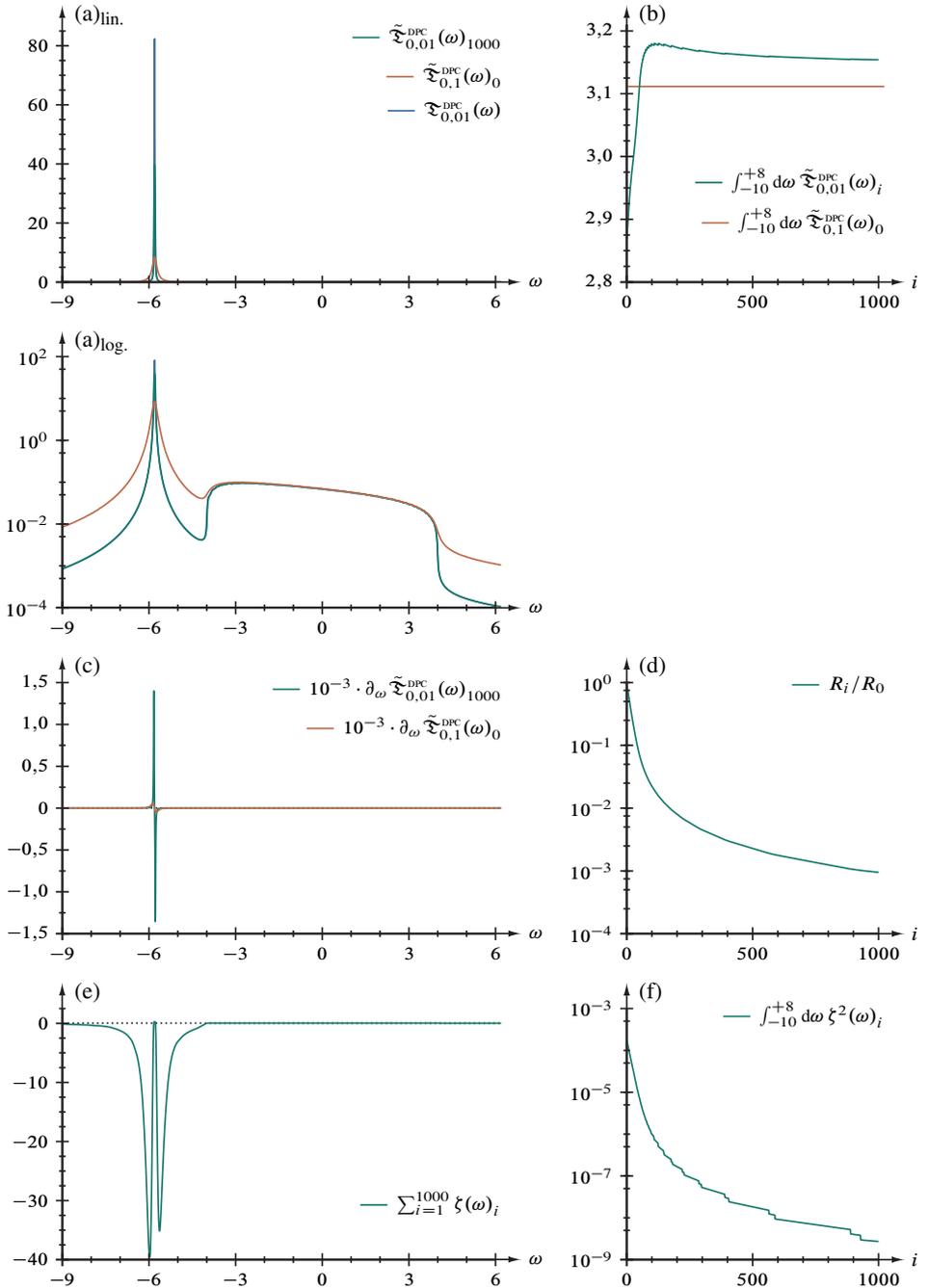


Abbildung 3.27. Entfaltung der künstlich verrauschten Deltafunktion nahe eines Kontinuums, $\tilde{\mathfrak{T}}_{\eta}^{\text{DPC}}(\omega)$, Gleichung (3.9), 1800 Gitterpunkte im Intervall $\omega \in [-10, +8]$.

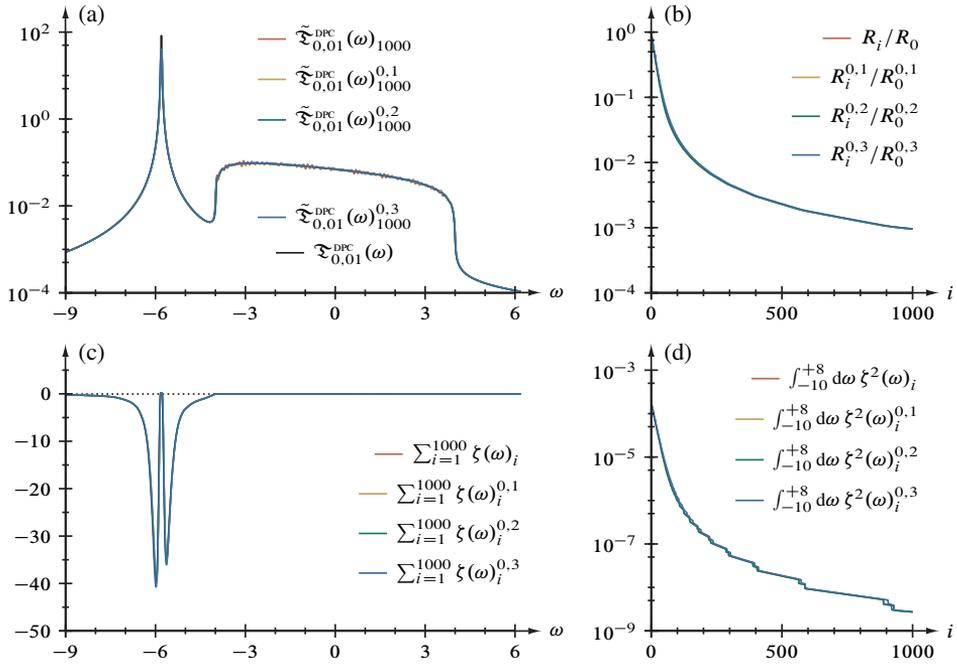


Abbildung 3.28. Entfaltung einer Deltafunktion nahe eines Kontinuums, Abbildung 3.27, mit unterschiedlich starker Rauschunterdrückung.

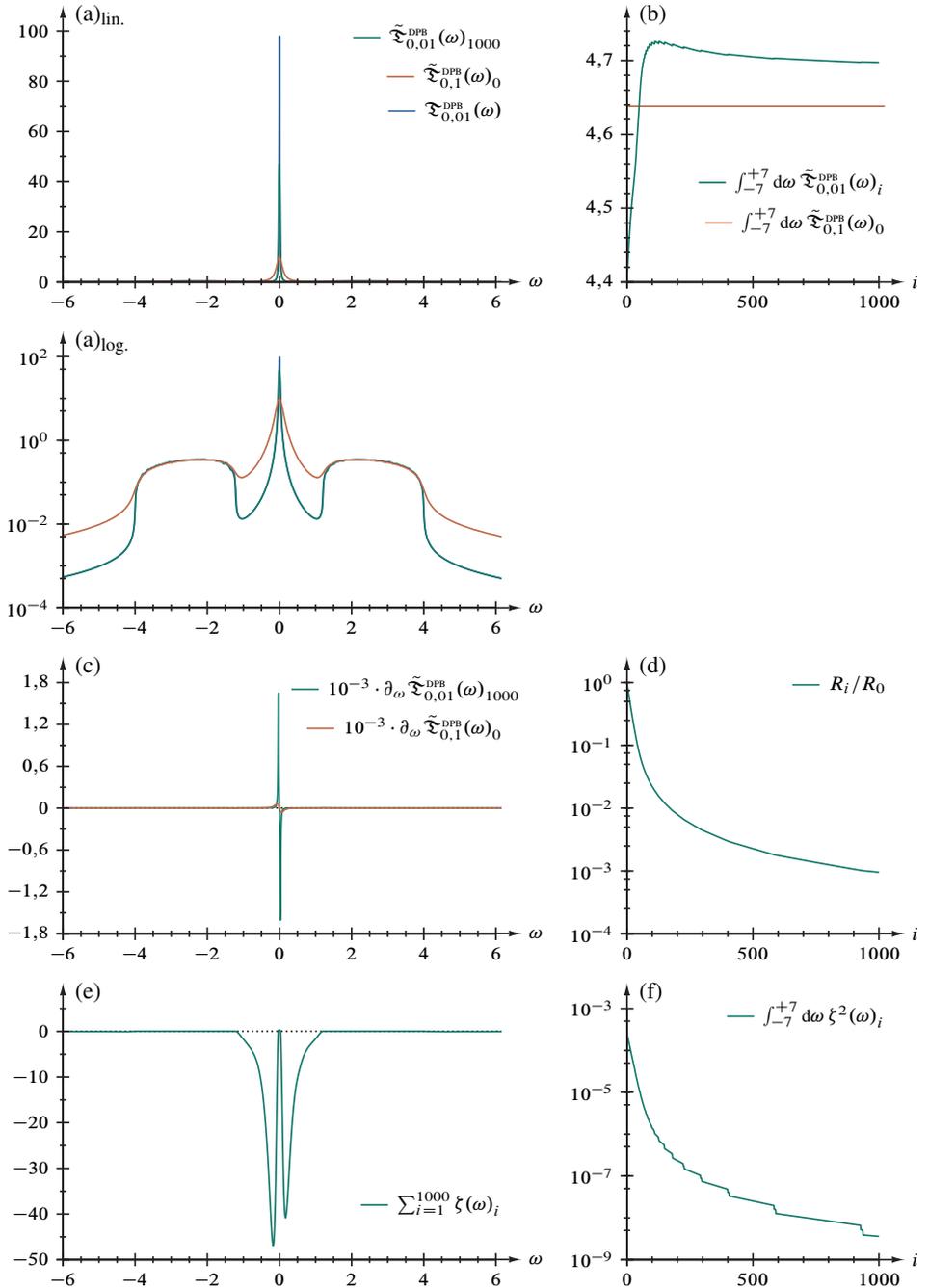


Abbildung 3.29. Entfaltung der künstlich verrauschten Deltafunktion zwischen zwei Bändern, $\tilde{\mathfrak{F}}_{\eta}^{\text{DPB}}(\omega)$, Gleichung (3.10), 1400 Gitterpunkte im Intervall $\omega \in [-7, +7]$.

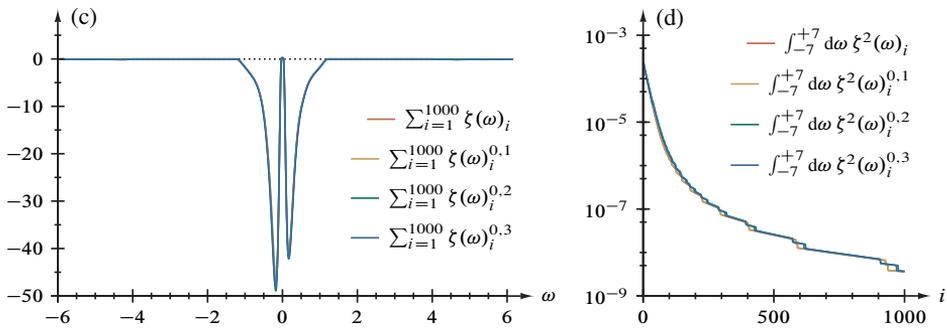
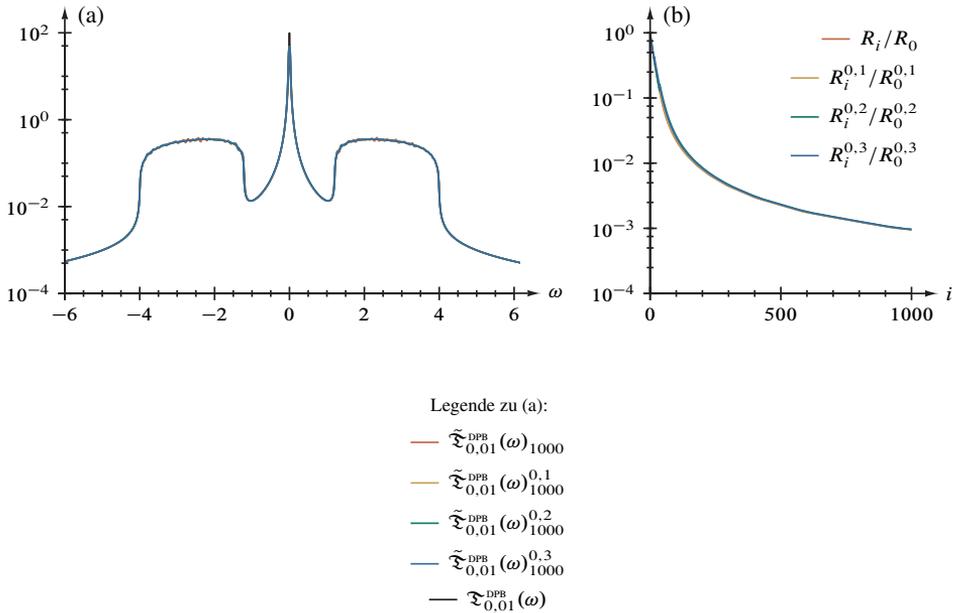


Abbildung 3.30. Entfaltung einer Deltafunktion zwischen zwei Bändern, Abbildung 3.29, mit unterschiedlich starker Rauschunterdrückung.

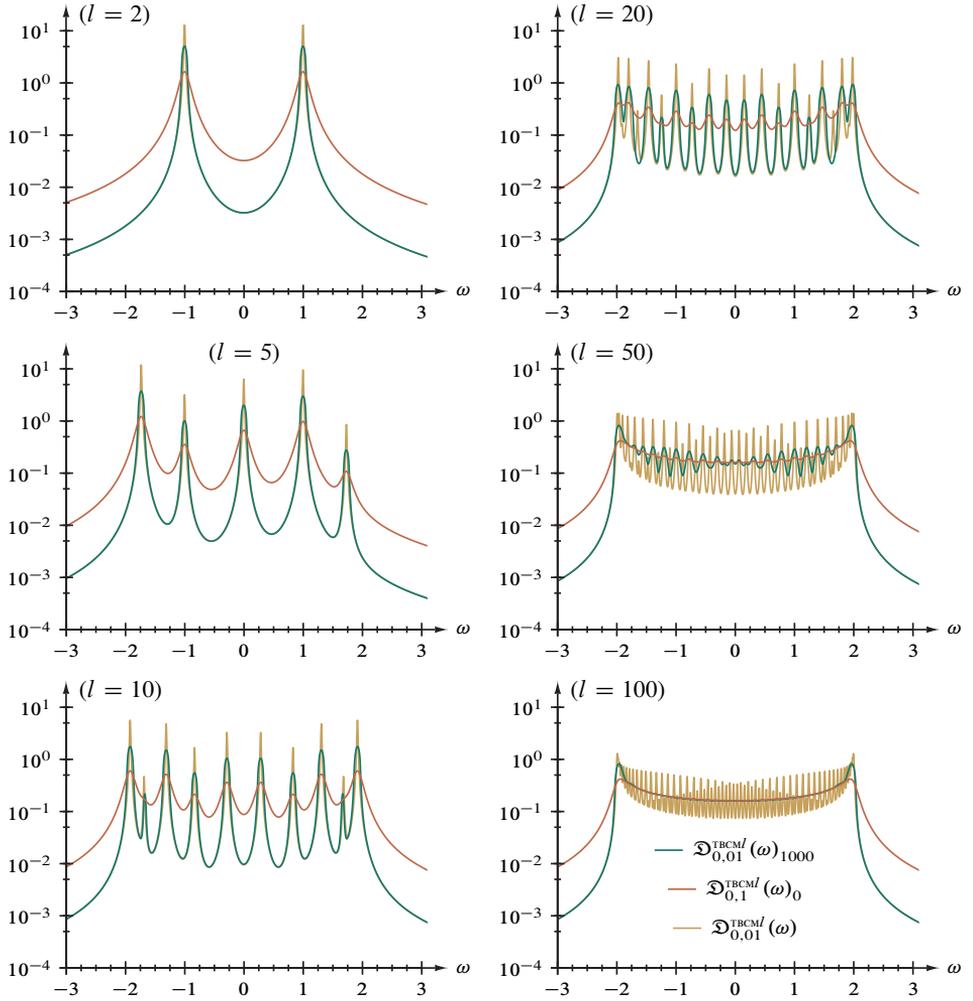


Abbildung 3.31. Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen (Mitte), $\mathcal{D}_{\eta}^{\text{BCM}l}(\omega)$, Gleichung (3.11), 8000 äquidistante Gitterpunkte im Intervall $\omega \in [-4, +4]$.

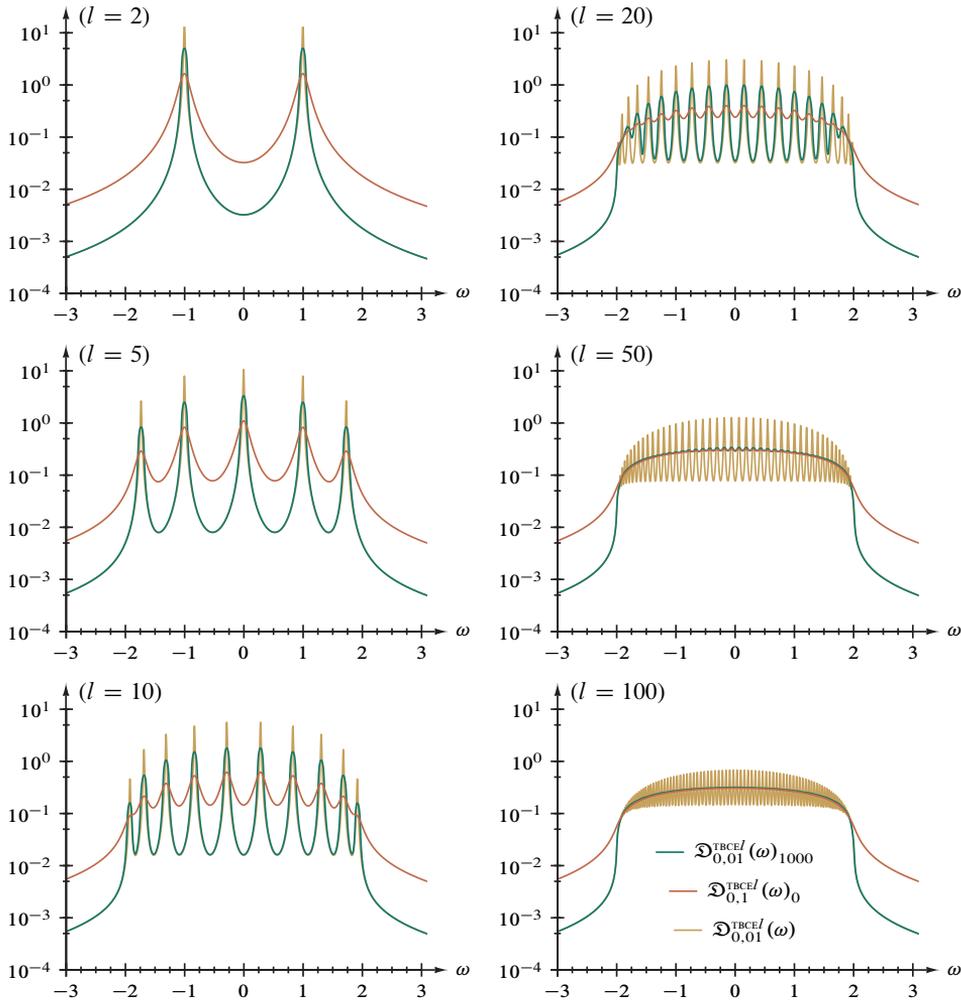


Abbildung 3.32. Entfaltung der Zustandsdichte einer Kette stark gebundener Elektronen (Enden), $\mathcal{D}_\eta^{\text{TBCEL}}(\omega)$, Gleichung (3.12), 8000 äquidistante Gitterpunkte im Intervall $\omega \in [-4, +4]$.

4. Anwendung auf Spektralfunktionen aus numerischen Untersuchungen

Die systematische Untersuchung des Leistungsvermögens der Rechenverfahren zur Faltungsminderung und Rauschdämpfung anhand des Prüffunktionsatzes im vorausgegangenen Kapitel 3 erlaubt eine – subjektiv – günstige Beurteilung: Bei sorgfältiger Wahl des Glättungsparameters, zuweilen ein Kompromiß zwischen verbleibenden Störungen, klarerer Nutzinformation und stabiler Konvergenz, sind die Ergebnisse im Vergleich mit den Referenzdaten mindestens zufriedenstellend – wenn nicht von hoher Qualität. Die neu hinzukommenden Störungen sind mittels der exakten Faltungen kleinerer Breite leicht zu isolieren und lassen sich phänomenologisch in drei Gruppen einteilen, Divergenzartefakte (Abschnitt 3.4 und Abbildung 3.6), Stufenartefakte (Abschnitt 3.6 und Abbildung 3.10) und Bandartefakte (Abschnitt 3.7 und Abbildung 3.12). Damit ist eine Grundlage für die Interpretation von Anwendungen auf DDMRG-Spektren geschaffen, welchen als exakte Lösung eine Testfunktion zugrunde liegt oder welche vergleichbare Merkmale aufweisen.

4.1. Vorbemerkungen.

4.1.1. *Modelle.* Die in den folgenden Abschnitten vorgelegten Ergebnisse beruhen auf DDMRG-Rechnungen am (fermionischen) *eindimensionalen Hubbardmodell* [37, 38],

$$\hat{H} = -t \sum_{i=1}^{N'} \sum_{\sigma=\uparrow,\downarrow} \left(\hat{c}_{i,\sigma}^+ \hat{c}_{i+1,\sigma} + \hat{c}_{i+1,\sigma}^+ \hat{c}_{i,\sigma} \right) + U \sum_{i=1}^N \hat{n}_{i,\uparrow} \hat{n}_{i,\downarrow}, \quad (4.1)$$

einem der elementaren Modelle der Physik stark wechselwirkender Elektronen von hoher Relevanz für Experimente³², sowie am allgemeineren *eindimensionalen Peierls-Hubbard-Modell*,

$$\hat{H} = \hat{T} + U \sum_{i=1}^N \left(\hat{n}_{i,\uparrow} - \frac{1}{2} \right) \left(\hat{n}_{i,\downarrow} - \frac{1}{2} \right) \quad (4.2)$$

$$\text{mit } \hat{T} = - \sum_{i=1}^{N'} \sum_{\sigma=\uparrow,\downarrow} \left(t - (-1)^i \frac{\Delta}{2} \right) \left(\hat{c}_{i,\sigma}^+ \hat{c}_{i+1,\sigma} + \hat{c}_{i+1,\sigma}^+ \hat{c}_{i,\sigma} \right),$$

aus welchem Gleichung (4.1) bei fehlender Dimerisierung ($\Delta = 0$) hervorgeht. Konventionsgemäß bezeichnet in beiden Gleichungen, die für beliebige Bandfüllungen gelten, $t > 0$ den Hüpfparameter (mit der Dimension einer Energie) zwischen zwei benachbarten Gitterplätzen und $\hat{c}_{i,\sigma}^+$ und $\hat{c}_{i,\sigma}$ den Erzeugungs- und den Vernichtungsoperator für ein Elektron mit dem Spin σ auf dem i ten der insgesamt N Gitterplätze; zudem steht $U \geq 0$ für die (lokale) Coulomb-Wechselwirkungsenergie der Elektronen an einem Gitterplatz und $\hat{n}_{i,\sigma} = \hat{c}_{i,\sigma}^+ \hat{c}_{i,\sigma}$ für den Dichteoperator. Zu beachten ist ferner, daß für eine offene Kette $N' = N - 1$ und unter periodischen Randbedingungen $N' = N$ (N dann

³²Siehe dazu beispielsweise die in Referenz [39] zusammengefaßten Arbeiten

geradzahlig) und somit $\hat{c}_{N+1,\sigma} = \hat{c}_{1,\sigma}$ gilt. Mißt man \hat{H} in Einheiten von t , dann bleiben als unabhängig variierbare Parameter U/t und Δ/t ; formal können wir also $t = 1$ setzen. Im Weiteren legen wir $\Delta = 6t/10$ fest, siehe Abschnitt 3.1.2.

Ergänzt wird unsere Untersuchung durch Resultate zu einem *erweiterten Hubbardmodell* [38],

$$\hat{H}_{\text{erw.}} = \hat{H} + V_1 \sum_{i=1}^N (\hat{n}_i - 1)(\hat{n}_{i+1} - 1) + V_2 \sum_{i=1}^N (\hat{n}_i - 1)(\hat{n}_{i+2} - 1), \quad (4.3)$$

welches durch die Potentiale V_1 und V_2 die Abstoßung zwischen nächsten und übernächsten Nachbargitterplätzen mit dem physikalisch relevanten Fall $U > V_1 > V_2 \geq 0$ berücksichtigt. In dieser Arbeit wird die Betrachtung im Gegensatz zu Referenz [36] auf $V_2 = 0$ beschränkt, siehe Abschnitt 4.4.2.

4.1.2. Gliederung der Abbildungen. Sofern zu hier behandelten Spektralfunktionsdaten Ergebnisse aus anderen Quellen vorliegen, sind vergleichende Abbildungen in den jeweiligen Abschnitt eingefügt; Darstellungen der – reproduzierbaren – Faltungsminderungsergebnisse sind aufgrund deren Umfangs hingegen im Abbildungsabschnitt 4.5, beginnend auf Seite 95, zusammengestellt. Wie im Abbildungsteil des vorausgegangenen Kapitels 3 enthalten die Doppelseiten links die verfahrensrelevanten Größen unter Anwendung des Glättungsparameters $\rho = \eta$ und rechts einen Vergleich zwischen unterschiedlichen Werten von ρ sowie der Faltungsminderung ohne Rauschdämpfung. Bezüglich der Rauschfilterung gelten zudem die in Abschnitt 3.1.4 dargelegten Erwägungen.

4.2. Zustandsdichte des eindimensionalen Hubbardmodells. Die Zustandsdichte,

$$D(\omega \leq 0) = \frac{1}{N} \sum_n \sum_{i=1}^N \sum_{\sigma=\uparrow,\downarrow} |\langle \psi_0 | \hat{c}_{i,\sigma} | \psi_n \rangle|^2 \delta(\omega - E_0 + E_n) \quad (4.4-1)$$

und

$$D(\omega \geq 0) = \frac{1}{N} \sum_n \sum_{i=1}^N \sum_{\sigma=\uparrow,\downarrow} |\langle \psi_0 | \hat{c}_{i,\sigma}^+ | \psi_n \rangle|^2 \delta(\omega + E_0 - E_n), \quad (4.4-2)$$

$|\psi_0\rangle$ sind darin Grundzustand und $|\psi_n\rangle$ weitere Eigenzustände von \hat{H} sowie E_0 und E_n die zugehörigen Eigenenergien, wird beispielsweise im Buchbeitrag [12] und eingehend im Artikel [44] behandelt; darin finden sich auch Betrachtungen zum Hintergrund der betreffenden DDMRG-Rechnungen.

4.2.1. Keine Wechselwirkung. Der Grenzfall $U = 0$ ist in dieser Arbeit bereits mehrfach erwähnt worden, beginnend mit Gleichung (1.7) in Abschnitt 1.1.2; die Zustandsdichte in der Kettenmitte weist die Form der Quadratwurzeldivergenz auf, siehe Gleichung (3.6) der entsprechenden Prüffunktion im Abschnitt 3.4.

Die Faltungsminderung der mit mehr als 6000 Datenpunkten vergleichsweise hoch aufgelösten DDMRG-Ergebnisrohdaten (128 Gitterplätze, $\eta = 0,08$), dargestellt in Abbildung 4.9 auf Seite 96, verläuft bis auf einen „Sprung“ der *cost function* nach dem

Schritt 799 ebenso unproblematisch wie jene der zugehörigen analytischen Lösung, siehe Abbildung 3.21 auf Seite 70. Auf eindrucksvolle Weise zeichnet das Entfaltungsverfahren die beiden Divergenzen bei $\omega = \pm 2$ (bis auf die „Restfaltung“ mit $\eta' = 0,008$ und somit minimales Spektralgewicht außerhalb von $|\omega| > 2$) scharf, ohne dabei die endliche Systemgröße hervortreten zu lassen – vergleiche Abschnitt 3.9.1. Wie zu erwarten, tritt symmetriegemäß zweimalig, bei $\omega \approx \pm 1,87$, das Divergenzartefakt auf. Teilabbildung (d) legt Konvergenz nach weniger als 500 Zyklen nahe.

Bereits in der soeben behandelten Hauptabbildung verwendete Rauschdämpfung mit $\rho = \eta = 0,08$ erweist sich als ausreichend wirksam; eine Erhöhung um die Faktoren 2 und 3 führt im Einklang zur gleichen Untersuchung an der Testfunktion, siehe Abbildung 3.22 auf Seite 71, nur zu einer verstärkten Instabilität des Verfahrens, wie insbesondere die Teile (b) und (d) in Abbildung 4.10 auf Seite 97 verdeutlichen. Überdieses leidet, ohne Qualitätsgewinn, die Funktionssymmetrie, wie Teil (a) an den Ergebnisdaten und Teil (c) an der Summe der Korrekturen zeigt.

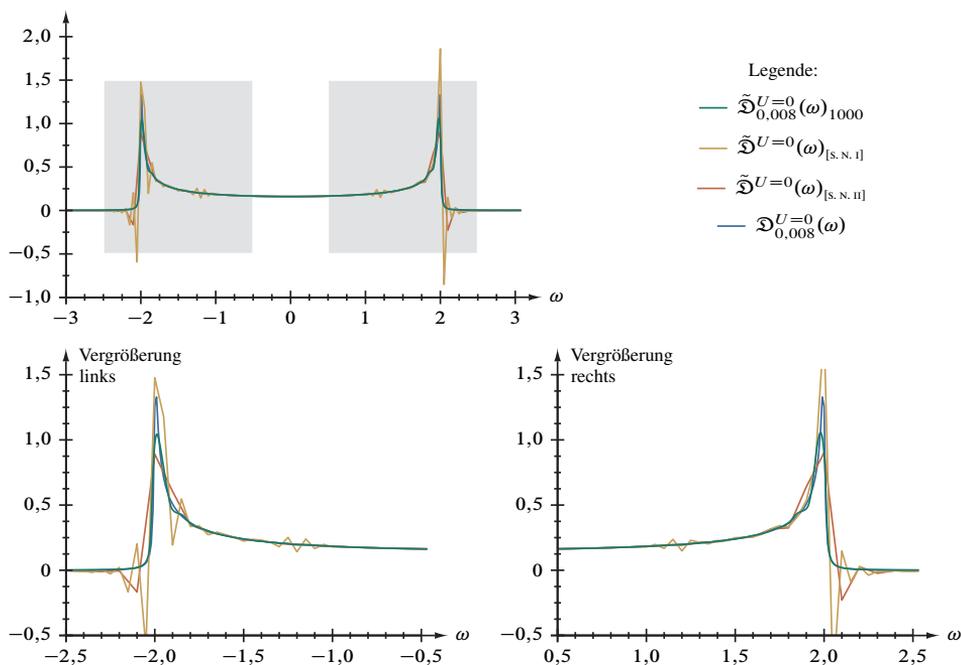


Abbildung 4.1. Vergleich verschiedener Entfaltungsverfahren anhand der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathcal{D}}_{0,08}^{U=0}(\omega)$.

Für den wechselwirkungsfreien Fall liegen auch Ergebnisse aus zwei Varianten eines anderen Entfaltungsverfahrens auf Grundlage linearer Regularisierung vor, das von SATOSHI NISHIMOTO entwickelt wurde [21]; Abbildung 4.1 kombiniert diese mit dem

Resultat hiesiger Methode und der entsprechend gefalteten Referenz $\mathfrak{D}_{0,008}^{U=0}(\omega)$. Deutlich belegbar ist insbesondere durch die Ausschnittsvergrößerungen, daß die Datensätze $\tilde{\mathfrak{D}}^{U=0}(\omega)_{[\text{s. n. I}]}$ und $\tilde{\mathfrak{D}}^{U=0}(\omega)_{[\text{s. n. II}]}$, ohne Berücksichtigung der geringen Auflösung von $\Delta\omega = 0,1$ respektive $\Delta\omega = 0,05$, symmetrisch starke Oszillationen und negatives Spektralgewicht aufweisen – sowohl im Frequenzbereich $1,7 \lesssim |\omega| \lesssim 2,3$ um die Quadratwurzeldivergenzen, als auch bei ersterem Ergebnis im Band bei $1,0 \lesssim |\omega| \lesssim 1,4$. Während im mit der ersten Variante entfalteten Spektrum die Rekonstruktionen der unteren und der oberen Divergenz deutlich voneinander abweichen, vermag die zweite Variante die Spiegelsymmetrie zu erhalten, jedoch unter deutlich höherer Restfaltung. Abgesehen vom Divergenzartefakt belegt somit das Entfaltungsergebnis $\tilde{\mathfrak{D}}_{0,008}^{U=0}(\omega)_{1000}$ ($\Delta\omega = 0,004 = \eta/20$) beeindruckend die Vorzüge der neuen Methode.

4.2.2. *Endliche Wechselwirkung.* Die Zustandsdichte in der Kettenmitte bei endlicher Elektronenkorrelation behandeln wir zunächst für den Wert $U = 4$; stärkere Wechselwirkung wird kurz im Abschnitt 2 des Anhangs betrachtet. Eine wesentliche Frage ist in diesem Zusammenhang die Form von $D_{U>0}(\omega)$ unterhalb der unteren und oberhalb der oberen Kante (*on-set*) der *optischen Bandlücke*³³ der Breite E_{opt} bei kleinen Werten von U . Während die Feldtheorie [40] für $\omega \geq E_{\text{opt}}/2$ mit einer Konstanten k

$$D(\omega) = \Theta(\omega - E_{\text{opt}}/2) \cdot k \quad \rightarrow \quad \text{---} \begin{array}{|c} \hline \\ \hline \end{array}, \quad (4.5)$$

also ein stufenfunktionsartiges Verhalten, siehe Abschnitt 3.6, vorhersagt, behauptet die Störungsrechnung im Limes starker Wechselwirkung [41] mit einer Funktion, für die $f'(\omega) < 0$ gilt,

$$D(\omega) = \Theta(\omega - E_{\text{opt}}/2) \cdot f(\omega) \quad \rightarrow \quad \text{---} \begin{array}{|c} \hline \\ \hline \\ \hline \end{array}, \quad (4.6)$$

folglich eine Quadratwurzeldivergenz, vergleiche Abschnitt 3.4 und 3.5; Θ bezeichnet jeweils die Heavysidefunktion.

Abbildung 4.11 auf Seite 98 präsentiert die Entfaltung des DDMRG-Rohdatensatzes (128 Gitterplätze, $\eta = 0,08$), welche laut Teil (f) nach etwa 300 Zyklen in den Konvergenzbereich eintritt und nach Schritt 349 stärkere Sprünge aufweist. Abbildungsteil (a) ist zu entnehmen, daß die weiteren Iterationen dennoch eine (wenn auch bloß marginale) Verbesserung hinsichtlich der Höhe der Extrema mit sich bringt, beispielsweise von $\tilde{\mathfrak{D}}_{0,08}^{U=4}(\omega)_0 \approx 0,278$ über $\tilde{\mathfrak{D}}_{0,008}^{U=4}(\omega)_{349} \approx 0,391$ nach $\tilde{\mathfrak{D}}_{0,008}^{U=4}(\omega)_{1000} \approx 0,406$, jeweils bei $\omega \approx 1,89$. Teil (e) zeigt die größten Änderungen bei den Bandkanten, die jedoch nicht vollständig symmetrisch verläuft, wiewohl strukturell im Ergebnis kaum ein Unterschied besteht; bestimmt aus der jeweiligen ersten Ableitung, $\partial_\omega \tilde{\mathfrak{D}}_\eta^{U=4}(\omega)_i$, Teil (c) ergeben sich zum Beispiel jeweils $\omega = E_{\text{opt}}/2 \approx 0,65$ für $i = 0$ (Ausgangsdaten, $\eta = 0,08$), für $i = 349$ und für $i = 1000$ ($\eta = 0,008$). Diesem Wert für die halbe Bandlücke steht, berechnet mit dem Betheansatz [38], $E_{\text{opt}}/2 = 0,643$ gegenüber.

³³Da hier (wie später nochmals in Abschnitt 4.4.1) die Mottsche Isolatorphase anhand des Hubbardmodells betrachtet wird, können wir – was im Allgemeinen nicht zutreffend ist – die *Einteilchen-* beziehungsweise *Mott-Bandlücke* mit der optischen Bandlücke identifizieren.

Auffällig am bisher diskutierten Entfaltungsergebnis sind die – in einigen Abschnitten, etwa nahe der Bandkanten bei $0,65 \lesssim |\omega| \lesssim 1,5$ und bei $3,0 \lesssim |\omega| \lesssim 4,5$ beträchtlichen – Störungen. Eine zweistufige vorsichtige Erhöhung des Rauschfilterwertes von $\rho = \eta = 0,08$ auf $\rho = 3/2\eta = 0,12$ und $\rho = 2\eta = 0,16$ liegt Abbildung 4.12 auf Seite 99 zugrunde. Hier zeigt sich insbesondere an Teil (d) durch heftige Sprünge die Unverträglichkeit stärkerer Rauschdämpfung für den Minimierungsalgorithmus der *cost function* an diesem Spektrum und somit eine zentrale Schwäche des Verfahrens; Abbildungsteil (a) weist dementsprechend nur eine geringfügige Verbesserung des Ergebnis' auf.

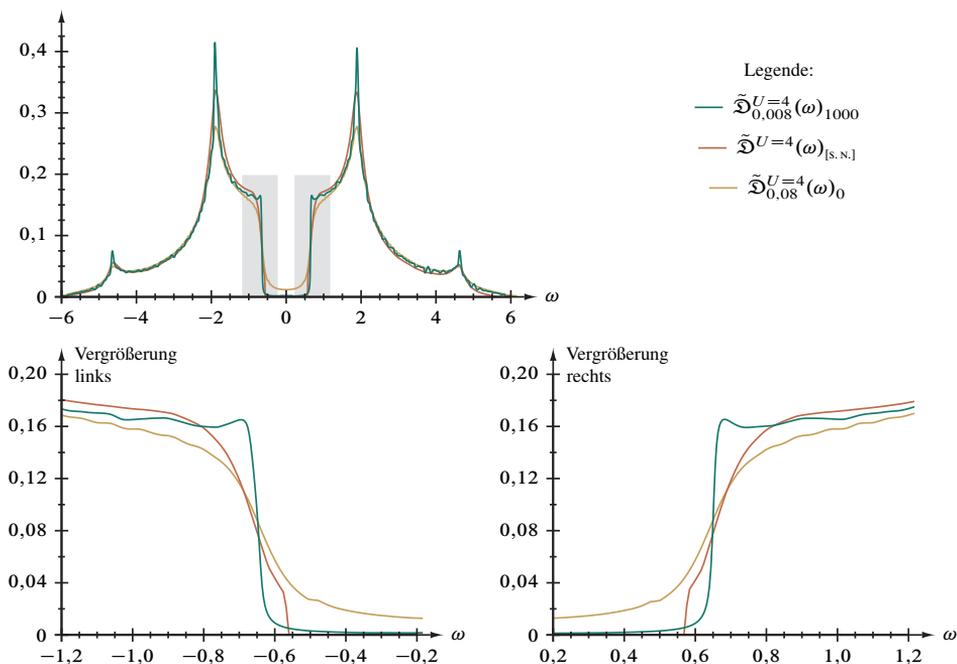


Abbildung 4.2. Vergleich verschiedener Entfaltungsverfahren anhand der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathcal{D}}_{0,08}^{U=4}(\omega)$.

Auch für den Fall endlicher Wechselwirkung mit $U = 4$ liegt das Ergebnis des bereits erwähnten älteren Entfaltungsverfahrens [5, 21, 26] vor, welches mit dem hiesigen in Abbildung 4.2 zusammengefasst ist. Eindeutig erweist sich der Fortschritt durch die in dieser Arbeit vorgestellte Methode: Die Bandkanten werden wesentlich deutlicher scharfgezeichnet, und obzwar das Resultat durch das Stufenartefakt beeinträchtigt wird, läßt sich die Kantenform zweifelsfrei dem Szenario nach Gleichung (4.5) zuordnen³⁴.

³⁴Wie im Abschnitt 2 des Anhanges kurz behandelt wird, bestätigt die Gaußrückfaltung diese Zuordnung noch weitaus klarer.

4.3. Zustandsdichte einer Luttingerflüssigkeit. Zur besonderen Physik eindimensionaler Materialien gehört, daß sich entsprechende Leiter nicht, wie gewöhnliche Metalle, als Fermiflüssigkeit verstehen lassen, sondern als Luttingerflüssigkeit beschrieben werden müssen [14, 15]. Diese weisen etwa, im Gegensatz zu Fermiflüssigkeiten, an der Fermienergie E_F statt einer endlichen Zustandsdichte eine potenzgesetzartige Pseudolücke auf,

$$D(\omega) \sim |\hbar\omega - E_F|^\alpha \quad \dashrightarrow \quad \bigvee, \quad (4.7)$$

worin $1 > \alpha > 0$. Weil allerdings in den noch gefalteten Ergebnissen entsprechender DDMRG-Rechnungen am Hubbardmodell (90 Gitterplätze) ein solches, für Luttingerflüssigkeiten charakteristisches Verhalten nicht eindeutig zu identifizieren ist, besteht hier herausragendes Interesse an der Anwendung des beschriebenen Entfaltungsverfahrens. Die im Folgenden behandelten Daten wurden jenseits der Halbfüllung für die mittlere elektronische Dichte $n = 0,6$ ermittelt, woraus sich $0 < \alpha < 1/8$ ergeben sollte.

4.3.1. Photoemission. Durch Gleichung (4.4-1) ist die Zustandsdichte der *Photoemission* gegeben; Abbildung 4.13 auf Seite 100 zeigt das zugehörige DDMRG-Ergebnis sowie dessen Entfaltung. Wie aus den Teilen (d) und (f) hervorgeht und ansatzweise auch im „Protokoll“ des Spektralen Gewichts in Teil (b) zu erkennen ist, erreicht die kombinierte Minimierungs- und Glättungsprozedur nach etwa 150 Zyklen den Konvergenzbereich und geht unmittelbar in ein stark ausgeprägtes Sägezahnprofil über. Daß bereits beim ersten Sprung nach Schritt 311 weitestgehende Faltungsminderung erreicht ist, zeigt der Vergleich mit dem ebenfalls eingezeichneten Ergebnis nach den vollen 1000 Iterationen: Das Hauptmaximum erhöht sich von $\tilde{\mathcal{D}}_{0,01}^{\text{PE}}(\omega)_{311} \approx 0,487$ bei $\omega \approx -0,730$ nur noch auf $\tilde{\mathcal{D}}_{0,01}^{\text{PE}}(\omega)_{1000} \approx 0,496$ bei $\omega \approx -0,735$, wobei der Anfangswert bei $\omega \approx -0,720$ den Wert $\tilde{\mathcal{D}}_{0,1}^{\text{PE}}(\omega)_0 \approx 0,277$ aufwies. Bei genauerer Betrachtung ist auch in diesem Beispiel zu erkennen, daß mit zunehmender „Verschärfung“ der wesentlichen Funktionsmerkmale im Verfahrensverlauf auch die Artefakte stärker hervortreten, da ein Extremumsabstand erreicht wird, den der Glättungsalgorithmus nicht mehr behandelt. Wird der für die Rauschdämpfung zuständige Parameter $\rho = \eta = 0,1$ verdoppelt, siehe Abbildung 4.14 auf Seite 101, ist der geringe subjektive Qualitätsgewinn mit einer Abstumpfung und Verschiebung der beiden Extrema erkauft, etwa für das Hauptmaximum auf $\tilde{\mathcal{D}}_{0,01}^{\text{PE}}(\omega)_{1000}^{0,2} \approx 0,464$ bei $\omega \approx -0,760$.

Eine Detailvergrößerung des Spektrums liegt in Abbildung 4.3 vor, worin deutlich zu erkennen ist, daß das Entfaltungsergebnis ebenso wenig wie das unbehandelte DDMRG-Ergebnis das in Gleichung (4.7) formulierte Potenzgesetz an der Fermienergie $E_F \approx 0$ bestätigen kann. Damit ist ein in diesem Zusammenhang zentrales Ziel der Verfahrensentwicklung noch nicht erreicht, wobei möglicherweise für diese Anwendung günstige Programmodifikationen, vornehmlich die Gauß-Rückfaltung, im Anhang kurz behandelt werden.

4.3.2. Inverse Photoemission. Gleichung (4.4-2) beschreibt die Zustandsdichte der *inversen Photoemission*, deren DDMRG-Ergebnis und dessen Entfaltung in Abbildung 4.15

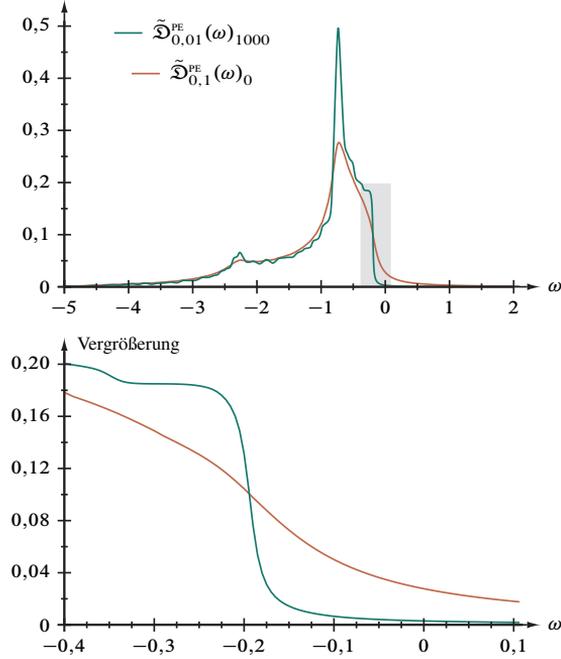


Abbildung 4.3. Ausschnitt aus der entfalteten Zustandsdichte einer Luttinger-
flüssigkeit bei $U = 4,9$ im Bereich der Photoemission (PE), $\tilde{\mathcal{D}}_{0,1}^{\text{PE}}(\omega)$.

auf Seite 102 dargestellt sind. Bemerkenswerterweise verläuft das Verfahren, siehe insbesondere Teil (f), weitgehend ohne jene Komplikationen, die beim Photoemissionsspektrum im vorausgegangenen Abschnitt 4.3.1 aufgetreten sind, und entsprechend ist das Endresultat nach 1000 Schritten und jenem vor den einzigen Sprung bei $i = 922$ ununterscheidbar. Gut zu erkennen ist im Entfaltungsergebnis die Auflösung des Hauptmaximums $\tilde{\mathcal{D}}_{0,1}^{\text{IPE}}(\omega)_0 \approx 0,221$ bei $\omega \approx 1,930$ in eine Doppelspitze, $\tilde{\mathcal{D}}_{0,01}^{\text{IPE}}(\omega_1)_{1000} \approx 0,284$ bei $\omega_1 \approx 2,070$ und $\tilde{\mathcal{D}}_{0,01}^{\text{IPE}}(\omega_2)_{1000} \approx 0,283$ bei $\omega_2 \approx 2,470$, einem Zwischenminimum $\tilde{\mathcal{D}}_{0,01}^{\text{IPE}}(\omega_3)_{1000} \approx 0,259$ bei $\omega_3 \approx 2,220$. Unter Verdoppelung des Rauschdämpfungsparameters auf $\rho = 2\eta = 0,2$ verschlechtert sich das Konvergenzverhalten drastisch, siehe Abbildung 4.16 auf Seite 103, besonders Teil (d). Dennoch erscheint das Spektrum glatter, wobei sich die Doppelspitze bloß unmerklich verschiebt und „abflacht“: $\tilde{\mathcal{D}}_{0,01}^{\text{IPE}}(\omega_1)_{1000} \approx 0,283$ bei $\omega_1 \approx 2,050$, $\tilde{\mathcal{D}}_{0,01}^{\text{IPE}}(\omega_2)_{1000} \approx 0,280$ bei $\omega_2 \approx 2,465$ und $\tilde{\mathcal{D}}_{0,01}^{\text{IPE}}(\omega_3)_{1000} \approx 0,260$ bei $\omega_3 \approx 2,230$.

Abbildung 4.4 zeigt jenen Ausschnitt des Spektrums um die Fermienergie $E_F \approx 0$ vergrößert, welcher die Vorhersage des Potenzgesetzes (4.7) zeigen sollte. Wie bereits im Falle des Photoemissionsspektrums findet sich ein entsprechender Funktionsverlauf

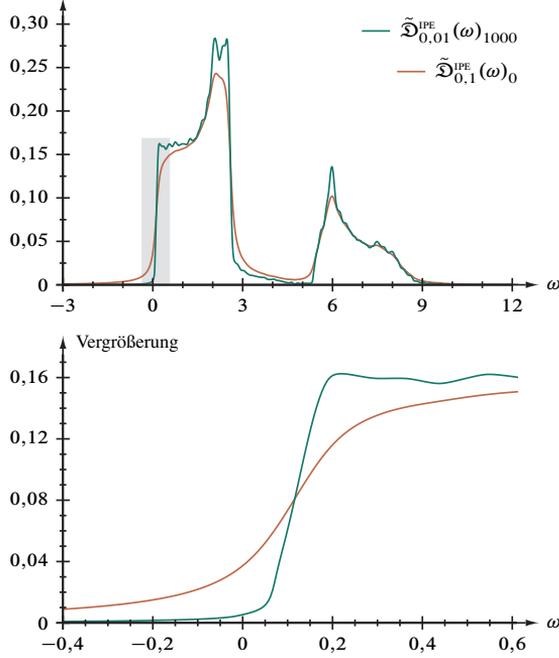


Abbildung 4.4. Ausschnitt aus der entfalteten Zustandsdichte einer Luttinger-Flüssigkeit bei $U = 4,9$ im Bereich der inversen Photoemission (IPE), $\tilde{\mathcal{D}}_{0,1}^{\text{IPE}}(\omega)$.

nicht und bleibt Gegenstand künftiger Verfeinerungen der am Entfaltungsverfahren beteiligten Algorithmen.

4.4. Optische Leitfähigkeit im Hubbard- und Peierls-Hubbard-Modell. Die *optische Leitfähigkeit*,

$$\sigma_1(\omega) = \frac{\pi}{Na\omega} \sum_{n>1} |\langle \psi_0 | \hat{J} | \psi_n \rangle|^2 \delta(\omega + E_0 - E_n), \quad (4.8)$$

a sind darin die Gitterkonstante, $-e$ die Elektronenladung und \hat{J} beispielsweise der Peierls-Hubbard-Stromoperator

$$\hat{J} = \frac{iae}{\hbar} \sum_{i=1}^{N'} \sum_{\sigma=\uparrow,\downarrow} \left(t - (-1)^i \frac{\Delta}{2} \right) \left(\hat{c}_{i,\sigma}^+ \hat{c}_{i+1,\sigma} - \hat{c}_{i+1,\sigma}^+ \hat{c}_{i,\sigma} \right) \quad (4.9)$$

bei zu Abschnitt 4.1.1 identischen Randbedingungen, ist Gegenstand einer umfassenden Studie im Zeitschriftenbeitrag [11], in welcher der physikalische Hintergrund sowie die

DDMRG-Technik betreffende Aspekte ausführlich dargelegt werden. Sämtliche Angaben in den folgenden Abschnitten sind daher dieser Arbeit entnommen.

4.4.1. *Optische Leitfähigkeit des Mott-Hubbard-Isolators.* Das Hubbardmodell, Gleichung (4.1), erfaßt für $U > 0$ die Eigenschaften des *Mott-Hubbard-Isolators*, dessen optische Leitfähigkeit die Form

$$\sigma_1(\omega) \sim \sqrt{\hbar\omega - E_{\text{opt.}}(U)} \quad (4.10)$$

besitzt. Über die exakte Lösung des Hubbardmodells durch den Betheansatz ist auch die optische Lücke analytisch zugänglich [10] ($\hbar = 1$),

$$E_{\text{opt.}}(U) = \frac{(4t)^2}{U} \int_1^{\infty} dy \frac{\sqrt{y^2 - 1}}{\sinh(2\pi ty/U)}; \quad (4.11)$$

das aus Referenz [11] vorliegende DDMRG-Ergebnis (128 Gitterplätze) wurde für $U = 3t$ berechnet, woraus sich nach vorgenannter Gleichung (4.11) mit $t = 1$ daher $E_{\text{opt.}} \approx 0,63139$ ergibt.

Die detaillierte Darstellung des Verlaufes der Faltungsminderung findet sich Abbildung 4.17 auf Seite 104, worin zunächst Teil (f) eine markante Unterbrechung der Konvergenz nach $i = 826$ Entfaltungsschritten aufweist. Tatsächlich tritt für das Funktionsmaximum, siehe Teil (a), kaum noch eine Änderung ein: Von $\tilde{\mathfrak{D}}_{0,1}^{\text{HMOC}}(\omega)_0 \approx 0,998$ bei $\omega \approx 0,890$ ausgehend wird $\tilde{\mathfrak{D}}_{0,01}^{\text{HMOC}}(\omega)_{826} \approx 1,282$ bei $\omega \approx 0,825$ erreicht; die verbleibenden Zyklen ändern die Lage auf $\tilde{\mathfrak{D}}_{0,01}^{\text{HMOC}}(\omega)_{1000} \approx 1,287$ bei gleicher Frequenz. Die obere Kante der optischen Lücke (Niedrigenergieschwellwert) kann aus dem Maximum der ersten Ableitung, $\partial_{\omega} \tilde{\mathfrak{D}}_{\eta}^{\text{HMOC}}(\omega)_i$, siehe Teil (b), bestimmt werden, und – genauigkeitsbeschränkt durch die Auflösung des Frequenzrasters – ergibt sich $E_{\text{opt.}} \approx 0,660$ für $i = 0$ (Ausgangsdaten, $\eta = 0,1$) und $E_{\text{opt.}} \approx 0,650$ für $i = 826$ und $i = 1000$ ($\eta = 0,01$).

Wird die Entfaltung mit höheren Rauschfilterwerten ρ durchgeführt, ist das Konvergenzverhalten deutlich schlechter, wie Abbildung 4.18 auf Seite 105 für $\rho = 2\eta = 0,2$ und für $\rho = 3\eta = 0,3$ in den Teilen (b) und insbesondere (d) belegt. Dennoch scheint die Funktion selbst „robust“; nach vollen 1000 Entfaltungsschritten ergeben sich das Maximum $\tilde{\mathfrak{D}}_{0,01}^{\text{HMOC}}(\omega)_{1000}^{0,2} \approx 1,288$ bei $\omega \approx 0,820$ und $\tilde{\mathfrak{D}}_{0,01}^{\text{HMOC}}(\omega)_{1000}^{0,3} \approx 1,292$ bei $\omega \approx 0,810$ sowie die Bandkante $E_{\text{opt.}} \approx 0,645$ für $\rho = 0,2$ und $E_{\text{opt.}} \approx 0,620$ für $\rho = 0,3$. Es läßt sich also schließen, daß eine stärkere Rauschfilterung in diesem Falle zu Ergebnissen führt, die – nur durch die Frequenzrasterweite begrenzt – nochmals deutlich näher an der exakten Lösung (4.11) liegen.

Während Referenz [11] unter anderem die noch gefalteten DDMRG-Rohdaten zeigt ($\eta = 0,1$) und vergleichshalber mit dem feldtheoretisch berechneten [10] Spektrum überlagert (dort Figur 3), werden nun in Abbildung 4.5 die mit dem hier entwickelten Verfahren auf $\eta' = 0,01$ faltungsgeminderte Daten zusammen mit dem Feldtheorie-Ergebnis und dem Resultat [5] des zuvor von SATOSHI NISHIMOTO implementierten Entfaltungsansatzes [21] dargestellt. Für die charakteristischen Merkmale der optischen

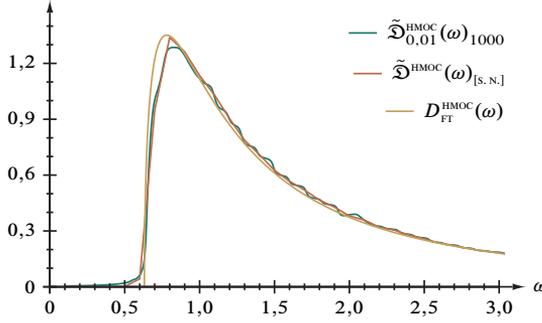


Abbildung 4.5. Vergleich verschiedener Entfaltungsverfahren anhand der optischen Leitfähigkeit im eindimensionalen Hubbardmodell, $\tilde{\mathfrak{D}}_{0,1}^{\text{HMOC}}(\omega)$, bei $U = 3$.

Leitfähigkeit ergeben sich folgende Werte: $D_{\text{FT}}^{\text{HMOC}}(\omega) \approx 1,351$ bei $\omega \approx 0,780$ und $\tilde{\mathfrak{D}}^{\text{HMOC}}(\omega)_{[\text{s. N.}]} \approx 1,340$ bei $\omega \approx 0,8$ für das Maximum, für die optische Lücke aus der Feldtheorie erwartungsgemäß $E_{\text{opt.}} \approx 0,632$ und aus dem älteren Entfaltungsfahren $E_{\text{opt.}} \approx 0,6$. Eine genauere Vergleich der beiden Entfaltungsmethoden ist aufgrund der Datenlage (Frequenzauflösung des älteren Ergebnisses bei $\Delta\omega = \eta = 0,1$) nicht möglich; qualitativ erscheint jedoch aufgrund ähnlicherer Maximumshöhe die Übereinstimmung des Resultats aus Referenz [5] mit dem feldtheoretischen Spektrum besser.

4.4.2. *Strom-Strom-Korrelationsfunktion im Hubbardmodell.* Im Thermodynamischen Limes ist die *Strom-Strom-Korrelationsfunktion* aus der DDMRG-Rechnung durch

$$\text{Im}(\chi(\omega)) = \lim_{\eta \rightarrow 0} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_n \frac{\eta |\langle \psi_0 | \hat{J} | \psi_n \rangle|^2}{(\omega - (E_n - E_0))^2 + \eta^2} \quad (4.12)$$

gegeben [36], worin \hat{J} für den Stromoperator steht.

Für das erweiterte Hubbardmodell, Gleichung (4.3), im Sonderfall $V_1 = 5$ und $V_2 = 0$ betrachten wir nun in Abbildung 4.19 auf Seite 106 das entfaltete DDMRG-Ergebnis (Rohdatenentfaltung mit $\eta = 0,1$, $L = 128$ Gitterplätze), wobei die Teile (b), (d) und (f) auf einen bezüglich der Konvergenz unproblematischen Verfahrensablauf schließen lassen – wie diesen in fast quantitativer Übereinstimmung auch die Testfunktion in Abbildung 3.27 auf Seite 76 aufweist. In Teil (a), besser noch in (a)_{log}, sind die wesentlichen Funktionsmerkmale zu erkennen, nämlich das Hauptexziton bei $\omega'_1 \approx -5,810$ mit $\tilde{\mathfrak{E}}_{0,01}^{\text{CCCF}}(\omega'_1)_{1000} \approx 99,292$, das Nebenexziton bei $\omega'_2 \approx -5,010$ mit $\tilde{\mathfrak{E}}_{0,01}^{\text{CCCF}}(\omega'_2)_{1000} \approx 1,713$ und die obere Bandkante³⁵ bei $\omega'_3 \approx 3,940$, wobei zwischen ω'_1 und ω'_2 stärkere und für $|\omega'| \leq 4$ sowie bei $\omega' \approx -7,860$ leichtere künstliche Strukturen auftreten.

³⁵Bestimmt aus der ersten Ableitung, $\partial_{\omega'} \tilde{\mathfrak{E}}_{0,01}^{\text{CCCF}}(\omega')_{1000}$

Die entsprechend gefaltete analytische Referenz, eine ‚Deltafunktion nahe eines Kontinuums‘, enthält nur das Hauptexziton bei $\omega_1 \approx -5,800$; für die Bandkante ist in der ungefalteten Testfunktion $\omega_3 = 4$ vorgegeben.

Ein ähnlicher Eindruck entsteht auch beim Vergleich unterschiedlicher Rauschfilterwerte ρ innerhalb und zwischen den Abbildungen 4.20 und 3.28 auf Seite 107 respektive 77; so weist die Entfaltung des DDMRG-Ergebnisses bei $\rho = 2\eta = 0,2$ durch eine feine ‚Sägezahnstruktur‘ nur geringfügige Konvergenzstörungen auf. Deutlich zeigt sich hier die *Strom-Strom-Korrelationsfunktion* die Notwendigkeit, bei der Wahl von ρ abzuwägen: Zwar werden mit $\rho = 0,2$ die Störungen für $-5,8 \lesssim \omega' \lesssim -5,0$ und für $-4,0 \lesssim \omega' \lesssim +4,0$ reduziert, aber dafür muß eine leicht auf $\omega'_2 \approx -4,995$ verschobene und auf $\tilde{\mathfrak{E}}_{0,01}^{\text{CCCF}}(\omega'_2)_{1000} \approx 1,283$ verminderte Exzitonenhöhe akzeptiert werden.

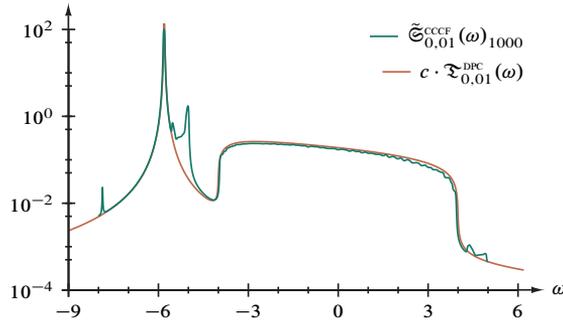


Abbildung 4.6. Vergleich des Endergebnis' der Entfaltung der Strom-Strom-Korrelationsfunktion (CCCF) im Hubbardmodell, $\tilde{\mathfrak{E}}_{0,01}^{\text{CCCF}}(\omega)_{1000}$ (Abkürzung: $\mathfrak{E}^{\text{CCCF}}(\omega) = \text{Im}(\chi(\omega))$), mit der analytischen Lösung, $c \cdot \mathfrak{T}_{0,01}^{\text{DPC}}(\omega)$. Aufgrund der abweichenden Normierung der Prüffunktion ist $c \approx 2,715$, siehe Text.

In Abbildung 4.6 sind nochmals die wesentliche Daten der Abbildungen 4.19 und 3.27, wobei für die Referenzfunktion aufgrund abweichender Norm der Faktor

$$c = \frac{\int_{-8}^{+5} d\omega \tilde{\mathfrak{E}}_{0,01}^{\text{CCCF}}(\omega)_{1000}}{\int_{-10}^{+8} d\omega \mathfrak{T}_{0,01}^{\text{DPC}}(\omega)} \approx 2,715$$

zu berücksichtigen ist. Die Höhe des Exzitons bei $\omega_1 \approx -5,800$ beträgt somit $c \cdot \mathfrak{T}_{0,01}^{\text{DPC}}(\omega_1) \approx 223,477$, wobei zu beachten ist, daß das Spektrale Gewicht des in der Testfunktion nicht enthaltenen kleineren Exzitons bei $\omega_2 \approx -5$ auf die ‚ δ -Spitze‘ und das Band verteilt sind. Bemerkenswert ist zudem das Fehlen des im Abschnitt 3.7 behandelte Bandartefakt links des Kontinuums: möglicherweise ‚verschiebt‘ das Nebenexziton diese Effekt in das Intervall $-5,8 \lesssim \omega \lesssim -5,0$.

4.4.3. *Optische Leitfähigkeit des Peierlsisolators.* Bei fehlender Wechselwirkung U beschreibt der Hamiltonoperator in Gleichung (4.2) den *Peierlsisolator*, ein System unabhängiger Elektronen, welches sich insbesondere für eine beliebige Dimerisierungsstärke Δ exakt lösen läßt. Die dimensionsbehaftete optische Leitfähigkeit ist mit

$$\sigma_1(\omega) = \frac{ae^2(4t)^2(2\Delta)^2}{2\hbar(\hbar\omega)^2} \frac{1}{\sqrt{((\hbar\omega)^2 - (2\Delta)^2)((4t)^2 - (\hbar\omega)^2)}} \quad (4.13)$$

wenn $2|\Delta| < \hbar\omega < 4t$, $\sigma_1(\omega) = 0$ sonst, berechenbar; diese Formel geht in Einheiten von ae^2/\hbar in Gleichung (3.7), die bereits in Abschnitt 3.5 diskutierte Asymmetrischen Quadratwurzeldivergenz, über, wenn dimensionslose Frequenzen durch formales Setzen von $\hbar = 1$ verwendet werden.

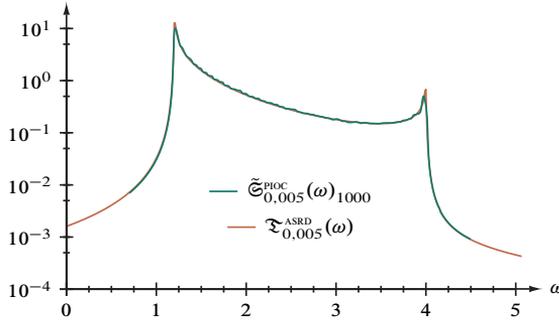


Abbildung 4.7. Vergleich des Entfaltungsergebnisses der optischen Leitfähigkeit (oc) eines Peierlsisolators, $\tilde{\mathfrak{S}}_{0,005}^{\text{PIOC}}(\omega)_{1000}$ (Abkürzung: $\mathfrak{S}^{\text{PIOC}}(\omega) = \sigma_1(\omega)$), mit der analytischen Lösung, $\tilde{\mathfrak{T}}_{0,005}^{\text{ASRD}}(\omega)$.

Abbildung 4.21 auf Seite 108 belegt, daß die Güte der Entfaltung der DDMRG-Daten (256 Gitterplätze) an jene der analytischen Prüffunktion in Abbildung 3.23 auf Seite 72 anschließt, siehe auch Abbildung 4.7: Die Maxima der Divergenzen in Teil (a) beziehungsweise (a)_{log} liegen für $\omega_1 \approx 1,215$ und $\omega_2 \approx 3,970$ bei $\tilde{\mathfrak{S}}_{0,005}^{\text{PIOC}}(\omega_1)_{1000} \approx 7,604$ und $\tilde{\mathfrak{S}}_{0,005}^{\text{PIOC}}(\omega_2)_{1000} \approx 0,382$ gegenüber $\tilde{\mathfrak{T}}_{0,01}^{\text{ASRD}}(\omega_1)_{1000} \approx 9,993$ und $\tilde{\mathfrak{T}}_{0,01}^{\text{ASRD}}(\omega_2)_{1000} \approx 0,500$ bei $\omega_1 \approx 1,210$ und $\omega_2 \approx 3,975$. Auch das Spektralgewicht, Teil (b), weist (unter Berücksichtigung der verschiedenen Frequenzintervalle, $\omega \in [0,7, 4,5]$ für $\mathfrak{S}^{\text{PIOC}}(\omega)$ und $\omega \in [-3, +7]$ für $T^{\text{ASRD}}(\omega)$) sehr ähnliche Werte auf: $\int_{0,7}^{4,5} d\omega \tilde{\mathfrak{S}}_{0,005}^{\text{PIOC}}(\omega)_{1000} \approx 2,182$ respektive $\int_{-3}^{+7} d\omega \tilde{\mathfrak{T}}_{0,01}^{\text{ASRD}}(\omega)_{1000} \approx 2,194$. Eine nahezu quantitative Übereinstimmung zeigt durch die Iterationsschrittnummern des jeweils ersten „Sprunges“, $i = 834$ für $\int_{0,7}^{4,5} d\omega \zeta^2(\omega)_i$ von $\mathfrak{S}^{\text{PIOC}}(\omega)$ und $i = 813$ für $\int_{-3}^{+7} d\omega \zeta^2(\omega)_i$ der Referenz $T^{\text{ASRD}}(\omega)$, auch Teil (f).

Die Nähe der Entfaltungsergebnisse zur Prüffunktion wird auch in Abbildung 4.22 auf Seite 109, welche die Variation des Rauschfilterwertes ρ behandelt, deutlich: Wiederum

erweist sich, daß eine Erhöhung von ρ über die DDMRG-Ausgangsfaltung von $\eta = 0,05$ hinaus die Konvergenz, vergleiche Teil (d), deutlich (obschon nicht so drastisch wie bei der Prüffunktion in Abbildung 3.24 auf Seite 73) beeinträchtigt und keine Verbesserung des Resultats bewirkt.

4.4.4. *Reduzierte optische Leitfähigkeit des Mott-Peierls-Isolators.* Im Limes starker Wechselwirkung, $U/t \rightarrow \infty$, beschreibt Gleichung (4.2) einen Sonderfall des *Mott-Peierls-Isolators*, für welchen die optische Leitfähigkeit ebenfalls analytisch angegeben werden kann. Unter der Annahme $0 < |\Delta| < 2t$ besteht das Spektrum im Bereich $2|\Delta| \leq |\hbar\omega - U| \leq 2t$ aus zwei Bändern,

$$\sigma_1(\omega) = \frac{ae^2 g_0}{8\hbar} \frac{\sqrt{((\hbar\omega - U)^2 - (2\Delta)^2)((4t)^2 - (\hbar\omega - U)^2)}}{\hbar\omega|\hbar\omega - U|}, \quad (4.14)$$

und einer diese in der Lücke (Breite: $4|\Delta|$) bei $\hbar\omega = U$ trennenden δ -Funktion

$$\sigma_1(\omega) = \frac{\pi ae^2 g_\pi t^2}{\hbar U} \delta(\hbar\omega - U). \quad (4.15)$$

Die darin auftretenden Spin-Formfaktoren g_0 und g_π werden in Referenz [11] behandelt. Auch dieser Funktionstypus wurde bereits in Form der ‚Deltafunktion zwischen zwei Bändern‘ im Unterabschnitt 3.8 vorbereitet; dortige Formel (3.10) ergibt sich aus obigen Gleichungen (4.14) und (4.15).

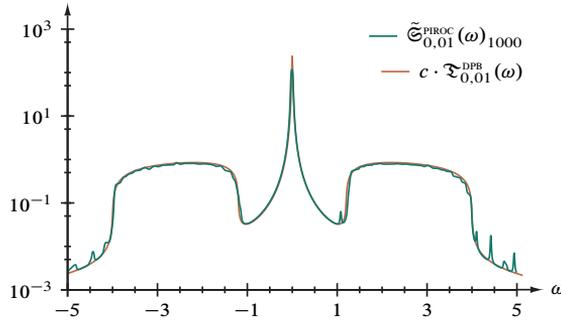


Abbildung 4.8. Vergleich des Entfaltungsergebnis' der reduzierten optischen Leitfähigkeit (ROC) eines Mott-Peierls-Isolators, $\tilde{\Sigma}_{0,01}^{\text{PIROC}}(\omega)_{1000}$ (Abkürzung: $\tilde{\Sigma}^{\text{PIROC}}(\omega) = \omega \sigma_1(\omega)$), mit der analytischen Lösung, $c \cdot \Sigma_{0,01}^{\text{DPB}}(\omega)$. Aufgrund der abweichenden Normierung der Prüffunktion ist $c \approx 2,410$, siehe Text.

Daß die Entfaltung des DDMRG-Resultats (128 Gitterplätze) nahezu gleichermaßen unproblematisch ist, wie die Behandlung der Referenzfunktion, belegt der Vergleich von Abbildung 4.23 auf Seite 110 mit Abbildung 3.29 auf Seite 78 im Besonderen mit Blick auf Teil (f). Die Teile (a)_{log.}, kombiniert in Abbildung 4.8, zeigen jedoch deutliche Artefakte zwischen der ‚ δ -Spitze‘ und dem oberen Band bei $\omega \approx 1,080$ sowie

außerhalb der Bänder für $|\omega| \gtrsim 4$. Überdieses ist die Spitze selbst verschoben und im Funktionswert kleiner: $\omega \approx -0,005$ mit $\tilde{\mathfrak{E}}_{0,01}^{\text{PIROC}}(\omega)_{1000} = 116,748$ gegenüber $\omega = 0$ mit $c \cdot \mathfrak{T}_{0,01}^{\text{DPB}}(\omega) \approx 236.205$, wobei sich

$$c = \frac{\int_{-5}^{+5} d\omega \tilde{\mathfrak{E}}_{0,01}^{\text{PIROC}}(\omega)_{1000}}{\int_{-7}^{+7} d\omega \mathfrak{T}_{0,01}^{\text{DPB}}(\omega)} \approx 2,410$$

aus der abweichenden Normierung der analytischen Funktion ergibt.

Entsprechende Ähnlichkeiten finden sich auch bei Gegenüberstellung der Entfaltungsergebnisse mit variierten Rauschdämpfungswerten ρ , vergleiche Abbildung 4.24 auf Seite 111 respektive Abbildung 3.30 auf Seite 79. Von einigem Vorteil für die Behandlung der reduzierten optischen Leitfähigkeit erscheint die Verdopplung $\rho = 2\eta = 0,2$, welche die oben erwähnten artifiziiellen Strukturen mildert, die „Schärfe“ bei $\omega \approx -0,005$ mit $\tilde{\mathfrak{E}}_{0,01}^{\text{PIROC}}(\omega)_{1000}^{0,2} \approx 117,128$ im Rahmen der numerischen Genauigkeit unbeeinflusst läßt.

4.5. Abbildungen. Die Hauptabbildungen in diesem Abschnitt haben die folgenden Gemeinsamkeiten:

- Teil (a): In $i = 1000$ Schritten rauschgedämpft ($\rho = \eta$) entfalteter und mit $\eta' = \eta/10$ rückgefalteter DDMRG-Datensatz der jeweiligen Spektralfunktion, gegebenenfalls um Zwischenergebnisse und analytische Lösungsfunktionen ergänzt (siehe Abbildungstitel);
- Teil (b): Spektralgewicht von (a) nach der i ten Iteration, siehe Gleichung (2.1);
- Teil (c): Gradienten von (a);
- Teil (d): Normierter Minimierungsparameter R_i/R_0 nach der i ten Iteration;
- Teil (e): Summe aller Korrekturen $\sum_{i=1}^{1000} \zeta(\omega)_i$ am Ausgangsdatensatz nach dem letzten der 1000 Schritte;
- Teil (f): Gesamtkorrektur $\int_{B' \subset \mathbb{R}} d\omega \zeta^2(\omega)_i$ am entfaltetem Spektrum nach der i ten Iteration.

Die jede Hauptdarstellung begleitende Ergänzungsabbildung ist folgendermaßen gegliedert:

- Teil (a): Unter Variation des Rauschdämpfungsparameters ρ in $i = 1000$ Schritten störungsgemindert entfalteter und mit $\eta' = \eta/10$ rückgefalteter DDMRG-Datensatz der jeweiligen Spektralfunktion;
- Teil (b): Normierter Minimierungsparameter R_i/R_0 nach der i ten Iteration;
- Teil (c): Summe aller Korrekturen $\sum_{i=1}^{1000} \zeta(\omega)_i$ am Ausgangsdatensatz nach dem letzten Entfaltungsschritt;
- Teil (d): Gesamtkorrektur $\int_{B' \subset \mathbb{R}} d\omega \zeta^2(\omega)_i$ am entfaltetem Spektrum nach der i ten Iteration.

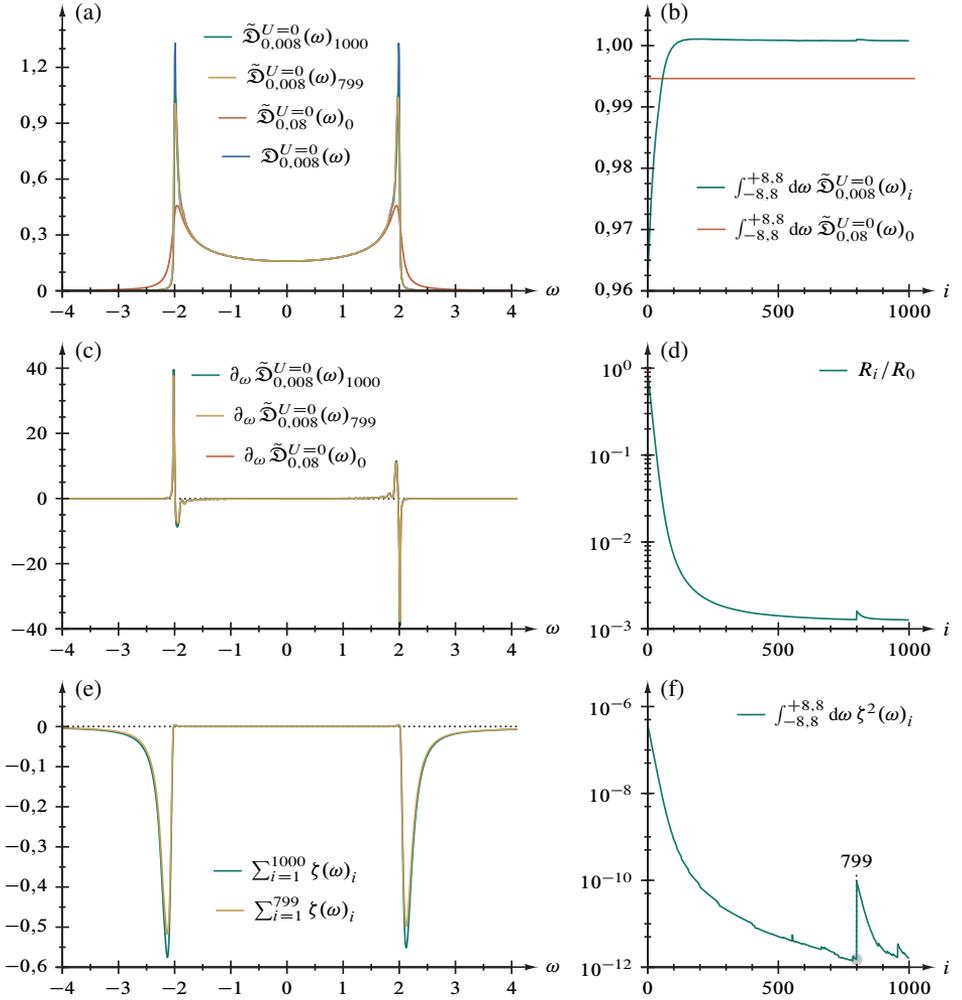


Abbildung 4.9. Entfaltung der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathfrak{D}}_{0,008}^{U=0}(\omega)$, 6372 irreguläre Gitterpunkte im Intervall $\omega \in [-8,800, +8,792]$. Zusätzlich ist das Zwischenergebnis nach 799 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

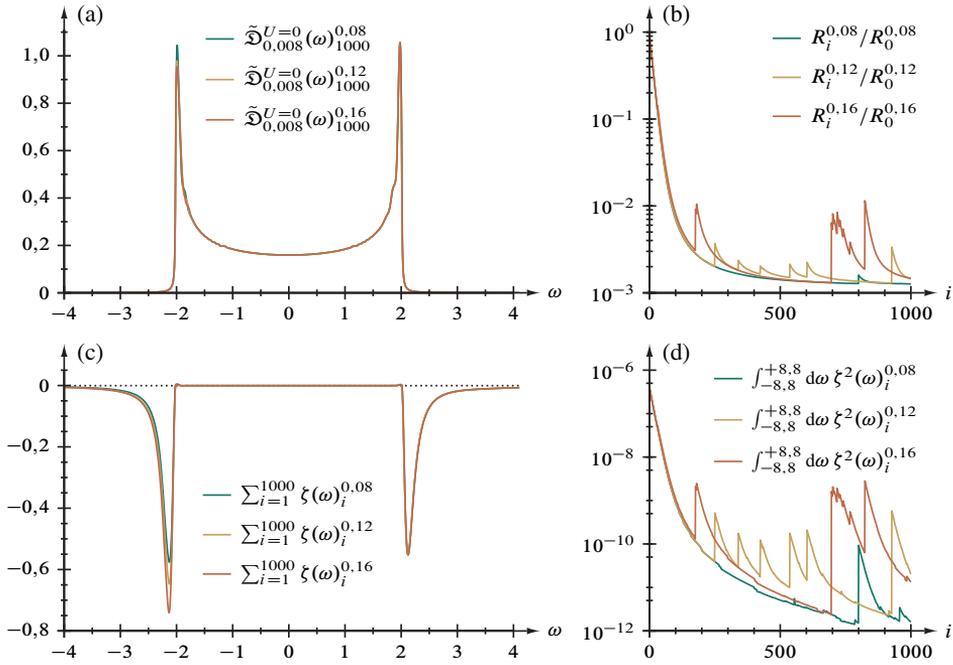


Abbildung 4.10. Entfaltung der Zustandsdichte des Hubbardmodells bei $U = 0$, Abbildung 4.9, mit unterschiedlich starker Rauschunterdrückung.

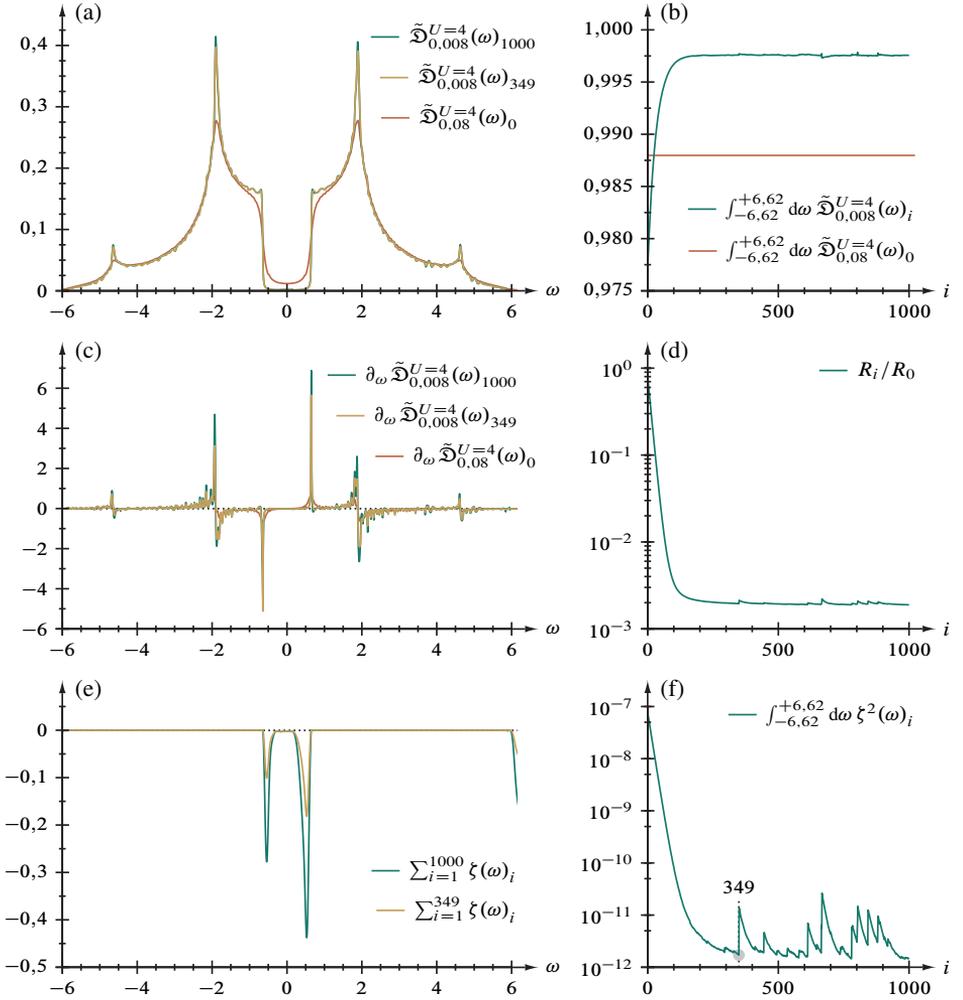


Abbildung 4.11. Entfaltung der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathcal{D}}_{0,08}^{U=4}(\omega)$, 1760 irreguläre Gitterpunkte im Intervall $\omega \in [-6,62, +6,62]$. Zusätzlich ist das Zwischenergebnis nach 349 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

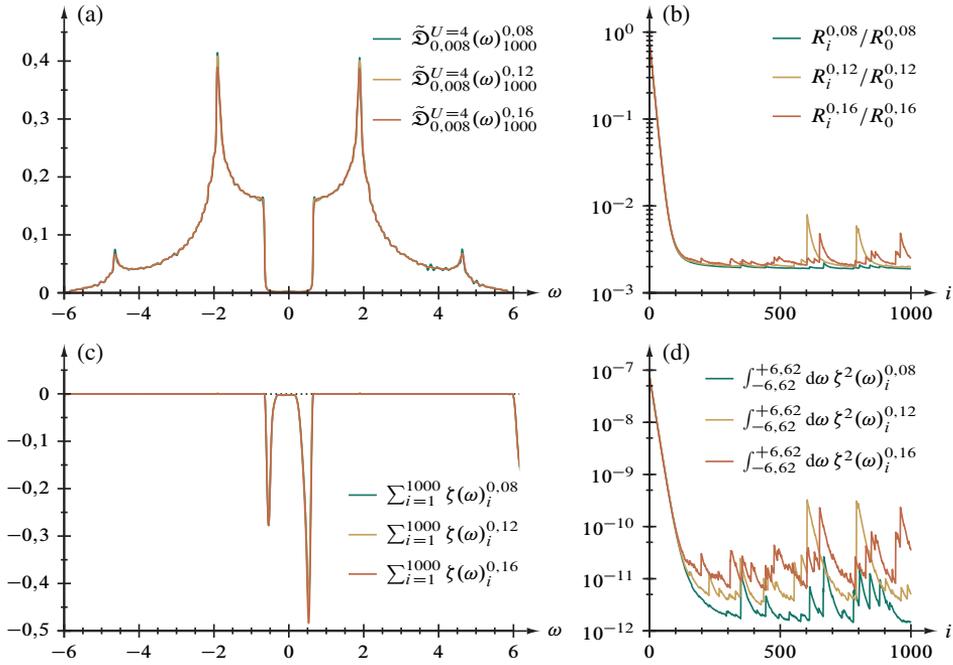


Abbildung 4.12. Entfaltung der Zustandsdichte des Hubbardmodells bei $U = 4$, Abbildung 4.11, mit unterschiedlich starker Rauschunterdrückung.

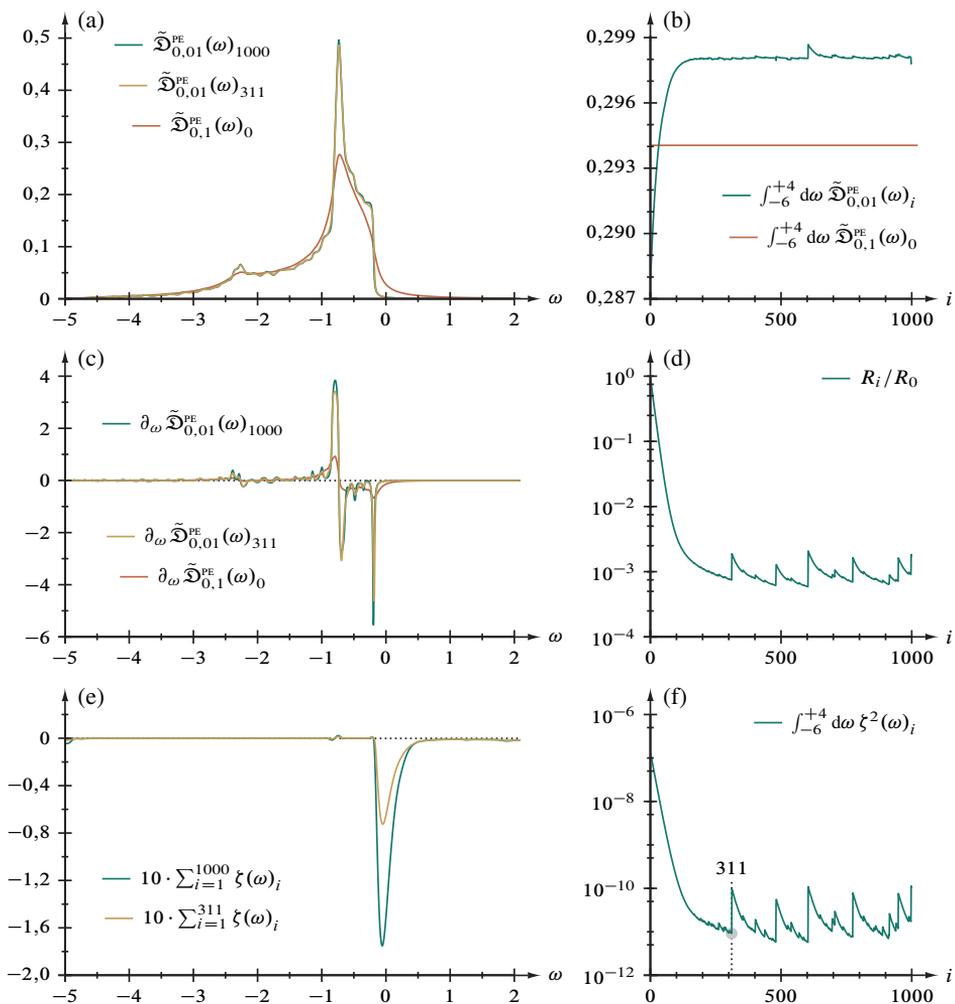


Abbildung 4.13. Entfaltung der Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ im Bereich der Photoemission (PE), $\tilde{\mathcal{D}}_{0,1}^{\text{PE}}(\omega)$, 1001 reguläre Gitterpunkte im Intervall $\omega \in [-6, +4]$. Zusätzlich ist das Zwischenergebnis nach 311 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

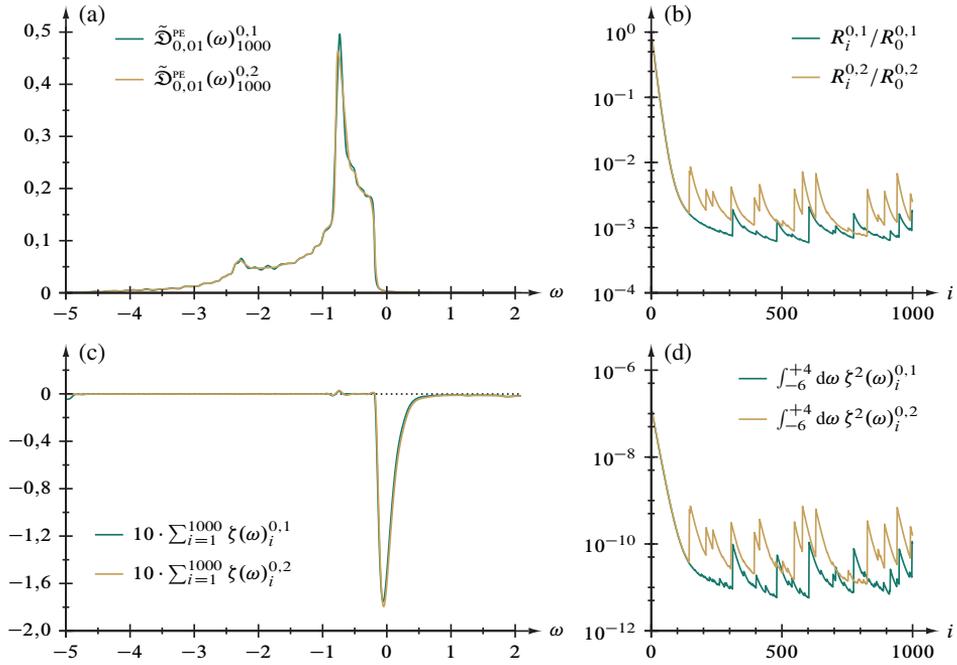


Abbildung 4.14. Entfaltung der Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ (Photoemission), Abbildung 4.13, mit unterschiedlich starker Rauschunterdrückung.

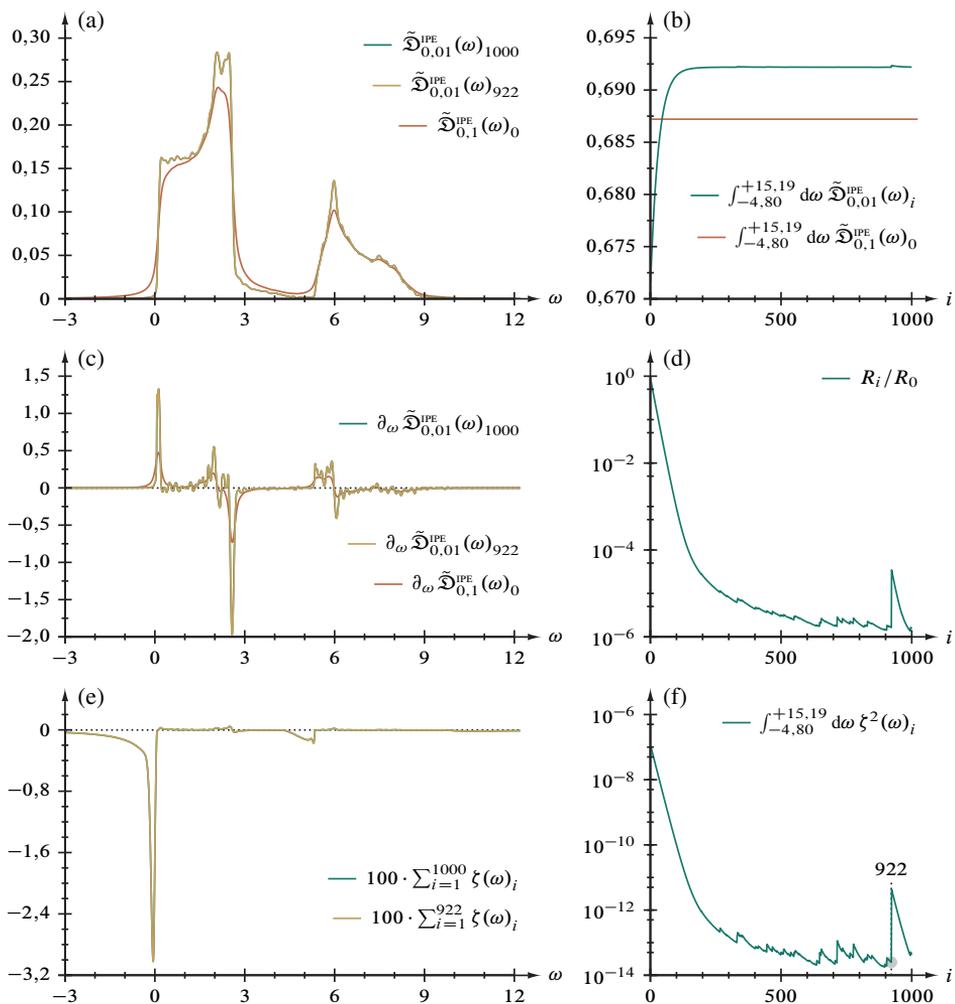


Abbildung 4.15. Entfaltung der Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ im Bereich der inversen Photoemission (IPE), $\tilde{\mathcal{D}}_{0,1}^{\text{IPE}}(\omega)$, 4000 reguläre Gitterpunkte im Intervall $\omega \in [-4,8085, +15,1865]$. Zusätzlich ist das Zwischenergebnis nach 922 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

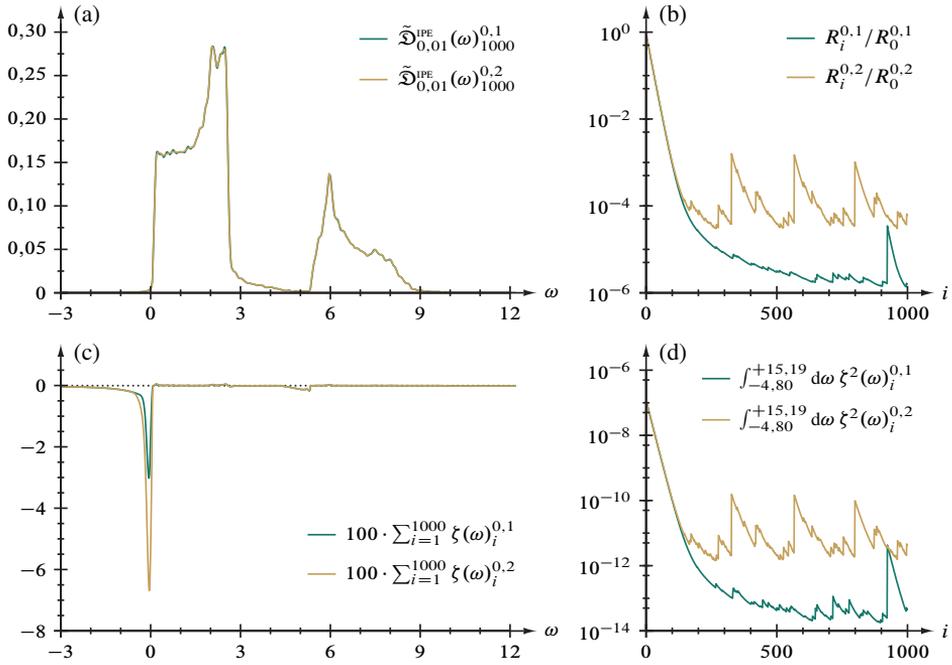


Abbildung 4.16. Entfaltung der Zustandsdichte einer Luttingerflüssigkeit bei $U = 4,9$ (inverse Photoemission), Abbildung 4.15, mit unterschiedlich starker Rauschunterdrückung.

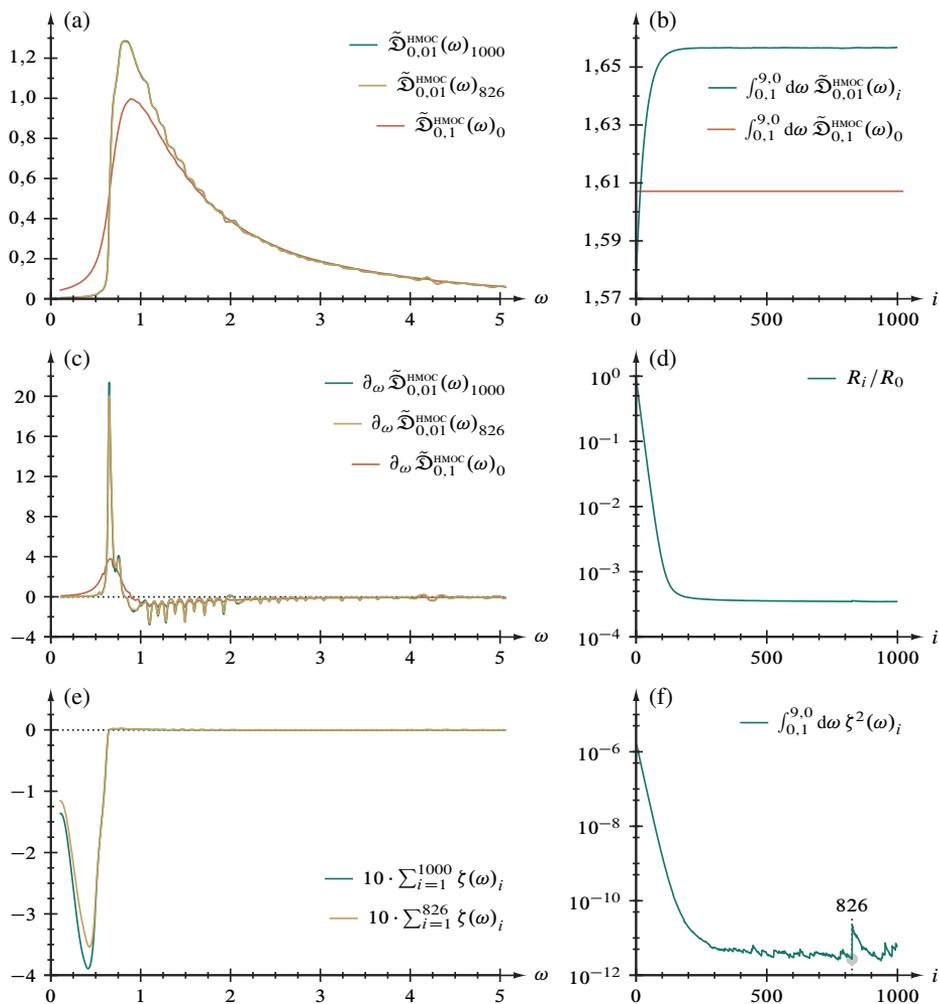


Abbildung 4.17. Entfaltung der optischen Leitfähigkeit (OC) im eindimensionalen Hubbardmodell bei $U = 3$, $\tilde{\mathcal{D}}_{0,1}^{\text{HMOC}}(\omega)$, 261 irreguläre Gitterpunkte im Intervall $\omega \in [0,1, 9,0]$. Zusätzlich ist das Zwischenergebnis nach 826 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

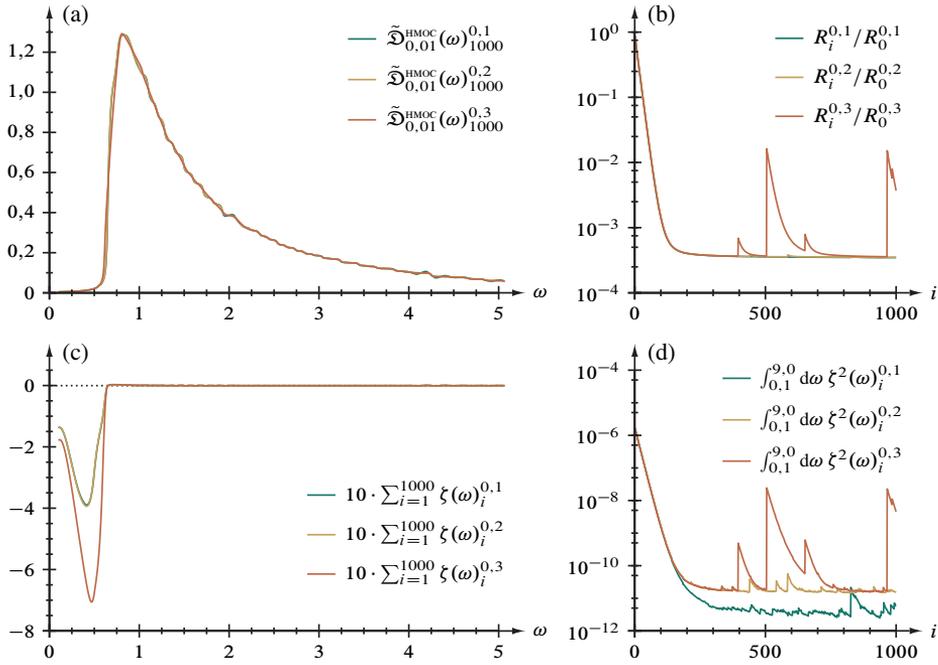


Abbildung 4.18. Entfaltung der optischen Leitfähigkeit im Hubbardmodell bei $U = 3$, Abbildung 4.17, mit unterschiedlich starker Rauschunterdrückung.

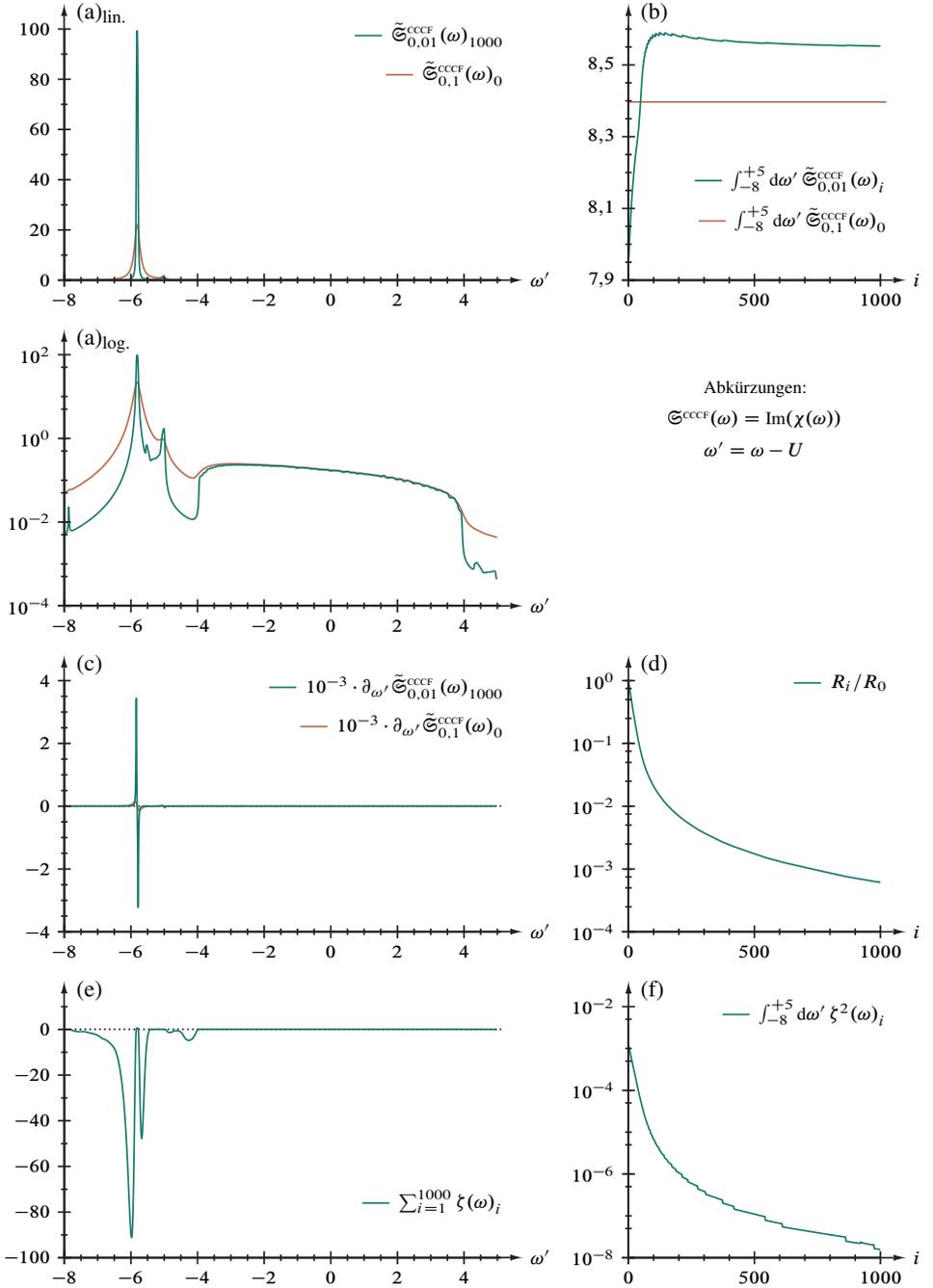


Abbildung 4.19. Entfaltung der Strom-Strom-Korrelationsfunktion (CCCF) im eindimensionalen Hubbardmodell, $\tilde{\mathcal{E}}_{0,1}^{\text{CCCF}}(\omega') = \text{Im}(\chi(\omega - U))_{0,1}$, 447 irreguläre Gitterpunkte im Intervall $\omega \in [-8, +5]$.

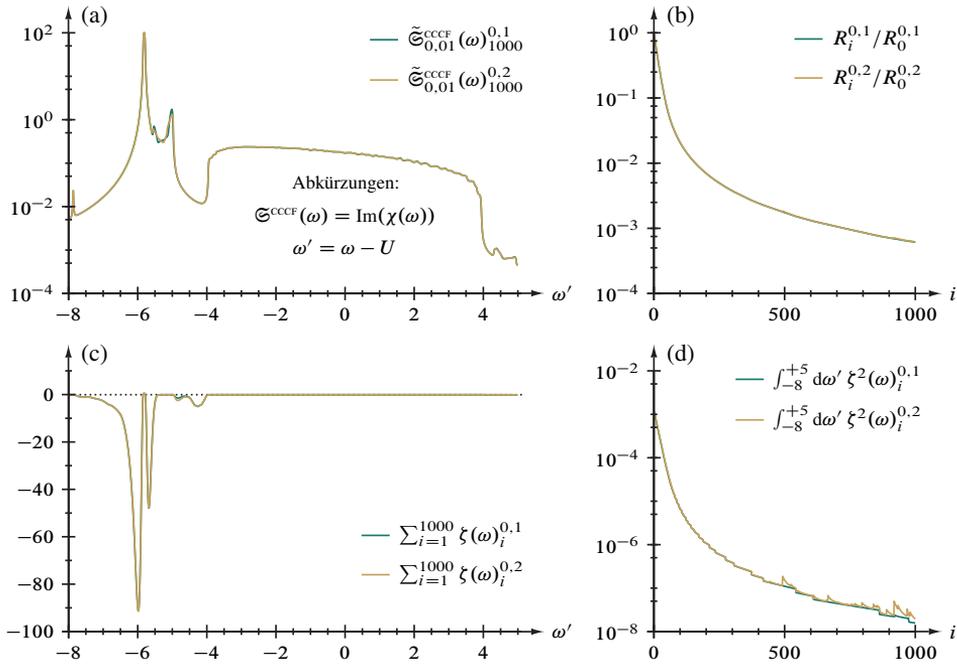


Abbildung 4.20. Entfaltung der Strom-Strom-Korrelationsfunktion im Hubbardmodell, Abbildung 4.19, mit Rauschunterdrückung zweier verschiedener Stärken.

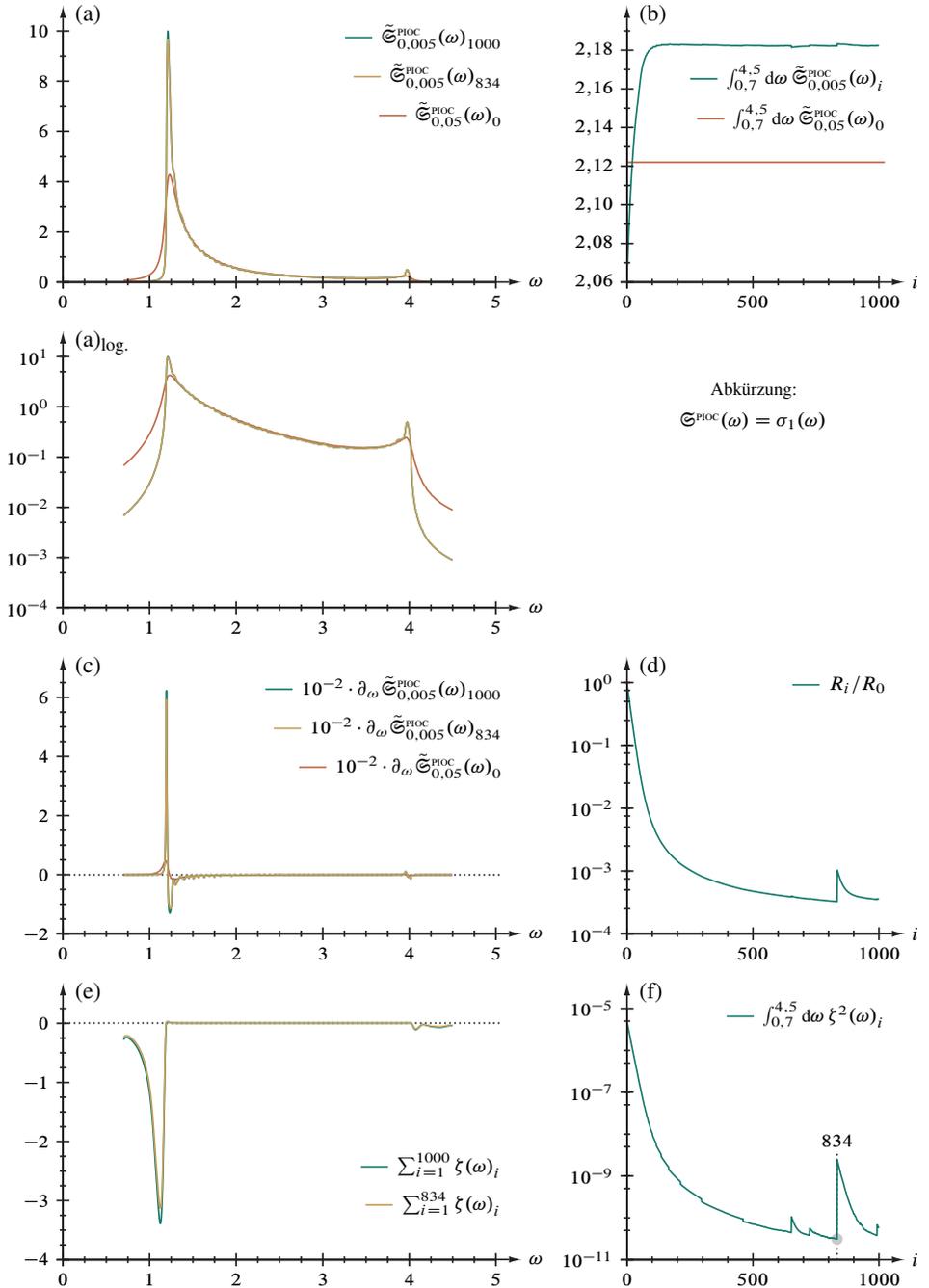


Abbildung 4.21. Entfaltung der optischen Leitfähigkeit (OC) eines Peierlsisolators, $\tilde{\mathfrak{E}}_{0,05}^{\text{pIOC}}(\omega) = \sigma_1(\omega)_{0,05}$, 759 reguläre Gitterpunkte im Intervall $\omega \in [0,7051, 4,4951]$. Zusätzlich ist das Zwischenergebnis nach 834 Entfaltungsschritten eingezeichnet, vergleiche „Sprung“ in Abbildungsteil (f).

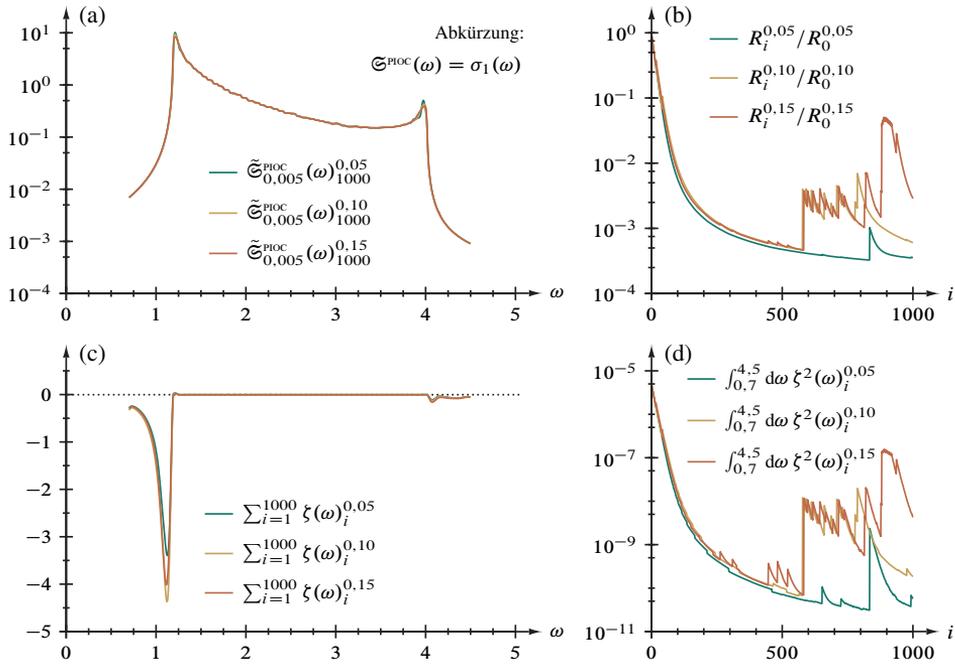


Abbildung 4.22. Entfaltung der optischen Leitfähigkeit eines Peierlsisolators, Abbildung 4.21, mit unterschiedlich starker Rauschunterdrückung.

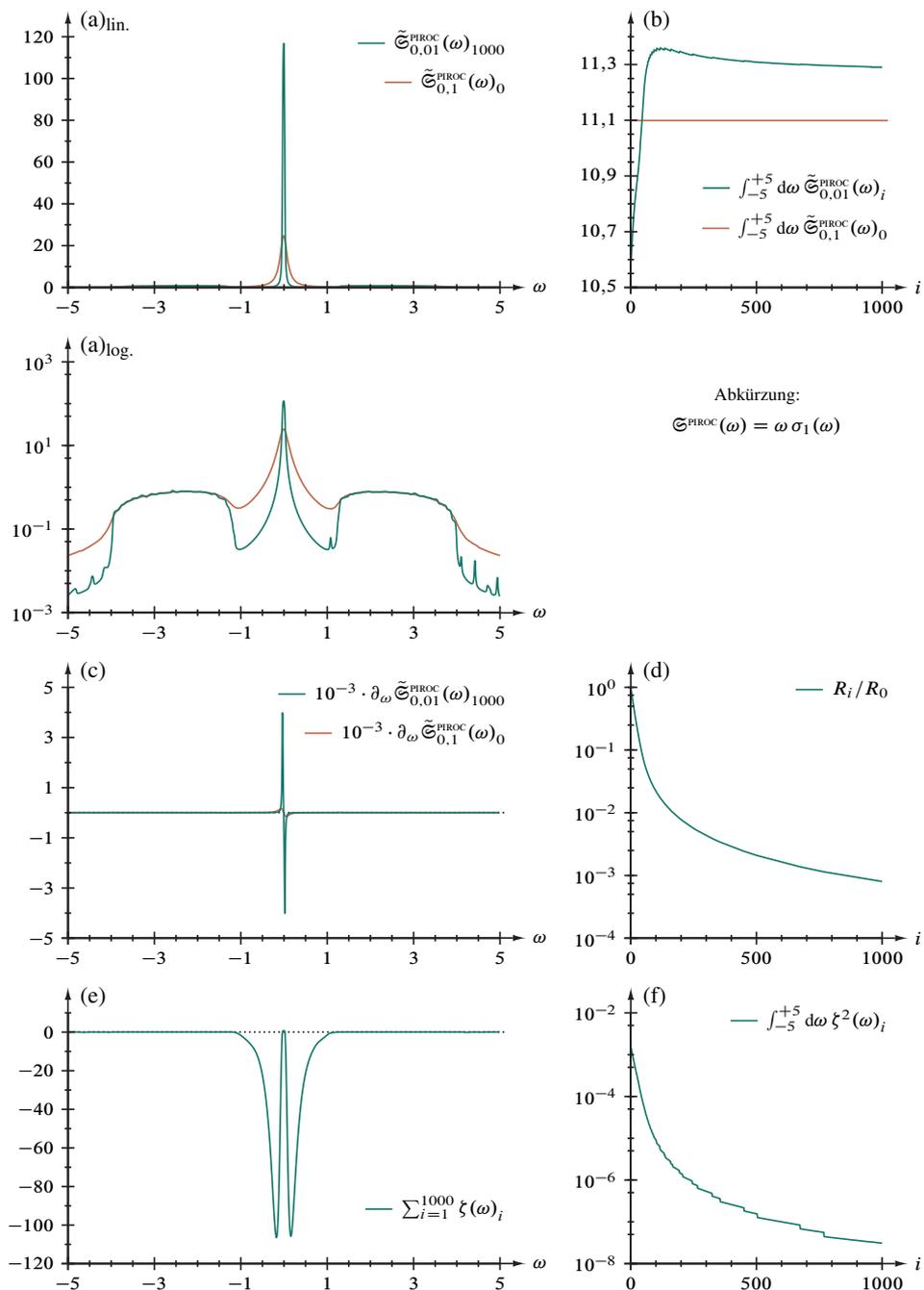


Abbildung 4.23. Entfaltung der reduzierten optischen Leitfähigkeit (ROC) eines Mott-Peierls-Isolators, $\tilde{\mathfrak{E}}_{0,1}^{\text{PIROC}}(\omega) = \omega \sigma_1(\omega)_{0,1}$, 1001 reguläre Gitterpunkte im Intervall $\omega \in [-5, +5]$.

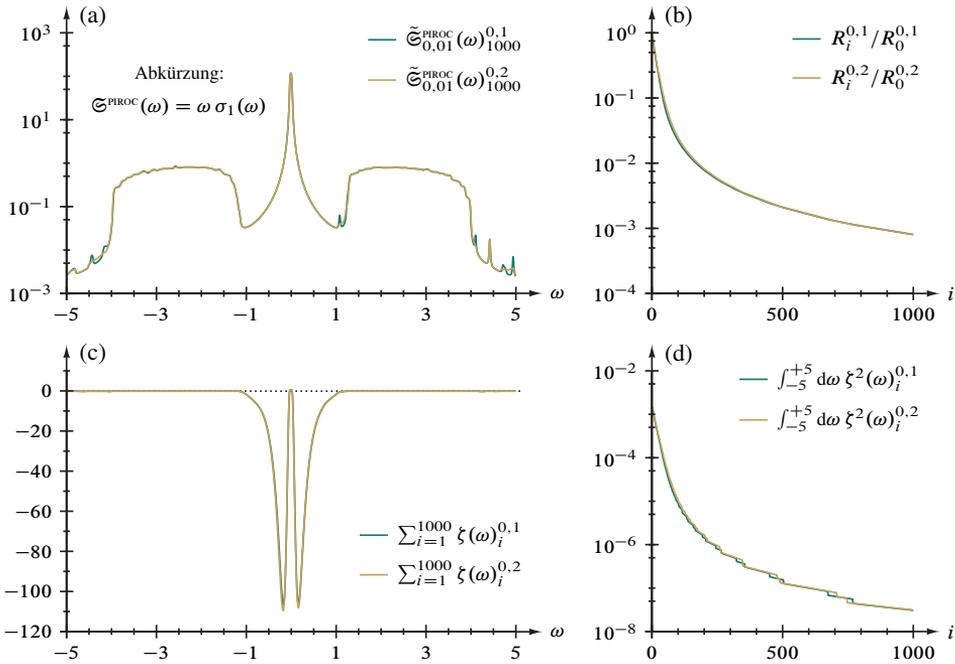


Abbildung 4.24. Entfaltung der reduzierten optischen Leitfähigkeit eines Mott-Peierls-Isolators, Abbildung 4.23, mit unterschiedlich starker Rauschunterdrückung.

5. Zusammenfassung und Ausblick

Mit dieser Arbeit wird erstmals ein Verfahren auf konzeptioneller Grundlage der blinden Entfaltung vorgelegt, welches im Thermodynamischen Limes, also für das makroskopische, quasi unendlich ausgedehnte Quantensystem, die Berechnung von abschnittsweise glatten und positiv-semidefiniten Spektralfunktionen endlicher Breite aus DDMRG-Ergebnissen erlaubt. Damit werden bei vergleichsweise geringer Rechenzeit die zwei zentralen Beschränkungen der numerischen Simulation überwunden, nämlich die endliche Systemgröße und die endliche Faltungsbreite – ohne aufwendige Extrapolationsanalysen durchführen zu müssen. Allerdings bleibt dabei zu bedenken, daß das Ergebnis aufgrund nicht-linearer Nebenbedingungen einen Kompromiß zwischen der Treue zu den Ausgangsdaten und zu deren Entfaltung darstellt. Praktisch reduziert sich das Verfahren auf eine Ausgleichsrechnung nach dem Ansatz kleinster Fehlerquadrate unter den seitens der Physik an das Spektrum gestellten, oben genannten Forderungen sowie dem Anspruch, das Spektrale Gewicht bleibe im Verfahrensverlauf konstant.

Zentrale Idee für dieses Entfaltungsverfahren ist, den wahren Faltungskern als aus einer exakt formulierbaren Lorentzverteilung und durchaus substantiellen nicht quantifizierbaren Beiträgen zusammengesetzt anzusehen, nämlich der endlichen Systemgröße und Datenpunkanzahl sowie den statistischen Ungenauigkeiten der Numerik. Dies überträgt sich einerseits auf die zu minimierende *cost function* R , Gleichung (1.15), welche explizit den „Lorentz-Anteil“ enthält, andererseits auf die physikalisch motivierten Ansprüche an die Minimierung von R , in welche die unbekanntenen Anteile eingehen. Damit wird die Prozedur zu einem Abwägen zwischen dem Einklang des Entfaltungsergebnisses mit den Ausgangsdaten, der Erfüllung der genannten Forderungen und der Stabilität der Lösung [28]; Konsequenz daraus ist wiederum, daß signifikante Abweichungen vom absoluten Minimum der *cost function* auftreten können. Damit ist die Neuheit dieses Verfahrens umrissen: Die unbekanntenen Beiträge zum Faltungskern werden nun – realitätsnäher – nicht mehr als kleine, die Übereinstimmung von Lösung und Ausgangsdaten nur wenig beeinflussende „Störungen“ behandelt, wie dieses in Ansätzen erfolgt, die eine fast vollständig dominierende Lorentzfaltung annehmen.

5.1. Derzeitiger Stand. Das Gesamtverfahren, von Vorbereitungs- und Nachbearbeitungsstufe umrahmt, besteht aus einem iterativen Hauptteil, der in jedem Zyklus zunächst durch den Optimierungsschritt das alte Zwischenergebnis entfaltet und anschließend das neue Zwischenergebnis glättet. Da insbesondere der wesentliche Glättungsschritt eines Ausgangsdaten-spezifisch zu wählenden Parameters für die Unterscheidung von „Nutzinformation“ und numerischem Rauschen bedarf, wurde die Methode umfassend an Spektren getestet, die im Thermodynamischen Limes und ohne Faltung mathematisch-exakt bekannt und physikalisch repräsentativ sind; dazu gehören im Besonderen die Einteilchen-Zustandsdichte und die optische Leitfähigkeit des eindimensionalen Mott-Hubbard-Isolators [10, 11, 36]. Dabei hat sich gezeigt, daß verschiedenartige Singularitäten – Potenzgesetz-Bandkanten, Stufen und exzitonische Spitzen – in den abschnittsweise glatten Spektralfunktionen zufriedenstellend verdeutlicht werden. Im

Besonderen ist eine Reduktion des Faltungsgrades um mehrere Größenordnungen möglich, wobei der Lorentzkern gegen einen generell „schmaleren“ Gaußkern ersetzt werden kann, wie in Abschnitt 1.1 des Anhanges vorgestellt wird.

Beeinträchtigt wird die Brauchbarkeit dieses Entfaltungsverfahrens durch das Auftreten artifizierlicher „Schulter-artiger“ Strukturen auf Energieskalen oberhalb des Wertes für den Rauschdämpfungsparameter, $\Delta\nu \gtrsim \rho$, die sich phänomenologisch nach den benachbarten Funktionsmerkmalen in Divergenzartefakte, Stufenartefakte und Bandartefakte unterteilen lassen. Dieses für sich gewichtige Problem relativiert sich allerdings, sofern diese Fehler keine Nutzinformationen überlagern und nicht als solche fehlinterpretiert werden können. Damit ist das Verfahren bereits eingeschränkt universell verwendbar, wengleich die optimale Einstellung der Glättung ausprobiert werden muß.

5.2. Aussichten.

5.2.1. *Offene Probleme.* Das Entfaltungsverfahren ist vor allem mit Blick auf die Artefakte im Endergebnis verbesserungsbedürftig. Überdies muß in einer weiterführenden Untersuchung der Frage nachgegangen werden, warum bezüglich der Ergebnisqualität einige Spektralfunktionsstypen, insbesondere die Lorentzverteilung und die Halbelliptische Verteilung, vergleiche Abschnitte 3.2 beziehungsweise 3.3 sowie Abbildungen 3.17 respektive 3.19, gegenüber den anderen Testfunktionen augenscheinlich benachteiligt sind. Bereits in diesem Zusammenhang durchgeführte Verfahrensmodifikationen haben gezeigt, daß dabei das Glättungsverfahren genauerer Betrachtung bedarf. Dabei könnte etwa der bislang nicht generell erfolgreiche Ansatz verfolgt werden, die zweite Ableitung der Spektralfunktion zur Rauscherkennung hinzuzuziehen; auch ist der Frage nachzugehen, ob die Parameter der verbesserten Extremumssuche, siehe Abschnitt 2.3.2 bereits optimal definiert sind, da beispielsweise die Höhendifferenz, Gleichung (2.10), bislang keinen nennenswerten günstigen Einfluß auf die Ergebnisqualität hat. Überdies steht noch in Anschluß an Abschnitt 2.1.2 die Formulierung eines Spektralfunktions-unabhängigen Abbruchkriteriums aus, das vorteilhaft für die angestrebte „Robustheit“ des Gesamtverfahrens als *DDMRG-postprocessing* ist.

5.2.2. *Physikalisch motivierte Erweiterungen.* Die hier beschriebene Methode kann auf verschiedenen Arten verallgemeinert werden. Beispielsweise liegt eine Anpassung auf irreguläre *DDMRG*-Daten nahe – solche also mit variabler Frequenzrastrerweite $\Delta\nu$, welche eine feinere Auflösung scharfer Spektralfunktionsmerkmale erlaubt. Da aber die limitierende Skala der nicht frequenzabhängig definierte Faltungsbreiteparameter η ist, wird dieser Ansatz die Qualität der Resultate allenfalls über eine möglicherweise wirksamere Glättung verbessern. Eine Herangehensweise, die auch η als Funktion der Frequenz ansieht, wurde bereits für Störstellenprobleme beschrieben [21]; eine derartige Generalisierung auch für die Entfaltung scheint möglich, wurde jedoch noch nicht versucht. Um Informationen über die Faltungsbreitenabhängigkeit des Spektrums in das Verfahren einzubringen, könnten die *cost functions* für *DDMRG*-Datensätze mit unterschiedlichem Wert von η zu einer gemeinsamen *cost function* summiert werden. Auch ein solcher Schritt scheint in weitergehenden Entwicklungen angeraten.

5.2.3. *Mathematisch motivierte Erweiterungen.* Routineanwendungen des Entfaltungsverfahrens in seiner hier genutzten Endfassung, vergleiche Abschnitt 2.5, stehen vor allem die in Abschnitt 5.2.1 dargelegten Probleme entgegen. Zu deren Lösung tragen möglicherweise Verbesserungen in den Numerikalgorithmen bei, welche sonst für eine flexiblere Nutzung bei „schwierigen“ Funktionsdetails nötig sind, siehe vorausgehender Abschnitt 5.2.2. Vornehmlich in der Rauschdämpfungsmethode und bei der Rediskretisierung und Regularisierung der Ausgangsdaten, Abschnitt 2.4, werden aber Näherungen, etwa durch lineare Interpolationen, vorgenommen, die feine Strukturen in den Spektren (zer)stören können; diesem entgegengereicht könnte künftig zum Beispiel ein *spline*-basierter Ansatz [26, 33, 34] zur Anwendung kommen. Die Rückfaltung mit einer Gaußverteilung wurde bereits erwähnt; im Abschnitt 1.1 des Anhangs werden die erforderlichen Programmodifikationen behandelt.

5.2.4. *Datenverarbeitungstechnisch motivierte Erweiterungen.* In der derzeitigen Implementierung der Methode sind auf modernen Arbeitsplatzcomputern³⁶ Verarbeitungszeit und Ressourcenbedarf gegenüber der eigentlichen Berechnung der Spektralfunktionen mit DDMRG unkritisch. Insbesondere der Zeitaufwand kann jedoch durch algorithmische Änderungen zwecks Überwindung der zuvor beschriebenen verbleibenden Einschränkungen sowie die vorgenannten durch Physik und Numerische Mathematik angeregten Erweiterungen signifikant anwachsen. Diesem entgegengereicht wäre eine (teilweise) Parallelisierung, welche aufgrund der Abhängigkeit der Daten mit den derzeitigen Algorithmen nicht möglich (Entfaltung) oder aufgrund des zu großen Aufwandes im Rahmen dieser Arbeit noch nicht erfolgt (Glättung) ist, fallweise zu überprüfen. Überdies besteht gegebenenfalls die Aussicht, mit Hilfe von Vektorisierungen, die teilweise auf Kompilerebene erfolgen können, den erweiterten Funktionsumfang der Fließkommaeinheiten neuerer Mikroprozessoren besser nutzbar zu machen.

³⁶Eine Sonderversion des Programmes, welche ohne graphische Kontrollausgaben über Grace arbeitet und sich für den Stapelbetrieb oder die automatisierte Anwendung in Skripten und somit für größere Rechenanlagen, beispielsweise vom Typ HP AH337A [48], eignet, liegt ebenfalls bereits vor.

Anhang: Ergänzungen zum Rechenverfahren

I. Programmmodifikationen

Die Entwicklung der in dieser Studie in Abschnitt 2.5 vorgestellten Methode zur Entfaltung von DDMRG-Spektren hat im Verlaufe der Entwicklung, wie im Kapitel 2 vereinfacht dargelegt wird, einige Varianten hervorgebracht. Während dabei der eigentliche „Programmkern“ bis auf eine Verbesserung bei der Erhaltung der Funktionssymmetrie, vergleiche Abschnitt 2.3.1, im Wesentlichen nur algorithmische Optimierungen hinsichtlich der Verarbeitungsgeschwindigkeit erfahren hat, sind aufgrund der vielfältigen numerischen Effekte mehrere Ansätze zur Glättung innerhalb des iterativen Verfahrensablaufs verfolgt worden – insbesondere zur Identifikation des Rauschens. Gegen Abschluß der Arbeit sind noch zwei weitere, die Rückfaltung betreffende Änderungen ins Blickfeld gekommen, einerseits die Verwendung der schmalen Gaußverteilung *vor der Endergebnisausgabe*, erläutert in Abschnitt 1.1, andererseits die Anwendung der Rückfaltungsprozedur (mit Lorentz- oder Gaußverteilung) *nach jedem Entfaltungsschritt*, skizziert in Abschnitt 1.2. Eine exemplarische Anwendung der Kombination beider Modifikationen wird in Abschnitt 2 vorgestellt.

1.1. Rückfaltung mit einer Gaußverteilung. Die Gauß(-Rück)faltung [17],

$$\mathfrak{S}_i = \frac{1}{\sqrt{2\pi\eta'}} \sum_{j=1}^L \Delta\mu_j S_j \exp\left(-\frac{1}{2}\left(\frac{\mu_i - \mu_j}{\eta'}\right)^2\right) \quad \text{für } i = 1, 2, \dots, L$$

stellt einen „Ersatz“ für Gleichung (2.5) dar und modifiziert folglich nur den Schritt 2 im Ablaufplan der Rückfaltungsprozedur, beschrieben in Abschnitt 2.2.3. Generell lassen sich an dieser Stelle im Sinne der Grundgleichung (1.5) beliebige Kerne gegeneinander substituieren, wodurch diesbezüglich weitergehende Untersuchungen vereinfacht werden.

Wesentlicher *Vorteil* gegenüber der Lorentzverteilung liegt im steileren Abfall der Flanken, wodurch Funktionsmerkmale schärfere hervortreten, ohne, wie bei Auslassung dieses Programmpunktes, die behandelten *finite-size*-Effekte zu rekonstruieren. Ein diesen Vorzug verdeutlichendes Beispiel wird in Abschnitt 2 dieses Anhangs behandelt. Problematisch dabei bleibt, daß selbstverständlich mit den realen, also für die Physik

bedeutsamen, auch die unerwünschten artifiziellen Strukturen betont und somit eindeutige Interpretationen in Sonderfällen schwieriger werden. Hauptsächlich, indes gegenüber dem Aufwand der DDMRG-Rechnung vernachlässigbarer Nachteil ist die merkliche erhöhte Rechenzeit durch die Numerik der auftretenden Exponentialfunktion: Bei 15000 Datenpunkten beläuft sich diese beispielsweise auf etwa 3 Stunden für 1000 Iterationen auf einer Verarbeitungseinheit (Kern) eines Mikroprozessors vom Typ Intel Itanium 9560 [48].

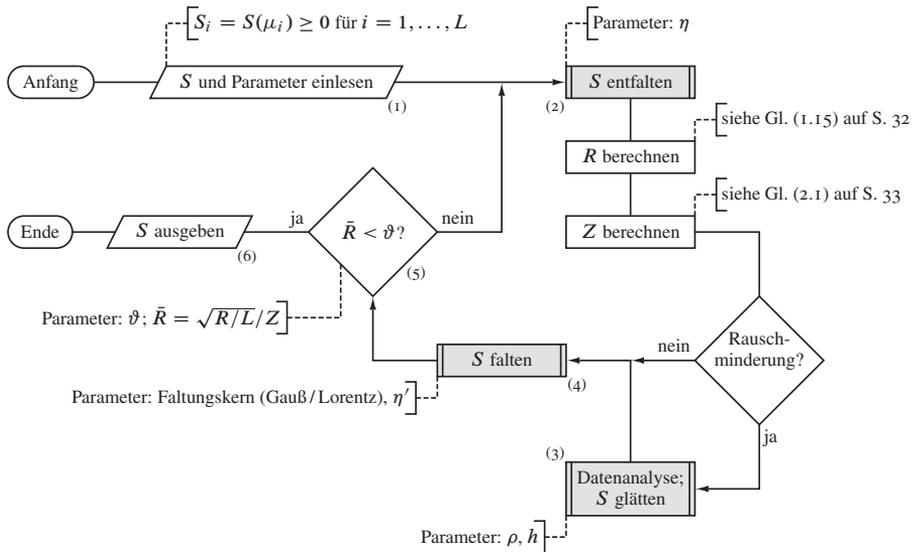


Abbildung 1.1. Programmablaufscha des alternativen Gesamtverfahrens, vergleiche Abbildung 2.9 (Abschn. 2.5 ab S. 46): Die Rückfaltung, Flußdiagrammelement (4), siehe Abbildung 2.4 (Abschn. 2.2.3 ab S. 38), wird nun optional innerhalb der Hauptschleife ausgeführt.

1.2. Alternativer Ablauf des Gesamtverfahrens. Eine naheliegende Variation des in Abschnitt 2.5 beschriebenen Programmablaufs, die Gegenstand künftiger Studien sein wird und dabei über die im Ausblick, siehe Abschnitt 5.2, erwähnten Modifikationen hinausgeht, besteht in der Vertauschung der dortigen Schritte 6 und 7, wodurch die Rückfaltung mit einem vorgegebenen Kern gegebenenfalls innerhalb der Hauptschleife ausgeführt wird.

- 6 (optional) Rückfaltungsunterprogramm, Element (4), Abschnitt 2.2.3 und Abbildung 2.4, aufrufen: (geglättetes) Endergebnis, Wahlparameter für den Kerntyp der Rückfaltung und dessen Breite η' übernehmen, erneut gefalteten Datensatz übergeben

- 7 Verfahren beenden, wenn beispielsweise gemäß Gleichung (2.2) $\bar{R} < \vartheta$; sonst zu Schritt 2 zurückkehren

Auf diesem Wege wird die zunächst (durch den iterativen Ansatz stets nur näherungsweise) vollständig erfolgende Entfaltung vermieden, welche möglicherweise die Wirksamkeit des Glättungsverfahrens durch das erneute Hervortreten der diskreten Struktur der Spektralfunktion, siehe Punkt (3) in Abschnitt 1.2.2, nachteilig beeinflusst: Effektiv tritt durch die Rückfaltung der Roh- beziehungsweise Zwischenergebnisdaten nach einem jeden Entfaltungszyklus eine *sukzessive* „Umfaltung“ von η nach η' und (optional) von der Lorentz- zur Gaußverteilung ein.

2. Probeanwendungen auf die Zustandsdichte des eindimensionalen Mottisolators

Als Anwendungsbeispiel für die im Abschnitt 1.1 des Anhanges beschriebene Rückfaltung mit einer Gaußverteilung in Kombination mit dem „Umfaltungsverfahren“ aus Abschnitt 1.2 betrachten wir in Anlehnung an Abschnitt 4.2.2 die lokale Zustandsdichte des Hubbardmodelles bei $U = 4$ in der Kettemitte. Eine umfassendere Untersuchung verschiedener Werte für die *on-site*-Wechselwirkung mit der gemäß Abschnitt 1 geänderten Entfaltungsmethode wird im Zeitschriftenbeitrag [44] vorgenommen.

Analog zu Abbildung 4.2 auf Seite 86 sind in Abbildung 2.1 die gesamte Spektralfunktion sowie die vergrößerten relevanten Ausschnitte des Bereichs der unteren und der oberen Bandkante veranschaulicht. Dabei wird, ergänzt durch die DDMRG-Rohdaten mit $\eta = 0,08$, das sukzessiv mit einem Gaußkern der Breite $\eta' = \eta/100 = 8 \cdot 10^{-4}$ in 1000 Zyklen umgefaltete Ergebnis dem bereits zuvor betrachteten Resultat gegenübergestellt, welches mit dem üblichen Lorentzkern der Breite $\eta' = \eta/10 = 8 \cdot 10^{-3}$ nach 1000 Entfaltungsschritten zurückgefaltet wurde. Ohne das bislang auftretende Bandartefakt zeigen sich nahezu fehlerfrei und mit bedeutend vergrößerter Schärfe die wiederum aus der ersten Ableitung ermittelten Stufen bei $\omega_1 = -0,638$ und bei $\omega_2 = 0,639$, woraus eine optische Lücke von $E_{\text{opt.}} = |\omega_1| + \omega_2 = 1,277$ in gutem Einklang mit dem exakten Ergebnis $E_{\text{opt.}} = 1,286$ folgt¹. Insgesamt wirkt darüber hinaus das neue Resultat subjektiv nicht so „rauh“ wie das vorherige, aber ein tiefergehender direkter Vergleich ist einerseits durch die um eine Größenordnung kleinere Rückfaltung mit einem zudem „schmaleren“ Kern, andererseits durch den (empirisch) doppelt so groß eingestellten Schwellenwert für die Rauscherkennung, $\rho = 2\eta = 0,16$, ohne weitere Prüfrechnungen schwierig.

¹Die Unsicherheit des scheinbar numerisch näheren Ergebnisses von $E_{\text{opt.}} = 1,29$ aus der Lorentzrückfaltung, siehe Abschnitt 4.2, ist durch den „breiteren“ Kern größer.

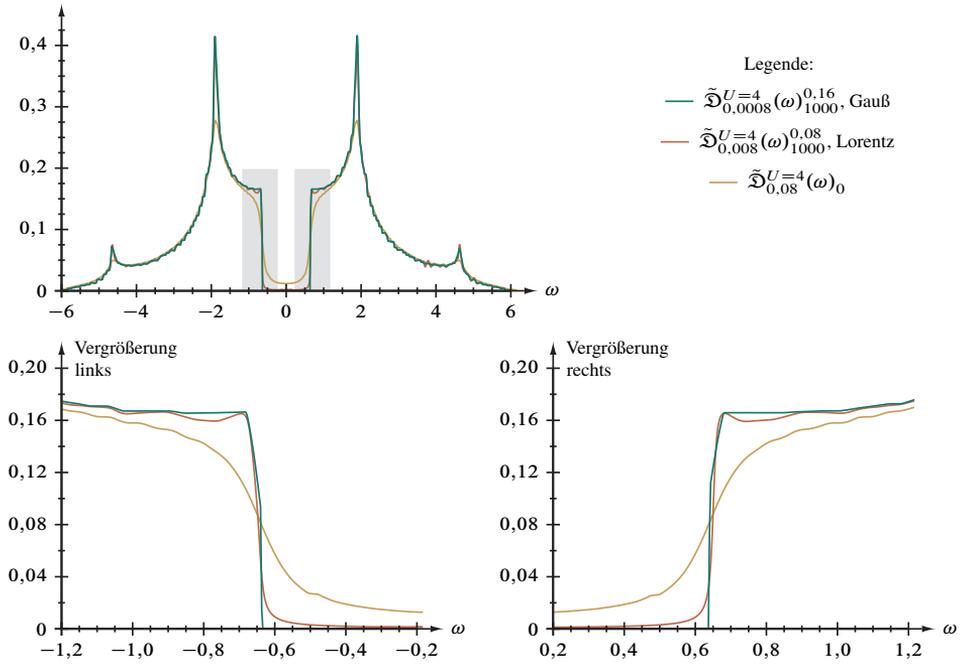


Abbildung 2.1. Vergleich von Gauß- und Lorentzrückfaltung anhand der Zustandsdichte des eindimensionalen Hubbardmodells, $\tilde{\mathcal{D}}_{0,08}^{U=4}(\omega)$.

Literatur- und Quellenverzeichnis

- [1] INGO PESCHEL, XIAOQUN WANG, MATTHIAS KAULKE, AND KAREN HALLBERG (Editors): *Density-Matrix Renormalization. A New Numerical Method in Physics*. Series: *Lecture Notes in Physics* 528. Berlin, Heidelberg: Springer, 1999.
- [2] INGO PESCHEL et al.: The Density Matrix Renormalization Group: Concepts and Applications. Part IX in: HOLGER FEHSKE, RALF SCHNEIDER, AND ALEXANDER WEISSE (Editors): *Computational Many-Particle Physics*. Series: *Lecture Notes in Physics* 739. Berlin, Heidelberg: Springer, 2008.
- [3] STEVEN R. WHITE: Density matrix formulation for quantum renormalization groups. – *Physical Review Letters* 69 (19), 2863–2866 (1992).
- [4] ERIC JECKELMANN: Density-Matrix Renormalization Group Algorithms. In [2], pages 597–619.
- [5] ERIC JECKELMANN AND HOLGER FEHSKE: Exact numerical methods for electron-phonon problems. – *Rivista del Nuovo Cimento* 30 (6), 259–292 (2007).
- [6] ULRICH SCHOLLWÖCK: The density-matrix renormalization group in the age of matrix product states. – *Annals of Physics* 326 (1), 96–192 (2011).
- [7] REINHARD M. NOACK AND SALVATORE R. MANMANA: Diagonalization- and Numerical Renormalization-Group-Based Methods for Interacting Quantum Systems. – *AIP Conference Proceedings* 789, 93–163 (2005).
- [8] TILL D. KÜHNER AND STEVEN R. WHITE: Dynamical correlation functions using the density matrix renormalization group. – *Physical Review B: Condensed matter and materials physics* 60 (1), 335–343 (1999).
- [9] SWAPAN K. PATI, SURYANARAYANA S. RAMASESHA, ZHIGANG SHUAI, AND JEAN-LUC BRÉDAS: Dynamical nonlinear optical coefficients from the symmetrized density-matrix renormalization-group method. – *Physical Review B: Condensed matter and materials physics* 59 (23), 14827–14830 (1999).
- [10] ERIC JECKELMANN, FLORIAN GEBHARD, AND FABIAN H. L. ESSLER: Optical Conductivity of the Half-Filled Hubbard Chain. – *Physical Review Letters* 85 (18), 3910–3913 (2000).
- [11] ERIC JECKELMANN: Dynamical density-matrix renormalization-group method. – *Physical Review B: Condensed matter and materials physics* 66 (4), 045114, 16 pages (2002).

- [12] ERIC JECKELMANN AND HOLGER BENTHIEN: Dynamical Density-Matrix Renormalization Group. In [2], pages 621–635.
- [13] ERIC JECKELMANN: Density-Matrix Renormalization Group Methods for Momentum- and Frequency-Resolved Dynamical Correlation Functions. – *Progress of Theoretical Physics Supplement* 176, 143–164 (2008).
- [14] KURT SCHÖNHAMMER: Luttinger Liquids: The Basic Concepts. Pages 93–136 in DIONYS BAERISWYL AND LEONARDO DEGIORGI (Editors): *Strong Interactions in Low Dimensions*. Series: *Physics and Chemistry of Materials with Low-Dimensional Structures* 25. Dordrecht: Kluwer, 2004.
- [15] ERIC JECKELMANN: Local density of states of the one-dimensional spinless fermion model. – *Journal of Physics: Condensed Matter* 25 (1), 014002, 9 pages (2013).
- [16] STEVEN R. WHITE AND IAN AFFLECK: Spectral function for the $S = 1$ Heisenberg antiferromagnetic chain. – *Physical Review B: Condensed matter and materials physics* 77 (13), 134437, 11 pages (2008).
- [17] ANDREAS WEICHELBAUM, FRANK VERSTRAETE, ULRICH SCHOLLWÖCK, JUAN IGNACIO CIRAC, AND JAN VON DELFT: Variational matrix-product-state approach to quantum impurity models. – *Physical Review B: Condensed matter and materials physics* 80 (16), 165117, 7 pages (2009).
- [18] ANDREAS HOLZNER, ANDREAS WEICHELBAUM, IAN P. MCCULLOCH, ULRICH SCHOLLWÖCK, AND JAN VON DELFT: Chebyshev matrix product state approach for spectral functions. – *Physical Review B: Condensed matter and materials physics* 83 (19), 195115, 20 pages (2011); PIET ERIC DARGEL, ANDREAS HONECKER, ROBERT PETERS, REINHARD M. NOACK, AND THOMAS PRUSCHKE: Adaptive Lanczos-vector method for dynamic properties within the density matrix renormalization group. – *ibid.* 83 (16), 161104, 4 pages (2011).
- [19] ROBERT PETERS: Spectral functions for single- and multi-impurity models using density matrix renormalization group. – *Physical Review B: Condensed matter and materials physics* 84 (7), 075139, 10 pages (2011).
- [20] FLORIAN GEBHARD, ERIC JECKELMANN, SANDRA MAHLERT, SATOSHI NISHIMOTO, AND REINHARD M. NOACK: Fourth-order perturbation theory for the half-filled Hubbard model in infinite dimensions. – *The European Physical Journal B: Condensed Matter and Complex Systems* 36 (4), 491–509 (2003).
- [21] SATOSHI NISHIMOTO AND ERIC JECKELMANN: Density-matrix renormalization group approach to quantum impurity problems. – *Journal of Physics: Condensed Matter* 16 (4), 613–625 (2004).
- [22] SATOSHI NISHIMOTO, FLORIAN GEBHARD, AND ERIC JECKELMANN: Dynamical density-matrix renormalization group for the Mott-Hubbard insulator in high dimensions. – *Journal of Physics: Condensed Matter* 16 (39), 7063–7083 (2004).
- [23] CARSTEN RAAS, GÖTZ S. UHRIG, AND FRITHJOF B. ANDERS: High-energy dynamics of the single-impurity Anderson model. – *Physical Review B: Condensed matter and materials physics* 69 (4), 041102, 4 pages (2004).

- [24] CARSTEN RAAS AND GÖTZ S. UHRIG: Spectral densities from dynamic density-matrix renormalization. – *The European Physical Journal B: Condensed Matter and Complex Systems* 45 (3), 293–303 (2005).
- [25] PETER SCHMITTECKERT: Calculating Green functions from finite systems. – *Journal of Physics: Conference Series* 220 (1), 012022, 19 pages (2010); TOBIAS ULBRICHT AND PETER SCHMITTECKERT: Tracking spin and charge with spectroscopy in spin-polarised 1D systems. – *Europhysics Letters* 89 (4), 47001, 6 pages (2010).
- [26] WILLIAM H. PRESS, SAUL TEUKOLSKY, WILLIAM T. VETTERLING, AND BRIAN P. FLANNERY: *Numerical Recipes. The Art of Scientific Computing*. In particular chapter 19. Third Edition (C++). Cambridge: Cambridge University Press, 2007.
- [27] DIANNE P. O’LEARY: *Scientific Computing with Case Studies*. Chapters 6 and 14. Philadelphia (Pennsylvania): SIAM Press, 2009.
- [28] TIM CORNWELL AND ALAN BRIDLE: *Deconvolution Tutorial*. Charlottesville (Virginia): National Radio Astronomy Observatory, 1996; PATRIZIO CAMPISI AND KAREN EGIAZARIAN (Editors): *Blind image deconvolution: theory and applications*. Boca Raton (Florida): CRC Press, 2007.
- [29] KONRAD KÖNIGSBERGER: *Analysis*. Band 1: Fünfte Auflage. Heidelberg: Springer, 2001; Band 2: Vierte Auflage. Heidelberg: Springer, 2002; KURT MEYBERG UND PETER VACHENAUER: *Höheren Mathematik*. Band 1: Sechste Auflage. Heidelberg: Springer, 2001; Band 2: Vierte Auflage. Heidelberg: Springer, 2001.
- [30] GERD CZYCHOLL: *Theoretische Festkörperphysik. Von den klassischen Modellen zu modernen Forschungsthemen*. Dritte Auflage. Heidelberg: Springer, 2008.
- [31] GABRIEL KOTLIAR AND DIETER VOLLHARDT: Strongly Correlated Materials: Insights From Dynamical Mean-Field Theory. – *Physics Today* 57 (3), 53–59 (2004).
- [32] GILBERT STRANG: *Wissenschaftliches Rechnen*. Heidelberg: Springer, 2010.
- [33] WOLFGANG DAHMEN UND ARNOLD REUSKEN: *Numerik für Ingenieure und Naturwissenschaftler*. Zweite Auflage. Berlin, Heidelberg: Springer, 2008.
- [34] HELMUTH SPÄTH: Zur Glättung empirischer Häufigkeitsverteilungen. – *Computing* 10 (4), 353–357 (1972).
- [35] SIR NEVILL FRANCIS MOTT: *Metal-Insulator Transitions*. Second Edition. London: Taylor & Francis, 1990; FLORIAN GEBHARD: *The Mott Metal-Insulator Transition. Models and Methods*. Series: *Springer Tracts in Modern Physics* 137. Berlin: Springer, 1997.
- [36] FABIAN H. L. ESSLER, FLORIAN GEBHARD, AND ERIC JECKELMANN: Excitons in one-dimensional Mott insulators. – *Physical Review B: Condensed matter and materials physics* 64 (12), 125119, 15 pages (2001).
- [37] JOHN HUBBARD: Electron Correlations in Narrow Energy Bands. – *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 276 (1365), 238–257 (1963).
- [38] ELLIOTT H. LIEB AND FA-YUEH WU: Absence of Mott Transition in an Exact Solution of the Short-Range, One-Band Model in One Dimension. – *Physical Review Letters* 20 (25), 1445–1448 (1968); Erratum. – *ibid.* 21 (3), 192 (1968);

- FABIAN H. L. ESSLER, HOLGER FRAHM, FRANK GÖHMANN, ANDREAS KLÜMPER, AND VLADIMIR E. KOREPIN: *The One-Dimensional Hubbard Model*. Cambridge: University Press, 2005.
- [39] MASAKI AZUMA, ZENJI HIROI, MIKIO TAKANO, KENJI ISHIDA, AND YOSHIO KITAOKA: Observation of a Spin Gap in SrCu_2O_3 Comprising Spin- $1/2$ Quasi-1D Two-Leg Ladders. – *Physical Review Letters* 73 (25), 3463–3466 (1994); YOUNG-JUNE KIM et al.: Resonant Inelastic X-Ray Scattering of the Holon-Antiholon Continuum in SrCuO_2 . – *ibid.* 92 (13), 137402, 4 pages (2004); YUZURU KUROSAKI, YOSHIMITSU SHIMIZU, KAZUYA MIYAGAWA, KAZUSHI KANODA, AND GUNZI SAITO: Mott Transition from a Spin Liquid to a Fermi Liquid in the Spin-Frustrated Organic Conductor $\kappa\text{-(ET)}_2\text{Cu}_2(\text{CN})_3$. – *ibid.* 95 (17), 177001, 4 pages (2005).
- [40] FABIAN H. L. ESSLER AND ALEXEI M. TSVELIK: Weakly coupled one-dimensional Mott insulators. – *Physical Review B: Condensed matter and materials physics* 65 (11), 115117, 13 pages (2002).
- [41] KARLO PENC (unveröffentlicht).
- [42] ALBERTO PAROLA AND SANDRO SORELLA: Spin-charge decoupling and the Green's function of one-dimensional Mott insulators. – *Physical Review B: Condensed matter and materials physics* 45 (22), 13156–13159 (1992).
- [43] KARLO PENC, KAREN HALLBERG, FRÉDÉRIC MILA, AND HIROYUKI SHIBA: Spectral functions of the one-dimensional Hubbard model in the $U \rightarrow +\infty$ limit: How to use the factorized wave function. – *Physical Review B: Condensed matter and materials physics* 55 (23), 15475–15488 (1997).
- [44] MARTIN PAECH AND ERIC JECKELMANN: Blind deconvolution of density-matrix renormalization-group spectra. – *Physical Review B: Condensed matter and materials physics* 89 (19), 195101, 8 pages (2014).

Datenverarbeitungsressourcen

- [45] *Mathematica*. Version 8.0.
Champaign (Illinois): Wolfram Research, Inc., 2011.
- [46] *Grace*. Version 5.1.22. © 1991–1995 by PAUL J. TURNER, Portland (Oregon), and © 1996–2008 by the Grace Development Team.
<http://plasma-gate.weizmann.ac.il/Grace>.

[47] RISC-workstation, Typ HP A9638A:

Rechner	Hewlett-Packard System 9000 J6750
Prozessoren	zwei, Typ PA-8700+
Hauptspeicher	2 GWorte (16 GBytes)
Betriebssystem	HP-UX Version B.11.11
Kompiler	ANSI-C B3899BA/B3901BA Version B.11.11.16
Stand	März 2009

[48] EPIC-server, Typ HP AH337A, bereitgestellt von Hewlett-Packard:

Rechner	Hewlett-Packard Integrity Superdome 2-16s
Prozessoren	16 zu je 8 Kernen, Typ Intel Itanium 9560
Hauptspeicher	192 GWorte (1536 GBytes)
Betriebssystem	HP-UX Version B.11.31
Kompiler	ANSI-C B9007AA Version C.11.31.07
Stand	April 2014

Graphentheoretische Störungsrechnung zum Hubbardmodell im Limes starker Wechselwirkung

Eine Vielzahl neuerer Arbeiten belegt, daß JOHN HUBBARDS *Modell stark korrelierter Elektronen im Gitterpotential eines Festkörpers* von 1963 [1, 2] für die Erforschung des kondensierten Zustandes der Materie ungebrochen Bedeutung hat [3]. Wiewohl es sich bei diesem Modell um das einfachste mögliche handelt, liegen selbst bei diesem Näherungsgrad bislang nur in wenigen Sonderfällen analytische Lösungen vor, vornehmlich jene für eindimensionale Systeme ausgehend von ELLIOTT LIEB und FA-YUEH WU auf Grundlage des Betheansatzes [4]. Die Gründe für das fortdauernde Interesse sind nahelegend: Einerseits und aus historischer Sicht eröffnet das Hubbardmodell den Zugang zu einigen fundamentalen und auch technisch ungemein bedeutsamen Effekten der Festkörperphysik – dem Magnetismus [5], der Supraleitung bei hohen Temperaturen [6] und insbesondere dem Mottübergang vom isolierenden zum leitenden Zustand [7], um nur die wichtigsten zu nennen. Andererseits ermöglicht die Reduktion auf die zwei wesentlichen Prozesse, die Metall-typische Mobilität der Elektronen im Gitter (kinetischer Term) und die dazu gegensätzliche korrelationsbedingte Isolator-typische Lokalisation der Elektronen auf den Gitterplätzen (Coulombterm), die numerische Simulation vergleichsweise großer Systeme und experimentell untersuchter chemischer Verbindungen [8]. Überdies besteht seitens der Quantenoptik ein außerordentliches Interesse an diesem Modell in seiner Formulierung für bosonische Kondensate (Phasenübergang vom Isolator zum Suprafluid) oder fermionische Kondensatmischungen in optischen Gittern [9–11].

Die vorliegende Studie widmet sich dem Ansatz einer algebraischen Lösung des Hubbardmodells auf dem *Bethegitter*, auf welchem jeder Gitterpunkt Z nächste Nachbarn hat, in den Limites *starker Wechselwirkung* und *hoher Koordinationszahl* ($Z \rightarrow \infty$) kraft einer von TOSIO KATO 1949 begründeten [12], 1959 von CLAUDE BLOCH aufgegriffenen [13] und MINORU TAKAHASHI 1977 auf das Hubbardmodell angewendeten [14] *Störungsrechnung*. Durch diese Reihenentwicklung ist es möglich, einen wichtigen bisher analytisch unlösbaren Sonderfall mit analytischer Strenge zu behandeln, ohne auf numerische Rechenverfahren ausweichen und dabei die vielfältigen Probleme endlich

genauer Algorithmen und Zahlen¹, somit nur ein genähertes Ergebnis in Kauf nehmen zu müssen.

Die *graphentheoretische* Durchführung dieser Störungsrechnung wurde von EVA KALINOWSKI zwischen 1999 und 2002 als eigenständige Methode umfassend ausgearbeitet [15–19] und im Rahmen einer ersten konzeptionellen Prüfung in hohen Ordnungen rechnergestützt auf Grundzustandseigenschaften angewendet, welche Aussagen über den Mott-Hubbard-Übergang [20] erlauben. Dieser Quanten-Phasenübergang (ohne Symmetriebruch) ist gekennzeichnet durch das Verschwinden der Bandlücke Δ , die unteres und oberes Hubbardband trennt. Oberhalb einer kritischen Wechselwirkung (kritischen Kopplung) $U > U_c$ weist das System zwei getrennte Bänder, also isolierendes (vergleiche folgende Abbildung), unterhalb metallisches Verhalten auf. Gegenstand einer Diskussion [19, 21] ist in diesem Zusammenhang noch immer, um welche Art Phasenübergang es sich handelt, und obschon sich das Quantensystem am kritischen Punkt nicht störungstheoretisch behandeln läßt, so kann man doch immerhin aus dem Ansatz für die kritische Energie $E_c(U) \sim (U - U_c)^{\tau-1}$ die kritische Kopplung U_c und den kritischen Exponenten τ abschätzen. Komplementär zur hiesigen Herangehensweise kann sich dem Übergangspunkt auch aus Richtung der Metallphase durch Störungstheorie genähert werden, folglich durch eine Behandlung des Grenzfalles *schwacher Wechselwirkung* [22, 23].

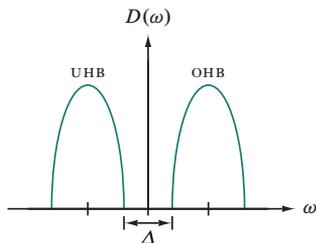


Abbildung zur Zustandsdichte $D(\omega)$ innerhalb der Mott-Hubbard-Isolatorphase. Schematisch dargestellt sind das untere (UHB) und obere (OHB) Hubbardband sowie die Bandlücke Δ . Details finden sich beispielsweise in Referenz [20].

Die hier betrachtete algebraisch-graphentheoretische Störungsrechnung nach $1/U$ liefert in jeder Ordnung für die *Grundzustandsenergie* – mathematisch formuliert ist das die *Erzeugende Funktion* der Folge der (rationalen [19, 24:A226584/A226585]) Potenzreihenoeffizienten – exakte Ergebnisse [15]. Neben deren unmittelbarem Wert für ein tieferes Verständnis des Quantensystems sind diese Ergebnisse auch Qualitätskriterium und potentiell Ergänzung für die analytisch-approximativen Methoden, beispielsweise

¹Umfassend wird auf die damit verbundenen Schwierigkeiten im ersten Teil dieser Arbeit eingegangen.

die Dynamische *mean-field*-Theorie (DMFT)², und die vielfältigen numerischen Verfahren [32]. Dem Konzept nach gehen die Möglichkeiten dieser Behandlung auf Grundlage von Graphen allerdings weit über die Berechnung der mit der Grundzustandsenergie verbundenen Systemeigenschaften hinaus, nächstliegend zur *Greenschen Funktion*. Darin besteht die Motivation für eine diesbezüglich offene und hiermit vorgelegte Neuimplementierung der Algorithmen, welche zugleich auf die Nutzung von Großrechenanlagen mit symmetrischer Multiprozessorarchitektur und linearer Speicheradressierung abzielt.

Untrennbar gehört zu diesem Schritt auch die in dieser Arbeit begonnene möglichst tiefgegründete formale Analyse dieses gesamten allgemeinen Rechenverfahrens, das unter den üblichen Methoden der Mathematischen Physik durch seine erwiesene Leistungsfähigkeit zu den beachtenswerten Neuerungen aus dem vergleichsweise jungen mathematischen Teilgebiet der *Diskreten Strukturen* zählt. Dementsprechend ist die von NEIL SLOANE begründete *on-line encyclopedia of integer sequences* (OEIS) [24] zentrales Hilfsmittel von Seiten der *Kombinatorik, Graphen- und Zahlentheorie* geworden. Einen ersten wesentlichen Schritt zu umfassenderem Verständnis stellt die für dieses Projekt aufgebaute Zusammenarbeit mit der Theoretischen Informatik dar, in deren Rahmen in der Gruppe von RITA LOOGEN am Fachbereich Mathematik und Informatik der Philipps-Universität Marburg die Diplomarbeit von GUNTHER GRUBER entstanden ist, eine eigenständige Umsetzung des Verfahrens in der funktionalen Programmiersprache Haskell [33, 79].

Dieser Abhandlung liegt folgende Gliederung zugrunde: Zunächst werden die physikalischen und mathematischen Grundlagen behandelt (Kapitel 1 und 2), dann umfassend das allgemeine Rechenverfahren sowie dessen konkrete optimierte Umsetzung einschließlich einer Problemkomplexitätsanalyse beschrieben (Kapitel 3 und 5). Eingeschoben ist die Betrachtung der Ergebnisse (Kapitel 4); abschließend wird ein Ausblick auf weitergehende Anwendungen und Verallgemeinerungen gegeben (Kapitel 6). Technische Einzelheiten zu den verwendeten Algorithmen befinden sich neben Übersichtstabellen einiger verfahrenszentraler Graphen im Anhang. Sind vor allem die Resultate und die Bewertung der DMFT-Näherung von Interesse, empfiehlt sich die Kapitelabfolge 1, 4

²Auf die DMFT und deren Verallgemeinerungen als einige der führenden Werkzeuge zum Studium korrelierter Elektronensysteme (bis hin zu Fragestellungen der Quantenchemie [25]) gehen beispielsweise die unter Referenz [26] zusammengefaßten Arbeiten ein. Wesentlich für die vorliegende Abhandlung ist der DMFT-Ansatz, welcher einen unendlich ausgedehnten Festkörper auf ein effektives selbstkonsistentes Quantenstörstellenmodell (*impurity model*) abbildet, im Falle des Hubbardmodelles das *single impurity*-Andersonmodell (SIAM). Im Sonderfall unendlicher Dimension oder Koordinationszahl – worin eine bedeutsame Überschneidung mit dieser Arbeit besteht – wird diese Abbildung zwar exakt, aber im Allgemeinen ist die Lösung des Störstellenproblems ein schwieriges Unterfangen: Bei der rekursiv auszuführenden numerischen Behandlung kommen beispielsweise (siehe dazu Teil I dieser Arbeit) NRG [27], DMRG [21, 23, 28, 29] und QMC [18, 30] als “*impurity solver*” zur Anwendung, welche innerhalb des DMFT-Schemas schwer abzuschätzende Fehler in das Ergebnis einbringen. Jenseits der Numerik kann zur Lösung der DMFT-Selbstkonsistenzgleichungen die Kato-Takahashi-Störungsrechnung genutzt werden [31].

und 6. Da es sprachlicherseits für die intensive Verwendung diskret-mathematischer Methoden auf solcherart komplexe Fragestellungen noch keine zusammenhängenden Konventionen gibt, ist angeraten, vor den Kapiteln 3 und 5 die Ausführungen in Kapitel 2 zu lesen.

I. Einleitung und physikalische Grundlagen

Gegenstand der gesamten Untersuchung auf Grundlage von EVA KALINOWSKIS Dissertation und nachfolgenden Arbeiten ist das Ein-Band-Hubbardmodell für Elektronen bei *halber Bandfüllung* und im Bereich der (paramagnetischen) *Mott-Isolatorphase* sowie dessen analytisch gelöste Vereinfachung, das *Falicov-Kimball-Modell* [34]. Gleichfalls bezieht sich unsere Untersuchung ferner auf das Bethegitter im Grenzfall unendlicher Konnektivität (Koordinationszahl); hiermit wird aufgrund Schleifenfreiheit und statistischer Unterdrückung von Mehrfachsprüngen die Rechnung entscheidend vereinfacht. Sämtlich werden diese zentralen Begriffe mit einfachen Beispielen im vorliegenden Kapitel eingeführt und im weiteren Verlauf der Abhandlung vertieft. Graphenbezogener Terminus wird dem Kapitel 2 voregreifend nur gebraucht, soweit intuitiv verständlich. Wir verwenden durchgängig die Notation aus Referenz [15], betrachten allerdings nur KATOS Störungstheorie und verzichten damit – ausreichend für die untersuchten Eigenschaften – auf die von TAKAHASHI eingeführte Γ -Transformation.

I.1. Modelle und deren Parameter; Gitter.

I.1.1. Hamiltonoperator des Hubbardmodells. Den Hamiltonoperator des Hubbardmodells (HM), bestehend aus dem kinetischen Anteil \hat{T} und dem Wechselwirkungsanteil \hat{D} , notieren wir in Teilchen-Loch-symmetrischer Formulierung als

$$\hat{H}^{\text{HM}} = \underbrace{-\frac{t}{\sqrt{Z}} \sum_{(i,j)} \sum_{\sigma=\uparrow,\downarrow} \left(\hat{c}_{i,\sigma}^+ \hat{c}_{j,\sigma} + \hat{c}_{j,\sigma}^+ \hat{c}_{i,\sigma} \right)}_{=: \hat{T}} + \underbrace{U \sum_i \left(\hat{n}_{i,\uparrow} - \frac{1}{2} \right) \left(\hat{n}_{i,\downarrow} - \frac{1}{2} \right)}_{=: \hat{D}}, \quad (\text{I.1})$$

worin $\hat{c}_{i,\sigma}^+$ und $\hat{c}_{i,\sigma}$ für den Erzeugungs- und den Vernichtungsoperator für ein Elektron mit dem Spin $\sigma = \uparrow, \downarrow$ am Gitterplatz i , $\hat{n}_{i,\sigma} = \hat{c}_{i,\sigma}^+ \hat{c}_{i,\sigma}$ für den Besetzungszahloperator, $t = t_\uparrow = t_\downarrow$ für die Hüpfamplitude zwischen benachbarten Gitterplätzen, U für die intraatomare Coulombabstoßung und Z für die Anzahl der Paare³ (i, j) nächster Nachbarn (auch *Koordinationszahl* genannt, siehe Abschnitt 2.1.2) stehen. Für die Anzahl an Elektronen jeder Spinsorte gilt $N_\sigma = \sum_i \hat{n}_{i,\sigma}$, für deren Gesamtzahl $N = \sum_\sigma N_\sigma$.

³Wir weichen hierin zugunsten einer später einheitlichen Schreibung von n -Tupeln, siehe Abschnitt 2.1, vom Standard ab, in welchem Gitterplatzpaare in spitzen Klammern notiert werden.

Zwecks vereinfachter Schreibweise setzen wir künftig $t = 1$, indem wir, soweit nicht anders angegeben, formal alle Energien in Einheiten der Hüpfstärke messen.

1.1.2. Hamiltonoperator des Falicov-Kimball-Modells. Unter der Annahme, nur eine der beiden Spinspezies (beispielsweise und im weiteren Verlauf $\hat{c}_i^{(+)}$ mit \uparrow) sei durch $t = t_\uparrow = 1$ auf dem Gitter beweglich, während die andere (entsprechend $\hat{d}_i^{(+)}$ mit \downarrow) durch $t_\downarrow = 0$ an ihren Ausgangsgitterplätzen fixiert bleibe, vereinfacht sich Gleichung (1.1) zum Falicov-Kimball-Modell (FKM),

$$\hat{H}^{\text{FKM}} = -\frac{t}{\sqrt{Z}} \sum_{(i,j)} \left(\hat{c}_i^+ \hat{c}_j + \hat{c}_j^+ \hat{c}_i \right) + U \sum_i \left(\hat{c}_i^+ \hat{c}_i - \frac{1}{2} \right) \left(\hat{d}_i^+ \hat{d}_i - \frac{1}{2} \right), \quad (1.2)$$

für welches die kritische Wechselwirkung $U_c = 2$ und für $U > U_c$ insbesondere die Grundzustandsenergie pro Gitterplatz,

$$E_0^{\text{FKM}}(U) = -\frac{1}{4} \frac{1}{U}, \quad (1.3)$$

exakt bekannt ist [15, 19], vergleiche Abschnitt 4.1. Auch für die maschinelle Reihenentwicklung der Grundzustandsenergie des Hubbardmodells ist diese Erkenntnis äußerst bedeutsam, da nämlich die analog am Falicov-Kimball-Modell ausgeführte Störungsrechnung in $1/U$ für alle Ordnungen über der ersten – sofern das Programm fehlerfrei arbeitet – keinen Beitrag zu E_0^{FKM} ergeben darf.

Die in den Gleichungen (1.1) und (1.2) im Anschluß an die Vorarbeiten explizit berücksichtigte Symmetrie zwischen Teilchen und Löchern findet im Folgenden keine Beachtung; daher können wir

$$\hat{D} \longrightarrow U \sum_i \hat{n}_{i,\uparrow} \hat{n}_{i,\downarrow} \quad (1.4)$$

notieren.

1.1.3. Bethegitter. Wir wählen für die Modelle (1.1) und (1.2) das Bethegitter, benannt nach HANS BETHE und von ihm 1935 als Modellstruktur in die Festkörpertheorie eingeführt [35, 36]. Dieses besitzt baumartigen Charakter, zwischen zwei Gitterplätzen existiert also genau ein Hin- und ein identischer Rückweg; Schleifen zu gehen ist unmöglich, und entsprechende Terme verschwinden in Rechnungen. *Translationsperiodizität* wird durch die unendliche Ausdehnung garantiert, und durch die allen Gitterplätzen gemeinsame Koordinationszahl $Z \rightarrow \infty$ haben Mehrfachsprünge zwischen einem Gitterplatzpaar (i, j) von i nach j ein mit $1/Z$ verschwindendes statistisches Gewicht. Eine in den Kontext der graphentheoretischen Begriffe eingebettete Definition des Bethegitters findet sich in Abschnitt 2.2.1.

1.2. Relevante Grundzustandseigenschaften. Unsere Aufmerksamkeit wollen wir unabhängig von der späteren formalen Analyse zunächst auf den Mottübergang richten; daher sind neben der Grundzustandsenergie, bezogen auf einen der L Gitterplätze,

$$E_0 = \frac{1}{L} \langle \psi_0 | \hat{H} | \psi_0 \rangle \quad (1.5)$$

der kritische Anteil der Energie E_c , die kritische Wechselwirkung U_c und der daraus abgeleitete kritische Exponent τ ,

$$E_c(U) \sim (U - U_c)^{\tau-1}, \quad (1.6)$$

unsere ersten Ziele. Die Bedeutung dieser Erkenntnisse geht aber über deren physikalische Relevanz hinaus: Weil methodisch die DMFT der Störungsrechnung gegenübersteht, stellen die algebraisch-exakten Ergebnisse letzterer auch ein Genauigkeitskriterium für numerisch-nähernde Lösungsverfahren der DMFT-Gleichungen⁴, insbesondere DDMRG [21] und QMC [18], bereit.

1.3. Lösung des Eigenwertproblems; Ausgangszustände. In dieser Arbeit betrachten wir den Fall $U \gg t$, starke Elektron-Elektron-Wechselwirkung bewirkt durch starke Elektron-Gitter-Kopplung⁵, und beschränken uns auf den halbgefüllten Fall bei Temperatur 0, den ($S_z = 0$)-Grundzustand mit $N_\uparrow = N_\downarrow$ bei $N = L$ Elektronen auf L Gitterplätzen. Unter starker Kopplung verstehen wir insbesondere $U > U_c$, also daß es sich bei dem betrachteten Quantensystem um einen Mott-Hubbard-Isolator handelt.

Die Behandlung folgt Referenz [15]; dort wird gezeigt, daß der Grundzustand des halbgefüllten Hubbardmodells, Gleichung (1.1), Ordnung für Ordnung in der Störungsrechnung in $1/U$ wie bei $U = \infty$ entartet bleibt. Dabei ist zu beachten, daß dies nur auf dem Bethegitter, bei $Z \rightarrow \infty$ und bei Annahme eines paramagnetischen Grundzustandes gilt; zudem beträgt die (homogene) Ladungsdichte genau ein Elektron pro Gitterplatz.

Wir bezeichnen nun mit \hat{P} den Projektor auf diesen entarteten Grundzustandsraum und mit $|\phi_0\rangle$ irgend einen einzelnen Eigenzustand des Problems bei $U = \infty$, welchen wir im Weiteren *Ausgangszustand* nennen. Dann ist $\hat{P}|\phi_0\rangle$ proportional zu einem Grundzustand und

$$E_0 = \frac{1}{L} \frac{\langle \phi_0 | \hat{P} \hat{H} \hat{P} | \phi_0 \rangle}{\langle \phi_0 | \hat{P} \hat{P} | \phi_0 \rangle} = \frac{1}{L} \frac{\langle \phi_0 | \tilde{H} | \phi_0 \rangle}{\langle \phi_0 | \hat{P} | \phi_0 \rangle} \quad (1.7)$$

die Grundzustandenergie. Darin wird $\hat{P} \hat{H} \hat{P} = \tilde{H}$ der effektive Hamiltonoperator, siehe Referenz [15]. Wir beachten $\hat{P}^2 \equiv \hat{P}$ und entnehmen \hat{P} gleichfalls aus Arbeit [15];

⁴Der Mottsche Metall-Isolator-Übergang (*metal-insulator transition*, MIT) im Hubbardmodell wurde eingehend mit der Dynamischen *mean-field*-Theorie studiert [21, 26–28, 30]. Auf dem Bethegitter mit $Z \rightarrow \infty$ läßt die DMFT mit zunehmendem U einen Quantenphasenübergang erster Ordnung von einer *Fermiflüssigkeit* zu einem Mottisolator erkennen: Der Grundzustand ist bis zu einer kritischen Wechselwirkung U_c^{MIT} metallisch (gemäß QMC-Rechnung [30] bei $U_c^{\text{MIT}} \approx 5,8$), darüber isolierend. Bei bis zu einem Wert $U_c < U_c^{\text{MIT}}$ abnehmender Kopplung, bei welchem sich die Mott-Hubbard-Lücke schließt, bleibt der Mottisolator-Zustand *metastabil*, sodaß dieser die Systemeigenschaften für $U_c < U < U_c^{\text{MIT}}$ beeinflusst; entsprechende Beobachtungen sollten an realen Systemen mit Hilfe zeitaufgelöster Spektroskopie möglich sein.

⁵Im *Limes starker Kopplung* geht das Hubbardmodell durch eine Störungsentwicklung zweiter Ordnung in t , bezogen auf das unterste Energieniveau, in das *t-J-Modell* über; bei Halbfüllung entsteht aus diesem das *Heisenbergmodell*. Daher werden bei starker Abstoßung die niedrigliegenden elementaren Anregungen des Hubbardmodells durch das antiferromagnetische Heisenbergmodell beschrieben.

abschließend mitteln wir über die Ausgangszustände,

$$E_0 = \text{Sp} \left(\frac{1}{L} \frac{\langle \phi_0 | \tilde{H} | \phi_0 \rangle}{\langle \phi_0 | \hat{P} | \phi_0 \rangle} \right), \quad (1.8)$$

und bezeichnen das Mittel symbolisch mit

$$\text{Sp}(\hat{O}) = \frac{\sum_{\phi_0} \langle \phi_0 | \hat{O} | \phi_0 \rangle}{\sum_{\phi_0} \langle \phi_0 | \phi_0 \rangle} := \langle \hat{O} \rangle. \quad (1.9)$$

1.4. Reihentwicklung nach der Wechselwirkung.

1.4.1. *Allgemein.* Wir greifen auf die in Arbeit [15] ausgeführte Entwicklung von \hat{P} in $1/U$ zurück,

$$\hat{P} = -\hat{S}^0 - \sum_{n=1}^{\infty} \frac{1}{U^n} \sum_{\kappa_r \geq 0} \hat{S}^{\kappa_1} \hat{T}^{\kappa_2} \hat{T} \dots \hat{T}^{\kappa_{n+1}}. \quad (1.10)$$

Hierin nehmen die $n+1$ auftretenden κ_r positiv-ganzzahlige Werte mit der Nebenbedingung $\kappa_1 + \kappa_2 + \dots + \kappa_{n+1} = n$ an.⁶ Mit \hat{P}_d als Projektor auf d Doppelbesetzungen ist der Operator \hat{S} folgendermaßen definiert:

$$\hat{S}^\kappa := \begin{cases} -\hat{P}_0 & \text{für } \kappa = 0 \\ \sum_{d>0} \frac{1}{(-d)^\kappa} \hat{P}_d & \text{für } \kappa \geq 1 \end{cases}. \quad (1.11)$$

Der Ausdruck für die Grundzustandsenergie nimmt nun sowohl für das Hubbard- wie auch das Falicov-Kimball-Modell die endgültige Form

$$E_0(U) = \sum_{n=1}^{\infty} \frac{1}{U^n} E_0^{(n)} = \frac{1}{L} \left\langle \frac{\sum_{n=1}^{\infty} \frac{1}{U^n} (\tilde{\mathcal{J}}^{(n)} + \tilde{\mathcal{D}}^{(n)})}{1 + \sum_{n=1}^{\infty} \frac{1}{U^n} \tilde{\mathcal{N}}^{(n)}} \right\rangle \quad (1.12)$$

⁶Die Lösung dieses kombinatorischen Problems können wir, vorgehend auf die rechenmaschinelle Entwicklung für prinzipiell beliebige Ordnungen von n im Abschnitt 3.3, als „Menge aller Kombinationen mit Wiederholung von $n+1$ Objekten zur Klasse n “ mit den Tupeln der \hat{S} -Exponenten wie folgt angeben:

$$\{(\kappa_1, \kappa_2, \dots, \kappa_{n+1}) \mid \kappa_r \in [0, n] \wedge \sum_{r=1}^{n+1} \kappa_r = n\}.$$

an, in welcher – vorteilhaft auch für die weitere rechnergestützte Verarbeitung zerlegt – folgende Bestandteile abgekürzt wurden:

$$\text{„}\mathcal{T}\text{-Produkt“: } \bar{\mathcal{T}}^{(n)} = \sum_{s=0}^n \langle \phi_0 | \hat{S}^{(n-s)} \hat{T} \hat{S}^{(s)} | \phi_0 \rangle, \quad (1.13)$$

$$\text{„}\mathcal{D}\text{-Produkt“: } \bar{\mathcal{D}}^{(n)} = \sum_{s=1}^n \langle \phi_0 | \hat{S}^{(n+1-s)} \hat{D} \hat{S}^{(s)} | \phi_0 \rangle, \quad (1.14)$$

$$\text{„Nenner“: } \bar{\mathcal{N}}^{(n)} = \langle \phi_0 | \hat{S}^{(n)} | \phi_0 \rangle, \quad (1.15)$$

$$\text{„}\mathcal{S}\text{-Faktor“: } \hat{S}^{(n)} = - \sum_{\substack{\kappa_1 + \kappa_2 + \dots + \kappa_{n+1} = n \\ \kappa_r \geq 0 \wedge 1 \leq r \leq n+1}} \hat{S}^{\kappa_1} \hat{T} \hat{S}^{\kappa_2} \hat{T} \dots \hat{T} \hat{S}^{\kappa_{n+1}}. \quad (1.16)$$

Infolge des Zusammenhangs

$$\frac{1}{L} \langle \hat{D} \rangle = U \frac{\partial}{\partial U} E_0(U) \Rightarrow \bar{\mathcal{D}}^{(n)} = -n E_0^{(n)} \quad (1.17)$$

brauchen wir das \mathcal{D} -Produkt nicht auszurechnen, und somit ergibt sich noch eine wesentliche Vereinfachung der Gleichung (1.12):

$$E_0(U) = \frac{1}{L} \left\langle \frac{\sum_{n=1}^{\infty} \frac{1}{n+1} \frac{1}{U^n} \bar{\mathcal{T}}^{(n)}}{1 + \sum_{n=1}^{\infty} \frac{1}{U^n} \bar{\mathcal{N}}^{(n)}} \right\rangle. \quad (1.18)$$

1.4.2. Beispiel. Die Grundzustandsenergie pro Teilchen in der Ordnung U^{-1} berechnet sich nun für das Hubbardmodell im Falle halber Bandfüllung (HBF) aus Gleichung (1.18) zu

$$E_0^{\text{HBF},(1)} = \frac{1}{2} \frac{1}{L} \left\langle \left(\hat{S}^{(0)} + \frac{1}{U} \hat{S}^{(1)} \right) \hat{T} \left(\hat{S}^{(0)} + \frac{1}{U} \hat{S}^{(1)} \right) \right\rangle_{|\phi_{(U^{-1})}} \quad (1.19-1)$$

$$= \frac{1}{2} \frac{1}{L} \left\langle \hat{S}^{(1)} \hat{T} \hat{S}^{(0)} + \hat{S}^{(0)} \hat{T} \hat{S}^{(1)} \right\rangle \quad (1.19-2)$$

$$= \frac{1}{L} \left\langle \underbrace{\hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0}_{\text{„Prozeß“}} \right\rangle = -\frac{1}{L} \langle \hat{T} \hat{P}_1 \hat{T} \rangle \quad (1.19-3)$$

$$\begin{aligned}
&= -\frac{1}{L} \left\langle \frac{1}{Z} \sum_{(i,j)} \sum_{\sigma} (\hat{c}_{i,\sigma}^+ \hat{c}_{j,\sigma} + \hat{c}_{j,\sigma}^+ \hat{c}_{i,\sigma}) \hat{P}_1 \right. \\
&\quad \left. \sum_{(i',j')} \sum_{\sigma'} (\hat{c}_{i',\sigma'}^+ \hat{c}_{j',\sigma'} + \hat{c}_{j',\sigma'}^+ \hat{c}_{i',\sigma'}) \right\rangle \quad (1.19-4)
\end{aligned}$$

unter Ausnutzung der Translationsinvarianz

$$= -\frac{1}{L} \left\langle \frac{1}{Z} \sum_{(i,j)} \sum_{\sigma} \sum_{\sigma'} \hat{c}_{i,\sigma}^+ \hat{c}_{j,\sigma} \cdot \hat{c}_{j,\sigma'}^+ \hat{c}_{i,\sigma'} \cdot 2 \right\rangle \quad (1.19-5)$$

mit $\sum_{(i,j)} 1 = \frac{LZ}{2}$ und $i \neq j$ sowie $\sigma = \sigma'$

$$= - \left\langle \sum_{\sigma} \hat{c}_{i,\sigma}^+ \hat{c}_{j,\sigma} \cdot \hat{c}_{j,\sigma}^+ \hat{c}_{i,\sigma} \right\rangle \quad (1.19-6)$$

$$= - \left\langle 2 \cdot \underbrace{\hat{c}_{i,\uparrow}^+ \hat{c}_{i,\uparrow}}_{=\frac{1}{2}} \cdot \underbrace{\hat{c}_{j,\uparrow} \hat{c}_{j,\uparrow}^+}_{=\frac{1}{2}} \right\rangle = -\frac{1}{2}, \quad (1.19-7)$$

und wir schreiben diese in

$$E_0^{(1),\text{HM}}(U) = -\frac{1}{2} \frac{1}{U} \quad (1.20)$$

als erstes Glied der $1/U$ -Reihe. Deutlich zeigt die Rechnung, wie sehr die Mittelung über die Ausgangszustände die Überlegungen vereinfacht: In Gleichungszeile (1.19-5) kann man die Summe über Paare nächster Nachbarn unter dem Mittel unabhängig von (i, j) ausführen.⁷

1.5. Prozesse, Bethegitterausschnitte, Sequenzen.

1.5.1. Hüpfprozesse. Eine Operatorenabfolge, beispielsweise die mit $\hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0$ kürzeste mögliche in Gleichungszeile (1.19-3), nennen wir verallgemeinert künftig einen *Elektronentransferprozeß* oder, verkürzend, *(Hüpf)prozeß*. Die konsequente Entwicklung des effektiven Hamiltonoperators, definiert in Gleichung (1.7),

$$\tilde{H}^{\text{HM}} = \tilde{h}_0 + \frac{1}{U} \tilde{h}_1 + \frac{1}{U^2} \tilde{h}_2 + \frac{1}{U^3} \tilde{h}_3 + \dots \quad (1.21)$$

⁷Die in Abschnitt 1.3 angeführte Erhaltung der Singulett-Grundzustandsentartung ist eine entscheidende Voraussetzung für die exakte Berechenbarkeit, siehe beispielhaft Gleichungszeile (1.19-7), des allgemeinen Erwartungswertes der Ordnung U^{-n}

$$\langle \phi_0 | \hat{c}_{i_1, \sigma_1}^+ \hat{c}_{i_2, \sigma_2} \hat{c}_{i_3, \sigma_3}^+ \hat{c}_{i_4, \sigma_4} \dots \hat{c}_{i_{2n+1}, \sigma_{2n+1}} \hat{c}_{i_{2n+2}, \sigma_{2n+2}} | \phi_0 \rangle$$

und somit der Reihenkoeffizienten. Dieser Sachverhalt kommt nochmals in Abschnitt 6.1.2 zur Sprache.

nach $1/U$ liefert⁸ für beliebige Gitterfüllungen ($2N \neq L$) bis einschließlich dritter Ordnung folgende, in jedem Term links mit \hat{P}_0 beginnende und rechts mit diesem Projektor endende Prozesse [15]:

$$\tilde{h}_0 = \hat{P}_0 \hat{T} \hat{P}_0, \quad (1.22)$$

$$\tilde{h}_1 = \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0, \quad (1.23)$$

$$\tilde{h}_2 = \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 - \frac{1}{2} \left(\hat{P}_0 \hat{T} \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 + \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 \hat{T} \hat{P}_0 \right) \quad (1.24)$$

und

$$\tilde{h}_3 = \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{S}^1 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \quad (1.25-1)$$

$$- \frac{1}{2} \left(\hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 + \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \right) \quad (1.25-2)$$

$$- \frac{1}{2} \left(\hat{P}_0 \hat{T} \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 + \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \hat{T} \hat{P}_0 \right) \quad (1.25-3)$$

$$+ \hat{P}_0 \hat{T} \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 + \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 \hat{T} \hat{P}_0 \quad (1.25-4)$$

$$+ \frac{1}{2} \left(\hat{P}_0 \hat{T} \hat{P}_0 \hat{T} \hat{P}_0 \hat{T} \hat{S}^3 \hat{T} \hat{P}_0 + \hat{P}_0 \hat{T} \hat{S}^3 \hat{T} \hat{P}_0 \hat{T} \hat{P}_0 \hat{T} \hat{P}_0 \right). \quad (1.25-5)$$

Wie leicht gezeigt werden kann [15], entfallen bei Halbfüllung des Gitters alle Terme mit einer ungeraden Anzahl an Elektronentransferoperatoren \hat{T} zwischen zwei Projektoren \hat{P}_0 auf den Raum anregungsfreier Ausgangszustände⁹. Unmittelbar folgt daraus, daß die geraden Ordnungen in U nicht zur Grundzustandsenergie beitragen, etwa (1.22) und (1.24) und sich die Prozeßanzahl in ungeraden Ordnungen drastisch verringert. Beispielsweise verkürzt sich Gleichung (1.25) durch Wegfall der Zeilen (1.25-3) bis (1.25-5) auf

$$\tilde{h}_3^{\text{HBF}} = \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{S}^1 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \quad (1.26-1)$$

$$- \frac{1}{2} \left(\hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 + \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \right). \quad (1.26-2)$$

Der Hinweis auf halbe Bandfüllung wird bis auf Weiteres unterlassen, sofern keine explizite Abgrenzung vom allgemeinen Fall notwendig ist.

1.5.2. Gitterparkettierung. Wie zuvor festgestellt, enthält in n ter Ordnung der Störungsrechnung jeder Term von \tilde{h}_n , also jeder Hüpfprozeß $n + 1$ Transferoperatoren; folglich muß in Gleichung (1.1) die äußere Summe über i_1, i_2, \dots, i_{n+1} und j_1, j_2, \dots, j_{n+1} ausgeführt werden. Auf dem Bethegitter sind zur abschließend geforderten Rückkehr in den

⁸Bis auf eine Konstante, etwa $U(L - 2N)/4$ im Falle berücksichtigter Teilchen-Loch-Symmetrie

⁹Durch das Bethegitter bedingt ist ein Zwischenzustand ohne Doppelbesetzungen, welcher nur nach einer geraden Anzahl an Hüpfritten auftreten kann, stets der Ausgangszustand.

Ausgangszustand geradzahlig viele Hüpf Schritte vonnöten, wobei der Grenzfall $Z \rightarrow \infty$ Mehrfachsprünge zwischen zwei Gitterplätzen „unterbindet“. Zum ersten folgt auch daraus nochmals anschaulich, daß keine gerade Ordnungen in $1/U$ auftreten können; zum zweiten resultiert, daß genau $(n + 1)/2$ zusammenhängende Gitterplatzpaare (i, j) für einen „erlaubten“ Hüpfprozeß benötigt werden, wobei jeweils ein Elektronensprung von i nach j durch einen zweiten in Gegenrichtung, insbesondere durch den gleichen Spin, nicht jedoch zwangsläufig das selbe¹⁰ Elektron, „ausgeglichen“ werden muß. Daß nur zusammenhängende Prozesse beiträgend sind (*linked-cluster-Theorem*) [15], folgt jedoch auch aus der Erkenntnis, mit der Grundzustandsenergie eine extensive Eigenschaft zu betrachten.

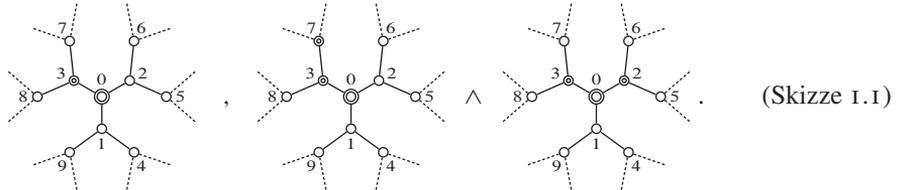
n	k	Gitterausschnitte (Kantentupel in Mengenklammern)
1	2	$\{(0, 3)\} \mapsto \textcircled{\circ}$
3	3	$\{(0, 3), (3, 7)\} \mapsto \textcircled{\circ} \wedge \{(0, 3), (0, 2)\} \mapsto \textcircled{\circ} \textcircled{\circ}$
5	4	$\textcircled{\circ} \wedge \textcircled{\circ} \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge \textcircled{\circ} \textcircled{\circ}$
7	5	$\textcircled{\circ} \wedge \textcircled{\circ} \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge \textcircled{\circ} \textcircled{\circ} \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge \textcircled{\circ} \textcircled{\circ} \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge \textcircled{\circ} \textcircled{\circ} \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge (\textcircled{\circ} \vee \textcircled{\circ}) \wedge \textcircled{\circ} \textcircled{\circ}$

Tabelle 1.1. Alle Möglichkeiten für die Bildung von Bethegitterausschnitten mit $k = (n + 3)/2$ Gitterplätzen bis einschließlich Ordnung $n = 7$ der Störungsrechnung; strukturell gleiche sind in runden Klammern zusammengefaßt [19, 24:A000081]. Die Gitterplatznummern in den Paarmengen ergeben sich aus Skizze 1.1.

An jedem Prozeß sind mithin $n + 1$ Gitterplätze beteiligt, von welchen aber $(n - 1)/2$ identisch sein müssen, weil auf dem Bethegitter keine Schleifenwege möglich sind. Somit erstreckt sich ein Prozeß über $k = (n + 3)/2$ Gitterplätze, siehe folgendes Beispiel für $Z = 3$ (Gitterplatznumerierung beliebig). Von den k prozeßrelevanten Gitterplätzen werden einer über das gesamte translationsinvariante Gitter, Faktor L , und $(n + 1)/2$ über nächste Nachbarn, Faktor $Z^{(n+1)/2}$, summiert; diese Faktoren kürzen sich, siehe

¹⁰Kehren Elektronen gleichen Spins am Ende eines Hüpfprozesses nicht wieder auf den jeweiligen Ausgangsgitterplatz zurück, so ändert sich bei ungeradzahli ger Vertauschung – bedingt durch die Antisymmetrie der fermionischen Wellenfunktion – das Zustandsvorzeichen, siehe Abschnitt 3.5.

Beispiel für $n = 1$ in Abschnitt 1.4.2. Jenen Gitterplatz, an welchem die Elektronenpropagation beginnt, kennzeichnen wir mit dem Symbol \odot ; alle anderen Beteiligten markieren wir mit \circ . Einsichtigerweise ist in erster Störungsrechnungsordnung (links), ein Gitterplatzpaar, nur ein Strukturtyp mit $k = 2$ Gitterplätzen möglich, während in dritter Ordnung (rechts), zwei Paare, bereits zwei strukturell verschiedene Möglichkeiten zur Wahl von $k = 3$ Gitterplätzen bestehen (das heißt zwei Terme in der Summe über Gitterplätze):



Alle aus dem Gitter „kombinatorisch extrahierten“ Wahlmöglichkeiten (Gitterausschnitte) bilden sämtlich umfassend in Abschnitt 2.2.2 erläuterten *Butcherbäume*, deren *Wurzel* der Ausgangsgitterplatz ist, siehe Tabelle 1.1 [19,24:A000081]¹¹. Jeder Butcherbaum entspricht also einem Term in der Summe über Gitterplätze, der zu einem nichtverschwindenden Beitrag zur Energie führt und umgekehrt. Bei endlicher Koordinationszahl gilt dies nur für Bäume mit höchstens $k = Z + 1$ Gitterplätzen; insbesondere der letzte Ausschnitt mit $k = 5$ in Tabelle 1.1 läßt sich beispielsweise nicht auf obigem Gitter mit $Z = 3$ einpassen. Man beachte ferner, daß beginnend mit $k = 4$ von vielen Gitterausschnitten „Symmetrieverwandte“ – oder genauer, wie im Abschnitt 2.1.12 diskutiert, *Isomorphismen* – existieren, von welchen nur ein Vertreter berücksichtigt werden darf.

Neben der kombinatorischen Herangehensweise eröffnet sich zu den Butcherbäumen auch anschaulicher Zugang aus den in $\hat{c}^+ \hat{c}$ -Paare „expandierten“ \hat{T} -Operatoren der Hamiltonoperatorterme, vergleiche Abschnitt 1.5.1, also der expliziten Ausführung der Summe über nächste Nachbarn. Ohne Berücksichtigung der Elektronenspins oder unter Annahme des Falicov-Kimball-Modells läßt sich Zeile $n = 1$ aus Tabelle 1.1 (siehe unten, vergleiche auch Skizze 1.1) beispielsweise mit Gleichung (1.23) als

$$\{(0, 3)\} \leftarrow \hat{c}_3^+ \hat{c}_0 \dots \hat{c}_0^+ \hat{c}_3$$

sowie Zeile $n = 3$ mit Gleichung (1.25) als

$$\begin{aligned} \{(0, 3), (3, 7)\} &\leftarrow \hat{c}_3^+ \hat{c}_0 \dots \hat{c}_7^+ \hat{c}_3 \dots \hat{c}_3^+ \hat{c}_7 \dots \hat{c}_0^+ \hat{c}_3 \quad \text{und} \\ \{(0, 3), (0, 2)\} &\leftarrow \hat{c}_3^+ \hat{c}_0 \dots \hat{c}_0^+ \hat{c}_2 \dots \hat{c}_2^+ \hat{c}_0 \dots \hat{c}_0^+ \hat{c}_3 \end{aligned}$$

¹¹Intuitiv erschließt sich, daß das $(Z \rightarrow \infty)$ -Bethegitter von allen Butcherbäumen beliebiger Ordnung vollständig nichtüberlappend überdeckt, *parkettiert* werden kann, folglich also kein Bereich auf dem Bethegitter existiert, der nicht erfaßt wird. Diese Erkenntnis gilt auch für das im folgenden Abschnitt 1.5.3 behandelte gefärbte (spinbelegte) Gitter.

schreiben. Faktisch führt jede „Permutation“ der jeweiligen Erzeuger-Vernichter-Paare zum gleichen, hier durch seine *Kantenmenge*¹² vertretenen Baum¹³; konventionsmäßig muß allerdings der erste Elektronensprung von der Wurzel zu einem Nachbargitterplatz erfolgen. Dieser Sachverhalt wird in Abschnitt 3.4 noch eingehend vertieft.

1.5.3. *Spinbelegte Gitterausschnitte.* Innerhalb der zuvor erläuterten Summe über sämtliche Kombinationen von Gitterplatzpaaren, welche die Butcherbäume „generiert“, ist nach Gleichung (1.1) die Summe über alle Spinkombinationen auszuführen. Bei $k = (n + 3)/2$ beteiligten Gitterplätzen sind mit $\sigma = \uparrow, \downarrow$ höchstens 2^k Spinbelegungen möglich; effektiv reduziert sich die Anzahl der zu berechnenden Beiträge jedoch zumeist durch die Tatsache, daß mehrere Spinbelegungen durch „Symmetrieoperationen“ – oder genauer, wie im Abschnitt 2.1.15 umfassend erläutert, durch *farbbezogene Isomorphismen (Isochromismen)* – in einander überführt werden können und daher nur ein Repräsentant zu berücksichtigen ist.

n	k	Spinbelegte Gitterausschnitte (Kantentupel in Mengenklammern)
1	2	$\{(0, 2)\} \mapsto \text{⬆} \wedge \{(0, 3)\} \mapsto \text{⬆} \wedge \{(3, 0)\} \mapsto \text{⬆} \wedge \{(3, 7)\} \mapsto \text{⬆}$
3	3	

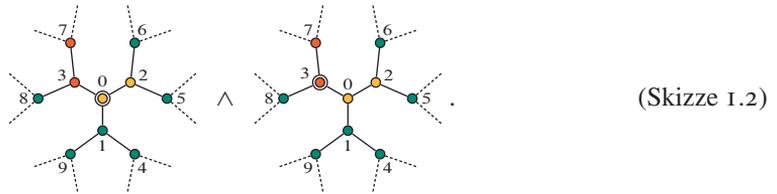
Tabelle 1.2. Alle Möglichkeiten für die Bildung von mit Elektronen besetzten (Farbkonvention: $\color{orange}\blacksquare \mapsto \uparrow$ -Spin, $\color{red}\blacksquare \mapsto \downarrow$ -Spin) Bethegitterausschnitten mit $k = (n + 3)/2$ Gitterplätzen in den Störungsordnungen $n = 1$ und $n = 3$; hinsichtlich der Spinbelegung gleiche sind in runden Klammern zusammengefaßt [24:A038055]. Die Gitterplatznummern in den Paarmengen ergeben sich aus Skizze 1.2.

Das folgende Beispiel für einen möglichen Spinzustand bedient sich wiederum des in Skizze 1.1 verwendeten Bethegitters mit $Z = 3$, welches nun, in Vertretung der den Elektronenspin symbolisierenden Pfeile, farbkodierte Gitterplätze aufweist. Dabei sind

¹²Die Menge der Kanten ist im Allgemeinen ungeordnet, siehe Abschnitt 2.1.1. Man beachte auch, daß die Tupelformulierung der Kanten die Richtung des Elektronensprunges reflektiert. Die jedoch in diesem und den folgenden beiden Abschnitten genutzte komprimierte Schreibweise der Kantenmengen enthält implizit auch die jeweilige „Gegenkante“ $\overline{(i, j)} = (j, i)$, siehe Fußnote 19 und im Besonderen Abschnitt 3.4.3.

¹³Eine Beweisführung ist aus der Baumdefinition, siehe Abschnitt 2.1.5, möglich: Jeder Baum \mathcal{T} ist vollständig durch seine Kantenmenge $E(\mathcal{T})$ gegeben, da sich aus ihr eindeutig auf seine Eckenmenge $V(\mathcal{T})$ schließen läßt, $E(\mathcal{T}) \mapsto V(\mathcal{T})$; für einen allgemeinen Graphen gilt dies nicht, da dieser unverbundene, also in keinem Element der Kantenmenge auftretende Ecken enthalten kann.

Berücksichtigte mit $\circ := \uparrow$ -Spin und $\bullet := \downarrow$ -Spin und Irrelevante mit $\bullet := \uparrow \vee \downarrow$ gekennzeichnet; der Ausgangsgitterplatz wird mit \odot beziehungsweise \ominus markiert:



Die in diesem Beispiel vorkommenden ($n = 1$) beziehungsweise generell möglichen ($n = 3$) Gitterausschnitte sind in Tabelle 1.2 zusammengestellt [24:A038055].

n	k	Ausgangszustände
1	2	$\odot \vee \ominus$
3	3	$(\odot \wedge \ominus \vee \ominus \wedge \odot) \wedge ((\odot \vee \ominus) \wedge \odot \vee (\ominus \vee \odot) \wedge \odot)$
5	4	$(\odot \wedge \odot \wedge \odot \wedge \odot \vee \odot \wedge \odot \wedge \odot \wedge \odot)$ $\wedge (\odot \wedge (\odot \vee \odot) \wedge \odot \vee \odot \wedge (\odot \vee \odot) \wedge \odot)$ $* \wedge (\odot \wedge \odot \wedge \odot \wedge \odot \wedge \odot \wedge \odot \vee \odot \wedge \odot \wedge \odot \wedge \odot \wedge \odot \wedge \odot)$ $\wedge ((\odot \vee \odot \vee \odot \vee \odot) \wedge (\odot \vee \odot \vee \odot \vee \odot) \wedge \odot \vee \odot)$ $\vee (\odot \vee \odot \vee \odot \vee \odot) \wedge (\odot \vee \odot \vee \odot \vee \odot) \wedge \odot$

Tabelle 1.3. Alle beitragenden Ausgangszustände für Elektronentransferprozesse (Farbkonvention: $\square \mapsto \uparrow$ -Spin, $\blacksquare \mapsto \downarrow$ -Spin) bis einschließlich Störungsordnung $n = 5$. Hinsichtlich der Spinbelegung gleiche sind in runden Klammern zusammengefaßt, bezüglich des Wurzelspins „invers“ belegte Bäume werden (zusätzlich) mit dem Symbol \vee einander gegenübergestellt [19]: [24:A198760]. Man beachte, daß für den Gitterausschnitt in der mit einem Stern gekennzeichneten Zeile ein hier überangenes „Spiegelbild“ existiert, siehe Zeile für $n = 5$ in Tabelle 1.1.

1.5.4. *Ausgangszustände.* Nicht auf allen der möglichen spinbelegten Gitterausschnitte, vergleiche Tabelle 1.2, ist ein erster Hüpfschritt („Sequenzinitialisierungssprung“) durchführbar: Bedingt durch das Pauliverbot, welches generell wie ein Filter die ausführbaren von allen Sprungpermutationen der (i, j) -Summe in Gleichung (1.1) trennt, muß mindestens einer von gegebenenfalls mehreren Nachbarn des Ausgangsgitterplatzes, zu welchen der erste Sprung zu erfolgen hat, ein Elektron der jeweils anderen Spinspezies

tragen. Neben dieser gewaltigen Aufwandsreduktion halbiert zudem die Symmetrie zwischen den beiden Spinsorten, symbolisiert durch \sphericalangle , die Anzahl der zu untersuchenden Ausgangszustände. Sämtliche zur Grundzustandsenergie beitragenden Ausgangszustände bis einschließlich $k = 4$ finden sich in Tabelle 1.3, siehe Referenz [19], aufgenommen in Referenz [24] als Ganzzahlfolge A198760.

1.5.5. Hüpfsequenzen am Beispiel erster Störungsordnung. Aufgrund des in diesem Abschnitt bisher Dargelegten kann Gleichungszeile (1.19-6) unter expliziter Ausführung der Spinsumme mit den beiden (in folgender Gleichung (1.27) eingerahmten) Ausgangszuständen, siehe Tabelle 1.3, durch die zwei eingeklammerten *Elektronenpropagationssequenzen* oder, kürzer, *Hüpfsequenzen* neu geschrieben und in Form eines *Diagramms* in Anlehnung an eine chemische Reaktionsgleichung dargestellt werden.

Dabei ist unter den Hüpfsequenzen die formale Beitragsberechnung angegeben, in welcher $b = 0$ die Anzahl der vertauschten Teilchen innerhalb einer Spinsorte, $d = 1$ die Anzahl an Doppelbesetzungen nach dem ersten Hüpfschritt und $\kappa = 1$ der Exponent des zwischen den Erzeuger-Vernichter-Paaren $\hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow}$ und $\hat{c}_{1,\uparrow}^+ \hat{c}_{2,\uparrow}$ eingeschobenen S -Operators ist. (Tatsächlich treten $b > 0$ und $d > 1$ erst in der Ordnung U^{-5} auf.) Angelehnt an Gleichungszeile (1.19-3) sind in Gleichung (1.27) die am zugrundeliegenden Prozeß beteiligten Operatoren angegeben, vergleiche dazu auch Abschnitt 1.5.1.

Für diese wie alle weiteren Graphendarstellungen und -rechnungen gelte die Farbkonvention $\square \mapsto \emptyset$ (Loch), $\blacksquare \mapsto \uparrow$ -Spin, $\blacksquare \mapsto \downarrow$ -Spin und $\blacksquare \mapsto \uparrow \wedge \downarrow = \uparrow\downarrow$ (Doppelbesetzung), sofern keine andere angegeben wird.

$$\begin{aligned}
 E_0^{(1)} &= \underbrace{\left(\begin{array}{c} \hat{P}_0 \\ \begin{array}{|c|} \hline \uparrow \\ \hline \end{array} \\ \hat{T}_{+1} \\ \begin{array}{|c|} \hline \uparrow \\ \hline \end{array} \\ \hat{T}_{-1} \\ \hat{P}_0 \end{array} \right)}_{= \frac{(-1)^b}{2 \cdot (-d)^\kappa \cdot 2}} + \underbrace{\left(\begin{array}{c} \hat{P}_0 \\ \begin{array}{|c|} \hline \downarrow \\ \hline \end{array} \\ \hat{T}_{+1} \\ \begin{array}{|c|} \hline \downarrow \\ \hline \end{array} \\ \hat{T}_{-1} \\ \hat{P}_0 \end{array} \right)}_{= \frac{(-1)^b}{2 \cdot (-d)^\kappa \cdot 2}} \\
 &= -\frac{1}{2} \quad \text{mit } b = 0, d = 1 \text{ und } \kappa = 1 \quad (1.27)
 \end{aligned}$$

Aus unmittelbar einsichtigem Grunde, nämlich der Symmetriebeziehung zwischen beiden Hüpfsequenzen, kann beispielsweise (und so künftig) der rechte Summand einfach durch den Faktor 2 im Energiebeitrag $E_0^{(1)}$ berücksichtigt werden, vergleiche auch Abschnitt 1.4.2, Gleichungszeile (1.19-7).

Die analog ausgeführte Rechnung für das Falicov-Kimball-Modell liefert den aus der analytischen Lösung bekannten (alleinigen) Beitrag zur Grundzustandsenergie

$$E_0^{(1),\text{FKM}} = -\frac{1}{4}, \quad (1.28)$$

da laut Gleichung (1.2) die Spinsumme entfällt, das heißt anschaulich gemäß unserer Konvention, daß der rechte Hüpfprozeß aufgrund der Immobilität der \downarrow -Elektronen nicht ausgeführt werden kann.

Unentbehrlich für die effiziente Behandlung höherer Ordnungen sind die bereits in Gleichung (1.27) auftretenden Operatoren \hat{T}_{+1} (Erhöhung der Doppelbesetzungszahl um eins) und \hat{T}_{-1} (Verringerung der Doppelbesetzungszahl um eins), welche ab $\mathcal{O}(U^{-3})$ von \hat{T}_0 (Unbeeinflussung der Doppelbesetzungszahl) ergänzt werden. Die allgemeine Formulierung dieser Operatoren lautet [19]

$$\hat{T}_{+1} = - \sum_{(i,j)} \sum_{\sigma} \hat{n}_{i,-\sigma} (1 - \hat{n}_{j,-\sigma}) \hat{c}_{i,\sigma}^+ \hat{c}_{j,\sigma}, \quad (1.29)$$

$$\hat{T}_{-1} = - \sum_{(i,j)} \sum_{\sigma} \hat{n}_{j,-\sigma} (1 - \hat{n}_{i,-\sigma}) \hat{c}_{i,\sigma}^+ \hat{c}_{j,\sigma} \quad \text{und} \quad (1.30)$$

$$\hat{T}_0 = - \sum_{(i,j)} \sum_{\sigma} (1 - \hat{n}_{i,-\sigma} - \hat{n}_{j,-\sigma} + 2\hat{n}_{i,-\sigma}\hat{n}_{j,-\sigma}) \hat{c}_{i,\sigma}^+ \hat{c}_{j,\sigma}; \quad (1.31)$$

im Besonderen gilt:

$$\hat{T} = \hat{T}_{+1} + \hat{T}_{-1} + \hat{T}_0. \quad (1.32)$$

1.6. Zusammenfassung. Die in diesem Kapitel weitgehend¹⁴ analog zu Referenz [19] aus der einfachen Kato-Störungstheorie hergeleitete Rechenvorschrift für die Ordnung n beinhaltet vier wesentliche Schritte, nämlich

- (1) die Entwicklung von \tilde{h}_n zwecks Gewinnung sämtlicher Hüpfprozesse in Umsetzung von Abschnitt 1.5.1,
- (2) die „Extraktion“ sämtlicher Ausgangszustände mit $k = (n + 3)/2$ Gitterplätzen aus dem Bethegitter gemäß den Skizzen in den Abschnitten 1.5.2 bis 1.5.4,
- (3) die Generierung aller Hüpfsequenzen aus den $n + 1 = 2k - 2$ Hüpfritten auf einem jeden der Ausgangszustände, siehe Beispiel in Abschnitt 1.5.5, und
- (4) abschließend die im selben Abschnitt beispielhaft durchgeführte Berechnung der Summanden von $E_0^{(n)}$ durch Gewichtung der Sequenzen anhand der Prozeßkoeffizienten unter Berücksichtigung der Grundzustandwiederherstellungen, der in den Hüpfprozessen enthaltenen \hat{S}^k , der Anzahl an Doppelbesetzungen d nach jedem Hüpfritt und der Gesamtzahl an Teilchenvertauschungen b .

Wie nun diese „Rezeptur“ in ihren Einzelteilen konkret in einen Gesamtalgorithmus zu übertragen ist, wird am Anfang von Kapitel 3, vornehmlich in Abbildung 3.1, skizziert und im weiteren Verlauf des Kapitels in allen Details beschrieben.

Da die Natur des Lösungsweges kombinatorischer Art ist, wächst – wie auch aus den Gleichungen (1.23) bis (1.26) sowie Tabelle 1.3 zu schließen – der Aufwand in höheren

¹⁴Die einzige sich hier zur Kato-Takahashi-Störungstheorie und somit zu EVA KALINOWSKIS Arbeiten ergebende Abweichung besteht in einer „Asymmetrie“ in den Vorfaktoren der Terme des \tilde{h}_n (Prozeßgewichte): Es fallen zwar, wie in Abschnitt 3.3 umfassend gezeigt wird, durch den Verzicht auf TAKAHASHIS unitäre Transformation prinzipiell beitragende Hüpfprozesse durch das Gewicht Null aus der Rechnung heraus, werden aber durch die entsprechend höheren Gewichte anderer Prozesse (beispielsweise durch den adjungierten Prozeß) „mitberücksichtigt“. Erwartungsgemäß sind folglich die Reihenoeffizienten aus beiden Verfahren Ordnung für Ordnung identisch.

Ordnungen der $1/U$ -Reihe schlicht gesagt „explosionsartig“ an. Findet sich in erster Ordnung effektiv für jede der beiden im Hubbardmodell möglichen Spinbelegungen der Wurzel nur eine Hüpfsequenz, so ergeben sich in dritter symmetriebereinigt 10, in fünfter 324, in siebenter bereits 22736 Sequenzen, siehe Referenz [19], aufgenommen in Referenz [24] als Ganzzahlfolge A198761, und Tabelle 3.8 in Abschnitt 3.4.6. Eine weitergehende Betrachtung der Problemkomplexität wird in Abschnitt 5.1.1 und folgendem vorgenommen.

2. Mathematische Grundlagen

2.1. Begriffe und Notationen. Zur Formalisierung der am Verfahren beteiligten Graphenstrukturen und der diese verknüpfenden Operationen auf einheitlicher Grundlage werden im Folgenden alle benötigten Begriffe aus der Graphentheorie eingeführt. Die Darstellung orientiert sich teilweise am Lehrbuch von REINHARD DIESTEL zu diesem Gebiet [37] sowie dem in die Diskrete Mathematik einführenden Werk von JIŘÍ MATOUŠEK und JAROSLAV NEŠETŘIL [38], weicht aber in der Bezeichnungsweise zum Teil ab, vergleiche Referenz [39] oder [40], und geht inhaltlich sachgemäß darüber hinaus. Dieser Abschnitt ist Voraussetzung für Abschnitt 2.2, *Problemspezifische Objekte und Operationen*, welcher seinerseits für Kapitel 3, die detaillierte Beschreibung der neuen „rechenmaschinellen“ Realisierung der in Kapitel 1 ausgeführten algebraisch-graphentheoretischen Störungstheorie, tragende Bedeutung hat.

Hauptsächlicher Gegenstand der Betrachtung sind einerseits „klassische“ *Mengen*, also Ansammlungen unterscheidbarer Objekte, welche in gekennzeichneten Sonderfällen *geordnet* sein müssen¹⁵, andererseits *Tupel*, in welchen die Reihenfolge der gegebenenfalls mehrfach auftretenden Elemente bedeutsam ist. Erstere werden, wie üblich, von geschweiften Klammern eingeschlossen, zum Beispiel $\{c, a, b\}$; letztere in runden Klammern notiert oder zwecks Vermeidung von Verwechslungen mit langen Tildezeichen überbrückt¹⁶, $(a, b, c, a) \equiv \overline{a, b, c, a}$.

Zusätzlich erfordert die formale Beschreibung des Rechenverfahrens die Verwendung von *Multimengen*, welche sich – ohne einen neuen mathematischen Grundbegriff einführen zu müssen – als Mengen gemäß obiger Definition von Paaren (n_i, o_i) schreiben lassen: Das zweite Element o_i eines solchen Multimengenelements repräsentiert das Objekt, nach welchem die Multimenge optional geordnet wird, das erste, $n_i \in \mathbb{N} \setminus \{0\}$, die Anzahl seines Vorkommens¹⁷, etwa $\{(1, b), (2, a), (1, c)\}$.

¹⁵Naturngemäß *sind* die Elemente aller durch ein Entwicklungsverfahren für eine vollständige Reihe erzeugten Mengen geordnet, und wir notieren die Mengen auch in dieser Weise, selbst wenn in einem betrachteten Verfahrensschritt die *Anfangsordnung* der Elemente irrelevant ist, weil beispielsweise *alle möglichen Anordnungen* berücksichtigt werden müssen.

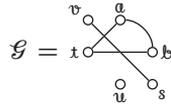
¹⁶Damit besteht unter anderem eine leichter ersichtliche Abgrenzung von der Regelung von Rechenoperationspräcedenzen durch runde Klammern.

¹⁷Wir erweitern diese Multiplizität eines Objektes später in Abschnitt 3.3.1 zu einem Vorfaktor $z_i \in \mathbb{Z} \setminus \{0\}$.

Die Vereinigung zweier Mengen wird, wie üblich, zum Beispiel mit $\{c, a\} \cup \{b, a\} = \{c, a, b\}$ notiert; im Falle von Tupeln betrachten wir ausschließlich deren (nicht kommutatives) Aneinanderhängen, die *Konkatenation*, $(a, b) \circ (c, a) = (a, b, c, a)$. Werden zwei Multimengen zusammengefaßt, so sind die „Anzahlen“ zu berücksichtigen, $\{(1, b), (1, a)\} \sqcup \{(1, a), (1, c)\} = \{(1, b), (2, a), (1, c)\}$. Betragsstriche kennzeichnen in jedem der Fälle die Berechnung der Kardinalzahl, wobei die Vielfachheiten der Multimengenelemente unberücksichtigt bleiben, $|\{(1, b), (2, a), (1, c)\}| = |\{b, a, c\}| = 3$.

Die bei endlicher Störungsrechnungsordnung stets endlichen (Multi)mengen und Tupel behandeln wir in der späteren datentechnischen Umsetzung als *Listen*: Sei beispielsweise $M = \{c, a, b\}$ eine Menge, dann stellt $\llbracket M \rrbracket_i$ den *Zugriff* auf deren *i*tes Element dar, $\llbracket M \rrbracket_3 = b$; praktisch ist $i \in \mathbb{N} \setminus \{0\}$, genauer $i = 1, 2, \dots, |M|$, als *Adresse* des *i*ten Listenelements aufzufassen. Entsprechendes gilt für Tupel¹⁸ und Multimengen, bei letzteren, beispielsweise \check{M} , ist $\llbracket \check{M} \rrbracket_i = (n_i, o_i)$.

2.1.1. *Graph, Teilgraph.* Man nennt $\mathcal{G} = (V, E)$ einen *Graphen*, $V = V(\mathcal{G})$ dessen *Ecken-* und $E = E(\mathcal{G}) \subseteq \{\{i, j\} \mid i, j \in V(\mathcal{G})\}$ dessen *Kantenmenge*¹⁹. Die *Ordnung* von \mathcal{G} wird als die Anzahl der Ecken $n = |V|$ definiert. Beispiel für einen Graphen \mathcal{G} in (wie anhand von Skizze 2.2 in Abschnitt 2.1.12 gezeigt wird) beliebiger Darstellung, $V(\mathcal{G}) = \{a, b, s, t, u, v\}$ ²⁰, $E(\mathcal{G}) = \{\{a, b\}, \{a, t\}, \{b, t\}, \{s, v\}\}$:



(Skizze 2.1)

Im Kontext des zugrundeliegenden festkörperphysikalischen Problem es verstehen wir unter einer Ecke einen *Gitterplatz*, in Zeichnungen durch \circ symbolisiert; zwei Gitterplätze k und l sind *nächste Nachbarn*, wenn $\{k, l\} \in E$ ist. Ecken und im Allgemeinen auch Kanten können, wie in Abschnitt 2.1.15 erläutert, eine oder mehrere *Eigenschaften* (Attribute) zugeordnet sein. Wir beschränken uns in dieser Abhandlung problemgemäß auf die *Besetzungszahlen* der Gitterplätze, welche wir entsprechend unterscheidbar *färben*.

Wir bezeichnen $g = (V', E')$ als einen *Teilgraphen* von $\mathcal{G} = (V, E)$, in Zeichen $g \subseteq \mathcal{G}$, wenn für dessen Eckenmenge $V' \subseteq V$ und für dessen Kantenmenge $E' \subseteq E$ gilt. Beispiel (siehe Skizze 2.1): $(\{b, s, t, v\}, \{\{s, v\}\}) \subset \mathcal{G}$.

¹⁸Aus Effizienzgründen werden Tupel im Computerprogramm zum Teil auch als Zeichenketten behandelt, geeignete Elemente vorausgesetzt, vergleiche Abschnitt 2.2.3 und insbesondere Fußnote 30.

¹⁹Wir betrachten zunächst nur *ungerichtete* Kanten, und definieren solche als zweielementige Teilmenge von V und nicht als Tupel $(i, j) \in V \times V = V^2$ mit $i \neq j$, welches aufgrund der Elementanordnung eine *Orientierung* der Kante impliziert. Nebenbemerkung: Da problembedingt zwei Ecken niemals mehrfach miteinander verbunden sind, können wir auf die Definition von E als Multimenge und entsprechend \mathcal{G} als *Multigraphen* verzichten. Diese Aufwandverringerung wird in verallgemeinerten Verfahren möglicherweise entfallen, siehe Kapitel 6.

²⁰Nach dem zuvor über den Mengenbegriff Gesagtem ist zu $V(\mathcal{G})$ beispielsweise $\{t, b, u, v, a, s\}$ äquivalent; wir haben uns hier willkürlich lexikographischer Ordnung bedient, $a < b < \dots < v$.

2.1.2. *Grad (Koordinationszahl)*. Sei $K(i) \subset V(\mathcal{G})$ die Menge aller nächsten Nachbarn der Ecke i , für welche $K(i) = \{j \in V(\mathcal{G}) \mid \{i, j\} \in E(\mathcal{G})\}$ gilt, dann nennen wir $Z(i) = |K(i)|$ den *Grad* beziehungsweise im physikalischen Kontext die *Koordinationszahl* von i . Beispiele (siehe Skizze 2.1): $K(\mathfrak{b}) = \{\mathfrak{a}, \mathfrak{t}\}$, $Z(\mathfrak{b}) = 2$; $Z(\mathfrak{u}) = |K(\mathfrak{u})| = |\emptyset| = 0$.

2.1.3. *Weg, Schleife*. Einen Kantenabfolge $E' = \{\{j, k_1\}, \{k_1, k_2\}, \dots, \{k_{n'-3}, k_{n'-2}\}, \{k_{n'-2}, l\}\} \subseteq E$ nennen wir einen die Ecken $j, l \in V$ verbindenden *Weg* auf dem Graphen $\mathcal{G} = (V, E)$. Mit der zugehörigen Eckenmenge $V' = \{j, k_1, k_2, \dots, k_{n'-3}, k_{n'-2}, l\} \subseteq V$ stellt der Weg einen speziellen Teilgraphen $\mathcal{g} = (V', E')$ von \mathcal{G} dar. $|E'| = n' - 1$ mit $n' = |V'|$ bezeichnen wir dabei als die Länge des Weges. Beispiel (siehe Skizze 2.1): Die Ecken \mathfrak{a} und \mathfrak{t} sind über den Weg $E' = \{\{\mathfrak{a}, \mathfrak{b}\}, \{\mathfrak{b}, \mathfrak{t}\}\} \subset E(\mathcal{G})$ der Länge $|E'| = |\{\mathfrak{a}, \mathfrak{b}, \mathfrak{t}\}| - 1 = 2$ miteinander verbunden.

Verbinden zwei oder mehr verschiedene Wege die Ecken j und l , so sprechen wir von einer oder mehreren *Schleifen* auf \mathcal{G} . Beispiel (siehe Skizze 2.1): Die Ecken \mathfrak{a} und \mathfrak{t} sind sowohl über den Weg $E' = \{\{\mathfrak{a}, \mathfrak{b}\}, \{\mathfrak{b}, \mathfrak{t}\}\}$, als auch über den (direkten) Weg $E'' = \{\{\mathfrak{a}, \mathfrak{t}\}\}$ miteinander verbunden.

2.1.4. *Zusammenhang*. Existieren von jeder Ecke eines Graphen \mathcal{G} ein oder mehrere Wege zu allen anderen Ecken, so nennt man \mathcal{G} *zusammenhängend*; es muß dann $|E(\mathcal{G})| \geq n - 1$ gelten. Notwendig, aber nicht hinreichend ist die Forderung $Z(i) > 0 \forall i \in V(\mathcal{G})$; eine Ecke i mit $Z(i) = 0$ heißt *isoliert*. Beispiel (siehe Skizze 2.1): \mathcal{G} ist (zweifach) unzusammenhängend; \mathfrak{u} ist isoliert.

2.1.5. *Baum; Wald*. Einen zusammenhängenden Graphen \mathcal{T} , für welchen einschränkend $|E(\mathcal{T})| = n - 1$ gilt, nennt man *Baum*; Bäume sind folglich *schleifenfrei* – ein für die Lösung des physikalischen Problems entscheidender Sachverhalt. Ein unzusammenhängender Graph \mathcal{W} ohne Schleifen ist ein *Wald*; jeder zusammenhängende Teilgraph t_i , also jede Komponente eines Waldes, ist entsprechend ein Baum, $\mathcal{W} = \bigcup_i t_i$.

2.1.6. *Wurzel, Blatt*. Falls eine Ecke $r \in V(\mathcal{T})$ in besonderer Weise ausgezeichnet ist, beispielsweise durch den baumerzeugenden Algorithmus als erste Ecke, so bezeichnet man diese als dessen *Wurzel*; wir verwenden dafür das Symbol \odot . Das formal entstehende Tripel $(V, E, r) = \tilde{\mathcal{T}}$ können wir wieder auf die Schreibweise $(V, E) = \mathcal{T}$ verkürzen, indem wir die Wurzelecke grundsätzlich an den Anfang der (dann geordneten) Menge der Ecken stellen. (Die Kantenmenge ist dann gegebenenfalls entsprechend neu zu ordnen.) Da wir jedoch künftig die „Benennung der Ecken“ mit $V = \{i \mid i = 1, 2, \dots, n, n \in \mathbb{N}\}$ festlegen wollen und beim Aufbau des Baumes an dessen Wurzel beginnen, ist ohnehin $r = 1$. Nebenbei gelte auch $k < l \forall \{k, l\} \in E(\mathcal{T})$. Eine Ecke $j \in V \setminus \{r\}$ mit dem Grad $Z(j) = 1$ wird *Blatt* genannt, in Zeichnungen: \circ . Veranschaulichungen finden sich nächstliegend in den Abschnitten 2.1.7, 2.1.10 und 2.1.11.

Eine im Weiteren wichtige Art von Weg $\mathcal{J} = (V', E')$ auf einem „verwurzelten“ Baum $\mathcal{T} = (V, E)$ gründet sich auf die Kantenmenge $E'_i = \{\{r, \dots\}, \dots, \{\dots, l_i\}\}$, welche die Wurzel r mit dem i ten der $I = |\{j \mid Z(j) = 1, j \in V \setminus \{r\}\}|$ Blätter l verbindet. Die Menge aller dieser Wege schreiben wir als $\mathbb{I} = \{(V'_i, E'_i) \mid i = 1, 2, \dots, I\}$.

2.1.7. *Eckenklasse, Baumrotation.* Zwei Ecken $i, j \in V$ eines Baumes $\mathcal{R} = (V, E)$ ohne Wurzel heißen zur selben *Eckenklasse* gehörig, wenn die beiden aus \mathcal{R} hervorgehenden Bäume mit Wurzel, $\tilde{\mathcal{R}} = (V, E, i)$ und $\tilde{\mathcal{R}}' = (V, E, j)$ gemäß obiger ausführlicher Notation, abgesehen von Neunumerierung der Ecken identisch sind. Ebensoles gilt auch für die Wurzel $r \in V$ und eine beliebige weitere Ecke $k \in V$ eines von $\tilde{\mathcal{S}} = (V, E, r)$ in $\tilde{\mathcal{S}}' = (V, E, k)$ umgewandelten „Wurzelbaumes“. Beide Transformationen bezeichnen wir mit dem Begriff *Baumrotation*, welcher bereits eine Neuausrichtung der Bäume (siehe Abschnitte 2.1.9) impliziert, in Zeichen $\mathcal{R} \rightsquigarrow \tilde{\mathcal{R}}^{21}$ und $\tilde{\mathcal{S}} \rightsquigarrow \tilde{\mathcal{S}}'$. Beispiel mit zwei Eckenklassen (Ausgangsbaum links ohne und rechts mit Wurzel, Markierung \circledast auf der jeweiligen zur Wurzel „rotierten“ Ecke):



2.1.8. *Verzweigungsgrad, Ast.* Unter dem *Verzweigungsgrad* innerhalb eines „Wurzelbaumes“ \mathcal{T} verstehen wir die Anzahl $|J(i)| = |K(i)| - 1 = m$ der aus Sicht der Wurzel r von einer Ecke $i \in V \setminus \{r\}$ abzweigenden *Äste* genannten Teilgraphen t (Menge $J(i) = \{t_1, \dots, t_m\}$ mit $t_j \subset \mathcal{T}$) beziehungsweise Blätter ($J = \emptyset$). Für den Verzweigungsgrad der Wurzel gilt $|J(r)| = |K(r)|$. Gleicher Verzweigungsgrad zweier Ecken ist für die Zugehörigkeit zur gleichen Eckenklasse zwar notwendige, nicht aber hinreichende Voraussetzung.

2.1.9. *Ausgerichtete und nicht ausgerichtete Bäume.* Wir unterscheiden mit einer gewissen Willkür anhand der von uns zweckbedingt bevorzugten graphischen Darstellung *ausgerichtete* und *nicht ausgerichtete Bäume*, unabhängig von möglicherweise gerichteten Kanten. Kriterium für die nicht ausgerichtete Zeichenweise ist eine im Aufbauprinzip der gesamten Baumart enthaltene Zentrosymmetrie, also die Anwendbarkeit der zyklischen Gruppe C_n , wobei $n \in \mathbb{N}$ der Koordinationszahl der Zentralecke entspricht (formaler Sonderfall: alle Bäume mit $|V| = 1 \Rightarrow n = 0 \Rightarrow C_\infty$). Überdies sind Bäume ohne eine besondere Ecke, „(wurzel)freie Bäume“, stets nicht ausgerichtet, während „Wurzelbäume“ – auch solche mit Symmetrie – bevorzugt als ausgerichtet angesehen werden. Anschaulich werden nicht ausgerichtete Bäume in Aufsicht, ausgerichtete Bäume hingegen in Seitenansicht dargestellt; in letzterem Fall ist die Wurzel immer *unten*.

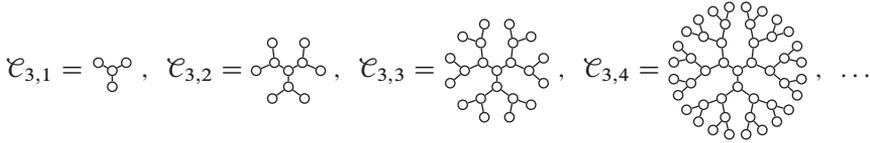
2.1.10. *Nicht ausgerichtete Bäume: Cayleybäume, Permutationsbäume.* Wesentlich für die weitere Untersuchung sind unter anderem die (vollständigen) *Cayleybäume* $\mathcal{C}_{z,g}$. Diese bestehen aus einer zentralen *Mutterecke*²², welche in erster von g *Generationen*²³ mit z *Tochterecken* durch z Kanten verbunden ist (Koordinationszahl z). Die Töchter

²¹Genaugenommen ist die Baumrotation $\mathcal{R} \rightsquigarrow \tilde{\mathcal{R}}$ lediglich die Auszeichnung einer Wurzel in \mathcal{R} , also dessen Überführung in einen Wurzelbaum und zeichnerische Ausrichtung.

²²Die „Mutterecke“ könnte mit obiger Begründung auch als Wurzel angesehen werden; im Falle unendlich ausgedehnter Cayleybäume, $\mathcal{C}_{z,g \rightarrow \infty}$, dem Bethegitter, kommt allerdings, wie noch gezeigt werden wird, keinem Gitterplatz eine ausgezeichnete Stellung zu (*Translationsinvarianz*, siehe Abschnitt 2.2.1).

²³Man beachte, daß – rückbezüglich auf Abschnitt 2.1.6 – der Zusammenhang $|E'(\mathcal{J})| = g \ \forall \ \mathcal{J} \in \mathbb{I}$ gilt, also die Länge des Weges von der zentralen Mutterecke zu einem jeden der Blätter gleich der Generationenanzahl ist.

ihrerseits haben, sofern $g > 1$, in nächster Generation $z - 1$ Tochterecken (Verzweigungsgrad $z - 1$). In g ter Generation sind alle Ecken Blätter, in allen vorausgehenden Generationen weisen die Ecken die Koordinationszahl z auf ($z - 1$ Kanten zu den Töchtern, eine Kante zur jeweiligen Mutterecke):

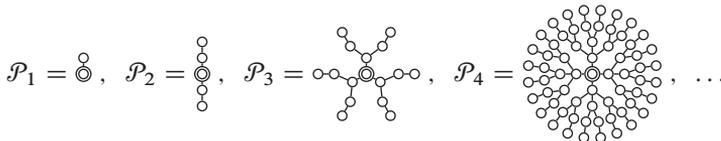


Sonderfälle sind $\mathcal{C}_{z,0} = \circ$ für $z \geq 0$ sowie $\mathcal{C}_{0,g} = \circ$ für $g \geq 0$. Die Eckengesamtzahl im Allgemeinen ergibt sich auf Grundlage einer geometrischen Reihe aus der (endlichen) Summe

$$|V(\mathcal{C}_{z,g})| = 1 + z \sum_{i=1}^g (z - 1)^{i-1} = \frac{z(z - 1)^g - 2}{z - 2}, \tag{2.1}$$

welche sich beispielsweise für obige $\mathcal{C}_{3,g}$ zu $|V| = 3 \cdot 2^g - 2$ vereinfacht [24:A033484]; die Kantenanzahl berechnet sich, wie für alle Bäume, mit $|E(\mathcal{C}_{z,g})| = |V(\mathcal{C}_{z,g})| - 1$.

Ebensfalls bedeutsam sind im Folgenden die (vollständigen) *Permutationsbäume* \mathcal{P}_n , wobei n die Anzahl der zu permutierenden Tupелеlemente bezeichnet. Ein Permutationsbaum besteht, gleich einem Cayleybaum, aus einer zentralen Mutter- oder Wurzecke, welche in erster von n Generationen mit n Töchtern umgeben ist. Sei i die betrachtete Generation ($1 \leq i \leq n$), so hat darin jede Tochterecke ihrerseits $n - i$ Töchter; demzufolge sind in n ter Ordnung alle Ecken Blätter, $|J| = 0$:



(Abbildung 3.7 basiert auf \mathcal{P}_5 .) Für $n = |\emptyset|$ ist formal $\mathcal{P}_{n=0} = \circ$. Die Anzahl der Blätter beträgt $n!$; für die Gesamtzahl der Ecken von \mathcal{P}_n , welche dementsprechend mit $n!$ anwächst, gilt

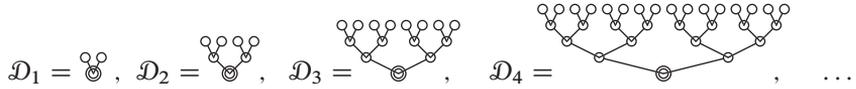
$$|V(\mathcal{P}_n)| = 1 + \sum_{i=1}^n \prod_{j=1}^i (n - j + 1) = e \Gamma(n + 1, 1), \tag{2.2}$$

worin $\Gamma(z, a)$ für die bezüglich der unteren Integrationsgrenze a parametrisierte („unvollständige“) Gammafunktion steht [24:A000522].

2.1.11. *Ausgerichtete Bäume: Binärbäume.* Die für die weitere Betrachtung wichtigen (vollständigen) *Binärbäume* \mathcal{D}_g (g : Generationsanzahl²⁴; im Falle einer ausgerichteten

²⁴Vergleiche Fußnote 23.

zeichnerischen Darstellung: Ebenenanzahl oder Tiefe) besitzen an jeder Ecke i ter Generation für $i < g$ den Verzweigungsgrad 2; in g ter Generation sind alle Ecken Blätter. Die Ecke nullter Generation wird als Wurzel bezeichnet.



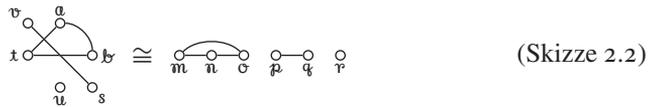
Formal ist $\mathcal{D}_0 = \circ$. Die Gesamtzahl der Ecken $|V|$ läßt sich, wiederum von einer geometrischen Reihe ausgehend, wie folgt berechnen [24:A000225]:

$$|V(\mathcal{D}_g)| = 1 + \sum_{i=1}^g 2^i = \sum_{i=0}^g 2^i = 2^{g+1} - 1. \tag{2.3}$$

2.1.12. *Isomorphismus.* Wir bezeichnen eine bijektive Abbildung

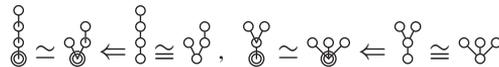
$$\varphi: V \leftrightarrow V' \quad \text{mit} \quad \{i, j\} \in E \Leftrightarrow \{\varphi(i), \varphi(j)\} \in E' \quad \forall i, j \in V, \tag{2.4}$$

welche den Graphen \mathcal{G} in den Graphen $\mathcal{G}' = \{V', E'\}$ überführt, als *Isomorphismus* und nennen \mathcal{G}' zu \mathcal{G} *isomorph*, $\mathcal{G}' \cong \mathcal{G}$. Isomorphe Graphen und Bäume werden im folgenden nicht unterschieden. Beispiel (vergleiche Skizze 2.1; für Bäume siehe Tabelle 1.1 und insbesondere Abschnitt 2.2.2):



Im Kontext dieser Arbeit ist die Einführung des Isomorphismusbegriffes insbesondere der Tatsache geschuldet, daß die systematische, also auf Algorithmen basierende und zudem rechenmaschinelle Erzeugung und Verarbeitung von Graphen eine Ordnung der Strukturen (angedeutet in Skizze 2.2, vertieft in Abschnitt 2.2.2 und folgenden) in Form von Listen bewirkt und verlangt.

2.1.13. *Topologische Äquivalenz, Topologieklassse.* Wir nennen in Abschwächung des Isomorphiekriteriums zwei „Wurzelbäume“ \mathcal{A} und \mathcal{A}' *topologisch äquivalent*, $\mathcal{A}' \simeq \mathcal{A}$, beziehungsweise der selben *Topologieklassse* angehörend, wenn die entsprechenden „unverwurzelten“ Bäume \mathcal{T} und \mathcal{T}' zueinander isomorph sind, siehe Tabelle 2.1. Beispiele:



2.1.14. *Astordnung.* Zwecks Eliminierung isomorpher Strukturen eines ausgerichteten, „verwurzelten“ Baumes \mathcal{A} ist es vorteilhaft, auf dessen vom Wurzelknoten abzweigende Äste eine *Ordnungsrelation* anzuwenden. (Bei nicht ausgerichteten Bäumen wird aufgrund deren symmetrischen Aufbaues – alle Hauptäste sind identisch – dieser Begriff nicht benötigt.) Formal kann eine solche *Astordnung* durch Umwandlung von \mathcal{A} unter Entfernung von dessen Wurzel und aller mit dieser inzidenten Kanten in einen Wald \mathcal{Z} erfolgen, beispielsweise mit

$$\mathcal{Z} = (V(\mathcal{A}) \setminus \{1\}, E(\mathcal{A}) \setminus \{\{1, i\} \mid i \in K(\mathcal{A})(1)\}) \tag{2.5}$$

(sofern $V(\mathcal{A}) = \{j \mid j = 1, 2, \dots, |V(\mathcal{A})|\}$), welcher sich seinerseits in $p = |K(\mathcal{A})|$ zusammenhängende Komponenten a , die Hauptäste von \mathcal{A} , zerlegen läßt ($\bigcup_p a_p = \mathcal{Z}$). Für die Hauptäste gilt $\bigcap_p V(a_p) = \emptyset$ und entsprechend $\bigcap_p E(a_p) = \emptyset$. Ist nun, gleichbedeutend mit $a_1 \preceq a_2 \preceq \dots \preceq a_p$, mit einer noch zu definierenden Gewichtungsfunktion g die Ordnungsrelation

$$g(a_1) \leq g(a_2) \leq \dots \leq g(a_p) \tag{2.6}$$

erfüllt, so nennen wir den Baum \mathcal{A} bezüglich seiner Hauptäste *geordnet*; jeder andere Baum \mathcal{A}' mit einer abweichenden Reihenfolge der selben Hauptäste ist zu \mathcal{A} folglich isomorph.

Wie später in den Abschnitten 3.1 und 3.2 ausführlicher dargelegt wird, ist als Gewichtungsfunktion $g(a_p) = |V(a_p)|$, also die Anzahl der Ecken von a_p , im Allgemeinen noch nicht ausreichend: Eine vollständige Ordnung von \mathcal{A} kann auf diesem Wege nur durch *rekursive* Anwendung der Ordnungsrelation (2.6) auf dessen Teilgraphen a – jedes Element des Waldes \mathcal{Z} ist ein wiederum in einen Wald zerlegbarer Baum – erreicht werden. Beispiel:



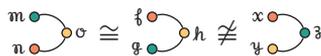
2.1.15. *Farbe, Färbung.* Sei X (geordnete) Menge der Attribute χ_q mit $q = 1, 2, \dots, \dots, |X|$, *Farben* genannt, beispielsweise $\{\blacksquare, \square, \blacksquare\}$. Wird jeder Ecke $i \in V(\mathcal{H})$ eines Graphen \mathcal{H} eine Farbe aus X als Attribut zugeordnet, so nennen wir \mathcal{H} bezüglich seiner Ecken *gefärbt*.

Das so entstehende Tupel der Farbattribute F , beispielsweise $(\blacksquare, \square, \blacksquare, \blacksquare, \square)$ bei einem Graphen mit fünf Ecken, ist mit $V = V(\mathcal{H})$ folglich durch eine bijektive Funktion f verknüpft, $f : F \leftrightarrow V, k \mapsto f(k) := k$; dem k ten Element in V wird also das k te Element in F zugeordnet, $|F| = |V|$. Einerseits wird $F(\mathcal{H}) := F$ auch die *Färbung* von \mathcal{H} genannt, andererseits und im Weiteren wird \mathcal{H} als *eine* Färbung des ungefärbten Graphen $\mathcal{G} = (V, E)$ unter der Voraussetzung $|V(\mathcal{G})| = |F|$ (*Färbbarkeit* von \mathcal{G} mit F) bezeichnet:

$$\mathcal{G} \circ (F) = (V, E) \circ (F) = (V, E, F) = \mathcal{H} . \tag{2.7}$$

Man beachte, daß entsprechend auch jeder Weg h auf \mathcal{H} als Teilgraph desselben das zur jeweiligen Eckenteilmenge V' gehörige Teiltupel F' der Farbattribute $F(\mathcal{H})$ enthält: $h = (V' \subseteq V, E' \subseteq E, F')$.

Zwei Färbungen $\mathcal{H} = (V, E, F)$ und $\mathcal{H}' = (V, E, F')$ werden im Sinne eines Isomorphismus *isochromatisch* zueinander genannt, $\mathcal{H}' \cong \mathcal{H}$, wenn eine Abbildungsvorschrift $\vartheta : F \leftrightarrow F'$ existiert, welche auf die Farbattribute topologisch äquivalenter Ecken wirkt – genau dann also, wenn $\vartheta : V \leftrightarrow V'$ auch den zugrundeliegenden ungefärbten Graphen \mathcal{G} in den isomorphen Graphen \mathcal{G}' überführt. Beispiel:

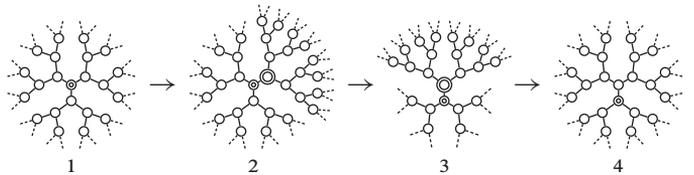


Stehe G für die Menge aller möglichen Färbungen von \mathcal{G} , dann ist unter Vernachlässigung isochromer Färbungen $|G| \leq c^{|\mathcal{V}(\mathcal{G})|}$ mit $c = |X|$.

2.2. Objekte und Operationen problemspezifischer Natur. Abzielend auf eine einerseits möglichst kompakte und andererseits möglichst einfach in Algorithmen übertragbare Schreibweise des graphentheoretischen Rechenverfahrens werden im Weiteren, neben für das zugrundeliegende physikalische Problem besonders wichtigen, aber universell bedeutsamen Objekten (Bethegitter, Butcherbaum), spezifische Operationen (Baumprodukt, Baumsumme) definiert, die von speziellen Annahmen in der Graphendefinition ausgehen.

2.2.1. Bethegitter; Translationsinvarianz bzw. -periodizität. Unter einem Bethegitter \mathcal{E}_z versteht man formal einen Cayleybaum $\mathcal{C}_{z,g}$ mit $z \geq 2$ im Grenzfall Generationsanzahl $g \rightarrow \infty$, also unendlicher Ausdehnung. Damit „erbt“ das Bethegitter – ausgenommen die Existenz von Blättern – alle Eigenschaften der (unverwurzelten) Bäume, von welchen die Tatsache, daß zwischen zwei beliebigen Ecken nur genau ein Weg, also genau eine verbindende Kantenabfolge besteht, für das Weitere besonders bemerkenswert ist. Referenz [36] behandelt den Zusammenhang zwischen Cayleygraph und Bethegitter sowie Eigenschaften desselben im Rahmen der Lösung des Isingmodells eingehender.

Die Struktur des Bethegitters verhält sich *unter Translation* oder, im Terminus der Bäume, unter Rotation durch die Zugehörigkeit aller Ecken zu nur einer Eckenklasse *invariant*; dies ist im Falle $z = 2$, einer endlosen, „periodischen“ Kette, intuitiv einsichtig und kann beispielsweise bei $z = 3$ graphisch leicht gezeigt werden:



Da es auf dem Bethegitter keine Wurzel- oder Mutterecke gibt, ist orientierungshalber in obiger vierstufiger Abbildung in jedem Gitterausschnitt der Bezugsgitterplatz mit einer Markierung (⊙) versehen, zentriert in den Stufen 1 und 2, nach unten verschoben in den Stufen 3 und 4; der in der dritten und vierten Stufe zentrale Gitterplatz, nächster Nachbar des Bezugsknotens, wird bereits in der zweiten Stufe eingekreist (⊙) dargestellt. Der Übergang zu Stufe 2 erfolgt durch teilweises „einblenden“ einer weiteren Generation, jener zu Stufe 3 durch Drehung und Verschiebung des gesamten Ausschnittes sowie „ausblenden“ eines Gitterteiles. Ein Aufweiten der nun nach oben links und oben rechts weisenden Äste und ein Zusammenstauchen des nach unten weisenden Zweiges überführt Stufe 3 in Stufe 4, welcher strukturell mit dem ersten identisch ist.

2.2.2. *Ausgerichtete Bäume: Butcherbäume.* Die bereits im Abschnitt 2.1.14 untersuchten geordneten Bäume werden nach deren „Erstanwender“ in der Mathematik [41] *Butcherbäume* genannt.²⁵ Diese eingrenzende und in der Literatur übliche Bezeichnung soll ausdrücklich auf die Wahl eines repräsentativen Vertreters aus isomorphen Bäumen hinweisen, welche für unsere weiteren Betrachtungen entscheidend ist. Aufgrund deren zentraler Rolle führen wir für die Butcherbäume auch eine unabhängige und konsistent verwendete Notation ein:

$k, l, m \in \mathbb{N} \setminus \{0\}$: Ordnungsnummer (identisch mit der Eckenanzahl), Strukturnummer, Färbungsnummer. Struktur und Färbung werden zwecks „Adressierung“ eines Butcherbaumes fortlaufend gezählt.

λ_k : Anzahl der strukturverschiedenen Butcherbäume k ter Ordnung, unabhängig von etwaiger Färbung. Man beachte, daß $\lambda_1 = 1$ a priori bekannt ist: Weil ein Baum erster Ordnung nur aus der Wurzel besteht, ist nur eine Struktur möglich. Für λ_k läßt sich für $k = 2, 3, \dots$ die Rekursionsformel (2.8)²⁶ mit $\lambda_1 = 1$ und $\llbracket D(i) \rrbracket_j$ als j ten Teiler von i und $\lambda_0 := 1$ angeben, siehe auch Tabelle 2.1 [24:A000081]. Im übrigen entspricht die Anzahl der im folgenden nicht weiter beachteten strukturverschiedenen Bäume k ter Ordnung einschließlich isomorpher der k ten *Catalanzahl* [24:A000108].

$$\lambda_k := \frac{1}{k-1} \sum_{i=1}^{k-1} \lambda_{k-i} \sum_{j=1}^{|D(i)|} \llbracket D(i) \rrbracket_j \lambda_{\llbracket D(i) \rrbracket_j} \quad (2.8)$$

γ_k : Anzahl der Butcherbäume unter Ausschluß der topologisch äquivalenten, also die Anzahl der Topologieklassen in k ter Ordnung, ohne Berücksichtigung etwaiger Färbung, siehe Tabelle 2.1 [24:A000055]. Diese Folge ist identisch mit der Anzahl „unverwurzelter“ Bäume aus k Ecken, vergleiche Abschnitt 2.1.13. Der zwischen γ_k und λ_k bestehende Zusammenhang läßt sich wie folgt formulieren²⁷ [33]:

$$\gamma_k := \lambda_k + \frac{(-1)^k + 1}{2} \binom{\lambda_{\lfloor k/2 \rfloor} + 1}{2} - \sum_{\substack{i+j=k \\ 0 < i \leq j}} \lambda_i \lambda_j \quad (2.9)$$

$\mathcal{A}_{k,l}$: Ungefärbter Butcherbaum mit k Ecken, $V = \{i \mid i = 1, 2, \dots, k\}$, und der Strukturnummer l . Beispiel: Sämtliche Butcherbäume bis einschließlich dritter Ordnung:

$$\mathcal{A}_{1,1} = \circ, \quad \mathcal{A}_{2,1} = \circ \circ, \quad \mathcal{A}_{3,1} = \circ \circ \circ, \quad \mathcal{A}_{3,2} = \circ \circ \circ$$

²⁵Die „Entdeckung“ dieser Bäume geht ebenfalls auf ARTHUR CAYLEY zurück und findet sich in einem bemerkenswerten Artikel im *Philosophical Magazine*, Jahrgang 1857, umfassend beschrieben (vergleiche insbesondere die dortige Tabelle auf Seite 175 in Referenz [42] mit hiesiger Tabelle 2.1).

²⁶Man beachte die „Verwandtschaft“ dieser Gleichung mit Rekursionsformel (2.11) zur Berechnung der Anzahl der Partitionen von $p = k - 1$.

²⁷Die darin auftretende *Gaußklammer* steht mit $\lfloor x \rfloor \in \mathbb{N}$ im Allgemeinen für die größte ganze Zahl, welche kleiner oder gleich $x \in \mathbb{R}$ ist.

k	1	2	3	4	5	6	7	8	9	10	11	12	13	...
λ_k	1	1	2	4	9	20	48	115	286	719	1842	4766	12486	...
γ_k	1	1	1	2	3	6	11	23	47	106	235	551	1301	...

Tabelle 2.1. Anzahl λ_k der Butcherbäume k ter Ordnung [24:A000081] und Anzahl γ_k der Butcherbäume k ter Ordnung ohne topologisch äquivalente [24:A000055].

\mathcal{A}_k : Geordnete Menge aller ungefärbten Butcherbäume k ter Ordnung, $\mathcal{A}_k = \{\mathcal{A}_{k,l} \mid l = 1, \dots, \lambda_k\}$; formal gilt $[\mathcal{A}_k]_l = \mathcal{A}_{k,l}$. Beispiel: Sämtliche Butcherbäume vierter Ordnung:

$$\mathcal{A}_4 = \left\{ \begin{array}{c} \circ \\ | \\ \circ \\ | \\ \circ \end{array} , \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} , \begin{array}{c} \circ \circ \\ | \\ \circ \circ \end{array} \cong \begin{array}{c} \circ \circ \\ | \\ \circ \circ \end{array} , \begin{array}{c} \circ \circ \\ | \\ \circ \circ \end{array} \right\}$$

Wie vorgreifend im Abschnitt 1.5.2 und folgendem bereits veranschaulicht, ist die Möglichkeit einer vollständigen *Parkettierung* des (gegebenenfalls auch gefärbten) Bethegitters \mathcal{E}_z einsichtig, minimal mit den Butcherbäumen $\mathcal{A}_{k=z+1}$.

$\eta_{k,l}$: Anzahl der Eckenklassen im Butcherbaum k ter Ordnung und l ter Struktur ohne Berücksichtigung etwaiger Färbung, siehe Tabelle 2.2.

l	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$\mathcal{A}_{1,l}$	1													
$\mathcal{A}_{2,l}$	1													
$\mathcal{A}_{3,l}$	2	2												
$\mathcal{A}_{4,l}$	2	2	2	2										
$\mathcal{A}_{5,l}$	3	4	4	2	3	4	3	4	2					
$\mathcal{A}_{6,l}$	3	5	4	4	5	2	4	4	2	3	5	4	4	...
\vdots														

Tabelle 2.2. Anzahl $\eta_{k,l}$ der Eckenklassen im Butcherbaum k ter Ordnung und l ter Struktur.

${}^c\mu_{k,l}$: Anzahl der strukturverschiedenen und nicht-isomorphen c gefärbten Butcherbäume k ter Ordnung und l ter Struktur. Man beachte, daß sich auch ${}^c\mu_{1,1} = c$ a priori angeben läßt: Der einzige Knoten des einen Baumes erster Ordnung, nämlich dessen Wurzel, kann auf c verschiedene Weisen gefärbt werden. Beispiel: Zwei- und viergefärbte Butcherbäume, Tabellen 2.3 und 2.4.

k	1	2	3	4	5	6	7	8	...
$\sum_{l=1}^{\lambda_k} {}^2\mu_{k,l}$	2	4	14	52	214	916	4116	18996	...
$\sum_{l=1}^{\lambda_k} {}^4\mu_{k,l}$	4	16	104	752	5996	50512	444256	4027360	...

Tabelle 2.3. Gesamtzahl zwei- [24:A038055] und vierfarbiger [24:A136793] Butcherbäume k ter Ordnung.

l	1	2	3	4	5	6	7	8	9	...
${}^2\mu_{1,l}$	2									
${}^2\mu_{2,l}$	4									
${}^2\mu_{3,l}$	8	6								
${}^2\mu_{4,l}$	16	12	16	8						
${}^2\mu_{5,l}$	32	24	32	16	32	24	20	24	10	
${}^2\mu_{6,l}$	64	48	64	32	64	48	40	48	20	...
\vdots										
${}^4\mu_{1,l}$	4									
${}^4\mu_{2,l}$	16									
${}^4\mu_{3,l}$	64	40								
${}^4\mu_{4,l}$	256	160	256	80						
${}^4\mu_{5,l}$	1024	640	1024	320	1024	640	544	640	140	
${}^4\mu_{6,l}$	4096	2560	4096	1280	4096	2560	2176	2560	560	...
\vdots										

Tabelle 2.4. Anzahl $\mu_{k,l}$ zwei- und vierfarbiger Butcherbäume k ter Ordnung und l ter Struktur (publiziert in der OEIS als Sequenzen A242353 und A242354), Aufschlüsselung von Tabelle 2.3.

Vereinfachend wird die Anzahl der Farben im Folgenden nur vermerkt, falls $c \neq 4$. ${}^c\mathcal{B}_{k,l,m}$: c gefärbter Butcherbaum mit k Ecken, $V = \{i \mid i = 1, 2, \dots, k\}$, der Strukturnummer l und der Färbungsnummer m . Beispiel: Sämtliche mit $X = \{\square, \blacksquare, \blacktriangle, \blacklozenge\}$ ($c = |X| = 4$) unter $\square < \blacksquare < \blacktriangle < \blacklozenge$ gefärbten Butcherbäume erster Ordnung, siehe auch Abschnitt 3.2 im Anhang:

$$\mathcal{B}_{1,1,1} = \odot, \mathcal{B}_{1,1,2} = \ominus, \mathcal{B}_{1,1,3} = \otimes, \mathcal{B}_{1,1,4} = \oplus$$

2.2.3. *Klammer- und Binärdarstellung.* Neben der Kodierung in einer Adjazenzmatrix, welche für beliebige Graphen möglich ist, jedoch im Allgemeinen bezüglich der Eckenanzahl quadratische Laufzeit zur Folge hat, lassen sich ausgerichtete Bäume durch einen *Tiefendurchlauf* auch eindeutig auf eine lineare Struktur, etwa eine Zeichenkette, abbilden. Dazu sind jeweils ein Symbol zum „Betreten“ und „Verlassen“ einer Ebene (etwa links beginnend) innerhalb eines Astes vonnöten, beispielsweise öffnende und schließende *Klammern* [43] oder die *Binärziffern*²⁹; Blätter, welche die Abschlussebene eines Astes darstellen, beziehungsweise der Butcherbaum erster Ordnung, \odot , werden so durch $()$ respektive 10_2 wiedergegeben. Die Gesamtlänge der zusammengesetzten Zeichenkette³⁰ beträgt bei $|V| = n$ Ecken folglich $2n$ Zeichen³¹. Beispiel: Sämtliche Butcherbäume zweiter und dritter Ordnung:

$$\odot \equiv (()) \equiv 1100_2, \quad \odot \equiv ((())) \equiv 111000_2, \quad \odot \equiv (()) \equiv 110100_2$$

Die Binärdarstellung spiegelt einerseits die (willkürlich gewählte) Strukturreihenfolge wider, wonach der „längste“, unverzweigte Baum gegebener Ordnung die niedrigste Strukturnummer l erhält.³² Beispiel: $k = 3$, erste Butcherbaumordnung mit $\lambda_3 = 2$ Strukturen:

$$\mathcal{A}_{3,1} > \mathcal{A}_{3,2} \leftarrow \odot > \odot \leftarrow (111000_2 \equiv 56_{10}) > (110100_2 \equiv 52_{10})$$

Zum anderen ermöglichen diese *bit*-Muster in eleganter Weise die in Abschnitt 2.1.14 geforderte eindeutige Gewichtung der Hauptäste eines Baumes anhand der Gesamtstruktur, also ohne eine rekursive Zerlegung durchführen zu müssen. Beispiel: $\mathcal{A}_{4,3}$, erster einen Isomorphismus aufweisender Butcherbaum:

$$\odot < \odot \leftarrow (11011000_2 \equiv 216_{10}) < (11100100_2 \equiv 228_{10})$$

²⁹Referenz [24:A057547] beispielsweise zeigt das Dezimalzahläquivalent der binärkodierte „*Catalanschen Berge ohne Meeresniveautäler*“, welche der Zahl an Butcherbäumen zuzüglich deren Isomorphismen entspricht. Unabhängig von der vorliegenden Arbeit wurden von ALOIS HEINZ in Referenz [24:A216648] die den Butcherbäumen entsprechenden *rooted trees* $\tilde{\mathcal{A}}_{k,l}$ publiziert; für diese gilt $\tilde{\mathcal{A}}_{k,\lambda_k-l+1} = \mathcal{A}_{k,l}$ mit $l = 1, 2, \dots, \lambda_k$.

³⁰Wir behandeln ohne weitere Diskussion Zeichenketten wie Tupel, deren Elemente einem vorgegebenen Alphabet entstammen, etwa $\Sigma = \{(), \emptyset, 1, \dots, 9, a, b, \dots, z\}$, und die in ihrer Abfolge festgelegten Bildungsgesetzen genügen. Zwecks Verzicht auf die sonst üblichen abgrenzenden Anführungszeichen werden Zeichenketten in dieser Arbeit in Schreibmaschinenschrift gesetzt.

³¹Für die datentechnische Behandlung ist ein hier der Übersichtlichkeit halber entfallendes *Terminationssymbol* erforderlich, beispielsweise \uparrow , welches den Speicherbedarf auf $2n + 1$ vergrößert.

³²Darin weicht unsere Baumsortierung, die zum Wirkungskreis von JOHN BUTCHERS Arbeiten [41, 80] konsistent ist, beispielsweise von der in Fußnote 29 erwähnten Ganzzahlfolge A216648 aus Referenz [24] ab, welcher die umgekehrte Ordnungsweise zugrundeliegt.

Unter Verwendung eines weiteren Zeichens aus einem entsprechend definierten Alphabet, welches der eine Ebene öffnenden Klammer vorangestellt ist, läßt sich ein Eckenattribut mitkodieren. Beispiel: Einige durch $\Sigma_X = \{a \leftarrow \blacksquare, b \leftarrow \blacksquare, c \leftarrow \blacksquare\}$ dreigefärbte Butcherbäume vierter Ordnung:

$$\begin{array}{c} \circ \circ \circ \\ \circ \circ \circ \\ \circ \circ \circ \end{array} \equiv a(c(b()c())) , \quad \begin{array}{c} \circ \circ \circ \\ \circ \circ \circ \\ \circ \circ \circ \end{array} \equiv c(a()b(a())) , \quad \begin{array}{c} \circ \circ \circ \\ \circ \circ \circ \\ \circ \circ \circ \end{array} \equiv b(b()c()c())$$

Die öffnende Klammer kann, wie ersichtlich, durch das eindeutig positionierte Farbsymbol ersetzt werden; der Beibehalt dient einzig besserer Lesbarkeit. Eine weitere Datenreduktion wird durch die Zusammenfassung identischer Äste innerhalb einer Ebene unter Nutzung eines Vorfaktors möglich³³. Beispiele (Wurzelvorfaktor aus Konsistenzgründen, stets 1):

$$\begin{array}{c} \circ \circ \\ \circ \circ \end{array} \equiv ((()()) \equiv 1(2()()) , \quad \begin{array}{c} \circ \circ \circ \\ \circ \circ \circ \\ \circ \circ \circ \end{array} \equiv b(b()c()c()) \equiv 1b(1b()2c())$$

Neben der Zeichenkettenrepräsentation ist – günstig für die maschinelle Verarbeitung – auch eine Binärdarstellung von Bäumen mit Eckenattribut möglich, indem etwa Struktur (S) und (beispielsweise) Färbung (F) in zwei Ganzzahlen getrennt behandelt werden. Beispiel: Zwei mit $\Sigma_X = \{00_2 \leftarrow \blacksquare, 01_2 \leftarrow \blacksquare, 10_2 \leftarrow \blacksquare\}$ gefärbte Butcherbäume:

$$\begin{array}{c} \circ \circ \\ \circ \circ \end{array} \equiv {}^S 11011000_2 {}^F 10000100_2 , \quad \begin{array}{c} \circ \circ \circ \\ \circ \circ \circ \\ \circ \circ \circ \end{array} \equiv {}^S 11010100_2 {}^F 01011111_2$$

Analog zur vorausgegangenen Ermittlung des „Strukturgewichts“ wird durch die Kodierung der Baumfärbung in einer Binärzahl die Auswahl eines Repräsentanten strukturell identischer isochromer Bäume anhand deren „Färbungsgewichte“ möglich. Beispiel: ${}^2\mathcal{B}_{3,2,2}$, erster einen „Isochromismus“³⁴ aufweisender durch $\Sigma_X = \{0_2 \leftarrow \blacksquare, 1_2 \leftarrow \blacksquare\}$ zweifarbigiger Butcherbaum:

$$\begin{array}{c} \circ \circ \\ \circ \circ \end{array} < \begin{array}{c} \circ \circ \\ \circ \circ \end{array} \leftarrow ({}^F 001_2 \equiv {}^F 110) < ({}^F 010_2 \equiv {}^F 210) \quad \text{mit} \quad {}^S 110100_2 \equiv {}^S 521_0$$

2.2.4. *Baumgruppe*. Wir nennen ein Objekt $\widetilde{\mathcal{T}}_1, \dots, \widetilde{\mathcal{T}}_n$, bestehend aus $n \in \mathbb{N}$ Bäumen \mathcal{T}_i , eine *Baumgruppe*, symbolisch

$$\mathcal{M} = \widetilde{\mathcal{T}}_1, \dots, \widetilde{\mathcal{T}}_n = \overline{(V_1, E_1), \dots, (V_n, E_n)} , \quad |\mathcal{M}| = n ; \quad (2.12)$$

im Falle $n = 1$ ist $\widetilde{\mathcal{T}} \equiv \mathcal{T}$. Formal ist $\llbracket \mathcal{M} \rrbracket_i = \mathcal{T}_i$. Baumgruppen, welche folglich Tupel untereinander nicht zusammenhängender Bäume sind und daher erst unter Ecken- und

³³Tatsächlich werden die Butcherbäume in der im Kapitel 3 vorgelegten Neuimplementierung des graphentheoretischen Rechenverfahrens zunächst in Zeichenkettenform erzeugt und erst später binärcodiert. Grund dafür ist die unmittelbare Lesbarkeit der Bäume, welche den ersten, für die Gesamtlaufzeit unbedeutenden Abschnitt der Programmentwicklung vereinfacht hat.

³⁴Da gefärbten Butcherbäumen in dieser gesamten Abhandlung zu EVA KALINOWSKIS graphentheoretischer Störungsrechnung tragende Bedeutung zukommt, erscheint die Bildung des Begriffes ‚hinsichtlich der Färbung Artgleiches‘ (Isochromismus) als Gegenstück zum Ausdruck ‚hinsichtlich der Form Artgleiches‘ (Isomorphismus) statthaft.

Kantenmengenvereinigung derselben in Walder uberfuhrt werden („Protowald“),

$$\mathcal{W} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_n = (V_1 \cup \dots \cup V_n, E_1 \cup \dots \cup E_n) \neq \mathcal{M}, \quad (2.13)$$

verhalten sich insbesondere unter nachfolgend definierter (gegebenenfalls mehrfacher) Produkt- sowie Potenzmengenbildung wie *ein* Baum.

2.2.5. *Baumprodukt.* Bei dem *einfachen Produkt zweier Butcherbaumengen* \mathbb{A}_{k_1} und \mathbb{A}_{k_2} der Ordnungen k_1 und k_2 handelt es sich um deren gewohnliches kartesisches Produkt (P),

$$\begin{aligned} \mathbb{A}_{k_1} \times \mathbb{A}_{k_2} &= \mathbb{A}_{k_1;k_2}^{\text{P}} \\ &= \left\{ \overline{\mathbb{A}_{k_1,l_1}, \mathbb{A}_{k_2,l_2}} \mid l_1 = 1, \dots, \lambda_{k_1}, l_2 = 1, \dots, \lambda_{k_2} \right\}_{<}, \end{aligned} \quad (2.14)$$

unter der Bedingung $k_1 < k_2$, auf welche durch das Zeichen]< hingewiesen wird. (Falls $k_1 = k_2$, ist der „Baumexponent“ $\mathbb{A}_{k_1}^2$ zu bilden, siehe Abschnitt 2.2.6; falls $k_1 > k_2$, formal ein Umbenennung der Indices, entsteht ein Isomorphismus, $\mathbb{A}_{k_1} \times \mathbb{A}_{k_2} \cong \mathbb{A}_{k_2} \times \mathbb{A}_{k_1}$, da nur die Reihenfolge der Elemente in jedem Produktpaar vertauscht werden.) Die Produktmenge $\mathbb{A}_{k_1;k_2}^{\text{P}}$ besteht aus

$$|\mathbb{A}_{k_1;k_2}^{\text{P}}| = \lambda_{k_1} \lambda_{k_2} \quad (2.15)$$

Paaren; ein solches Produktpaar bildet eine Baumgruppe. Beispiel ($\lambda_2 = 1, \lambda_3 = 2$):

$$\mathbb{A}_{2;3}^{\text{P}} = \left\{ \overline{\mathbb{A}_{2,l_1}, \mathbb{A}_{3,l_2}} \mid l_1 = 1, l_2 = 1, 2 \right\}_{<} = \left\{ \overline{\mathbb{A}_{2,1}, \mathbb{A}_{3,1}}, \overline{\mathbb{A}_{2,1}, \mathbb{A}_{3,2}} \right\}$$

Entsprechende Definition ist fur das (bezuglich der Indizierung auere) „*Strukturprodukt*“ (PS) zweier Gesamtmengen *c*gefarbter Butcherbume ${}^c\mathbb{B}_{k_1}$ und ${}^c\mathbb{B}_{k_2}$ der Ordnungen $k_1 < k_2$ gultig,

$$\begin{aligned} {}^c\mathbb{B}_{k_1} \times {}^c\mathbb{B}_{k_2} &= {}^c\mathbb{B}_{k_1;k_2}^{\text{PS}} \\ &= \left\{ {}^c\mathbb{B}_{k_1,l_1} \times {}^c\mathbb{B}_{k_2,l_2} \mid l_1 = 1, \dots, \lambda_{k_1}, l_2 = 1, \dots, \lambda_{k_2} \right\}_{<}, \end{aligned} \quad (2.16)$$

worin das (bezuglich der Indizierung innere) „*Farbungsprodukt*“ (PF) der Teilmengen ${}^c\mathbb{B}_{k_1,l_1}$ und ${}^c\mathbb{B}_{k_2,l_2}$ der Ordnungen k_1 und k_2 und Strukturnummern l_1 und l_2 explizit folgendermaen zu schreiben ist:

$$\begin{aligned} {}^c\mathbb{B}_{k_1,l_1} \times {}^c\mathbb{B}_{k_2,l_2} &= {}^c\mathbb{B}_{k_1,l_1;k_2,l_2}^{\text{PF}} \\ &= \left\{ \overline{{}^c\mathbb{B}_{k_1,l_1,m_1}, {}^c\mathbb{B}_{k_2,l_2,m_2}} \mid m_1 = 1, \dots, {}^c\mu_{k_1,l_1}, m_2 = 1, \dots, {}^c\mu_{k_2,l_2} \right\}_{<}. \end{aligned} \quad (2.17)$$

Fur die Produktmengen ${}^c\mathbb{B}_{k_1,l_1;k_2,l_2}^{\text{PF}}$ und ${}^c\mathbb{B}_{k_1;k_2}^{\text{PS}}$ gilt

$$|{}^c\mathbb{B}_{k_1,l_1;k_2,l_2}^{\text{PF}}| = {}^c\mu_{k_1,l_1} {}^c\mu_{k_2,l_2} \quad \text{und} \quad |{}^c\mathbb{B}_{k_1;k_2}^{\text{PS}}| = \sum_{l_1=1}^{\lambda_{k_1}} \sum_{l_2=1}^{\lambda_{k_2}} {}^c\mu_{k_1,l_1} {}^c\mu_{k_2,l_2}. \quad (2.18)$$

Im Falle des *gemischten Produktes* aus einer Baummenge und einer Baumproduktmenge, etwa $\mathbb{A}_3 \times \mathbb{A}_{2;4}^{\text{P}}$, oder aus zwei Produktmengen, beispielsweise $\mathbb{A}_{1;3}^{\text{P}} \times \mathbb{A}_{2;4}^{\text{P}}$, ist

oben genannte Ordnungsrelation im Allgemeinen nicht mehr direkt anwendbar; jedes implizite Mehrfachprodukt aus n Faktoren läßt sich jedoch in eine geordnete Abfolge von $n - 1$ Einfachproduktbildungen zerlegen, welche zu geordneten Produkttupeln (Baumgruppen) führt:

$$\begin{aligned} \mathbb{A}_{k_1} \times \mathbb{A}_{k_2} \times \cdots \times \mathbb{A}_{k_n} &= \mathbb{A}_{k_1; k_2; \dots; k_n}^p \quad \text{mit } k_1 < k_2 < \cdots < k_n \\ &= (\mathbb{A}_{k_1} \times \mathbb{A}_{k_2}) \times \cdots \times \mathbb{A}_{k_n} \\ &= \mathbb{A}_{k_1; k_2}^p \times \cdots \times \mathbb{A}_{k_n} = \mathbb{A}_{k_1; k_2; \dots; k_{n-1}}^p \times \mathbb{A}_{k_n}. \end{aligned} \quad (2.19)$$

Da bei $\mathbb{A}_{k_i} \times \mathbb{A}_{k_{i+1}} \times \cdots \times \mathbb{A}_{k_{i+m}}$ mit $k_i = k_{i+1} = \cdots = k_{i+m}$ eine Ordnungsrelation auf Strukturebene anzuwenden (und bei farbigen strukturgleichen Bäumen auf Färbungseben zu sortieren) ist, hat der (nachfolgend definierte) Baumexponent $\mathbb{A}_{k_i}^{m+1}$ Vorrang vor dem Baumprodukt.

2.2.6. Baumexponent. Unter dem *Baumexponenten* ist das \varkappa fache kartesische Produkt der Menge aller Butcherbäume \mathbb{A}_k der Ordnung k mit sich selbst,

$$\begin{aligned} \mathbb{A}_k^\varkappa &= \prod_{j=1}^{\varkappa} \left\{ \mathbb{A}_{k, l_j} \mid l_j = 1, \dots, \lambda_k \right\} \Big]_{\geq} \\ &= \left\{ \mathbb{A}_{k, l_1=1}, \dots, \mathbb{A}_{k, l_1=\lambda_k} \right\} \times \cdots \times \left\{ \mathbb{A}_{k, l_\varkappa=1}, \dots, \mathbb{A}_{k, l_\varkappa=\lambda_k} \right\} \Big]_{\geq} \\ &= \left\{ \overbrace{\mathbb{A}_{k, l_1=1}, \dots, \mathbb{A}_{k, l_\varkappa=1}, \dots, \mathbb{A}_{k, l_1=\lambda_k}, \dots, \mathbb{A}_{k, l_\varkappa=\lambda_k}} \right\} \\ &= \left\{ \mathcal{X}_1, \dots, \mathcal{X}_\varkappa \right\} = \mathcal{X}, \end{aligned} \quad (2.20)$$

unter Entfernung gleicher (\varkappa elementiger) Produkttupel – also Gruppen gemäß Definition in Abschnitt 2.2.4 aus jeweils \varkappa Bäumen – nach Anwendung der Ordnungsrelation $l_1 \geq l_2 \geq \cdots \geq l_\varkappa$, welche durch das Zeichen $\Big]_{\geq}$ symbolisiert wird, zu verstehen. Die Ursache dieser Ordnung liegt in der konventionsmäßigen, in der Produktbaumgruppe beizubehaltenden strukturellen Gewichtung der Butcherbäume innerhalb k ter Ordnung, wie in deren Binärdarstellung besonders leicht nachvollziehbar ist:

$$\left[\mathbb{A}_{k, l+1} \right]_2 < \left[\mathbb{A}_{k, l} \right]_2 \quad \forall l \in \{1, \dots, \lambda_k - 1\}. \quad (2.21)$$

Das bei der Berechnung des Baumexponenten zugrundeliegende kombinatorische Problem entspricht der Bildung \varkappa elementiger Multimengen mit Elementen aus \mathbb{A}_k , wie es beim \varkappa maligen Ziehen einer von λ_k verschiedenen Kugel aus einer Urne unter Zurücklegen und unter Vernachlässigung der Reihenfolge („Kombination mit Wiederholung“) auftritt,

$$\left| \mathbb{A}_k^\varkappa \right| = \binom{\lambda_k + \varkappa - 1}{\varkappa} = \frac{(\lambda_k + \varkappa - 1)!}{(\lambda_k - 1)! \varkappa!} = \varkappa. \quad (2.22)$$

Der Fall $g = 1$ mit $|A_k^1| = \lambda_k$ wird offenkundig „korrekt behandelt“ ($\widetilde{\mathcal{A}}_{k,l} \equiv \mathcal{A}_{k,l}$):

$$A_k^1 = \prod_{j=1}^1 \{ \mathcal{A}_{k,l_j} \mid l_j = 1, \dots, \lambda_k \}]_{\geq} = \{ \mathcal{A}_{k,1}, \dots, \mathcal{A}_{k,\lambda_k} \}.$$

Beispiel³⁵ für die Butcherbäume dritter Ordnung, $\lambda_3 = 2$, $|A_3^2| = \binom{3}{2} = 3$:

$$\begin{aligned} A_3^2 &= \prod_{j=1}^2 \{ \mathcal{A}_{3,l_j} \mid l_j = 1, 2 \}]_{\geq} = \{ \mathcal{A}_{3,1}, \mathcal{A}_{3,2} \} \times \{ \mathcal{A}_{3,1}, \mathcal{A}_{3,2} \}]_{\geq} \\ &= \{ \overbrace{\mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}} \}]_{\geq} \\ &\cong \{ \overbrace{\mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}} \}]_{\geq} \\ &= \{ \overbrace{\mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}} \}. \end{aligned}$$

Bei Behandlung gefärbter Butcherbäume gilt Entsprechendes sowohl für die Teilmenge ${}^c B_{k,l} \subseteq {}^c \mathcal{B}_k$ k ter Ordnung und l ter Struktur („Färbungsexponent“, Ordnungsrelation $m_1 \leq m_2 \leq \dots \leq m_x$, Zeichen $]_{\leq}$)³⁶,

$${}^c B_{k,l}^x = \prod_{j=1}^x \{ {}^c \mathcal{B}_{k,l,m_j} \mid m_j = 1, \dots, {}^c \mu_{k,l} \}]_{\leq} = \{ \mathcal{Y}_1, \dots, \mathcal{Y}_y \} = Y, \quad (2.23)$$

als auch, analog zu Gleichung (2.20), für die Gesamtmenge ${}^c \mathcal{B}_k$ k ter Ordnung („Struktur exponent“, Ordnungsrelation wiederum $l_1 \geq l_2 \geq \dots \geq l_x$, Zeichen $]_{\geq}$),

$${}^c \mathcal{B}_k^x = \prod_{j=1}^x \{ {}^c B_{k,l_j} \mid l_j = 1, \dots, \lambda_k \}]_{\geq} = \{ Y_1, \dots, Y_y \} = Y. \quad (2.24)$$

Eine Besonderheit gegenüber den strukturbezogenen Potenzen der gefärbten (wie ungefärbten, siehe Gleichung (2.21)) Butcherbäume stellt die Sortierung der farbbezogenen Potenz dar, welche sich im Gegensatz zu

$${}^S [{}^c B_{k,l+1}]_2 < {}^S [{}^c B_{k,l}]_2 \quad \forall l \in \{1, \dots, \lambda_k - 1\} \quad (2.25)$$

³⁵Abschnitt 3.1.2 greift erneut dieses „lehrreiche“ Beispiel auf, das einfachste mit $x > 1$ und $\lambda > 1$.

³⁶Diese vom Struktur exponenten, welcher die Reihenfolge der Äste festlegt, abweichende Ordnung wird nur bis einschließlich Abschnitt 3.2 (Erzeugung gefärbter Butcherbäume) verwendet; darüberhinaus gilt die in der Neuimplementierung verwendete, welche die Färbungen ebenfalls absteigend sortiert. Ursächlich dafür ist, daß einerseits die Ordnung *innerhalb der strukturgenerierenden Partitionen*, siehe Abschnitt 3.1, durch EVA KALINOWSKIS Grundlagenarbeiten [18, 19, 44] aufsteigend vorgegeben ist und zwecks Vergleichbarkeit eingehalten werden mußte. Dadurch erscheint die gleichlautende Färbungsordnung zunächst „didaktisch“ sinnvoll. Weil jedoch, begünstigt durch die Zeichenketten- oder *bit*-Muster-Kodierung, bei der Übersetzung der Butcherbaumfärbung in den zugehörigen quantenmechanischen (Ausgangs)zustand analog zur Schreibrichtung ein Tiefendurchlauf von links nach rechts vorgenommen werden muß und die Elektronenpropagationen konventionsmäßig entsprechend beginnen sollen, ist die gegenläufige Färbungsreihenfolge ab Abschnitt 3.4 notwendig.

auf die Ordnung

$${}^F[{}^c\mathcal{B}_{k,l,m+1}]_2 > {}^F[{}^c\mathcal{B}_{k,l,m}]_2 \quad \forall m \in \{1, \dots, {}^c\mu_{k,l} - 1\} \quad (2.26)$$

gründet. Für die speziellen Potenzmengen ${}^c\mathcal{B}_{k,l}^\varkappa$ und ${}^c\mathcal{B}_k^\varkappa$ gilt

$$|{}^c\mathcal{B}_{k,l}^\varkappa| = \binom{{}^c\mu_{k,l} + \varkappa - 1}{\varkappa} = y \quad \text{und} \quad |{}^c\mathcal{B}_k^\varkappa| = \binom{\lambda_k + \varkappa - 1}{\varkappa} = y. \quad (2.27)$$

Man beachte, daß die Produkttupel des Strukturexponenten Gruppen aus jeweils \varkappa *Baum-mengen* darstellen, welche sich erst nach Bildung des gegebenenfalls gemischten Färbungsproduktes innerhalb einer Gruppe in *Baumgruppen* „auflösen“; ein umfangreiches Beispiel findet sich in Abschnitt 3.2.

2.2.7. *Baumsumme.* Unter der *Addition zweier (wurzelloser) Bäume* \mathcal{R} und \mathcal{S} verstehen wir im Allgemeinen die Vereinigung beider Ecken- und Kantenmengen unter Ergänzung einer \mathcal{R} und \mathcal{S} verbindenden Kante $\{i, j\}$ zu einem neuen Baum \mathcal{T} , in Zeichen

$$\begin{aligned} \mathcal{T} &= \mathcal{R} \underset{i,j}{\uplus} \mathcal{S} \\ &= \{V(\mathcal{R}) \cup V(\mathcal{S}), E(\mathcal{R}) \cup E(\mathcal{S}) \cup \{i, j\} \mid i \in V(\mathcal{R}), j \in V(\mathcal{S})\}. \end{aligned} \quad (2.28)$$

Unter der Voraussetzung disjunkter Eckenmengen, $V(\mathcal{R}) \cap V(\mathcal{S}) = \emptyset$ (äquivalent dazu: $i \notin V(\mathcal{S}) \wedge j \notin V(\mathcal{R})$), welche trotz gleicher Numerierung der Ecken stets erfüllt ist (formal: $V(\mathcal{R}) = \{1_{\mathcal{R}}, 2_{\mathcal{R}}, \dots\}$ und $V(\mathcal{S}) = \{1_{\mathcal{S}}, 2_{\mathcal{S}}, \dots\}$), ist die Addition kommutativ,

$$\mathcal{R} \underset{i,j}{\uplus} \mathcal{S} = \mathcal{S} \underset{j,i}{\uplus} \mathcal{R}, \quad (2.29)$$

und assoziativ,

$$\left(\mathcal{R} \underset{i,j}{\uplus} \mathcal{S} \right) \underset{k,l}{\uplus} \mathcal{T} = \mathcal{R} \underset{i,j}{\uplus} \left(\mathcal{S} \underset{k,l}{\uplus} \mathcal{T} \right). \quad (2.30)$$

Um im „Summenbaum“ \mathcal{T} , Gleichung (2.28), zu einer konsistenten Benennung der Ecken (und Kanten) zu gelangen (formal: $V(\mathcal{T}) = \{1_{\mathcal{T}}, 2_{\mathcal{T}}, \dots\}$), ist eine Transformationsvorschrift $\boxplus : \mathcal{S} \rightarrow \mathcal{S}'$ vonnöten, welche gemäß vorausgegangener Definition in Abschnitt 2.1.12 einen Isomorphismus darstellt und bis auf Neunummerierung der Ecken den Graphen unverändert läßt:

$$\mathcal{S}' = \boxplus_k \mathcal{S} = \left\{ \boxplus_k V(\mathcal{S}), \boxplus_k E(\mathcal{S}) \right\}, \quad \text{wurin} \quad (2.31)$$

$$\boxplus_k V(\mathcal{S}) = \{k+1, k+2, \dots, k+|V(\mathcal{S})|\} \quad \text{und} \quad (2.32)$$

$$\begin{aligned} \boxplus_k E(\mathcal{S}) &= \{\{i_1+k, j_1+k\}, \{i_2+k, j_2+k\}, \dots \\ &\dots, \{i_{|E(\mathcal{S})|}+k, j_{|E(\mathcal{S})|}+k\}\}. \end{aligned} \quad (2.33)$$

Für den weiteren, eingeschränkten Gebrauch der Baumsumme betrachten wir fortan nur noch die Addition eines (Butcher-)Baumes \mathcal{S} mit dessen erster Ecke, der Wurzel, *an* die Wurzel eines Baumes \mathcal{R} . Da einerseits die stärker bindende, also vor der Addition

auszuführende Transformation \boxplus nur von der Eckenanzahl $|V(\mathcal{R})|$ des ersten Summanden abhängt und andererseits auch die zu verbindenden Ecken 1 und $|V(\mathcal{R})| + 1$ fest vorgegeben sind, lassen sich beide Zeichen ohne Mehrdeutigkeit zusammenfassen:

$$\begin{aligned} \mathcal{T} &= \mathcal{R} \underset{1, |V(\mathcal{R})|+1}{\boxplus} \left(\underset{|V(\mathcal{R})|}{\boxplus} \mathcal{S} \right) = \mathcal{R} \underset{1, |V(\mathcal{R})|+1}{\boxplus} \mathcal{S}' = \mathcal{R} \boxplus \mathcal{S} \\ &= \{V(\mathcal{R}) \cup V(\mathcal{S}'), E(\mathcal{R}) \cup E(\mathcal{S}') \cup \{1, |V(\mathcal{R})| + 1\}\}. \end{aligned} \quad (2.34)$$

Mögliche Eckenattribute sind entsprechend zu berücksichtigen, $F(\mathcal{R}) \circ F(\mathcal{S}')$. Anschaulich wird durch Vorschrift (2.34) \mathcal{S} im „Summenbaum“ \mathcal{T} zu einem Ast des Stammes \mathcal{R} , welcher an dessen Wurzel abzweigt. Man beachte, daß eine solche Additionsanweisung außer im trivialen Falle $\mathcal{S} = \mathcal{R}$ zwar nicht mehr kommutativ ist, $\mathcal{R} \boxplus \mathcal{S} \neq \mathcal{S} \boxplus \mathcal{R}$, wohl aber zu topologisch äquivalenten Summen führt, $\mathcal{R} \boxplus \mathcal{S} \simeq \mathcal{S} \boxplus \mathcal{R}$. Beispiel: $(\mathcal{A}_{1,1} \boxplus \mathcal{A}_{2,1} = \mathcal{A}_{3,1}) \neq (\mathcal{A}_{2,1} \boxplus \mathcal{A}_{1,1} = \mathcal{A}_{3,2})$; das Blatt von $\mathcal{A}_{2,1}$ ist mit einer Markierung versehen:

$$\left(\begin{array}{c} \circ \\ \circ \boxplus \circ \\ \circ \end{array} \right) = \begin{array}{c} \circ \\ \circ \\ \circ \end{array} \neq \left(\begin{array}{c} \circ \\ \circ \boxplus \circ \\ \circ \end{array} \right) \quad (\text{Skizze 2.3})$$

Eine besondere Eleganz weist die (oben definierte) „Wurzeladdition“ in der Binär-darstellung (\boxplus_2) auf. Anknüpfend an vorausgegangenes Beispiel, Skizze 2.3. läßt sich mit $a = 2$ Dualstellen pro Ecke schreiben:

$$\mathcal{R} \boxplus_2 \mathcal{S}: \mathcal{A}_{1,1} \boxplus_2 \mathcal{A}_{2,1} = \mathcal{A}_{3,1} \text{ mit } k_{\mathcal{S}} = |V(\mathcal{S} = \mathcal{A}_{2,1})| = 2$$

$$10_2 \boxplus_2 1100_2 = \left(\underbrace{\overbrace{10_2/2 \cdot 2^{ak_{\mathcal{S}}}}^{=10000_2}}_{=1_2} + 1100_2 \right) \cdot 2 = 11100_2 \cdot 2 = 111000_2$$

Das „Öffnen“ und „Schließen“ einer Wurzel zum Anfügen eines neuen Astes entspricht der Division (auf Maschinenebene: *bit shift* nach rechts) beziehungsweise Multiplikation (*bit shift* nach links) mit 2 (eine Dualstelle), die Verschiebung der Indices zur Neu-numerierung der Multiplikation mit $2^{ak_{\mathcal{S}}}$ ($ak_{\mathcal{S}}$ Dualstellen). Man beachte, daß – nur scheinbar, aufgrund der Schreibrichtung – die „Richtung“ der Numerierungsverschiebung, siehe Gleichungen (2.31) und (2.34), bei binärer Addition (\boxplus_2) folglich nicht von der Eckenanzahl des ersten Summanden (dort $|V(\mathcal{R})|$), sondern vom zweiten bestimmt wird. Entsprechend folgt bei Vertauschung:

$$\mathcal{S} \boxplus_2 \mathcal{R}: \mathcal{A}_{2,1} \boxplus_2 \mathcal{A}_{1,1} = \mathcal{A}_{3,2} \text{ mit } k_{\mathcal{R}} = |V(\mathcal{R} = \mathcal{A}_{1,1})| = 1$$

$$1100_2 \boxplus_2 10_2 = \left(\underbrace{\overbrace{1100_2/2 \cdot 2^{ak_{\mathcal{R}}}}^{=11000_2}}_{=110_2} + 10_2 \right) \cdot 2 = 11010_2 \cdot 2 = 110100_2$$

Im Allgemeinen gilt für ungefärbte Butcherbäume

$$\mathcal{A}_{k_{\mathcal{R}}, l_{\mathcal{R}}} \boxplus_2 \mathcal{A}_{k_{\mathcal{S}}, l_{\mathcal{S}}} = 2 \cdot \left(2^{ak_{\mathcal{S}}-1} \cdot \mathcal{A}_{k_{\mathcal{R}}, l_{\mathcal{R}}} + \mathcal{A}_{k_{\mathcal{S}}, l_{\mathcal{S}}} \right) \Big]_2, \quad (2.35)$$

worin das Symbol $]_2$ explizit auf deren Binärdarstellung hinweist. Bei c gefärbten Butcherbäumen müssen Struktur- und Färbungszahl, gekennzeichnet durch $^S[$ und $^F[$, getrennt behandelt und zudem die Anzahl $b = \lceil \log_2(c) \rceil$ der Dualstellen für die Kodierung der Farben berücksichtigt werden:

$${}^c\mathcal{B}_{k_{\mathcal{R}},l_{\mathcal{R}},m_{\mathcal{R}}} \uplus_2 {}^c\mathcal{B}_{k_{\mathcal{S}},l_{\mathcal{S}},m_{\mathcal{S}}} = {}^S[2 \cdot (2^{ak_{\mathcal{S}}-1} \cdot {}^c\mathcal{B}_{k_{\mathcal{R}},l_{\mathcal{R}},m_{\mathcal{R}}} + {}^c\mathcal{B}_{k_{\mathcal{S}},l_{\mathcal{S}},m_{\mathcal{S}}})]_2 \\ {}^F[2^{bk_{\mathcal{S}}} \cdot {}^c\mathcal{B}_{k_{\mathcal{R}},l_{\mathcal{R}},m_{\mathcal{R}}} + {}^c\mathcal{B}_{k_{\mathcal{S}},l_{\mathcal{S}},m_{\mathcal{S}}}]_2 \cdot \quad (2.36)$$

2.3. Zusammenfassung. Eingebettet in den Kontext grundlegender Begriffe der Graphentheorie sind in diesem Kapitel zunächst in Abschnitt 2.1 zielgerichtet wesentliche „Objekte“ für das Störungsrechnungsverfahren und deren Eigenschaften eingeführt und untersucht worden. Erstrangig wurde dabei die Definition des Baumbegriffes als Sonderfall allgemeiner Graphen (Abschnitt 2.1.5) behandelt, gefolgt von der weitergehenden problemmotivierten Spezialisierung gemäß dem jeweiligen „Aufbauprinzip“ auf Permutationsbäume \mathcal{P}_n (Abschnitt 2.1.10) und Binärbäume \mathcal{D}_g (Abschnitt 2.1.11). Überdies wurde vorbereitend Farbe als Attribut der Baumecken (Abschnitt 2.1.15) in die Notation integriert, durch welche später Besetzungszahlen dargestellt werden können.

Hauptsächlicher Gegenstand von Abschnitt 2.2 ist nach der formalen Definition von Bethegittern durch Cayleybäume $\mathcal{C}_{z,g}$ mit $g \rightarrow \infty$ (Abschnitt 2.2.1) der zentrale Begriff des Butcherbaumes, ungefärbt als $\mathcal{A}_{k,l} \in \mathcal{A}_k$, c gefärbt als ${}^c\mathcal{B}_{k,l,m} \in {}^c\mathcal{B}_{k,l} \in {}^c\mathcal{B}_k$, welcher einen (in der vorliegenden Arbeit elektronenbesetzten) Bethegitterausschnitt repräsentiert. Nach Vorstellung effizienter Kodierungsmöglichkeiten von Baumstruktur und -färbung (Abschnitt 2.2.3) wurde abschließend auf spezielle Produkte von Baumengen, beispielsweise im ungefärbten Falle $\mathcal{A}_{k_1} \times \mathcal{A}_{k_2}$ und \mathcal{A}_k^{\times} (Abschnitte 2.2.5 und 2.2.6), sowie die Vereinigung zweier Bäume zu einem „Summenbaum“, im ungefärbten Falle $\mathcal{A}_{k_1,l_1} \uplus \mathcal{A}_{k_2,l_2}$ (Abschnitt 2.2.7), eingegangen.

Während letztgenannte Operationen ausschließlich für die Beschreibung der Baumerzeugung in den Abschnitten 3.1 und 3.2 relevant sind, haben die Formulierungen des ersten Teilabschnitts bis einschließlich Klammer- und Binärdarstellung von Butcherbäumen tragende Bedeutung für die Abschnitte 3.4 und 3.5 des sich anschließenden Kapitels.

3. Optimierter Gesamtalgorithmus

Im hier beginnenden zentralen Abschnitt dieser Studie werden jene Verfahren beschrieben, welche für die praktische maschinelle Ausführung der eingangs dargelegten algebraisch-graphentheoretischen Störungsrechnung vonnöten sind. Eingeschlossen ist jeweils die Illustration durch zumindest ein ausführliches Beispiel, das die jeweiligen teils erst in höheren Ordnungen auftretenden Besonderheiten an möglichst einfachen Fällen erfaßt. Sämtlich beruhen diese hinsichtlich Nutzung jenseits der Grundzustandseigenschaften ansatzweise verallgemeinerten und in einigen Details technisch verfeinerten [44] Algorithmen auf der von EVA KALINOWSKI entwickelten und innerhalb eines umfangreichen

Machbarkeitsbeweises im Anschluß an ihre Dissertation bis $\mathcal{O}(U^{-11})$ angewendeten Methode [18, 19, 21].

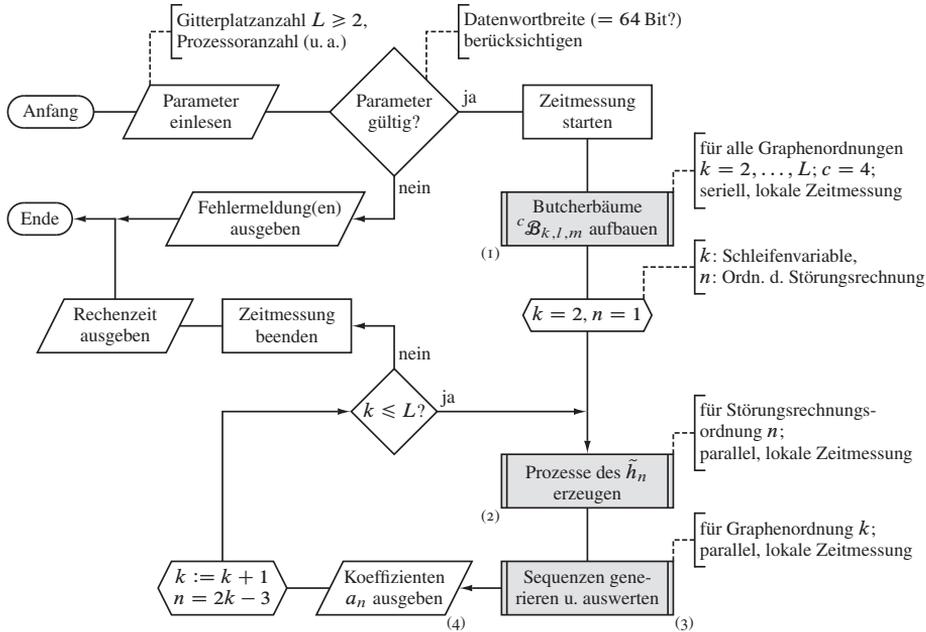


Abbildung 3.1. Ablaufplan des Hauptprogramms. Das Unterprogramm zu Flußdiagrammelement (1) ist in Abbildung 3.4 (Abschn. 3.2 ab S. 169) „expandiert“ dargestellt; ebenso sind die Programmteile Element (2) in Abbildung 3.5 (Abschn. 3.3 ab S. 176) und Element (3) in den Abbildungen 3.6 und 3.9 (Abschn. 3.4 u. 3.5 ab S. 194) detailliert ausgeführt.

Eine schematische Übersicht über den Programmablauf gibt Abbildung 3.1, welche sich unter Bezugnahme auf Abschnitt 1.6 wie folgt beschreiben läßt:

- I Ausführungparameter einlesen: Obergrenze der Gitterplatzanzahl $L \geq 2$, also die maximale Größe der Bethegitterausschnitte³⁷ und somit die höchste berücksichtigte Störungsrechnungsordnung $n = 2L - 3$, Anzahl der in den parallelisierten Abschnitten genutzten Prozessoren sowie Steuerparameter für den Umfang an Kontrollausgaben (Speichernutzung etc.) und Zwischenergebnissen

³⁷Daß der zentrale Hauptprogrammparameter nicht die Ordnung des Koeffizienten der $1/U$ -Reihe, sondern die Ordnung der zugehörigen Butcherbäume ist, hat entwicklungsgeschichtliche Gründe. Die Unterprogramme sind hingegen aus Effizienzgründen so formuliert, daß der primäre (Aufruf)parameter der „algorithmisch natürliche“ ist – je nachdem, ob beispielsweise durch das Gitter $k = (n + 3)/2$ oder etwa durch die Elektromobewegung darauf $n = 2k - 3$ im Vordergrund stehen.

- 2 Parameter auf Gültigkeit prüfen; bei Negativbefund Programm mit entsprechenden Fehlermeldung(en) beenden. Dabei ist die Rechnerarchitektur zu berücksichtigen, da alle graphenbezogenen Objekte und viele der weiteren Datenstrukturen in „langen“ vorzeichenlosen ganzen Zahlen mit bis zu 64 genutzten Dualstellen gespeichert werden. Insbesondere wird für Zwischenergebnisse für $\mathcal{O}(U^{-15})$ auf zusammengesetzte Ganzzahlen im Intervall $[0, 2^{128} - 1]$ zurückgegriffen.
- 3 Globale Zeitmessung starten
- 4 Innerhalb eines Unterprogrammes, Flußdiagrammelement (1), mit lokaler Zeitmessung sämtliche vierfarbigen Butcherbäume $\mathcal{B}_{k,l,m}$ für alle Graphenordnungen $k = 2, \dots, L$ unter Vorgabe der Graphen mit $k = 1$ gemäß Abschnitt 3.2 seriell-iterativ aufbauen³⁸
- 5 Schleife³⁹ (explizit) über alle Graphenordnungen $k = 2, 3, \dots, L$ und somit (implizit) über alle Störungsordnungen $n = 1, 3, \dots, 2L - 3$:
 - 5.1 Hüpfprozesse $\hat{\mathcal{P}}^{(n)}$ des \tilde{h}_n , bestehend aus Beitragsgewichten und Operatorexponenten, für die Grundzustandsenergie n ter Ordnung der Störungsrechnung in einem teilweise parallelisierten Unterprogramm mit lokaler Zeitmessung gemäß Abschnitt 3.3 berechnen, Flußdiagrammelement (2)
 - 5.2 Hüpfsequenzen $\mathcal{H}^{(n)}$ der Ordnung n der durch die Butcherbäume k ter Ordnung repräsentierten Bethegitterausschnitte (Ausgangszustände) in einem vollständig parallelisierten Unterprogramm mit lokaler Zeitmessung gemäß Abschnitt 3.4 rekursiv generieren, den jeweiligen Sequenzbetrag zur Grundzustandsenergie Abschnitt 3.5 entsprechend bestimmen und zum Zwischenergebnis a_n addieren, Flußdiagrammelement (3)
 - 5.3 Koeffizienten $a_n = E_0^{(n)}$ des Korrekturterms $E_0^{(n)}(U)$ in n ter Ordnung der Störungsreihe ausgeben, siehe Gleichung (3.1)
- 6 Zur nächsten Ordnung $k := k + 1$ (und entsprechend $n = 2k - 3$) übergehen, Schritt 5, falls $k < L$; sonst die globale Zeitmessung anhalten, Rechenzeit ausgeben und Hauptprogramm beenden

Abschließend kann die Grundzustandsenergie nach Gleichung (1.12) in der Form

$$\begin{aligned}
 E_0(U) &= \sum_{k=2}^L \frac{1}{U^{2k-3}} E_0^{(2k-3)} + \mathcal{O}\left(\frac{1}{U^{2L-2}}\right) \\
 &= a_1 U^{-1} + a_3 U^{-3} + \dots + a_{2L-3} U^{-2L+3} + \mathcal{O}(U^{-2L+2}) \quad (3.1)
 \end{aligned}$$

³⁸Aufgrund der aufwandsbedingt ineffizienten Synchronisation von Schreibzugriffen auf die dreifach indizierte Datenstruktur der zentralen „Baumtafel“ erfolgt die Ausführung derzeit nicht parallel, vergleiche dazu Fußnote 41 und Abschnitt 5.1.

³⁹Optional kann diese aufgrund eines entsprechenden Steuerparameters auch nur für $k = L$ (respektive $n = 2L - 3$) ausgeführt werden, wiewohl der Zeitaufwand für die Berechnung der Vergleichsergebnisse aller Ordnungen bis $L - 1$ im Kontrast zur Ordnung L vernachlässigbar ist, siehe Abschnitt 5.1.1.

niedergeschrieben werden. Programmschritt 5.2 und folgender sind im Übrigen derart umgesetzt, daß die für das Falicov-Kimball-Modell relevanten Sequenzen aus der „Obermenge“ der Hubbard-Sequenzen separat zum Reihenkoeffizienten $a_n^{\text{FKM}} = E_0^{(n),\text{FKM}}$ verarbeitet werden, für welche bei korrektem Ablauf $a_n^{\text{FKM}} = 0$ für $n > 1$ gemäß Gleichung (1.3) gelten muß.

Sämtliche beteiligte Unterprogramme werden im Weiteren detailliert dargelegt; die Beschreibung der darin enthaltenen Teilalgorithmen und Hilfsfunktionen findet sich in den Anhängen 1 und 2.

3.1. Erzeugung der ungefärbten Butcherbäume. Der hier beschriebene Algorithmus, welcher iterativ sämtliche ungefärbte Butcherbäume \mathcal{A}_k der Ordnungen $k = 2, \dots, L$ unter alleiniger Vorgabe des ersten Baumes $\mathcal{A}_{1,1} = (\{1\}, \{\})$ und $\lambda_1 = |\mathcal{A}_1| = 1$ generiert, ist in dieser geschlossenen Form für das Gesamtverfahren in der vorgelegten Neuimplementierung nicht von Belang. Dennoch erfolgt eine (informelle) Darstellung aus zwei Gründen: Einerseits kommen erstmalig insbesondere die im Abschnitt 2.2 beschriebenen Formalismen – vorbereitend auf die Erzeugung gefärbter Butcherbäume, eine Verallgemeinerung – beispielhaft zur Anwendung; andererseits sind die ungefärbten Bäume für eine mögliche künftige vereinfachte Nutzung des algebraisch-graphentheoretischen Rechenverfahrens für bosonische Quantensysteme mit $S = 0$ (nicht entartet) grundlegend⁴⁰, siehe Abschnitt 6.1.

3.1.1. Allgemeines Vorgehen.

- 1 Ausführungparameter vom aufrufenden Hauptprogramm übernehmen: Gitterplatzanzahl $L \geq 2$ sowie Steuerparameter für den Umfang an Kontrollausgaben (Speichernutzung etc.) und Zwischenergebnissen, siehe Flußdiagrammelement (1) in Abbildung 3.3
- 2 Lokale Zeitmessung beginnen (Schritt aus Platzgründen nicht im Programmablaufschema eingezeichnet)
- 3 Für $k = 2, \dots, L$: Die innerhalb gegebener Baumordnung k fortlaufende Strukturnummer mit $l = 0$ initialisieren
- 4 Partitionierung der Zahl $p = k - 1$ in q lexikographisch (aufsteigend) geordnete Summanden r_a ($a = 1, \dots, q$), beginnend mit $q = 1$ (also $r_1 = p$), endend mit $q = p$ (folglich $r_1 = r_2 = \dots = r_p = 1$)⁴¹:

$$p = \sum_{a=1}^q r_a \quad \text{mit} \quad 0 < r_1 \leq r_2 \leq \dots \leq r_q \quad (3.2)$$

⁴⁰Für die Behandlung von Bosonensystemen mit $S > 0$ sind gefärbte Butcherbäume notwendig. Die Frage nach der Farbanzahl wird nochmals in Fußnote 61 angeschnitten und insbesondere in Abschnitt 3.4.1 vertieft, dort bezogen auf Teilchen mit halbzahligen Spin.

⁴¹Da für p insgesamt ν_p Partitionen existieren, siehe Gleichung (2.11) und Tabelle 2.5, welche entsprechend dem weiteren Programmablauf unabhängig verarbeitet werden können – es wird darin nur auf Butcherbäume der Ordnungen $k' = 1, 2, \dots, k - 1$ zurückgegriffen –, besteht die Möglichkeit zur Parallelisierung.

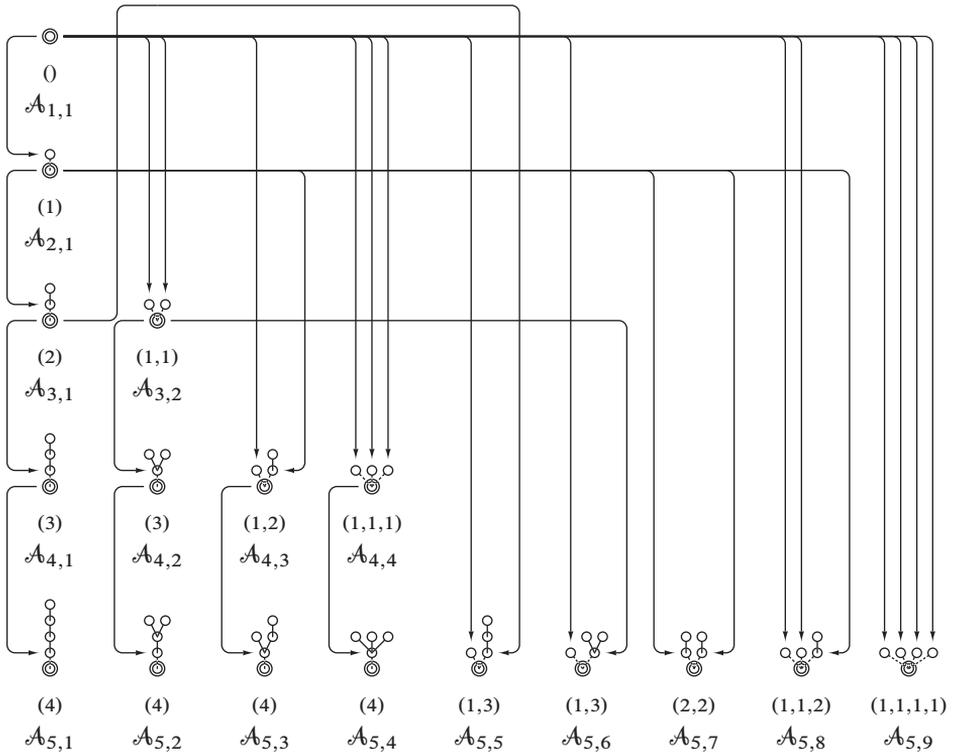


Abbildung 3.2. Iterative Konstruktion aller ungefärbten Butcherbäume $\mathcal{A}_{k,l}$ (Ordnung k , Struktur l) mit bis zu $k = 5$ Ecken. Die zugrundeliegenden Partitionen von $p = k - 1$ sind in runden Klammern als Tupel angegeben.

Das Tupel der Summanden einer Partition, welche der Ordnung der späteren Äste entsprechen, wird im Weiteren mit R bezeichnet; es gilt: $|R| = q$.

5 Einteilung der r_a in u Gruppen ϱ_α gleicher Summanden ($\alpha = 1, \dots, u$), Element (2); r'_α ist dabei ein „repräsentativer Vertreter der jeweiligen Gruppe“ aus R :

$$p = \sum_{a=1}^q r_a = \sum_{\alpha=1}^u \varrho_\alpha r'_\alpha, \quad \sum_{\alpha=1}^u \varrho_\alpha = q, \quad (3.3)$$

in Tupelschreibweise, $g_\xi = \sum_{\alpha=1}^\xi \varrho_\alpha$:

$$R = \underbrace{(r_1, r_2, \dots, r_{g_1})}_{\varrho_1, r'_1=r_{g_1}} \underbrace{(r_{g_1+1}, r_{g_1+2}, \dots, r_{g_2})}_{\varrho_2, r'_2=r_{g_2}} \dots \underbrace{(r_{g_{u-1}+1}, r_{g_{u-1}+2}, \dots, r_{g_u})}_{\varrho_u, r'_u=r_{g_u}} \quad (3.4)$$

mit $r_1 = r_2 = \dots = r_{g_1} \neq r_{g_1+1} = r_{g_1+2} = \dots$

6 Berechnung des folgenden Mehrfachproduktes von Baumexponenten, Element (3):

$$A_{r'_1}^{q_1} \times A_{r'_2}^{q_2} \times \dots \times A_{r'_u}^{q_u} = \prod_{\alpha=1}^u A_{r'_\alpha}^{q_\alpha} = \mathbb{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_x\} \quad (3.5)$$

darin mit der Strukturnummer s_a von $[\mathcal{X}_i]_a$, also jener des a ten Baumes ($a = 1, \dots, q$) in der Baumgruppe \mathcal{X}_i ,

$$\mathcal{X}_i = \underbrace{\mathcal{A}_{r'_1, s_1}, \dots, \mathcal{A}_{r'_1, s_w}}_{q_1}, \dots, \underbrace{\mathcal{A}_{r'_u, s_w}, \dots, \mathcal{A}_{r'_u, s_q}}_{q_u} \quad \forall \mathcal{X}_i \in \mathbb{X} \quad (3.6)$$

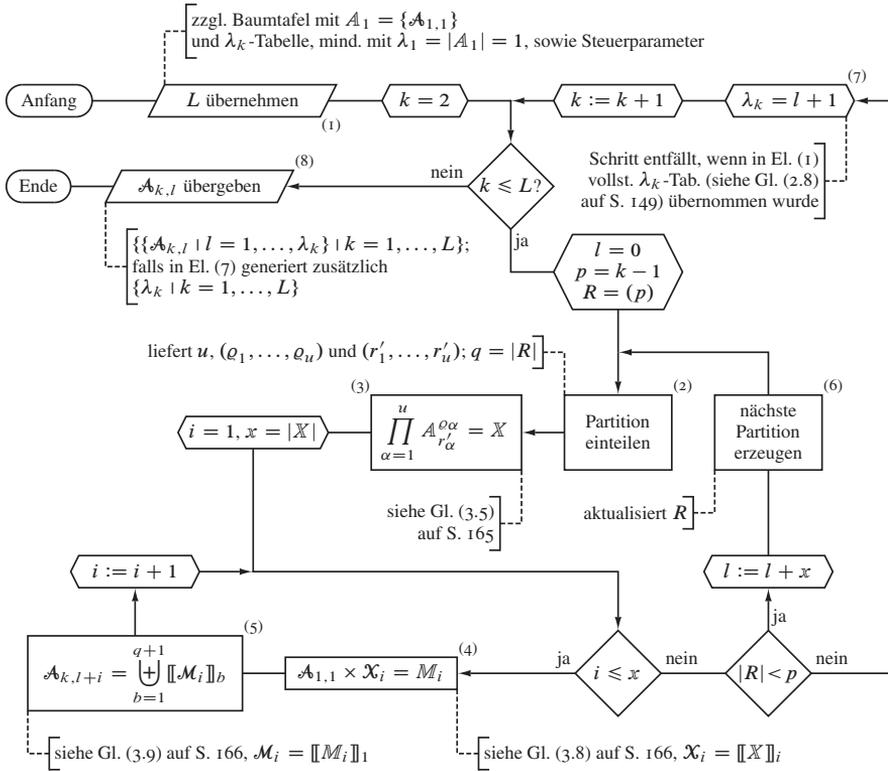


Abbildung 3.3. Ablaufplan eines Unterprogramms für die Erzeugung der ungefärbten Butcherbäume.

(ausführlicher)

$$\begin{aligned}
\mathcal{X}_1 &= \underbrace{\mathcal{A}_{r'_1,1}, \dots, \mathcal{A}_{r'_1,1}, \dots}_{\varrho_1}, \underbrace{\mathcal{A}_{r'_u,1}, \dots, \mathcal{A}_{r'_u,1}, \dots}_{\varrho_u} \\
\mathcal{X}_2 &= \underbrace{\mathcal{A}_{r'_1,1}, \dots, \mathcal{A}_{r'_1,1}, \dots}_{\varrho_1}, \underbrace{\mathcal{A}_{r'_u,1}, \dots, \mathcal{A}_{r'_u,2}}_{\varrho_u} \\
&\vdots \\
\mathcal{X}_x &= \underbrace{\mathcal{A}_{r'_1,\lambda_{r'_1}}, \dots, \mathcal{A}_{r'_1,\lambda_{r'_1}}}_{\varrho_1}, \underbrace{\mathcal{A}_{r'_u,\lambda_{r'_u}}, \dots, \mathcal{A}_{r'_u,\lambda_{r'_u}}}_{\varrho_u}
\end{aligned}$$

wobei

$$x = |\mathbb{X}| = \binom{\lambda_{r'_1} + \varrho_1 - 1}{\varrho_1} \cdot \dots \cdot \binom{\lambda_{r'_u} + \varrho_u - 1}{\varrho_u} = \prod_{\alpha=1}^u \binom{\lambda_{r'_\alpha} + \varrho_\alpha - 1}{\varrho_\alpha} \quad (3.7)$$

7 Bildung intermediärer Produkte für $i = 1, 2, \dots, x$, Element (4):

$$\mathcal{A}_{1,1} \times \mathcal{X}_i = \left\{ \overbrace{\mathcal{A}_{1,1}, \mathcal{A}_{r'_1,s_1}, \dots, \mathcal{A}_{r'_1,s_w}, \dots, \mathcal{A}_{r'_u,s_w}, \dots, \mathcal{A}_{r'_u,s_q}} \right\} = \mathbb{M}_i \quad (3.8)$$

(Formal handelt es sich bei \mathbb{M}_i um eine einelementige Menge, bestehend aus der Baumgruppe $\mathcal{M}_i = \llbracket \mathbb{M}_i \rrbracket_1$.)8 Für $i = 1, 2, \dots, x$: Addition der q Äste $\llbracket \mathcal{M}_i \rrbracket_b$, b ter Baum der Gruppe $\llbracket \mathbb{M}_i \rrbracket_1$ mit $b = 2, 3, \dots, q + 1$, an die Wurzel $\llbracket \mathcal{M}_i \rrbracket_1$, also $\mathcal{A}_{1,1}$, des neuen Butcherbaumes $\mathcal{A}_{k,l+i}$, Element (5):

$$\mathcal{A}_{k,l+i} = \bigoplus_{b=1}^{q+1} \llbracket \mathcal{M}_i \rrbracket_b = \left((\mathcal{A}_{1,1} \uplus \mathcal{A}_{r'_1,s_1}) \uplus \dots \right) \uplus \mathcal{A}_{r'_u,s_q} \quad (3.9)$$

- 9 Strukturnummernzähler $l := l + x$ aktualisieren und zur nächsten Partition von p übergehen, Schritt 4, Element (6), falls $|R| < p$; sonst Schleife (nach insgesamt ν_p Zyklen, siehe Gleichung (2.11)) beenden
- 10 Gegebenenfalls $\lambda_k = l + x$ speichern (beim letzten Schleifendurchlauf, $|R| = p$, ist $x = 1$ für alle Werte von k), Element (7), und zur nächsten Baumordnung $k := k + 1$ übergehen, Schritt 1, falls $k < L$; sonst Schleife beenden
- 11 Lokale Zeitmessung beenden und Rechenzeit ausgeben (Schritt nicht im Programmablaufschema eingezeichnet)
- 12 Verfahren beenden und Butcherbäume bis einschließlich Graphenordnung L dem aufrufenden Hauptprogramm übergeben, Element (8)

3.1.2. *Beispiel.* Als beispielhafte Anwendung des zuvor beschriebenen Verfahrens betrachten wir die Erzeugung jener Butcherbäume, bei welchen erstmals $q = 3$ Partitionssummanden $R = (r_1, r_2, r_3)$ auftreten, derer zwei mit einem zugehörigen $\lambda > 1$ „für einen interessanten Baumexponenten“ gleich sind und deren erster mit $\lambda_{r_1} = 1$ hinsichtlich Aufwandverringerung von möglichst kleiner Ordnung ist, folglich $(r_1 = 1) < (r_2 = r_3)$ mit $u = 2$. Laut Tabelle 2.1 existieren zuerst für die dritte Ordnung⁴², somit $r_2 = 3$, zwei strukturell verschiedene Bäume; daher ist $p = r_1 + r_2 + r_3 = 7$ und $k = 8$.

Das Vorgehen für den Aufbau der \mathcal{A}_8 , bereits $\lambda_8 = 115$ strukturell unterschiedliche nicht-isomorphe Bäume (siehe Tabelle 2.1), verlangt, hier nur formal erwähnt, die Bildung aller Partitionen der Zahl 7, siehe folgende Tabelle, innerhalb einer Schleife über die Programmschritte 4 bis 9.

i	1	2	...	7	8	9	...	15
R_i	(7)	(1, 6)	...	(1, 1, 1, 4)	(1, 3, 3)	(2, 2, 3)	...	(1, 1, 1, 1, 1, 1, 1)
q_i	1	2	...	4	3	3	...	7
λ'_i	48	20	...	4	3	2	...	1

Gemäß genannten Kriterien betrachten wir, beginnend mit Schritt 5, die achte der insgesamt $\nu_7 = 15$ Partitionen (vergleiche Tabelle 2.5), die nach Gleichung (3.3) wie folgt aufzuteilen ist:

$$R = \left(\underbrace{1}_{q_1=1}, \underbrace{3, 3}_{r'_1=1, \quad q_2=2, \quad r'_2=3} \right).$$

Daraus ergibt sich, vergleiche Schritt 6 und insbesondere Gleichung (3.5) in der allgemeinen Vorgehensweise, ein solches Produkt zweier Baumexponenten:

$$\begin{aligned} \mathcal{A}_1^1 \times \mathcal{A}_3^2 &= \left\{ \overbrace{\mathcal{A}_{1,1}} \right\} \times \left\{ \overbrace{\mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}} \right\} \\ &= \left\{ \overbrace{\mathcal{A}_{1,1}} \right\} \times \left\{ \overbrace{\mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}} \right\} \\ &= \left\{ \overbrace{\mathcal{A}_{1,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,1}, \mathcal{A}_{1,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}, \mathcal{A}_{1,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}} \right\} \\ &= \{ \mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3 \} = \mathcal{X} \end{aligned}$$

⁴²Dieses Beispiel wurde aus gleichem Grunde schon in Abschnitt 2.2.6 verwendet; Details zur Berechnung von \mathcal{A}_3^2 siehe dort.

mit $|\mathbb{X}| = \binom{1}{1} \cdot \binom{3}{2} = 3$. Im folgenden Programmschritt 7 werden dementsprechend die drei „Zwischenprodukte“

$$\begin{aligned} \llbracket \mathcal{A}_{1,1} \times \mathcal{X}_1 \rrbracket_1 &= \overbrace{\mathcal{A}_{1,1}, \mathcal{A}_{1,1}, \mathcal{A}_{3,1}, \mathcal{A}_{3,1}} = \mathcal{M}_1, \\ \llbracket \mathcal{A}_{1,1} \times \mathcal{X}_2 \rrbracket_1 &= \overbrace{\mathcal{A}_{1,1}, \mathcal{A}_{1,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,1}} = \mathcal{M}_2 \quad \text{und} \\ \llbracket \mathcal{A}_{1,1} \times \mathcal{X}_3 \rrbracket_1 &= \overbrace{\mathcal{A}_{1,1}, \mathcal{A}_{1,1}, \mathcal{A}_{3,2}, \mathcal{A}_{3,2}} = \mathcal{M}_3 \end{aligned}$$

gebildet, aus welchen im letzten Schritt 8, rückgreifend auf die Butcherbäume erster und dritter Ordnung,

$$\mathcal{A}_{1,1} = \circ, \quad \mathcal{A}_{3,1} = \begin{array}{c} \circ \\ | \\ \circ \end{array}, \quad \mathcal{A}_{3,2} = \begin{array}{c} \circ \circ \\ | \\ \circ \end{array},$$

drei neue Bäume mit sich (siehe obige Tabelle) an $l = \sum_{n=1}^7 \lambda'_n = 102$ anschließenden Strukturnummern erwachsen:

$$\mathcal{A}_{8,103} = \bigoplus_{b=1}^4 \llbracket \mathcal{M}_1 \rrbracket_b = \left((\circ \uplus \circ) \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} = \left(\begin{array}{c} \circ \\ | \\ \circ \end{array} \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \\ | \\ \circ \circ \end{array},$$

$$\mathcal{A}_{8,104} = \bigoplus_{b=1}^4 \llbracket \mathcal{M}_2 \rrbracket_b = \left((\circ \uplus \circ) \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} = \left(\begin{array}{c} \circ \\ | \\ \circ \end{array} \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \circ \\ | \\ \circ \end{array} \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \circ \\ | \\ \circ \circ \end{array},$$

$$\mathcal{A}_{8,105} = \bigoplus_{b=1}^4 \llbracket \mathcal{M}_3 \rrbracket_b = \left((\circ \uplus \circ) \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} = \left(\begin{array}{c} \circ \\ | \\ \circ \end{array} \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \circ \\ | \\ \circ \circ \end{array} \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \circ \circ \\ | \\ \circ \circ \end{array}.$$

Deutlich zeigt sich an diesem Beispiel die Wirkung des in Abschnitt 2.2.6 definierten und in Schritt 6 angewendeten geordneten Baumexponenten, da der Isomorphismus zu $\mathcal{A}_{8,104}$,

$$\left((\circ \uplus \circ) \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} = \left(\begin{array}{c} \circ \\ | \\ \circ \end{array} \uplus \begin{array}{c} \circ \\ | \\ \circ \end{array} \right) \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \\ | \\ \circ \circ \end{array} \uplus \begin{array}{c} \circ \circ \\ | \\ \circ \end{array} = \begin{array}{c} \circ \circ \circ \\ | \\ \circ \circ \end{array} \cong \begin{array}{c} \circ \circ \circ \\ | \\ \circ \circ \end{array},$$

gemäß Festlegung für die Butcherbäume nicht generiert wird.

Wird die Butcherbaumerzeugung auf Grundlage der in Abschnitt 2.2.3 erläuterten komprimierten Zeichenkettenrepräsentation ausgeführt, so liegen die im hiesigen Beispiel benötigten Bäume in folgender Gestalt vor:

$$\mathcal{A}_{1,1} = 1(), \quad \mathcal{A}_{3,1} = 1(1(1())), \quad \mathcal{A}_{3,2} = 1(2()).$$

Für die Ausführung von Programmschritt 8 sind zunächst die konsistenter Form halber angegebene Wurzelmultiplizitäten zu entfernen und bei der Konkatenation der *strings*⁴³

⁴³Siehe Fußnote 30

die neu auftretenden Mehrzähligkeiten der Äste einzufügen. Obige Baumadditionen sind somit wie folgt zu schreiben:

$$\begin{aligned}\mathcal{A}_{8,103} &= 1 \circ (\circ 1 \circ \circ) \circ 2 \circ (1(1(\circ))) \circ = 1(1(\circ)2(1(1(\circ)))) , \\ \mathcal{A}_{8,104} &= 1 \circ (\circ 1 \circ \circ) \circ 1 \circ (2(\circ)) \circ 1 \circ (1(1(\circ))) \circ = 1(1(\circ)1(2(\circ))1(1(1(\circ)))) , \\ \mathcal{A}_{8,105} &= 1 \circ (\circ 1 \circ \circ) \circ 2 \circ (2(\circ)) \circ = 1(1(\circ)2(2(\circ))) .\end{aligned}$$

3.2. Erzeugung gefärbter Butcherbäume. Das iterative Verfahren zur Erzeugung gefärbter Butcherbäume stellt im Ablaufplan der graphentheoretischen Störungsrechnung vor der Reihenentwicklung des Hamiltonoperators, Abschnitte 1.4 und 3.3, und der Generierung der Hüpfsequenzen, Abschnitte 1.5.5 und 3.4, den ersten der drei Hauptschritte dar, siehe Element 1 in Flußdiagramm 3.1. Da sich die vorliegende Arbeit auf Elektronen beschränkt, genügte bei Ausgangszuständen mit einem Teilchen auf jedem Gitterplatz die Zweifärbung. Um das Verfahren jedoch auch auf Anregungen ausweiten zu können, also auch doppelt- und unbesetzte Ecken zuzulassen, erfolgt praktisch der Aufbau viergefärbter Bäume⁴⁴. Technisch werden, wie in Abschnitt 5.1 noch umfassender erläutert, sämtliche Strukturen im Sinne von Abschnitt 2.2.3 binärkodiert verarbeitet; insbesondere gilt dies für die Baumaddition.

3.2.1. Allgemeines Vorgehen. Die Schritte 1 bis 5 entsprechen jenen im Algorithmus zur Erzeugung der ungefärbten Butcherbäume, siehe Abschnitt 3.1.1. Bei der Datenübernahme vom aufrufenden Hauptprogramm, vergleiche Flußdiagrammelement (1) in Abbildung 3.4, müssen jedoch die Farbenanzahl c und die entsprechenden Bäume erster Ordnung ${}^c\mathcal{B}_{1,1,\tau} = (\{1\}, \{\}, (\chi_\tau))$ mit $\tau = 1, 2, \dots, c$ berücksichtigt werden.

6 Berechnung des folgenden Mehrfachproduktes (bezüglich der Struktur) von Baumengenexponenten, Element (2):

$${}^c\mathbb{B}_{r'_1}^{\varrho_1} \times {}^c\mathbb{B}_{r'_2}^{\varrho_2} \times \dots \times {}^c\mathbb{B}_{r'_u}^{\varrho_u} = \prod_{\alpha=1}^u {}^c\mathbb{B}_{r'_\alpha}^{\varrho_\alpha} = \mathbb{Y} = \{Y_1, Y_2, \dots, Y_y\} \quad (3.10)$$

darin mit der Strukturnummer s_a von $\llbracket Y_i \rrbracket_a$, also jener der a ten Baummenge ($a = 1, \dots, q$) in der Baumengengruppe Y_i ,

$$Y_i = \overbrace{({}^c\mathbb{B}_{r'_1, s_1}, \dots, {}^c\mathbb{B}_{r'_1, s_w}, \dots, {}^c\mathbb{B}_{r'_u, s_w}, \dots, {}^c\mathbb{B}_{r'_u, s_q})}^{\forall Y_i \in \mathbb{Y}} \quad (3.11)$$

$\underbrace{\hspace{10em}}_{\varrho_1} \qquad \qquad \qquad \underbrace{\hspace{10em}}_{\varrho_u}$

wobei

$$y = |\mathbb{Y}| = \binom{\lambda_{r'_1} + \varrho_1 - 1}{\varrho_1} \cdot \dots \cdot \binom{\lambda_{r'_u} + \varrho_u - 1}{\varrho_u} = \prod_{\alpha=1}^u \binom{\lambda_{r'_\alpha} + \varrho_\alpha - 1}{\varrho_\alpha} \quad (3.12)$$

⁴⁴„Unter- und oberhalb“ der Gitterhalbfüllung sind nur drei Farben erforderlich: $\{\emptyset \leftarrow \square, \uparrow \leftarrow \blacksquare, \downarrow \leftarrow \blacksquare\}$ respektive $\{\uparrow \leftarrow \blacksquare, \downarrow \leftarrow \blacksquare, \updownarrow \leftarrow \blacksquare\}$. Die Frage der Farbenanzahl und -mengen wird nochmals in Abschnitt 3.4.1 aufgegriffen und vertieft.

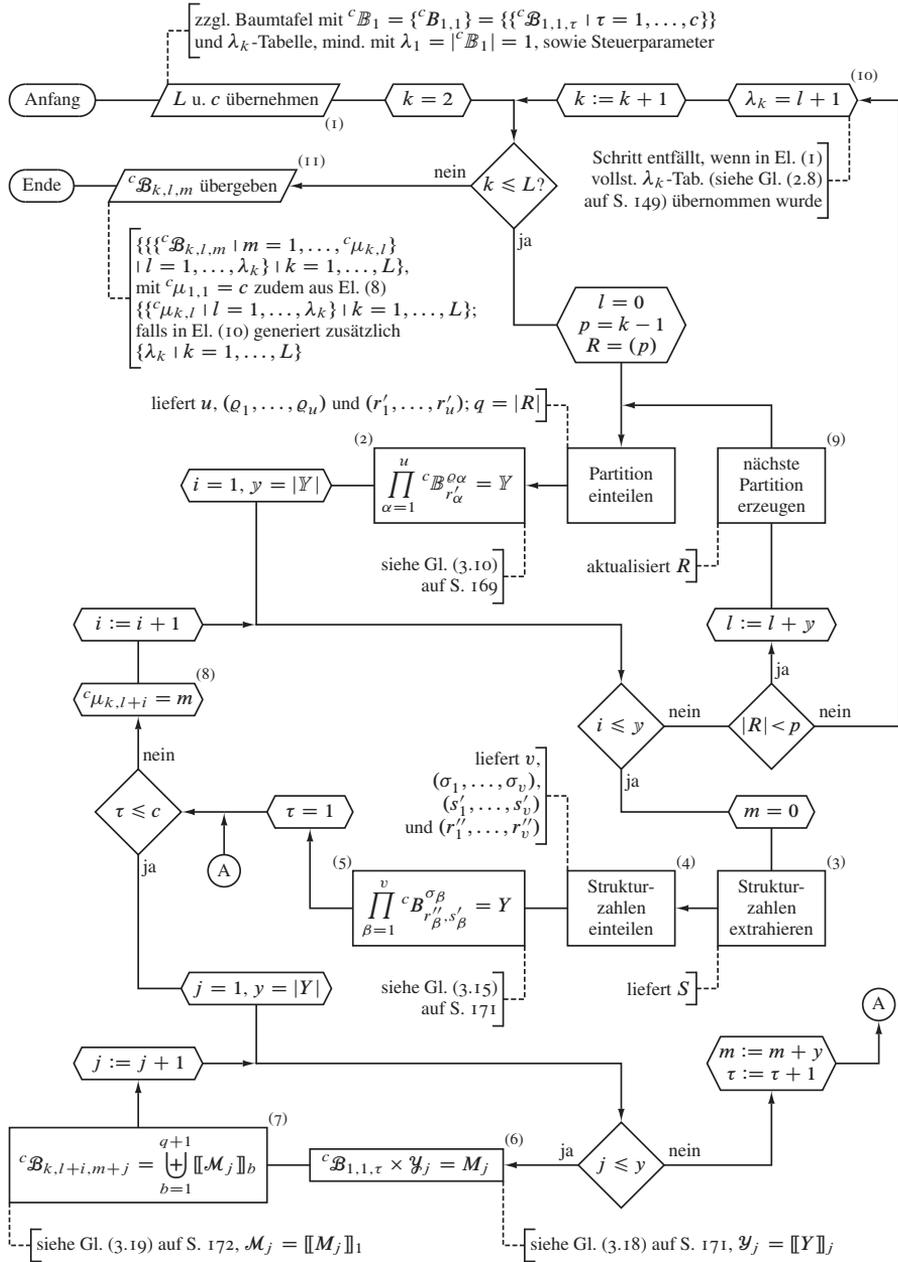


Abbildung 3.4. Ablaufplan des Unterprogramms für die Erzeugung der c -gefärbten Butcherbäume, Flußdiagrammelement (1) in Abbildung 3.1.

Das Tupel der Strukturzahlen s_a aus Y_i wird im Weiteren mit S bezeichnet; es gilt:
 $|S| = |R| = q$.

7 Für $i = 1, 2, \dots, y$:

7.1 Fortlaufende Färbungsnummer mit $m = 0$ initialisieren

7.2 „Extraktion“ und Einteilung der s_a in v Gruppen σ_β benachbarter gleicher Strukturzahlen ($\beta = 1, \dots, v$) unter der Voraussetzung gleicher Ordnung ϱ_α ($\alpha = 1, \dots, u$), Elemente (3) und (4):

$$\sum_{\beta=1}^v \sigma_\beta = q, \quad \sum_{\alpha=1}^u \varrho_\alpha = q, \quad (3.13)$$

in Tupelschreibweise, $h_\xi = \sum_{\beta=1}^\xi \sigma_\beta$:

$$S = \left(\underbrace{s_1, \dots, s_{h_1}}_{\sigma_1, s'_1=s_{h_1}, r''_1=r'_1}, \underbrace{s_{h_1+1}, \dots, s_{h_2}}_{\sigma_2, s'_2=s_{h_2}, r''_2=r'_1}, \dots, \underbrace{s_{h_{v-1}+1}, \dots, s_{h_v}}_{\sigma_v, s'_v=s_{h_v}, r''_v=r'_u} \right) \quad (3.14)$$

mit $s_1 = s_2 = \dots = s_{h_1} \neq s_{h_1+1} = s_{h_1+2} = \dots \wedge r'_1 \neq r'_2 \neq \dots \neq r'_u$

7.3 Berechnung des folgenden Mehrfachproduktes (bezüglich der Färbung) von Baumexponenten, Element (5):

$${}^c B_{r''_1, s'_1}^{\sigma_1} \times {}^c B_{r''_2, s'_2}^{\sigma_2} \times \dots \times {}^c B_{r''_v, s'_v}^{\sigma_v} = \prod_{\beta=1}^v {}^c B_{r''_\beta, s'_\beta}^{\sigma_\beta} = Y = \{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_y\} \quad (3.15)$$

darin mit der Färbungsnummer t_a von $[[\mathcal{Y}_j]]_a$, also jener des a ten Baumes ($a = 1, \dots, q$) in der Baumgruppe \mathcal{Y}_j ,

$$\mathcal{Y}_j = \underbrace{{}^c \mathcal{B}_{r''_1, s'_1, t_1}, \dots, {}^c \mathcal{B}_{r''_1, s'_1, t_w}}_{\sigma_1}, \dots, \underbrace{{}^c \mathcal{B}_{r''_u, s'_v, t_{\tilde{w}}}, \dots, {}^c \mathcal{B}_{r''_u, s'_v, t_q}}_{\sigma_v} \quad \forall \mathcal{Y}_j \in Y \quad (3.16)$$

wobei

$$y = |Y| = \prod_{\beta=1}^v \binom{{}^c \mu_{r''_\beta, s'_\beta}^{\sigma_\beta} + \sigma_\beta - 1}{\sigma_\beta} \quad (3.17)$$

7.4 Für $\tau = 1, 2, \dots, c$: Schleife über alle Farben der Wurzel:

7.4.1 Bildung intermediärer Produkte für $j = 1, 2, \dots, y$, Element (6):

$${}^c \mathcal{B}_{1,1,\tau} \times \mathcal{Y}_j =$$

$$\left\{ {}^c \mathcal{B}_{1,1,\tau}, {}^c \mathcal{B}_{r''_1, s'_1, t_1}, \dots, {}^c \mathcal{B}_{r''_1, s'_1, t_w}, \dots, {}^c \mathcal{B}_{r''_u, s'_v, t_{\tilde{w}}}, \dots, {}^c \mathcal{B}_{r''_u, s'_v, t_q} \right\} = M_j \quad (3.18)$$

(Formal handelt es sich bei M_j um eine einelementige Menge, bestehend aus der Baumgruppe $\mathcal{M}_j = [[M_j]]_1$.)

7.4.2 Für $j = 1, 2, \dots, y$: Addition der q Äste $\llbracket \mathcal{M}_j \rrbracket_b$, b ter Baum der Gruppe $\llbracket \mathcal{M}_j \rrbracket_1$ mit $b = 2, 3, \dots, q + 1$, an die Wurzel $\llbracket \mathcal{M}_j \rrbracket_1$, also ${}^c\mathcal{B}_{1,1,\tau}$, des neuen Butcherbaumes ${}^c\mathcal{B}_{k,l+i,m+j}$, Element (7):

$${}^c\mathcal{B}_{k,l+i,m+j} = \bigoplus_{b=1}^{q+1} \llbracket \mathcal{M}_j \rrbracket_b = \left(({}^c\mathcal{B}_{1,1,\tau} \uplus {}^c\mathcal{B}_{r'_1,s'_1,t'_1}) \uplus \dots \right) \uplus {}^c\mathcal{B}_{r'_u,s'_v,t'_q} \quad (3.19)$$

7.4.3 Färbungsnummernzähler $m := m + y$ aktualisieren und zur nächsten Wurzelfarbe übergehen, Schritt 7.4, falls $\tau < c$; sonst Schleife beenden und ${}^c\mu_{k,l+i} = m$ abspeichern, Element (8)

7.5 Zur nächsten Strukturproduktzeile übergehen, Schritt 7, falls $i < y$; sonst Schleife beenden

- 8 Strukturnummernzähler $l := l + y$ aktualisieren und zur nächsten Partition von p übergehen, Schritt 4, Element (9), falls $|R| < p$; sonst Verfahren (nach insgesamt ν_p Zyklen, siehe Gleichung (2.11)) beenden
- 9 Gegebenenfalls $\lambda_k = l + y$ speichern (beim letzten Schleifendurchlauf, $|R| = p$, ist $y = 1$ für alle Werte von k), Element (10), und zur nächsten Baumordnung $k := k + 1$ übergehen, Schritt 1, falls $k < L$; sonst Schleife beenden
- 10 Lokale Zeitmessung beenden und Rechenzeit ausgeben (Schritt nicht im Programmablaufschema eingezeichnet)
- 11 Verfahren beenden und Butcherbäume bis einschließlich Graphenordnung L dem aufrufenden Hauptprogramm übergeben, Element (11)

Die Anzahl der Butcherbäume mit $c = 4$ in der Rechnung für $L = 10$ (Störungsrechenungsordnung U^{-17}) sowie den jeweiligen Rechenzeitbedarf auf einem CISC-server vom Typ HP AM447A [84] bei Algorithmus-bedingt serieller Verarbeitung (ein Prozessorkern belegt) zeigt Tabelle 3.1. Man beachte, daß aufgrund der iterativen Baumkonstruktion die Gesamtzeit für die Ordnung L die Summe aller Rechenzeiten für $k = 2, \dots, L$ ist; diese beträgt somit beispielsweise für die zehnte Ordnung circa 45100 Sekunden⁴⁵.

3.2.2. *Beispiel.* Zur Demonstration des Algorithmus' zur Erzeugung gefärbter Butcherbäume formulieren wir zum ungefärbten Falle ähnliche Bedingungen: $q = 3$ Partitionssummanden $R = (r_1, r_2, r_3)$, von welchen zwei gleich und sämtliche kleinstmöglich sind, also $r_1 = r_2 = 1$ und $r_3 = 2$ mit $u = 2$ sowie $p = r_1 + r_2 + r_3 = 4$ und folglich $k = 5$. Da die Anwendung des strukturbezogenen Baumexponenten bereits im Abschnitt 3.1.2 erläutert wurde, hier abweichend nur auf Baumengen verallgemeinert wird und an dieser Stelle der neuauftretende färbungsbezogene Baumexponent im Vordergrund steht, ist das dreimalige Auftreten von $\lambda = 1$, siehe Tabelle 2.1, akzeptabel.

⁴⁵Diese Zeitangaben beziehen sich auf den mit Zeichenketten operierenden Algorithmus, vergleiche Fußnote 33; bei Verwendung der sehr effizienten Ganzzahlkodierung, siehe dazu auch das Beispiel in Abschnitt 2.2.7, darf mindestens eine Größenordnung weniger veranschlagt werden.

k	5	6	7	8	9	10
\mathfrak{b}_k	5996	50512	444256	4027360	37383044	353486320
Pz./s	$\leq 0,04$		1,09	35,8	1240	43900

Tabelle 3.1. Gesamtzahl vierfarbiger Butcherbäume in Fortsetzung von Tabelle 2.3. In der letzten Zeile ist orientierungshalber jeweils die Prozessorzeit (Pz.) für die k te Ordnung angegebenen, welche bei hiesiger serieller Verarbeitung identisch mit der Maschinen- beziehungsweise Realzeit ist; technische Aspekte dazu finden sich detailliert im Abschnitt 5.1.

Um die ohnehin eindrucksvolle Problemkomplexität nicht weiter zu erhöhen, beschränken wir uns auf die $c = 2$ Farben $\{\blacksquare, \blacksquare\}$, was der primären Anwendung des Gesamtverfahrens – Ausgangszustände mit halber Bandfüllung ohne Anregung – entspricht.

Innerhalb der Butcherbaumengen ${}^c\mathcal{B}_5$ existieren (unabhängig von der Farbenanzahl c) $\lambda_5 = 9$ strukturell unterschiedliche nicht-isomorphe Bäume, welche aus den Partitionen der Zahl 4, siehe folgende Tabelle, hervorgehen; diese Zerlegungen werden in der Schleife über die Programmschritte 4 bis 8 generiert.

i	1	2	3	4	5
R_i	(4)	(1, 3)	(2, 2)	(1, 1, 2)	(1, 1, 1, 1)
q_i	1	2	2	3	4
λ'_i	4	2	1	1	1

In Erfüllung der obigen Anforderungen legen wir dem weiteren Programmablauf, beginnend mit Schritt 5, die vierte der insgesamt $\nu_4 = 5$ Partitionen (vergleiche Tabelle 2.5) zugrunde und teilen diese, Gleichung (3.3) entsprechend, folgendermaßen auf:

$$R = \left(\underbrace{1, 1}_{e_1=2, r'_1=1}, \underbrace{2}_{e_2=1, r'_2=2} \right).$$

Das nun mit Gleichung (3.10) in Schritt 6 zu bildende Mehrfachprodukt bezüglich der Struktur lautet

$$\begin{aligned} {}^c\mathcal{B}_1^2 \times {}^c\mathcal{B}_2^1 &= \left\{ \overbrace{{}^c\mathcal{B}_{1,1}, {}^c\mathcal{B}_{1,1}} \right\}_{\geq} \times \left\{ \overbrace{{}^c\mathcal{B}_{2,1}} \right\}_{\geq} = \left\{ \overbrace{{}^c\mathcal{B}_{1,1}, {}^c\mathcal{B}_{1,1}} \right\} \times \left\{ \overbrace{{}^c\mathcal{B}_{2,1}} \right\} \\ &= \left\{ \overbrace{{}^c\mathcal{B}_{1,1}, {}^c\mathcal{B}_{1,1}, {}^c\mathcal{B}_{2,1}} \right\} = \{Y\} = \mathbb{Y} \end{aligned}$$

mit $|\mathbb{Y}| = \binom{2}{2} \cdot \binom{1}{1} = 1$; dementsprechend ist die sich anschließende Schleife über alle im Produkt enthaltenen Baumengengruppen ab Programmschritt 7 nur einmal auszuführen. Nach Initialisierung des Färbungsnummernzählers $m = 0$, Schritt 7.1, erfolgt

die Gruppierung der Strukturnummern aus Y , Schritt 7.2,

$$S = \left(\underbrace{1, 1}_{r'_1=1}, \underbrace{2}_{r'_2=2} \right),$$

$$\begin{array}{cc} \sigma_1=2, & s'_1=1 \\ r''_1=1 & \sigma_2=1, & s'_2=2 \\ & r''_2=2 \end{array}$$

woraus unmittelbar (Schritt 7.3) das Mehrfachprodukt bezüglich der Färbung, nunmehr unter eingangs erwähnter Festlegung auf $c = 2$ mit ${}^2\mu_{1,1} = 2$ und ${}^2\mu_{2,1} = 4$ (siehe Tabelle 2.4), zusammengesetzt werden kann:

$$\begin{aligned} {}^2B_{1,1}^2 \times {}^2B_{2,1}^1 &= \left\{ \overbrace{{}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}} \right\} \times \\ &\quad \times \left\{ \overbrace{{}^2\mathcal{B}_{2,1,1}, {}^2\mathcal{B}_{2,1,2}, {}^2\mathcal{B}_{2,1,3}, {}^2\mathcal{B}_{2,1,4}} \right\} \Big]_{\leq} \\ &= \left\{ \overbrace{{}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}} \right\} \times \\ &\quad \times \left\{ \overbrace{{}^2\mathcal{B}_{2,1,1}, {}^2\mathcal{B}_{2,1,2}, {}^2\mathcal{B}_{2,1,3}, {}^2\mathcal{B}_{2,1,4}} \right\} \\ &= \left\{ \overbrace{{}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{2,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{2,1,2}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{2,1,3}}, \right. \\ &\quad \overbrace{{}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{2,1,4}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,2}}, \\ &\quad \overbrace{{}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,3}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,4}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,1}}, \\ &\quad \left. \overbrace{{}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,3}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,4}} \right\} \\ &= \{y_1, \dots, y_{12}\} = Y; \end{aligned}$$

$|Y| = \binom{3}{2} \cdot \binom{4}{1} = 12$. Im Rahmen der Schleife über alle „Wurzelfarben“, Abschnitt 7.4, sind jeweils zwölf „Zwischenprodukte“ mit ${}^2\mathcal{B}_{1,1,\tau} \forall \tau \in \{1, 2\}$, zu bilden, dargestellt zunächst mit $\tau = 1$:

$$\begin{aligned} \llbracket {}^2\mathcal{B}_{1,1,1} \times y_1 \rrbracket_1 &= \overbrace{{}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{2,1,1}} = \mathcal{M}_1, \\ &\quad \vdots \\ \llbracket {}^2\mathcal{B}_{1,1,1} \times y_{12} \rrbracket_1 &= \overbrace{{}^2\mathcal{B}_{1,1,1}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{1,1,2}, {}^2\mathcal{B}_{2,1,4}} = \mathcal{M}_{12}. \end{aligned}$$

Unter Rückgriff auf die zweifarbigen Butcherbäume erster und zweiter Ordnung,

$${}^2\mathcal{B}_{1,1,1} = \odot, \quad {}^2\mathcal{B}_{1,1,2} = \odot, \quad {}^2\mathcal{B}_{2,1,1} = \odot, \quad {}^2\mathcal{B}_{2,1,2} = \odot, \quad {}^2\mathcal{B}_{2,1,3} = \odot, \quad {}^2\mathcal{B}_{2,1,4} = \odot,$$

können nun im letzten Schritt 7.4.2 jede der intermediären Baumgruppen zu einem neuen Butcherbaum mit der Strukturnummer $l = \sum_{i=1}^4 \lambda'_i = 8$ „aufsummiert“ werden:

$$\begin{aligned}
 {}^2\mathcal{B}_{5,8,1} &= \bigoplus_{b=1}^4 \llbracket \mathcal{M}_1 \rrbracket_b = \left((\odot \uplus \odot) \uplus \odot \right) \uplus \odot = \left(\odot \uplus \odot \right) \uplus \odot = \odot \uplus \odot = \odot \uplus \odot, \\
 &\vdots \\
 {}^2\mathcal{B}_{5,8,5} &= \bigoplus_{b=1}^4 \llbracket \mathcal{M}_5 \rrbracket_b = \left((\odot \uplus \odot) \uplus \odot \right) \uplus \odot = \left(\odot \uplus \odot \right) \uplus \odot = \odot \uplus \odot = \odot \uplus \odot, \\
 &\vdots \\
 {}^2\mathcal{B}_{5,8,12} &= \bigoplus_{b=1}^4 \llbracket \mathcal{M}_{12} \rrbracket_b = \left((\odot \uplus \odot) \uplus \odot \right) \uplus \odot = \left(\odot \uplus \odot \right) \uplus \odot = \odot \uplus \odot = \odot \uplus \odot.
 \end{aligned}$$

Die zweite Hälfte der insgesamt $24 = {}^2\mu_{5,8}$ (Programmschritt 7.4.3) zweifarbigen Butcherbäume fünfter Ordnung und achter Struktur entsteht im erneuten Durchlauf der Schleife 7.4 aus den mit $\tau = 2$ berechneten intermediären Produkten

$$\llbracket {}^2\mathcal{B}_{1,1,2} \times \mathcal{Y}_1 \rrbracket_1 = \mathcal{M}_1, \dots, \llbracket {}^2\mathcal{B}_{1,1,2} \times \mathcal{Y}_{12} \rrbracket_1 = \mathcal{M}_{12};$$

zusammengefaßt sind diese:

$${}^2\mathcal{B}_{5,8} = \left\{ \begin{array}{l} \odot \uplus \odot, \\ \odot \uplus \odot, \odot \uplus \odot \end{array} \right\}.$$

Man beachte, daß der geordnete färbungsbezogene Baumexponent, eingeführt in Abschnitt 2.2.6, das Entstehen von Isochromismen verhindert, im hiesigen Beispiel insgesamt acht, etwa jenem zu ${}^2\mathcal{B}_{5,8,5}$:

$$\left((\odot \uplus \odot) \uplus \odot \right) \uplus \odot = \left(\odot \uplus \odot \right) \uplus \odot = \odot \uplus \odot = \odot \uplus \odot \cong \odot \uplus \odot.$$

Aus entwicklungsgeschichtlichen Gründen⁴⁶ werden in der technischen Umsetzung des baumerzeugenden Algorithmus' gemäß Abschnitt 2.2.3 Zeichenketten mit Kompression gleicher Äste verwendet. Dabei wird auf die mit $\Sigma_X = \{u \leftarrow \blacksquare, d \leftarrow \blacksquare\}$ folgendermaßen kodierten Ausgangsbäume zurückgegriffen:

$$\begin{aligned}
 {}^2\mathcal{B}_{1,1,1} &= 1u() \quad , \quad {}^2\mathcal{B}_{1,1,2} = 1d() \quad , \\
 {}^2\mathcal{B}_{2,1,1} &= 1u(1u()) \quad , \quad {}^2\mathcal{B}_{2,1,2} = 1u(1d()) \quad , \\
 {}^2\mathcal{B}_{2,1,3} &= 1d(1u()) \quad , \quad {}^2\mathcal{B}_{2,1,4} = 1d(1d()) \quad .
 \end{aligned}$$

Wie bereits in Abschnitt 3.1.2 bei der beispielhaften Erzeugung ungefärbter Butcherbäume erläutert, müssen zur Ausführung von Programmschritt 7.4.2 die Wurzelmultiplizitäten der „Summanden“ entfernt und vor der Konkatenation der neuen Äste, beginnend an

⁴⁶Siehe Fußnote 33

der öffnenden Klammer der neuen Wurzel, deren Mehrzähligkeiten eingefügt werden⁴⁷. Somit erhalten die Baumadditionen folgende Form:

$$\begin{aligned}
 {}^2\mathcal{B}_{5,8,1} &= 1 \circ u(\circ 2 \circ u(\circ 1 \circ u(1u(\circ)) \circ)) = 1u(2u(\circ)1u(1u(\circ))), \\
 &\vdots \\
 {}^2\mathcal{B}_{5,8,5} &= 1 \circ u(\circ 1 \circ u(\circ 1 \circ d(\circ) \circ 1 \circ u(1u(\circ)) \circ)) = 1u(1u(\circ)1d(\circ)1u(1u(\circ))), \\
 &\vdots \\
 {}^2\mathcal{B}_{5,8,24} &= 1 \circ d(\circ 2 \circ d(\circ 1 \circ d(1d(\circ)) \circ)) = 1d(2d(\circ)1d(1d(\circ))).
 \end{aligned}$$

3.3. Reihenentwicklung des effektiven Hamiltonoperators. Die iterative Prozedur zur Entwicklung der die Elektronentransferprozesse bewirkenden „Operationen“ definierter Störungsreihenordnung ist der zweite wesentliche Schritt im Gesamtverfahren. Voraus geht diesem die Erzeugung aller elektronenbesetzten Bethegitterausschnitte, vergleiche Abschnitt 3.2 und Element 2 in Flußdiagramm 3.1; nachgeschaltet ist der „Hüpfsequenzgenerator“, Abschnitt 3.4.

Gleichwohl wir bei der Probeanwendung auf die Grundzustandsenergie bei Gitterhalbfüllung die Vorgehensweise deutlich vereinfachen können, siehe Abschnitt 3.3.2, sind die meisten Zwischenschritte bereits auf die Behandlung allgemeinerer Probleme (Greensche Funktion) ausgelegt. Die technische Umsetzung beinhaltet in dieser Neuimplementierung teilweise parallelisierte Verarbeitung der Daten; weitere Informationen finden sich im Abschnitt 5.1.

3.3.1. Vorbemerkungen. Da die Entwicklung der Terme des Hamiltonoperators gegebener Ordnung ein rein kombinatorisches Problem ist, lassen sich für die Anzahl der Komponenten zum Teil bekannte, zum Teil aus bekannten abgeleitete zahlentheoretische Folgen angeben. Diese sind speziell bei halber Bandfüllung mit $i \in \mathbb{N}$:

m_i : i te Motzkinzahl [24:A001006], für welche mit $m_0 = m_1 = 1$ folgende Rekursionsformel gilt:

$$m_i := m_{i-1} + \sum_{j=0}^{i-2} m_j m_{i-j-2}. \quad (3.20)$$

α_i : i tes Element der Sequenz A005773 aus der OEIS-Datenbank, Referenz [24]. Es gilt mit $\alpha_0 = 1$ die folgende Rekursionsformel:

$$\alpha_i := m_i + \sum_{j=0}^{i-1} \alpha_j m_{i-j-1}. \quad (3.21)$$

⁴⁷Siehe Fußnote 30

\mathfrak{b}_i : i tes Element der Sequenz A002426 aus Referenz [24]. Es gilt mit $\mathfrak{b}_0 = \mathfrak{b}_1 = 1$ die folgende Rekursionsformel:

$$\mathfrak{b}_i := \frac{1}{i} \left((2i-1)\mathfrak{b}_{i-1} + (3i-3)\mathfrak{b}_{i-2} \right). \quad (3.22)$$

\mathfrak{q}_i : Gesamtanzahl⁴⁸ der möglichen $\mathcal{S}^{(i)}$ (siehe unten) für welche die folgende Formel mit $\mathfrak{q}_0 = 1$ gilt [24:A025565]:

$$\mathfrak{q}_i := 2\mathfrak{a}_{i-1}. \quad (3.23)$$

\mathfrak{u}_i : Anzahl der möglichen $\mathcal{S}^{(i)}$ mit $\kappa_1 = 0 \wedge \kappa_{i+1} > 0 \vee \kappa_1 > 0 \wedge \kappa_{i+1} = 0$ sowie, bei Halbfüllung, in ungeraden Ordnungen Gesamtanzahl unterschiedlicher $\mathcal{T}^{(i)}$ und $\mathcal{D}^{(i)}$ (siehe unten), gemäß folgender Formel mit $\mathfrak{u}_0 = 0$:

$$\mathfrak{u}_i := \mathfrak{b}_{i-1} \quad (3.24)$$

\mathfrak{v}_i : Anzahl der möglichen $\mathcal{S}^{(i)}$ mit $\kappa_1 = 0 \wedge \kappa_{i+1} = 0$ (siehe unten) gemäß folgender Formel mit $\mathfrak{v}_0 = 1$:

$$\mathfrak{v}_i := \frac{1}{2}(\mathfrak{q}_i - 2\mathfrak{u}_i). \quad (3.25)$$

\mathfrak{w}_i : Anzahl der möglichen $\mathcal{S}^{(i)}$ mit $\kappa_1 > 0 \wedge \kappa_{i+1} > 0$ (siehe unten) gemäß folgender Formel mit $\mathfrak{w}_0 = 0$:

$$\mathfrak{w}_i := \mathfrak{v}_i \quad \text{für } i > 0. \quad (3.26)$$

Im besonderen gilt $\mathfrak{q}_i = 2\mathfrak{u}_i + \mathfrak{v}_i + \mathfrak{w}_i$. Tabelle 3.2 zeigt die Glieder sämtlicher Folgen für $0 \leq i \leq 11$.

Um den im Weiteren vorgestellten Algorithmus kompakter formulieren zu können, schreiben wir die sogenannten \mathcal{S} -Faktoren sowie die \mathcal{T} - und die \mathcal{D} -Produkte⁴⁹ aus Abschnitt 1.4 um, und führen dabei mit der Störungsrechnungsordnung n (ungerade) ergänzende Symbole ein:

$\mathcal{S}^{(i)}$, $i = 0, 1, \dots, n$: Eines der \mathfrak{q}_i $\hat{\mathcal{S}}$ -Exponenten-Tupel $(\kappa_j \mid j = 1, 2, \dots, i+1)$, für welches Gleichung (1.16) gemäß $\sum_{j=1}^{i+1} \kappa_j = i$ und $\kappa_j \geq 0$ und durch halbe Bandfüllung nicht $\kappa_j = \kappa_{j+1} = 0$ gilt, im Folgenden \mathcal{S} -Elementarfaktor genannt. Summation über alle möglichen $\hat{\mathcal{S}}$ -Exponenten-Kombinationen liefert den Zusammenhang

$$\sum_{\substack{\kappa_1 + \kappa_2 + \dots + \kappa_{i+1} = i \\ \kappa_j \geq 0 \wedge \neg(\kappa_j = \kappa_{j+1} = 0)}} \hat{\mathcal{S}}^{\kappa_1} \hat{\mathcal{T}}^{\kappa_2} \hat{\mathcal{T}} \dots \hat{\mathcal{T}}^{\kappa_{i+1}} := \sum_{r=1}^{\mathfrak{q}_i} \mathcal{S}_r^{(i)}. \quad (3.27)$$

⁴⁸Es gilt bereits der Ausschluß von $\kappa_j = 0 \wedge \kappa_{j+1} = 0$ für $j = 1, 2, \dots, i$ bei Halbfüllung. Ohne diese Einschränkung entspricht die Anzahl der $\mathcal{S}^{(i)}$ jener der zentralen Binomialkoeffizienten, siehe beispielsweise Referenz [24:A000984].

⁴⁹Nach den Gleichungen (1.17) und (1.18) können die \mathcal{D} -Produkte durch einen Vorfaktor im \mathcal{T} -Produkt berücksichtigt werden; dennoch werden erstgenannte und zugehörige Mengen in den folgenden Ausführungen mit Hinblick auf eine verallgemeinerte Anwendung der Algorithmen behandelt.

i	0	1	2	3	4	5	6	7	8	9	10	11	Sequenz
m_i	1	1	2	4	9	21	51	127	323	835	2188	5798	A001006
α_i	1	2	5	13	35	96	267	750	2123	6046	17303	49721	A005773
\mathfrak{b}_i	1	1	3	7	19	51	141	393	1107	3139	8953	25653	A002426
\mathfrak{g}_i	1	2	4	10	26	70	192	534	1500	4246	12092	34606	A025565
u_i	0	1	1	3	7	19	51	141	393	1107	3139	8953	Gl. (3.24)
v_i	1	0	1	2	6	16	45	126	357	1016	2907	8350	A005717
w_i	0	0	1	2	6	16	45	126	357	1016	2907	8350	Gl. (3.26)

Tabelle 3.2. Hilfstabelle zur Erzeugung der \mathcal{T} - und \mathcal{D} -Produkte für $i = 0, 1, \dots, 11$. In der letzten Spalte findet sich die Eintragsbezeichnung der Sequenz in der OEIS-Datenbank, Referenz [24]. Man beachte, daß $u_i = \mathfrak{b}_{i-1}$ mit $u_0 = 0$ und $w_i = v_i$ mit $w_0 = 0$ gilt.

Ferner führen wir eine abkürzende Schreibweise der beteiligten \hat{S} -Exponenten ein,

$$\begin{aligned}
 \mathfrak{g}^{(i)} &= \hat{S}^{\kappa_1} \hat{T} \hat{S}^{\kappa_2} \hat{T} \dots \hat{T} \hat{S}^{\kappa_{i+1}} \leftarrow (\kappa_1, \kappa_2, \dots, \kappa_{i+1}) \\
 &\leftarrow \boxed{\kappa_1 \ \kappa_2 \ \dots \ \kappa_{i+1}}, \quad (3.28)
 \end{aligned}$$

und verstehen das Symbol \circ als ein „Zusammenkleben“ beider, formal die Konkatenation zweier \mathcal{S} -Elementarfaktoren,

$$\begin{aligned}
 \mathfrak{g}^{(u)} \hat{\mathfrak{g}} \mathfrak{g}^{(v)} &:= \boxed{\alpha_1 \ \alpha_2 \ \dots \ \alpha_{u+1}} \circ \hat{\mathfrak{g}} \boxed{\beta_1 \ \beta_2 \ \dots \ \beta_{v+1}} \\
 &= \boxed{\alpha_1 \ \alpha_2 \ \dots \ \alpha_{u+1} \ \beta_1 \ \beta_2 \ \dots \ \beta_{v+1}} \quad (3.29)
 \end{aligned}$$

mit $u, v \in \mathbb{N}$ sowie (beispielhaft) dem Einsoperator $\hat{\mathfrak{g}}$, welche in der Sprache der zugrundeliegenden Operatorabfolgen als Produktbildung aufgefaßt werden kann. Wie mit den Grenzelementen an der Klebestelle, hier etwa α_{u+1} und β_1 , und gegebenenfalls Vorfaktoren zu verfahren ist, hängt vom eingeschobenen Operator⁵⁰ ab und wird beim jeweiligen Produkt erläutert.

⁵⁰ Genaugenommen kann in diesem Zusammenhang das Konkatenationssymbol nicht alleine auftreten und bildet erst mit dem folgenden Operator eine „funktionstüchtige“ Einheit, $(\circ \hat{X})$, welche sowohl auf den rechts- wie auch auf den linksstehenden Operanden wirkt, siehe Gleichungen (3.34), (3.38) und (3.43). Die dieses verdeutlichendere, indes umständlicher erscheinende Schreibweise ist

$$(\circ \hat{X}) \left(\boxed{\alpha_1 \ \alpha_2 \ \dots \ \alpha_{u+1}} \cdot \boxed{\beta_1 \ \beta_2 \ \dots \ \beta_{v+1}} \right).$$

$\check{\mathfrak{S}}^{(i)}$: Menge aller Elementarfaktoren $\mathfrak{S}^{(i)}$.

$$\check{\mathfrak{S}}^{(i)} = \bigcup_{j=1}^{\mathfrak{q}_i} \mathfrak{S}_j^{(i)} = \check{\mathfrak{S}}_{\text{EE}}^{(i)} \cup \check{\mathfrak{S}}_{\text{EG}}^{(i)} \cup \check{\mathfrak{S}}_{\text{GE}}^{(i)} \cup \check{\mathfrak{S}}_{\text{GG}}^{(i)}, \quad (3.30)$$

bestehend aus den vier Teilmengen

$$\check{\mathfrak{S}}_{\text{EE}}^{(i)} \subseteq \check{\mathfrak{S}}^{(i)}, \text{ beginnend mit } \kappa_1 = 0 \text{ und endend mit } \kappa_{i+1} = 0, |\check{\mathfrak{S}}_{\text{EE}}^{(i)}| = \mathfrak{v}_i, \quad (3.31-1)$$

$$\check{\mathfrak{S}}_{\text{EG}}^{(i)} \subset \check{\mathfrak{S}}^{(i)}, \text{ beginnend mit } \kappa_1 = 0 \text{ und endend mit } \kappa_{i+1} > 0, |\check{\mathfrak{S}}_{\text{EG}}^{(i)}| = \mathfrak{u}_i, \quad (3.31-2)$$

$$\check{\mathfrak{S}}_{\text{GE}}^{(i)} \subset \check{\mathfrak{S}}^{(i)}, \text{ beginnend mit } \kappa_1 > 0 \text{ und endend mit } \kappa_{i+1} = 0, |\check{\mathfrak{S}}_{\text{GE}}^{(i)}| = \mathfrak{u}_i, \quad (3.31-3)$$

$$\check{\mathfrak{S}}_{\text{GG}}^{(i)} \subset \check{\mathfrak{S}}^{(i)}, \text{ beginnend mit } \kappa_1 > 0 \text{ und endend mit } \kappa_{i+1} > 0, |\check{\mathfrak{S}}_{\text{GG}}^{(i)}| = \mathfrak{w}_i. \quad (3.31-4)$$

Es gilt $\mathfrak{S}_{\text{EE},j}^{(i)} \in \check{\mathfrak{S}}_{\text{EE}}^{(i)}$, $\mathfrak{S}_{\text{EG},j}^{(i)} \in \check{\mathfrak{S}}_{\text{EG}}^{(i)}$, $\mathfrak{S}_{\text{GE},j}^{(i)} \in \check{\mathfrak{S}}_{\text{GE}}^{(i)}$ und $\mathfrak{S}_{\text{GG},j}^{(i)} \in \check{\mathfrak{S}}_{\text{GG}}^{(i)}$.

$\hat{\mathfrak{S}}^{(i)}$: Ein \mathfrak{S} -Faktor im Sinne der Definition der \mathcal{T} - und \mathcal{D} -Produkte, siehe Gleichungen (3.37) und (3.41).

$$\hat{\mathfrak{S}}^{(i)} := - \sum_{j=1}^{\mathfrak{q}_i} \llbracket \check{\mathfrak{S}}^{(i)} \rrbracket_j = - \sum_{j=1}^{\mathfrak{q}_i} \mathfrak{S}_j^{(i)} \quad (3.32)$$

$\mathcal{P}^{(i)}$, $i = 1, 3, \dots, n$: Ein Prozeß im Sinne von Abschnitt 1.5.1, beispielsweise der Gleichung (1.22) und folgende; auch *Elementarprozeß* genannt. $\mathcal{P}^{(i)}$ formulieren wir als Paar $(e, \mathfrak{p}^{(i)})$, worin $e \in \mathbb{N} \setminus \{0\}$ der *Prozeßkoeffizient* und $\mathfrak{p}^{(i)}$ das Tupel $(\kappa_j \mid j = 1, 2, \dots, i+2)$ der $\hat{\mathcal{S}}$ -Exponenten ist. Formal handelt sich bei den im Weiteren eingeführten *Elementarprodukten* $\mathcal{T}^{(i)}$, $\mathcal{D}^{(i)}$, $\mathcal{Z}^{(i)}$ und $\mathcal{N}^{(i)}$ ebenfalls um Prozesse, für deren Koeffizienten im Allgemeinen jedoch $e \in \mathbb{Z}$ gilt.

Zwei Elementarprozesse $\mathcal{P}_1^{(i)}$ und $\mathcal{P}_2^{(i)}$ mit beliebigen Prozeßkoeffizienten e_1 und e_2 lassen sich zu einem Prozeß *reduzieren* (zusammenfassen),

$$\mathcal{P}_1^{(i)} \triangleleft \mathcal{P}_2^{(i)} = \mathcal{P}^{(i)} := (e_1 + e_2, \mathfrak{p}_1^{(i)}) \quad (3.33)$$

wenn für deren $\hat{\mathcal{S}}$ -Exponenten-Tupel $\mathfrak{p}_1^{(i)} = (\alpha_1, \alpha_2, \dots, \alpha_{i+1}, \alpha_{i+2})$ und $\mathfrak{p}_2^{(i)} = (\beta_1, \beta_2, \dots, \beta_{i+1}, \beta_{i+2})$ gilt: $\alpha_j = \beta_j$ für $1 \leq j \leq i+2$. Die *Reduktion* ist folglich kommutativ, $\mathcal{P}_1^{(i)} \triangleleft \mathcal{P}_2^{(i)} = \mathcal{P}_2^{(i)} \triangleleft \mathcal{P}_1^{(i)}$. Anstatt des Reduktionssymbols \triangleleft wird für das entsprechende Zusammenfassen von Mengen aus „prozeßwertigen“ Elementen⁵¹, beispielsweise M_1 und M_2 , folgende Schreibweise verwendet:

$M_1 \sqsupset M_2$, M_1 „absorbiert“ M_2 vollständig, vorausgesetzt $M_2 \subseteq M_1$ ⁵².

$\mathcal{P}^{*(i)} = \text{adj}(\mathcal{P}^{(i)})$: Wir nennen zwei Elementarprozesse $\mathcal{P}_1^{(i)}$ und $\mathcal{P}_2^{(i)}$ mit beliebigen Prozeßkoeffizienten e_1 und e_2 zueinander *adjungiert* und schreiben $\mathcal{P}_1^{(i)} = \mathcal{P}_2^{*(i)}$,

⁵¹Diese entsprechen formal Multimengen, welche wir als Mengen von Paaren definiert haben, vergleiche Abschnitt 2.1.

⁵²Es handelt sich unter dieser Voraussetzung um einen Sonderfall der Vereinigung von Multimengen, $M_1 \sqcup M_2$.

wenn für deren \hat{S} -Exponenten-Tupel $\mathfrak{p}_1^{(i)} = (\alpha_1, \alpha_2, \dots, \alpha_{i+1}, \alpha_{i+2})$ und $\mathfrak{p}_2^{(i)} = (\beta_1, \beta_2, \dots, \beta_{i+1}, \beta_{i+2})$ gilt: $\alpha_j = \beta_{i+3-j}$ für $1 \leq j \leq i+2$. Adjungierte Prozesse, die – bildlich gesprochen – demzufolge zueinander spiegelsymmetrisch sind, lassen sich gemäß Gleichung (3.33) reduzieren; wie in Abschnitt 3.3.4 dargelegt wird, ermöglicht dieses eine weitere Verringerung des Rechenauswandes.

Die den Gleichungen (I.12) sowie (I.18) entsprechende Entwicklung des Zählers in einer $1/U$ -Reihe erfolgt in direkter Umsetzung der in Abschnitt I.4.1 beschriebenen Zerlegung. Die darin enthaltenen Komponenten setzen sich folgendermaßen zusammen:

$\mathcal{T}^{(i)}$, $i = 1, 3, \dots, n$: Ein \mathcal{T} -Elementarprodukt bei beliebiger Füllung; $i = u + v$ und $u, v \geq 0$.

$$\begin{aligned} \mathcal{T}^{(i)} &= \mathfrak{S}^{(u)} \hat{T} \mathfrak{S}^{(v)} := \boxed{\alpha_1 \ \alpha_2 \ \dots \ \alpha_{u+1}} \circ \hat{T} \boxed{\beta_1 \ \beta_2 \ \dots \ \beta_{v+1}} \\ &= +1 \boxed{\alpha_1 \ \alpha_2 \ \dots \ \alpha_{u+1} \ \beta_1 \ \beta_2 \ \dots \ \beta_{v+1}} \\ &= \left(+1, (\kappa_1, \kappa_2, \dots, \kappa_{(u+v+2)=i+2}) \right) \end{aligned} \quad (3.34)$$

$\check{\mathcal{T}}^{(i)}$: Multimenge aller (mit Vorfaktoren behafteten) Elementarprodukte $\mathcal{T}^{(i)}$, speziell bei halber Füllung mit der Zusammensetzung

$$\check{\mathcal{T}}^{(i)} = \bigsqcup_{j=1}^{u_i} \mathcal{T}_j^{(i)} = \check{\mathcal{T}}_{\text{EGEE}}^{(i)} \sqcup \check{\mathcal{T}}_{\text{EEGE}}^{(i)} \sqcup \check{\mathcal{T}}_{\text{EGGE}}^{(i)}, \quad (3.35)$$

bestehend aus den drei Teilmengen

$$\check{\mathcal{T}}^{(i)} \supseteq \check{\mathcal{T}}_{\text{EGEE}}^{(i)} = \bigsqcup_{r=0}^i \check{\mathfrak{S}}_{\text{EG}}^{(i-r)} \times \hat{T} \check{\mathfrak{S}}_{\text{EE}}^{(r)} \quad (3.36-1)$$

$$\begin{aligned} &= \left\{ \mathfrak{S}_{\text{EG},s}^{(i)} \hat{T} \mathfrak{S}_{\text{EE},t}^{(0)} \mid s = 1, \dots, u_i, t = 1, \dots, v_0 \right\} \sqcup \dots \\ &\quad \dots \sqcup \left\{ \mathfrak{S}_{\text{EG},s}^{(0)} \hat{T} \mathfrak{S}_{\text{EE},t}^{(i)} \mid s = 1, \dots, u_0, t = 1, \dots, v_i \right\} \\ &= 2 \bigsqcup_{r=0}^{(i-1)/2} \check{\mathfrak{S}}_{\text{EG}}^{(i-r)} \times \hat{T} \check{\mathfrak{S}}_{\text{EE}}^{(r)} \quad \text{mit} \quad |\check{\mathcal{T}}_{\text{EGEE}}^{(i)}| = 2 \sum_{r=0}^{(i-1)/2} u_{i-r} v_r, \end{aligned} \quad (3.36-2)$$

$$\check{\mathcal{T}}^{(i)} \supset \check{\mathcal{T}}_{\text{EEGE}}^{(i)} = 2 \bigsqcup_{r=0}^{(i-1)/2} \check{\mathfrak{S}}_{\text{EE}}^{(i-r)} \times \hat{T} \check{\mathfrak{S}}_{\text{GE}}^{(r)} \quad \text{mit} \quad |\check{\mathcal{T}}_{\text{EEGE}}^{(i)}| = 2 \sum_{r=0}^{(i-1)/2} v_{i-r} u_r, \quad (3.36-3)$$

$$\check{\mathcal{T}}^{(i)} \supset \check{\mathcal{T}}_{\text{EGGE}}^{(i)} = 2 \bigsqcup_{r=1}^{(i-1)/2} \check{\mathfrak{S}}_{\text{EG}}^{(i-r)} \times \hat{T} \check{\mathfrak{S}}_{\text{GE}}^{(r)} \quad \text{mit} \quad |\check{\mathcal{T}}_{\text{EGGE}}^{(i)}| = 2 \sum_{r=1}^{(i-1)/2} u_{i-r} u_r. \quad (3.36-4)$$

$\check{\mathcal{T}}_{\text{red.}}^{(i)}$: Menge aller bis auf einen repräsentativen Vertreter reduzierten und entsprechend mit Vorfaktoren behafteten Elementarprodukte $\mathcal{T}^{(i)}$; es gilt $|\check{\mathcal{T}}_{\text{red.}}^{(i)}| = u_i \ll |\check{\mathcal{T}}_{\text{EGEE}}^{(i)}| + |\check{\mathcal{T}}_{\text{EEGE}}^{(i)}| + |\check{\mathcal{T}}_{\text{EGGE}}^{(i)}|$.

$\hat{\mathcal{T}}^{(i)}$: Menge aller \mathcal{T} -Produkte, siehe Gleichung (I.13).

$$\sum_{r=0}^i \langle \phi_0 | \hat{\mathcal{S}}^{(i-r)} \hat{T} \hat{\mathcal{S}}^{(r)} | \phi_0 \rangle = \sum_{j=1}^{u_i} \langle \phi_0 | \hat{\mathcal{T}}_j^{(i)} | \phi_0 \rangle \quad (3.37)$$

$\mathfrak{D}^{(i)}$, $i = 1, 3, \dots, n$: Ein \mathfrak{D} -Elementarprodukt bei beliebiger Füllung; $i = u + v$ und $u, v > 0$.

$$\begin{aligned} \mathfrak{D}^{(i)} &= \mathfrak{S}^{(u+1)} \hat{D} \mathfrak{S}^{(v)} := \boxed{\alpha_1 \ \alpha_2 \ \dots \ \alpha_{u+2}} \circ \hat{D} \boxed{\beta_1 \ \beta_2 \ \dots \ \beta_{v+1}} \\ &= -1 \boxed{\alpha_1 \ \alpha_2 \ \dots \ (\alpha_{u+2} + \beta_1 - 1) \ \beta_2 \ \dots \ \beta_{v+1}} \\ &= (-1, (\kappa_1, \kappa_2, \dots, \kappa_{(u+v+2=i+2)})) \end{aligned} \quad (3.38)$$

$\check{\mathfrak{D}}^{(i)}$: Multimenge aller (mit Vorfaktoren behafteten) Elementarprodukte $\mathfrak{D}^{(i)}$, speziell bei halber Füllung mit der Zusammensetzung

$$\check{\mathfrak{D}}^{(i)} = \bigsqcup_{j=1}^{u_i} \mathfrak{D}_j^{(i)} = \check{\mathfrak{D}}_{\text{EGGE}}^{(i)}, \quad (3.39)$$

worin

$$\check{\mathfrak{D}}_{\text{EGGE}}^{(i)} = \bigsqcup_{r=1}^i \check{\mathfrak{S}}_{\text{EG}}^{(i+1-r)} \times \hat{D} \check{\mathfrak{S}}_{\text{GE}}^{(r)} \quad (3.40-1)$$

$$= \left(2 \bigsqcup_{r=1}^{(i-1)/2} \check{\mathfrak{S}}_{\text{EG}}^{(i+1-r)} \times \hat{D} \check{\mathfrak{S}}_{\text{GE}}^{(r)} \right) \sqcup \left(\check{\mathfrak{S}}_{\text{EG}}^{((i+1)/2)} \times \hat{D} \check{\mathfrak{S}}_{\text{GE}}^{((i+1)/2)} \right) \quad (3.40-2)$$

$$\text{mit } |\check{\mathfrak{D}}_{\text{EGGE}}^{(i)}| = \sum_{r=1}^i u_{i+1-r} u_r = 2 \sum_{r=1}^{(i-1)/2} u_{i+1-r} u_r + u_{(i+1)/2}^2.$$

$\check{\mathfrak{D}}_{\text{red}}^{(i)}$: Menge aller bis auf einen repräsentativen Vertreter reduzierten und entsprechend mit Vorfaktoren behafteten Elementarprodukte $\mathfrak{D}^{(i)}$; es gilt $|\check{\mathfrak{D}}_{\text{red}}^{(i)}| = u_i \ll |\check{\mathfrak{D}}_{\text{EGGE}}^{(i)}|$.

$\hat{\mathfrak{D}}^{(i)}$: Menge aller \mathfrak{D} -Produkte, siehe Gleichung (I.14).

$$\sum_{r=1}^i \langle \phi_0 | \hat{\mathcal{S}}^{(i+1-r)} \hat{D} \hat{\mathcal{S}}^{(r)} | \phi_0 \rangle = \sum_{j=1}^{u_i} \langle \phi_0 | \hat{\mathfrak{D}}_j^{(i)} | \phi_0 \rangle \quad (3.41)$$

$\check{\mathcal{Z}}_{\text{red}}^{(i)}$, „Zähler“: Gesamtmenge $\check{\mathcal{T}}_{\text{red}}^{(i)} \sqsupset \check{\mathfrak{D}}_{\text{red}}^{(i)}$. Da die symmetrische Differenz beider Mengen (ohne Berücksichtigung der Vorfaktoren) $\check{\mathcal{T}}_{\text{red}}^{(i)} \triangle \check{\mathfrak{D}}_{\text{red}}^{(i)} = \emptyset$ ist, können alle Elementarprodukte $\mathfrak{D}^{(i)}$ in den Vorfaktoren der $\mathcal{T}^{(i)}$ berücksichtigt werden, wodurch $|\check{\mathcal{Z}}_{\text{red}}^{(i)}| = u_i$ erhalten bleibt.

Die Entwicklung des Nenners in einer $1/U$ -Reihe gemäß der Gleichung (1.12) und deren Halbfüllungsgrenzfall (1.18) ist durch

$$\begin{aligned} \frac{1}{1 + \sum_{j=1}^{\infty} \frac{\Omega_j}{U^{2j}}} &= 1 - \frac{\Omega_1}{U^2} + \frac{\Omega_1^2 - \Omega_2}{U^4} + \frac{-\Omega_1^3 + 2\Omega_1\Omega_2 - \Omega_3}{U^6} + \dots \\ &= 1 - \sum_{j=1}^{\infty} \left(\frac{1}{U^{2j}} \sum_{q=1}^j \sum_{\substack{g_1+g_2+\dots+g_q=j \\ g_r>0, 1\leq r\leq q}} (-1)^q \Omega_{g_1} \Omega_{g_2} \dots \Omega_{g_q} \right) \end{aligned} \quad (3.42)$$

gegeben. Die nun im Zähler stehenden Polynome lassen sich in unseren Formalismus durch $\Omega_{g_r} \mapsto \mathcal{S}^{(v_r)}$ mit $v_r = 2g_r$ wie folgt übertragen:

$\mathcal{N}^{(i)}$, $i = 3, 5, \dots, n$: Ein \mathcal{N} -Elementarprodukt bei beliebiger Füllung; $\mathcal{Z}^{(i)} \in \check{\mathcal{Z}}_{\text{red.}}^{(i)}$,
 $i = u + v_1 + v_2 + \dots + v_q$, $u = 1, 3, \dots, i - 2$ und $v_j > 0 \forall j$ sowie $q = 1, 2, \dots, (i - u)/2$ (Wertebereich der q).

$$\begin{aligned} \mathcal{N}^{(i)} &= \mathcal{Z}^{(u)} \prod_{j=1}^q \hat{\mathcal{N}} \mathcal{S}^{(v_j)} = \mathcal{Z}^{(u)} \hat{\mathcal{N}} \mathcal{S}^{(v_1)} \hat{\mathcal{N}} \mathcal{S}^{(v_2)} \dots \hat{\mathcal{N}} \mathcal{S}^{(v_q)} \\ &:= \boxed{\alpha_1 \ \alpha_2 \ \dots \ \alpha_{u+2}} \circ \hat{\mathcal{N}} \underbrace{\boxed{\beta_{1,1} \ \beta_{1,2} \ \dots \ \beta_{1,v_1+1}}}_{\text{gemäß } \Omega_{g_1}, \text{ Gl. (3.42)}} \\ &\quad \circ \hat{\mathcal{N}} \underbrace{\boxed{\beta_{2,1} \ \beta_{2,2} \ \dots \ \beta_{2,v_2+1}}}_{\Omega_{g_2}} \dots \circ \hat{\mathcal{N}} \underbrace{\boxed{\beta_{q,1} \ \beta_{q,2} \ \dots \ \beta_{q,v_q+1}}}_{\Omega_{g_q}} \\ &= (-1)^q \boxed{\alpha_1 \ \dots \ \alpha_{u+1} \ \beta_{1,1} \ \dots \ \beta_{1,v_1} \ \beta_{2,1} \ \dots \ \beta_{2,v_2} \ \dots \ \beta_{q,1} \ \dots \ \beta_{q,v_q+1}} \\ &= ((-1)^q, (\kappa_1, \kappa_2, \dots, \kappa_{(u+v_1+v_2+\dots+v_q+2)=i+2})) \end{aligned} \quad (3.43)$$

Der hier notierte formale Operator $\hat{\mathcal{N}}$ bewirkt in Einheit mit dem Konkatenationszeichen (vergleiche Fußnote 50) den Wegfall des letzten $\hat{\mathcal{S}}$ -Exponenten, einer Null, im linken Operanden; jedes Auftreten ändert zudem das Vorzeichen des Produkts.

$\check{\mathcal{N}}^{(i)}$: Multimenge aller (mit Vorfaktoren behafteten) Elementarprodukte $\mathcal{N}^{(i)}$. Speziell bei halber Füllung gilt für deren Zusammensetzung

$$\check{\mathcal{N}}^{(i)} = \bigsqcup_{\substack{u+v_1+v_2+\dots+v_q=i \\ u=1,3,\dots,i-2; v_j>0, 1\leq j\leq q}} \check{\mathcal{Z}}_{\text{red.}}^{(u)} \times \hat{\mathcal{N}} \check{\mathcal{S}}_{\text{EE}}^{(v_1)} \dots \times \hat{\mathcal{N}} \check{\mathcal{S}}_{\text{EE}}^{(v_q)}. \quad (3.44)$$

$\check{\mathcal{N}}_{\text{red.}}^{(i)}$, „Nenner“: Menge aller bis auf einen repräsentativen Vertreter reduzierten und entsprechend mit Vorfaktoren behafteten Elementarprodukte $\mathcal{N}^{(i)}$.

$\check{\mathcal{P}}_{\text{red.}}^{(i)}$, „Prozesse“: Gesamtmenge $\check{\mathcal{Z}}_{\text{red.}}^{(i)} \sqsupset \check{\mathcal{N}}_{\text{red.}}^{(i)}$; $\check{\mathcal{P}}_{\text{red.}}^{(i)} \ni \mathcal{P}^{(i)}$. Da (ohne Berücksichtigung der Vorfaktoren) $\check{\mathcal{N}}_{\text{red.}}^{(i)} \subset \check{\mathcal{Z}}_{\text{red.}}^{(i)}$ gilt, können alle Elementarprodukte $\mathcal{N}^{(i)}$ in

den Vorfaktoren der $\mathcal{Z}^{(i)}$ berücksichtigt werden, wodurch die Anzahl der unterschiedlichen Prozesse $|\check{\mathcal{P}}_{\text{red}}^{(i)}| = u_i$ erhalten bleibt.

$\check{\mathcal{P}}_{\text{eff}}^{(i)}$: Menge aller Elementarprozesse $\mathcal{P}^{(i)} \in \check{\mathcal{P}}_{\text{red}}^{(i)}$ mit einem Prozeßkoeffizienten $e \neq 0$; $\check{\mathcal{P}}_{\text{eff}}^{(i)} \subseteq \check{\mathcal{P}}_{\text{red}}^{(i)}$.

$\hat{\mathcal{P}}^{(i)}$: Menge aller Elementarprozesse $\mathcal{P}^{(i)} \in \check{\mathcal{P}}_{\text{eff}}^{(i)}$, für welche $\mathfrak{p} \ni \kappa_j > 0$ mit $j = 4, 6, \dots, i-1$ gilt. (Die Forderung $\kappa_2 > 0$ ist bereits erfüllt, da bei halber Bandfüllung \mathfrak{S} -Faktoren mit $\kappa_j = 0 \wedge \kappa_{j+1} = 0$ für $j = 1, 2, \dots, i$ ausgeschlossen sind.)
 $\hat{\mathcal{P}}^{(i)} \subseteq \check{\mathcal{P}}_{\text{eff}}^{(i)}$.

$f \in \mathbb{N}$: Nummer der *Operator*- beziehungsweise *Prozeßklasse*,
 $f = 0, 1, \dots, 2^{(i-1)/2} - 1$

$\hat{\mathcal{P}}_f^{(i)}$: Menge aller Elementarprozesse $\mathcal{P}^{(i)} \in \hat{\mathcal{P}}^{(i)}$ der Operatorklasse f , welche sich aus der Position der verbleibenden Grundzustandsprojektoren $\hat{S}^{\kappa=0}$ in der Operatorsequenz und entsprechend aus der Position der inneren ($\kappa = 0$)-Elemente in $\mathfrak{p}^{(i)}$ (die Außenelemente κ_1 und κ_{i+1} sind, wie zuvor ausgeführt, bei halber Bandfüllung stets null) folgendermaßen berechnen läßt:

$$f(\hat{\mathfrak{p}}^{(i)}) = \sum_{u=3,5,\dots}^i 2^{(u-3)/2} \hat{\mathfrak{p}}_u^{(i)}, \quad \text{worin} \quad (3.45)$$

$$\hat{\mathfrak{p}}^{(i)} = (\hat{\kappa}_v \mid v = 1, 2, \dots, i+2) \quad \text{mit} \quad \hat{\kappa}_v = \begin{cases} 1 & \text{falls } \kappa_v = \llbracket \mathfrak{p}^{(i)} \rrbracket_v = 0 \\ 0 & \text{sonst} \end{cases} \quad (3.46)$$

Es gilt ferner

$$\hat{\mathcal{P}}^{(i)} = \bigsqcup_{g=1}^{2^{(i-1)/2}} \hat{\mathcal{P}}_{g-1}^{(i)} \quad \text{und entsprechend} \quad |\hat{\mathcal{P}}^{(i)}| = \sum_{g=1}^{2^{(i-1)/2}} |\hat{\mathcal{P}}_{g-1}^{(i)}|. \quad (3.47)$$

Man beachte, daß einige Operatorklassen aufgrund $e(\mathcal{P}^{(i)}) = 0$ und somit effektiv $\hat{\mathcal{P}}_f^{(i)} = \emptyset$ „verschwinden“.

3.3.2. Grundsätzliches Vorgehen für halbe Bandfüllung.

- 1 Ausführungsparameter vom aufrufenden Hauptprogramm übernehmen: Störungsordnung n , Anzahl der in den parallelisierten Abschnitten genutzten Prozessoren sowie Steuerparameter für den Umfang an Kontrollausgaben (Speicherbedarf etc.) und Zwischenergebnissen
- 2 Lokale Zeitmessung beginnen
- 3 Tabelle 3.2 mit Hilfsdaten (Indexobergrenzen) aufbauen
- 4 (Elementarfaktoren) Für $i = 0, 1, \dots, n$, siehe Flußdiagrammelement (1) in Abbildung 3.5:

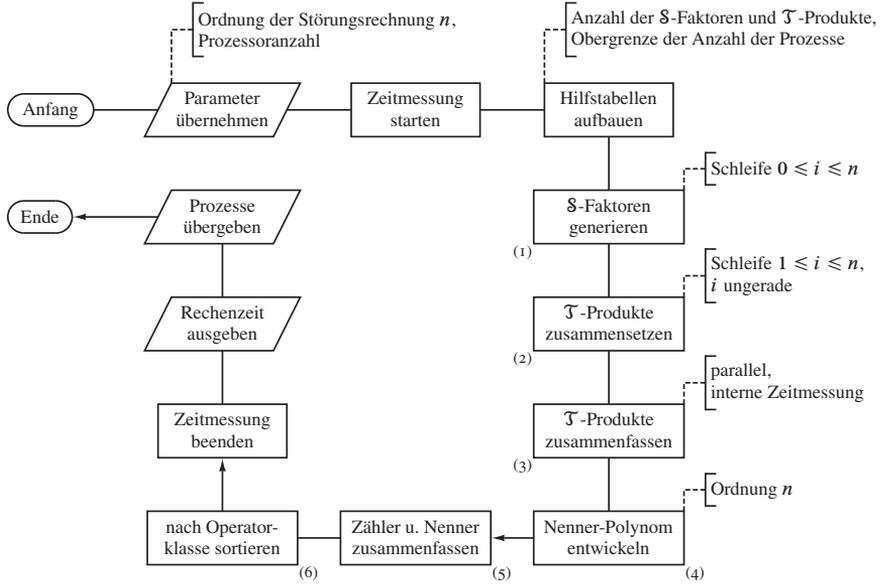


Abbildung 3.5. Ablaufplan des Unterprogramms zur Erzeugung der Prozesse bei halber Bandfüllung, Flußdiagrammelement (2) in Abbildung 3.1.

- 4.1 Partitionierung der Zahl i in q lexikographisch (absteigend) geordnete Summanden κ_a ($a = 1, \dots, q$), formal beginnend mit $q = 1$ (also $\kappa_1 = i$)⁵³, endend mit $q = i$ (folglich $\kappa_1 = \kappa_2 = \dots = \kappa_i = 1$):

$$i = \sum_{a=1}^q \kappa_a \quad \text{mit} \quad \kappa_1 \geq \kappa_2 \geq \dots \geq \kappa_q > 0 \quad (3.48)$$

Das Tupel der Summanden einer Partition wird im Weiteren mit K bezeichnet; es gilt: $|K| = q$.

- 4.2 Anhängen von $\dot{q} = i + 1 - q$ Nullen an K ,

$$\dot{K} = K \circ (0 \mid j = 1, 2, \dots, \dot{q}), \quad (3.49)$$

so daß $|\dot{K}| = i + 1$

⁵³In diesem Falle kann auf die allgemeine Partitionierungsprozedur zurückgegriffen werden, wie diese bereits im Programmschritt 4 in Abschnitt 3.1.1 verwendet wurde. Wie im Weiteren zu sehen, werden jedoch damit auch Partitionen erzeugt, die zu mehr als der Hälfte aus Nullen bestehen, und von welchen zwangsläufig jene Permutationen mit mindestens zwei aufeinanderfolgende Nullen herausfallen: Bei halber Bandfüllung verschwinden Operatortermine der Art $\hat{P}_0 \dots \hat{P}_0 \hat{T} \hat{P}_0 \dots \hat{P}_0 = \hat{S}^0 \dots \hat{S}^{\kappa_g=0} \hat{T} \hat{S}^{\kappa_{g+1}=0} \dots \hat{S}^0$; um den Filteraufwand in Programmschritt 4.4 zu verringern, sollte folglich mit $q = \lceil i/2 \rceil$ begonnen werden.

4.3 Bildung sämtlicher Permutationen \tilde{K} des $(i + 1)$ -Tupels \dot{K} , $|\tilde{K}| = i + 1$, deren Gesamtmenge $\check{K} = \{\tilde{K}_1, \tilde{K}_2, \dots\}$ mit $\tilde{K}_1 = \dot{K}$ sei.

$$\begin{aligned} \tilde{K}_1 &= (\kappa_1, \kappa_2, \dots, \kappa_{g_1}, \kappa_{g_1+1}, \kappa_{g_1+2}, \dots, \kappa_{g_2}, \dots, \kappa_{g_{u-1}+1}, \kappa_{g_{u-1}+2}, \dots, \kappa_{g_u}) \\ &\vdots \\ \tilde{K}_{\check{k}-1} &= (\kappa_{g_{u-1}+1}, \kappa_{g_{u-1}+2}, \dots, \kappa_{g_u}, \dots, \kappa_1, \kappa_2, \dots, \kappa_{g_1}, \kappa_{g_1+1}, \kappa_{g_1+2}, \dots, \kappa_{g_2}) \\ \tilde{K}_{\check{k}} &= (\kappa_{g_{u-1}+1}, \kappa_{g_{u-1}+2}, \dots, \kappa_{g_u}, \dots, \kappa_{g_1+1}, \kappa_{g_1+2}, \dots, \kappa_{g_2}, \kappa_1, \kappa_2, \dots, \kappa_{g_1}) \end{aligned}$$

Zur kontrollweisen Berechnung von $\dot{k} = |\check{K}|$ bedient man sich der Vorgehensweise in Schritt 5 im Erzeugungsalgorithmus der (ungefärbten) Butcherbäume, Abschnitt 3.1.1: Einteilung der κ_a in u Gruppen ϱ_α gleicher Summanden ($\alpha = 1, \dots, u$); κ'_α ist dabei ein „repräsentativer Vertreter der jeweiligen Gruppe“ aus \dot{K} :

$$i = \sum_{a=1}^q \kappa_a = \sum_{\alpha=1}^{u-1} \varrho_\alpha \kappa'_\alpha, \quad \sum_{\alpha=1}^u \varrho_\alpha = i + 1,$$

	n	1	3	5	7	9	11	Sequenz
$ \check{\mathcal{T}}(n) $	nach Schritt 6.2	1	5	40	344	3014	26620	
$ \check{\mathcal{T}}_{\text{red.}}(n) $	nach Schritt 6.3	1	3	19	141	1107	8953	\mathfrak{u}_n
$ \check{\mathcal{D}}(n) $	nach Schritt 6.4	1	7	61	547	4921	44287	
$ \check{\mathcal{D}}_{\text{red.}}(n) $	nach Schritt 6.5	1	3	19	141	1107	8953	\mathfrak{u}_n
$ \check{\mathcal{Z}}(n) $	(entfällt)	2	10	80	688	6028	53240	
$ \check{\mathcal{Z}}_{\text{red.}}(n) $	nach Schritt 6.6	1	3	19	141	1107	8953	\mathfrak{u}_n
$ \check{\mathcal{N}}(n) $	nach Schritt 8.4	0	1	10	98	950	9131	
$ \check{\mathcal{N}}_{\text{red.}}(n) $	nach Schritt 10	0	1	8	68	586	5075	
$ \check{\mathcal{P}}_{\text{red.}}(n) $	nach Schritt 11	1	4	25	168	1163	8176	
$ \hat{\mathcal{P}}(n) $	in Schritt 12	1	2	11	65	386	2413	

Tabelle 3.3. Anzahl der Prozesse nach verschiedenen Programmschritten.

in Tupelschreibweise, $g_\xi = \sum_{\alpha=1}^{\xi} \varrho_\alpha$:

$$\dot{K} = \left(\underbrace{\kappa_1, \kappa_2, \dots, \kappa_{g_1}}_{\varrho_1, \kappa'_1 = \kappa_{g_1}}, \underbrace{\kappa_{g_1+1}, \kappa_{g_1+2}, \dots, \kappa_{g_2}}_{\varrho_2, \kappa'_2 = \kappa_{g_2}}, \dots, \underbrace{\kappa_{g_{u-1}+1}, \kappa_{g_{u-1}+2}, \dots, \kappa_{g_u}}_{\varrho_u = \dot{q}, \kappa'_u = \kappa_{g_u} = 0} \right)$$

mit $\kappa_1 = \kappa_2 = \dots = \kappa_{g_1} \neq \kappa_{g_1+1} = \kappa_{g_1+2} = \dots$

Die Gesamtanzahl⁵⁴ der Permutationen von \dot{K} mit klassenweise äquivalenten Elementen ist dann

$$|\dot{K}| = |\dot{K}| = (i+1)! \left(\prod_{\alpha=1}^u \varrho_\alpha! \right)^{-1} = \frac{(i+1)!}{\dot{q}!} \left(\prod_{\alpha=1}^{u-1} \varrho_\alpha! \right)^{-1}. \quad (3.50)$$

4.4 Aufteilung der Permutationsmenge \dot{K} in die vier Gruppen der Elementarfaktoren $\mathfrak{S}^{(i)}$, Gleichungen (3.31-1) bis (3.31-4), unter Eliminierung solcher Permutationen, in denen das „Muster“ $\kappa_h = \kappa_{h+1} = 0$ für $h = 1, 2, \dots, i$ auftritt

4.5 Zur nächsten Partition von i übergehen, Schritt 4.1, falls $|K| < i$; sonst implizite Schleife (nach insgesamt v_i Zyklen, siehe Gleichung (2.11)) beenden

5 Zu $i := i + 1$ übergehen, Schritt 4, falls $i < n$; sonst Schleife beenden

6 (Zählerprodukte) Für $i = 1, 3, \dots, n$, siehe Flußdiagrammelement (2) und (in dieser Implementierung darin eingeschlossen) Element (3):

6.1 Gemäß Gleichungen (3.36-2) bis (3.36-4) die drei Teilmengen der \mathcal{T} -Produkte i ter Ordnung zusammensetzen

6.2 Vereinigungsmenge nach Gleichung (3.35) bilden

6.3 Identische \mathcal{T} -Produkte zusammenfassen: Nur einen Repräsentanten speichern, alle übrigen im Vorfaktor berücksichtigen⁵⁵

⁵⁴Die vollständige Zerlegung des rechtsseitigen Ausdrucks in Gleichung (3.50) legt die Optimierung nahe, zunächst nur alle Permutationen des auf das „ $(\kappa > 0, \kappa = 0)$ -Tupel“ reduzierten Tupels \dot{K} mit $(i+1)!/(q! \cdot \dot{q}!)$ Möglichkeiten zu bilden, dann alle Vertauschungen mit zwei oder mehr aufeinanderfolgenden Nullen zu entfernen und abschließend sämtliche Permutationen von \dot{K} auf die „ $(\kappa > 0)$ -Plätze“ zu verteilen, entsprechend mit $q!/\prod_{\alpha=1}^{u-1} \varrho_\alpha!$ Möglichkeiten.

⁵⁵Aufgrund der Vorsortierung reduziert sich die Anzahl unterschiedlicher \mathcal{T} -Produkte i ter Ordnung immer auf jene der Elementarfaktoren $\mathfrak{S}_{EG}^{(i)}$ im $\tilde{\mathcal{T}}_{EGEE}^{(i)}$ -Produkt, siehe Gleichung (3.36-2) ($\mathfrak{S}_{EG}^{(i)}$ links stehend für $r = 0$ mit von rechts konkateniertem $\mathfrak{S}_{EE}^{(0)}$); Tabelle 3.3 belegt eindrucksvoll die „Datenreduktion“. Die Kenntnis der Struktur der geordneten \mathcal{T} -Produkt-Menge ermöglicht überdies eine Parallelisierung dieses Programmschrittes ohne aufwendige Synchronisation der Vorfaktoraktualisierung. Die geschieht durch die Verteilung der Elementarprodukte von $\mathcal{T}_1^{(i)}$ bis $\mathcal{T}_{u_i}^{(i)}$ an mehrere „Rechenprozeßfäden“ (*threads*) und nachfolgendes paralleles Durchsuchen der Vereinigungsmenge Gleichung (3.35) ab Element $u_i + 1$. Daß aus datentechnischer Sicht die Schreibvorgänge unabhängig voneinander sind, ermöglicht den Beibehalt einer *in-place*-Ausführung und somit den Verzicht auf (später abzuleichende) Hilfslisten.

- 6.4 Optional⁵⁶ die Menge der \mathfrak{D} -Produkte i ter Ordnung gemäß Gleichung (3.40-2) zusammensetzen
- 6.5 Optional identische \mathfrak{D} -Produkte zusammenfassen: Nur einen Repräsentanten speichern, alle übrigen im Vorfaktor berücksichtigen⁵⁷
- 6.6 „Zählerprozesse“ in $\check{\mathfrak{Z}}_{\text{red}}^{(i)}$ zusammenfassen, optional \mathfrak{T} - und \mathfrak{D} -Produkte zuvor miteinander verrechnen
- 7 Zur nächsten Störungsordnung $i := i + 2$ übergehen, Schritt 6, falls $i < n$; sonst Schleife beenden
- 8 (Nennerprodukte) Für $i = n - 1, n - 3, \dots, 2$, siehe Flußdiagrammelement (4):
- 8.1 Partitionierung der Zahl $p = i/2$ in q lexikographisch (absteigend) geordnete Summanden v_a ($a = 1, \dots, q$), formal beginnend mit $q = 1$ (also $v_1 = p$), endend mit $q = p$ (folglich $v_1 = v_2 = \dots = v_p = 1$):

$$p = \sum_{a=1}^q v_a \quad \text{mit} \quad v_1 \geq v_2 \geq \dots \geq v_q > 0 \quad (3.51)$$

Das Tupel der Summanden einer Partition wird im Weiteren mit \mathcal{Y} bezeichnet; es gilt: $|\mathcal{Y}| = q$.

- 8.2 Bildung sämtlicher Permutationen $\tilde{\mathcal{Y}}$ des q -Tupels \mathcal{Y} , $|\tilde{\mathcal{Y}}| = q$, deren Gesamtmenge $\mathcal{Y} = \{\tilde{\mathcal{Y}}_1, \tilde{\mathcal{Y}}_2, \dots\}$ mit $\tilde{\mathcal{Y}}_1 = \mathcal{Y}$ sei.

$$\tilde{\mathcal{Y}}_1 = (v_1, v_2, \dots, v_{g_1}, v_{g_1+1}, v_{g_1+2}, \dots, v_{g_2}, \dots, v_{g_{u-1}+1}, v_{g_{u-1}+2}, \dots, v_{g_u})$$

⋮

$$\tilde{\mathcal{Y}}_{v-1} = (v_{g_{u-1}+1}, v_{g_{u-1}+2}, \dots, v_{g_u}, \dots, v_1, v_2, \dots, v_{g_1}, v_{g_1+1}, v_{g_1+2}, \dots, v_{g_2})$$

$$\tilde{\mathcal{Y}}_v = (v_{g_{u-1}+1}, v_{g_{u-1}+2}, \dots, v_{g_u}, \dots, v_{g_1+1}, v_{g_1+2}, \dots, v_{g_2}, v_1, v_2, \dots, v_{g_1})$$

Zur kontrollweisen Berechnung von $v = |\mathcal{Y}|$ bedient man sich der Vorgehensweise in Schritt 4.3: Einteilung der v_a in u Gruppen ϱ_α gleicher Summanden ($\alpha = 1, \dots, u$); v'_α ist dabei ein „repräsentativer Vertreter der jeweiligen Gruppe“ aus P :

$$p = \sum_{a=1}^q v_a = \sum_{\alpha=1}^{u-1} \varrho_\alpha v'_\alpha, \quad \sum_{\alpha=1}^u \varrho_\alpha = q,$$

⁵⁶Auf die rechentechnisch effizienzoptimierte Realisierung des Gesamtverfahrens abzielend wird ab dem Anwendungsabschnitt 3.5 davon ausgegangen, daß die \mathfrak{D} -Elementarprodukte noch *nicht* im Prozeßkoeffizienten enthalten sind, sondern gemäß Gleichung (1.17) erst durch den globalen Vorfaktor $(n+1)^{-1}$ Berücksichtigung finden. Der Widerspruch zum laufenden Abschnitt, welcher die \mathfrak{D} -Produkte – explizit oder implizit – stets mit einberechnet, besteht somit nur scheinbar, da hier die Reihenentwicklung des Hamiltonoperators zunächst als in sich geschlossener, von der späteren konkreten Anwendung unabhängiger Algorithmus beschrieben wird.

⁵⁷Eingedenk Gleichung (3.40-2) kann analog zu Programmschritt 6.3 vorgegangen werden (siehe Fußnote 55).

in Tupelschreibweise, $g_\xi = \sum_{\alpha=1}^{\xi} \varrho_\alpha$:

$$\Upsilon = \left(\underbrace{v_1, v_2, \dots, v_{g_1}}_{\varrho_1, v'_1=v_{g_1}}, \underbrace{v_{g_1+1}, v_{g_1+2}, \dots, v_{g_2}}_{\varrho_2, v'_2=v_{g_2}}, \dots, \underbrace{v_{g_{u-1}+1}, v_{g_{u-1}+2}, \dots, v_{g_u}}_{\varrho_u, v'_u=v_{g_u}} \right)$$

mit $v_1 = v_2 = \dots = v_{g_1} \neq v_{g_1+1} = v_{g_1+2} = \dots$

Die Gesamtanzahl der Permutationen von Υ mit klassenweise äquivalenten Elementen ist dann

$$v = |\Upsilon| = q! \left(\prod_{\alpha=1}^u \varrho_\alpha! \right)^{-1}. \tag{3.52}$$

8.3 Aus allen $\tilde{\Upsilon} \in \Upsilon$ die \mathcal{N} -Produkte n ter Ordnung gemäß Gleichung (3.44), darin $u = n - i$ und $v_j = 2\llbracket \tilde{\Upsilon} \rrbracket_j$, zusammensetzen

8.4 Zur nächsten Partition von p übergehen, Schritt 8.1, falls $|\Upsilon| < p$; sonst implizite Schleife (nach insgesamt v_i Zyklen, siehe Gleichung (2.11)) beenden

9 Zur Störungsordnung $i := i - 2$ übergehen, Schritt 8, falls $i > 2$; sonst Schleife beenden

10 Identische \mathcal{N} -Produkte, die „Nennerprozesse“, in $\check{\mathcal{N}}_{\text{red.}}^{(n)}$ zusammenfassen: Nur einen Repräsentanten speichern, alle übrigen im Vorfaktor berücksichtigen

11 Zähler und Nenner in $\check{\mathcal{P}}_{\text{red.}}^{(n)}$ zusammenfassen, siehe Flußdiagrammelement (5)

12 Prozeßmenge auf $\hat{\mathcal{P}}^{(n)}$ reduzieren und entsprechend der Operatorklassenzugehörigkeit, Gleichungen (3.45) und (3.46), auf die $\hat{\mathcal{P}}_f^{(n)}$ verteilen, siehe Element (6)

13 Lokale Zeitmessung beenden und Rechenzeit ausgeben

14 Verfahren beenden und die Hüpfprozesse der Störungsordnung n dem aufrufenden Hauptprogramm übergeben

3.3.3. *Beispiel.* Die im vorigen Abschnitt dargelegte allgemeine Vorgehensweise zur Erzeugung sämtlicher relevanter Hamiltonoperatorterme, also der den Hüpfprozessen zugrundeliegenden Operatorsequenzen, wollen wir nun bei halber Bandfüllung exemplarisch auf die Störungsreihenordnung $n = 3$ anwenden. Dabei beginnen wir mit der Herleitung der \mathcal{S} -Elementarfaktoren, Programmschritt 4, wie in folgender Tabelle zusammengefaßt; darin ist $i = 0, 1, \dots, n$ und $j = 1, 2, \dots, v_i$ die Nummer der jeweiligen Partition, v_i siehe Tabelle 2.5. Man beachte, daß die Faktoren in der mit einem Stern gekennzeichneten Zeile ($i = 3$ und $j = 1$) im Weiteren durch zwei beziehungsweise drei aufeinanderfolgende Nullen entfallen⁵⁸.

i	j	\dot{K}	$\dot{\kappa}$	$\dot{\mathcal{K}}$	$\check{\mathcal{S}}^{(i)}$	g_i
0	1	$\boxed{0}$	1	$\{\boxed{0}\}$		
					$\check{\mathcal{S}}_{\text{EE}}^{(0)} = \{\boxed{0}\}$	1

Fortsetzung auf der nächsten Seite

⁵⁸Siehe Fußnote 53

i	j	\dot{K}	\dot{k}	\dot{K}	$\check{S}^{(i)}$	\mathfrak{F}_i
<i>Fortsetzung</i>						
1	1	$\boxed{10}$	2	$\{\boxed{10}, \boxed{01}\}$	$\check{S}_{EG}^{(1)} = \{\boxed{01}\}, \check{S}_{GE}^{(1)} = \{\boxed{10}\}$	2
2	1	$\boxed{200}$	3	$\{\boxed{200}, \boxed{020}, \boxed{002}\}$	$\check{S}_{EE}^{(2)} = \{\boxed{020}\}, \check{S}_{EG}^{(2)} = \{\boxed{011}\},$ $\check{S}_{GE}^{(2)} = \{\boxed{110}\}, \check{S}_{GG}^{(2)} = \{\boxed{101}\}$	4
	2	$\boxed{110}$	3	$\{\boxed{110}, \boxed{101}, \boxed{011}\}$		
3	1	$\boxed{3000}$	4	$\{\boxed{3000}, \boxed{0300}, \boxed{0030}, \boxed{0003}\}^*$	$\check{S}_{EE}^{(3)} = \{\boxed{0120}, \boxed{0210}\},$ $\check{S}_{EG}^{(3)} = \{\boxed{0111}, \boxed{0102}, \boxed{0201}\},$ $\check{S}_{GE}^{(3)} = \{\boxed{1110}, \boxed{1020}, \boxed{2010}\},$ $\check{S}_{GG}^{(3)} = \{\boxed{1011}, \boxed{1101}\}$	10
	2	$\boxed{2100}$	12	$\{\boxed{2100}, \boxed{2010}, \boxed{2001}, \boxed{0210},$ $\boxed{0201}, \boxed{0021}, \boxed{1200}, \boxed{1020},$ $\boxed{1002}, \boxed{0120}, \boxed{0102}, \boxed{0012}\}$		
	3	$\boxed{1110}$	4	$\{\boxed{1110}, \boxed{1101}, \boxed{1011}, \boxed{0111}\}$		

Im sich anschließenden Programmschritt 6 werden für $i = 1, 3, \dots, n$ zunächst (und in dieser Implementierung gemäß Gleichung (1.18) einzig) die Mengen der \mathcal{T} -Elementarprodukte zusammengesetzt, wie im Folgenden mit $i' = i - r$ tabelliert.

i	i'	r	Typ	$\check{T}^{(i)}$	$\check{T}_{red.}^{(i)}$	u_i
1	1	0	EGEE	$\{\boxed{01} \circ \hat{T} \boxed{0}\}$	$\{ + 2 \boxed{010} \}$	1
	0	1	EEGE	$\{\boxed{0} \circ \hat{T} \boxed{10}\}$		
3	3	0	EGEE	$\{\boxed{0111} \circ \hat{T} \boxed{0}, \boxed{0102} \circ \hat{T} \boxed{0}, \boxed{0201} \circ \hat{T} \boxed{0}\}$		
	2	1	EEGE	$\{\boxed{020} \circ \hat{T} \boxed{10}\}$		

Fortsetzung auf der nächsten Seite

i	i'	r	Typ	$\check{\mathcal{T}}^{(i)}$	$\check{\mathcal{T}}_{\text{red.}}^{(i)}$	u_i
<i>Fortsetzung</i>						
1	2		EGGE	$\{\boxed{011} \circ \hat{T} \boxed{10}\}$		
			EGEE	$\{\boxed{01} \circ \hat{T} \boxed{020}\}$		
			EGGE	$\{\boxed{01} \circ \hat{T} \boxed{110}\}$		
0	3		EEGE	$\{\boxed{0} \circ \hat{T} \boxed{1110}, \boxed{0} \circ \hat{T} \boxed{1020}, \boxed{0} \circ \hat{T} \boxed{2010}\}$		
				$\{+4 \boxed{01110}, +3 \boxed{01020}, +3 \boxed{02010}\}$		
						3

Man beachte, daß die Produktbildung abweichend von Programmschritt 6.1 hier nach Gleichung (3.36-1) und den analog formulierten Gleichungen (3.36-3) und (3.36-4) erfolgt ist, nicht aber nach der verkürzenden Vorgehensweise in Gleichung (3.36-2) welche

$$\check{\mathcal{T}}_{\text{red.}}^{(3')} = \{+4 \boxed{01110}, +2 \boxed{01020}, +4 \boxed{02010}\}$$

zum Ergebnis hat. Diese „Asymmetrie“ entsteht aus $(\mathcal{S}_{\text{EE},1}^{(2)} \hat{T} \mathcal{S}_{\text{GE},1}^{(1)}) = \text{adj}(\mathcal{S}_{\text{EG},1}^{(1)} \hat{T} \mathcal{S}_{\text{EE},1}^{(2)})$: Das Produkt $\boxed{020} \circ \hat{T} \boxed{10}$ wird Gleichung (3.36-3) entsprechend mit dem Vorfaktor 2 erzeugt, das Produkt $\boxed{01} \circ \hat{T} \boxed{020}$ hingegen formal übergangen.

Im Falle der Berücksichtigung angeregter Ausgangszustände werden zusätzlich die \mathfrak{D} -Elementarproduktmengen benötigt, deren Berechnung für den Produkttyp EGGE bei Halbfüllung ohne Anregung in der folgenden Tabelle mit $i' = i + 1 - r$ demonstriert wird.

i	i'	r	$\check{\mathfrak{D}}^{(i)}$	$\check{\mathfrak{D}}_{\text{red.}}^{(i)}$	u_i	
1	1	1	$\{\boxed{01} \circ \hat{D} \boxed{10}\}$			
				$\{-1 \boxed{010}\}$	1	
3	3	1	$\{\boxed{0111} \circ \hat{D} \boxed{10}, \boxed{0102} \circ \hat{D} \boxed{10}, \boxed{0201} \circ \hat{D} \boxed{10}\}$			
			2			$\{\boxed{011} \circ \hat{D} \boxed{110}\}$
			1			$\{\boxed{01} \circ \hat{D} \boxed{1110}, \boxed{01} \circ \hat{D} \boxed{1020}, \boxed{01} \circ \hat{D} \boxed{2010}\}$
				$\{-3 \boxed{01110}, -2 \boxed{01020}, -2 \boxed{02010}\}$	3	

Auch diese Produktbildung ist abweichend von der allgemeinen Prozedur, vergleiche Programmschritt 6.4, nicht nach Gleichung (3.40-2), sondern nach Gleichung (3.40-1) vorgenommen worden, welche jedoch die gleichen reduzierten Produktmengen, $\check{\mathfrak{D}}_{\text{red.}}^{(3')} = \check{\mathfrak{D}}_{\text{red.}}^{(3)}$, zur Folge haben.

Bei „ausführlichem“ \mathcal{T} -Elementarprodukt (siehe obige Tabelle) ergibt sich dementsprechend die Zählerprozeßmenge zu

$$\check{\mathcal{Z}}_{\text{red.}}^{(3)} = \check{\mathcal{T}}_{\text{red.}}^{(3)} \sqsupset \check{\mathcal{D}}_{\text{red.}}^{(3)} = \{ +1 \boxed{01110}, +1 \boxed{01020}, +1 \boxed{02010} \}$$

und bei optimierter Rechnung zu

$$\check{\mathcal{Z}}_{\text{red.}}^{(3)'} = \check{\mathcal{T}}_{\text{red.}}^{(3)'} \sqsupset \check{\mathcal{D}}_{\text{red.}}^{(3)} = \{ +1 \boxed{01110}, 0 \boxed{01020}, +2 \boxed{02010} \};$$

aufgrund $\boxed{02010} = \text{adj}(\boxed{01020})$ sind beide Ergebnisse gleich.

Der nun folgende Programmschritt 8, Bildung der \mathcal{N} -Elementarproduktmengen für die Berücksichtigung der „Nennerprozesse“, wird durch $i = n-1, n-3, \dots, 2$ erstmals für die betrachtete Ordnung $n = 3$ mit $i = 2$ und entsprechend (Schritt 8.1) $p = 1 = q$ ausgeführt. Für die einzige ($v_1 = 1$) sich somit ergebende Partition $\Upsilon_1 = (1)$ existiert bei streng formalem Vorgehen nur eine ($v = 1$) Permutation, $\tilde{\Upsilon}_1 = (1)$. Mit $u = n-i = 1$ und $v_1 = 2\llbracket \tilde{\Upsilon}_1 \rrbracket_1 = 2$ folgt mit

$$\check{\mathcal{Z}}_{\text{red.}}^{(1)} = \check{\mathcal{Z}}_{\text{red.}}^{(1)'} = \{\mathcal{Z}^{(1)}\} = \{ +1 \boxed{010} \}$$

aus Gleichung (3.43),

$$\mathcal{N}^{(3)} = \mathcal{Z}^{(u=1)} \hat{\mathcal{N}} \mathcal{S}_{\text{EE}}^{(v_1=2)} = +1 \boxed{010} \circ \hat{\mathcal{N}} \boxed{020} = +1 \cdot (-1)^{q=1} \boxed{01020},$$

über Schritt 10 die Elementarproduktmenge $\check{\mathcal{N}}_{\text{red.}}^{(3)} = \{\mathcal{N}^{(3)}\} = \{ -1 \boxed{01020} \}$.

Schritt 11 der allgemeinen Vorgehensweise liefert nun die Vereinigung von Zähler- und Nennerprozessen, bei ausführlicher Rechnung

$$\check{\mathcal{P}}_{\text{red.}}^{(3)} = \check{\mathcal{Z}}_{\text{red.}}^{(3)} \sqsupset \check{\mathcal{N}}_{\text{red.}}^{(3)} = \{ +1 \boxed{01110}, 0 \boxed{01020}, +1 \boxed{02010} \},$$

optimiert

$$\check{\mathcal{P}}_{\text{red.}}^{(3)'} = \check{\mathcal{Z}}_{\text{red.}}^{(3)'} \sqsupset \check{\mathcal{N}}_{\text{red.}}^{(3)} = \{ +1 \boxed{01110}, -1 \boxed{01020}, +2 \boxed{02010} \}. \quad (3.53)$$

Unter Berücksichtigung der Gleichheit adjungierter Elementarprozesse erhalten wir das Endergebnis

$$\hat{\mathcal{P}}^{(3)} = \{ +1 \boxed{01110}, +1 \boxed{02010} \}, \quad (3.54)$$

welches in zwei Prozeßklassen zerfällt (Programmschritt 12):

$$\begin{aligned} \hat{\mathcal{P}}_0^{(3)} &= \{ +1 \boxed{01110} \} \quad \text{und} \\ \hat{\mathcal{P}}_1^{(3)} &= \{ +1 \boxed{02010} \}. \end{aligned}$$

Unter Berücksichtigung des Zusammenhanges

$$\hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 = \text{adj}(\hat{P}_0 \hat{T} \hat{S}^1 \hat{T} \hat{P}_0 \hat{T} \hat{S}^2 \hat{T} \hat{P}_0)$$

zeigt der Vergleich der Prozeß- beziehungsweise Termkoeffizienten und \mathcal{S} -Exponenten mit \tilde{h}_3^{HBF} , Gleichung (1.26), Übereinstimmung.

Werden nach Gleichung (1.17) die \mathfrak{D} -Elementarprodukte erst in den Elementarprozessen durch den Vorfaktor $(n+1)^{-1}$ berücksichtigt, ändern sich die Zwischenergebnisse entsprechend:

$$\begin{aligned}\check{Z}_{\text{red.}}^{(1)'} &= \check{J}_{\text{red.}}^{(1)'} = \{ + 2 \boxed{010} \} \quad \text{und} \\ \check{Z}_{\text{red.}}^{(3)'} &= \check{J}_{\text{red.}}^{(3)'} = \{ + 4 \boxed{01110}, + 2 \boxed{01020}, + 4 \boxed{02010} \}\end{aligned}$$

sowie

$$\check{N}^{(3)} = \{ \llbracket \check{Z}_{\text{red.}}^{(1)'} \rrbracket_1 \hat{N} s_{\text{EE}}^{(2)} \} = \{ - 2 \boxed{01020} \}.$$

Dadurch gewinnt nach $\check{Z}_{\text{red.}}^{(3)'} \sqsupset \check{N}^{(3)}$ das mit obigem Resultat identische Endergebnis formal folgendes Aussehen:

$$\hat{\rho}^{(3)} = \left\{ + \frac{4}{4} \boxed{01110}, + \frac{4}{4} \boxed{02010} \right\}.$$

3.3.4. Optimierungen. Der zunächst und in dieser Arbeit betrachtete Fall halbgefüllter und anregungsfreier Ausgangszustände legt zwei rechentechnische Vereinfachungen nahe, welche über die in Abschnitt 3.3.2 beschriebene Prozeßgenerierung hinausgehen. Unmittelbar ist dies einerseits die Zusammenfassung zueinander adjungierter Prozesse, wie bereits im vorausgehenden Beispiel beim Übergang von Gleichung (3.53) zu Gleichung (3.54) angewendet. Andererseits lassen sich mittelbar, nämlich unter Berücksichtigung der höchstmöglichen Doppelbesetzungsanzahl d_j für jeden Zwischenzustand $j = 2, 3, \dots, n+1$ einer Hüpfsequenz⁵⁹ der n ten Störungsrechnungsordnung Prozesse zu $\check{\rho}^{(n)} = (e, \check{\rho}^{(n)})$ komprimieren, indem nach Programmschritt II die zu $d_j = 1$ gehörigen \hat{S} -Exponenten gleich eins gesetzt und abschließend sämtliche Prozesse erneut zusammengefaßt werden. Darin sind auch neu entstehende Adjunktionen einzubeziehen, wie unten dargestelltes Beispiel für $f = 1$ verdeutlicht.

Eingehender werden die möglichen Doppelbesetzungsverläufe anhand der möglichen *Hüpfsequenzen* in den Abschnitten 3.4.7 und 3.5 untersucht, wiewohl eine entsprechende Betrachtung auch aufgrund der möglichen *Hüpfprozesse* durchführbar ist. Für die hier beschriebene Optimierung genügt die Überlegung, welche Zwischenzustände genau ein Elektronenpaar aufweisen können, nämlich problembedingt ausschließlich solche, die dem (wiederhergestellten) Ausgangszustand im Sequenzverlauf benachbart sind, welche ihrerseits als Null im \hat{S} -Exponenten-Tupel auftreten.

Die Änderung von \hat{S} -Exponenten wirkt sich auf das Vorzeichen des Prozeßkoeffizienten e und somit des Energiebeitrags, formal

$$\begin{aligned}e \cdot (-d_j)^{k_j} &= (-1)^{k_j} e \cdot (d_j)^{k_j} \\ &= (-1)^{k_j} e \cdot 1^1 \Leftrightarrow d_j = 1, \end{aligned} \quad (3.55)$$

⁵⁹Forderungsgemäß gilt für den Ausgangs- und den Endzustand $d_1 = d_{n+2} = 0$.

effektiv nicht aus: Im Falle halber Bandfüllung läßt sich generell das negative Vorzeichen der Doppelbesetzungszahl vorziehen, da immer eine ungerade Anzahl an ungeraden \hat{S} -Exponenten und folglich eine ungerade Summe derselben vorliegt:

$$\begin{aligned} \left(\prod_{j=2}^{n+1} (-1)^{\kappa_j} \right) e &= (-1)^{\kappa_2 + \kappa_3 + \dots + \kappa_{n+1}} e \\ &= -1e \Leftarrow \sum_{j=2}^{n+1} \kappa_j \text{ ungerade.} \end{aligned} \tag{3.56}$$

Beispiel: Sämtliche Hüpfprozesse $\hat{P}^{(n)}$ der Ordnung $n = 5$, aufgeteilt in die $2^{(n-1)/2} = 4$ möglichen Prozeßklassen f [15]:

f	ohne Kompression		mit Kompression		zusammengefaßt				
	e	p	e	\check{p}	e	\check{p}			
0	$+\frac{6}{6}$	<table border="1"><tr><td>0 1 1 1 1 1 0</td></tr></table>	0 1 1 1 1 1 0	$+\frac{6}{6}$	<table border="1"><tr><td>0 1 1 1 1 1 0</td></tr></table>	0 1 1 1 1 1 0	$+\frac{6}{6}$	<table border="1"><tr><td>0 1 1 1 1 1 0</td></tr></table>	0 1 1 1 1 1 0
0 1 1 1 1 1 0									
0 1 1 1 1 1 0									
0 1 1 1 1 1 0									
1	$-\frac{6}{6}$	<table border="1"><tr><td>0 1 1 1 0 2 0</td></tr></table>	0 1 1 1 0 2 0	$-\frac{6}{6}$	<table border="1"><tr><td>0 1 1 1 0 1 0</td></tr></table>	0 1 1 1 0 1 0	(enthalten in $f = 2$)		
0 1 1 1 0 2 0									
0 1 1 1 0 1 0									
2	$-\frac{4}{6}$	<table border="1"><tr><td>0 1 0 1 1 2 0</td></tr></table>	0 1 0 1 1 2 0	$-\frac{4}{6}$	<table border="1"><tr><td>0 1 0 1 1 1 0</td></tr></table>	0 1 0 1 1 1 0	$-\frac{18}{6}$	<table border="1"><tr><td>0 1 0 1 1 1 0</td></tr></table>	0 1 0 1 1 1 0
	0 1 0 1 1 2 0								
	0 1 0 1 1 1 0								
0 1 0 1 1 1 0									
$-\frac{6}{6}$	<table border="1"><tr><td>0 1 0 1 2 1 0</td></tr></table>	0 1 0 1 2 1 0	$-\frac{6}{6}$	<table border="1"><tr><td>0 1 0 1 2 1 0</td></tr></table>	0 1 0 1 2 1 0	$-\frac{6}{6}$	<table border="1"><tr><td>0 1 0 1 2 1 0</td></tr></table>	0 1 0 1 2 1 0	
0 1 0 1 2 1 0									
0 1 0 1 2 1 0									
0 1 0 1 2 1 0									
$-\frac{8}{6}$	<table border="1"><tr><td>0 1 0 2 1 1 0</td></tr></table>	0 1 0 2 1 1 0	$-\frac{8}{6}$	<table border="1"><tr><td>0 1 0 1 1 1 0</td></tr></table>	0 1 0 1 1 1 0				
0 1 0 2 1 1 0									
0 1 0 1 1 1 0									
3	$+\frac{6}{6}$	<table border="1"><tr><td>0 1 0 2 0 2 0</td></tr></table>	0 1 0 2 0 2 0	$+\frac{6}{6}$	<table border="1"><tr><td>0 1 0 1 0 1 0</td></tr></table>	0 1 0 1 0 1 0	$+\frac{12}{6}$	<table border="1"><tr><td>0 1 0 1 0 1 0</td></tr></table>	0 1 0 1 0 1 0
	0 1 0 2 0 2 0								
	0 1 0 1 0 1 0								
0 1 0 1 0 1 0									
$+\frac{4}{6}$	<table border="1"><tr><td>0 1 0 1 0 3 0</td></tr></table>	0 1 0 1 0 3 0	$+\frac{4}{6}$	<table border="1"><tr><td>0 1 0 1 0 1 0</td></tr></table>	0 1 0 1 0 1 0				
0 1 0 1 0 3 0									
0 1 0 1 0 1 0									
$+\frac{2}{6}$	<table border="1"><tr><td>0 1 0 3 0 1 0</td></tr></table>	0 1 0 3 0 1 0	$+\frac{2}{6}$	<table border="1"><tr><td>0 1 0 1 0 1 0</td></tr></table>	0 1 0 1 0 1 0				
0 1 0 3 0 1 0									
0 1 0 1 0 1 0									

Die auf diesen Überlegungen beruhende drastische Verminderung der Prozeßanzahl zeigt bis einschließlich Störungsrechnungsordnung U^{-17} mit dem jeweiligen Rechenzeitbedarf auf einem CISC-server des Typs HP AM447A [84] bei paralleler Verarbeitung (alle 80 Prozessorkerne mit jeweils zwei threads belegt) Tabelle 3.4.

3.3.5. *Graphische Darstellung von Prozessen.* Eine naheliegende Veranschaulichung der Hüpfprozesse beruht auf einer abstrahierten, „Fieberkurven-ähnlichen“ Auftragung⁶⁰ der \hat{S} -Exponenten in $\check{p}^{(n)}$ gegen die Nummer des Zustands, auf welche die entsprechenden \hat{S} -Operatoren jeweils wirken. In Tabelle 3.5 sind in dieser Weise die Prozesse bis $n = 5$ dargestellt; die vorteilhafte Verwendung dieser Notation wird in Abschnitt 3.5

⁶⁰Diese eingängige Visualisierungsmethode ähnelt einer von EDWARD TUFTE 2004 insbesondere für Zeitreihen (beispielsweise Finanzmarktdaten) entwickelten Darstellungsart [45].

n	1	3	5	7	9	11	13	15	17
$ \check{Z}_{\text{red.}}^{(n)} $	1	3	19	141	1107	8953	73789	616227	5196627
$ \check{N}^{(n)} $	0	1	10	98	950	9131	87230	829629	7863926
$ \hat{\rho}^{(n)} $	1	2	11	65	386	2413	*	*	*
ohne Adjungierte	1	2	8	41	236	1413	*	*	*
nach Kompression	1	2	4	14	48	193	795	3412	14803
Pz./s		$\leq 0,25$			1,29	21,8	232	53600	5460000

Tabelle 3.4. Anzahl der Prozesse nach verschiedenen Optimierungsstufen ausgehend vom Basisalgorithmus, Abschnitt 3.3.2, in Fortsetzung von Tabelle 3.3. Die durch einen Stern vertretenen Zahlen werden in der auf minimale Rechenzeit hin optimierten „Produktionsversion“ des Programmes nicht mehr kontrollweise ausgegeben. In der letzten Zeile ist orientierungshalber jeweils die Prozessorzeit (Pz.) für die n te Ordnung angegeben, also die akkumulierte Rechenzeit aller Prozessoren bei hiesiger paralleler Verarbeitung; technische Aspekte dazu finden sich detailliert im Abschnitt 5.1.

n	(f, e, \check{p})
1	$(0, +1, \boxed{0\ 1\ 0}) \mapsto (0, +1, \begin{array}{ c c c } \hline \square & \square & \square \\ \hline \end{array})$
3	$(0, +1, \boxed{0\ 1\ 1\ 1\ 0}) \mapsto (0, +1, \begin{array}{ c c c c } \hline \square & \square & \square & \square \\ \hline \end{array}) \wedge (1, -1, \boxed{0\ 1\ 0\ 1\ 0}) \mapsto (1, -1, \begin{array}{ c c c c } \hline \square & \square & \square & \square \\ \hline \end{array})$
5	$(0, +1, \begin{array}{ c c c c c } \hline \square & \square & \square & \square & \square \\ \hline \end{array}) \wedge (2, -3, \begin{array}{ c c c c c } \hline \square & \square & \square & \square & \square \\ \hline \end{array}) \wedge (2, -1, \begin{array}{ c c c c c } \hline \square & \square & \square & \square & \square \\ \hline \end{array}) \wedge (3, +2, \begin{array}{ c c c c c } \hline \square & \square & \square & \square & \square \\ \hline \end{array})$

Tabelle 3.5. Graphische Veranschaulichung der Hüpfprozesse bis einschließlich Störungsordnung $n = 5$ [15].

beschrieben. Da die Hüpfprozesse mit den im folgenden Abschnitt 3.4 diskutierten Hüpfsequenzen „verschränkt“ sind, kann diese Darstellungsart auf die Anzahl der Doppelbesetzungen des Bethegitterausschnittes nach jedem Hüpfschritt übertragen werden, siehe Abschnitt 3.4.7.

3.4. Generierung der Hüpfsequenzen. Der dritte wesentliche Schritt des Gesamtverfahrens und eigentlicher Programmkern, siehe Element 3 in Flußdiagramm 3.1, besteht

in der iterativ-rekursiven Generierung sämtlicher Hüpfsequenzen, welche wir im Folgenden auch als *Sequenzierung der Ausgangszustände* bezeichnen. Während die im vorigen Abschnitt 3.3 hergeleiteten Prozesse und deren Gewichte erst bei der Berechnung des Sequenzbeitrages zum Störungsreihenkoeffizienten im Abschnitt 3.5 gebraucht werden, stützt sich der hier detailliert dargelegte Algorithmus auf die im ersten Programmteil, Abschnitt 3.2, generierten Bethegitterausschnitte. Weil deren Verarbeitung in hohen Ordnungen oder bei komplexeren Fragestellungen, etwa der Greenschen Funktion („Lochhüpfen“, das virtuelle Propagieren von „Unbesetzungen“ von Gitterplatz zu Gitterplatz wird relevant), neben der Parallelisierung höchstmögliche Effizienz verlangt, ist eingangs die Definition einer größeren Anzahl an teilweise binärkodierten Hilfsstrukturen notwendig.

3.4.1. Farbseparation. Um den besonderen quantenmechanischen Gesetzen für fermionische Systeme – oder solche mit $(S > 0)$ -Bosonen – in der rechentechnischen Anwendung Genüge zu tun, müssen wir zur Definition der Zustände und deren Übergänge ineinander durch die Elektronenbewegungen den insbesondere in den Abschnitten 2.2.2 und 2.2.3 eingeführten Formalismus gefärbter Bäume erweitern.

Auf ein intuitives Verständnis bauend haben wir bereits im einleitenden Kapitel 1 den Plätzen auf den Bethegitterausschnitten je nach Besetzung jeweils eine von vier Farben $X = \{\chi_0, \chi_1, \chi_2, \chi_3\} \mapsto \{\square, \blacksquare, \blacklozenge, \blacktriangledown\}$ ($|X| = c = 4$) zugewiesen: Weiß für ein „Loch“ (\emptyset), für ein Elektron je nach Spinsorte Gelb-Orange (\uparrow) oder Rot-Orange (\downarrow) und Blau für eine Doppelbesetzung ($\uparrow\downarrow$). Sehen wir ersteren und letzteren Gitterplatzzustand als Besetzung mit einem *Quasiteilchen* (Holon und Doublon) an, die jedoch nicht Gegenstand dieser Betrachtung sind, liegen bezüglich der Elektronen effektiv nur zwei „Grundfarben“ $X \supset X' = \{\chi_1, \chi_2\}$ ($|X'| = c' = 2$) vor, aus welchen die Doppelbesetzungsfarbe formal als Mischung hervorgeht. Generell läßt sich für Fermionen der Zusammenhang

$$c = \sum_{i=0}^{c'} \binom{c'}{i} = 2^{c'} = 1 + \sum_{i=0}^{c'-1} 2^i \quad (3.57)$$

angeben⁶¹. Die Summe über die „Kombinationen variabler Länge ohne Wiederholungen“ verdeutlicht nach Umschrift auf die rechte Gleichungsseite die Systematik beim Aufbau der Farben: Der Summand 1 steht als Betrag für das eine ein Loch kennzeichnende Attribut χ_0 und $2^0, 2^1, \dots, 2^i, \dots, 2^{c'-1}$ die mit jeder neuen „reinen“ Farbe χ_{2^i} hinzukommende Mischungsanzahl. Deutlicher wird dies bei der Formulierung der geordneten

⁶¹Gleiches gilt auch für Fermionenmischungen, etwa aus solchen der Spinquantenzahlen $\pm 1/2$ und $\pm 3/2$ ($c' = 4, c = 16, X' = \{\chi_1, \chi_2, \chi_4, \chi_8\}$), wie diese beispielsweise auf optischen Gittern realisiert werden können [9]. Technisch konzeptionell berücksichtigt lassen sich auch solche Systeme – neben bosonischen mit $S = 0$ oder $S > 0$, siehe Abschnitt 6.1.5 – mit einigen Anpassungen in den Teilalgorithmen mit diesem Rechenverfahren behandeln.

Gesamtfarbmenge:

$$X = \underbrace{\{\chi_0\}}_{\not\subseteq X'} \cup \bigcup_{i=0}^{c'-1} \left(\underbrace{\{\chi_{2^i}\}}_{\subset X'} \cup \overbrace{\{\chi_{2^i+j} \mid j = 1, 2, \dots, 2^i - 1\}}^{\text{Mischfarbe}} \right) \underbrace{\not\subseteq X'}_{\not\subseteq X'} \\ = \{\chi_0\} \cup \bigcup_{i=0}^{c'-1} \bigcup_{j=0}^{2^i-1} \{\chi_{2^i+j}\}. \quad (3.58)$$

Für das weitere Vorgehen erweist es sich nun als günstig, die Färbung

$$F({}^c\mathcal{B}_{k,l,m}) = (F(i) \mid i = 1, 2, \dots, k) \quad (3.59) \\ \text{mit } F(i) \in X = \{\chi_q \mid q = 0, 1, \dots, c-1\}$$

eines allgemein c gefärbten Butcherbaumes mit einer Trennfunktion

$$C({}^c\mathcal{B}_{k,l,m}) = \{({}_s C(i) \mid i = 1, 2, \dots, k) \mid s = 1, 2, \dots, c'\} \quad (3.60) \\ \text{mit } {}_s C(i) \in {}_s X = \{\chi_0\} \cup \{\llbracket X \rrbracket_j\} = \{\chi_0, \chi_j\}, \quad j = 2^{s-1},$$

in die Grundfarbebenen ${}_s C$ zu separieren, wodurch die Zerlegung

$${}^c\mathcal{B}_{k,l,m} = (V, E, F) \leftarrow (V, E, C) \\ = (V, E, \{{}_s C\}) = \{(V, E, {}_s C)\} \\ = \{{}_s^c\mathcal{B}_{k,l,m}\} \quad \text{mit } s = 1, 2, \dots, c' \quad (3.61)$$

in Erweiterung von Gleichung (2.7) formuliert werden kann. Beispiel⁶²:

$${}^4\mathcal{B}_{4,4,15} = \begin{array}{c} \text{V} \qquad \qquad \qquad \text{E} \qquad \qquad \qquad \text{F} \\ \text{---} \qquad \qquad \qquad \text{---} \qquad \qquad \qquad \text{---} \\ \text{---} \end{array} = \left(\{1, 2, 3, 4\}, \{\{1, 2\}, \{1, 3\}, \{1, 4\}\}, (\chi_0, \chi_3, \chi_2, \chi_1) \right) \leftarrow \\ \left(V, E, \underbrace{\{(\chi_0, \chi_1, \chi_0, \chi_1)\}}_{1C}, \underbrace{\{(\chi_0, \chi_2, \chi_2, \chi_0)\}}_{2C} \right) = \left\{ \begin{array}{c} \text{---} \\ \text{---} \end{array} \right\} = \{{}_1^4\mathcal{B}_{4,4,15}, {}_2^4\mathcal{B}_{4,4,15}\} \\ \text{(Skizze 3.1)}$$

Diese farbseparierte Darstellung erlaubt die Formulierung der Möglichkeiten für Elektronenbewegungen mit nur einer Regel, daß nämlich ein Sprung der Spinsorte χ_s vom Gitterplatz a zum Platz b aufgrund des Pauliprinzips nur dann erfolgen kann, wenn

$${}_s C(a) = \chi_s \quad \text{und} \quad (3.62-1)$$

$${}_s C(b) = \chi_0 \quad (3.62-2)$$

gilt. Besondere Eleganz wird offenkundig, wird die Propagation der Teilchen auf deren jeweiliger Farbebene verfolgt: Bei halbgefüllten Ausgangszuständen ohne Anregung

⁶²Hier beginnend wird in dieser Arbeit die Färbungsordnung umgekehrt und entspricht somit der absteigenden Ordnung der Strukturen, während die Astörung innerhalb einer gegebenen Struktur l aufsteigend bleibt. Eine Erläuterung dazu befindet sich in Fußnote 36. Hinweis: Die Färbungsnummer m ist von der Wahl der Färbungsordnung unabhängig und ändert sich folglich nicht bei deren Umkehrung.

q	0	1	2	3	4	5	6	7	8	...
$[q]_2$	0_2	1_2	10_2	11_2	100_2	101_2	110_2	111_2	1000_2	...
${}_1C$		+1/2		+1/2		+1/2		+1/2		...
${}_2C$			-1/2	-1/2			-1/2	-1/2		...
${}_3C$					+3/2	+3/2	+3/2	+3/2		...
${}_4C$									-3/2	...
\vdots										\ddots
$\chi_q \in$	X	X'	X'	X	X'	X	X	X	X'	

Tabelle 3.6. Schema zur Spin- beziehungsweise Farbebenenentrennung für Elektronen ($q = 0, 1, 2, 3$) und beliebige Fermionenmischungen, siehe Gleichung (3.58). Man beachte, daß sich die „Mischfarbzusammensetzung“ unmittelbar aus der Binärdarstellung des Farbsymbolindex q ablesen läßt, da s in ${}_sC$ für die Dualstelle in $[q]_2$ steht.

genügt es aufgrund der von der Quantenstatistik „geerbten“ Symmetriebeziehung, den ersten Sprung, bei Betrachtung des Falicov-Kimball-Modells sogar sämtliche Sprünge einer Sequenz, stets auf nur einer willkürlich gewählten Ebene auszuführen.

GUNTHER GRUBERS funktionale Studie zum Algorithmus, welche umfassend in Referenz [33] dargelegt ist, macht noch weitergehenden Gebrauch von der Ebenentrennung: Aus den farbseparierten und zunächst „entwurzelten“ zweifarbigen Butcherbäumen (Ausgangszustände; deren Anzahl ist γ_k in k ter Baumordnung) werden Produkt-Metagraphen gebildet, symbolisch $\{(V, E, {}_1C)\}_{k,g} \times \{(V, E, {}_2C)\}_{k,g}$ mit $g = 1, 2, \dots, \dots, \gamma_k$; jede Ecke darin repräsentiert einen – forderungsgemäß jedoch nur halbgefüllten – (gegebenenfalls vierfarbigen Zwischen)zustand, jede Kante einen Übergang zwischen zusammengehörigen Zuständen. Auf diesen sogenannten *Hüpfschrittgraphen* werden dann alle bei jeweils einem Ausgangszustand beginnenden und endenden Pfade der Länge $2k - 1$ gesucht, welche nach Rückübertragung auf („verwurzelte“) Butcherbäume gültigen Hüpfsequenzen entsprechen.

3.4.2. *Ausgangs- und Zwischenzustände.* Neben dem im vorausgegangenen Abschnitt explizit im Graphenformalismus berücksichtigten Paulischen Ausschlußprinzips muß im Weiteren auch der Antisymmetrie der fermionischen Wellenfunktion Rechnung getragen werden. Um Teilchenvertauschungen zur Vorzeichenbestimmung algorithmisch effizient erfassen zu können, „personalisieren“ wir die Elektronen auf der jeweiligen Farbebene

zweckmäßigerweise mit der Nummer ihres Ausgangsgitterplatzes:

$$M({}^c\mathcal{B}_{k,l,m}) = \{({}_sM(i) \mid i = 1, 2, \dots, k) \mid s = 1, 2, \dots, c'\}$$

$$\text{mit } {}_sM(i) = \begin{cases} 0 & \text{falls } {}_sC(i) = \chi_0 \\ i & \text{falls } {}_sC(i) = \chi_s \end{cases} \quad (3.63)$$

Damit läßt sich die Butcherbaumdefinition, Gleichung (3.61), mit

$${}^c\mathcal{B}_{k,l,m} = (V, E, F) \leftarrow (V, E, C) \leftarrow (V, E, M)$$

nochmals verfeinern. Dabei gilt für die Farbe der Ecke $i = 1, 2, \dots, k$ der später ab Abschnitt 3.4.9 bedeutsame Zusammenhang

$$F(V(i)) = \chi_{q_i} \quad \text{mit} \quad q_i = \sum_{s=1}^{c'} s \cdot \begin{cases} 1 & \text{falls } {}_sM(i) > 0 \text{ bzw. } {}_sC(i) = \chi_s \\ 0 & \text{sonst} \end{cases} \quad (3.64)$$

zwischen dem attributbasierten und den ebenengetrennten „Farbsystemen“. Beispiel (vergleiche Skizze 3.1):

$${}^4\mathcal{B}_{4,4,15} \leftarrow \left\{ \begin{array}{c} \text{orange} \\ \text{orange} \\ \text{orange} \\ \text{orange} \end{array} \circledast, \begin{array}{c} \text{orange} \\ \text{orange} \\ \text{orange} \\ \text{orange} \end{array} \circledast \right\} \leftarrow \left(\underbrace{\{1, 2, 3, 4\}}_V, E, \underbrace{\{(0, 2, 0, 4), (0, 2, 3, 0)\}}_{1M} \right)$$

Im Rahmen des Hubbardmodells können wir nun für ein Elektronen-beziehungweise ein beliebiges Fermionensystem auf einem Butcherbaum k ter Ordnung und l ter Struktur den vollständigen „Zustandsvektor“ als

$$|{}_1M, {}_2M\rangle_{k,l} \mapsto \vartheta({}^4\mathcal{B}_{k,l,m}) \quad \text{bzw.} \quad |{}_1M, {}_2M, \dots, {}_{c'}M\rangle_{k,l} \mapsto \vartheta({}^c\mathcal{B}_{k,l,m}) \quad (3.65)$$

schreiben, worin die Funktion ϑ einen Isochromismus, siehe Abschnitt 2.1.15 und Fußnote 34, darstellt: Ausgangszustände *sind* der Menge der gefärbten Butcherbäume angehörig, $\vartheta : F \leftrightarrow F$, Zwischenzustände *können* entweder ein gefärbter Butcherbaum oder isochrom zu einem solchen sein, $\vartheta : F \leftrightarrow F'$. Beispiel (vergleiche Skizze 3.1):

$$\begin{aligned} \text{Ausgangszustand:} & \quad |(1, 0, 0, 4), (0, 2, 3, 0)\rangle_{4,4} \mapsto \begin{array}{c} \text{orange} \\ \text{orange} \\ \text{orange} \\ \text{orange} \end{array} \circledast = {}^4\mathcal{B}_{4,4,34} \\ \text{Zwischenzustand I:} & \quad |(0, 1, 0, 4), (0, 2, 3, 0)\rangle_{4,4} \mapsto \begin{array}{c} \text{orange} \\ \text{orange} \\ \text{orange} \\ \text{orange} \end{array} \circledast = {}^4\mathcal{B}_{4,4,15} \\ \text{Zwischenzustand II:} & \quad |(0, 0, 1, 4), (0, 2, 3, 0)\rangle_{4,4} \mapsto \begin{array}{c} \text{orange} \\ \text{orange} \\ \text{orange} \\ \text{orange} \end{array} \circledast \cong {}^4\mathcal{B}_{4,4,15} \end{aligned}$$

(Skizze 3.2)

Durch die Einschränkung auf anregungsfreie halbgefüllte Ausgangszustände zwecks Berechnung der Grundzustandsenergie bedient sich die Hüpfsequenzgenerierung in k ter Ordnung einer Untermenge der viergefärbten Butcherbäume ${}^4\mathcal{B}_k \equiv \mathcal{B}_k$. Wir übernehmen die vorgenannte Farbkonvention und definieren folgende Mengen:

$B'_{k,l}$: Menge aller mit X gefärbten Butcherbäume k ter Ordnung ($k > 1 \Leftrightarrow K(1) \neq \emptyset$) und l ter Struktur $\mathcal{B}_{k,l,m}$ ohne unbesetzte und ohne doppelt besetzte Ecken. Die

Wurzel trägt die Farbe $F(1) = \chi_1$, und zumindest einer der zur Wurzel benachbarten Gitterplätze $b \in K(1)$ ist mit $F(b) = \chi_2$ gefärbt.

$$B_{k,l} \supset B'_{k,l} \ni \mathcal{B}_{k,l,m} \Leftrightarrow \forall a \in V : F(a) \notin \{\chi_0, \chi_3\} \\ \wedge \underbrace{F(1) = \chi_1}_{\text{Wurzel}} \wedge \exists b \in K(1) : F(b) = \chi_2 \quad (3.66) \\ \forall m \in \left[\frac{\mu_{k,l}}{4} + 1, \frac{\mu_{k,l}}{2} \right]$$

Wie anhand der Auswahlkriterien zu sehen, sind die Ecken sämtlicher $B'_{k,l}$ effektiv nur mit $X \supset X' = \{\chi_1, \chi_2\}$ gefärbt.

\mathbb{B}'_k : Menge aller $B'_{k,l}$ mit $l = 1, \dots, \lambda_k$. Die Gesamtzahl der Ausgangszustände, siehe Gleichung (2.10), bis einschließlich $k = 10$ (Störungsrechnungsordnung U^{-17}) ist in Tabelle 3.7 aufgeführt. Durch die Wahl einer Wurzelfarbe aus zwei Möglichkeiten handelt es sich jeweils um die Hälfte der Zahlenfolgenwerte in Referenz [24:AI98760]. Es gilt $b'_k = \|\mathbb{B}'_k\|$.

k	1	2	3	4	5	6	7	8	9	10
b'_k		1	4	16	68	298	1356	6321	30117	145920
${}^2b_k/2$	1	2	7	26	107	458	2058	9498	44947	216598
q		0,5	0,571	0,615	0,636	0,651	0,659	0,666	0,670	0,674

Tabelle 3.7. Gesamtzahl der Ausgangszustände bezogen auf eine der beiden möglichen Wurzelfarben, vergleiche Tabelle 1.3. Zum Vergleich ist zusätzlich entsprechend die Hälfte der Gesamtzahl ${}^2b_k = \|\mathbb{B}_k\| = \sum_{l=1}^{\lambda_k} {}^2\mu_{k,l}$ zweifarbigter Butcherbäume, siehe auch Tabelle 2.3 und [24:A038055], sowie das gerundete Verhältnis $q = 2b'_k/{}^2b_k$ angegeben.

3.4.3. *Hüpfkantenmengen.* Unabhängig von den Ausgangszuständen, also den relevanten gefärbten Butcherbäumen des zu lösenden Problems, werden folgende Symbole und Begriffe im Weiteren benötigt:

j : Nummer des Hüpfes, $j = 1, 2, \dots, 2k - 2$. Wird insbesondere ein (zumeist mit einem Subskript versehener) Zustand nach einem bestimmten Hüpfes referenziert, so wird die Hüpfesnummer, eingeschlossen in spitze Klammern, hochgestellt notiert. Damit besteht eine eindeutige Abgrenzung vom Hinweis auf die Graphen- (k) beziehungsweise Störungsrechnungsordnung (n) eines Objektes, deren Angabe zwar ebenfalls hochgestellt, jedoch in runden Klammer erfolgt.

G : Hüpfkantenmenge, bestehend aus der Menge aller Kanten E (Hinsprung) eines Butcherbaumes einschließlich derer formalen „Gegenkanten“ \bar{E} (Rücksprung). Die folgende, eine *Richtung* implizierende Definition setzt voraus, daß eine Kante im

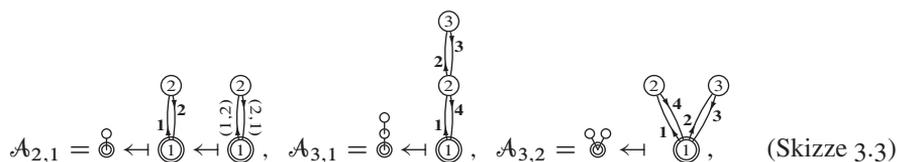
Gegensatz zu Abschnitt 2.1.1 im Weiteren als *Paar*; also ein (geordnetes) 2-Tupel behandelt wird, siehe auch Fußnoten 12 und 19.

$$G = E \cup \bar{E} = E \cup \bigcup_{u=1}^{k-1} \llbracket \bar{E} \rrbracket_{k-u} \quad \text{mit} \quad (3.67)$$

$$\llbracket \bar{E} \rrbracket_v = \overline{\llbracket E \rrbracket_v} = \overline{(a_v, b_v)} = (b_v, a_v) \quad \text{für } v = 1, 2, \dots, k-1 \quad (3.68)$$

Der erste Sprung muß voraussetzungsgemäß über $(1, b)$ mit $b \in K(1)$ erfolgen; alle weiteren $|G| - 1 = 2k - 3$ Sprünge sind wahlfrei. Ohne Berücksichtigung der färbungsbedingten Ausführbarkeit ergeben sich höchstens $|K(1)| \cdot (2k - 3)!$ Möglichkeiten.

${}^mG, {}^mE$: Indexmengen der Kantenmengen G und E , auch Meta-Kantenmengen genannt, die für alle (ungefärbten) Butcherbäume gleicher Ordnung k durch ${}^mG = \{j \mid j = 1, 2, \dots, 2(k-1)\}$ beziehungsweise ${}^mE = \{j' \mid j' = 1, 2, \dots, k-1\}$ und folglich $|{}^mG| = |G|$ und $|{}^mE| = |E|$ definiert sind. Im Folgenden ist insbesondere die Meta-Hüpfkantenmenge mG bedeutsam, deren Elemente durch Permutation eine *festgelegte Reihenfolge* erhalten. Beispiel: Sämtliche ungefärbte Butcherbäume zweiter und dritter Ordnung:



worin, bedingt durch die definitionsgemäß kraft eines links beginnenden Tiefendurchlaufs geordnet angelegte Eckenmengen E und obige Definition von G ,

$${}^mG(\mathcal{A}_{2,1}) = \{1, 2\} \leftarrow G(\mathcal{A}_{2,1}) = \{(1, 2), (2, 1)\},$$

$${}^mG(\mathcal{A}_{3,1}) = \{1, 2, 3, 4\} \leftarrow G(\mathcal{A}_{3,1}) = \{(1, 2), (2, 3), (3, 2), (2, 1)\},$$

$${}^mG(\mathcal{A}_{3,2}) = \{1, 2, 3, 4\} \leftarrow G(\mathcal{A}_{3,2}) = \{(1, 2), (1, 3), (3, 1), (2, 1)\}$$

Für den Baum $\mathcal{A}_{2,1}$ gilt somit beispielsweise $\llbracket {}^mG \rrbracket_1 \leftarrow (1, 2)$ und $\llbracket {}^mG \rrbracket_2 \leftarrow (2, 1)$; die Indexmengen können datentechnisch somit als *Adreßlisten* aufgefaßt werden.

$r^{(j)}$: Nummer der im j ten Hüpfschritt übersprungenen Kante. Für $j = 1$ gilt $r^{(j)} = r' \in {}^mE$, wie im Folgenden erläutert wird.

$s^{(j)}$: Nummer der im j ten Hüpfschritt gesprungenen Farbe. Für $j = 1$ gilt kraft der Spinsymmetrie $s^{(j)} = s' \in \{1\}$, sonst $s^{(j)} \in \{1, 2\}$.

3.4.4. *Hüpf Schritte*. Der Erzeuger-Vernichter-Formalismus für die Elektronenpropagation, siehe Abschnitt 1.1 und folgende (dort mit den benachbarten Gitterplätzen (u, v) und $\sigma = \uparrow, \downarrow$), läßt sich nun auf Grundlage der (Meta-)Hüpfkantenmenge in eine für die

Rechentechnik praktikable Form übertragen⁶³:

$$\underbrace{\hat{c}_{v,\sigma}^+ \hat{c}_{u,\sigma}}_{\text{Gl. (1.1) ff.}} \longrightarrow \check{c}_{b,s}^+ \check{c}_{a,s} \quad \text{mit } (a, b) \in G \text{ und } s \in \{1, 2\}$$

$$= \check{T}_{r,s} \quad \text{mit } r \in {}^m G \leftarrow (a, b) = \llbracket G \rrbracket_r . \quad (3.69)$$

Sei ${}_s m_b$ die Nummer des Elektrons mit der „Spinfarbe“ s am Gitterplatz b , dann kann – sofern der Gitterplatz a nicht von einem Elektron gleichen Spins belegt ist – die Wirkung von $\check{c}_{a,s}^+ \check{c}_{b,s}$ auf einen Zustand k ter Baumordnung und l ter Struktur,

$$|{}_s M \mid s = 1, 2\rangle_{k,l} = |({}_s m_i \mid i = 1, 2, \dots, k) \mid s = 1, 2\rangle_{k,l} , \quad (3.70)$$

wie folgt formuliert werden, wobei die Reihenfolge von ${}_s m_a$ und ${}_s m_b$ in ${}_s M$ beliebig ist:

$${}_s m_a \neq 0 \wedge {}_s m_b = 0 \Rightarrow \check{c}_{b,s}^+ \check{c}_{a,s} |{}_s M, {}_{s'} M\rangle_{k,l} :=$$

$$|({}_s m_1, {}_s m_2, \dots, {}_s m_a := 0, \dots, {}_s m_b := {}_s m_a, \dots, {}_s m_k), {}_{s'} M\rangle_{k,l} . \quad (3.71)$$

Anders gesagt „liest“ und „löscht“ in dieser Zuweisungsabfolge $\check{c}_{a,s}$ die Nummer des auf Platz a sitzenden Elektrons, während $\check{c}_{b,s}^+$ diese Nummer auf den Platz b „schreibt“. Beispiel (vergleiche Skizze 3.2):

$$\text{Zwischenzustand I: } \check{c}_{2,1}^+ \check{c}_{1,1} | (1, 0, 0, 4), (0, 2, 3, 0) \rangle_{4,4} := | (0, 1, 0, 4), (0, 2, 3, 0) \rangle_{4,4}$$

$$\text{Zwischenzustand II: } \check{c}_{3,1}^+ \check{c}_{1,1} | (1, 0, 0, 4), (0, 2, 3, 0) \rangle_{4,4} := | (0, 0, 1, 4), (0, 2, 3, 0) \rangle_{4,4}$$

(Skizze 3.4)

Um hier passend einen kurzen Ausblick auf die konkrete Umsetzung in Binärdarstellung geben zu können, müssen wir zunächst eine Notation⁶⁴ der *bit*-weisen Logikoperatoren, welche sich aus den Booleschen Schaltfunktionen ergeben [39], vereinfacht an einem alle Fälle umfassenden Beispiel einführen:

$$\text{not}(01_2) = 10_2 , \quad (3.72)$$

$$\text{and}(0101_2, 0011_2) = 0001_2 , \quad (3.73)$$

$$\text{or}(0101_2, 0011_2) = 0111_2 . \quad (3.74)$$

Die „einstellige“ *Nicht*-Operation, Gleichung (3.72), invertiert den Operanden, schaltet also jede Null zu einer Eins und umgekehrt. Die „zweistellige“ *Und*-Verknüpfung, Gleichung (3.73), schaltet im Ergebnis eine Eins, sofern die beiden Dualziffern an gleicher Stelle in den Operanden ebenfalls Einsen sind; anderenfalls ist das Ergebnis Null. Die ebenso „zweistellige“ *Oder*-Verknüpfung, Gleichung (3.74), schaltet im Ergebnis eine

⁶³Man beachte, daß es sich bei dieser Formulierung nur um eine symbolhafte Übertragung der quantenmechanischen Operanden und Operatoren handelt, die nicht der zugehörigen Algebra genügt. So muß insbesondere die Bestimmung der resultierenden Zustandsvorzeichen durch eine separate Berechnung erfolgen, wie später gezeigt wird.

⁶⁴Um keine weiteren Symbole definieren zu müssen, bedienen wir uns dabei der Präfix-Schreibweise und verwenden Abkürzungen, wie diese in der Assembler-Programmierung üblich sind.

Eins, sofern eine der beiden oder beide Dualziffern an gleicher Stelle in den Operanden eine Eins ist; sonst ist das Ergebnis Null.

Zudem definieren wir auch die bereits in Abschnitt 2.2.7 erwähnten *bit-shift*-Funktionen als „zweistellige“ Operationen mit $\alpha, \beta \in \mathbb{N}$, welche wie alle obengenannten auf den genutzten Rechnerarchitekturen direkt als *Maschinenbefehle*⁶⁵ vorliegen:

$$\text{shl}(\alpha, \beta) := \alpha \cdot 2^\beta, \quad (3.75)$$

$$\text{shr}(\alpha, \beta) := \alpha \cdot 2^{-\beta}; \quad (3.76)$$

der *bit shift* nach links, gemäß Gleichung (3.75) eine Multiplikation, verschiebt also die Ziffern der Ganzzahl α in Dualdarstellung um β Positionen in die linke Richtung, der *bit shift* nach rechts, nach Gleichung (3.76) ein (Ganzzahl)division, wirkt umgekehrt.

Im Gegensatz zu den Struktur und Färbung betreffenden Beispielen im Abschnitt 2.2.3 kodieren wir die Teilchennummern ${}_s m$ in der Binärdarstellung der Reihenfolge ihrer Ausgangsgitterplätze entsprechend von rechts nach links in der zur Farbebene s gehörigen Dualzahl ${}_s M]_2 \in \mathbb{N}$. Für höchstens k Elektronen einer Spinsorte auf k Gitterplätzen⁶⁶ werden für eine Teilchennummer ein *Block* aus $w(k) = \lceil \log_2(k+1) \rceil$ Dualstellen benötigt. Da, wie bereits ausgeführt, die Gitterplätze mit $i = 1, 2, \dots, k$ numeriert sind und folglich für alle Teilchen ${}_s m_i > 0$ gilt, kann ein freier Gitterplatz durch eine „ $w(k)$ stellige“ Null im *bit*-Muster repräsentiert werden⁶⁷; somit erhalten wir unter Rückgriff auf Gleichung (3.70) für jede der beiden Spinspezies $s = 1, 2$

$${}_s M]_2 = \sum_{i=1}^k \text{shl}({}_s m_i]_2, i w(k)) = \sum_{i=1}^k \text{shl}(\llbracket {}_s M \rrbracket_i]_2, i w(k)). \quad (3.77)$$

Durch diese Definition besitzt ${}_s M]_2$ stets die Länge $w'(k) = (k+1)w(k)$ und entstammt dem Wertebereich $[0, 2^{w'(k)} - 1]$ ⁶⁸; der erste Block rechts, formal den „nullten“ Gitterplatz repräsentierend, ist mit Nullen belegt und bleibt ungenutzt⁶⁹.

⁶⁵Tatsächlich existieren in den Befehlssätzen üblicherweise mehrere Varianten von *bit shifts*, welche auf hier nicht belangvolle Aspekte wie die durch die endlichen Wertebereiche der Datentypen bedingten unterschiedlichen Überlaufbehandlungen (vor allem bei Linksverschiebungen) Rücksicht nehmen. [46]

⁶⁶Hier sind bereits Anwendungen jenseits der Gitterhalbfüllung berücksichtigt.

⁶⁷Wie unmittelbar einsichtig bleibt die Eindeutigkeit der Abbildung des Tupels ${}_s M$ auf eine (binäre) Ganzzahl davon unberührt.

⁶⁸Dieses Intervall ist „löcherig“, da forderungsbedingt niemals zwei von null verschiedene Blöcke den gleichen Zahlenwert haben, das heißt die gleiche Teilchennummer enthalten.

⁶⁹Dieser *off set* ließe sich nur durch erhöhten Rechenaufwand bei der Blockadressierung im Verlauf der Teilchenverschiebungen verhindern, was aber hinsichtlich des Gesamt Speicherbedarfs durch die Maschinentyp-gegebene Registerbreite keine Vorteile brächte, siehe Fußnote 71.

Mit den Abkürzungen $a' = aw(k)$ und $b' = bw(k)$ sowie ${}_sZ]_2$ für das Zwischen- und ${}_s\tilde{M}]_2$ für das Endergebnis können wir einen Hüpfschritt in Binärform, der Übersichtlichkeit halber in zwei Teile zerlegt, folgendermaßen schreiben:

$$\text{Erzeugung: } {}_sZ]_2 = \text{or} \left({}_sM]_2, \text{shl} \left(\text{and} \left(\text{shr}({}_sM]_2, a'), \gamma_{(1)}^{\text{pos.}}, b' \right) \right) \right), \quad (3.78-1)$$

$$\text{Vernichtung: } {}_s\tilde{M}]_2 = \text{and} \left({}_sZ]_2, \gamma_{(1+a)}^{\text{neg.}} \right). \quad (3.78-2)$$

Die darin auftretenden „Masken“

$$\gamma_{(1)}^{\text{pos.}} = 2^{w(k)} - 1 \quad \text{und} \quad (3.79)$$

$$\gamma_{(1+a)}^{\text{neg.}} = \text{not} \left(\text{shl}(2^{w(k)} - 1, a') \right) \quad (3.80)$$

bewirken mit der Und-Operation folgendes (die Blocknummer steht eingeklammert im Subskriptum): Mit der Dualzahl aus Gleichung (3.79) wird aus einem *bit*-Muster der erste, am rechten Rand ausgerichtete Block „ausgestanzt“, also alle Dualstellen mit einer Position größer als $w(k)$ auf Null gesetzt; mit der Dualzahl aus Gleichung (3.80) hingegen wird aus einem *bit*-Muster der rechts an Position $a' + 1 = aw(k) + 1$ beginnende Block „gelöscht“, also alle Dualstellen innerhalb Blockes auf Null gesetzt.⁷⁰ Man beachte, daß insbesondere $\gamma_{(1+a)}^{\text{neg.}}$ aus $w'(k)$ Dualstellen mit $kw(k)$ Einsen und $w(k)$ Nullen besteht, damit die Maske über die volle *Datenwortbreite*⁷¹ von ${}_sM]_2$ respektive ${}_sZ]_2$ und ${}_s\tilde{M}]_2$ wirkt.

Verdeutlichen wir diese maschinennahe Vorgehensweise an einem Beispiel⁷², das die gesamte HüpfschrittAusführung umfaßt, und gehen dabei vom Zustand

$${}^4\mathcal{B}_{3,1,26} = \begin{array}{c} \circ \\ \circ \\ \circ \\ \circ \\ \circ \end{array} \leftarrow |(1, 0, 3), (0, 2, 0)\rangle_{3,1}$$

aus:

$$\check{c}_{2,1}^+ \check{c}_{3,1} |(1, 0, 3), (0, 2, 0)\rangle_{3,1} := |(1, 3, 0), (0, 2, 0)\rangle_{3,1} \mapsto \begin{array}{c} \circ \\ \circ \\ \circ \\ \circ \\ \circ \end{array} = {}^4\mathcal{B}_{3,1,29}. \quad (\text{Skizze 3.5})$$

Bei $w(3) = 2$ Dualstellen pro Gitterplatz und $w'(3) = 8$ Dualstellen gesamter Wortbreite wird die fragliche Farbebene $s = 1$ nach Gleichung (3.77) als ${}_1M]_2 = 11000100_2$

⁷⁰Weil diese Masken in der späteren Anwendung von der Färbung des Butcherbaumes, also vom Besetzungszustand des Bethegitterausschnittes unabhängig ist und für die gesamte Rechnung der Graphenordnung k gilt, empfiehlt sich aus Leistungsgründen vor allem die Tabellierung von $\gamma_{(1+a)}^{\text{neg.}}$ für $a = 1, 2, \dots, k$, sofern der Wertzugriff schneller als dessen Berechnung erfolgen kann, siehe Fußnote 128.

⁷¹Die Datenwort- beziehungsweise Registerbreite für (vorzeichenlose) Ganzzahlen beträgt auf Universalprozessoren seit den späten 70er Jahren (CRAY-1 [47]) gleichbleibend 64 Dualstellen (64 Bit); damit ergibt sich eine maximale in *hardware* behandelbare Problemgröße von $k = 15$ Gitterplätzen, welche jeweils $w(15) = 4$ Dualstellen beanspruchen (insgesamt 60 Bit zuzüglich 4 Bit *off set*). Auf technische Details der genutzten Datenverarbeitungsanlagen wird umfassend in Abschnitt 5.1.3 eingegangen.

⁷²Der hier erläuterte Hüpfschritt tritt tatsächlich in der Störungsrechnung dritter Ordnung auf und ist in der in Gleichung (3.103-4) dargestellten Hüpfsequenz an dritter Stelle zu finden, siehe Abschnitt 3.5.2.

kodiert, für die unbeteiligte ergibt sich dementsprechend ${}_2M]_2 = 00100000_2$. Vorgegeben sind des weiteren $a = 3$ und $b = 2$, folglich $a' = 6$ und $b' = 4$; zudem ergibt sich die „Positiv-Maske“ zu

$$\gamma_{(1)}^{\text{pos.}} = 00000011_2$$

und die „Negativ-Maske“ zu

$$\gamma_{(4)}^{\text{neg.}} = \text{not} \left(\underbrace{\text{shl}(00000011_2, 6)}_{=11000000_2} \right) = 00111111_2 .$$

Der Erzeugungsschritt, Gleichung (3.78-1), und Vernichtungsschritt, Gleichung (3.78-2) nehmen nun im konkreten Fall folgendes Aussehen an:

$$\begin{aligned} {}_1Z]_2 &= \text{or} \left(11000100_2, \text{shl} \left(\underbrace{\text{and} \left(\underbrace{\text{shr}(11000100_2, 6), 00000011_2}_{=00000011_2} \right), 4 \right)}_{=00000011_2} \right) \right) \\ &= \text{or} \left(11000100_2, \underbrace{\text{shl}(00000011_2, 4)}_{=00110000_2} \right) = 11110100_2 , \\ {}_1\tilde{M}]_2 &= \text{and}(11110100_2, 00111111_2) = 00110100_2 . \end{aligned}$$

Die Rückkonvertierung des Ergebnisses in die Tupeldarstellung der Farbebene, quasi die Umkehrung von Gleichung (3.77), läßt sich beispielsweise durch

$${}_sM = \left(\text{and} \left(\text{shr}({}_sM]_2, iw(k)), \gamma_{(1)}^{\text{pos.}} \right) \mid i = 1, 2, \dots, k \right) \quad (3.81)$$

beschreiben. Für $[[{}_1\tilde{M}]_i]$ aus obigem Beispiel ergibt sich für $i = 1$, also die Nummer des Teilchens auf dem ersten Gitterplatz, definitionsgemäß der Wurzel, folglich

$$[[{}_1\tilde{M}]_1]_1 = \text{and} \left(\underbrace{\text{shr}(00110100_2, 2)}_{=00001101_2}, 00000011_2 \right) = 00000001_2 = 1 ;$$

in Übereinstimmung mit Skizze 3.5 lautet das vollständige Tupel ${}_1\tilde{M} = (1, 3, 0)$.

3.4.5. *Kantenklassen und Metagraphen.* Dem Kern des Graphenrechenverfahrens liegen zwei kombinatorische Fragen zugrunde: Wieviele unterschiedliche Möglichkeiten existieren für die Elektronenpropagation auf dem jeweiligen Bethegitterausschnitt (Butcherbaum), und wie oft wird eine jede davon realisiert. Durch das erstgenannte Problem

erlangen zwei Arten von Metagraphen im Algorithmus tragende Bedeutung, nämlich Permutationsbäume und Binärbäume, deren Ecken mit Attributen versehen sind.

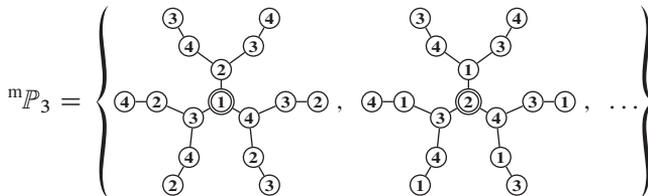
Im Falle der zunächst eingeführten Permutationsbäume ist mit jedem Knoten die Nummer einer Sprungkante auf den der jeweiligen Ordnung zugrundeliegenden Butcherbäumen assoziiert; auf den später erläuterten Binärbäumen ist jeder Knoten mit dem farb-kodierten Spin des jeweils springenden Elektrons belegt. Einer jeden Hüpfsequenz liegt ein Weg auf den Metagraphen von der Wurzel zu einem Blatt zugrunde, wie in Abschnitt 3.4.6 vertiefend behandelt wird. Da aber durch den aktuellen Besetzungszustand sowie das Pauliprinzip nicht alle Wege ausgeführt werden können, sind sowohl die Permutationsbäume als auch die Binärbäume durch „ausbrechbare“ Äste *unvollständig*, wie später erläutert.

${}^m\mathcal{P}_{2k-3,r}$: Vollständiger „Meta-Permutationsbaum“ zur Meta-Hüpfkantenmenge mG der Butcherbaumordnung $k \geq 2$ mit dem Wurzelattribut $r = 1, 2, \dots, 2k - 2$, also $r \in {}^mG$, von welchem nur der Wertebereich $1 \leq r' \leq k - 1$, folglich $r' \in {}^mE$, im Weiteren Bedeutung hat. Beispiel: Sämtliche Meta-Permutationsbäume für $k = 2$:

$${}^m\mathcal{P}_{1,1} = \begin{array}{c} \textcircled{2} \\ | \\ \textcircled{1} \end{array}, \quad {}^m\mathcal{P}_{1,2} = \begin{array}{c} \textcircled{1} \\ | \\ \textcircled{2} \end{array}$$

Da im Verlauf einer Hüpfsequenz auf dem betrachteten, den Ausgangszustand repräsentierenden Butcherbaum der Meta-Permutationsbaum stets von der Wurzel zu einem Blatt *in dessen Tiefe durchlaufen* wird, notieren wir die sich ergebenden Abfolge der Eckenattribute als Menge; diese stellt dementsprechend eine Permutation der Meta-Hüpfkantenmenge dar, beispielsweise ${}^m\mathcal{P}_{1,2} = \{2, 1\}$.

${}^m\mathcal{P}_{2k-3}$: Geordnete Menge aller vollständigen Meta-Permutationsbäume zur Meta-Hüpfkantenmenge mG der Butcherbaumordnung $k \geq 2$, ${}^m\mathcal{P}_{2k-3} = \{ {}^m\mathcal{P}_{2k-3,r} \mid r = 1, 2, \dots, 2k - 2 \}$ mit $|{}^m\mathcal{P}_{2k-3}| = 2k - 2$; formal gilt $[{}^m\mathcal{P}_{2k-3}]_r = {}^m\mathcal{P}_{2k-3,r}$. Beispiel: Menge der Meta-Permutationsbäume für $k = 3$ (gezeichnet sind nur die später relevanten Bäume mit $r' = 1$ und $r' = 2$):



In Mengennotation der Eckenattribute läßt sich dieses Beispiel auch folgendermaßen schreiben, indem diese geordnet entlang der Pfade von der Wurzel zu den Blättern – hier im Falle von Kantennummern als geordnete Menge – „protokolliert“ werden (konventionsgemäß mit den niedrigsten Knotenattributen beginnend, dann

entgegen dem Uhrzeigersinn)⁷³:

$$\mathbb{F}({}^m\mathcal{P}_3) = \left\{ \left\{ \{1, 2, 3, 4\}, \{1, 2, 4, 3\}, \{1, 3, 2, 4\}, \{1, 3, 4, 2\}, \{1, 4, 2, 3\}, \{1, 4, 3, 2\} \right\}, \right. \\ \left. \left\{ \{2, 1, 3, 4\}, \{2, 1, 4, 3\}, \{2, 3, 1, 4\}, \{2, 3, 4, 1\}, \{2, 4, 1, 3\}, \{2, 4, 3, 1\} \right\}, \dots \right\}$$

\dot{E} : Menge aller sogenannten Sequenzinitialisierungskanten eines Baumes aus $B'_{k,l}$, welche die Wurzel mit einem Nachbargitterplatz der Farbe χ_2 verbinden. Der erste Sprung kann nur, bedingt durch das Pauliprinzip, über eine solche Kante erfolgen.

$$\forall \mathcal{B}_{k,l,m} \in B'_{k,l} : \dot{E} = \{(1, b) \mid \forall b \in K(1) \Leftrightarrow F(b) = \chi_2\} \quad (3.82)$$

Es gilt $|\dot{E}| \leq |K(1)|$. Mit Berücksichtigung der Anzahl tatsächlich ausführbarer Erstsprünge ergeben sich höchstens $|\dot{E}| \cdot (2k - 3)!$ mögliche Sprungkombinationen, da das Wurzelattribut der Meta-Permutationsbäume nur \dot{E} entnommen werden kann.

$\dot{E}_{\text{CNL}}, E_{\text{CNL}}$: Nach topologischer Äquivalenz in Klassen aufgeteilte Kantenmengen E und \dot{E} eines gefärbten Butcherbaumes; es gilt $\dot{E}_{\text{CNL}} \subseteq E_{\text{CNL}}$. Um bei der Sequenzierung der Ausgangszustände keine „gültigen“, also einen Beitrag zur Grundzustandsenergie leistenden Sprungpermutationen zu übergehen, muß die Klasseneinteilung bezüglich aller Ecken lokal (*classes—node-local*, CNL) erfolgen, beispielsweise rekursiv. Betrachtet werden dabei alle von der jeweiligen Ecke ausgehenden Kanten in Richtung der Blätter: Sei $\tilde{Z}(i)$ die (voraussetzungsgemäß geordnete) Menge der von der Ecke $i \in V$ abzweigenden Äste a_1, a_2, \dots, a_q , dann gilt mit $a_1 = a_2 = \dots = a_{g_1} \neq a_{g_1+1} = a_{g_1+2} = \dots$ und

$$\tilde{Z} = \tilde{Z}_1 \cup \tilde{Z}_2 \cup \dots \cup \tilde{Z}_u \\ = \left\{ \underbrace{a_1, a_2, \dots, a_{g_1}}_{\tilde{Z}_1}, \underbrace{a_{g_1+1}, a_{g_1+2}, \dots, a_{g_2}}_{\tilde{Z}_2}, \dots, \underbrace{a_{g_{u-1}+1}, a_{g_{u-1}+2}, \dots, a_{g_u}}_{\tilde{Z}_u} \right\}$$

sowie der Wurzecke r des Astes a' folgende Festlegung:

$$E_{\text{CNL}} = \bigcup_{i \in V} \left\{ \{(i, r(a')) \mid \forall a' \in \tilde{Z}_v(i)\} \mid v = 1, 2, \dots, u(i) \right\}. \quad (3.83)$$

Damit weicht die hiesige Kantenklassendefinition von der allgemeinen ab, welche bezüglich der Wurzel baumweit (*classes—tree-wide*, CTW) alle Kanten erfaßt⁷⁴.

⁷³Die „Funktionswirkung“ der Menge \mathbb{F} besteht darin, geordnet die Attributmengen F sämtlicher von der Wurzel zu den Blättern verlaufender Wege $\mathcal{J} = (V, E, F)$ auf ${}^m\mathcal{P}_{2k-3}$ ($|V| = 2k - 2$, entsprechend der Anzahl der Hüpf Schritte auf dem Butcherbaum) zu „extrahieren“.

⁷⁴Da die Sequenzinitialisierungskanten ausschließlich die Wurzel mit nächsten Nachbarn verbinden, gilt allerdings $\dot{E}_{\text{CNL}} = \dot{E}_{\text{CTW}}$. Der Zusammenhang beider Definitionen wird eingehender in Abschnitt 3.4.9 behandelt.

Beispiele: Butcherbaum vierter Ordnung (trotz Zweifärbung bezieht sich die Färbungsnummer auf $c = 4$) mit $|E_{\text{CNL}}| = 2$ und $|\dot{E}_{\text{CNL}}| = 1$:

$$\mathcal{B}_{4,4,34} = \begin{array}{c} \bullet \bullet \bullet \\ \bullet \\ \bullet \end{array}, \quad (\text{Skizze 3.6})$$

worin (die Ecken werden definitionsgemäß durch einen links beginnenden Tiefendurchlauf numeriert)

$$E_{\text{CNL}} = \left\{ \{(1, 2), (1, 3)\}, \{(1, 4)\} \right\} \quad \text{aus} \quad E = \{(1, 2), (1, 3), (1, 4)\} \quad \text{und} \\ \dot{E}_{\text{CNL}} = \left\{ \{(1, 2), (1, 3)\} \right\} \quad \text{aus} \quad \dot{E} = \{(1, 2), (1, 3)\}.$$

Butcherbaum sechster Ordnung (Färbungsnummer ebenfalls auf $c = 4$ bezogen) mit $|E_{\text{CNL}}| = 4$ und $|\dot{E}_{\text{CNL}}| = 1$:

$$\mathcal{B}_{6,18,871} = \begin{array}{c} \bullet \bullet \bullet \\ \bullet \bullet \bullet \\ \bullet \end{array},$$

worin

$$E_{\text{CNL}} = \left\{ \{(1, 2)\}, \{(1, 3), (1, 5)\}, \{(3, 4)\}, \{(5, 6)\} \right\} \quad \text{und} \quad \dot{E}_{\text{CNL}} = \left\{ \{(1, 2)\} \right\}.$$

Man beachte, daß bei der zugrundeliegenden Baumstruktur $\mathcal{A}_{6,18}$, ebenso wie erstmalig bei $\mathcal{A}_{5,7}$ (siehe Abbildung 3.2), die eckenbezogene nicht mehr mit der baumweiten Kantenklassifikation,

$$E_{\text{CTW}} = \left\{ \{(1, 2)\}, \{(1, 3), (1, 5)\}, \{(3, 4), (5, 6)\} \right\},$$

übereinstimmt: Aus Sicht der Wurzel sind die Kanten (3, 4) und (5, 6) topologisch gleich; somit ergibt sich $|E_{\text{CTW}}| = 3$.

${}^m\dot{E}_{\text{CNL}}, {}^mE_{\text{CNL}}$: Den Kantenklassenmengen \dot{E}_{CNL} und E_{CNL} zugehörige Indexmengen, auch als Meta-Kantenklassenmengen bezeichnet; es gilt ${}^m\dot{E}_{\text{CNL}} \subseteq {}^mE_{\text{CNL}}$. Beispiel: Butcherbaum der Ordnung $k = 6$ (trotz Zweifärbung bezieht sich die Färbungsnummer auf $c = 4$) mit $|{}^mE_{\text{CNL}}| = |E_{\text{CNL}}| = 4$ und $|{}^m\dot{E}_{\text{CNL}}| = |\dot{E}_{\text{CNL}}| = 2$:

$$\mathcal{B}_{6,19,514} = \begin{array}{c} \bullet \bullet \bullet \\ \bullet \bullet \bullet \\ \bullet \end{array},$$

mit (zur Kantennumerierung siehe Skizze 3.3)

$${}^mE_{\text{CNL}} = \left\{ \{1, 2\}, \{3\}, \{4\}, \{5\} \right\} \quad \text{aus} \\ E_{\text{CNL}} = \left\{ \{(1, 2), (1, 3)\}, \{(1, 4)\}, \{(1, 5)\}, \{(5, 6)\} \right\},$$

woraus sich ${}^m\dot{E}_{\text{CNL}} = \left\{ \{1, 2\}, \{4\} \right\}$ durch obige Definition von \dot{E} ableiten läßt

Beim Generieren aller Permutationen ${}^m\tilde{G}$ der Meta-Hüpfkantenmenge mG , deren Gesamtmenge ${}^mG = \{{}^m\tilde{G}_0, {}^m\tilde{G}_1, \dots\}$ mit ${}^m\tilde{G}_0 = {}^mG$ sei, sind solche zu übergehen, bei welchen zwei oder mehr der selben Klasse $\llbracket {}^mE_{\text{CNL}} \rrbracket_i$ ($i = 1, 2, \dots, |{}^mE_{\text{CNL}}|$)

angehörigen Kanten vertauscht sind. Das gilt insbesondere beim Aufbau der Meta-Permutationsbäume ${}^m\mathcal{P}'_{2k-3,r'} \in {}^m\mathcal{P}'_{2k-3}$ (siehe unten), für deren Wurzelattribut r' entsprechend nur *ein Repräsentant einer jeden Klasse* der Sequenzinitialisierungskanten gewählt werden darf, in dieser Arbeit konventionsmäßig der erste, $[[{}^m\dot{E}_{\text{CNL}}]]_{i,1} = [{}^m\dot{E}_{\text{CNL}}]_{i,1}$ mit $i = 1, 2, \dots, |{}^m\dot{E}_{\text{CNL}}|$. Somit reduziert sich die Anzahl möglicher Sprungkombinationen nochmals, nämlich mit ${}^mE'_{\text{CNL}} = {}^mE_{\text{CNL}} \setminus {}^m\dot{E}_{\text{CNL}}$ und $e' = |{}^mE'_{\text{CNL}}|$ auf höchstens

$$|{}^m\mathcal{G}| = |{}^m\dot{E}_{\text{CNL}}| (2k-3)! \left(\prod_{i=1}^{e'} |[[{}^mE'_{\text{CNL}}]]_i|! \right)^{-1}. \quad (3.84)$$

Im obigen Beispiel des Baumes $\mathcal{B}_{6,19,514}$ verbleiben daher mit $[[{}^m\dot{E}_{\text{CNL}}]]_{1,1} = r' = 1$ und $[[{}^m\dot{E}_{\text{CNL}}]]_{2,1} = r' = 4$ nur die (in diesem Falle vollständigen) Permutationsbäume ${}^m\mathcal{P}_{9,1}$ und ${}^m\mathcal{P}_{9,4}$.

${}^m\ddot{E} = {}^m\ddot{E}_{\text{CNL}}$: Menge aller „repräsentativen Vertreter“ aus der Menge der klassifizierten Meta-Sequenzinitialisierungskanten:

$${}^m\ddot{E}_{\text{CNL}} = \{[[{}^m\dot{E}_{\text{CNL}}]]_{i,1} \mid i = 1, 2, \dots, |{}^m\dot{E}_{\text{CNL}}|\}. \quad (3.85)$$

Insbesondere gilt $|{}^m\ddot{E}_{\text{CNL}}| = |{}^m\dot{E}_{\text{CNL}}|$. Da in dieser Arbeit dadurch keine Uneindeutigkeiten auftreten, entfällt im Folgenden der Subskript CNL. Beispiel: Butcherbaum $\mathcal{B}_{6,19,514}$:

$${}^m\ddot{E} = \{1, 4\} \quad \text{aus} \quad {}^m\dot{E}_{\text{CNL}} = \{\{1, 2\}, \{4\}\}$$

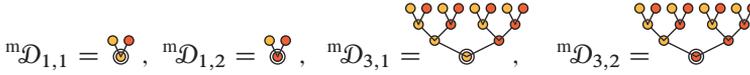
${}^m\mathcal{P}'_{2k-3}$: Geordnete Menge aller vollständigen Meta-Permutationsbäume zur Meta-Hüpfkantenmenge ${}^m\mathcal{G}$ und der (struktur- und färbungsspezifischen) Meta-Sequenzinitialisierungskantenklassenmenge ${}^m\ddot{E}$ eines Baumes der Ordnung $k \geq 2$; es gilt ${}^m\mathcal{P}'_{2k-3} \supseteq {}^m\mathcal{P}_{2k-3} = \{{}^m\mathcal{P}_{2k-3,r'} \mid r' \in {}^m\ddot{E} \text{ und entsprechend } |{}^m\mathcal{P}'_{2k-3}| = |{}^m\ddot{E}|\}$. Beispiel (siehe oben): $\mathcal{B}_{6,19,514} \Rightarrow {}^m\mathcal{P}'_9 = \{{}^m\mathcal{P}_{9,1}, {}^m\mathcal{P}_{9,4}\}$

${}^m\mathcal{D}_{2k-3,s}$: Zweifarbigem vollständiger „Meta-Binärbaum“ zur Butcherbaumordnung $k \geq 2$ mit der Färbungsnummer $s = 1, 2$, welche die Farbe der Wurzel, $F(1) = \chi_s$, bei sonst gleicher Färbung der übrigen Gitterplätze angibt⁷⁵. Werden die Gitterplätze $i = 2, 3, \dots, |V(\mathcal{D}_{2k-3})|$, siehe Gleichung (2.3), in einem Breitendurchlauf numeriert, so ergibt sich die einfache Färbungsregel

$$F(i) = \begin{cases} \chi_1 & \text{falls } i \text{ gerade} \\ \chi_2 & \text{falls } i \text{ ungerade} \end{cases}. \quad (3.86)$$

⁷⁵Man beachte, daß bei künftigen Erweiterungen des gesamten Rechenverfahrens auf mehr als zwei „Grundfarben“, beispielsweise im Falle von Fermionenmischungen in optischen Gittern, auch diese Metagraphen bei c' Farben auf den Verzweigungsgrad c' und das Färbungsnummernintervall $s = 1, 2, \dots, c'$ verallgemeinert werden müssen.

Beispiel: Sämtliche Meta-Binärbäume für $k = 2$ und $k = 3$:



Auch Meta-Binärbäume können in eine verkürzende Notation der Eckenattribute – bei Farben definitionsgemäß als geordnete Menge von Tupeln – durch einen Tiefendurchlauf (konventionsgemäß von links nach rechts) überführt werden, beispielsweise

$$F({}^m\mathcal{D}_{3,1}) = \{(1, 1, 1, 1), (1, 1, 1, 2), (1, 1, 2, 1), (1, 1, 2, 2), \\ (1, 2, 1, 1), (1, 2, 1, 2), (1, 2, 2, 1), (1, 2, 2, 2)\}$$

${}^m\mathcal{D}_{2k-3}$: Geordnete Menge aller zweifarbigen vollständigen Meta-Binärbäume zur Butcherbaumordnung $k \geq 2$, ${}^m\mathcal{D}_{2k-3} = \{{}^m\mathcal{D}_{2k-3,s} \mid s = 1, 2\}$ beziehungsweise $|{}^m\mathcal{D}_{2k-3}| = 2$; formal gilt $[{}^m\mathcal{D}_{2k-3}]_s = {}^m\mathcal{D}_{2k-3,s}$.

${}^m\mathcal{D}'_{2k-3}$: Geordnete (und im hier diskutierten zweifarbigen Falle nur einelementige) Menge aller vollständigen Meta-Binärbäume zur Butcherbaumordnung $k \geq 2$, welche nur Wurzelfarben $F(1) = \chi_s$ mit ungeraden Werten von s besitzen. Aufgrund der durch die Spinsymmetrie möglichen Wahl nur einer der beiden Färbungen der Butcherbaumwurzel und somit der Festlegung auf die „Farbe“ des ersten hüpfenden Elektrons kann die Menge der Meta-Binärbäume schlicht auf ${}^m\mathcal{D}_{2k-3} \supset {}^m\mathcal{D}'_{2k-3} = \{{}^m\mathcal{D}_{2k-3,s'} \mid s' = 1\}$ reduziert werden.

3.4.6. *Hüpfsequenzen, deren Eigenschaften und Notation.* Unter einer Hüpfsequenz n ter Störungsrechenungsordnung, symbolisch $H^{(n)}$, verstehen wir im Weiteren ein Tupel aus $n+2 = 2k-1 = |H^{(n)}|$ Strukturen, welche die eingenommenen Zustände des betrachteten Quantensystems im Verlauf der Elektronenpropagation vollständig beschreiben. Das erste Element darin ist der Ausgangszustand $[H^{(n)}]_0 = |{}_1M, {}_2M\rangle_{k,l}^{(0)} = |M\rangle_{k,l}^{(0)} \mapsto \mathcal{B}_{k,l,m} \in B'_{k,l}$, das letzte der Endzustand $[H^{(n)}]_{2k-2} = |M\rangle_{k,l}^{(2k-2)} \mapsto \mathcal{B}_{k,l,m} \in B_{k,l}$. Für die Hüpfsequenzelemente gilt mit der durch Gleichung (3.69) eingeführten Transferoperation $\check{T}_{r,s}$ in ausführlicher Schreibweise die rekursive Beziehung

$$H^{(n)} = (|{}_1M, {}_2M\rangle_{k,l}^{(0)}, |{}_1M, {}_2M\rangle_{k,l}^{(j)} = \check{T}_{r^{(j)},s^{(j)}} |{}_1M, {}_2M\rangle_{k,l}^{(j-1)}, \\ j = 1, 2, \dots, 2k-2). \quad (3.87)$$

Das in Kapitel 1 dargelegte Problem fordert, daß bis auf Vertauschung von Teilchen gleichen Spins Ausgangs- und Endzustand hinsichtlich der Färbung identisch sind, also den selben Butcherbaum darstellen. Ohne die Vertauschungsfunktion $\zeta : M \leftrightarrow M'$, welche auch $\zeta : V \leftrightarrow V'$, $\zeta : E \leftrightarrow E'$ und $\zeta : F \leftrightarrow F'$ mit $F' \equiv F$ beziehungsweise $\zeta : C \leftrightarrow C'$ mit $C' \equiv C$ impliziert, zunächst genauer zu spezifizieren⁷⁶, läßt sich

⁷⁶ ζ kann als auf eines der (vektoriellen) Elemente, also eine Farbebene des Zustandsvektors wirkende Permutationsmatrix dargestellt werden; dann ist gerade die Determinante $\det({}_s\zeta) = (-1)^{s^b} = {}_s s \in \{-1, +1\}$

daher formal mit $j = 2k - 2$

$$|M\rangle_{k,l}^{(j)} = \varsigma (|M\rangle_{k,l}^{(0)}) \quad (3.88)$$

schreiben. Gleiches gilt – je nachdem, welcher Operatorklasse f der der Sequenz zugrunde liegende Hüpfprozeß angehört – auch für gerade Werte von j .

Die problemspezifisch für die Teilchentransferoperationen $\check{T}_{r^{(j)},s^{(j)}}$ des j ten Schrittes zu wählenden Parameter $r^{(j)}$ (Sprungkante) und $s^{(j)}$ (Sprungfarbe) erlauben die präzisere Angabe der Zustände in Gleichung (3.87):

$$|M\rangle_{k,l}^{(1)} := \check{T}_{r^{(1)},s^{(1)}} |M\rangle_{k,l}^{(0)} \quad \text{mit} \quad (3.89-1)$$

$$r^{(1)} = F^{(\mathfrak{m}\mathcal{P})}(1) = \llbracket F^{(\mathfrak{m}\mathcal{P})} \rrbracket_{1,1} \quad \forall \mathfrak{m}\mathcal{P} \in \mathfrak{m}\mathcal{P}'_{2k-3} \Leftrightarrow r^{(1)} \in \mathfrak{m}\ddot{E} \quad \text{und}$$

$$s^{(1)} = F^{(\mathfrak{m}\mathcal{D})}(1) = \llbracket F^{(\mathfrak{m}\mathcal{D})} \rrbracket_{1,1} = 1 \quad \forall \mathfrak{m}\mathcal{D} \in \mathfrak{m}\mathcal{D}'_{2k-3},$$

$$|M\rangle_{k,l}^{(j)} := \check{T}_{r^{(j)},s^{(j)}} |M\rangle_{k,l}^{(j-1)} \quad \text{für } j = 2, 3, \dots, 2k - 2 \quad \text{mit} \quad (3.89-2)$$

$$r^{(j)} = \llbracket F^{(\mathfrak{m}\mathcal{P})} \rrbracket_{u,j} \quad \forall \mathfrak{m}\mathcal{P} \in \mathfrak{m}\mathcal{P}'_{2k-3} \quad \text{und}$$

$$s^{(j)} = \llbracket F^{(\mathfrak{m}\mathcal{D})} \rrbracket_{v,j} \quad \forall \mathfrak{m}\mathcal{D} \in \mathfrak{m}\mathcal{D}'_{2k-3}.$$

Mit $\mathbb{H}^{(n)}$ bezeichnen wir die Menge aller Hüpfsequenzen n ter Ordnung; folglich ist $H^{(n)} \in \mathbb{H}^{(n)}$. Die Anzahl der Hüpfsequenzen $\mathfrak{h}_n = |\mathbb{H}^{(n)}|$ ist für das Hubbardmodell bis $n = 15$ und für das Falicov-Kimball-Modell bis $n = 17$ in Tabelle 3.8 zusammengestellt. Die darin teilweise enthaltenen Rechenzeitangaben wurden auf einem CISC-server des Typs HP AM447A [84] bei paralleler Verarbeitung (alle 80 Prozessorkerne mit jeweils zwei *threads* belegt) ermittelt.

Die Erzeugbarkeit von Hüpfsequenzen muß für alle Äste des betrachteten Permutationsbaumes, $u = 1, 2, \dots, (2k - 3)!$, sowie für einen gegebenen Wert von u für alle Äste des zugrunde liegenden Binärbaumes, $v = 1, 2, \dots, 2^{2k-3}$, *geprüft* werden, da aufgrund der Forderungen (3.62) unter Berücksichtigung des Zusammenhanges (3.69) nicht jeder sich formal ergebende Hüpfschritt ausgeführt werden kann. Während, wie im folgenden Abschnitt noch erläutert wird, eine *brute-force*-Implementierung des Algorithmus' *alle* Werte von u und v beispielsweise iterativ in geschachtelten Schleifen prüft, überspringt das vorliegende iterativ-rekursive Verfahren nicht ausführbare Äste – sozusagen „gierig“ voranschreitend – nach dem *greedy*-Prinzip. Damit wird die Effizienz der birekursiven Architektur der Erstimplementierung [18, 19] mit der Nutzbarkeit von (massiv-)parallelen Großrechenanlagen verbunden.

3.4.7. Doppelbesetzungsverlauf und dessen graphische Darstellung. Eine direkt aus der Hüpfsequenz ableitbare physikalisch und algorithmisch bedeutsame Information ist der *Doppelbesetzungsverlauf* \mathfrak{d} , welcher ebenfalls ein Tupel der Länge $n + 2 = 2k - 1$ darstellt und wie ein „Protokoll“ zu jedem (Zwischen)zustand die zugehörige Anzahl der

der gegebenenfalls das Vorzeichen der fermionischen Wellenfunktion bezüglich der betrachteten Farbebene $s = 1, 2, \dots, c'$ wechselnde Faktor. Für das Gesamtvorzeichen, siehe beispielsweise Gleichung (1.27), gilt entsprechend $(-1)^b = s = \prod_s \det(s_\varsigma)$. Eine beispielhafte Anwendung findet sich in Fußnote 99.

k	n	Hubbardmodell			Falicov-Kimball-Modell		
		$\#_n$	G.	Pz./s	$\#_n$	G.	Pz./s
2	1	2	10^0		1	10^0	
3	3	20	10^1		4	10^0	
4	5	648	10^2	0,46	54	10^1	
5	7	45472	10^4	2,98	1568	10^3	
6	9	5644880	10^6	33,9	80680	10^4	
7	11	1099056000	10^9	6060	6510624	10^6	
8	13	310007943616	10^{11}	2140000	761286848	10^8	
9	15	119777421416192	10^{14}		121944722176	10^{11}	*
10	17	**			25668462562560	10^{13}	

Tabelle 3.8. Anzahl der Hüpfsequenzen $\#_n$ im Hubbard- und im Falicov-Kimball-Modell, Referenzen [24:A198761] und [24:A225823]. Orientierungshalber ist zusätzlich die Größenordnung (G.) von $\#_n$ vermerkt. Bei der für einige Werte angegebenen Prozessorzeit (Pz.) handelt es sich um die akkumulierte Rechenzeit aller Prozessoren bei hiesiger paralleler Verarbeitung; der einem Stern zugeordnete Wert beträgt 3060000 Sekunden. Technische Aspekte zu diesen Angaben finden sich detailliert im Abschnitt 5.1. Die durch zwei Sterne vertretene Sequenzanzahl für das Hubbardmodell in der Ordnung U^{-17} ist aus Gründen des enormen Aufwandes, siehe Abschnitt 5.1.1, noch nicht zugänglich.

Doppelbesetzungen angibt. Unter der Voraussetzung $[[d]]_0 = [[d]]_{2k-2} = 0 \Leftrightarrow \chi_3 \notin F(\mathcal{B}_{k,l,m})$ (Ausgangs- und Endzustand enthalten keine Doppelbesetzung) und mit der Zerlegung $|M\rangle_{k,l}^{(j)} = |{}_1M, {}_2M\rangle_{k,l}^{(j)}$ kann folgende Definition formuliert werden:

$$d = (0, d_j, 0 \mid j = 1, 2, \dots, 2k - 3) \quad \text{mit} \quad (3.90)$$

$$\sum_{i=1}^k d'_{j,i} = d_j \in \mathbb{N} \quad \text{und} \quad d'_{j,i} = \begin{cases} 1 & \text{falls } [{}_1M^{(j)}]_i > 0 \wedge [{}_2M^{(j)}]_i > 0 \\ 0 & \text{sonst} \end{cases} .$$

Analog zur Visualisierung der \hat{S} -Exponenten-Abfolgen der Hüpfprozesse durch Pfadgraphen, siehe Abschnitt 3.3.5, deren \hat{S} -Operatoren auf jeden Zustand innerhalb einer Sequenz wirken und die somit ebenfalls ein $n + 2$ -Tupel bilden, lassen sich die Doppelbesetzungsverläufe darstellen⁷⁷, wobei allerdings die Doppelbesetzungsanzahl

⁷⁷Im Gegensatz zu den Hüpfprozesspfadgraphen handelt es sich hierbei jedoch um eine Untermenge der „Motzkinpfade“ [48], da sich die Doppelbesetzungsanzahl allenfalls „um eine Höheneinheit ändern darf“, siehe auch

gemäß Schlüsselgleichung (1.11) bereits ein negatives Vorzeichen erhält. Die zugrundeliegenden Hüpfsequenzen können in gleicher Weise wie die Hüpfprozesse nach Gleichung (3.45) klassifiziert werden (Index f); auf die Ausnutzung dieser bedeutsamen „inneren Symmetrie“ zwischen Sequenzen und Prozessen wird detailliert in Abschnitt 3.5 eingegangen. Tabelle 3.9 faßt die möglichen Verläufe bis zur fünften Ordnung der Störungsrechnung zusammen, Tabelle 3.10 enthält deren empirisch ermittelte Gesamtzahl⁷⁸ bis einschließlich $n = 15$, wobei für eine gegebene Klasse $d_{n,f}$ die Anzahl aller klas-senzugehörigen und somit

$$d_n = \sum_{g=1}^{2^{(n-1)/2}} d_{n,g-1}$$

die Gesamtzahl aller möglichen Doppelbesetzungsverläufe ist. Der Vergleich der Tabelle 3.9 mit 3.5 respektive 3.10 mit 3.4 zeigt $d_n \geq p_n$; darin ist die Gesamtzahl der bereits nach Abschnitt 3.3.4 komprimierten Prozesse

$$p_n = \sum_{g=1}^{2^{(n-1)/2}} |\hat{\mathcal{P}}_{g-1}^{(n)}|,$$

siehe die dazu analoge Gleichung (3.47).

Wie aus Kapitel 1 folgt und insbesondere in Gleichung (1.27) des Beispiels in Abschnitt 1.5.5 zu sehen, sind für die Berechnung der Grundzustandsenergie aus den Hüpfsequenzen nur zwei Informationen vonnöten, nämlich die Anzahl der Doppelbesetzungen d nach jedem Hüpfschritt sowie die Gesamtzahl b der vertauschten Fermionen. Mit dem Wellenfunktionsvorzeichen⁷⁹ $s = (-1)^b$ kann damit eine Hüpfsequenz der Störungsrechnungsordnung n auf das Paar

$$\mathcal{H}^{(n)} = (s, d^{(n)}) \quad (3.91-1)$$

reduziert und analog zum Elementarprozeß

$$\mathcal{P}^{(n)} = (e, p^{(n)}) \quad (3.91-2)$$

mit dem Zusammenhang $|d^{(n)}| = |p^{(n)}| = n+2$ sowie den Eigenschaften $s \in \{-1, +1\}$ und $e \in \mathcal{Q} \setminus \{0\}$ notiert werden. Unter Übernahme der Bezeichnungsweise von $p^{(n)}$ als

Abschnitte 5.2.1 und 6.3.3 sowie Anhang 1.1, während die \hat{S} -Exponentenwerte auch um zwei und mehr Stufen springen.

⁷⁸Aus der weiteren Beschreibung der Verfahrensumsetzung wird deutlich, daß eine direkte Ermittlung dieser Zahlenfolge im Verlauf der Rechnung aufgrund des erheblichen Aufwands – man beachte das Anwachsen der Sequenzanzahlen in Tabelle 3.8 – nicht vorgesehen ist. Für Tabelle 3.10 wurde bei serieller Programmausführung der Doppelbesetzungsverlauf einer jeden zur Grundzustandsenergie beitragenden Hüpfsequenz ausgegeben, in einer Verarbeitungskette mit UNIX-Befehlen auf einen Repräsentanten einschränkend sortiert und abschließend gezählt. Anhang 1.1 beschreibt darüber hinausgehend ein Verfahren, mit dem sich die Doppelbesetzungsverlaufstypen a priori vollständig erzeugen lassen.

⁷⁹Siehe Fußnote 76

n	(f, \mathfrak{d})
1	$(0, (0, 1, 0)) \mapsto (0, \begin{array}{ c } \hline \nabla \\ \hline \end{array})$
3	$(0, (0, 1, 1, 1, 0)) \mapsto (0, \begin{array}{ c c } \hline \nabla & \nabla \\ \hline \end{array}) \wedge (1, (0, 1, 0, 1, 0)) \mapsto (1, \begin{array}{ c c } \hline \nabla & \nabla \\ \hline \end{array})$
5	$(0, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array}) \wedge (0, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array}) \wedge (0, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array}) \wedge (0, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array}) \wedge (0, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array})$ $\wedge (1, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array}) \wedge (1, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array}) \wedge (2, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array}) \wedge (2, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array})$ $\wedge (3, \begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array})$

Tabelle 3.9. Graphische Veranschaulichung (analog zu Tabelle 3.5) der möglichen Doppelbesetzungsverläufe bis einschließlich Störungsordnung $n = 5$.

n	1	3	5	7	9	11	13	15
\mathfrak{d}_n	1	2	10	59	397	2878	21266	162732

Tabelle 3.10. Gesamtzahl \mathfrak{d}_n der möglichen Doppelbesetzungsverläufe aller Klassen bis einschließlich Störungsordnung $n = 15$. Angegeben sind die aus Rechenzeitgründen nur bis $n = 13$ auf Übereinstimmung mit dem Hubbardmodell geprüften Werte für das Falicov-Kimball-Modell; auf die Frage der Gleichheit von \mathfrak{d}_n für beide Modelle wird im Abschnitt 6.3.3 im Ausblick kurz eingegangen.

\hat{S} -Exponenten-Tupel kann $\mathfrak{d}^{(n)}$ bezugnehmend auf Gleichung (1.11) auch ein *Basentupel* n ter Störungsrechnungsordnung genannt werden. Man beachte, daß die Verkürzung des Hüpfsequenzstupels laut Definition in Abschnitt 3.4.6 auf die hiesige Paardarstellung, $H^{(n)} \mapsto \mathcal{H}^{(n)}$, auch auf deren Gesamt(multi)menge zu übertragen ist, $\mathbb{H}^{(n)} \mapsto \hat{\mathcal{H}}^{(n)}$; damit erhalten wir $\mathcal{H}^{(n)} \in \hat{\mathcal{H}}^{(n)}$ „spiegelbildlich“ zu $\mathcal{P}^{(n)} \in \hat{\mathcal{P}}^{(n)}$.

3.4.8. *Hüpfspintupel.* Neben dem Doppelbesetzungsverlauf ist das *Hüpfspintupel* \mathfrak{z} der Länge $n + 1 = 2k - 2$ verfahrensnötwendig, welches für jeden Hüpfschritt den Spin des gesprungenen Teilchens verzeichnet. Unter der Bedingung aus Gleichung (3.89-1), daß eine Sequenz stets mit $s^{(1)} = 1$ beginne, also $\llbracket \mathfrak{z} \rrbracket_0 = 1$ sei, kann folgendermaßen festgelegt werden:

$$\mathfrak{z} = (1, s^{(j)} \mid j = 2, 3, \dots, 2k - 2) \quad \text{mit} \quad (3.92)$$

$$s^{(j)} = \begin{cases} 1 & \text{falls } {}_1M^{(j)} \neq {}_1M^{(j-1)} \\ 2 & \text{sonst, also falls } {}_2M^{(j)} \neq {}_2M^{(j-1)} \end{cases} .$$

Die Anzahl der formal möglichen verschiedenen Hüpfspintupel entspricht $|\mathbb{F}({}^m\mathcal{D})|$, siehe Gleichung (3.89-2).

3.4.9. *Iterativer Rahmen.* Der zentrale Programmbereich zur Ausgangszustandssequenzierung, welchem wir uns nun zuwenden, ist seinerseits wiederum in drei Teile gegliedert, deren erster, iterativ über alle relevanten Butcherbäume laufend, die beiden anderen, siehe Abschnitte 3.4.11 und 3.4.13, rahmenartig einfaßt. In diesem Rahmen wird wie folgt vorgegangen:

- 1 Ausführungsparameter vom aufrufenden Hauptprogramm übernehmen: Graphenordnung $k = (n + 3)/2$, Butcherbaumtafel \mathbb{B}_k , Anzahl der im parallelisierten Abschnitt genutzten Prozessoren sowie Steuerparameter für den Umfang an Kontrollausgaben (Speichernutzung etc.) und Zwischenergebnissen, vergleiche Flußdiagrammelement (1) in Abbildung 3.6
- 2 Lokale Zeitmessung beginnen (Schritt aus Platzgründen nicht im Programmablaufschema eingezeichnet)
- 3 Ausgangszustandsmenge \mathbb{B}'_k aufbauen, Element (2): Parallel⁸⁰ für $l = 1, 2, \dots, \lambda_k$ alle $\mathcal{B}_{k,l,m}$ gemäß der in Gleichung (3.66) formulierten Kriterien aus $\mathcal{B}_{k,l}$ herausfiltern. Dazu kann eine entsprechend vereinfachte Variante des im folgenden Abschnitt 3.4.11 beschriebenen Dekodierers der Zeichenkettendarstellung der Butcherbäume dienen, nach dessen Ausführung ${}^m\bar{E}(\mathcal{B}_{k,l,m}) \neq \emptyset$ gelten muß. Effektiv werden die gültigen Färbungsnummern, zusammen mit der zugehörigen Strukturnummer, linearisiert⁸¹ in einer Liste abgelegt:

$$\bar{\mathbb{B}}_k = \bigcup_{l=1}^{\lambda_k} \{(l, m) \mid \forall m \in [1, \mu_{k,l}] : \mathcal{B}_{k,l,m} \in \mathcal{B}'_{k,l}\}. \quad (3.93)$$

Es gilt entsprechend $|\bar{\mathbb{B}}_k| = \mathbb{b}'_k = \|\mathbb{B}'_k\|$.

- 4 Anfang des parallelisierten Bereichs⁸². Für $i = 1, 2, \dots, |\bar{\mathbb{B}}_k|$: Schleife über alle Ausgangszustände mit $l' = \llbracket \bar{\mathbb{B}}_k \rrbracket_{1,i}$ und $m' = \llbracket \bar{\mathbb{B}}_k \rrbracket_{2,i}$:

⁸⁰Wie im Abschnitt 5.1 ausführlicher dargelegt, erfolgt in dieser Implementierung, ungleicher Lastverteilung durch unterschiedliche Werte von $\mu_{k,l}$ und folglich ungleichen Schleifenlaufzeiten entgegenwirkend, die Filterung in einer Schleife über $m = 1, 2, \dots, \mu_{k,l}$ für eine durch die äußere Schleife gegebene Strukturnummer l . Hat ein *thread* das diesem vom Auftragsverteiler (*dispatcher/scheduler*) zugewiesene (im einfachsten Falle zusammenhängende) Färbungsnummernintervall $1 \leq m_1, m_1 + 1, \dots, m_2 \leq \mu_{k,l}$ abgearbeitet und sind die restlichen Intervalle der inneren Schleife bereits verteilt, so wird – ohne auf die anderen *threads* zu warten (“*no-wait clause*”) – das Verfahren nach Erhöhung von l in der äußeren Schleife fortgesetzt und ein Bereich $1 \leq m_3, m_3 + 1, \dots, m_4 \leq \mu_{k,l+1}$ zugeteilt.

⁸¹Diese Vorgehensweise stellt bereits eine Optimierung dar, siehe Abschnitt 5.1: Der Test auf Zugehörigkeit eines Baums zur Ausgangszustandsmenge ließe sich naheliegender auch in einer geschachtelten Doppelschleife über die Indices l und m vor dem gegebenenfalls erfolgenden Aufruf der Propagationsfunktion durchführen – allerdings mit merklichen Leerlaufzeiten der Aufträge erwartenden *threads*.

⁸²Sämtliche Arbeitsvariablen und -datenstrukturen mit Schreibzugriff müssen bei paralleler Verarbeitung *thread-unabhängig* (*private*) angelegt werden; entsprechende mit Lesezugriff werden gemeinsam verwendet (*shared*).

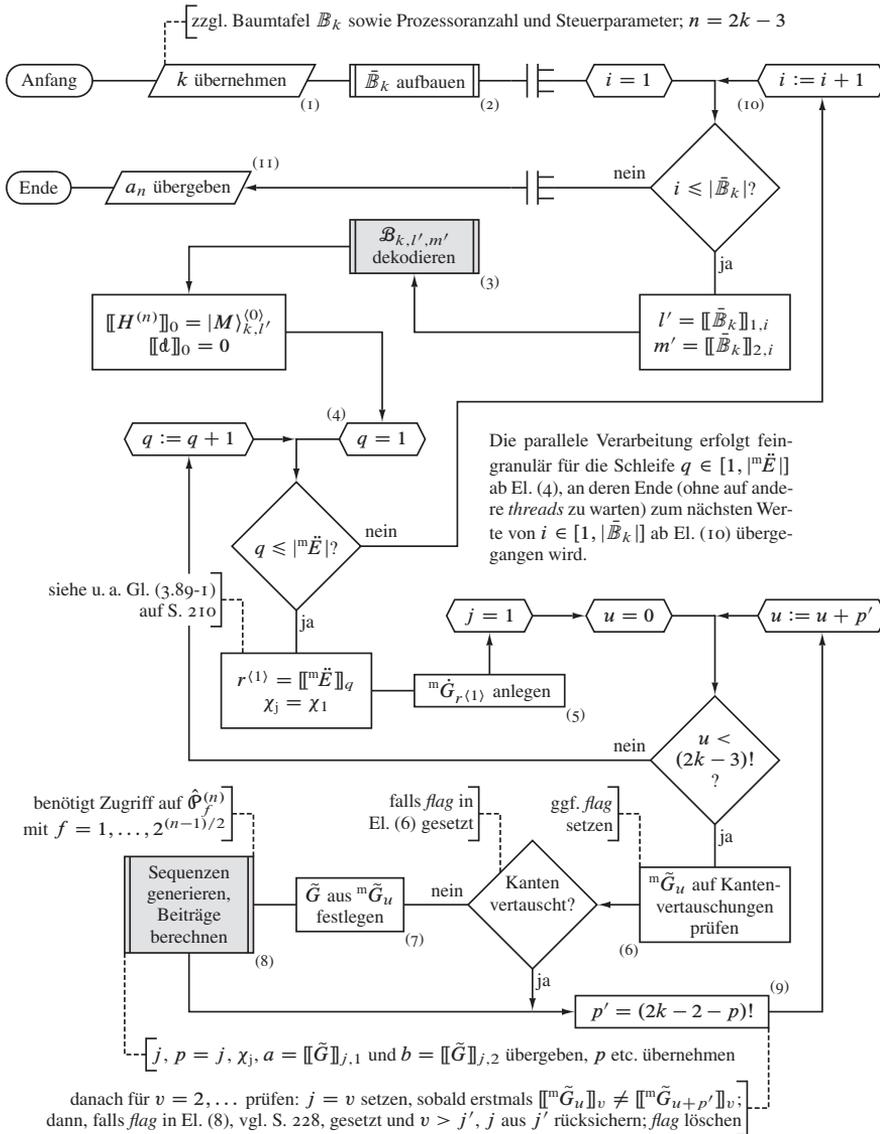


Abbildung 3.6. Ablaufplan des iterativen Rahmens, vertreten durch Flußdiagrammelement (3) in Abbildung 3.1. Das Unterprogramm zu hiesigem Element (3) ist in Abbildung 3.8 (Abschn. 3.4.11 ab S. 221), die rekursive Prozedur Element (8) in Abbildung 3.9 (Abschn. 3.4.13 ab S. 227) gezeichnet.

- 4.1 Nach Initialisierung der erforderlichen internen Variablen die Vorbereitungsstufe, Abschnitt 3.4.1.1, aufrufen, Element (3): den zu $\llbracket \mathbb{B}_k \rrbracket_i$ gehörigen Baum $\mathcal{B}_{k,l',m'}$ (nach dortiger Programmformulierung in Zeichenkettendarstellung mit dem *Zeiger* z auf das erste Zeichen) und $g = 0$ als anfängliche Baumebene (Wurzel) übergeben, ${}^m E_{\text{CNL}}$, ${}^m \ddot{E}$, G und M für $|M\rangle_{k,l'}^{(0)}$ übernehmen
- 4.2 $\llbracket H^{(n)} \rrbracket_0 = |M\rangle_{k,l'}^{(0)}$ und entsprechend $\llbracket d \rrbracket_0 = 0$ initialisieren
- 4.3 Für $q = 1, 2, \dots, |{}^m \ddot{E}(\mathcal{B}_{k,l',m'})|$: Parallelisierte⁸³ Schleife über alle Sequenzinitialisierungskanten:
- 4.3.1 $r^{(1)} = \llbracket {}^m \ddot{E} \rrbracket_q$ initialisieren und voraussetzungsgemäß $s^{(1)} = 1$ durch $\chi_j = \chi_1$ festlegen, siehe Gleichung (3.89-1)
- 4.3.2 Formal ${}^m \mathcal{P} := {}^m \mathcal{P}_{2k-3, r^{(1)}}$ generieren, konkret zunächst initiale Liste der Hüpfkantenindices als Meta-Kantenmenge
- $${}^m \dot{G}_{r^{(1)}} = \{r^{(1)}, 1, 2, \dots, r^{(1)} - 1, r^{(1)} + 1, \dots, 2k - 3\} \quad (3.94)$$
- anlegen, Element (5)
- 4.3.3 Sprungnummer $j = 1$ setzen
- 4.3.4 Für $u = 0, \dots$: Schleife über ${}^m \tilde{G}_u = \llbracket F({}^m \mathcal{P}) \rrbracket_{u+1}$, also sämtliche Permutationsbaumäste, beziehungsweise (indirekter Weg) über alle Hüpfkantenvertauschungen, beginnend mit ${}^m \tilde{G}_0 = {}^m \dot{G}_{r^{(1)}}$, gegebenenfalls endend mit ${}^m \tilde{G}_{(2k-3)!-1} = \{r^{(1)}, 2k - 3, 2k - 4, \dots, r^{(1)} + 1, r^{(1)} - 1, \dots, 1\}$:
- 4.3.4.1 Hüpfkantenabfolge auf Vertauschung zweier klassengleicher Kanten in einer impliziten Schleife über alle ${}^m E_{\text{CNL}}$ überprüfen⁸⁴, Element (6), und bei entsprechendem Befund ein *flag* setzen. Rückgabewert der Funktion⁸⁵ ist die Elementnummer der ersten der klassengleichen Kanten in ${}^m \tilde{G}_u$, welche mit der Ebene p des zugrundeliegenden Permutationsbaumes ${}^m \mathcal{P}$ übereinstimmend ist⁸⁶.

⁸³Der parallelisierte Bereich erstreckt sich in der vorliegenden Neuimplementierung über alle Ausgangszustände innerhalb der mit Schritt 4 beginnenden Schleife; falls sich sämtliche Initialisierungskanten eines gegebenen Ausgangszustands bereits in der Verarbeitung befinden, wird durch die *no-wait*-Auftragsverteilung (vergleiche auch Fußnote 80) durch den *dispatcher* mit der Bearbeitung des nächsten Ausgangszustands begonnen und Aufträge auf Ebene der Initialisierungskanten an die wartenden *threads* verteilt (Details siehe Abschnitt 5.1).

⁸⁴Siehe Fußnote 95. Implementierungsabhängig kann ein Rechenzeitgewinn mit dem vorausgehenden Test auf $|{}^m E_{\text{CNL}}| < k - 1$ verbunden sein: Nur wenn es weniger Kantenklassen als Kanten gibt, also überhaupt zwei oder mehr klassengleiche Kanten existieren, muß auf Vertauschung geprüft werden. Im Falle des mit dieser Arbeit vorgelegten Programmes wird in der zur Meta-Kantenklassenmenge gehörigen Datenstruktur nur dann eine Klasse in Form einer Unterliste angelegt, wenn Kantengleichheit besteht; falls $|{}^m E_{\text{CNL}}| = k - 1$ ist die Kantenklassenliste folglich leer.

⁸⁵Aus Gründen der Ausführungsgeschwindigkeit ist diese in der vorgelegten Implementierung *in-line* ausgeführt worden.

⁸⁶Scheinbar von Abschnitt 2.1.10 abweichend, jedoch mit der Berechtigung, daß die Permutationsbäume ${}^m \mathcal{P}'_{2k-3} = \{{}^m \mathcal{P}_{2k-3, r'} \mid r' \in {}^m \ddot{E}\}$ formal als Äste einen (stets unvollständigen, da $|{}^m \ddot{E}| \leq |{}^m G|/2$) „Über-Permutationsbaum“ mit eigener (attributloser) Wurzel bilden, wird in diesem Zusammenhang die Wurzel eines jeden ${}^m \mathcal{P}_{2k-3, r'}$ als dessen erste Ebene betrachtet.

- 4.3.4.2 Falls *flag* im vorausgehenden Programmschritt gesetzt, zum Schleifenende springen, Schritt 4.3.5
 4.3.4.3 Hüpfkantenabfolge anhand der Indexmenge festlegen, Element (7),

$$\tilde{G} = \{ \llbracket G \rrbracket_h \mid h = \llbracket \tilde{G}_u \rrbracket_j \mid j = 1, 2, \dots, 2k - 2 \} \quad (3.95)$$

Es genügt, j von p bis $2k - 2$ laufen zu lassen, da in ${}^m\tilde{G}_u$ keine Mengenelemente unter $j = p$ vertauscht wurden.

- 4.3.4.4 Nach Initialisierung der erforderlichen internen Variablen das Unterprogramm für die Elektronenpropagation aufrufen⁸⁷, vergleiche Flußdiagrammelement (8) und siehe zudem Abschnitt 3.4.13
 4.3.5 Formal $p' = (2k - 2 - p)!$ Äste auf ${}^m\mathcal{P}$ überspringen, konkret zu $u := u + p'$ übergehen, vergleiche Element (9), und *ute* Permutation der Kantenindices in einem Unterprogramm erzeugen, dann zu Schritt 4.3.4 rückspringen, falls $u < (2k - 3)!$; sonst Schleife beenden.⁸⁸
 4.4 Falls $q < |{}^m\tilde{E}(\mathcal{B}_{k,l',m'})|$, zu $q := q + 1$ übergehen, Schritt 4.3; sonst Schleife beenden
 5 Falls $i < |\tilde{B}_k|$, zu $i := i + 1$ übergehen, Schritt 4; sonst Schleife beenden und parallelisierten Bereich verlassen⁸⁹
 6 Lokale Zeitmessung beenden und Rechenzeit ausgeben (Schritt nicht im Programmablaufschema eingezeichnet)

⁸⁷Übergabeparameter sind dabei insbesondere die Sprungnummer j , der Permutationsbaumebenenanzähler $p = j$, die nach Gleichung (3.64) aus $|M\rangle_{k,l'}^{(j-1)}$ ermittelte „springende Besetzungsfarbe“ $\chi_j \in X$, welche bei $\chi_j = \chi_3$ (Doppelbesetzung) in den hüpfenden Spin aus X' aufzulösen oder sonst ($\chi_j = \chi_0$ (Loch) führt naturgemäß zum Abbruch) mit diesem gleichzusetzen ist, ferner der Quellgitterplatz $a = \llbracket \tilde{G} \rrbracket_{j,1}$ und der Zielgitterplatz $b = \llbracket \tilde{G} \rrbracket_{j,2}$; Rückgabewert ist – neben den Beiträgen zum Reihenkoeffizienten – die Permutationsbaumebene p , in welcher die Rekursion beendet beziehungsweise abgebrochen wurde, die Hüpf-schritt-nummer j sowie gegebenenfalls deren Sicherungsvariable j' nebst zugehörigem *flag*.

⁸⁸Im Programm wird der Schleifenabbruch durch den Rückgabewert des „Permutierers“ ausgelöst, siehe Anhang, Abschnitt 2.2, sobald beim rechts in der Kantenindexmenge beginnenden Verfahren deren erstes Element, also der Index der Initialisierungskante erreicht wird. In diesem Schritt ist wichtig, die „Ausstiegsebenennummer“ j des Hüpfkerns mit der jener Permutationsbaumebene v in ${}^m\mathcal{P}$ abzugleichen, in welcher im Ast $u + p'$ gegenüber Ast u das Eckenattribut wechselt, also das niedrigste in ${}^m\tilde{G}_{u+p'}$ gegenüber ${}^m\tilde{G}_u$ geänderte Element v zu finden: wenn erstmals $\llbracket \tilde{G}_u \rrbracket_v \neq \llbracket \tilde{G}_{u+p'} \rrbracket_v$ für $v = 2, 3, \dots$, dann $j = v$ setzen. Damit aber in der neuen Permutation nicht durch eine unter v liegende Binärbaumgabelung, ab welcher separat zwei Zwischenzustände weiterzuverfolgen sind, übergangen wird, muß die neue „Einstiegsebene“ j auf die Ebene der ersten überhaupt auftretenden Gabelung j' gesetzt werden. (Alternativ bietet sich an, einen zusätzlichen Binärbaum-rekursiven Stapelspeicher einzuführen, dessen Aufgabe in der Pufferung der Hüpf-schritt-nummern j , der zugehörigen Zustandsvektoren $|M\rangle^{(j)}$ und der Listenelemente $\llbracket d \rrbracket_j$ und $\llbracket \tilde{z} \rrbracket_j$ an den Verzweigungspunkten besteht, sowie einen Programmenschub, welcher im Falle einer neuen Permutation gegebenenfalls alle zwischengespeicherten Zustände des Systems bis zum Einstiegshüpfschritt j berücksichtigt.) Diese wird im Kern erfaßt und dem Rahmen übergeben; der „den Mechanismus verriegelnde Schalter“ muß entsprechend vor dem nächsten Permutationsschleifendurchlauf wieder geöffnet werden.

⁸⁹Dabei sind *thread*-lokale Zwischenergebnisse, insbesondere für die Reihenkoeffizienten a_n , in „globalen“ Variablen zusammenzufassen.

- 7 Verfahren beenden und Reihenkoeffizienten a_n mit $n = 2k - 3$ sowie weitere graphentheoretisch relevante Daten an das aufrufende Hauptprogramm übergeben, Element (11)

3.4.10. *Beispiel.* Die vorausgehend beschriebene Arbeitsweise des Rahmens im Allgemeinen an einem konkreten Beispiel vollständig und übersichtlich zu illustrieren, scheidet an der auftretenden Datenmenge: Insbesondere erfordert der Teilalgorithmus zur permutationsbezogenen Kantenabfolgeprüfung innerhalb einer jeden Kantenklasse mindestens einen Butcherbaum der Ordnung $k = 4$ mit entsprechenden Symmetrien, konkret zum Beispiel $\mathcal{B}_{4,2,65} = 1u(1d(2u()))$ oder $\mathcal{B}_{4,2,68} = 1u(1d(2d()))$ mit ${}^mE_{\text{CNL}} = \{\{1\}, \{2, 3\}\}$, welcher mit $j' = 2(k - 1) = 6$ Hüpfritten die Behandlung von Kantenpermutationen in der Größenordnung $(j' - 1)! = 120$ nach sich zieht. Um uns eine Übersicht zu verschaffen und über den „trivialen“ Fall der Störungsrechnungsordnung $n = 1$ in Abschnitt 1.5.5 hinauszugehen, betrachten wir daher zunächst $n = 3$, also die Ausgangszustände der Baumordnung $k = (n + 3)/2 = 3$.

Nach den beiden ersten initialisierenden Programmpunkten filtert Schritt 3 die vierfarbigen Butcherbäume $\mathcal{B}_3 = \{\{\mathcal{B}_{3,1,m} \mid m = 1, \dots, 64\}, \{\mathcal{B}_{3,2,m} \mid m = 1, \dots, 40\}\}$. An Ausgangszuständen verbleiben gemäß den Auswahlkriterien $\mathcal{B}'_3 = \{\{\mathcal{B}_{3,1,26}, \mathcal{B}_{3,1,27}\}, \{\mathcal{B}_{3,2,16}, \mathcal{B}_{3,2,18}\}\}$, abgelegt in der – rechentechnisch vorteilhaft – linear lesebaren Liste $\bar{\mathcal{B}}_3 = \{(1, 26), (1, 27), (2, 16), (2, 18)\}$.

In der mit Schritt 4 beginnenden Schleife schränken wir uns auf $\llbracket \bar{\mathcal{B}}_3 \rrbracket_i$ mit $i = 1$, also $l' = 1$ und $m' = 26$, ein und untersuchen folglich

$$\mathcal{B}_{3,1,26} = \begin{array}{c} \circ \\ \circ \\ \circ \end{array} \equiv 1u(1d(1u())) .$$

Programmpunkt 4.1 übergibt den später noch umfassend beschriebenen *parser* hauptsächlich die Baumzeichenkette und erhält die folgenden, für den weiteren Ablauf wesentlichen Daten: mit ${}^mE_{\text{CNL}} = \{\{1\}, \{2\}\}$ die Menge der in jeweils einer Menge zusammengefaßten Kanten gleicher Klasse (in Adreßform) und mit ${}^m\vec{E} = \{1\}$ die Menge der Sequenzinitialisierungskanten (ebenfalls in Form der Kantennummern) sowie mit $G = \{(1, 2), (2, 3), (3, 2), (2, 1)\}$ die Menge der Hin- und Rücksprungkanten (jeweils als Eckenpaar). Zudem wird in $M = \{(1, 0, 3), (0, 2, 0)\}$ als Paar von Dreitupeln der elektronische (Ausgangs)zustand des Gitterausschnittes kodiert, wobei jedes Teilchen innerhalb des Tupels seiner Spinquantenzahl („Farbeebene“) mit der Nummer seines Ausgangsplatzes versehen ist. Im anschließenden Schritt angelegt beginnen folglich die Hüpfsequenz mit $\llbracket H^{(3)} \rrbracket_0 = |M\rangle_{3,1}^{(0)} = |(1, 0, 3), (0, 2, 0)\rangle_{3,1}^{(0)}$ und die Doppelbesetzungssequenz (voraussetzungsgemäß) mit $\llbracket d \rrbracket_0 = 0$.

Die Schleife über alle Initialisierungskanten, beginnend mit Programmschritt 4.3, ist im betrachteten Fall nur mit $q = 1$ aufgrund $|{}^m\vec{E}| = 1$ zu durchlaufen; entsprechend werden $r^{(1)} = \llbracket {}^m\vec{E} \rrbracket_q = 1$ und (wiederum gemäß Definition) $s^{(1)} = 1$ zugewiesen.

Die anfängliche Hüpfkanten(meta)liste lautet somit ${}^m\dot{G}_1 = \{1, 2, 3, 4\}$; diese entspricht der Attributmengende des ersten Astes $[[\mathcal{F}]]_1$ des Permutationsbaums ${}^m\mathcal{P}_{3,1}$, welche vollständig $\mathcal{F}({}^m\mathcal{P}_{3,1}) = \{\{1, 2, 3, 4\}, \{1, 2, 4, 3\}, \{1, 3, 2, 4\}, \{1, 3, 4, 2\}, \{1, 4, 2, 3\}, \{1, 4, 3, 2\}\}$ lautet. Dementsprechend beginnt Schleife 4.3.4 mit der – formal betrachtet „nullten“ – Kantenpermutation ${}^m\tilde{G}_0 = {}^m\dot{G}_1$; im weiteren Ablauf werden sukzessiv nachfolgend aufgeführte Vertauschungen erzeugt (Nr. steht für die Nummer des Schleifendurchlaufs) und in Schritt 4.3.4.3 in die Hüpfkantenabfolgen \tilde{G} umgewandelt, vergleiche dazu auch Skizze 3.3:

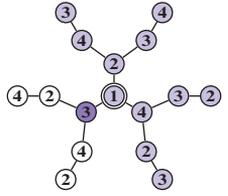
Nr.	u	${}^m\tilde{G}_u$	\tilde{G}	Bemerkung
1	0	{1, 2, 3, 4}	{(1, 2), (2, 3), (3, 2), (2, 1)}	
2	1	{1, 2, 4, 3}	{(1, 2), (2, 3), (2, 1), (3, 2)}	
3	2	{1, 3, 2, 4}	{(1, 2), (3, 2), (2, 3), (2, 1)}	
–	3	{1, 3, 4, 2}	{(1, 2), (3, 2), (2, 1), (2, 3)}	entfällt, siehe unten
4	4	{1, 4, 2, 3}	{(1, 2), (2, 1), (2, 3), (3, 2)}	
5	5	{1, 4, 3, 2}	{(1, 2), (2, 1), (3, 2), (2, 3)}	

Die Aufruf- und Rückgabewerte der sich anschließenden rekursiven Elektronenpropagationsprozedur, Programmpunkt 4.3.4.4, lassen sich wie folgt tabellieren, wobei die Permutationsebene v nur angegeben wird, wenn $v < p$ sowie $v < j'$ gilt und diese somit maßgeblich den Einstiegshüpfschritt j in der nachfolgenden Kantenvertauschung verringert:

Nr.	u	${}^m\tilde{G}_u$	j	j'	p	p'	v	flag?	gültige Sequenzen/ Bemerkung
1	0	{1, 2, 3, 4}	1	4	3	1	–	ja	(3.103-1)
2	1	{1, 2, 4, 3}	3	4	3	1	2	ja	(3.103-2)
3	2	{1, 3, 2, 4}	2	–	2	2	–	nein	nicht vollst. ausführbar, $p = 2$
–	3	{1, 3, 4, 2}	–	–	–	–	–	—	übersprungen, da $p' = 2$
4	4	{1, 4, 2, 3}	2	2	3	1	–	ja	(3.103-3)
5	5	{1, 4, 3, 2}	2	2	3	1	–	ja	(3.103-4)
–	6	{2, 1, 3, 4}	–	–	–	–	–	—	Abbruch, da $u = 3!$

Die wesentlichen Informationen daraus lassen sich übersichtlich im Permutationsbaum visualisieren (vergleiche Tabelle 3.11); dabei werden geprüfte und ausführbare Hüpf-schritte hellviolett und geprüfte und nicht ausführbare Schritte dunkelviolett gefärbt und

übergangene Hüpf Schritte weiß belassen:



Entfallen können in diesem Beispiel bei allen Permutationen die Programmschritte 4.3.4.1 und 4.3.4.2, da $|{}^m E_{\text{CNL}}| = 2 \not\leq k - 1$, folglich jede Kante einer eigenen Klasse angehört. Da mit $u = 5$ und $q = 1$ die Obergrenzen der Schleifen 4.3.4 (siehe Programmschritt 4.3.5) und 4.3 erreicht sind, erfolgt der Übergang zum nächsten Ausgangszustand $\llbracket \tilde{\mathcal{B}}_3 \rrbracket_i$ mit $i = 2$ ($l' = 1$ und $m' = 26$, also $\mathcal{B}_{3,1,27} = 1u(1d(1d(\cdot)))$) in Schleife 4, dessen Verarbeitung „aus Sicht“ des Kernes umfassend in Abschnitt 3.4.14 beschrieben wird.

Abschließend wenden wir uns stark verkürzt dem bereits oben erwähnten Butcherbaum

$$\mathcal{B}_{4,2,65} = \text{graph symbol} \equiv 1u(1d(2u(\cdot)))$$

zu, bei welchen erstmalig eine Kantenklasse mehr als ein Element enthält: $\llbracket {}^m E_{\text{CNL}} \rrbracket_2 = \{2, 3\}$. In der mit Programmpunkt 4.3.4 beginnenden Schleife zur Behandlung des ${}^m \mathcal{P}_{5,1}$ treten nachfolgende Ablaufparameter auf, wobei entsprechend alle Permutation mit der Kantenabfolge ${}^m \tilde{G} = \{1, \dots, 3, \dots, 2, \dots\}$ übergangen werden. Auch hier wird die Permutationsebene v nur angegeben, falls $v < p$ sowie $v < j'$ und dadurch die Ebene des ersten Hüpf Schritts j in der nachfolgenden Kantenvertauschung abgesenkt wird.

Nr.	u	${}^m \tilde{G}_u$	j	j'	p	p'	v	flag?	Ergebnis
1	0	{1, 2, 3, 4, 5, 6}	1	-	3	6	-	nein	nicht vollst. ausführbar
2	6	{1, 2, 4, 3, 5, 6}	3	-	3	6	-	nein	nicht vollst. ausführbar
3	12	{1, 2, 5, 3, 4, 6}	3	6	5	1	-	ja	1 gültige Sequenz
4	13	{1, 2, 5, 3, 6, 4}	5	6	5	1	4	ja	1 gültige Sequenz
5	14	{1, 2, 5, 4, 3, 6}	4	-	4	2	-	nein	nicht vollst. ausführbar
6	16	{1, 2, 5, 6, 3, 4}	4	4	5	1	-	ja	1 gültige Sequenz
7	17	{1, 2, 5, 6, 4, 3}	4	4	5	1	3	ja	1 gültige Sequenz
8	18	{1, 2, 6, 3, 4, 5}	3	-	4	2	-	nein	nicht vollst. ausführbar
9	20	{1, 2, 6, 4, 3, 5}	4	6	5	1	-	ja	1 gültige Sequenz
10	21	{1, 2, 6, 4, 5, 3}	5	6	5	1	4	ja	1 gültige Sequenz

Fortsetzung auf der nächsten Seite

Nr.	u	${}^m\tilde{G}_u$	j	j'	p	p'	v	$flag?$	Ergebnis
<i>Fortsetzung</i>									
11	22	{1, 2, 6, 5, 3, 4}	4	4	5	1	—	ja	1 gültige Sequenz
12	23	{1, 2, 6, 5, 4, 3}	4	4	5	1	—	ja	1 gültige Sequenz
13	24	{1, 3, 2, 4, 5, 6}	—	—	2	24	—	—	Kantenvertauschung
14	48	{1, 4, 2, 3, 5, 6}	2	—	2	24	—	nein	nicht vollst. ausführbar
15	72	{1, 5, 2, 3, 4, 6}	2	—	2	24	—	nein	nicht vollst. ausführbar
16	96	{1, 6, 2, 3, 4, 5}	2	2	4	2	—	ja	nicht vollst. ausführbar
17	98	{1, 6, 2, 4, 3, 5}	2	2	5	1	—	ja	1 gültige Sequenz
18	99	{1, 6, 2, 4, 5, 3}	2	2	5	1	—	ja	1 gültige Sequenz
19	100	{1, 6, 2, 5, 3, 4}	2	2	5	1	—	ja	1 gültige Sequenz
20	101	{1, 6, 2, 5, 4, 3}	2	2	5	1	—	ja	1 gültige Sequenz
21	102	{1, 6, 3, 2, 4, 5}	—	—	3	6	—	—	Kantenvertauschung
22	108	{1, 6, 4, 2, 3, 5}	2	2	5	1	—	ja	1 gültige Sequenz
23	109	{1, 6, 4, 2, 5, 3}	2	2	5	1	—	ja	1 gültige Sequenz
24	110	{1, 6, 4, 3, 2, 5}	—	—	4	2	—	—	Kantenvertauschung
25	112	{1, 6, 4, 5, 2, 3}	2	2	4	2	—	ja	nicht vollst. ausführbar
26	114	{1, 6, 5, 2, 3, 4}	2	2	5	1	—	ja	1 gültige Sequenz
27	115	{1, 6, 5, 2, 4, 3}	2	2	5	1	—	ja	1 gültige Sequenz
28	116	{1, 6, 5, 3, 2, 4}	—	—	4	2	—	—	Kantenvertauschung
29	118	{1, 6, 5, 4, 2, 3}	2	2	4	2	—	ja	nicht vollst. ausführbar
—	120	{2, 1, 3, 4, 5, 6}	—	—	—	—	—	—	Abbruch, da $u = 5!$

Man beachte den Wirkungsgrad des „gierigen“ Kombinationsverfahrens aus iterativem Rahmen und rekursivem Kern, welcher bereits in einem Beispiel der Störungsrechenungsordnung U^{-5} offenkundig wird: Von 120 Kantenpermutationen werden nur noch 29 untersucht, von welchen 16 zur Grundzustandsenergie beitragende Hüpfsequenzen erzeugen; damit erhöht sich die „Trefferquote“ für diesen Ausgangszustand von etwa 13,3 % auf gerundet 55,2 %.

Vorteilhaft werden auch hier die Daten im entsprechend eingefärbten Permutationsbaum veranschaulicht, siehe Abbildung 3.7: Für die 16 Sequenzen sind mathematisch formal 96, vom Programm nur 48 Sprünge zuzüglich 9 Fehlversuchen auszuführen; somit ergibt sich eine Erfolgsquote von etwa 84,2 %. Von den nach Gleichung (2.2) insgesamt 326 Ecken des Permutationsbaums, die jeweils einen Sprung repräsentieren, beansprucht das Verfahren nur die zusammengenommen 57 markierten, rund 17,5 %.

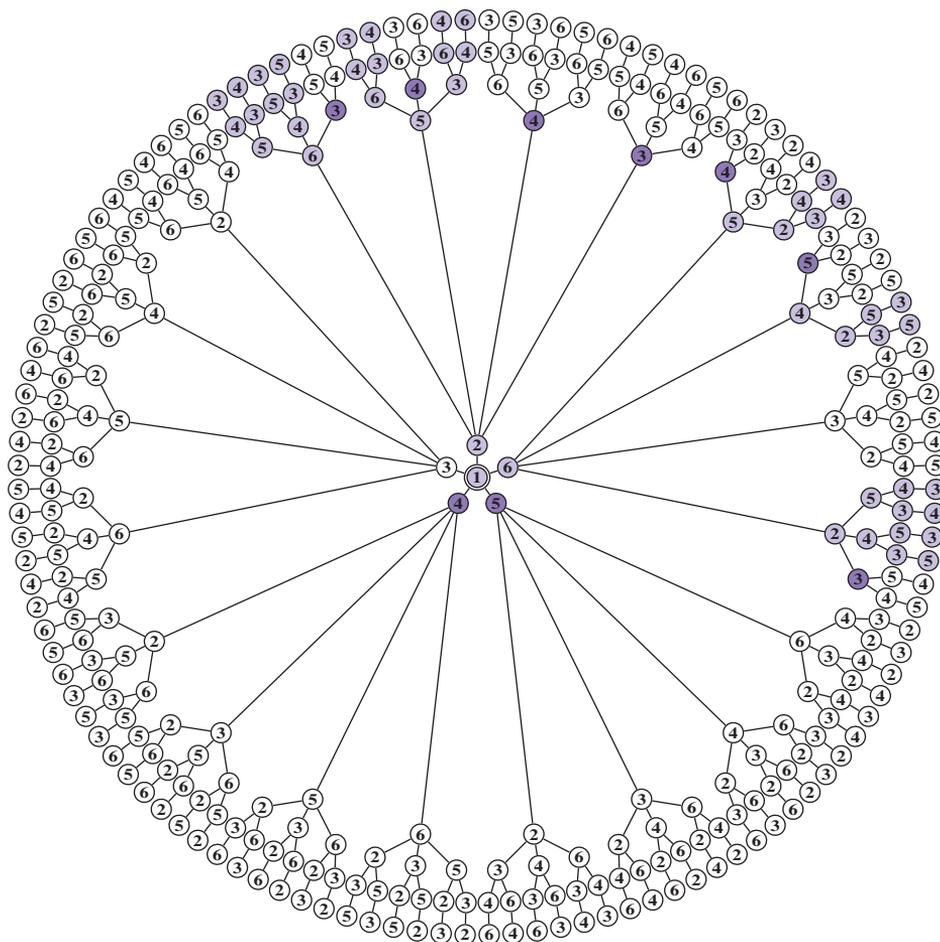


Abbildung 3.7. Beispiel zur Sequenzierung: Zum Ausgangszustand $\mathcal{B}_{4,2,65}$ gehöriger Permutationsbaum ${}^m\mathcal{P}_{5,1}$. Vom Kernalgorithmus geprüfte und ausführbare Hüpf Schritte sind *hellviolett*, geprüfte und nicht ausführbare Schritte sind *dunkelviolett* und vom Rahmenalgorithmus übergangene Schritte sind *weiß* markiert; Eckenattribut ist die Hüpfkantennummer im Butcherbaum. Aus Platzgründen wurde in der Darstellung der Abstand zwischen der zweiten und dritten Schale stark vergrößert.

3.4.11. *Rekursive Vorbereitungsstufe.* Innerhalb des zuvor in Abschnitt 3.4.9 beschriebenen Rahmenunterprogrammes müssen vor Ausführung des (im Abschnitt 3.4.13 detailliert dargelegten) Kernes für das Verfahren wesentliche Strukturen bereitgestellt werden. Diese sind insbesondere die Hüpfkantenmenge G sowie die Meta-Sequenzinitialisierungskantenmenge ${}^m\ddot{E}$ und die Meta-Kantenklassenmenge ${}^mE_{\text{CNL}}$; zudem ist der zu

den Ausgangszuständen gehörige Baum $\mathcal{B}_{k,l',m'}$ gemäß Gleichung (3.65) über M in den zugehörigen Zustandsvektor $|M\rangle_{k,l}^{(0)}$ unter „Personalisierung“ der Elektronen nach Gleichung (3.63) umzuschreiben.

Unter Berücksichtigung der bereits definierten Kriterien für Ausgangszustände, insbesondere der Wahl der Wurzelfarbe und somit die Vorgabe der ersten drei Zeichen der

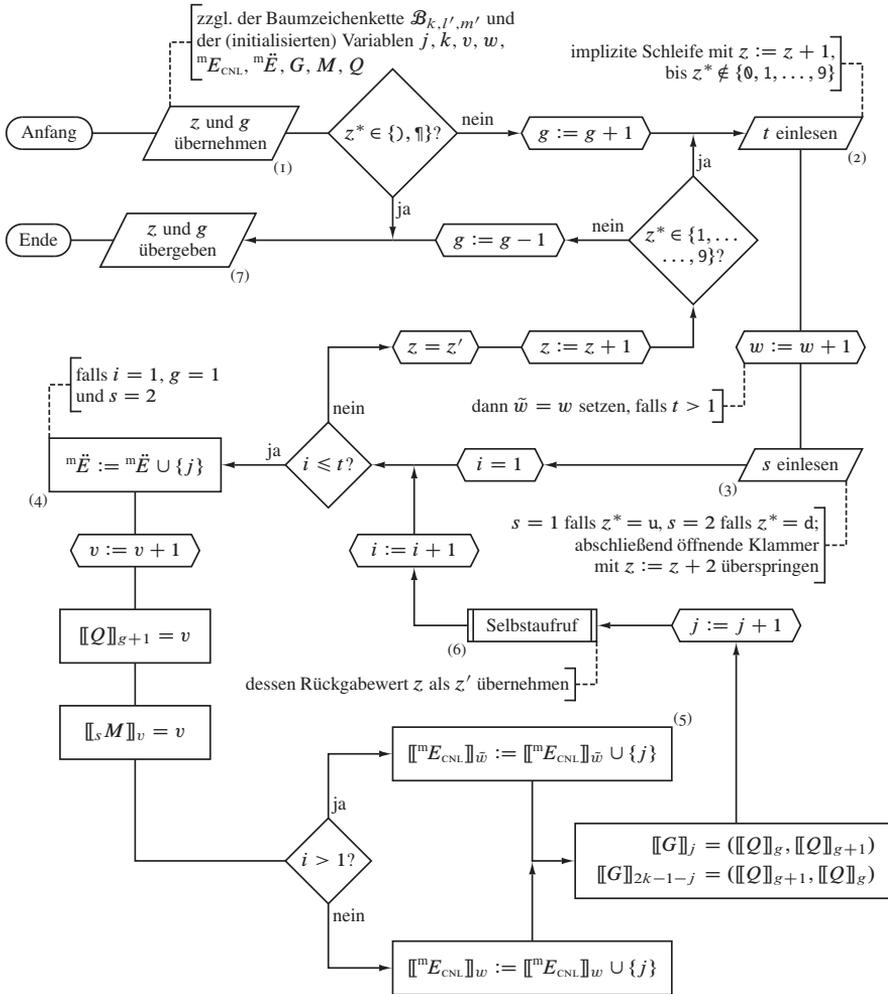


Abbildung 3.8. Ablaufplan des rekursiven Dekodierers der Baumzeichenkette, Flußdiagrammelement (3) in Abbildung 3.6.

Baumzeichenkette, stets

$$\underset{2}{1} \underset{4}{u} (\underset{3}{z^*}, (\underset{5}{z+1})^*, \dots) \quad \text{mit } z^* \in \{\emptyset, 1, \dots, 9\} \quad \text{für } k > 2$$

werden die internen Variablen der rekursiv durch den Butcherbaum laufenden Funktion folgendermaßen initialisiert: $z = 4$ (Zeiger z^* auf das vierte Zeichen der Zeichenkette⁹⁰), $g = 0$ (Ebene des Baumes; konstruktionsbedingt unterhalb der Wurzel beginnend), $j = 1$ (Nummer des Schritts für den Hinsprung, $1 \leq j \leq k - 1$), $v = 1$ (Nummer der betrachteten Ecke, $1 \leq v \leq k$; beginnend mit der Wurzel), $w = 0$ (Nummer der behandelten Eckenklasse), $\llbracket Q \rrbracket_1 = 1$ (Hilfsliste; Nummer der Wurzel an unterster Position) und $\llbracket M \rrbracket_1 = 1 \leftarrow {}_1 M = (1, 0 \mid i = 2, 3, \dots, k)$ (in der Farbebene der \uparrow -Spins ist bereits die Wurzel verzeichnet).

- 1 Rekursionsebene betreten: Hauptprogramm-globale (${}^m E_{\text{CNL}}$, ${}^m \ddot{E}$, G , M : Zugriff schreibend) über *Referenzadressen* und Funktions-lokale ($\mathcal{B}_{k,l',m'}$: lesend; z, g, j, k, v, w ; Q : lesend / schreibend) Ausführungsparameter und Datenstrukturen mit ihrem Wert vom Rahmenprogramm oder der aufrufenden Ebene übernehmen, siehe Flußdiagrammelement (1) in Abbildung 3.8
- 2 Zeichen $z^* = \llbracket \mathcal{B}_{k,l',m'} \rrbracket_z$ prüfen und zu Schritt 11 springen, falls Baumebenen-() oder Zeichenkettenende (¶) erreicht⁹¹; sonst fortsetzen
- 3 Baumebenen-zähler g um eins erhöhen
- 4 Astmultiplizität t einlesen, Element (2): in einer (impliziten) Schleife den Zeiger z so lange um jeweils eine *string*-Position, bis keine Ziffern mehr eingelesen werden, dann Ziffernfolge in Ganzzahl konvertieren
- 5 Kantenklassennummer w um eins erhöhen. Falls der Ast mehr als einmal von der betrachteten Ecke abzweigt ($t > 1$), w in \tilde{w} temporär speichern
- 6 Symbol des Teilchenspins (hier u oder d), im allgemeinen Fall angeregter oder nicht-halbgefüllter Ausgangszustände Besetzungszahl-Kode einlesen und s entsprechend setzen, Element (3). Die folgende öffnende Klammer (()) mit $z := z + 2$ übergehen
- 7 Für $i = 1, 2, \dots, t$: die in Multiplizität „komprimierten“ identischen Äste expandieren
 - 7.1 Sprungnummer j der Menge ${}^m \ddot{E}$ hinzufügen (genauer: anhängen, da Menge geordnet), falls die Kante von der Wurzel abzweigt ($g = 1$) und der Endgitterplatz gemäß Gleichung (3.82) mit einem \downarrow -Spin besetzt ist ($s = 2$), Element (4). Da diese Anweisung nur beim ersten Schleifendurchlauf ausgeführt wird, speichert ${}^m \ddot{E}$ nur einen Repräsentanten einer jeden Klasse der Initialisierungskanten.

⁹⁰Die Behandlung von Zeichenketten als Tupel, siehe Fußnote 30, ermöglicht, den Zugriff auf ein einzelnes Zeichen z^* notationskonsistent zu formulieren, beispielsweise mit $z^* = \llbracket \mathcal{B}_{k,l,m} \rrbracket_z$, den Baum in *string*-Darstellung vorausgesetzt.

⁹¹In der vorgelegten Neufassung wird die unterste Rekursionsebene des *parsers* mit der schließenden Klammer der untersten Butcherbaumebene verlassen.

- 7.2 Eckenzähler v um eins erhöhen
- 7.3 Eckennummer v in der Hilfsliste Q an Position $g + 1$ ablegen
- 7.4 Im zum aktuellen Spin s gehörigen Tupel ${}_sM$ das Teilchen an der aktuellen Position v mit der Nummer seines Ausgangsgitterplatzes, also v versehen
- 7.5 Beim ersten Durchlauf der Schleife ($i = 1$) die Sprungnummer j der (neuen) Kantenklasse w hinzufügen (genauer: anhängen, da Menge geordnet); bei mehrmaligem Schleifendurchlauf ($i > 1$) Kante in der bereits bestehenden Klasse (mit der „Adresse“ \tilde{w}) speichern, Element (5)⁹²
- 7.6 Die Hinsprungkante als Paar aus Anfangs- und Endecke (aus Sicht der Wurzel) an der Position j in G speichern; ebenso mit der Rücksprungkante als Paar aus End- und Anfangsecke verfahren
- 7.7 Sprungnummernzähler j um eins erhöhen
- 7.8 Neue Rekursionsebene durch Selbstaufwurf der Funktion betreten, Element (6), dabei Werte der lokalen und Referenzen auf die globalen Variablen und Datenstrukturen übergeben, siehe Schritt 1; bei Rückkehr die Adresse z des aktuellen Zeichens der Baumzeichenkette als z' entgegennehmen
- 8 Zu $i := i + 1$ übergehen, Schritt 7, falls $i \leq t$; sonst Schleife beenden
- 9 Wert des Zeichenkettenzeigers z aus z' übernehmen und um eins erhöhen
- 10 Zeichen z^* prüfen: falls eine Ziffer vorliegt, zum Einlesen der Multiplizität im Programmpunkt 4 zurückspringen; sonst Baumebenenzähler g um eins verringern
- 11 Rekursionsebene verlassen, Element (7): aktualisierte Ausführungsparameter und Datenstrukturen zurückgeben

3.4.12. *Beispiel.* In exemplarischer Anwendung der Vorbereitungsstufe betrachten wir den Butcherbaum

$$\mathcal{B}_{3,2,18} = \text{☉} \text{ in seiner Zeichenkettendarstellung } \begin{matrix} 1 & 3 & 5 & 7 \\ 1u(2d()) \\ 2 & 4 & 6 & 8 \end{matrix}$$

und berücksichtigen die Vorgaben des Rahmenprogramms, insbesondere $\llbracket Q \rrbracket_1 = 1$ und ${}_1M = (1, 0, 0)$. Tabellarisch (E. steht für Rekursionsebene, S. für Programmschritt) läßt sich der Ablauf folgenderweise zusammenfassen:

E.	S.	Programmschritt-beteiligte Variablen										Bemerkung
		g	j	v	z	z^*	s	i	t	w	\tilde{w}	
1	1	0	1	1	4		-	-	-	0	-	(Anfangswerte)
	2				4	2						z^* zeigt auf Ziffer
	3	1										

Fortsetzung auf der nächsten Seite

⁹²Die Programmstruktur dieses *parsers* erlaubt neben der hier beschriebenen Verwendung auch die Gewinnung anderer graphentheoretisch relevanter Informationen aus den komprimierten Baumzeichenketten. Die abschließende Zuweisung von $w = \tilde{w}$ falls $i = t$ in Flußdiagrammelement (5) bewirkt die baumweite Kantenklasseneinteilung, liefert statt ${}^mE_{\text{CNL}}$ also ${}^mE_{\text{CTW}}$ als Meta-Menge von E_{CTW} .

E.	S.	Programmschritt-beteiligte Variablen										Bemerkung
		g	j	v	z	z^*	s	i	t	w	\tilde{w}	
<i>Fortsetzung</i>												
	4				5	d			2			z^* zeigt auf Spinsymbol
	5							2	1	1		$\tilde{w} = w$, da $t > 1$
	6				7)	2					z^* zeigt auf Ebenenende
	7							1	2			Multiplizitätsschleife
	7.1	1	1				2	1				${}^m\tilde{E} = \{1\}$
	7.2			2								
	7.3	1		2								$\llbracket Q \rrbracket_2 = 2$
	7.4			2			2					$\llbracket {}_2M \rrbracket_2 = 2; {}_2M = (0, 2, 0)$
	7.5		1					1		1		${}^mE_{\text{CNL}} = \{\{1\}\}$, da $i \neq 1$
	7.6	1	1									$\llbracket G \rrbracket_1 = (1, 2), \llbracket G \rrbracket_4 = (2, 1)$
	7.7		2									
	7.8											Selbstaufruf
2	I	1	2	2	7		-	-	-	1	-	(Übernahmewerte)
	2				7)						z^* zeigt auf Ebenenende
	II											Rückkehr, $z' = 7$
1	8							2	2			zurück zu Schritt 7, da $i \leq 2$
	7.1	1	2				2	2				entfällt, da $i = 2$
	7.2			3								
	7.3	1		3								$\llbracket Q \rrbracket_2 = 3$
	7.4			3			2					$\llbracket {}_2M \rrbracket_3 = 3; {}_2M = (0, 2, 3)$
	7.5		2					2		1		${}^mE_{\text{CNL}} = \{\{1, 2\}\}$, da $i > 1$
	7.6	1	2									$\llbracket G \rrbracket_2 = (1, 3), \llbracket G \rrbracket_3 = (3, 1)$
	7.7		3									
	7.8											Selbstaufruf
2	I	1	3	3	7		-	-	-	1	-	(Übernahmewerte)
	2				7)						z^* zeigt auf Ebenenende
	II											Rückkehr, $z' = 7$
1	8							3	2			weiter zu Schritt 9, da $i \neq 2$

Fortsetzung auf der nächsten Seite

E.	S.	Programmschritt-beteiligte Variablen										Bemerkung
		g	j	v	z	z^*	s	i	t	w	\tilde{w}	
<i>Fortsetzung</i>												
	9				8							
	10	0			8)						z^* zeigt auf Ebenenende
	11											Rückkehr, da unterste Ebene

Dem aufrufenden Rahmen stehen nach dem Verlassen der untersten Rekursionsebene die folgenden Daten wunschgemäß zu Verfügung: ${}^mE_{\text{CNL}} = \{\{1, 2\}\}$, ${}^m\vec{E} = \{1\}$, $G = \{(1, 2), (1, 3), (3, 1), (2, 1)\}$ und ${}_2M = (0, 2, 3)$, mit ${}_1M$ zusammengefaßt folglich $M = \{(1, 0, 0), (0, 2, 3)\}$.

3.4.13. *Rekursiver Kern.* Die im rekursiv angelegten eigentlichen Zentralteil des Programmes auszuführende Elektronenpropagation auf den Bethegitterausschnitten erfolgt geradlinig gemäß dem in Abschnitt 3.4.4 entwickelten Formlismus: Aus den Erzeugungs- und Vernichtungsoperatoren der Modellgleichungen (1.1) und (1.2) werden datenverarbeitungstechnisch die entsprechenden Operationen. Der Algorithmus hat, wie im Anhang der Methodenerstveröffentlichung Referenz [19] ausgeführt, zudem nur die „Einhaltung des Pauliverbots zu überwachen“ sowie Sprungversuche von unbesetzten Gitterplätzen „herauszufiltern“ und in beiden Fällen die Behandlung dieses Kantenpermutationsbaumastes abzubrechen.

Da sich das Unterprogramm zunächst am Besetzungszustand, das heißt an der Farbe der Quell-Baumecke $\chi_a := F(a) \in X$ orientiert und dieser den Spin des hüpfenden Teilchens aus X' entnimmt, muß das Verfahren über das zuvor verlangte hinaus Sprünge von doppeltbesetzten Gitterplätzen in separate Zweige des Binärbaums ${}^m\mathcal{D}_{2k-3,s^{(1)}}$ auflösen, welche – anschaulich formuliert – den zugrundeliegenden Permutationsbaum ${}^m\mathcal{P}_{2k-3,r^{(1)}}$ überlagern. Konkret werden diese Forderungen mit $\chi_b := F(b) \in X$ als Farbe der Ziel-Baumecke folgendermaßen umgesetzt:

- I Ausführungparameter vom aufrufenden Rahmenprogramm oder dem vorausgegangenen Rekursionsschritt übernehmen, teilweise in Form der Referenzadressen: Neben den Konstanten (Störungsrechnungsordnung n beziehungsweise Graphenordnung k) und den zuvor erzeugten oder bearbeiteten Datenstrukturen (insbesondere für $H^{(n)}$ (Zugriff lesend/schreibend) und \tilde{G} (lesend) sowie für die Zwischensumme zu a_n (schreibend) des bearbeitenden Rechenprozeßfadens), Hüpfschrittnummer j , Permutationsbaumebenenzähler $p = j$, das aus $|M|^{(j-1)}$ extrahierte Hüpfarbsymbol χ_j , Quellgitterplatz $a = \llbracket \tilde{G} \rrbracket_{j,1}$ und Zielgitterplatz $b = \llbracket \tilde{G} \rrbracket_{j,2}$, siehe Flußdiagrammelement (1) in Abbildung 3.9. Die Übernahme der Gitterplätze a und b stellt durch Vermeidung zweier Feldzugriffe vor Element (2) und folgenden bereits eine naheliegende Laufzeitverkürzung dar.

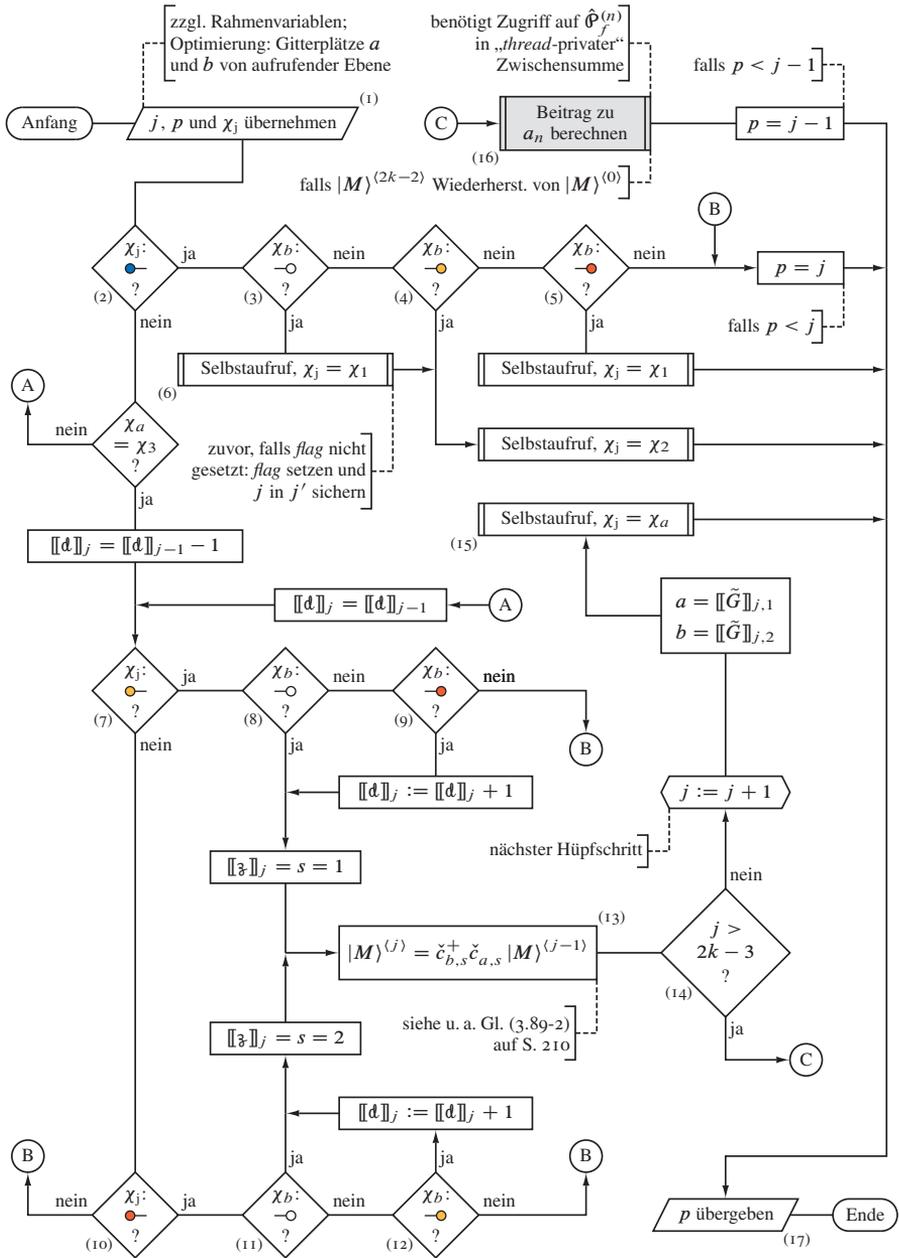


Abbildung 3.9. Ablaufplan des rekursiven Kernes, Flußdiagrammelement (8) in Abbildung 3.6. Das Unterprogramm zu hiesigem Element (16) ist in Abbildung 3.10 (Abschn. 3.5 ab S. 246) ausgeführt.

- 2 Sprungausführung von einem doppeltbesetzten Quellgitterplatz, hüpfender „Spin“ χ_j trägt (formal) die Farbe $\chi_3 = \blacksquare$, fallweise folgendermaßen auflösen:
 - \circ , Sprung zu einem unbesetzten Zielgitterplatz, $\chi_b = \chi_0 = \square$, Elemente (2) und (3): Falls *flag* nicht gesetzt, vor Element (6) *flag* setzen und j in j' sichern, dann Selbstaufwurf mit Sprung des \uparrow -Spins, $\chi_j = \chi_1 = \blacksquare$, nach Rückkehr zu dieser Rekursionsverzweigung Selbstaufwurf mit Sprung des \downarrow -Spins, $\chi_j = \chi_2 = \blacksquare$, dann nach erneuter Rückkehr Programmende durch Übergang zu Punkt 12
 - \circ , Sprung zu einem Zielgitterplatz mit einem \uparrow -Spin, $\chi_b = \chi_1$, Elemente (2) und (4): Selbstaufwurf mit Pauli-erlaubtem Sprung des \downarrow -Spins, $\chi_j = \chi_2$, nach Rückkehr zu dieser Rekursionsverzweigung Programmende durch Übergang zu Punkt 12
 - \circ , Sprung zu einem Zielgitterplatz mit einem \downarrow -Spin, $\chi_b = \chi_2$, Elemente (2) und (5): Selbstaufwurf mit Pauli-erlaubtem Sprung des \uparrow -Spins, $\chi_j = \chi_1$, nach Rückkehr zu dieser Rekursionsverzweigung Programmende durch Übergang zu Punkt 12
 - \bullet , Sprung zu einem doppeltbesetzten Zielgitterplatz, $\chi_b = \chi_3$: kraft des Pauliverbots Programmende durch Übergang zu Punkt 12, zuvor Änderung der Permutationsbaumebene auf $p = j$, falls $p < j$
- 3 Doppelbesetzungszahl vom vorausgegangenen Hüpfschritt übernehmen, $\llbracket d \rrbracket_j = \llbracket d \rrbracket_{j-1}$, oder, falls a eine Doppelbesetzung trägt, $\chi_a = \chi_3$, Doppelbesetzungszahl übernehmen und dabei um eins verringern, $\llbracket d \rrbracket_j = \llbracket d \rrbracket_{j-1} - 1$
- 4 Wenn es sich bei dem hüpfenden Spin mit der Farbe χ_j um einen \uparrow -Spin handelt, folgende Fälle unterschieden:
 - \circ , Sprung zum unbesetzten Zielgitterplatz, $\chi_b = \chi_0$, Elemente (7) und (8): Hüpfspintupelement des Sprunges auf eins setzen, $\llbracket \mathfrak{z} \rrbracket_j = s = 1$, und zu Programmpunkt 7 übergehen
 - \circ , Sprung zu einem \downarrow -Spin-besetzten Zielgitterplatz, $\chi_b = \chi_2$, Elemente (7) und (9): Doppelbesetzungszahl um eins erhöhen, $\llbracket d \rrbracket_j := \llbracket d \rrbracket_j + 1$, Hüpfspintupelement des Sprunges auf eins setzen, $\llbracket \mathfrak{z} \rrbracket_j = s = 1$, und zu Programmpunkt 7 übergehen
 - $\circ \vee \bullet$, Sprung zu einem \uparrow -Spin-, $\chi_b = \chi_1$, oder doppeltbesetzten Zielgitterplatz, $\chi_b = \chi_3$: kraft des Pauliverbots Programmende durch Übergang zu Punkt 12; zuvor die Permutationsbaumebene auf $p = j$ ändern, falls $p < j$
- 5 Wenn es sich bei dem hüpfenden Spin mit der Farbe χ_j um einen \downarrow -Spin handelt, folgende Fälle unterschieden:
 - \circ , Sprung zu einem unbesetzten Zielgitterplatz, $\chi_b = \chi_0$, Elemente (10) und (11): Hüpfspintupelement des Sprunges auf zwei setzen, $\llbracket \mathfrak{z} \rrbracket_j = s = 2$, und zu Programmpunkt 7 übergehen

- , Sprung zu einem \uparrow -Spin-besetzten Zielgitterplatz, $\chi_b = \chi_1$, Elemente (10) und (12): Doppelbesetzungszahl um eins erhöhen, $[[d]]_j := [[d]]_j + 1$, Hüpfspintupelement des Sprunges auf zwei setzen, $[[z]]_j = s = 2$, und zu Programmpunkt 7 übergehen
 - \vee •—•, Sprung zu einem \downarrow -Spin-, $\chi_b = \chi_2$, oder doppelbesetzten Zielgitterplatz, $\chi_b = \chi_3$: kraft des Pauliverbots Programmende durch Übergang zu Punkt 12; zuvor die Permutationsbaumebene auf $p = j$ ändern, falls $p < j$
- 6 Falls im Programmfluß in den Elementen (2), (7) oder (10) keine Abzweigung beschritten wurde, also der hüpfende „Spin“ χ_j als verbleibende Möglichkeit (formal) die Farbe χ_3 eines Loches trägt, Programm mit Schritt 12 beenden, zuvor Änderung der Permutationsbaumebene auf $p = j$, falls $p < j$
- 7 Neuen Zustand $|M\rangle^{(j)}$ in Element (13) durch Transfer des Elektrons mit dem Spin s vom Gitterplatz a zum Gitterplatz b aus dem vorherigen Zustand $|M\rangle^{(j-1)}$ erzeugen, siehe Gleichungen (3.89-1) für $j = 1$ und (3.89-2) für $j > 1$:

$$|M\rangle^{(j)} = \check{c}_{b,s}^+ \check{c}_{a,s} |M\rangle^{(j-1)}$$

Die Definition der Transferoperation, Details siehe Gleichung (3.71), gewährleistet durch die Verschiebung der Teilchennummer, daß die Anzahl der Teilchenvertauschungen zwischen Ausgangs- und Endzustand ermittelt werden kann, siehe Abschnitt 3.5. Beispielhaft ist die konkrete Umsetzung dieses für das gesamte Konzept wesentlichen Schrittes auf Ebene des „Maschinenprogrammes“ für Datenverarbeitungsanlagen in Abschnitt 3.4.4 beschrieben.

- 8 Falls alle Hüpf Schritte bereits ausgeführt wurden (Sprungzähler $j > 2k - 3$), im Element (16) Hüpfsequenzbeitrag zum Reihenkoeffizienten berechnen, siehe Abschnitt 3.5, Permutationsbaumebene auf $p = j - 1$ setzen, wenn $p < j - 1$, und Programm durch Übergang zu Punkt 12 beenden
- 9 Hüpfschrittnummer um eins erhöhen, $j := j + 1$
- 10 Neue Quell- und Zielgitterplatznummer der Sprungkante entnehmen: $a = [[\tilde{G}]]_{j,1}$, $b = [[\tilde{G}]]_{j,2}$
- 11 Neue Rekursions- beziehungsweise Permutationsbaumebene durch Selbstaufwurf mit $\chi_j = \chi_a$ betreten, Element (15)
- 12 Aktuelle Rekursionsebene verlassen, Element (17): Permutationsbaumebenenähler p und Hüpfschrittnummer j sowie gegebenenfalls deren Sicherungsvariable j' nebst zugehörigem *flag* zurückgeben

3.4.14. *Beispiel.* Die Arbeitsweise des Kernes in Wechselwirkung mit der permutationsgenerierenden Schleife 4.3.4 des Rahmenprogramms, siehe Abschnitt 3.4.9, veranschaulichen wir uns anhand der vollständigen Sequenzierung des Butcherbaums

$$\mathcal{B}_{3,1,27} = \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \circ \end{array},$$

wobei unter anderem (siehe unten) $|M\rangle^{(0)} = |(1, 0, 0), (0, 2, 3)\rangle^{(0)}$ und somit $\llbracket d \rrbracket_0 = 0$ vom Rahmen übernommen werden. Analog zum Beispiel der Vorbereitungsstufe, Abschnitt 3.4.12, wählen wir eine tabellarische Darstellung der umfangreichen Informationen des Ablaufs (E. steht für Rekursionsebene, S. für Programmschritt, Nr. für laufende Anweisungs- beziehungsweise Zeilennummer).

E.	S.	Nr.	Programmschritt-beteiligte Variablen							$\llbracket d \rrbracket_j$	$\llbracket \checkmark \rrbracket_j$	Bemerkung
			j	χ_j	s	a	b	$ _s M\rangle^{(j)}$				
$\tilde{G} = \{(1, 2), (2, 3), (3, 2), (2, 1)\}$ aus ${}^m \tilde{G}_0 = \{1, 2, 3, 4\}$; $p = j = 1$												
1	I	I	1	○	–	1	2	–	–	–	(Anfangswerte)	
		2	2	○							$\chi_j \neq \blacksquare$	
		3	3	1			1		0		$\llbracket d \rrbracket_0$, da $\chi_a = \blacksquare$	
		4	4	1	○	1		2		1	1	○—○
		7	5	1		1	1	2	$ _1(0, 1, 0)\rangle^{(1)}$			$\checkmark_{2,1}^+ \checkmark_{1,1} _1(1, 0, 0)\rangle^{(0)}$
		8	6	1								$j \neq 3$
		9	7	2								
		10	8				2	3				
		11	9		●							Selbstaufruf
	2	I	10	2	●	–	2	3	–	–	–	(Übernahmewerte)
			11		○			3				●—○: Selbstaufruf
3	I	12	2	○	–	2	3	–	–	–	(Übernahmewerte)	
		13		○							$\chi_j \neq \blacksquare$	
		14	2				2		0		$\llbracket d \rrbracket_1 - 1$, da $\chi_a = \blacksquare$	
		15	2	○	1		3		1	1	○—○	
		7	16	2		1	2	3	$ _1(0, 0, 1)\rangle^{(2)}$			$\checkmark_{3,1}^+ \checkmark_{2,1} _1(0, 1, 0)\rangle^{(1)}$
		8	17	2								$j \neq 3$
		9	18	3								
		10	19				3	2				
		11	20		●							Selbstaufruf
	4	I	21	3	●	–	3	2	–	–	–	(Übernahmewerte)

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung	
			j	χ_j	s	a	b	$ _s M^{(j)}$	$[[d]]_j$	$[[z]]_j$		
<i>Fortsetzung</i>												
	2	22		○			2				●→○: Selbstaufruf	
5	I	23	3	○	–	3	2	–	–	–	(Übernahmewerte)	
	2	24		○							$\chi_j \neq \blacksquare$	
	3	25	3			3			0		$[[d]]_2 - 1$, da $\chi_a = \blacksquare$	
	4	26	3	○	1		2		1	1	○→●	
	7	27	3		1	3	2	$ _1(0, 1, 0)^{(3)}$			$\check{c}_{2,1}^+ \check{c}_{3,1} _1(0, 0, 1)^{(2)}$	
	8	28	3								$j \neq 3$	
	9	29	4									
	10	30					2	1				
	11	31			●							Selbstaufruf
	6	I	32	4	●	–	2	1	–	–	–	(Übernahmewerte)
		2	33		○			1				●→○: 1. Selbstaufruf
7	I	34	4	○	–	2	1	–	–	–	(Übernahmewerte)	
	2	35		○							$\chi_j \neq \blacksquare$	
	3	36	4			2			0		$[[d]]_3 - 1$, da $\chi_a = \blacksquare$	
	4	37	4	○	1		1		0	1	○→○	
	7	38	4		1	2	1	$ _1(1, 0, 0)^{(4)}$			$\check{c}_{1,1}^+ \check{c}_{2,1} _1(0, 1, 0)^{(3)}$	
	8	39	4								Sequenz gültig, a_n -Beitrag berechnen	
											$j > 3$	
	12	40	4								Rückkehr, $p = 3$	
6	2	41		○			1				●→○: 2. Selbstaufruf	
7	I	42	4	○	–	2	1	–	–	–	(Übernahmewerte)	
	2	43		○							$\chi_j \neq \blacksquare$	
	3	44	4			2			0		$[[d]]_3 - 1$, da $\chi_a = \blacksquare$	
	4	45		○							$\chi_j \neq \blacksquare$	
	5	46	4	○	2		1		0	2	○→○	
	7	47	4		2	2	1	$ _2(2, 0, 3)^{(4)}$			$\check{c}_{1,2}^+ \check{c}_{2,2} _2(0, 2, 3)^{(3)}$	
											Sequenz nicht gültig	
	8	48	4								$j > 3$	

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung	
			j	χ_j	s	a	b	$ _s M^{(j)}$	$\llbracket d \rrbracket_j$	$\llbracket \tilde{z} \rrbracket_j$		
<i>Fortsetzung</i>												
	I2	49	4								Rückkehr, $p = 3$	
$\tilde{G} = \{(1, 2), (2, 3), (2, 1), (3, 2)\}$ aus ${}^m\tilde{G}_1 = \{1, 2, 4, 3\}$; $p = j = 3$												
1	I	50	3	●—	—	2	1	—	—	—	(Übernahmewerte)	
	2	51		●—							$\chi_j \neq \blacksquare$	
	3	52	3			2			1		$\llbracket d \rrbracket_2$, da $\chi_a = \blacksquare$	
	4	53		●—							$\chi_j \neq \blacksquare$	
	5	54	3	●—	2		1		1	1	●—○	
	7	55	3		2	2	1	$ _2(2, 0, 3)^{(3)}$			$\check{c}_{1,2}^+, \check{c}_{2,2} _2(0, 2, 3)^{(2)}$	
	8	56	3								$j \neq 3$	
	9	57	4									
	I0	58				3	2					
	I1	59		●—								Selbstaufruf
	2	I	60	4	●—	—	3	2	—	—	—	(Übernahmewerte)
2		61		●—			2				●—○: 1. Selbstaufruf	
3	I	62	4	●—	—	3	2	—	—	—	(Übernahmewerte)	
	2	63		●—							$\chi_j \neq \blacksquare$	
	3	64	4			3			0		$\llbracket d \rrbracket_3 - 1$, da $\chi_a = \blacksquare$	
	4	65	4	●—	1		2		0	1	●—○	
	7	66	4		1	3	2	$ _1(0, 1, 0)^{(4)}$			$\check{c}_{2,1}^+, \check{c}_{3,1} _1(0, 0, 1)^{(3)}$	
	8	67	4					Sequenz nicht gültig			$j > 3$	
	I2	68	4								Rückkehr, $p = 3$	
2	2	69		●—			2				●—○: 2. Selbstaufruf	
3	I	70	4	●—	—	3	2	—	—	—	(Übernahmewerte)	
	2	71		●—							$\chi_j \neq \blacksquare$	
	3	72	4			3			0		$\llbracket d \rrbracket_3 - 1$, da $\chi_a = \blacksquare$	
	4	73		●—							$\chi_j \neq \blacksquare$	
	5	74	4	●—	2		2		0	2	●—○	

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung
			j	χ_j	s	a	b	$ _s M^{(j)}$	$[[d]]_j$	$[[z]]_j$	
<i>Fortsetzung</i>											
	7	75	4		2	3	2	$ _2(2, 3, 0)^{(4)}$			$\check{c}_{2,2}^+ \check{c}_{3,2} _2(2, 0, 3)^{(3)}$
	8	76	4					Sequenz nicht gültig			$j > 3$
	I2	77	4								Rückkehr, $p = 3$
$\tilde{G} = \{(1, 2), (3, 2), (2, 3), (2, 1)\}$ aus ${}^m\tilde{G}_2 = \{1, 3, 2, 4\}$; $p = j = v = 2$											
1	I	78	2		-	3	2	-	-	-	(Übernahmewerte)
	2	79									$\chi_j \neq \blacksquare$
	3	80	2			3			1		$[[d]]_1$, da $\chi_a = \blacksquare$
	4	81									$\chi_j \neq \blacksquare$
	5	82	2				2				
	I2	83									
$\tilde{G} = \{(1, 2), (2, 1), (2, 3), (3, 2)\}$ aus ${}^m\tilde{G}_4 = \{1, 4, 2, 3\}$; $p = j = 2$											
1	I	84	2		-	2	1	-	-	-	(Übernahmewerte)
	2	85					1				\circ : I. Selbstaufruf
2	I	86	2		-	2	1	-	-	-	(Übernahmewerte)
	2	87									$\chi_j \neq \blacksquare$
	3	88	2			2			0		$[[d]]_1 - 1$, da $\chi_a = \blacksquare$
	4	89	2		1	1			0	1	\circ
	7	90	2		1	2	1	$ _1(1, 0, 0)^{(2)}$			$\check{c}_{1,1}^+ \check{c}_{2,1} _1(0, 1, 0)^{(1)}$
	8	91	2								$j \neq 3$
	9	92	3								
I0	93				2	3					
I1	94										Selbstaufruf
3	I	95	3		-	2	3	-	-	-	(Übernahmewerte)
	2	96									$\chi_j \neq \blacksquare$
	3	97	3			2			0		$[[d]]_2$, da $\chi_a = \blacksquare$
	4	98									$\chi_j \neq \blacksquare$
	5	99	3				3				

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung	
			j	χ_j	s	a	b	$ _s M^{(j)}$	$[[d]]_j$	$[[z]]_j$		
<i>Fortsetzung</i>												
	I2	I00									Abbruch, $p = 3$	
1	2	I01		●-			1				●-○: 2. Selbstaufruf	
2	1	I02	2	●-	-	2	1	-	-	-	(Übernahmewerte)	
	2	I03		●-							$\chi_j \neq \blacksquare$	
	3	I04	2			2			0		$[[d]]_1 - 1$, da $\chi_a = \blacksquare$	
	4	I05		●-							$\chi_j \neq \blacksquare$	
	5	I06	2	●-	2		1		0	2	●-○	
	7	I07	2		2	2	1	$ _2(2, 0, 3)^{(2)}$			$\check{c}_{1,2}^+ \check{c}_{2,2} _2(0, 2, 3)^{(1)}$	
	8	I08	2								$j \neq 3$	
	9	I09	3									
	10	I10				2	3					
	11	I11		○-							Selbstaufruf	
	3	1	I12	3	○-	-	2	3	-	-	-	(Übernahmewerte)
2		I13		○-							$\chi_j \neq \blacksquare$	
3		I14	3			2			0		$[[d]]_2$, da $\chi_a = \blacksquare$	
4		I15	3	○-	1		3		1	1	○-●	
7		I16	3		1	2	3	$ _1(0, 0, 1)^{(3)}$			$\check{c}_{3,1}^+ \check{c}_{2,1} _1(0, 1, 0)^{(2)}$	
8		I17	3								$j \neq 3$	
9		I18	4									
10		I19				3	2					
11		I20		●-							Selbstaufruf	
4		1	I21	4	●-	-	3	2	-	-	-	(Übernahmewerte)
		2	I22		○-			2				●-○: 1. Selbstaufruf
5	1	I23	4	○-	-	3	2	-	-	-	(Übernahmewerte)	
	2	I24		○-							$\chi_j \neq \blacksquare$	
	3	I25	4			3			0		$[[d]]_3 - 1$, da $\chi_a = \blacksquare$	
	4	I26	4	○-	1		2		0	1	○-○	
	7	I27	4		1	3	2	$ _1(0, 1, 0)^{(4)}$			$\check{c}_{2,1}^+ \check{c}_{3,1} _1(0, 0, 1)^{(3)}$	

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung
			j	χ_j	s	a	b	$ _s M^{(j)}$	$\llbracket d \rrbracket_j$	$\llbracket \mathfrak{z} \rrbracket_j$	
<i>Fortsetzung</i>											
	8	128	4	Sequenz nicht gültig							$j > 3$
	12	129	4								Rückkehr, $p = 3$
4	2	130		●—○			2				●—○: 2. Selbstaufruf
5	1	131	4	●—○	—	3	2	—	—	—	(Übernahmewerte)
	2	132		●—○							$\chi_j \neq \blacksquare$
	3	133	4			3			0		$\llbracket d \rrbracket_3 - 1$, da $\chi_a = \blacksquare$
	4	134		●—○							$\chi_j \neq \blacksquare$
	5	135	4	●—○	2		2		0	2	●—○
	7	136	4		2	3	2	$ _2(2, 3, 0)^{(4)}$			$\check{c}_{2,2}^+ \check{c}_{3,2}^- _2(2, 0, 3)^{(3)}$
	8	137	4	Sequenz nicht gültig							$j > 3$
	12	138	4								Rückkehr, $p = 3$
$\tilde{G} = \{(1, 2), (2, 1), (3, 2), (2, 3)\}$ aus ${}^m \tilde{G}_5 = \{1, 4, 3, 2\}$; $p = j = j' = 2$											
1	1	139	2	●—○	—	2	1	—	—	—	(Übernahmewerte)
	2	140		○—●			1				●—○: 1. Selbstaufruf
2	1	141	2	○—●	—	2	1	—	—	—	(Übernahmewerte)
	2	142		○—●							$\chi_j \neq \blacksquare$
	3	143	2			2			0		$\llbracket d \rrbracket_1 - 1$, da $\chi_a = \blacksquare$
	4	144	2	○—●	1		1		0	1	○—●
	7	145	2		1	2	1	$ _1(1, 0, 0)^{(2)}$			$\check{c}_{1,1}^+ \check{c}_{2,1}^- _1(0, 1, 0)^{(1)}$
	8	146	2								$j \neq 3$
	9	147	3								
	10	148				3	2				
11	149		●—○								Selbstaufruf
3	1	150	3	●—○	—	3	2	—	—	—	(Übernahmewerte)
	2	151		●—○							$\chi_j \neq \blacksquare$
	3	152	3			3			0		$\llbracket d \rrbracket_2$, da $\chi_a = \blacksquare$
	4	153		●—○							

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung
			j	χ_j	s	a	b	$ _s M)^{(j)}$	$\llbracket d \rrbracket_j$	$\llbracket \tilde{z} \rrbracket_j$	
<i>Fortsetzung</i>											
	5	154	3	●—			2				●—○
	12	155									Abbruch, $p = 3$
1	2	156		●—			1				●—○: 2. Selbstaufruf
2	1	157	2	●—	—	2	1	—	—	—	(Übernahmewerte)
	2	158		●—							$\chi_j \neq \blacksquare$
	3	159	2			2			0		$\llbracket d \rrbracket_1 - 1$, da $\chi_a = \blacksquare$
	4	160		●—							$\chi_j \neq \blacksquare$
	5	161	2	●—	2		1		0	2	●—○
	7	162	2		2	2	1	$ _2(2, 0, 3))^{(2)}$			$\check{c}_{1,2}^+ \check{c}_{2,2} _2(0, 2, 3))^{(1)}$
	8	163	2								$j \neq 3$
	9	164	3								
	10	165				3	2				
	11	166		●—							Selbstaufruf
	3	1	167	3	●—	—	3	2	—	—	—
2		168		●—							$\chi_j \neq \blacksquare$
3		169	3			3			0		$\llbracket d \rrbracket_2$, da $\chi_a = \blacksquare$
4		170		●—							$\chi_j \neq \blacksquare$
5		171	3	●—	2		2		1	2	●—○
7		172	3		2	3	2	$ _2(2, 3, 0))^{(3)}$			$\check{c}_{2,2}^+ \check{c}_{3,2} _2(2, 0, 3))^{(2)}$
8		173	3								$j \neq 3$
9		174	4								
10		175				2	3				
11		176		●—							Selbstaufruf
4		1	177	4	●—	—	2	3	—	—	—
	2	178		●—			3				●—○: 1. Selbstaufruf
5	1	179	4	●—	—	2	3	—	—	—	(Übernahmewerte)
	2	180		●—							$\chi_j \neq \blacksquare$
	3	181	4			2			0		$\llbracket d \rrbracket_3 - 1$, da $\chi_a = \blacksquare$

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen							Bemerkung	
			j	χ_j	s	a	b	$ _s M^{(j)}$	$[[d]]_j$		$[[z]]_j$
<i>Fortsetzung</i>											
	4	182	4	○-	1		3		0	1	○-○
	7	183	4		1	2	3	$ _1(0, 0, 1)^{(4)}$			$\check{c}_{3,1}^+ \check{c}_{2,1} _1(0, 1, 0)^{(3)}$
	8	184	4					Sequenz nicht gültig			$j > 3$
	12	185	4								Rückkehr, $p = 3$
4	2	186		●-			3				●-○: 2. Selbstaufruf
5	1	187	4	●-	-	2	3	-	-	-	(Übernahmewerte)
	2	188		●-							$\chi_j \neq \blacksquare$
	3	189	4			2			0		$[[d]]_3 - 1$, da $\chi_a = \blacksquare$
	4	190		●-							$\chi_j \neq \blacksquare$
	5	191	4	●-	2		3		0	2	●-○
	7	192	4		2	2	3	$ _2(2, 0, 3)^{(4)}$			$\check{c}_{3,2}^+ \check{c}_{2,2} _2(2, 3, 0)^{(3)}$
	8	193	4					Sequenz nicht gültig			$j > 3$
	12	194	4								Rückkehr, $p = 3$

Der vorausgegangenen Befehlsabfolge des Kernes liegen die folgenden Aufruf- und Rückgabewerte des Rahmens zugrunde (Sequenzgleichungen siehe unten):

u	${}^m \tilde{G}_u$	j	j'	p	p'	v	$flag?$	Sequenzen/Bemerkung
0	{1, 2, 3, 4}	1	4	3	1	-	ja	(3.96-1), (3.96-2)
1	{1, 2, 4, 3}	3	4	3	1	2	ja	(3.96-3), (3.96-4)
2	{1, 3, 2, 4}	2	-	2	2	-	nein	(3.96-5)
3	{1, 3, 4, 2}	-	-	-	-	-	—	übersprungen, da $p' = 2$
4	{1, 4, 2, 3}	2	2	3	1	-	ja	(3.96-6), (3.96-7), (3.96-8)
5	{1, 4, 3, 2}	2	2	3	1	-	ja	(3.96-9), (3.96-10), (3.96-11)
6	{2, 1, 3, 4}	-	-	-	-	-	—	Abbruch, da $u = 6$

Das Ergebnis der Propagationsprozedur läßt sich in Anlehnung an Gleichung (1.27) in *Sequenzform* schreiben. Dabei werden über den Pfeilen die Transferoperationen und unter den Pfeilen die zugehörigen Anweisungsnummern aus obiger Tabelle notiert; eingerahmte Bäume stehen für den Ausgangszustand, Auslassungspunkte kennzeichnen

die Zustandsübernahme aus einer vorausgegangenen Permutation, schraffierte Flächen vertreten Pauli-verbotene Zwischenzustände. Man beachte, daß nur die Hüpfsequenz in Gleichungszeile (3.96-1) („nullte“ Kantenpermutation, also die nach unserer Baumdefinition natürliche Abfolge der Kanten, ${}^m\tilde{G}_0$) zur Grundzustandsenergie, hierfür beispielhaft in Abschnitt 3.5.2 berechnet, beiträgt. Die Permutation ${}^m\tilde{G}_3 = \{1, 3, 4, 1\}$ mit der Kantenabfolge $\tilde{G} = \{(1, 2), (3, 2), (2, 1), (2, 3)\}$ entfällt gänzlich, da bereits der Hüffschritt zweiter Ebene im vorausgehenden Permutationsbaumast ${}^m\tilde{G}_2 = \llbracket \mathbb{F}({}^m\mathcal{P}_{3,1}) \rrbracket_3$ nicht ausführbar ist, siehe Gleichung (3.96-5) und Anweisung 82.

$$\tilde{G} = \{(1, 2), (2, 3), (3, 2), (2, 1)\} \text{ aus } {}^m\tilde{G}_0 = \{1, 2, 3, 4\}:$$

$$\begin{array}{c} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[5]{\check{c}_{2,1}^+ \check{c}_{1,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[16]{\check{c}_{3,1}^+ \check{c}_{2,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[27]{\check{c}_{2,1}^+ \check{c}_{3,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[38]{\check{c}_{1,1}^+ \check{c}_{2,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \end{array} \quad (3.96-1)$$

$$\dots \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[47]{\check{c}_{1,2}^+ \check{c}_{2,2}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \quad (3.96-2)$$

$$\tilde{G} = \{(1, 2), (2, 3), (2, 1), (3, 2)\} \text{ aus } {}^m\tilde{G}_1 = \{1, 2, 4, 3\}:$$

$$\dots \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[55]{\check{c}_{1,2}^+ \check{c}_{2,2}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[67]{\check{c}_{2,1}^+ \check{c}_{3,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \quad (3.96-3)$$

$$\dots \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[76]{\check{c}_{2,2}^+ \check{c}_{3,2}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \quad (3.96-4)$$

$$\tilde{G} = \{(1, 2), (3, 2), (2, 3), (2, 1)\} \text{ aus } {}^m\tilde{G}_2 = \{1, 3, 2, 4\}:$$

$$\dots \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[82]{\check{c}_{2,2}^+ \check{c}_{3,2}} \text{///} \quad (3.96-5)$$

$$\tilde{G} = \{(1, 2), (2, 1), (2, 3), (3, 2)\} \text{ aus } {}^m\tilde{G}_4 = \{1, 4, 2, 3\}:$$

$$\dots \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[90]{\check{c}_{1,1}^+ \check{c}_{2,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[99]{\check{c}_{3,2}^+ \check{c}_{2,2}} \text{///} \quad (3.96-6)$$

$$\dots \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[107]{\check{c}_{1,2}^+ \check{c}_{2,2}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[116]{\check{c}_{3,1}^+ \check{c}_{2,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[127]{\check{c}_{2,1}^+ \check{c}_{3,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \quad (3.96-7)$$

$$\dots \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \xrightarrow[136]{\check{c}_{2,2}^+ \check{c}_{3,2}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \end{array} \quad (3.96-8)$$

$$\tilde{G} = \{(1, 2), (2, 1), (3, 2), (2, 3)\} \text{ aus } {}^m\tilde{G}_5 = \{1, 4, 3, 2\}:$$

$$\dots \quad \begin{array}{c} \bullet \\ \circ \end{array} \xrightarrow[\text{145 (90)}]{\check{c}_{1,1}^+ \check{c}_{2,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \circ \end{array} \xrightarrow[\text{154}]{\check{c}_{2,2}^+ \check{c}_{3,2}} \text{///} \quad (3.96-9)$$

$$\dots \quad \begin{array}{c} \bullet \\ \bullet \\ \circ \end{array} \xrightarrow[\text{162 (107)}]{\check{c}_{1,2}^+ \check{c}_{2,2}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \\ \circ \end{array} \xrightarrow[\text{172}]{\check{c}_{2,1}^+ \check{c}_{3,1}} \begin{array}{c} \circ \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \circ \end{array} \xrightarrow[\text{183}]{\check{c}_{3,1}^+ \check{c}_{2,1}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \circ \end{array} \quad (3.96-10)$$

$$\dots \quad \begin{array}{c} \circ \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \circ \end{array} \xrightarrow[\text{193}]{\check{c}_{3,2}^+ \check{c}_{2,2}} \begin{array}{c} \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \circ \end{array} \quad (3.96-11)$$

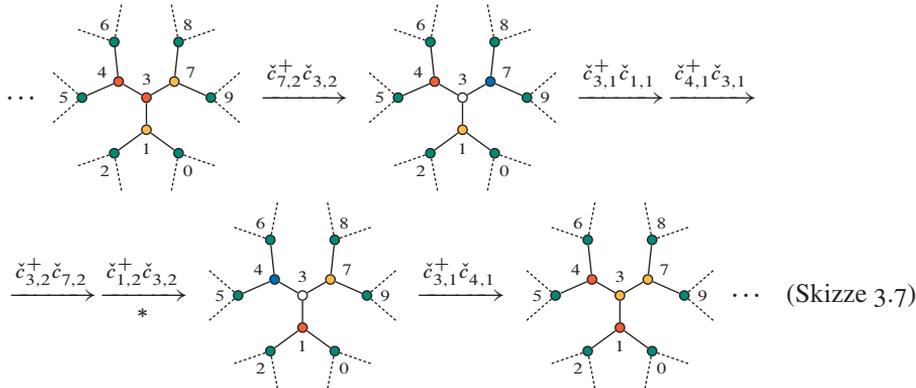
Die Gleichungszeilen (3.96-1) und (3.96-2), (3.96-3) und (3.96-4), (3.96-7) und (3.96-8) sowie (3.96-10) und (3.96-11) gehören jeweils paarweise zwar zum selben Ast des Permutationsbaumes, liegen jedoch auf verschiedenen Zweigen des mit ${}^m\mathcal{P}_{3,1}$ „verwobenen“ Binärbaumes ${}^m\mathcal{D}_{3,1}$. Eine weitere Binärbaumgabelung tritt in den besagten Anweisungen 90 und 107 auf und trennt die Zeile (3.96-6) von den Zeilen (3.96-7) und (3.96-8); diese wiederholt sich in den drei Formeln der letzten Kantenpermutation ${}^m\tilde{G}_5$: Zeile (3.96-9) beginnt effektiv mit dem aus Anweisung 90 hervorgehenden Zustand, die Zeilen (3.96-10) und (3.96-11) greifen genaugenommen auf den auf Anweisung 107 folgenden Zwischenzustand zurück.⁹³

3.4.15. Präemptive Hüpfspinwahl. Bei der Auflösung einer Doppelbesetzung zu einem Loch besteht die Wahl zwischen zwei (zunächst) äquivalenten Zweigen des Binärbaums, und die zuvor dargelegte Umsetzung des Algorithmus' beschreibt beide durch sequenzielle Selbstaufrufe, symbolisch $\bullet \circ \rightsquigarrow \circ \bullet \wedge \bullet \circ$, mit jeweils einem der beiden möglichen Hüpfspins mit dem Farbsymbol χ_j , siehe Programmschritt 2. Der im Verlauf der Elektronenpropagation eingeschlagene Weg auf dem Binärbaum ${}^m\mathcal{D}_{2k-3,s^{(1)}}$ ist jedoch vom gerade betrachteten Permutationsbaumast, also der leitenden Kantenpermutation ${}^m\tilde{G}_u = \llbracket \mathbb{F}({}^m\mathcal{P}_{2k-3,r^{(1)}}) \rrbracket_{u+1}$ nicht unabhängig, da eine Kante – unabdingbar für die Wiederherstellung des Ausgangszustands – stets in beide Richtungen von Elektronen gleichen Spins übersprungen werden muß.

Diese Bedingung tritt, neben leicht erkennbaren Fällen wie etwa in Anweisung 55 in der Sequenzgleichung (3.96-3), auch in nicht so offensichtlicher Weise bei unmittelbarer Ausführung des Hüpfens eines Elektrons von einem einfach besetzten Quell- auf einen mehr als zweifach koordinierten Zielgitterplatz („Ausweichstellen“ auf Butcherbäumen mit $k \leq 4$) durch die Programmschritte 4 und 5 in kraft. Skizze 3.7, worin neben der üblichen Farbzweisung beliebig besetzte Gitterplätze durch \bullet gekennzeichnet sind, verdeutlicht diese Situation: Die ursprünglich auf den Plätzen 1 und 3 sitzenden spinverschiedenen Teilchen vertauschen ihre Position im mit einem Stern markierten Schritt,

⁹³Entscheidend für die Generierung dieser Sequenzen ist der im Programmpunkt 4.3.5 des Rahmenalgorithmus' beschriebene Rücksprung zur ersten Binärbaumgabelung oder die Wiederherstellung der beiden erforderlichen Programmzustände, skizziert in Fußnote 88.

während die Auflösung der Doppelbesetzungen auf den Plätzen 4 und 7 zu Platz 3 jeweils durch die diese aufbauende Spinsorte erfolgt, also keine Vertauschung stattfindet.



Der im vorausgegangenen Abschnitt 3.4.13 beschriebene einfache Ansatz nimmt im Gegensatz zu der Erst- [19] und der hier vorgelegten vervollständigten Neuimplementierung auf diesen Sachverhalt keine Rücksicht und überläßt die Kontrolle der Übereinstimmung von Ausgangs- und Endzustand einer entsprechenden Prozedur vor oder in der Berechnung des Hüpfsequenzbeitrages, Programmpunkt 8. Diese künstlich vereinfachte Darstellung ist rein didaktischer Natur, da sich die Wirksamkeit der aus der eingangs beschriebenen Forderung ableitbaren Optimierung besonders leicht verdeutlichen läßt. Um nun die Generierung nicht beitragender Sequenzen und besonders die Schrittweite $p' = 0!$ beim Übergehen von Permutationsbaumästen zu vermeiden – der unzulässige Rücksprung könnte bereits in der zweiten von $2k - 3$ Ebenen erfolgt sein –, kann obiges Verfahren beispielsweise wie folgt modifiziert werden, wobei zunächst Programmschritt 1, vergleiche Abschnitt 3.4.13, ergänzt werden muß:

- 1 [...] Wenn $\bar{j} < j$ (Hinsprung wurde bereits ausgeführt) und $\llbracket M \rrbracket^{(j-1)}_{s,a} = 0$ (Hüpfspin auf Quellgitterplatz nicht vorhanden), dann mit $p = j$ falls $p < j$ zu Programmschritt 12 übergehen. Darin ist $\bar{j} = \llbracket \tilde{G} \rrbracket_j$ die Nummer des Sprunges in Gegenrichtung, $s = \llbracket \tilde{z} \rrbracket_{\bar{j}}$ der auf dem Hinweg gesprungene Spin und $a = \llbracket \tilde{G} \rrbracket_{j,1}$ der Ausgangsgitterplatz.

Die Schritte 2 bis 10 können unverändert übernommen, Schritt 11 muß um die Rücksprungbedingung erweitert werden:

- 11 Neue Rekursions- beziehungsweise Permutationsbaumebene durch Selbstaufwurf betreten, Element (15): falls $\bar{j} < j$ mit $\chi_{\bar{j}} = \chi_s$, worin $s = \llbracket \tilde{z} \rrbracket_{\bar{j}}$, sonst mit $\chi_{\bar{j}} = \chi_a$
- 12 Aktuelle Rekursionsebene verlassen, Element (17): Permutationsbaumebenenzähler p und Hüpfschrittnummer j sowie gegebenenfalls deren Sicherungsvariable j' nebst zugehörigem *flag* zurückgeben

Entsprechend ist im Rahmenprogramm, Abschnitt 3.4.9, insbesondere der den Kern aufrufende Schritt 4.3.4.4 um zwei Anweisungen zu ergänzen:

4.3.4.3 [...] Gegebenenfalls⁹⁴ die Hilfsliste ${}^m\tilde{G}$ neu aufbauen, siehe Gleichung (3.97)

4.3.4.4 [...] Falls $\bar{j} < j$ Unterprogramm für die Elektronenpropagation mit $\chi_j = \chi_s$, worin $s = \llbracket \mathfrak{z} \rrbracket_{\bar{j}}$, sonst mit $\chi_j = \chi_a$ aufrufen und zusätzlich ${}^m\tilde{G}$ übergeben

Für die Festlegung der Hüpfspinfarbe aus dem Spintupel \mathfrak{z} in den optimierten Schritten des Kern- und des Rahmenprogrammes muß für $i = 1, 2, \dots, 2k - 2$ die Metamenge

$$\llbracket {}^m\tilde{G} \rrbracket_i = \llbracket {}^m\acute{G} \rrbracket_{2k-1-\bar{i}}, \quad (3.97)$$

deren „Konstruktion“ auf

$$\llbracket {}^m\acute{G} \rrbracket_{\bar{i}} = i \quad \text{mit} \quad \bar{i} = \llbracket {}^m\tilde{G} \rrbracket_i \quad (3.98)$$

zurückgreift, vom Rahmenprogramm, Abschnitt 3.4.9, nach dortigem Schritt 4.3.4.3 bereitgestellt und dem Kern übergeben werden. Die in den komplementären „Adreßlisten“ (bezüglich G formal Metamengen zweiter und dritter Art) abgelegten Auskünfte lassen sich anschaulich beschreiben: Der Zugriff auf das j te Element von ${}^m\tilde{G}$ beantwortet die hier wesentliche Frage, wo sich die Gegenkante zum Sprung Nummer j , genaugenommen die Rücksprungnummer \bar{j} innerhalb des Kantenpermutationstupels ${}^m\tilde{G}$ befindet; der Zugriff auf das i te Element von ${}^m\acute{G}$ antwortet hingegen auf die Frage, wo sich die Kante Nummer i innerhalb des Kantenpermutationstupels ${}^m\tilde{G}$ befindet, also in welchem Hüpfschritt über i gesprungen wird⁹⁵.

Ist nur das Falicov-Kimball-Modell Gegenstand der Simulation, können a priori bereits im Rahmenprogramm alle Elemente des Hüpfspintupels konventionsgemäß mit einem \uparrow -Spin, $\mathfrak{z} = (1 \mid j = 1, 2, \dots, 2k - 2)$, und folglich in Schritt 11 für die Hüpfspinfarbe stets $\chi_j = \chi_1$ vorgegeben werden.

Um die Auswirkung des verbesserten Verfahrens zu illustrieren, wiederholen wir das Beispiel für $\mathcal{B}_{3,1,27}$ aus Abschnitt 3.4.14. Dabei sei exemplarisch nochmals die oben erwähnte Permutation ${}^m\tilde{G}_1 = \llbracket \mathbb{F}({}^m\mathcal{P}_{3,1}) \rrbracket_2 = \{1, 2, 4, 3\}$ herausgehoben: Während der einfachere Ansatz mit den Gleichungszeilen (3.96-3) und (3.96-4) zwei ungültige Sequenzen generiert, „scheitert“ die neue Vorgehensweise wunschgemäß bereits in Hüpfschritt $j = 3$ mit Anweisung 50 (Programmschritt 1), da für den Rücksprung über die Kante 4 nur ein \downarrow -Spin Verfügung steht ($\chi_a = \blacksquare$), wohingegen auf dem Hinweg in Hüpfschritt $j = 1$ über die Kante 1, Anweisungen 1 bis 5, ein \uparrow -Spin propagiert ist

⁹⁴Siehe Fußnote 95

⁹⁵Je nach Umsetzung der Prüfung auf Vertauschung klassengleicher Kanten im Rahmenprogramm kann ${}^m\acute{G}$ auch bereits im dort auszuführenden Schritt 6 Verwendung finden. In der vorliegenden Neuimplementierung wird geprüft, ob mit $u = 1, 2, \dots, |{}^mE_{\text{CNL}}|$, $v = 1, 2, \dots, \llbracket \llbracket {}^mE_{\text{CNL}} \rrbracket_u \rrbracket - 1$, $w = v + 1, v + 2, \dots, \llbracket \llbracket {}^mE_{\text{CNL}} \rrbracket_u \rrbracket$ und mit $v' = \llbracket \llbracket {}^mE_{\text{CNL}} \rrbracket_{u,v} \rrbracket$, $w' = \llbracket \llbracket {}^mE_{\text{CNL}} \rrbracket_{u,w} \rrbracket$ sowie $v'' = \llbracket \llbracket {}^m\acute{G} \rrbracket_{v'} \rrbracket$, $w'' = \llbracket \llbracket {}^m\acute{G} \rrbracket_{w'} \rrbracket$ gilt: $v'' \neq w''$; falls nicht, also Kanten vertauscht sind, wird $p = w''$ zurückgegeben.

($\chi_j = \blacksquare \Rightarrow \llbracket \check{z} \rrbracket_1 = 1$); entsprechend wird in dieser Ebene abgebrochen. Die Details für den gesamten Permutationsbaum ${}^m\mathcal{P}_{3,1}$ finden sich in folgender Tabelle:

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung	
			j	χ_j	s	a	b	$ {}_sM\rangle^{(j)}$	$\llbracket d \rrbracket_j$	$\llbracket \check{z} \rrbracket_j$		
$\tilde{G} = \{(1, 2), (2, 3), (3, 2), (2, 1)\}$ aus ${}^m\tilde{G}_0 = \{1, 2, 3, 4\}$; $p = j = 1$												
1	I	I	1	○	–	1	2	–	–	–	(Anfangswerte)	
		2	2	○							$\chi_j \neq \blacksquare$	
		3	3	1			1		0		$\llbracket d \rrbracket_0$, da $\chi_a = \blacksquare$	
		4	4	1	○	1		2		1	1	○—○
		7	5	1		1	1	2	$ {}_1(0, 1, 0)\rangle^{(1)}$			$\check{c}_{2,1}^+ \check{c}_{1,1} {}_1(1, 0, 0)\rangle^{(0)}$
		8	6	1								$j \neq 3$
		9	7	2								
		10	8				2	3				
		11	9		●							Selbstaufruf
	2	I	10	2	●	–	2	3	–	–	–	(Übernahmewerte)
		2	11		○			3				●—○: Selbstaufruf
3	I	12	2	○	–	2	3	–	–	–	(Übernahmewerte)	
		13		○							$\chi_j \neq \blacksquare$	
		14	2			2			0		$\llbracket d \rrbracket_1 - 1$, da $\chi_a = \blacksquare$	
		15	2	○	1		3		1	1	○—○	
		7	16	2		1	2	3	$ {}_1(0, 0, 1)\rangle^{(2)}$			$\check{c}_{3,1}^+ \check{c}_{2,1} {}_1(0, 1, 0)\rangle^{(1)}$
		8	17	2								$j \neq 3$
		9	18	3								
		10	19				3	2				
		11	20		○	1		2				Selbstaufruf, $\bar{j} = 2$
	4	I	21	3	○	–	3	2	–	–	–	(Übernahmewerte)
		22		○							$\chi_j \neq \blacksquare$	
		3	23	3			3		0		$\llbracket d \rrbracket_2 - 1$, da $\chi_a = \blacksquare$	
		4	24	3	○	1		2		1	1	○—○
		7	25	3		1	3	2	$ {}_1(0, 1, 0)\rangle^{(3)}$			$\check{c}_{2,1}^+ \check{c}_{3,1} {}_1(0, 0, 1)\rangle^{(2)}$
		8	26	3								$j \neq 3$

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung	
			j	χ_j	s	a	b	$ _s M^{(j)}$	$[[d]]_j$	$[[\tilde{z}]]_j$		
<i>Fortsetzung</i>												
	9	27	4									
	10	28				2	1					
	11	29		○	1		1				Selbstaufruf, $\bar{j} = 1$	
5	1	30	4	○	-	2	1	-	-	-	(Übernahmewerte)	
	2	31		○							$\chi_j \neq \blacksquare$	
	3	32	4			2			0		$[[d]]_3 - 1$, da $\chi_a = \blacksquare$	
	4	33	4	○	1		1		0	1	○—○	
	7	34	4		1	2	1	$ _1(1, 0, 0)^{(4)}$			$\check{c}_{1,1}^+ \check{c}_{2,1} _1(0, 1, 0)^{(3)}$	
	8	35	4	Sequenz gültig, a_n -Beitrag berechnen								$j > 3$
	12	36	4								Rückkehr, $p = 3$	

$$\tilde{G} = \{(1, 2), (2, 3), (2, 1), (3, 2)\} \text{ aus } {}^m\tilde{G}_1 = \{1, 2, 4, 3\}; p = j = 3$$

1	1	37	3	○	-	2	1	-	-	-	(Übernahmewerte)
---	---	----	---	---	---	---	---	---	---	---	------------------

nicht ausführbar, da $\chi_a = \blacksquare$, aber $s = 1$ durch $\bar{j} = 1$ gefordert

	12	38	3								Abbruch, $p = 3$
--	----	----	---	--	--	--	--	--	--	--	------------------

$$\tilde{G} = \{(1, 2), (3, 2), (2, 3), (2, 1)\} \text{ aus } {}^m\tilde{G}_2 = \{1, 3, 2, 4\}; p = j = v = 2$$

1	1	39	2	○	-	3	2	-	-	-	(Übernahmewerte)
---	---	----	---	---	---	---	---	---	---	---	------------------

	2	40		○							$\chi_j \neq \blacksquare$
--	---	----	--	---	--	--	--	--	--	--	----------------------------

	3	41	2			3			1		$[[d]]_1$, da $\chi_a = \blacksquare$
--	---	----	---	--	--	---	--	--	---	--	----------------------------------------

	4	42		○							$\chi_j \neq \blacksquare$
--	---	----	--	---	--	--	--	--	--	--	----------------------------

	5	43	2	○			2				○—○
--	---	----	---	---	--	--	---	--	--	--	-----

	12	44									Abbruch, $p = 3$
--	----	----	--	--	--	--	--	--	--	--	------------------

$$\tilde{G} = \{(1, 2), (2, 1), (2, 3), (3, 2)\} \text{ aus } {}^m\tilde{G}_4 = \{1, 4, 2, 3\}; p = j = 2$$

1	1	45	2	○	-	2	1	-	-	-	(Übernahmewerte)
---	---	----	---	---	---	---	---	---	---	---	------------------

	2	46		○							$\chi_j \neq \blacksquare$
--	---	----	--	---	--	--	--	--	--	--	----------------------------

	3	47	2			2			0		$[[d]]_1 - 1$, da $\chi_a = \blacksquare$
--	---	----	---	--	--	---	--	--	---	--	--------------------------------------------

	4	48	2	○	1		1		0	1	○—○
--	---	----	---	---	---	--	---	--	---	---	-----

	7	49	2		1	2	1	$ _1(1, 0, 0)^{(2)}$			$\check{c}_{1,1}^+ \check{c}_{2,1} _1(0, 1, 0)^{(1)}$
--	---	----	---	--	---	---	---	----------------------	--	--	--------------------------------------------------------

Fortsetzung auf der nächsten Seite

E.	S.	Nr.	Programmschritt-beteiligte Variablen								Bemerkung
			j	χ_j	s	a	b	$ _s M^{(j)}$	$\llbracket d \rrbracket_j$	$\llbracket \tilde{z} \rrbracket_j$	
<i>Fortsetzung</i>											
	8	50	2								$j \neq 3$
	9	51	3								
	10	52				2	3				
	11	53		●							Selbstaufruf
2	1	54	3	●	-	2	3	-	-	-	(Übernahmewerte)
	2	55		●							$\chi_j \neq \blacksquare$
	3	56	3			2			0		$\llbracket d \rrbracket_2$, da $\chi_a = \blacksquare$
	4	57		●							$\chi_j \neq \blacksquare$
	5	58	3	●			3				●—●
	12	59									Abbruch, $p = 3$

$$\tilde{G} = \{(1, 2), (2, 1), (3, 2), (2, 3)\} \text{ aus } {}^m\tilde{G}_5 = \{1, 4, 3, 2\}; p = j = j' = 2$$

1	1	60	2	●	-	2	1	-	-	-	(Übernahmewerte)
	2	61		●							$\chi_j \neq \blacksquare$
	3	62	2			2			0		$\llbracket d \rrbracket_1 - 1$, da $\chi_a = \blacksquare$
	4	63	2	●	1		1		0	1	●—○
	7	64	2		1	2	1	$ _1(1, 0, 0)^{(2)}$			$\check{c}_{1,1}^+ \check{c}_{2,1} _1(0, 1, 0)^{(1)}$
	8	65	2								$j \neq 3$
	9	66	3								
	10	67				3	2				
	11	68		●							Selbstaufruf
2	1	69	3	●	-	3	2	-	-	-	(Übernahmewerte)
	2	70		●							$\chi_j \neq \blacksquare$
	3	71	3			3			0		$\llbracket d \rrbracket_2$, da $\chi_a = \blacksquare$
	4	72		●							$\chi_j \neq \blacksquare$
	5	73	3	●			2				●—●
	12	74									Abbruch, $p = 3$

Verglichen mit dem einfacheren Verfahren bleiben durch den verbesserten Algorithmus nur 74 von 194 Verarbeitungsanweisungen an die Rechenanlage übrig; von den Hüpfsequenzen verbleiben die beitragende erste, Gleichungszeile (3.96-1), sowie die beiden

Fragmente in den Zeilen (3.96-5) und (3.96-6). In Tabelle 3.11 werden die Ergebnisse einer gestrafften Effizienzanalyse für die gesamte dritte Störungsrechenungsordnung dargelegt. Der drastischen Zunahme der Sequanzanzahl entsprechend steigt in höheren Ordnungen die Wirksamkeit des um die „vorausschauende“ Hüpfspinwahl erweiterten *greedy*-Ansatzes.

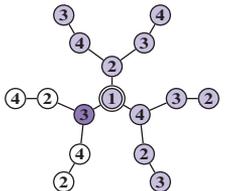
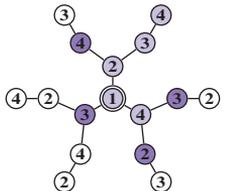
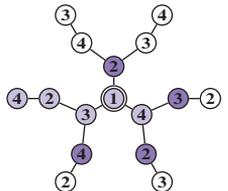
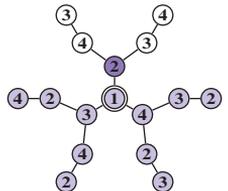
Butcherbaum des Ausgangszustandes			
$\mathcal{B}_{3,1,26} = $ 	$\mathcal{B}_{3,1,27} = $ 	$\mathcal{B}_{3,2,16} = $ 	$\mathcal{B}_{3,2,18} = $ 
Anzahl der Hüpfsequenzen (und dafür formal nötiger Schritte)			
4 (16)	1 (4)	1 (4)	4 (16)
zugrundeliegender Permutationsbaum ${}^m\mathcal{P}_{3,1}$ mit $ V(\mathcal{P}_3) = 16$			
			
Anzahl geprüfter und ausführbarer Hüpf Schritte (hellviolett)			
11	5	5	11
~, davon (nicht) sequenzrelevant			
11 (0)	4 (1)	4 (1)	11 (0)
Anzahl geprüfter und nicht ausführbarer Hüpf Schritte (dunkelviolett)			
1	4	4	1
Anzahl übergangener Hüpf Schritte (weiß)			
4	7	7	4

Tabelle 3.11. Beispielhafte Auswertung der den Hüpfsequenzen der Störungsordnung $n = 3$ zugrundeliegenden Permutationsbäume, ausgehend von der vervollständigten neuen *greedy*-Implementierung des Rechenverfahrens. Mit „nicht ausführbar“ sind sowohl Pauli-verbotene wie auch durch die Rücksprungbeschränkung (zum Hinsprung identische Spinspezies) verhinderte Hüpf Schritte gemeint.

3.5. Berechnung des Sequenzbeitrages zum Reihenkoeffizienten.

3.5.1. *Allgemein.* Das im Folgenden beschriebene Unterprogramm zur Berechnung des Hüpfsequenzbeitrages zum Reihenkoeffizienten setzt voraus, daß entweder in Erweiterung von Flußdiagrammelement (14) in Abbildung 3.9, siehe Abschnitt 3.4.13, bereits

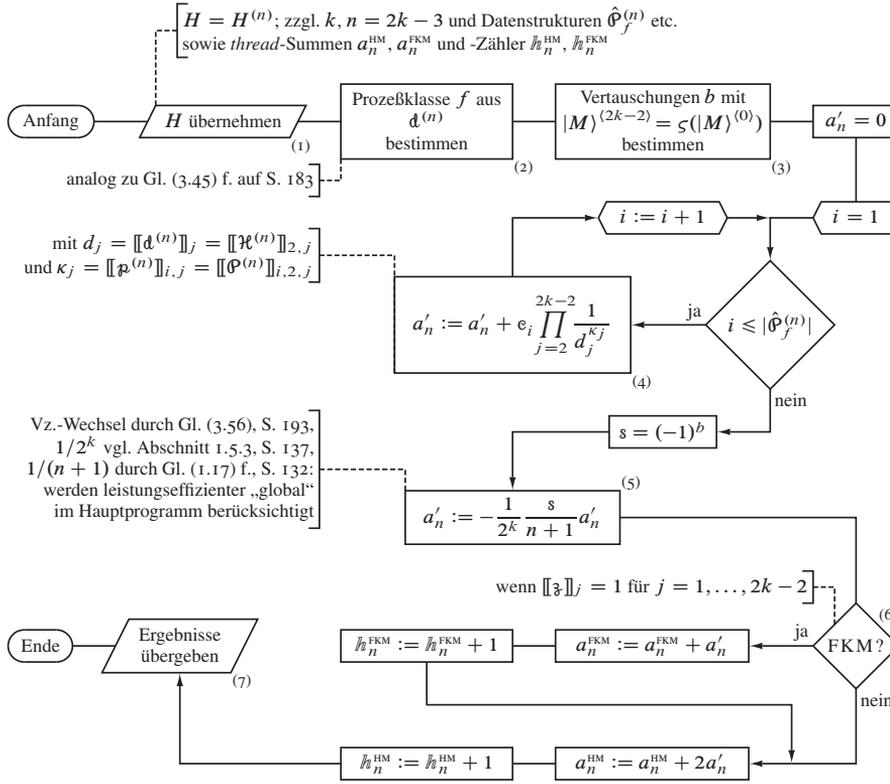


Abbildung 3.10. Ablaufplan des Unterprogramms für die Berechnung des Hüpfsequenzbeitrages zum Reihenkoeffizienten, Flußdiagrammelement (16) in Abbildung 3.9.

vor Aufruf die Ausgangszustandswiederherstellung geprüft oder durch den verbesserten Algorithmus, Abschnitt 3.4.15, sichergestellt wurde.

- 1 Neben der Rechnungsordnung k beziehungsweise $n = 2k - 3$ Ausführungsparameter vom aufrufenden Kern, siehe Abschnitt 3.4.13, und vom Rahmenprogramm, siehe Abschnitt 3.4.9 übernehmen: Hüpfsequenz $H^{(n)}$ und daraus abgeleitet \mathfrak{d} und \mathfrak{z} , Hüpfprozesse $\hat{\Phi}_f^{(n)}$ der Klassen $f = 1, 2, \dots, 2^{(n-1)/2}$ und für den rechnenden Prozeßfaden im Parallelbetrieb Reihenkoeffizientenzwischensummen a_n^{HM} (Hubbardmodell) und a_n^{FKM} (Falicov-Kimball-Modell) sowie zugehörige Sequenzzähler i_n^{HM} und i_n^{FKM} , siehe Flußdiagrammelement (1) in Abbildung 3.10

- 2 Aus dem Doppelbesetzungsverlauf $\mathfrak{d}^{(n)}$ die Prozeßklasse f der Hüpfsequenz Gleichung (3.45) entsprechend ableiten, Element (2):

$$f(\mathfrak{d}^{(n)}) = \sum_{u=3,5,\dots}^n 2^{(u-3)/2} \mathfrak{d}_u^{(n)}, \quad \text{worin} \quad (3.99)$$

$$\mathfrak{d}^{(n)} = (\mathfrak{d}'_v \mid v = 1, 2, \dots, n+2) \quad \text{mit} \quad \mathfrak{d}'_v = \begin{cases} 1 & \text{falls } d_v = \llbracket \mathfrak{d}^{(n)} \rrbracket_v = 0 \\ 0 & \text{sonst} \end{cases}$$

- 3 Aus dem Endzustand $|{}_1M, {}_2M\rangle^{(2k-2)}$ die Anzahl der vertauschten Elektronen innerhalb einer jeden der zwei Spinspezies bestimmen und summieren, Element (3). Da es hinreichend ist, die *minimale* Gesamtzahl zu kennen, genügt ausgehend von der Relation in Gleichung (3.88) ein direkter Vergleich von $|M\rangle^{(2k-2)}$ mit dem Ausgangszustand $|M\rangle^{(0)}$. Praktisch werden die Elektronen beider Sorten an ihren jeweiligen Ausgangsgitterplatz mit einem modifizierten *bubble-sort*-Algorithmus zurückgetauscht, welcher die Anzahl b der nötigen Vertauschungsoperationen zählt⁹⁶; dabei wird die gegenüber effizienteren Sortieralgorithmen besondere Eigenschaft dieses Verfahrens genutzt, daß zwei Tupelelemente bei „ordnungswidriger“ Reihenfolge nur *gegeneinander ausgetauscht* und nicht verschoben werden, siehe Anhang, Abschnitt 2.1.
- 4 Zwischensumme des Unterprogramms mit $a'_n = 0$ initialisieren
- 5 Summationsschleife über alle Prozesse der Klasse f , Index $i = 1, 2, \dots, |\hat{\mathcal{P}}_f^{(n)}|$, mit $e_i = \llbracket \mathcal{P}^{(n)} \rrbracket_{i,1}$:

$$a'_n := a'_n + e_i \prod_{j=2}^{2k-2} \frac{1}{d_j^{\kappa_j}} = a'_n + e_i \begin{bmatrix} p_i \\ d \end{bmatrix}, \quad \text{worin} \quad (3.100)$$

$$d_j = \llbracket \mathfrak{d} \rrbracket_j = \llbracket \mathcal{H}^{(n)} \rrbracket_{2,j} \quad \text{und} \quad \kappa_j = \llbracket p_i \rrbracket_j = \llbracket p \rrbracket_{i,j} = \llbracket \mathcal{P}^{(n)} \rrbracket_{i,2,j},$$

siehe Flußdiagrammelement (4)

- 6 Berechnung des Sequenzvorzeichens mit $\llbracket \mathcal{H}^{(n)} \rrbracket_1 = s = (-1)^b$
- 7 Skalierung des Sequenzbeitrages mit

$$a'_n := -\frac{1}{2^k} \frac{s}{n+1} a'_n. \quad (3.101)$$

Der Vorzeichenwechsel ist auf Gleichung (3.56) zurückzuführen, der Faktor $1/2^k$ resultiert aus der in Abschnitt 1.5.3 diskutierten Anzahl möglicher Spinbelegungen,

⁹⁶Dieser einfache Ansatz erlaubt nur die wesentliche Aussage zu treffen, ob gerad- oder ungeradzahlig viele Sortiertvertauschungen auf den (linearen) Färbungstupeln – genauer Spinnummerstupeln – stattgefunden haben, welche (linearisiert) den Butcherbaum abbilden.

sowie

$$\mathfrak{z}^{(5)} = (1, 1, 1, 1, 1, 1).$$

Programmschritt 2 ermittelt daraus gemäß Gleichung (3.99) zunächst die Prozeßklasse f , ausführlich notiert mit $\hat{d}^{(5)} = (1, 0, 0, 0, 0, 1)$ und $\hat{d}_u^{(5)} = \llbracket \hat{d}^{(5)} \rrbracket_u$:

$$f(\hat{d}^{(5)}) = 2^{(u-3)/2} \hat{d}_u^{(5)}|_{u=3} + 2^{(u-3)/2} \hat{d}_u^{(5)}|_{u=5} = 0.$$

Daran anschließend bestimmt Schritt 3 das Betragsvorzeichen bezüglich allfälliger Elektronenvertauschungen⁹⁹: In der „Farbebene“ der \uparrow -Spins kann der Endzustand

$${}_1|(4, 0, 0, 1)\rangle_{4,2}^{(6)}$$

in den Ausgangszustand

$${}_1|(1, 0, 0, 4)\rangle_{4,2}^{(0)}$$

offenbar durch einen einzigen Vertauschungsschritt überführt werden, ${}_1b = 1$; hinsichtlich der \downarrow -Spins stimmen End- und Ausgangszustand überein, ${}_2b = 0$.

Die nun anstehende zentrale Summationsschleife, Schritt 5 im Programmablauf, welche insbesondere die Doppelbesetzungszahlen mit den \hat{S} -Exponenten „verrechnet“, erstreckt sich durch $|\hat{p}_0^{(5)}| = 1$ nur über den Indexwert $i = 1$. Mit $e_i|_{i=1} = 6$ und der (hier streng formal dem Algorithmus folgenden) Initialisierung der Unterprogrammzwischensumme $a'_5 = 0$ ergibt sich

$$\begin{aligned} a'_5 &:= a'_5 + \left(e_i \begin{bmatrix} p_i \\ d \end{bmatrix} \right) \Big|_{i=1} = a'_5 + 6 \cdot \left[\begin{array}{c} \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|} \hline \square & \square & \square & \square \\ \hline \end{array} \end{array} \right] \\ &= a'_5 + 6 \cdot \frac{1}{1^1} \cdot \frac{1}{1^1} \cdot \frac{1}{2^1} \cdot \frac{1}{1^1} \cdot \frac{1}{1^1} = a'_5 + 3. \end{aligned}$$

Dieser Sequenzbeitrag wird mit $s = (-1)^{1^{b+2b}} = -1$ aus Programmschritt 6 gemäß Gleichung (3.101) in Schritt 7 zu

$$a'_5 := -\frac{1}{2^4} \frac{s}{6} a'_5 = \frac{1}{96} \cdot 3 = \frac{1}{32}$$

⁹⁹Nach Fußnote 76 wäre formal wie folgt zu rechnen:

$${}_1|(4, 0, 0, 1)\rangle_{4,2}^{(6)} = {}_1\zeta \cdot {}_1|(1, 0, 0, 4)\rangle_{4,2}^{(0)} \quad \text{und} \quad {}_2|(0, 2, 3, 0)\rangle_{4,2}^{(6)} = {}_2\zeta \cdot {}_2|(0, 2, 3, 0)\rangle_{4,2}^{(0)},$$

worin

$${}_1\zeta = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad \text{und} \quad {}_2\zeta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

ist. Daraus ergibt sich $s = \det({}_1\zeta) \cdot \det({}_2\zeta) = -1 \cdot 1 = -1$.

zusammengezogenen Schleife zu demonstrieren. Mit $g' = |\hat{\mathcal{P}}_4^{(7)}| = 4$ wird diese folgendermaßen ausgewertet:

$$\begin{aligned}
 a'_7 &= e_1 \begin{bmatrix} p_1 \\ d \end{bmatrix} + e_2 \begin{bmatrix} p_2 \\ d \end{bmatrix} + e_3 \begin{bmatrix} p_3 \\ d \end{bmatrix} + e_4 \begin{bmatrix} p_4 \\ d \end{bmatrix} \\
 &= -24 \cdot \begin{bmatrix} \text{Diagram 1} \\ \text{Diagram 2} \end{bmatrix} - 6 \cdot \begin{bmatrix} \text{Diagram 3} \\ \text{Diagram 4} \end{bmatrix} - 8 \cdot \begin{bmatrix} \text{Diagram 5} \\ \text{Diagram 6} \end{bmatrix} - 10 \cdot \begin{bmatrix} \text{Diagram 7} \\ \text{Diagram 8} \end{bmatrix} = -\frac{9}{2}.
 \end{aligned}$$

Mit $b = 0 \Rightarrow s = 1$ ergibt sich der Hüpfsequenzbeitrag

$$a'_7 := -\frac{1}{25} \frac{s}{8} a'_7 = -\frac{1}{256} \cdot \frac{9}{2} = -\frac{9}{512},$$

welcher zweifach in das Hubbardmodell, durch Propagation von \downarrow -Spins zwischen dem vierten und dem achten Zwischenzustand nicht jedoch in das Falicov-Kimball-Modell eingeht.

Abschließend wenden wir uns der Auswertung der dritten Störungsrechnungsordnung zu und vervollständigen damit auch die Betrachtungen im Abschnitt 3.4.14 zum zweiten der vier Ausgangszustände, $\mathcal{B}_{3,1,27}$, sowie die Übersichtstabelle 3.11 in Abschnitt 3.4.15. Die auftretenden Elektronenbewegungen haben in Sequenzform mit Operatornotation folgendes Aussehen, wobei der dritte Hüpfschritt in Gleichung (3.103-4) der Demonstration dessen faktischer Umsetzung in Binärdarstellung in Skizze 3.5 dient, siehe Abschnitt 3.4.4, und die Gleichungszeile (3.103-5) der Zeile (3.96-1) im Beispiel zum Kernalgorithmus entspricht:

Ausgangszustand $\mathcal{B}_{3,1,26}$:

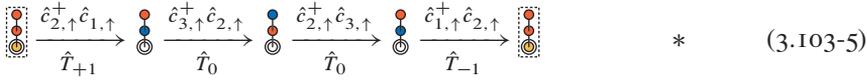
$$\begin{array}{c}
 \begin{array}{ccccccc}
 \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{+1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_0} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_0} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} \\
 \hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow} & & \hat{c}_{3,\downarrow}^+ \hat{c}_{2,\downarrow} & & \hat{c}_{2,\downarrow}^+ \hat{c}_{3,\downarrow} & & \hat{c}_{1,\uparrow}^+ \hat{c}_{2,\uparrow} \\
 \end{array} \quad (3.103-1)
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{ccccccc}
 \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{+1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_0} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_0} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} \\
 \hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow} & & \hat{c}_{3,\downarrow}^+ \hat{c}_{2,\downarrow} & & \hat{c}_{1,\uparrow}^+ \hat{c}_{2,\uparrow} & & \hat{c}_{2,\downarrow}^+ \hat{c}_{3,\downarrow} \\
 \end{array} \quad (3.103-2)
 \end{array}$$

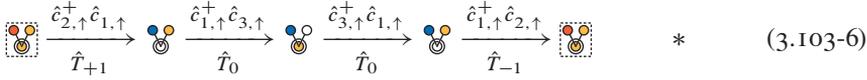
$$\begin{array}{c}
 \begin{array}{ccccccc}
 \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{+1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{-1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{+1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} \\
 \hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow} & & \hat{c}_{1,\uparrow}^+ \hat{c}_{2,\uparrow} & & \hat{c}_{3,\downarrow}^+ \hat{c}_{2,\downarrow} & & \hat{c}_{2,\downarrow}^+ \hat{c}_{3,\downarrow} \\
 \end{array} \quad (3.103-3)
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{ccccccc}
 \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{+1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{-1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} & \xrightarrow{\hat{T}_{+1}} & \begin{array}{|c|} \hline \bullet \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} \\
 \hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow} & & \hat{c}_{1,\uparrow}^+ \hat{c}_{2,\uparrow} & & \hat{c}_{2,\uparrow}^+ \hat{c}_{3,\uparrow} & & \hat{c}_{3,\uparrow}^+ \hat{c}_{2,\uparrow} \\
 \end{array} \quad * \quad (3.103-4)
 \end{array}$$

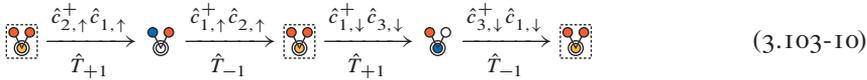
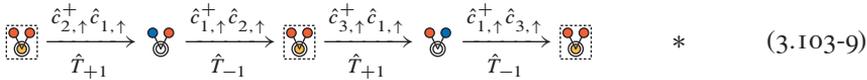
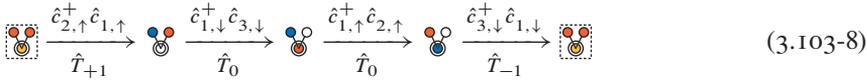
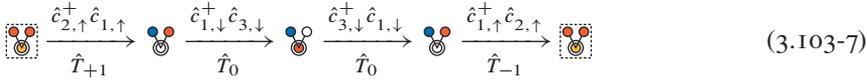
Ausgangszustand $\mathcal{B}_{3,1,27}$:



Ausgangszustand $\mathcal{B}_{3,2,16}$:



Ausgangszustand $\mathcal{B}_{3,2,18}$:



Laut Tabelle 3.5 existieren in dritter Ordnung zwei Prozeßklassen mit jeweils einem Prozeß; gleiches gilt für die Sequenzen nach Tabelle 3.9. Somit lassen sich die Hüpfsequenzen – zum Falicov-Kimball-Modell gehörige sind oben mit einem Stern gekennzeichnet – wie folgt einteilen¹⁰⁰:

Modell	f	Anzahl	Gleichungszeilen
HM	0	6	(3.103-1), (3.103-2), (3.103-5), (3.103-6), (3.103-7), (3.103-8)
	1	4	(3.103-3), (3.103-4), (3.103-9), (3.103-10)
FKM	0	2	(3.103-5), (3.103-6)
	1	2	(3.103-4), (3.103-9)

¹⁰⁰Die Zusammengehörigkeit der Hüpfsequenzen gemäß der topologischen Isomorphie der („entwurzelten“) und gegebenenfalls durch wechselseitigen Austausch der Spinspezies farbinversen Ausgangszustände wird in Abschnitt 6.2.2 angesprochen.

Da in Butcherbäumen mit $k = 3$ Ecken noch keine Möglichkeit zur Teilchenvertauschung besteht, folglich $s = (-1)^0 = 1$ ist, können die Beiträge zum Reihenkoeffizienten pauschal¹⁰¹ für jede der beiden Klassen berechnet werden, nämlich zu

$$a'_3|_{f=0} = -\frac{1}{2^3} \frac{1}{4} \cdot 4 \cdot \left[\begin{array}{|c|} \hline \square \square \\ \hline \square \square \\ \hline \end{array} \right] = -\frac{1}{8} \quad \text{und} \quad a'_3|_{f=1} = +\frac{1}{2^3} \frac{1}{4} \cdot 4 \cdot \left[\begin{array}{|c|} \hline \square \square \\ \hline \square \square \\ \hline \end{array} \right] = +\frac{1}{8} .$$

Zusammenfassend erhalten wir in Fortsetzung der Gleichungen (1.27) und (1.28)

$$\frac{1}{2} a_3^{\text{HM}} = 6a'_3|_{f=0} + 4a'_3|_{f=1} = -\frac{1}{4} , \quad \text{somit} \quad E_0^{(3),\text{HM}} = -\frac{1}{2} , \quad (3.104)$$

und erwartungsgemäß

$$a_3^{\text{FKM}} = 2a'_3|_{f=0} + 2a'_3|_{f=1} = 0 , \quad \text{somit} \quad E_0^{(3),\text{FKM}} = 0 . \quad (3.105)$$

3.6. Zusammenfassung. In diesem Kapitel wurde entlang der zuvor durchgeführten konkreten Neuimplementierung gemäß dem Programmablaufplan in Abbildung 3.1 eine *abstrakt-mathematisierte* Detailbeschreibung für die graphentheoretische Störungsrechnung am Hubbardmodell auf dem Bethegitter gegeben. Dabei wurden in den Grenzfällen starker Wechselwirkung und unendlicher Koordinationszahl auch methodische Ansatzpunkte für Untersuchungen jenseits der Gitterhalbfüllung, die Behandlung von Anregungen sowie die Beschreibung anderer fermionischer und bosonischer Quantensysteme aufgezeigt. Die problemspezifische und daher notwendigerweise sehr komplexe Notation gründet sich im Wesentlichen auf das vorangestellte Kapitel 2 und wurde in den einzelnen Abschnitten vollständig ausentwickelt. Mit dieser Art der Darlegung des Gesamtverfahrens ist zunächst eine Basis geschaffen, welche die Verwendung anderer Programmierparadigmen und -sprachen erlaubt, wie dies im Falle des funktionalen Zugangs [33] bereits erfolgt ist. Vor allem jedoch wird dadurch die Anwendung auf umfassendere physikalische Fragestellungen wie die Zweipunkt-Korrelationsfunktionen erleichtert, welche sich aus der Reihenentwicklung der Greenschen Funktion erschließen.

Die Umsetzung des von EVA KALINOWSKI ausgearbeiteten und in Kapitel 1 vorgestellten Algorithmus' für prinzipiell beliebige zur Grundzustandsenergie $E_0^{(n)}$ beitragende Ordnungen n der $1/U$ -Reihe beruht im wesentlichen auf der Entwicklung zweier Hauptbestandteile. Zwischen beiden – den Hüpfsequenzen $\mathcal{H}^{(n)} = (s, \mathfrak{d}^{(n)})$ einerseits, den Hüpfprozessen $\mathcal{P}^{(n)} = (e, \mathfrak{p}^{(n)})$ andererseits – besteht eine deutlich erkennbare struktureypologische Verwandtschaft.

Zum einen ist es möglich, die Ausgangszustände systematisch zu erzeugen, deren jeder ein Repräsentant der endlich vielen verschieden gefärbten, stets aber aus k Gitterplätzen bestehenden Auschnitte des unendlich ausgedehnten des Bethegitters ist. Dabei handelt es sich um eine (im hier betrachteten Fall der anregungsfreien Gitterhalbfüllung) zweifarbige Untermenge $B'_{k,l} \subset B_{k,l}$ der viergefärbten Butcherbäume $\mathcal{B}_{k,l,m} \in$

¹⁰¹Ein artverwandtes und hocheffizientes Verfahren für beliebige Ordnungen wird in Abschnitt 6.3.3 vorgestellt und im Kapitel 1 des Anhangs detailliert erläutert.

$B_{k,l} \in \mathbb{B}_k$ der Baumordnung $k = (n + 3)/2$, deren Aufbau Inhalt der Abschnitte 3.1 und insbesondere 3.2 ist. Damit besteht die Voraussetzung für die eigentliche Elektronenpropagation auf den Gitterausschnitten, welche die Hüpfsequenzen $\mathcal{H}^{(n)} \in \hat{\mathcal{H}}^{(n)}$ in Form von Tupeln aus $n + 2 = 2k - 1$ viergefärbten Bäumen generiert, dem Ziel der Darstellung in Abschnitt 3.4.

Zum anderen läßt sich der effektive Hamiltonoperator \tilde{H} konsequent nach der Wechselwirkung entwickeln, wodurch die Hüpfprozesse $\mathcal{P}^{(n)} \in \hat{\mathcal{P}}^{(n)}$, Gegenstand des Abschnitts 3.3, zugänglich werden. Die Gewichtung einer jeden Hüpfsequenz – in das Vorzeichen s durch Elektronenvertauschungen und den Doppelbesetzungsverlauf $d^{(n)}$ transformiert – durch den zugehörigen Hüpfprozeßtyp – aus dem Koeffizienten e und dem Exponententupel $\mathfrak{p}^{(n)}$ zusammengesetzt – erlaubt nun die Berechnung des jeweiligen Energiebeitrages, wie abschließend in Abschnitt 3.5 ausgeführt wurde.

4. Ergebnisse

Nach der umfassenden Erläuterung des gesamten algebraisch-graphentheoretischen Rechenverfahrens im vorausgegangenen Kapitel wendet sich diese Arbeit wieder den eingangs formulierten physikalischen Fragestellungen zu. Dabei wird zunächst an die Berechnung der dafür wesentlichen Ausgangsdaten in Abschnitt 3.5 angeknüpft; nachfolgend werden die zu Abschnitt 1.2 gehörigen Ergebnisse vorgestellt.

Unter Berücksichtigung des Spinsymmetriefaktors $\gamma = 1$ für das Falicov-Kimball-Modell und $\gamma = 2$ für das Hubbardmodell gilt auf Grundlage der Kapitel 1 und 3 und insbesondere in Erweiterung der Gleichungen (3.100) und (3.102) auf die gesamte Störungsrechnung der Ordnung U^{-n} bei halber Bandfüllung

$$a_n = -\frac{\gamma}{2^{(n+3)/2}} \frac{1}{n+1} \sum_{g=1}^{2^{(n-1)/2}} \sum_{h=1}^{|\hat{\mathcal{H}}_{g-1}^{(n)}|} s_h \sum_{i=1}^{|\hat{\mathcal{P}}_{g-1}^{(n)}|} e_i \begin{bmatrix} \mathfrak{p}_i \\ d_h \end{bmatrix} \quad \text{mit} \quad \begin{bmatrix} \mathfrak{p}_i \\ d_h \end{bmatrix} = \prod_{j=2}^{n+1} \frac{1}{d_j^{\kappa_j}} \quad (4.1)$$

sowie $s_h = \llbracket \hat{\mathcal{H}}_{g-1}^{(n)} \rrbracket_{h,1}$, $d_h = \llbracket \hat{\mathcal{H}}_{g-1}^{(n)} \rrbracket_{h,2}$, $e_i = \llbracket \hat{\mathcal{P}}_{g-1}^{(n)} \rrbracket_{i,1}$, $\mathfrak{p}_i = \llbracket \hat{\mathcal{P}}_{g-1}^{(n)} \rrbracket_{i,2}$
 und $d_j = \llbracket \hat{\mathcal{H}}_{g-1}^{(n)} \rrbracket_{h,2,j}$, $\kappa_j = \llbracket \hat{\mathcal{P}}_{g-1}^{(n)} \rrbracket_{i,2,j}$.

Anschaulich formuliert „summiert“ in dieser die beiden Hauptbestandteile des Gesamtalgorithmus’ umfassenden Gleichung der Index g über alle Hüpfsequenz- beziehungsweise Hüpfprozeßklassen; darin wiederum laufen die Summationsindices h und i über alle Hüpfsequenzen des betrachteten Modells und die (modellunabhängigen) Hüpfprozesse mit den jeweiligen Sequenzvorzeichen und Prozeßkoeffizienten¹⁰². Die davon umschlossene Produktbildung mit dem Index j „verrechnet“ die sequenzeigenen Doppelbesetzungszahlen mit den prozeßzugehörigen \hat{S} -Exponenten. Die Notation im Einzelnen ist umfassend in den Abschnitten 3.3 bis 3.5 erklärt.

¹⁰²Siehe Fußnote 56

4.1. Grundzustandsenergie und Doppelbesetzungszahl. Die Grundzustandsenergie des Hubbardmodells als Funktion der Wechselwirkung – das zentrale Ergebnis der graphentheoretischen Störungsrechnung – erhält nun auf Grundlage von Gleichung (1.12) und in der allgemeinen Form gemäß (3.1) mit den Reihenkoeffizienten $a_n^{\text{HM}} = a_n|_{r=2}$ aus (4.1) folgendes Aussehen:

$$E_0^{\text{HM}}(U) = \sum_{n=1,3,\dots}^{\infty} a_n^{\text{HM}} \frac{1}{U^n} = -\frac{1}{2} \frac{1}{U} - \frac{1}{2} \frac{1}{U^3} - \frac{19}{8} \frac{1}{U^5} - \frac{593}{32} \frac{1}{U^7} - \frac{23877}{128} \frac{1}{U^9} - \frac{4496245}{2048} \frac{1}{U^{11}} - \frac{1588528613}{55296} \frac{1}{U^{13}} - \frac{12927125815211}{31850496} \frac{1}{U^{15}} + \mathcal{O}\left(\frac{1}{U^{17}}\right). \quad (4.2)$$

Alle im Folgenden daraus abgeleiteten Größen erweitern somit die Resultate der Fundamentalarbeit [19] von EVA KALINOWSKI und WŁADYSŁAW GLUZA um zwei beitragende Ordnungen. Mit den zum Falicov-Kimball-Modell gehörigen Hüpfsequenzen kann, eine Ordnung über das Hubbardmodell hinausgehend,

$$E_0^{\text{FKM}}(U) = -\frac{1}{4} \frac{1}{U} + \mathcal{O}\left(\frac{1}{U^{19}}\right) \quad (4.3)$$

als Endergebnis im Rahmen dieser Arbeit angegeben werden. Entsprechend läßt sich die *Doppelbesetzungszahl* $L^{-1}D(U) = dE(U)/dU$ bestimmen [19]; für das Hubbardmodell lautet diese

$$\frac{1}{L}D^{\text{HM}}(U) = \sum_{n=1,3,\dots}^{\infty} n a_n^{\text{HM}} \frac{1}{U^{n+1}} = \frac{1}{2} \frac{1}{U^2} + \frac{3}{2} \frac{1}{U^4} + \frac{95}{8} \frac{1}{U^6} + \frac{4151}{32} \frac{1}{U^8} + \frac{214893}{128} \frac{1}{U^{10}} + \frac{49458695}{2048} \frac{1}{U^{12}} + \frac{20650871969}{55296} \frac{1}{U^{14}} + \frac{64635629076055}{10616832} \frac{1}{U^{16}} + \mathcal{O}\left(\frac{1}{U^{18}}\right) \quad (4.4)$$

und für das Falicov-Kimball-Modell

$$\frac{1}{L}D^{\text{FKM}}(U) = \frac{1}{4} \frac{1}{U^2} + \mathcal{O}\left(\frac{1}{U^{20}}\right). \quad (4.5)$$

Tabelle 4.1 zeigt die Primfaktoren der Koeffizienten der Grundzustandsenergie, worauf in Abschnitt 5.2.1 zurückgegriffen wird. Zusätzlich sind auch die (gerundeten) Dezimalwerte angegeben, welche einen direkten Vergleich mit den Vorhersagen der *extrapolierten Störungstheorie* [18], auf die der folgende Abschnitt eingeht, erlaubt.

4.2. Vergleich zur Numerik. Die numerische Auswertung der Gleichungen der Dynamischen *mean-field*-Theorie (DMFT) für das Hubbardmodell ist insbesondere und quasi zueinander methodisch komplementär mit DDMRG und QMC als *impurity solver*¹⁰³ durchgeführt worden; Referenz [21] respektive Referenz [18] geben über Details Auskunft

¹⁰³Bei der Behandlung eines (unendlich ausgedehnten) Quantensystems mit der Methode der DMFT wird dieses, wie bereits in Fußnote 2 erwähnt, auf ein effektives selbstkonsistentes Störstellenmodell (im Limes hoher Dimension exakt) abgebildet, hier das *single impurity Anderson model*. In den meisten Fällen ist es aber äußerst schwierig, das Störstellenproblem zu lösen; praktisch wird auf numerischen Wege und rekursiv vorgegangen, wobei die Abschätzung der dadurch auftretenden Fehler ein zusätzliches Problem darstellt.

n	k	Betrag des Reihenkoeffizienten a_n^{HM}			Abschätzung lt. Ref. [18]
		Zähler	Nenner	dezimal	
1	2	1	2	0,5	
3	3	1	2	0,5	
5	4	19	2^3	2,375	
7	5	593	2^5	18,5313	
9	6	$3^2 \cdot 7 \cdot 379$	2^7	186,539	
11	7	$5 \cdot 13^2 \cdot 17 \cdot 313$	2^{11}	2195,43	
13	8	$7 \cdot 109 \cdot 967 \cdot 2153$	$2^{11} \cdot 3^3$	28727,7	28705,6 (0,77‰ Abw.)
15	9	$12653 \cdot 1021664887$	$2^{17} \cdot 3^5$	405869	405006 (2,13‰ Abw.)

Tabelle 4.1. Einige weitere Angaben zu den Reihenkoeffizienten der Grundzustandsenergie des Hubbardmodells. Zähler und Nenner sind jeweils in ihrer Primfaktorzerlegung angegeben.

und verweisen auf weitergehende Literatur. Für die Grundzustandsenergie $E_{\text{DMFT}}(U)$ und die Doppelbesetzungszahl $D_{\text{DMFT}}(U)$ des Quantensystems in Abhängigkeit von der Wechselwirkung ergeben sich dabei die in Tabelle 4.2 zusammengefaßten und in den Teilen (a) und (b) von Abbildung 4.1 dargestellten Werte. Ebenfalls in dieser Tabelle verzeichnet sind für jeden der beiden DMFT-*solver* die Abweichungen

$$\Delta E_{\text{DMFT}}^{(n)}(U) = |E_{\text{DMFT}}(U) - E^{(n)}(U)| \quad \text{und}$$

$$\Delta D_{\text{DMFT}}^{(n+1)}(U) = |D_{\text{DMFT}}(U) - D^{(n+1)}(U)|$$

der numerischen Ergebnisse von den algebraischen Werten aus den endlichen Reihen

$$E^{(n)}(U) = \sum_{i=1,3,\dots}^n a_i^{\text{HM}} \frac{1}{U^i} \quad (4.6)$$

$$D^{(n+1)}(U) = \sum_{i=1,3,\dots}^n i a_i^{\text{HM}} \frac{1}{U^{i+1}} \quad \text{und} \quad (4.7)$$

für die bislang höchste erreichte Störungsrechnungsordnung $n = 15$. Systematisch verringern sich diese Abweichungen bei steigender Wechselwirkungsstärke, hier im Intervall $4,5 \leq U \leq 6,0$ dargestellt; im Verlauf von $n = 1, 3, \dots, 15$ nimmt die Abweichung mit jedem weiteren Term notwendigerweise ab, wie die Abbildungsteile (c) und (d) beispielhaft für $U = 4,8$ zeigen.

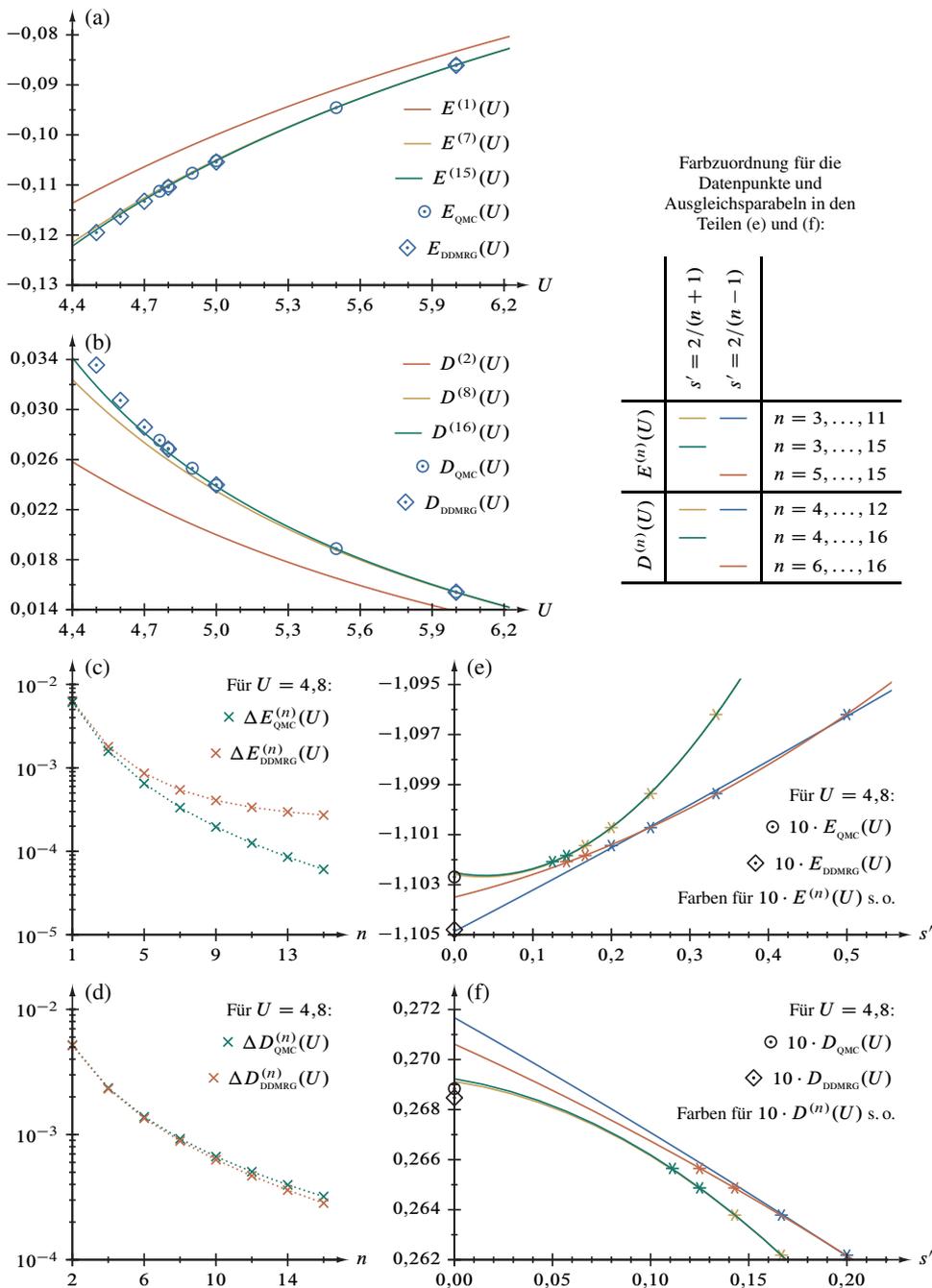


Abbildung 4.1. Grundzustandsenergie und Doppelbesetzungszahl: Vergleich der Störungsrechnung mit DMFT-Ergebnissen. Man beachte in den Teilen (b), (d) und (f), daß die Ordnungen von $D(U)$ als erste Ableitung von $E(U)$ nach der Wechselwirkung U , siehe Gleichung (4.4), geradzahlig sind, woraus auch die Vereinfachung für s' in der Farbzunordnungslegende folgt.

U	$-E_{\text{DMFT}}(U)$		$\Delta E_{\text{DMFT}}^{(15)}(U)$	
	DDMRG	QMC	DDMRG	QMC
4,5	0,11951	*	$5,82 \cdot 10^{-4}$	*
4,6	0,11632	*	$4,78 \cdot 10^{-4}$	*
4,7	0,11326	*	$3,18 \cdot 10^{-4}$	*
4,763318	*	0,11126724	*	$7,43 \cdot 10^{-5}$
4,8	0,11048	0,11026919	$2,72 \cdot 10^{-4}$	$6,08 \cdot 10^{-5}$
4,9	*	0,10766116	*	$3,71 \cdot 10^{-5}$
5,0	0,10543	0,10519814	$2,55 \cdot 10^{-4}$	$2,36 \cdot 10^{-5}$
5,5	*	0,09457761	*	$3,37 \cdot 10^{-6}$
6,0	0,08612	0,08604804	$7,26 \cdot 10^{-5}$	$6,42 \cdot 10^{-7}$

U	$D_{\text{DMFT}}(U)$		$\Delta D_{\text{DMFT}}^{(16)}(U)$	
	DDMRG	QMC	DDMRG	QMC
4,5	0,033558442	*	$1,68 \cdot 10^{-3}$	*
4,6	0,030728864	*	$8,36 \cdot 10^{-4}$	*
4,7	0,028588998	*	$4,54 \cdot 10^{-4}$	*
4,763318	*	0,02754891	*	$4,29 \cdot 10^{-4}$
4,8	0,026846772	0,02688322	$2,83 \cdot 10^{-4}$	$3,20 \cdot 10^{-4}$
4,9	*	0,02532133	*	$1,74 \cdot 10^{-4}$
5,0	0,023985481	0,02396695	$1,21 \cdot 10^{-4}$	$1,03 \cdot 10^{-4}$
5,5	*	0,01888665	*	$1,20 \cdot 10^{-5}$
6,0	0,015405815	0,01542583	$1,80 \cdot 10^{-5}$	$2,00 \cdot 10^{-6}$

Tabelle 4.2. Grundzustandsenergie (oberer Tabellenteil) und Doppelbesetzungszahl (unterer Tabellenteil) aus der Störungsrechnung im Vergleich mit DMFT-Ergebnissen. Die Lösung der *mean-field*-Gleichungen erfolgte mit DDMRG [21] und QMC [18]. Die durch einen Stern vertretenen Werte liegen nicht vor.

Genauere Betrachtung erschließt einerseits, daß die Abweichung der QMC-Energie von der Störungsrechnung stets kleiner als jene mit DDMRG gewonnene ist, Teil (c) in Abbildung 4.1, sich jedoch andererseits dieses Bild für die Doppelbesetzungszahl umkehrt, Teil (d). Da überdies der jeweilige Unterschied zwischen beiden DMFT-Lösungen im Vergleich kaum signifikant ist, kann auf Grundlage dieser einfachen Analyse weder eine Entscheidung zugunsten eines der beiden Verfahren noch eine Bewertung der DMFT-Näherung als solche vorgenommen werden. Bedingt durch die Störungstheorie ist ein Ergebnisvergleich prinzipiell nur für $U \geq U_c$, also im Bereich des Mottisolators erlaubt, siehe auch Abschnitt 4.3.1.

Insofern ist der Fall $U = 4,8$ besonders interessant, da für beide *solver* der Feldgleichungen Daten vorhanden sind, die nahe der kritischen Wechselwirkung liegen: Im Zuge der Erstimplementierung wurden 2004 die Reihenkoeffizienten bis einschließlich elfter Ordnung veröffentlicht [18, 19]; darauf gegründete lineare Ausgleichsrechnungen für die Grundzustandsenergiewerte für $n = 3, 5, \dots, 11$ aus Gleichung (4.6), siehe Tabelle 4.3, mit dem quadratischen Polynom

$$E^{(n)}(U) = \alpha_{(s)} + \beta_{(s)} \frac{2}{s} + \gamma_{(s)} \frac{4}{s^2} \quad \text{mit } s = n + 1 \quad \text{und } s = n - 1 \quad (4.8)$$

liefern für $U = 4,8$ die für $n \rightarrow \infty$ relevanten (die anderen sind dort tabelliert) genäherten Koeffizienten $\alpha_{(n+1)} = -0,11026$ und $\alpha_{(n-1)} = -0,11049$ und lassen somit eine Entscheidung durch die Extrapolation zwischen QMC mit $E_{\text{DMFT}}(U) = -0,11027 \approx \alpha_{(n+1)}$ und DDMRG mit $E_{\text{DMFT}}(U) = -0,11048 \approx \alpha_{(n-1)}$ ebenfalls offen.

Hingegen ergibt die analoge Rechnung mit den beiden neuen Reihenkoeffizienten für $n = 5, 7, \dots, 15$ näherungsweise $\alpha_{(n-1)} = -0,11035$, während für $n = 3, 5, \dots, \dots, 15$ der Regressionsparameter $\alpha_{(n+1)} = -0,11025$ weitgehend konstant bleibt. Daraus klar erkennbar ist ein deutliches Abrücken des Koeffizienten $\alpha_{(n-1)}$ vom DDMRG-Ergebnis bei gleichbleibender Übereinstimmung von $\alpha_{(n+1)}$ mit dem QMC-Wert in Bestätigung der Genauigkeit dieses Näherungsverfahrens.

Eine ebenfalls für $U = 4,8$ entsprechend verlaufende Analyse der Doppelbesetzungszahlen aus Gleichung (4.7), siehe gleichfalls Tabelle 4.3, mit der Parabel

$$D^{(n+1)}(U) = \alpha_{(s)} + \beta_{(s)} \frac{2}{s} + \gamma_{(s)} \frac{4}{s^2} \quad \text{mit } s = n + 2 \quad \text{und } s = n \quad (4.9)$$

führt bis einschließlich $n = 11$ auf $\alpha_{(n+2)} = 0,26911$ sowie $\alpha_{(n)} = 0,27166$; unter Berücksichtigung der Störungsreihenkoeffizienten dreizehnter und fünfzehnter Ordnung ergibt sich (beginnend mit $n = 3, 5, \dots$) $\alpha_{(n+2)} = 0,26922$ sowie (beginnend mit $n = 5, 7, \dots$) $\alpha_{(n)} = 0,27061$. Neben der Tatsache, daß die DMFT-Ergebnisse, siehe Tabelle 4.2, aus QMC mit $10 \cdot D_{\text{DMFT}}(U) = 0,26883$ und DDMRG mit $10 \cdot D_{\text{DMFT}}(U) = 0,26847$ untereinander im Einklang stehen, erweist sich bereits in elfter Ordnung – insbesondere für QMC – eine hervorragende Übereinstimmung mit $\alpha_{(n+2)}$. Bezüglich dieses Parameters ändert sich in fünfzehnter Ordnung nur Unwesentliches; demgegenüber nähert sich $\alpha_{(n)}$, in elfter Ordnung noch stark abweichend, durch die beiden folgenden Ordnungen den DMFT-Lösungen an. Die Teile (e) und (f) in Abbildung 4.1 stellen sämtliche Resultate dieser Extrapolationsmethode, die in ähnlicher Weise für den nächsten

n	1	3	5	7	9	11	13	15
$-E^{(n)}(U)$	0,104167	0,108688	0,10962	0,109936	0,110073	0,110144	0,110184	0,110208
$D^{(n+1)}(U)$	0,0217014	0,0245271	0,025498	0,0259584	0,0262169	0,0263784	0,0264867	0,0265634

	s	$\alpha_{(s)}$	$\beta_{(s)}$	$\gamma_{(s)}$	s	$\alpha_{(s)}$	$\beta_{(s)}$	$\gamma_{(s)}$
$E^{(n)}(U)$	$n + 1$	Datenpunkte für $n = 3, 5, \dots, 11$			$n - 1$	Datenpunkte für $n = 3, 5, \dots, 11$		
		-0,110259	-0,000545931	0,00737651		-0,110487	0,00163621	0,000164198
		Datenpunkte für $n = 3, 5, \dots, 15$				Datenpunkte für $n = 5, 7, \dots, 15$		
		-0,110252	-0,000586838	0,00743282		-0,110350	0,000769601	0,00138424
$D^{(n+1)}(U)$	$n + 2$	Datenpunkte für $n = 3, 5, \dots, 11$			n	Datenpunkte für $n = 3, 5, \dots, 11$		
		0,269113	-0,0117321	-0,179363		0,271661	-0,0441543	-0,0172808
		Datenpunkte für $n = 3, 5, \dots, 15$				Datenpunkte für $n = 5, 7, \dots, 15$		
		0,269222	-0,0126559	-0,177544		0,270606	-0,0352284	-0,0350123

Tabelle 4.3. Regressionsanalyse der Störungsrechnungsergebnisse für $E^{(n)}(U)$ und $D^{(n+1)}(U)$ bei $U = 4,8$ (oberer Tabellenteil). Für die Extrapolation auf $n \rightarrow \infty$ ist der Parabelparameter $\alpha_{(s)}$ (Ordinatenabschnitt; unterer Tabellenteil) bestimmend, siehe Gleichungen (4.8) und (4.9). Zur graphischen Darstellung der Ergebnisse vergleiche die Teile (e) und (f) in Abbildung 4.1.

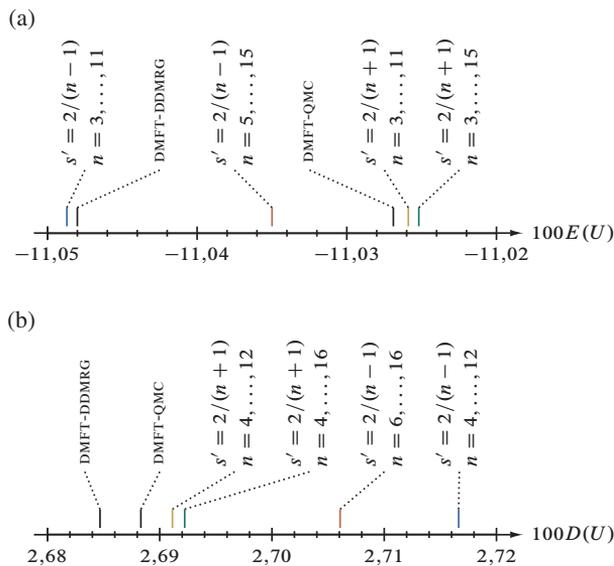


Abbildung 4.2. Graphischer Vergleich der Extrapolationsergebnisse für die Grundzustandsenergie $E(U)$ und die Doppelbesetzungszahl $D(U)$ mit DMFT-DMRG- und DMFT-QMC-Ergebnissen. Abbildungsteil (a) zeigt die Ordinatenschnitte in Teil (e) von Abbildung 4.1, dortiger Teil (f) wird hier durch Abbildungsteil (b) ergänzt.

Abschnitt tragend ist, einander graphisch gegenüber; die zugehörigen Ordinatenschnitte für $E(U)$ und $D(U)$ sind nochmals in Abbildung 4.2 zwecks leichterem Vergleichs mit den QMC- und DDMRG-Werten vergrößert aufgetragen.

4.3. Kritische Parameter des Hubbardmodells. Zur Bestimmung der kritischen Parameter folgen wir der insbesondere in den Referenzen [18] und [19] beschriebenen Vorgehensweise über die Konvergenzbetrachtung zur Störungsreihe anhand des Quotientenkriteriums, extrapolierte Störungstheorie (*extrapolated perturbation-theory*, EPT) genannt, welche auf der *Domb-Sykes-Analyse* [49] zur Deutung des asymptotischen Verhaltens von Reihen beruht. Eine verwandte Technik wurde bereits im vorausgegangenen Abschnitt verwendet; alle Hilfsdaten für die vorzunehmende Extrapolation sind in der Tabelle 4.4 zusammengestellt¹⁰⁴.

4.3.1. Definitionen. Die *kritische Wechselwirkung (Kopplung)* ist durch das Quotientenkriterium

$$U_c = \lim_{n \rightarrow \infty} \sqrt{\left| \frac{a_n^{\text{HM}}}{a_{n-2}^{\text{HM}}} \right|} = \lim_{n \rightarrow \infty} w_n \quad (4.10)$$

¹⁰⁴Vervollständigend wurden in Tabelle 4.4 auch die Ergebnisse für das hier nicht vertiefte Wurzelkriterium aufgenommen.

n	k	$\sqrt[n]{ a_n^{\text{HM}} }$	q_n	w_n
1	2	0,5		
3	3	0,793701	1	1
5	4	1,18887	4,75	2,17945
7	5	1,51750	7,80263	2,79332
9	6	1,78775	10,0662	3,17273
11	7	2,01268	11,7693	3,43064
13	8	2,20265	13,0852	3,61735
15	9	2,36533	14,1281	3,75874

Tabelle 4.4. Hilfsdaten für die Berechnung der kritischen Parameter. Abgekürzt sind $|a_n^{\text{HM}}/a_{n-2}^{\text{HM}}| = q_n$ und $\sqrt[n]{q_n} = w_n$. Orientierungshalber verzeichnet die dritte Spalte die Ergebnisse für das Wurzelkriterium.

als Konvergenzradius der Reihe (4.2) in $1/U$ gegeben. Die auftretende Wurzel wird für endliche große Werte von n durch ein Polynom zweiten Grades genähert,

$$U_c(s) \approx U_c + \frac{u_1}{s} + \frac{u_2}{s^2}, \quad (4.11)$$

und dessen drei Koeffizienten durch Ausgleichsrechnung auf Grundlage der oben tabellierten Daten bestimmt. Auf diesem Wege ist eine Näherung für U_c zugänglich, wobei die vorausgehende Wahl von $s = s(n)$ genauerer Untersuchung bedarf; diese wird im folgenden Abschnitt behandelt. Der kinetische Anteil der *kritischen Energie* E_c ist unmittelbar durch

$$E_c^{(n)} = E^{(n)}(U_c^{(n)}) \quad (4.12)$$

gegeben.

Der *kritische Exponent* $\tau \in \mathcal{Q}$, genauer $2\tau = 1, 2, \dots$, siehe Gleichung (1.6), muß der asymptotischen Relation $w_n \approx (1 - \tau/n)U_c$ für $n \gg 1$ genügen [49, 50] und läßt sich folglich genähert durch den Zusammenhang

$$\tau = -\frac{u_1}{U_c} \quad (4.13)$$

berechnen. Ist τ beispielsweise aus anderen Rechenverfahren bekannt oder existiert eine „naheliegende Annahme“ [19, 21], dann kann dies im Fitmodell (4.11) unter Reduktion auf zwei Regressionsparameter berücksichtigt werden:

$$U_c(s) \approx U_c - \tau \frac{U_c}{s} + \frac{u_2}{s^2}. \quad (4.14)$$

4.3.2. Systematische Extrapolationsanalyse. Eingeführt wurde die EPT 2004 von NILS BLÜMER und EVA KALINOWSKI [18] zur Bewertung der numerischen Ergebnisse aus den insbesondere mit QMC gelösten DMFT-Gleichungen, siehe Abschnitt 4.2, anhand der

n	s	Regressionskoeffizienten					τ
		$u_1 = -\frac{7}{2}U_c$		alle Parameter frei			
		U_c	u_2	U_c	u_1	u_2	
5	6	4,80463	6,39074	*	*	*	*
7	8	4,79207	6,43360	4,73151	-16,0850	4,63582	3,39955
9	10	4,78663	6,45836	4,74602	-16,2419	5,03292	3,42222
11	12	4,78386	6,47346	4,75458	-16,3407	5,29332	3,43684
13	14	4,78231	6,48314	4,76008	-16,4079	5,47633	3,44697
15	16	4,78136	6,48961	4,76384	-16,4560	5,61111	3,45436

Tabelle 4.5. Ergebnisse für die kritische Wechselwirkung U_c und zugehörige Fitparameter sowie den kritischen Exponenten τ . Für die Ausgleichsrechnung wurde $s = n + 1$ in den Gleichungen (4.11) und (4.14) substituiert; dabei können durch einen Stern vertretenen Werte mit zwei Datenpunkten nicht eindeutig bestimmt werden.

neunten Störungsrechnungsordnung; diese Untersuchungsmethode wurde in der physikalisch orientierten Veröffentlichung des Graphenalgorithmus' [19] auch auf die elfte Ordnung angewendet. Um einen einfachen Ergebnisvergleich auf Grundlage der beiden neuen Reihenoeffizienten zu ermöglichen, folgen wir zunächst für die Gleichungen (4.11) und (4.14) dem in den genannten Publikationen gewählten Ansatz $s = n + 1$, welcher der Ordnung der Hüpfamplitude t entspricht.

Für die in Tabelle 4.4 aufgeführten Datenpunkte bis zu $n = 15$, also die Störungsrechnungsordnung U^{-15} (respektive t^{16}), ergeben sich somit $U_c = 4,764$ und $\tau = 3,454$. Unter der Vermutung, exakt sei $\tau = 7/2$, bestimmt sich die kritische Wechselwirkung zu $U_c = 4,781$; sämtliche Daten dieser Ausgleichsrechnungen finden sich in Tabelle 4.5. Abbildung 4.3 zeigt in Teil (a) die Daten w_n aus Tabelle 4.4 mit $s = n + 1$ sowie die Wirkungsweise der EPT durch Extrapolation auf U_c für $1/s \rightarrow 0$ und in Teil (b) die Ergebnisse für U_c aus der unbeschränkten und der auf $\tau = 7/2$ eingeschränkten Ausgleichsrechnung. Man beachte, daß die Regressionskurven für U^{-11} und U^{-15} in (a) und auch in der Ausschnittsvergrößerung kaum unterscheidbar aufeinander liegen; gleiches gilt für die Datenpunkte w_{13} und w_{15} im Vergleich zu den vorhergesagten Werten auf Basis der elften Ordnung. Die (willkürlich) letzte Markierung im Extrapolationsbereich steht für w_{29} , also dem Resultat einer (hypothetischen) Rechnung der Ordnung U^{-29} . Die Werte für U_c in (b) sind monoton steigend mit n für die unbeschränkte, monoton fallend mit n für die eingeschränkte Ausgleichsrechnung. Für die kritische Energie erhalten wir

n	s	Regressionskoeffizienten					τ
		$u_1 = -\frac{5}{2}U_c$		alle Parameter frei			
		U_c	u_2	U_c	u_1	u_2	
5	4	4,41017	8,41017	*	*	*	*
7	6	4,39590	8,39998	4,35214	-10,6772	7,94591	2,45333
9	8	4,40904	8,40681	4,42697	-11,1693	8,63258	2,52302
11	10	4,42796	8,41400	4,47907	-11,5379	9,16434	2,57597
13	12	4,44715	8,41924	4,51739	-11,8259	9,59071	2,61786
15	14	4,46505	8,42264	4,54675	-12,0581	9,94175	2,65203

Tabelle 4.6. Ergebnisse für die kritische Wechselwirkung U_c und zugehörige Fitparameter sowie den kritischen Exponenten τ . Die Darstellung, hier für $s = n - 1$, erfolgt analog zur vorausgegangenen Tabelle 4.5.

durch Einsetzen des jeweiligen $U_c^{(15)}$ in Gleichung (4.2) einerseits $E_c^{(15)} = -0,1112$ für das offene (4.11), andererseits $E_c^{(15)} = -0,1107$ für das eingeschränkte Fitmodell (4.14); alle weiteren Werte sind in Tabelle 4.7 verzeichnet.

Die obige Wahl von s für die beiden Extrapolationsmodelle ist zwar naheliegend, fällt aber keineswegs eindeutig. Auf Basis der Störungsrechnung elfter Ordnung zeigt sich – wie später betrachtet wird – beispielsweise, daß die mit DDMRG berechneten DMFT-Ergebnisse [21] durch $s = n - 1$ eine ähnlich gute Übereinstimmung mit jenen der Reihenentwicklung aufweisen, wie obiger Ansatz für QMC liefert. Seitens DDMRG liegen für $n = 15$ dem Vergleich beider *solver* im folgenden Abschnitt 4.3.3 $U_c = 4,547$ und $\tau = 2,652$ zugrunde; wird $\tau = 5/2$ angenommen, so ergibt sich $U_c = 4,465$. Für $n = 5, 7, \dots, 15$ sind alle Fitkoeffizienten in Tabelle 4.6 aufgeführt. In gleicher Weise wie für die Erstanwendung der EPT zeigen die Teile (c) und (d) der Abbildung 4.3 die Resultate der Extrapolation mit $s = n - 1$; auffällig ist hier jedoch, daß die Ausgleichsparabeln für U^{-11} und U^{-15} deutlich voneinander abweichen („instabile“ Extrapolation), wie noch besser aus der Ausschnittsvergrößerung ersichtlich ist. Im Kontrast zum ursprünglichen EPT-Ansatz zeigt auch die durch die Gleichungen (4.11) und (4.14) gegebene kritische Wechselwirkung ein anderes Verhalten: Während deren Werte für den uneingeschränkten Fit ebenfalls mit zunehmenden n steigen, gilt dies gegensätzlich zu $s = n + 1$ ebenfalls (nach einem Minimum bei $n = 7$) für die beschränkte Ausgleichsrechnung. Ergänzend sind die kritischen Energien für $s = n - 1$ ebenfalls in Tabelle 4.7 eingetragen: $E_c^{(15)} = -0,1175$ für das offene, $E_c^{(15)} = -0,1201$ für das eingeschränkte Fitmodell.

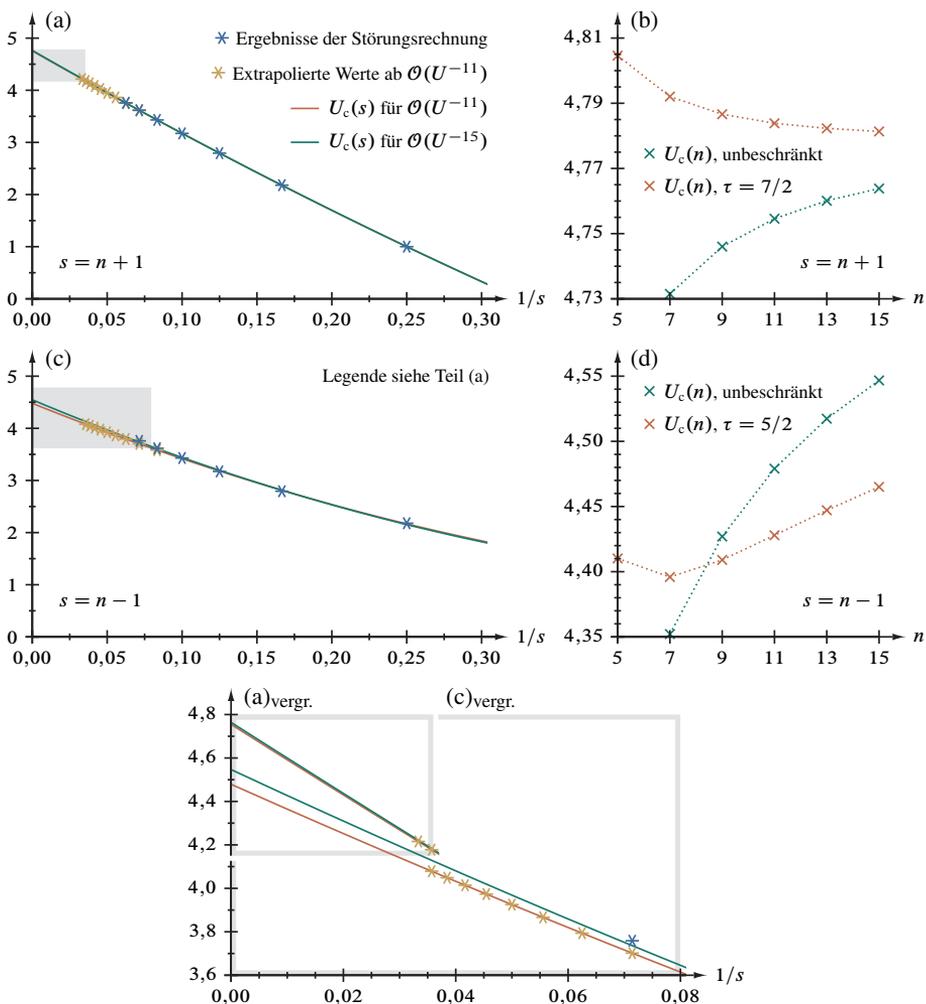


Abbildung 4.3. Kritische Wechselwirkung: Graphische Darstellung der wesentlichen Ergebnisse. In Domb-Sykes-Auftragung veranschaulichen die Abbildungsteile (a) und (c) die EPT mit dem unbeschränkten quadratischen Ausgleichspolynom für $s = n + 1$ nach Referenz [19] respektive $s = n - 1$ in Anlehnung an Referenz [21]; die grau hinterlegten Ausschnitte sind unten, auf gleichen Maßstab vergrößert, zusammengefasst dargestellt. Bei den Störungsrechnungsergebnissen handelt es sich um die Wurzeln w_n aus Tabelle 4.4. Die Teilabbildungen (b) und (d) stellen mit unterschiedlichen Ordinatenkalierungen die Werte für U_c aus unbeschränktem (4.11) und beschränktem (4.14) Fitmodell einander gegenüber, jeweils passend zu $s(n)$ aus (a) und (c); dabei dienen die Linien zwischen den diskreten Daten der Verdeutlichung der Tendenz.

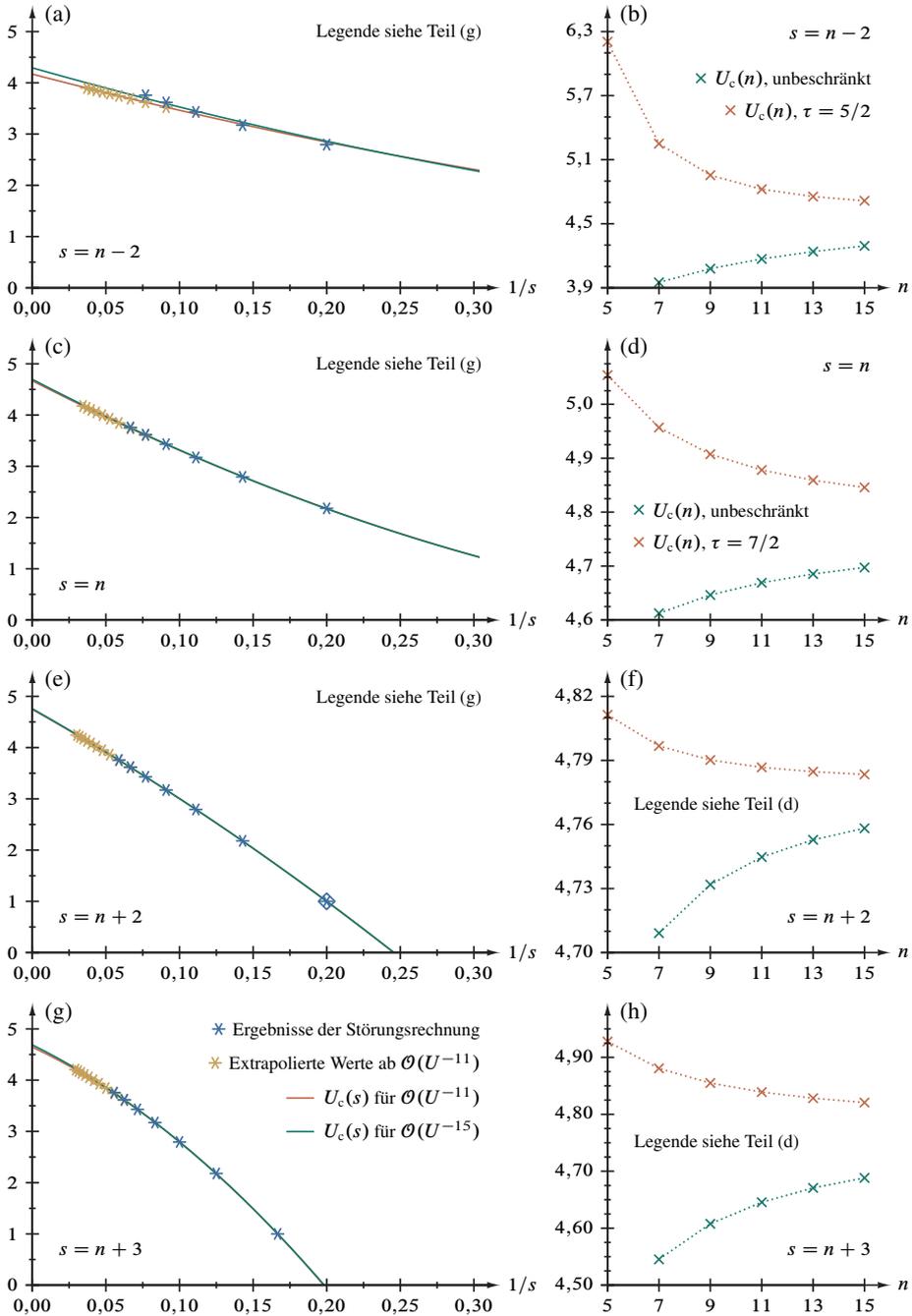


Abbildung 4.4. Kritische Wechselwirkung: Graphische Darstellung weiterer Ergebnisse in Ergänzung zu Abbildung 4.3 mit identischer Gliederung, Abbildungsteile (a) und (b) entsprechend für $s = n - 2$, Teile (c) und (d) für $s = n$, Teile (e) und (f) für $s = n + 2$ und Teile (g) und (h) für $s = n + 3$.

s	τ	$-E_c$ der Störungsrechnungsordnung n					
		5	7	9	11	13	15
$n - 1$	offen	*	0,123099	0,120946	0,119432	0,118316	0,117460
	5/2	0,120627	0,121660	0,121531	0,121084	0,120566	0,120055
$n + 1$	offen	*	0,111745	0,111509	0,111357	0,111253	0,111179
	7/2	0,109502	0,110142	0,110425	0,110571	0,110655	0,110706

Tabelle 4.7. Ergebnisse für die kritische Energie E_c . Die durch einen Stern vertretenen Werte können mit zwei Datenpunkten nicht eindeutig bestimmt werden, siehe auch Tabelle 4.5.

In konsequenter Erweiterung der beiden zuvor untersuchten Extrapolationsschemata wird nun allgemein $s = n + i$ mit $i = -2, -1, \dots, 3$ angesetzt. Ohne hier einzelne Werte der kritischen Parameter zu diskutieren, die sämtlich in Tabelle 4.8 in Ergänzung zu den Tabellen 4.5 bis 4.7 verfügbar und in Abbildung 4.4 in Erweiterung von Abbildung 4.3 dargestellt sind, ist insbesondere auffällig, daß sich die Ergebnisse für die kritische Wechselwirkung unter Einbeziehung der neuen Störungsreihenoeffizienten überwiegend auf den Bereich $4,65 \lesssim U_c \lesssim 4,85$ „konzentrieren“, siehe Abbildung 4.5.

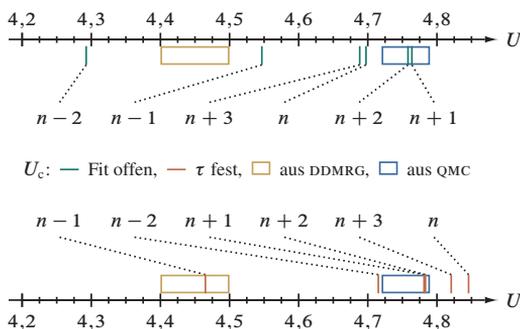


Abbildung 4.5. Graphischer Vergleich der Ergebnisse der verschiedenen Extrapolationsschemata $s(n)$ für die kritische Wechselwirkung U_c , siehe Gleichungen (4.11) für den „offenen Fit“ und (4.14) für vorgegebenes τ .

Diese Abschätzung läßt sich durch eine erneute Extrapolation für $n \rightarrow \infty$ („*advanced extrapolated perturbation-theory*“, AEPT) der oben berechneten und tabellierten Werte von U_c für die untersuchten Schemata $s(n)$ anhand des Ansatzes

$$U_c^{(n)} = \alpha + \beta \frac{1}{n} + \gamma \frac{1}{n^2} \quad (4.15)$$

s	τ	U_c der Störungsrechnungsordnung n					
		5	7	9	11	13	15
$n - 2$	offen	*	3,95036	4,07883	4,17034	4,23903	4,29262
	5/2	6,20502	5,24985	4,95437	4,82306	4,75446	4,71524
n	offen	*	4,61252	4,64638	4,66910	4,68532	4,69743
	7/2	5,05403	4,95681	4,90739	4,87816	4,85912	4,84586
$n + 2$	offen	*	4,70910	4,73182	4,74473	4,75280	4,75820
	7/2	4,81135	4,79675	4,79020	4,78675	4,78473	4,78345
$n + 3$	offen	*	4,54530	4,60773	4,64561	4,67069	4,68834
	7/2	4,92785	4,88072	4,85499	4,83906	4,82836	4,82073

s	τ der Störungsrechnungsordnung n				
	7	9	11	13	15
$n - 2$	1,64388	1,73110	1,79759	1,85050	1,89393
n	3,06926	3,11036	3,14064	3,16400	3,18265
$n + 2$	3,31543	3,36147	3,38963	3,40842	3,42171
$n + 3$	2,61628	2,78371	2,89029	2,96425	3,01863

s	τ	$-E_c$ der Störungsrechnungsordnung n					
		5	7	9	11	13	15
$n - 2$	offen	*	0,138385	0,133640	0,130335	0,127904	0,126039
	5/2	**	***	0,106185	0,109539	0,111405	0,112515
n	offen	*	0,115050	0,114273	0,113725	0,113322	0,113014
	7/2	0,103524	0,106023	0,107336	0,108125	0,108643	0,109006
$n + 2$	offen	*	0,112352	0,111894	0,111624	0,111451	0,111332
	7/2	0,109331	0,110020	0,110331	0,110495	0,110590	0,110650
$n + 3$	offen	*	0,117015	0,115387	0,114396	0,113737	0,113271
	7/2	0,106460	0,107883	0,108653	0,109124	0,109439	0,109661

Tabelle 4.8. Weitere Ergebnisse für die kritische Wechselwirkung U_c (oben), den kritischen Exponenten τ (mitte) und die kritische Energie E_c (unten) in Fortsetzung der Tabellen 4.5 bis 4.7. Die durch einen Stern vertretenen Werte können mit zwei Datenpunkten nicht eindeutig bestimmt werden; zwei Sterne stehen unten bei $s = n - 2$ und $\tau = 5/2$ für 0,082931, drei für 0,0994607.

s	Ausgehend von U_c aus der Regressionsrechnung mit					
	freien Parametern, Gl. (4.11)			vorgegebenem τ , Gl. (4.14)		
	α	β	γ	α	β	γ
$n-2$	4,68920	-6,63756	10,2648	4,91382	-7,43350	69,3453
$n-1$	4,75966	-3,49407	4,49101	4,61036	-2,81022	9,05949
n	4,78188	-1,33740	1,06273	4,76672	1,05583	1,90584
$n+1$	4,78492	-0,264411	-0,766912	4,77775	0,0124696	0,610073
$n+2$	4,78557	-0,299921	-1,64866	4,77773	0,0431088	0,625540
$n+3$	4,79381	-1,44120	-2,09059	4,77114	0,720160	0,318234

Tabelle 4.9. Ergebnisse der verschiedenen Extrapolationsschemata $s(n)$ für die kritische Wechselwirkung $U_c = \alpha$ und zugehörige Fitparameter im Grenzfall $n \rightarrow \infty$, siehe Gleichung (4.15).

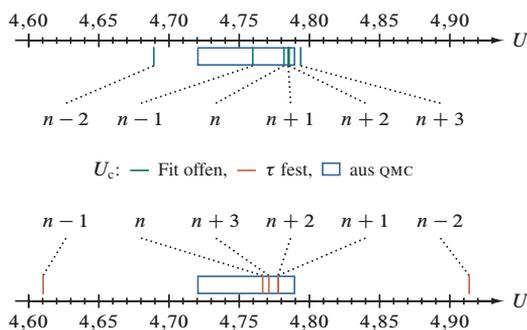


Abbildung 4.6. Graphischer Vergleich der Ergebnisse der verschiedenen Extrapolationsschemata $s(n)$ für die kritische Wechselwirkung $U_c = \alpha$ im Grenzfall $n \rightarrow \infty$, siehe Gleichung (4.15).

quantifizieren, siehe Tabelle 4.9. Wesentlicher Parameter ist dabei der Ordinatenabschnitt α , welcher U_c im Grenzfall unendlicher Störungsrechnungsordnung entspricht; ein graphischer Vergleich der Resultate für die sechs Extrapolationsschemata ist in Abbildung 4.6 zu finden. Bleiben die Befunde für $s = n-2$ und, im Falle eines vorgegeben kritischen Exponenten von $\tau = 5/2$, von $s = n-1$ unberücksichtigt, so erlaubt dieses Verfahren eine stärkere Eingrenzung des Intervalls auf $4,76 \lesssim U_c \lesssim 4,80$.

4.3.3. *Bewertung der DMFT-Befunde.* Der DDMRG-solver liefert im Rahmen der numerischen Genauigkeit für die kritische Wechselwirkung $4,4 \lesssim U_c \lesssim 4,5$ und postuliert

einen kritischen Exponenten $\tau \approx 2,5$ [21]. Beide Ergebnisse wurden 2004, ausgehend von der Störungsreihe einschließlich elfter Ordnung, durch den Extrapolationsansatz $s = n - 1$ mit $U_c = 4,479$ respektive $\tau = 2,576$ für die uneingeschränkte Ausgleichsrechnung nach Gleichung (4.11) und mit $U_c = 4,428$ unter Vorgabe von $\tau = 5/2$ gemäß Gleichung (4.14) für den kritischen Exponenten hervorragend bestätigt.

Die Einbeziehung der dreizehnten und fünfzehnten Störungsrechnungsordnung verändert dieses Bild: Bei fixiertem τ bleibt U_c zwar mit 4,465 im Bereich der DMFT-Lösung, der „offene Fit“ jedoch wandert mit $U_c = 4,547$ und $\tau = 2,652$ zu höheren Werten heraus, siehe Tabelle 4.6 und Abbildung 4.5. Auch keines der anderen EPT-Schemata unterstützt das Resultat der DDMRG, vergleiche Tabelle 4.8; ebenso folgen aus dem Extrapolationsansatz (4.15) nur noch deutliche höhere Werte der Wechselwirkung, siehe Tabelle 4.9 und Abbildung 4.6.

Das QMC-Verfahren zur Auswertung der DMFT sagt $4,72 \lesssim U_c \lesssim 4,79$ und $3,36 \lesssim \tau \lesssim 3,53$ voraus [18] – hinsichtlich der kritischen Wechselwirkung ein Ergebnis, daß auf Basis der 2004 veröffentlichten EPT mit dem Ansatz $s = n + 1$ bestens zur Störungsrechnung inklusiv elfter Ordnung mit $U_c = 4,755$ respektive $\tau = 3,437$ paßt. Wird der kritische Exponent auf $7/2$ bei der Regression festgesetzt, ergibt sich $U_c = 4,784$. Diese aus der Störungsreihe gewonnenen Daten ändern sich aus heutiger Sicht nur unwesentlich: Ohne Vorgabe von τ resultiert $U_c = 4,764$ respektive $\tau = 3,454$, mit Restriktion $U_c = 4,781$, siehe Tabelle 4.5 und Abbildung 4.5. Dieses Szenario bestätigt sich auch durch den Ansatz $s = n + 2$, die anderen Schemata liegen jedoch deutlich außerhalb der von QMC vorhergesagten Intervalle. In zweiter Extrapolationsebene, Gleichung (4.15), zeigt sich hingegen eine „Ansammlung“ von Ergebnissen im QMC-Bereich der kritischen Wechselwirkung – sowohl für den offenen wie auch den eingeschränkten Fit. Dabei ist insbesondere festzustellen, daß $s = n - 1$, welches im Rahmen der EPT das DDMRG-Ergebnis stark unterstützt hat, nun mitten im QMC-Bereich liegt, siehe Tabelle 4.9 und Abbildung 4.6.

Die Gesamtschau des Datenmaterials läßt den Schluß zu, daß die QMC-Lösungen der DMFT-Gleichungen mit der graphentheoretischen Störungsrechnung deutlich besser übereinstimmen, als die DDMRG-Ergebnisse. Während sich dieses mit den Reihenkoeffizienten einschließlich elfter Ordnung, vor allem durch das EPT-Schema $s = n + 1$ gestützt, nur andeutet und die Resultate mit $s = n - 1$ durchaus auch für DDMRG sprechen, bringen die beiden neuen Koeffizienten erheblich mehr Sicherheit. Damit erweist sich der erwartete Einklang zwischen den QMC-Werten der kritischen Parameter des Mottübergangs und den in Abschnitt 4.2 behandelten Befunden zu Grundzustandsenergie und Doppelbesetzungszahl der Isolatorphase des Hubbardmodells.

5. Parallelisierte Implementierung

Die Beantwortung der höchst bedeutsamen Frage, inwieweit sich die in Kapitel 3 umfassend beschriebene Vorgehensweise bei der graphentheoretischen Störungsrechnung prinzipiell weiter vereinfachen läßt, gar eine (teilweise) algebraische Lösung möglich

ist und ob die bereits zur Anwendung kommenden Algorithmen optimal sind, bedarf intensiver fächerumfassender Zusammenarbeit mit der Theoretischen Informatik und der Mathematik. Viele der bereits gemeinsam gewonnenen Erkenntnisse bedürfen noch eingehenderer Überlegungen; einige davon sind bereits in die vorgelegte Umsetzung eingeflossen und Gegenstand des Abschnittes 5.2 dieses Kapitels. Vorrangig jedoch wendet sich der nun folgenden Abschnitt 5.1 der praktischen (groß)rechnerischen Ausführung auf symmetrischen Multiprozessorsystemen (SMP) mit einem gemeinsamen linear adressierten Arbeitsspeicher (*shared memory*) zu; hinsichtlich des Ressourcenbedarfs genügen dabei zumeist halb-quantitative Aussagen, beispielsweise mittlere Anzahl oder asymptotische Zeit.

5.1. Technische Details. Der konkreten Programmierung der Methode geht die Wahl der Programmiersprache in Abhängigkeit von den zu nutzenden Datenverarbeitungsanlagen voraus. Grundlage für deren Auswahl ist wiederum eine Betrachtung der Skalierung von Speicher- und Rechenzeitverbrauch, die in den folgenden beiden Abschnitten umrissen wird.

5.1.1. Problemkomplexität. Ohne tiefergehende Analyse, die den Rahmen dieser Arbeit überschreitet, erschließt sich aus den bisherigen Betrachtungen nur vage, daß das Wachstum der Anzahl der zentralen Objekte des graphentheoretischen Verfahrens¹⁰⁵ für hohe Ordnungen zumindest „gedämpft“ exponentiell verläuft [24,43], $\mathcal{O}(e^{\delta n})$ mit $0 \ll \delta < 1$. Ursache dieser Unsicherheit ist insbesondere, daß für eine genauere Abschätzung der Asymptotik wesentliche Zahlenfolgen auch mit siebzehnter Störungsrechnungsordnung noch viel zu kurz sind, was Tabelle 5.1 zusammen mit dem Beispiel für λ_k belegt. Um dennoch eine Vorstellung zu erlangen, wurde auf Grundlage der Funktion $(n, p_n \in \mathbb{N} \setminus \{0\}$ und $\alpha, \beta, \gamma \in \mathbb{R}$)

$$\log(p_n) \approx \alpha + \beta n^\gamma \quad (5.1)$$

für alle auftretenden Größen p_n jeweils eine Ausgleichsrechnung durchgeführt¹⁰⁶, deren Ergebnisse ebenfalls in genannter Tabelle aufgeführt sind. Im Folgenden beschreiben wir anhand dieser empirischen Betrachtung das Skalierungsverhalten der Teilalgorithmen.

¹⁰⁵Nur für wenige „elementare Bestandteile“ kann, wie mehrfach festgestellt, die Problemskalierung unmittelbar angegeben werden [78]. Dazu gehört beispielsweise die Anzahl der möglichen \hat{S} -Exponenten-Tupel für beliebige Gitterfüllungen, siehe Fußnoten 6 und 48 sowie Gleichung (1.10), mit

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2}.$$

Ferner zeigt sich, daß etwa die Gesamtzahl der möglichen Doppelbesetzungsverläufe, d_n , vergleiche Abschnitt 3.4.7, generell bei L betrachteten Gitterplätzen exponentiell mit der Störungsrechnungsordnung n skaliert, genauer mit L^{n-1} . Auch läßt sich auf einem allgemeinen Bethegitter \mathcal{E}_z , vergleiche Abschnitt 2.2.1, grob die Anzahl der unterscheidbaren Hüpfsequenzen für einen gegebenen Hüpfprozeß mit $j = 1, 2, \dots, 2k - 2$ Schritten abschätzen, nämlich zu $(2zL)^j$, siehe dazu auch Fußnote 7.

¹⁰⁶Mit *Mathematica* [80]. Da es sich um ein nicht-lineares Fitmodell handelt, kann nur ein lokales Optimum bestimmt werden.

n	k	$\log(\lambda_k)$	$\log_2\left(\frac{b'_k}{\lambda_k}\right)$	$\log\left(\frac{p_n}{2^{(n-1)/2}}\right)$	$\log\left(\frac{d_n}{2^{(n-1)/2}}\right)$	$\log\left(\frac{h_n^{\text{HM}}}{2b_k}\right)$	$\log\left(\frac{h_n^{\text{FKM}}}{b'_k}\right)$
1	2	0	0	0	0	0	0
3	3	0,301030	1	0	0	0,397940	0
5	4	0,602060	2	0	0,397940	1,30643	0,528274
7	5	0,954243	2,91754	0,243038	0,867762	2,52421	1,36284
9	6	1,30103	3,89724	0,477121	1,39467	3,97641	2,43255
11	7	1,68124	4,82018	0,780407	1,95394	5,60773	3,68136
13	8	2,06070	5,78045	1,09419	2,52151	7,38956	5,08076
15	9	2,45637	6,71842	1,42580	3,10426	9,29853	6,60735
17	10	2,85673	7,66497	1,76211	*	*	8,24529
	$-\alpha$	0,103709	0,512905	0,377572	0,463096	0,209847	0,504832
	β	0,109322	0,519859	0,0386812	0,110237	0,114219	0,0680036
	γ	1,16427	0,972250	1,41736	1,28506	1,63496	1,71590

Tabelle 5.1. Daten zur Wachstumsanalyse der Anzahl verschiedener Objekte des graphentheoretischen Verfahrens. Durch einen Stern vertretene Werte lassen sich aus technischen Gründen derzeit noch nicht ermitteln. Die Parameter der linearen Ausgleichsrechnung auf Grundlage von Gleichung (5.1) sind in den letzten drei Zeilen angegeben.

Für die Anzahl λ_k der (zunächst „ungefärbten“) Bethegitterausschnitte A_k , deren Aufbauprinzip in Abschnitt 3.1 erklärt wurde, existiert Formel (2.8). Somit ist eine Regressionsuntersuchung beispielsweise mit $k = 200$ möglich, welche zu Gleichung (5.1) die Parameter $\alpha = -0,762871$, $\beta = 0,195815$ und $\gamma = 1,02670$ findet und auf den Grenzfall $\mathcal{O}(e^k)$ schließen läßt. Man beachte die Abweichung der Koeffizienten zum tabellierten Ergebnis für $k = 10$: Während sich der Exponent γ zu eins hin und die Konstante α über null hinaus verringert, vergrößert sich der Vorfaktor β .

Bekannt ist im Gegensatz zur Baumanzahl eine Obergrenze für die Skalierung der Ausgangszustandsmenge B'_k (Betrag: b'_k) im Falle von Elektronen für jede Struktur bezogen auf eine Wurzelfarbe¹⁰⁷, nämlich $\mathcal{O}(2^{k-1})$. Gleiches gilt mit $\mathcal{O}(2^{(n-1)/2})$ für die Klassenanzahl der Hüpfprozesse (Summe über alle Klassen: p_n) und der in dieser Arbeit

¹⁰⁷Zum Verhältnis der Ausgangszustandsanzahl zur Anzahl zweifarbiger Butcherbäume, $2b'_k/2^k b_k$, siehe Tabelle 3.7.

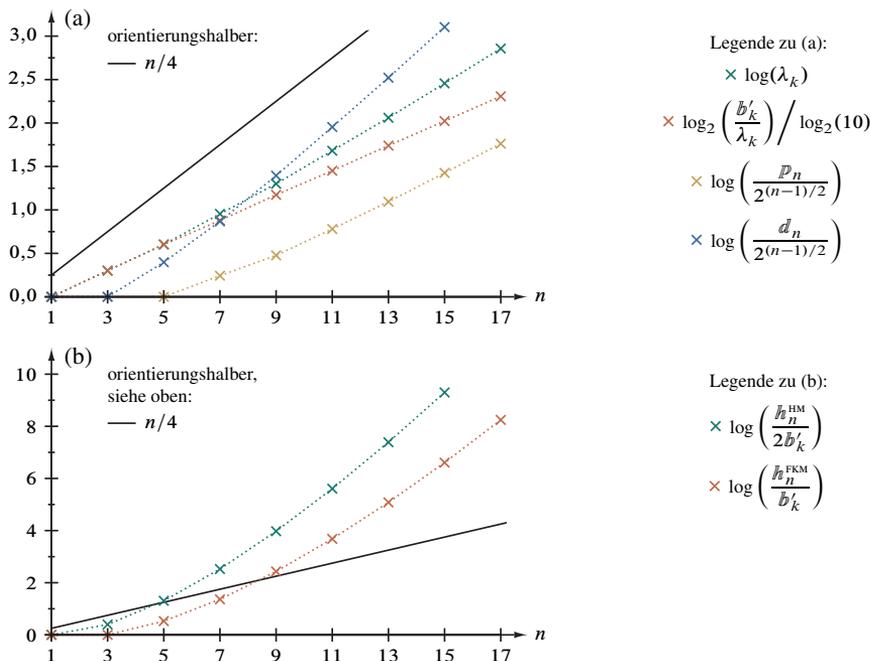


Abbildung 5.1. Graphische Darstellung der Daten zur Wachstumsanalyse aus Tabelle 5.1. Abbildungsteil (a) zeigt die normierten Werte (siehe Legende) von Butcherbaum-, Ausgangszustands-, Hüpfprozess- und Doppelbesetzungssequenzanzahl; Teil (b) veranschaulicht die Hüpfsequenzanzahl für das Hubbard- (bis einschließlich U^{-15}) und das Falicov-Kimball-Modell (bis einschließlich U^{-17}). Die Linien zwischen den diskreten Daten dienen der Verdeutlichung der Tendenz.

nur intermediär in den Hüpfsequenzen auftretenden Doppelbesetzungsverläufe (d_n); innerhalb einer Klasse wächst deren Anzahl stärker als exponentiell, aber schwächer als die Fakultät: $\mathcal{O}(n!^\delta)$ mit $0 \ll \delta < 1$.

Faktorielle Skalierung weisen die Hüpfsequenzanzahlen pro Ausgangszustand für das Falicov-Kimball- (Summe über alle Ausgangszustände: h_n^{FKM}) und das Hubbardmodell (h_n^{HM}) auf, da der Sequenzierungsalgorithmus im Grenzfall – wenn $k - 1$ Gitterplätze für einen Initialisierungssprung in Frage kommen und jede Kantenpermutation eine vollständige Hüpfsequenz liefert – den Arbeitsaufwand $\mathcal{O}((2k - 2)!/2)$ hat. Zwar können durch Binärbaumverzweigungen ab $n = 5$ verstärkend zwei, ab $n = 9$ vier Sequenzen und so weiter pro Permutation auftreten, aber durch die praktische Anwendung der *greedy*-Methode wird dieser Zuwachs weit überkompensiert. Zahlenwerte zu diesen Abschätzungen sind gleichfalls in Tabelle 5.1 numerisch aufgeführt und sämtliche Daten in Abbildung 5.1 graphisch dargestellt.

Um eine quantitative Vorstellung vom effektiven Anwachsen des Gesamtaufwandes zu erhalten, untersuchen wir die Folge der Hüpfsequenzensummen des jeweiligen Modells mit (zu Vergleichszwecken) zwei Näherungsfunktionen $(n, \mathbb{h}_n \in \mathbb{N} \setminus \{0\}$ und $\beta \in \mathbb{R})$. Einerseits ist dies, ausgehend von der Stirling-Formel,

$$\ln(\mathbb{h}_n) \approx \beta(n \ln(n) + \frac{1}{2} \ln(2\pi n) - n) , \tag{5.2}$$

andererseits eine *approximative inverse Faktorielle*,

$$\text{aif}(\mathbb{h}_n) \approx \beta n ; \tag{5.3}$$

letztenannte stellt eine numerisch-nähernde Umkehrfunktion der Fakultät dar und ist auf eine (definitionsbereichsseitig naturgemäß eingeschränkte) Inverse der Gammafunktion [51], $\text{aig}: [1, \infty) \subset \mathbb{R}^+ \rightarrow \mathbb{R}^+$, gegründet¹⁰⁸. Erwartungsgemäß zeigen von den vorgenannten Gleichungen ausgehende Regressionsrechnungen übereinstimmende Ergebnisse für den freien Parameter, siehe Tabelle 5.2: $\beta > 1$ für das Hubbard- und $\beta < 1$ für das Falicov-Kimball-Modell, wobei $\beta = 1$ einem exakt Fakultätsfunktions-gleichen Anwachsen entspräche. Berücksichtigt muß aber werden, daß der Zugang über die inverse Faktorielle in Gleichung (5.3) bei kleinen Ordnungen genauer als der „Stirling-Ansatz“ in Gleichung (5.2) ist¹⁰⁹ und folglich das bessere Werkzeug für Komplexitätsuntersuchungen dieser Art darstellt.

Übersichtshalber sind in Tabelle 5.3 die Gesamtanzahlen der für das Verfahren charakteristischen diskreten Strukturen in Exponentialdarstellung mit drei signifikanten Stellen angegeben. Ergänzt werden diese, vorbereitend auf den folgenden Abschnitt, um die Gesamtrechenzeiten für beide Modelle in den jeweiligen Ordnungen. Abbildung 5.2 vermittelt anhand der Zahlenwerte einen graphischen Eindruck der Problemkomplexität.

5.1.2. Laufzeit- und Speicherbedarf. Auf Grundlage der Betrachtungen des vorausgegangenen Abschnitts ist noch keine konkrete Angabe des Bedarfs an Datenverarbeitungsressourcen möglich, denn dazu bedürfte es einer tiefergehenden Algorithmenanalyse hinsichtlich des Zeitbedarfs für die Ausführung der jeweiligen Rechenoperationen und des Speicherbedarfs der jeweils beteiligten Datenstrukturen. Letztgenannte Größe kann nach

¹⁰⁸Die approximative inverse Faktorielle (aif) und die in Referenz [51] hergeleitete approximative inverse Gammafunktion (aig) unterschieden sich durch $n! = \Gamma(n + 1)$ entsprechend nur in einem Vorzeichen. Beide Umkehrfunktionen sind mit W_0 , dem oberen Ast der Lambertschen W -Funktion, wie folgt definiert:

$$\left. \begin{aligned} \text{aif}(p_n) \\ \text{aig}(p_n) \end{aligned} \right\} = \frac{\ell(p_n)}{W_0(\ell(p_n)/e)} + \frac{1}{2} \cdot \begin{cases} -1 \\ +1 \end{cases} \text{ mit } \ell(p_n) = \ln\left(\frac{p_n + c}{\sqrt{2\pi}}\right) \text{ und } c = \frac{\sqrt{2\pi}}{e} - \Gamma(d) .$$

Darin ist $d \approx 1,46163214496836234$ das numerisch zu bestimmende Minimum im positiven Definitionsbereich der Gammafunktion (respektive die Nullstelle deren Ableitung in Form der Digammafunktion) und somit eben jener Punkt, ab welchem die Gammafunktion im offenen Intervall $[d, \infty)$ bijektiv, also eindeutig umkehrbar ist.

¹⁰⁹Der relative Fehler liegt bei $2,08 \cdot 10^{-2}$ gegenüber $-8,44 \cdot 10^{-2}$ für 1!, bei $8,63 \cdot 10^{-4}$ gegenüber $-4,22 \cdot 10^{-2}$ für 2! und bei $-1,69 \cdot 10^{-4}$ gegenüber $-8,37 \cdot 10^{-3}$ für 10!.

n	k	$\ln(\beta_n^{HM}/2)$	$\text{aif}(\beta_n^{HM}/2)$	$\ln(\beta_n^{FKM})$	$\text{aif}(\beta_n^{FKM})$
1	2	0	1,02120	0	1,02120
3	3	2,30259	3,38490	1,38629	2,66056
5	4	5,78074	5,56197	3,98898	4,51461
7	5	10,0317	7,72795	7,35756	6,40577
9	6	14,8531	9,89120	11,2982	8,31907
11	7	20,1246	12,0530	15,6889	10,2457
13	8	25,7667	14,2135	20,4505	12,1816
15	9	31,7235	16,3727	25,5268	14,1242
17	10	*	*	30,8763	16,0720
	β	1,14587	1,09567	0,910602	0,936469

Tabelle 5.2. Daten zur Wachstumsanalyse der Hüpfsequenzanzahl im Vergleich. Durch einen Stern vertretene Werte lassen sich aus technischen Gründen derzeit noch nicht ermitteln. Der Parameter der linearen Ausgleichsrechnung auf Grundlage der Gleichungen (5.2) und (5.3) ist in der letzten Zeile angegeben.

Festlegung der Datenkodierung, zumeist in Ganzzahlfeldern aus Worten halber oder voller Registerbreite (64 Bit), weitgehend maschinenunabhängig angegeben werden. Im hohen Grade maschinenspezifisch und somit für allgemeine Aussagen schwer zugänglich ist hingegen die Ausführungszeit für die einzelnen Operationen und die Latenzen für Speicherzugriffe (*non-uniform memory access*, NUMA)¹¹⁰. Im Rahmen dieser Arbeit verfolgen wir daher einen ebenso praktischen wie einfachen Ansatz, indem wir anhand der Skalierungsuntersuchungen und der „Meßwerte“, insbesondere für die Rechenzeit, *Abschätzungen* für die nächsthöhere Ordnung vornehmen.

Tabelle 5.3 und Teil (a) der Abbildung 5.2 zeigen, daß der Speicherplatzbedarf für die dem Kernalgorithmus bereitgestellten Daten, vornehmlich die Ausgangszustände und die Hüpfprozesse, im wesentlichen exponentiell mit der Störungsrechnungsordnung zunehmen. Diesem steht die faktoriell wachsende Anzahl an Hüpfsequenzen gegenüber, Abbildungsteil (b), aus welchen jedoch unmittelbar nach Generierung der Betrag zum Reihenkoeffizienten berechnet wird und die keiner dauerhaften Speicherung bedürfen. Entsprechend verhält sich die Entwicklung des Laufzeitbedarfs: Der Rechenaufwand

¹¹⁰Obwohl der gesamte Arbeitsspeicher auf den großen Parallelrechnern in der verwendeten Konfiguration aus *software*-Sicht allen Prozessoren zur gemeinsamen Verfügung steht, ist der Zugriff darauf nicht von einheitlicher Zeit, da die Speicherverwaltung *hardware*-seitig (Speicherbausteine und -kontrollogik) auf alle Prozessoren aufgeteilt ist und die Datenwege gegebenenfalls über Zwischeninstanzen verlaufen.

n	k	λ_k	b'_k	p_n	d_n	Modell				R.
						Falicov-Kimball		Hubbard		
						h_n	Mz./s	$h_n/2$	Mz./s	
1	2	$1 \cdot 10^0$	*	$1 \cdot 10^0$	*	[85]				
3	3	$2 \cdot 10^0$	$4 \cdot 10^0$	$2 \cdot 10^0$	$2 \cdot 10^0$	$4 \cdot 10^0$	*	$1,0 \cdot 10^1$	*	[85]
5	4	$4 \cdot 10^0$	$1,6 \cdot 10^1$	$4 \cdot 10^0$	$1,0 \cdot 10^1$	$5,4 \cdot 10^1$	$1 \cdot 10^{-2}$	$3,24 \cdot 10^2$	$1 \cdot 10^{-2}$	[85]
7	5	$9 \cdot 10^0$	$6,8 \cdot 10^1$	$1,4 \cdot 10^1$	$5,9 \cdot 10^1$	$1,57 \cdot 10^3$	$6 \cdot 10^{-2}$	$2,27 \cdot 10^4$	$1,2 \cdot 10^{-1}$	[85]
9	6	$2,0 \cdot 10^1$	$2,98 \cdot 10^2$	$4,8 \cdot 10^1$	$3,97 \cdot 10^2$	$8,07 \cdot 10^4$	$1,32 \cdot 10^0$	$2,82 \cdot 10^6$	$1,08 \cdot 10^1$	[85]
11	7	$4,8 \cdot 10^1$	$1,36 \cdot 10^3$	$1,93 \cdot 10^2$	$2,88 \cdot 10^3$	$6,51 \cdot 10^6$	$6,29 \cdot 10^0$	$5,50 \cdot 10^8$	$8,29 \cdot 10^0$	[86]
13	8	$1,15 \cdot 10^2$	$6,32 \cdot 10^3$	$7,95 \cdot 10^2$	$2,13 \cdot 10^4$	$7,61 \cdot 10^8$	$9,70 \cdot 10^1$	$1,55 \cdot 10^{11}$	$1,04 \cdot 10^3$	[86]
15	9	$2,86 \cdot 10^2$	$3,01 \cdot 10^4$	$3,41 \cdot 10^3$	$1,63 \cdot 10^5$	$1,22 \cdot 10^{11}$	$4,52 \cdot 10^3$	$5,99 \cdot 10^{13}$	$4,18 \cdot 10^5$	[86]
17	10	$7,19 \cdot 10^2$	$1,46 \cdot 10^5$	$1,48 \cdot 10^4$	**	$2,57 \cdot 10^{13}$	$6,86 \cdot 10^5$	**	**	[86]
		2.1	1.3 u. 3.7	3.4	3.10	3.8		3.8		

Referenznummer der Tabelle mit den exakten Werten

Tabelle 5.3. Komplexität des Störungsrechnungsproblems. In dieser Tabelle steht Mz. für Maschinenzeit, also die „Realzeit“ der Computernutzung in Sekunden für die n te Ordnung nach dem Ablaufschema des Gesamtverfahrens, Abbildung 3.1. Man beachte, daß für $n = 11, 13, 15$ nicht mit „dynamisch wachsenden“ (*multiple precision*), sondern konstant 128 Bit langen Ganzzahlen bei circa 0,65fachem Aufwand gerechnet wurde. In der Spalte R. wird auf den verwendeten Rechner verwiesen (s. Verz. d. Datenverarbeitungsressourcen): [85] RISC-workstation vom Typ HP A9638A, zwei Prozessoren (seriell, Halbauslastung); [86] EPIC-computeserver vom Typ HP AH353A, 256 Prozessorkerne (parallel, Vollauslastung). Mit einem Stern gekennzeichnete Rechenzeiten liegen unterhalb zuverlässiger Messbarkeit (< 10 ms); durch zwei Sterne vertretene Werte lassen sich aus technischen Gründen derzeit noch nicht ermitteln.

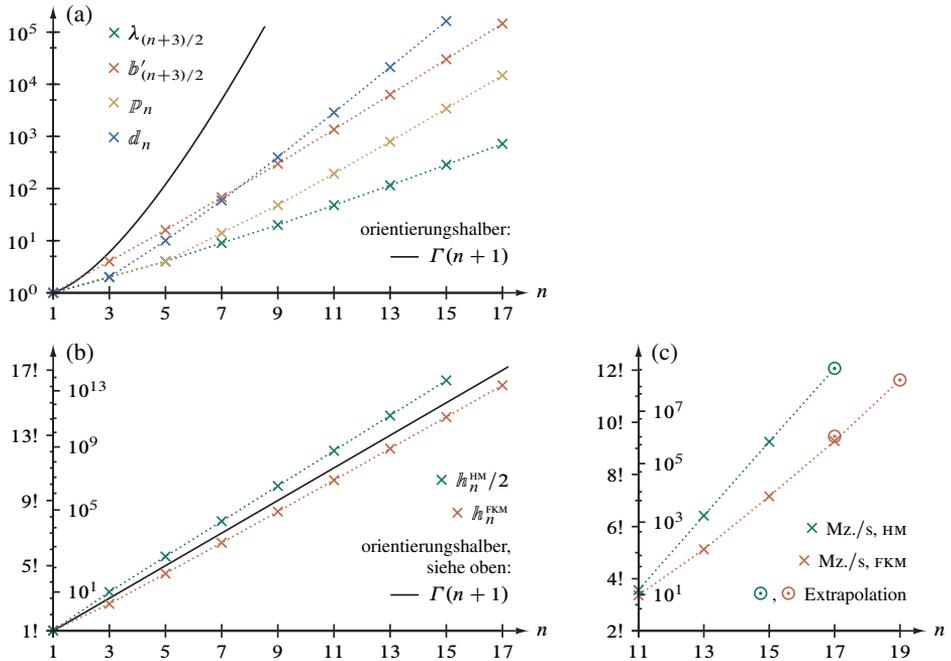


Abbildung 5.2. Veranschaulichung der Komplexität des Störungsrechnungsproblems auf Grundlage der Daten aus Tabelle 5.3. In Abbildungsteil (a) sind halblogarithmisch Butcherbaum-, Ausgangszustands-, Hüpfprozeß- und Doppelbesetzungssequenzanzahl dargestellt. Teil (b) zeigt die Hüpfsequenzanzahl für Hubbard- (bis einschließlich U^{-15}) und Falicov-Kimball-Modell (bis einschließlich U^{-17}) in faktorieller Auftragung. Bei den zugehörigen Rechenzeiten in Teil (c), angegeben als Maschinenzeit (Mz.) in Sekunden, handelt es sich um die Gesamtnutzungsdauer des EPIC-computeservers HP AH353A [86], wobei die Werte für $n = 17$ und $n = 19$ extrapoliert wurden. Die Linien zwischen den diskreten Daten dienen der Verdeutlichung der Tendenz.

für die Ausgangsdatenerzeugung wächst durch kombinatorische Algorithmentelemente¹¹¹ teilweise zwar stärker als exponentiell¹¹², zumal intermediär erheblich mehr Daten entstehen¹¹³, ist aber gegenüber dem eindeutig faktoriell ansteigendem Aufwand für die

¹¹¹Siehe dazu Gleichung (3.7) in Abschnitt 3.1.1 und Gleichungen (3.12) und (3.17) in Abschnitt 3.2.1 (Baumerzeugung) sowie Gleichung (3.50) und (3.52) in Abschnitt 3.3.2 (Hüpfprozeßerzeugung).

¹¹²Siehe dazu auch die Anzahlzerlegungen in Tabelle 5.1 und Abbildung 5.1.

¹¹³Insbesondere vierfarbige Butcherbäume (Abschnitt 3.2) in komprimierter Zeichenkettendarstellung für künftige Erweiterungen des Verfahrens. Man beachte auch, daß nach Zwischenschritten in der Reihenentwicklung des Hamiltonoperators (Abschnitt 3.3) – spezifisch für die Grundzustandsenergie – eine deutliche Verminderung des Datenvolumens vorgenommen wird.

Sequenzierung der ihrerseits exponentiell zunehmenden Bethegitterausschnitte in jeder Ordnung vernachlässigbar.

Ebenso fällt der Bedarf an Arbeitsspeicher gegenüber der Rechenzeit nicht ins Gewicht: Die für die bisher erreichten Ordnungen mindestens erforderlichen Anlagen der Klasse HP AH353A stellen bis zu $512 \cdot 2^{30}$ Datenworte¹¹⁴ bereit, während – modellunabhängig – die Rechnung für U^{-13} lediglich $188 \cdot 2^{20}$, für U^{-15} nur $2,25 \cdot 2^{30}$ und auch für U^{-17} nicht mehr als $16,3 \cdot 2^{30}$ Worte benötigt. Folglich ist davon auszugehen, daß selbst für die Störungsrechnungsordnung U^{-19} kein Mangel an dynamischen Speicher eintritt, während (wie unten abgeschätzt wird) die Rechenzeit bereits um Größenordnungen oberhalb des auf solchen Maschinen verfügbaren liegt. Das verhältnismäßige „Überangebot“ an Speicher, welches sich auf größeren Systemen linear fortsetzt, ermöglicht aber einige Verbesserungen datentechnischer Art. Bei einem Verhältnis von $2 \cdot 2^{30}$ Worten pro Prozessor(kern) können künftig globale Datenstrukturen, beispielsweise die gesamte Prozeßexponenten- und Prozeßkoeffiziententafel (für $n = 17$ etwa $32,5 \cdot 2^{20}$ Worte), mit signifikanter Einsparung an Zugriffszeit problemlos in *thread*-lokaler Kopie bereitgestellt werden. Auf solche für die Rechnerarchitektur spezifischen Aspekte wird kurz im folgenden Abschnitt eingegangen.

Eindrucksvoll belegen die Tabellen 3.3 und 3.4 (Prozeßanzahl), 3.1 und 3.7 (Ausgangszustandsanzahl) sowie 3.8 (Sequenzanzahl), daß die Prozessorzeit, also die akkumulierte reale Rechenzeit aller Prozessoren, für Baum- und Prozeßerzeugung um Größenordnungen unter jener für die Sequenzerzeugung liegen (Angaben für das Hubbardmodell und einen *CISC-server* vom Typ HP AM447A [84]): Für U^{-11} sind dies Sekunden (36,8 beziehungsweise 21,8) gegenüber Stunden (1,68), für U^{-13} bereits Minuten (20,7 beziehungsweise 3,87) gegenüber Wochen (3,54). Konkreter noch wird die Skalierung des Gesamtaufwandes bei Betrachtung der Zeit, aufgeführt in Tabelle 5.3 und veranschaulicht in Teil (c) der Abbildung 5.2, welche die Maschine mit der Abarbeitung aller Programmteile der Störungsrechnung gegebener Ordnung beschäftigt ist (Angaben für das Hubbardmodell und einen *EPIC-computeserver* vom Typ HP AH353A [86]): 256 Prozessorkerne berechnen die Ordnung U^{-11} in 8,29 Sekunden, U^{-13} in 17,3 Minuten, U^{-15} in 4,83 Tagen.

Der in Abschnitt 5.1.1 vorgestellte Ansatz zur „Datenlinearisierung“ über die Fakultätsumkehrfunktion ermöglicht nun die grobe Abschätzung des Laufzeitbedarfs der folgenden Störungsrechnungsordnung; dazu bedienen wir uns wiederum einer Ausgleichsrechnung auf Grundlage eines quadratischen Polynoms. Ausgehend von den Ergebnissen¹¹⁵ für $n = 11, 13, 15$ kann die Qualität der Vorhersage geprüft werden, da die Maschinenzeit für das Falicov-Kimball-Modell für U^{-17} mit 7,94405 Tagen vorliegt: Die Extrapolation überschätzt mit 12,0543 Tagen um circa 51,7 %, wobei allerdings zu berücksichtigen ist, daß der für sich geringe Fehler der Regressionsrechnung von 1,98 %

¹¹⁴ $512 \cdot 2^{30}$ Worte = 512 GWorte. Bei einer Wortbreite von 64 Bit entspricht diese Angabe 4096 GBytes.

¹¹⁵Die tabellierten Werte sind mit dem Faktor 1,54912 zu skalieren, da Rechnungen ab $n \geq 17$ zeitaufwendigere Ganzzahlen über 2^{128} erfordern, also mehr als zwei 64 Bit-Register zu kodieren vermögen.

sich mit der Fakultät vergrößert. Wird der exakte Wert für $n = 17$ in die Ausgleichsrechnung einbezogen, dann müssen für U^{-19} auf [86] etwa 5,92 Jahre veranschlagt werden, wobei ein kubisches Fitpolynom bloß 2,66 Jahre ansetzt. Eine analog erfolgende Rechnung für das Hubbardmodell läßt 18,1 Jahre für U^{-17} erwarten; da nur drei Meßpunkte zur Verfügung stehen, ist zwar eine kubische Abschätzung hier nicht durchführbar, aber die Ergebnisse des Falicov-Kimball-Modell legen eine deutlich zu hohe Schätzung nahe. Mögen zwar diese Zahlen auf den ersten Eindruck hin „astronomisch“ wirken, so ist doch die Berechnung dieser Ordnungen auf *cluster*-basierten Großrechenanlage mit mehreren hunderttausend Prozessorkernen vollkommen realistisch, wie in Kapitel 6 angesprochen wird.

5.1.3. *Programm und Rechner.* Für die konkrete rechenmaschinelle Umsetzung des optimierten Gesamtalgorithmus' wurde die Programmiersprache C gewählt und weitestmöglich durch Prüfung mit *Splint* [81] der ANSI-Standard eingehalten. Diese Entscheidung kommt innerhalb des imperativen Paradigmas¹¹⁶ einerseits der iterativ-rekursiven Charakteristik, andererseits den Datentypen¹¹⁷ des Verfahrens am weitesten entgegen; zudem zeigt keine andere Sprache eine derartig ideale Verbindung von Effizienz, Portabilität und Compilerqualität¹¹⁸. Um den zum Teil erheblichen Entwicklungsaufwand nicht weiter zu erhöhen und damit auch die Übertragbarkeit auf andere und künftige Prozessorbefehlssätze [52] nicht zu erschweren, wurde im Rahmen dieser Arbeit zunächst auf die direkte Einbindung von *Assembler*-Anweisungen [53] verzichtet, wiewohl dieser Programmiererebene erhebliches und im Bereich des Hochleistungsrechnens oft ungenutztes Potential innewohnt.

Die Neuimplementierung der Teilalgorithmen zur Reihenentwicklung des Hamiltonoperators und insbesondere zur Erzeugung der Hüpfsequenzen baut mit hoher Wirksamkeit vollständig auf die (im Rahmen der Datengranularität¹¹⁹ ideale) intrinsische Datenparallelität. Dabei erschien für die erstmalige Parallelisierung ein durch den Compiler-Präprozessor bewirktes *multi threading* auf Grundlage des effizienten, sehr ausgereiften und Hersteller-übergreifend standardisierten *OpenMP* als bester Kompromiß. Im Gegensatz zur Verwaltung parallellaufender Prozesse mit ihren jeweiligen „privaten“ und gemeinsam genutzten Daten sowie der entsprechenden Interprozeßkommunikation auf Sprachebene¹²⁰, wird damit zwar eine immense Verringerung von Programmieraufwand

¹¹⁶Komplementär ist dazu die Implementierung [33] mit *Haskell*, welches unter den funktionalen Sprachen mit den höchsten Entwicklungsgrad aufweist [79].

¹¹⁷Da sich alle diskreten Strukturen und Operationen daran auf einfache Weise maschinennah-elementar kodieren lassen, siehe dazu Abschnitt 2.2.3, teilweise Abschnitt 2.2.7 und insbesondere Abschnitt 3.4.4, wurde keine Notwendigkeit für objekt-orientierte Ansätze wie C++ gesehen.

¹¹⁸Allenfalls *Fortran*, dessen unzweifelhafte Stärke aber in der Fließkommanumerik (beispielsweise durch die „natürliche“ Behandlung komplexer Zahlen) zu sehen sind.

¹¹⁹Die Ausführungszeit des parallelen Verarbeitungsabschnitts ist bestenfalls jene des „größten Einzelauftrags“ darin (Teilaussage des Amdahlschen Gesetzes, siehe dazu auch Abschnitt 6.3.4).

¹²⁰Einen naheliegenden portablen Standard stellt *POSIX Threads* dar. Ebenfalls als den Sprachumfang erweiternde Bibliothek ist beispielsweise das derzeit bedeutende *MPI (message passing interface)* umgesetzt.

und Quelltextkomplexität erreicht – bedeutsam für die naheliegende und bereits teilweise vorbereitete Erweiterung der Methode –, indes auch die Beschränkung auf SMP-Architekturen mit bestenfalls 512 Prozessoren in Kauf genommen. Da nach heutigem Stand größere universelle Anlagen jedoch nahezu ausschließlich als *cluster* aus Multiprocessorsystemen oder als Massivparallelrechner (*massively parallel-processing*, MPP)¹²¹ realisiert werden, ist eine Überwindung der Ressourcenobergrenze durch eine künftige *Hybridparallelisierung* auf Compiler- und Sprachebene ohne Modifikation des Bisherigen uneingeschränkt möglich.

Die Länge des Quelltextes beträgt bei kompakter Formatierung und nach Entfernen der Kommentare etwa 3500 Zeilen; die „Rohfassung“ mit Unterprogrammvarianten für spätere Anwendungen umfaßt etwa 7100 Zeilen.¹²² Darin sind bereits alle über den genormten Sprachumfang hinausgehenden benötigten Funktionen, im Besonderen kombinatorische wie spezielle Permutations- und Partitionierungsverfahren (siehe Anhang) enthalten, die auf modifizierten Standardalgorithmen [43] beruhen. Folglich kann auf erweiternde Bibliotheken diesbezüglich verzichtet und somit – neben teilweise erheblichen Leistungsgewinnen durch die Berücksichtigung von Sonderfällen und die *in-line*-Ausführung von Hilfsroutinen, beispielsweise zum Sortieren – auch die Verwendung auf unterschiedlichen Betriebssystemen vereinfacht werden. Einzig durch die Bereitstellung eines Ganzzahldatentyps mit dynamisch beliebig vergrößertem Wertebereich und der zugehörigen Arithmetik durch GMP [82] wird von diesem Grundsatz abgewichen¹²³; hierin

¹²¹In diesen Kontext fällt auch die derzeit populäre, ursprünglich zweckentfremdende Nutzung von Graphikprozessoren (*general purpose computations on graphics processing units*, GPGPU) mit den Entwicklungsumgebungen OpenCL und CUDA. Diese auf Algorithmen-intrinsische Parallelität abzielende Technik stellt zwar sehr viele Recheneinheiten, in der Größenordnung einiger tausend, mit in Summe immenser Leistung bereit, welche jedoch herkunftsgemäß nur über einen im Wesentlichen auf Gleitkommaoperationen für Vektoren und Matrizen hin herstellerspezifisch eingeschränkten Befehlssatz sowie ein vergleichsweise ungünstiges Verhältnis zwischen Kernanzahl und gemeinsamen (Steckkarten-lokalen) Arbeitsspeicher verfügen (maximal 546 kWorte/Kern bei 12 GBytes/Karte, Speicherfehlerkorrektur verringert diesen Wert um den Faktor 15/16). Vor allem letztgenannte Einschränkung – unerheblich, sofern alle *threads* nur wenig „eigenen“ Speicher benötigen – steht jedoch Daten-parallelen Verfahren wie dem vorgelegten mit relativ hohem Speicherbedarf für jeden Rechenprozeßfaden entgegen.

Eine durch bedeutend höhere Flexibilität und geringen Programmiermehraufwand attraktive Alternative dazu bilden die Intel Xeon Phi-Kopprozessoren mit bis zu 61 Kernen (à 4 Registersätze, siehe Fußnote 130), welchen bei einem besseren Speicher-Kern-Verhältnis (33,6 MWorte/Kern ohne Fehlerkorrektur, zum Vergleich HP AM447A [84]: 1,60 GWorte/Kern mit Korrektur) eine zu den Hauptprozessoren des notwendigen *host*-Rechners eng verwandte Architektur (CISC-SMP) und daher bei ähnlicher Leistung ein nahezu identisches Programmiermodell zugrunde liegen. Folglich wäre mit einer Übertragung der Algorithmen eher auf dieser Grundlage anzusetzen.

¹²²Präprozessordirektiven steuern die Generierung einer stark ausführungsbeschleunigten Produktions- oder einer funktionserweiterten Entwicklungsversion.

¹²³Die Funktionen der GNU-*multiple-precision*-Bibliothek sind zwar auf Niveau der Maschinensprachen sämtlicher genutzter Mikroprozessoren durch Assembler-Instruktionen optimiert [82], ein effizienterer Doppelwortbreiter vorzeichenloser Datentyp mit dem Wertebereich $[0, 2^{128} - 1]$ (64 Bit-Ganzzahl und -Überlaufzähler in *hardware*, Arithmetik in *software*) steht hingegen neben vorzeichenbehafteten Rationalzahlen (64 Bit-Zähler und -Nenner in *hardware*, Arithmetik in *software*) bereits im Quelltext zur Verfügung.

besteht auch der einzige Portierungsaufwand. Um eine ausgewogene Auslastung (*load balancing*) der Parallelrechner zu erreichen, wurde (neben entscheidenden algorithmischen Anpassungen, siehe Abschnitt 5.2.2 und darin besonders die Lastprofile in den Abbildungen 5.3 und 5.4) auf OpenMP-Ebene umfassend die Auftragsvergabe vom *master* an die *slave threads* untersucht und *vor* der Übersetzung in den Maschinenkode auf vereinfachtem Wege empirisch angepaßt¹²⁴.

Für die Übersetzung in das Maschinenprogramm wurde im Rahmen der Entwicklung [85] und der „Produktionsrechnungen“ [86] unter dem Betriebssystem HP-UX der ANSI-C-Kompiler des Rechnerherstellers (Hewlett-Packard) genutzt; für die umfangreichen Prüfrechnungen [87–89] und Laufzeitmessungen [84] unter Linux wurden ausschließlich die Kompiler des Prozessorherstellers (Intel) verwendet. Beide kommerziellen Kompiler unterstützen somit die *hardware* (vornehmlich Maschinenbefehlssatz und Pufferspeichergrößen) in optimaler Weise und führen zu einem Leistungsgewinn von bis zu 35 % gegenüber dem lizenzkostenfreien GNU-Kompiler. Geschwindigkeitsverbesserungen des Maschinenkodes wurden kompilerseitig auf zwei Ebenen vorgenommen, konventionell zunächst während der Übersetzung aus dem Quellcode auf „konservativer Stufe“, anschließend nach Programmausführungen für unterschiedliche Ausführungsparameter mit Laufzeitdatensammlung auf „höchstmöglich progressiver Stufe“ einschließlich Schleifenauflösungen, Vektorisierungen, funktionsübergreifende Optimierungen (*inter-procedural optimizations*, IPO) und – insbesondere – Laufzeitprofilgeführte Optimierungen (*profile-guided optimization*, PGO).

Das Ablaufverhalten des Gesamtverfahrens wurde einerseits auf RISC- (*reduced instruction-set computing*) [85]¹²⁵ und CISC-Prozessoren (*complex instruction-set computing*) [84, 87, 88]¹²⁶, andererseits auf EPIC-Prozessoren (*explicitly parallel instruction computing*) [86, 89]¹²⁷ studiert. Sämtliche verwendete Architekturen unterstützen dabei auf Befehlssatzebene fortgeschrittene kombinierte *bit*-Operationen [52]; im Besonderen sind jedoch in den RISC- und EPIC-Entwürfen viele Instruktionen auf Transistorebene in den *Arithmetisch-logischen Einheiten* festverschaltet realisiert, so daß eine Auflösung in elementarere Anweisungen unnötig wird und zur Verarbeitung weniger Taktzyklen – zumeist sogar nur einer – gebraucht werden. Vornehmlich auf Abschnitt 3.4.4 rückblickend wird damit idealerweise den Anforderungen des zentralen Programmteils für den Elektronentransfer entsprochen, welcher, integriert in den Programmkern, Abschnitt 3.4.13, einen Hauptteil der gesamten Rechenzeit benötigt.

In dieser Untersuchung hat sich, auf Maschinen mit dem Betriebssystem HP-UX [85, 86] unter Zuhilfenahme von Profilern, zunächst gezeigt, daß die EPIC-Architektur die

¹²⁴Siehe Fußnoten 80 und 83

¹²⁵Hewlett-Packard Precision Architecture (Version 2.0)

¹²⁶Effektiv enthalten moderne CISC-Mikroprozessoren der x86-Familien verschiedener Hersteller mit 64 Bit-Registern und vergrößertem Adreßraum, hier insbesondere der verwendete Typ Intel Xeon (IA-32), viele Merkmale der RISC-Architektur.

¹²⁷Intel Itanium (IA-64) [46]

anderen Superskalarsysteme hinsichtlich Nutzungseffizienz der Recheneinheiten übertrifft: Offenbar ist die konzeptbedingte Optimierung der Anweisungsabfolge zur Übersetzungszeit durch die „Kompilerlogik“ merklich wirkungsvoller, als jene zur Laufzeit durch die „Prozessorlogik“ durchgeführte. Gegensätzlich bestätigt sich aber, daß Fehlzugriffe auf die Pufferinstanzen zwischen den Registern und dem Arbeitsspeicher (*cache*) beim Nachladen von Daten (*data-cache misses*) stärker verzögernd wirken, da hier vermutlich die ausgeprägtere Fähigkeit zur Umorganisation der Anweisungsabfolge während der Programmausführung durch die RISC- und CISC-Prozessoren eingreift. Diese *cache misses* treten in der Neuimplementierung allerdings durch eine blockbildende Indizierung innerhalb der Datenstrukturen nur noch im nicht leistungskritischen Maße auf, wodurch die EPIC-Systeme insgesamt im Vorteil sind¹²⁸.

Überdies zeigt sich die Wirksamkeit der doppelten Auslegung der Registersätze auf den genutzten CISC-Systemen, welche das „gefahrlose¹²⁹ Überladen“ eines Prozessorkernes mit zwei *threads* erlaubt¹³⁰: Der beispielsweise während Arbeitsspeicherzugriffen wartenden Prozeßfaden kann die noch ungenutzten Ausführungseinheiten des Kernes einem anderen Faden überlassen, bis dieser die Kontrolle, etwa durch eigene Wartezyklen bedingt, wieder an den ersten zurückgibt. Im hiesigen Falle beträgt die Laufzeitverkürzung durch diese Technik beachtenswerte 10%.¹³¹

5.2. Weitergehende Optimierungen. Die im folgenden Abschnitt dargelegten weiteren Verbesserungen am Gesamtalgorithmus sind zwar nicht für ein Verständnis des prinzipiellen Programmablaufs, wie dieser in Kapitel 3 vorgestellt wurde, notwendig, wohl aber technische Voraussetzung für die in Kapitel 4 umfassend behandelten Ergebnisse der Störungsrechnungsordnungen U^{-13} und U^{-15} . Daher sind die hier skizzierten Detailänderungen bereits in der konkreten Neuimplementierung und somit insbesondere

¹²⁸Zudem verfügen die jeweils acht Kerne der Itanium 9560-Prozessoren des HP AH353A (2,53 GHz Taktfrequenz, also circa 0,395 Nanosekunden Zyklusdauer) über einen gemeinsamen schnellen Zwischenspeicher (*level-III cache*) von $4 \cdot 2^{20}$ Worten (entspricht 32 MBytes; nominal 50 Taktzyklen Latenz); alle beispielsweise für die Ausgangszustandssequenzierung (siehe Abschnitte 3.4.9 und 3.4.13) wesentlichen Datenstrukturen passen jedoch in den mit nur 8 Taktzyklen Wartezeit schneller „erreichbaren“ *level-II data-cache* von $32 \cdot 2^{10}$ Worten pro Kern. (Pro Kern besitzt dieser Mikroprozessor darüberhinaus $64 \cdot 2^{10}$ Worte *level-II cache* mit 9 Taktzyklen Latenz für Befehle; der *level-I cache*, jeweils $2 \cdot 2^{10}$ Worte für Daten und Instruktionen, ist von den Registern beim Lesen nur „einen Taktzyklus weit entfernt“.) [54]

¹²⁹Beim Wechsel zwischen zwei Rechenprozessen muß typischerweise der jeweils zugehörige Inhalt sämtlicher Prozessorregister zeitaufwendig in andere Speicherinstanzen ausgelagert beziehungsweise aus diesen zurückgesichert werden.

¹³⁰Intel HyperThreading. Diese Technologie bieten auch die Itanium-Prozessoren des HP AH353A, welche architekturbedingt jedoch nicht bei allen Kontextwechseln innerhalb eines *multi threading* nutzenden Programmes vorteilhaft ist. Um mögliche Leistungseinbußen zu verhindern, wurden die Produktionsrechnungen daher mit unter HP-UX ausgeschaltetem LCPU-Systemkernparameter durchgeführt.

¹³¹Eine geringfügige Leistungsverbesserung hat sich bei einem Prozeßfaden-Kern-Verhältnis von 2 : 1 auch auf den älteren RISC- [85] und EPIC-Systemen [89] gezeigt, obwohl nur ein Registersatz zur Verfügung steht. Möglicherweise werden beim Kontextwechsel dessen Inhalte (siehe Fußnote 129) nicht in den Arbeitsspeicher, sondern in eine prozessornähere Pufferinstanz geschrieben, so daß im Mittel ebenfalls eine Laufzeitverkürzung eintritt.

in der Analyse des Rechenzeit- und Speicherbedarfs im vorausgegangenen Abschnitt berücksichtigt.

5.2.1. *Ganzzahlbehandlung der Energiebeiträge.* Die insbesondere für die Berechnung des Sequenzbeitrages zur Grundzustandsenergie von Gleichung (3.100) und folgender erforderte rechentechnische Behandlung rationaler Zahlen ist schwerfällig, da solche einschließlich ihrer durch Erweitern und Kürzen aufwendigen Arithmetik nicht als Datentypen in Prozessoren implementiert sind. Selbst für die Numerik optimierte Programmiersprachen wie Fortran stellen in ihrem standardisierten Umfang rationale Zahlen nicht bereit; folglich bleibt nur der Rückgriff auf Bibliotheksfunktionen¹³², welche etwa den (vorzeichenbehafteten) Zähler und den Nenner in zwei getrennten ganzen Zahlen speichern und entsprechend zusammengesetzte Rechenoperationen bereitstellen. Die Abbildung dieser – einschließlich einer Variante des Euklidischen Algorithmus’ zur Bestimmung des größten gemeinsamen Teilers (ggT) und anderer Sonderfunktionen – in elementare Maschinenbefehle führt zu jedoch zu Routinen mit vergleichsweise immensen Rechenzeitbedarf.

Naheliegender ist daher der Ansatz, einen Hauptnenner z_n für die gesamte n te Störungsrechnungsordnung zu finden und nur noch die entsprechend erweiterten Zähler der Hüpfsequenzbeiträge als Ganzzahlen zu verarbeiten und zu speichern. Aufgrund des hohen Geschwindigkeitsvorteiles kann dabei auch in Kauf genommen werden, daß der Wertebereich der längstmöglichen Ganzzahlen mit Vorzeichen auf *hardware*-Ebene, typischerweise von der Breite der Prozessorregister (64 Bit: $[-1 \cdot 2^{63}, 2^{63} - 1]$), bereits in Ordnungen mit realistischem Gesamtaufwand überschritten (U^{-13} und U^{-15}) und eine wiederum auf *software*-Ebene implementierte Behandlung größerer Zahlen vonnöten wird.

Ausgehend von Gleichung (4.1) gilt für $i = 1, 2, \dots, i'$ und $h = 1, 2, \dots, h'$ aller Hüpfprozeß- beziehungsweise Hüpfsequenzklassen $g = 1, 2, \dots, g'$ in n ter Störungsrechnungsordnung mit $i' = |\hat{\rho}_{g-1}^{(n)}|$, $h' = |\hat{\rho}_{g-1}^{(n)}|$ und $g' = 2^{(n-1)/2}$:

$$\mathbb{N} \setminus \{0\} \ni \begin{bmatrix} p_i \\ d_h \end{bmatrix}^{-1} = \prod_{j=2}^{n+1} d_j^{\kappa_j} \Leftrightarrow d_j \in \mathbb{N} \setminus \{0\} \wedge \kappa_j \in \mathbb{N}, \quad (5.4)$$

worin $d_j = \llbracket \hat{\rho}_{g-1}^{(n)} \rrbracket_{h,2,j}$ und $\kappa_j = \llbracket \hat{\rho}_{g-1}^{(n)} \rrbracket_{i,2,j}$. Da also der Kehrwert des (hier noch nicht mit dem Prozeßkoeffizienten e_i skalierten) Sequenzbeitrages natürlich ist, können wir das kleinste gemeinsame Vielfache (kgV) zunächst folgendermaßen ansetzen¹³³:

$$z_n = \text{kgV} \left(\left[\begin{bmatrix} p_i \\ d_h \end{bmatrix}^{-1} \mid i = 1, \dots, i' \mid h = 1, \dots, h' \mid g = 1, \dots, g' \right) \right). \quad (5.5)$$

¹³²Die bereits in Abschnitt 5.1.3 erwähnte GMP-Bibliothek [82] stellt beispielsweise einen rationalen Datentyp „beliebig mitwachsender“ Genauigkeit sowie die zugehörige Arithmetik bereit, siehe auch Fußnote 123.

¹³³Die „Operation“ kgV wirkt, wie auch ggT, im hiesigen Gebrauch auf ein *Tupel* aus zwei oder mehr natürlichen Zahlen größer null; genau genommen genügt jedoch eine Festlegung als *Mengenoperation*, da die Reihenfolge der Elemente belanglos ist und deren möglicherweise mehrfaches Auftreten das Ergebnis nicht ändert.

n	k	d'	z_n	m_n	G.	D.	aif(z_n)
3	3	1	1	1	$1 \cdot 10^0$	1	1,02120
5	4	2	2^2	1	$4 \cdot 10^0$	3	2,66056
7	5	2	2^5	1	$3,2 \cdot 10^1$	6	4,18344
9	6	3	$2^7 \cdot 3^3$	1	$3,456 \cdot 10^3$	13	6,80878
11	7	3	$2^9 \cdot 3^7$	1	$1,11974 \cdot 10^6$	22	9,49297
13	8	4	$2^{11} \cdot 3^9$	2^4	$4,03108 \cdot 10^7$	27	11,0025
15	9	4	$2^{20} \cdot 3^{11}$	2^2	$1,85752 \cdot 10^{11}$	39	14,2808
17	10	5	$2^{24} \cdot 3^{13} \cdot 5^5$	2^2	$8,35884 \cdot 10^{16}$	58	18,8728
19	11	5	$2^{28} \cdot 3^{15} \cdot 5^{11}$	2^2	$1,88074 \cdot 10^{23}$	79	77,3156
21	12	6	$2^{32} \cdot 3^{17} \cdot 5^{13}$	2^2	$6,77066 \cdot 10^{26}$	91	26,1575
23	13	6	$2^{36} \cdot 3^{19} \cdot 5^{15}$	2^2	$2,43744 \cdot 10^{30}$	102	28,6181
25	14	7	$2^{40} \cdot 3^{21} \cdot 5^{17} \cdot 7^7$	2^2	$7,22641 \cdot 10^{39}$	134	34,8994
27	15	7	$2^{44} \cdot 3^{23} \cdot 5^{19} \cdot 7^{15}$	2^2	$1,49972 \cdot 10^{50}$	168	41,4019
29	16	8	$2^{48} \cdot 3^{25} \cdot 5^{21} \cdot 7^{17}$	2^{10}	$2,64550 \cdot 10^{55}$	186	44,6041

Tabelle 5.4. Mit Mathematica nach Ausdruck (5.8) berechneter Hauptnenner z_n bis einschließlich Störungsordnung $n = 29$ in den zahlentheoretisch bedeutsamen Primfaktoren (vergleiche Tabelle 4.1). In Vorgriff auf Tabelle 5.6 (Symbolerläuterungen siehe dort) mit dem genäherten Wert \tilde{z}_n ist zudem mit $m_n = \tilde{z}_n/z_n$ das Verhältnis zwischen exaktem Wert und Approximation angegeben.

Diese Herangehensweise setzt allerdings alle Sequenzen als bekannt und somit die Störungsrechnung als durchgeführt voraus. Um a priori den Wert von z_n berechnen zu können, sind einige Vorüberlegungen notwendig, für welche wir von der Störungsrechnungsordnung n in die Graphenordnung $k = (n + 3)/2$ mit der durch $d' = \lfloor k/2 \rfloor$ gegebenen maximalen Doppelbesetzungsanzahl wechseln. Zunächst ist dabei die Frage zu behandeln, ob die Doppelbesetzungszahl d_j oder der \hat{S} -Exponent κ_j des j ten Hüpfes, welche ja voneinander aufgrund der Partitionierungsbedingung aus Abschnitt 1.4.1, siehe Gleichung (1.11), abhängig sind, das Produkt in Gleichung (5.4) maximieren; dazu wählen wir den einfachsten nicht-trivialen Fall mit $d' > 1$.

Wenn in *gerader Baumordnung* höchstmögliche Anregung vorliegt, sind bei Gitterhalbfüllung von den k Ecken des Butcherbaumes $k/2$ doppelt- und $k/2$ unbesetzt. Ein

weiterer Hüpfschritt muß demnach eine Doppelbesetzung in zwei Halbbesetzungen auflösen; folglich kann d' im Sequenzverlauf nicht „durchgehend“ über $2k - 1 - 2d'$ Zwischenzustände erhalten bleiben. Ohne und mit einer Wiederherstellung des Grundzustandes zeigt sich exemplarisch für $k = 4$

$$\underbrace{\begin{bmatrix} \text{Grid} \\ \text{Grid} \end{bmatrix}^{-1}}_{=4} = \underbrace{\begin{bmatrix} \text{Grid} \\ \text{Grid} \end{bmatrix}^{-1}}_{=4} .$$

Bei höchstmöglicher Anregung in *ungerader Baumordnung* sind bei Gitterhalbfüllung von den k Ecken des Butcherbaumes $(k - 1)/2$ doppelt-, eine Ecke halb und $(k - 1)/2$ Ecken unbesetzt. Ein weiterer Hüpfschritt löst zwar eine Doppelbesetzung – sofern Sprungstart – auf, kann aber sogleich auf dem halbbesetzten Gitterplatz – sofern Sprungziel – eine neue erzeugen, und daher kann d' im Sequenzverlauf über $2k - 1 - 2d'$ Zwischenzustände erhalten bleiben. Ohne, mit einer und mit zwei Wiederherstellungen des Grundzustandes findet man beispielhaft für $k = 5$

$$\underbrace{\begin{bmatrix} \text{Grid} \\ \text{Grid} \end{bmatrix}^{-1}}_{=32} > \underbrace{\begin{bmatrix} \text{Grid} \\ \text{Grid} \end{bmatrix}^{-1}}_{=16} > \underbrace{\begin{bmatrix} \text{Grid} \\ \text{Grid} \end{bmatrix}^{-1}}_{=8} = \underbrace{\begin{bmatrix} \text{Grid} \\ \text{Grid} \end{bmatrix}^{-1}}_{=8} .$$

Dies erlaubt folgenden Schluß, der auf Grundlage der „inneren Systematik“ der zugrundeliegenden Butcherbäume und Hüpfprozesse induktiv für beliebige Ordnungen gilt: Jede Grundzustandswiederherstellung reduziert die Potenzen der Doppelbesetzungszahlen $d_j \geq 2$ mindestens genauso stark, wie die Potenzen durch die Exponenten $\kappa_j \geq 2$ erhöht werden. Daher führt immer der eine Prozeß der Klasse $g = 1$ zum höchsten Wert des Nenners, für welche zwar nur $\kappa_j = 1$ für $j = 2, 3, \dots, 2k - 2$, aber auch $d_j \geq 2$ für $j = 3, 4, \dots, 2k - 3$ gilt. Gleichung (5.5) läßt sich somit zunächst zu

$$\begin{aligned} z_n &= \text{kgV} \left(\left[\begin{array}{c} \mathfrak{p}_i \\ \mathfrak{d}_h \end{array} \right]^{-1} \mid i = 1, \dots, i' \mid h = 1, \dots, h' \right) \Big|_{g=1} \\ &= \text{kgV} \left(\left[\begin{array}{c} \mathfrak{p}_1 \\ \mathfrak{d}_h \end{array} \right]^{-1} \mid h = 1, \dots, |\mathfrak{H}_0^{(n)}| \right) \quad \text{da} \quad i' = |\hat{\mathfrak{P}}_{g-1}^{(n)}|_{g=1} = 1 \end{aligned} \quad (5.6)$$

vereinfachen, worin allgemein

$$\mathfrak{p}_1 = (0) \circ (1 \mid u = 1, 2, \dots, 2k - 3) \circ (0) .$$

Mit den vorausgegangenen Betrachtungen kann auch die Anzahl der relevanten Doppelbesetzungstupel d weiter eingegrenzt werden, nämlich auf solche, in welchen der Doppelbesetzungswert $d = 2, 3, \dots, d'$ möglichst häufig auftritt. Mit

$$d_d = (0) \circ (1) \circ \dots \circ \overbrace{(d)}^{(2k-1-2d)\text{mal}} \circ \dots \circ (1) \circ (0) \quad \text{und} \quad (5.7-1)$$

$$d_{d'} = (0) \circ (1) \circ \dots \circ \underbrace{(d', d' - 1)}_{(k-1-d')\text{mal}} \circ (d') \circ \dots \circ (1) \circ (0) \quad (5.7-2)$$

ist dabei in der zusammenfassenden Formel¹³⁴, die erst für $k > 3$ gilt¹³⁵, zwischen geraden und ungeraden Baumordnungen zu unterscheiden:

$$z_n = \begin{cases} \text{kgV} \left(\begin{bmatrix} p_1 \\ d_{d'} \end{bmatrix}^{-1}, \begin{bmatrix} p_1 \\ d_d \end{bmatrix}^{-1} \mid d = 2, \dots, d' - 1 \right) & \text{falls } k \text{ gerade} \\ \text{kgV} \left(\begin{bmatrix} p_1 \\ d_d \end{bmatrix}^{-1} \mid d = 2, \dots, d' \right) & \text{falls } k \text{ ungerade} \end{cases} \quad (5.8)$$

Die nun vorzunehmende Primfaktorzerlegung der darin verbleibenden reziproken Hüpfsequenzbeiträge läßt sich, um den Verfahrensaufwand nicht weiter zu erhöhen, durch die Näherung $\tilde{z}_n = m_n z_n$ mit $m_n \in \mathbb{N} \setminus \{0\}$ umgehen. Eine einfache Approximation führt abschließend auf den für $k \geq 2$ gültigen Ausdruck

$$\tilde{z}_n = \begin{cases} \mathcal{M}_{d'}^{d'} \prod_{j=2}^{d'-1} \mathcal{M}_j^{2k-1-2j} & \text{falls } k \text{ gerade, mit } d' = k/2 \\ \prod_{j=2}^{d'} \mathcal{M}_j^{2k-1-2j} & \text{falls } k \text{ ungerade, mit } d' = (k-1)/2 \end{cases}, \quad (5.9)$$

worin $\mathcal{M}_i = \exp(\Lambda(i))$ die aus der Mangoldtschen Funktion $\Lambda(k)$ folgende i te gleichnamige Zahl ist [24:A014963], siehe Tabelle 5.5. Auf diese zahlentheoretisch bedeutsame

¹³⁴Die Umschrift der darin verwendeten Klammernotation in Produkte und anschließendes Umformen ergibt

$$\begin{bmatrix} p_1 \\ d_d \end{bmatrix}^{-1} = d^{2k-1-2d} ((d-1)!)^2 \quad \text{und} \quad \begin{bmatrix} p_1 \\ d_{d'} \end{bmatrix}^{-1} = 2^{-k+1} k((k-2)k)^{\frac{k}{2}-1} \left(\left(\frac{k}{2} - 1 \right) ! \right)^2 .$$

¹³⁵In der Betrachtung kann der triviale Fall $d' = 1 = d$ für $k = 3$ berücksichtigt werden; formal notiert ändert sich Gleichung (5.7-1) dann zu

$$d_d = (0) \circ \dots \circ \overbrace{(1)}^{3\text{mal}} \circ \dots \circ (0) .$$

Sequenz, nur aus Einsen und Primzahlen bestehend, wird im Abschnitt 2.4 des Anhanges kurz eingegangen und ein einfacher Algorithmus zur Erzeugung angegeben.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
\mathcal{M}_i	1	2	3	2	5	1	7	2	3	1	11	1	13	1	1	2	17

Tabelle 5.5. Die Folge der Mangoldtzahlen \mathcal{M}_i , eingetragen in der OEIS-Datenbank [24] als Sequenz A014963, bis einschließlich $i = 17$.

Der in Abschnitt 3.5 vorgestellte Algorithmus zur Berechnung des Sequenzbeitrages zum Reihenkoeffizienten kann nun unter Übernahme der ersten vier und der letzten drei Schritte für die Ganzzahlbehandlung wie folgt modifiziert werden:

- 5 Summationsschleife über alle Prozesse der Klasse f , Index $i = 1, 2, \dots, |\hat{\mathcal{P}}_f^{(n)}|$, mit $e_i = \llbracket \mathcal{P}^{(n)} \rrbracket_{i,1}$:

$$a'_n := a'_n + \tilde{z}_n e_i \prod_{j=2}^{2k-2} \frac{1}{d_j^{\kappa_j}} = a'_n + \tilde{z}_n e_i \begin{bmatrix} p_i \\ d \end{bmatrix}, \quad \text{worin} \quad (5.10)$$

$$d_j = \llbracket d \rrbracket_j = \llbracket \mathcal{H}^{(n)} \rrbracket_{2,j} \quad \text{und} \quad \kappa_j = \llbracket p_i \rrbracket_j = \llbracket p \rrbracket_{i,j} = \llbracket \mathcal{P}^{(n)} \rrbracket_{i,2,j}, \quad (5.11)$$

siehe Flußdiagrammelement (4)

- 6 Berechnung des Sequenzvorzeichens mit $s = \llbracket \mathcal{H}^{(n)} \rrbracket_1 = (-1)^b$

- 7 Berücksichtigung des Vorzeichens

$$a'_n := s a'_n .$$

Der Vorzeichenwechsel ist auf Gleichung (3.56) zurückzuführen.

Entsprechend ändert sich auch Gleichung (4.1) zu

$$a_n = r \sum_{g=1}^{2^{(n-1)/2}} \sum_{h=1}^{|\mathcal{H}_{g-1}^{(n)}|} s_h \sum_{i=1}^{|\hat{\mathcal{P}}_{g-1}^{(n)}|} \tilde{z}_n e_i \begin{bmatrix} p_i \\ d_h \end{bmatrix}; \quad (5.12)$$

darin treten nach Erweiterung der Sequenzbeiträge nur noch ganze Zahlen auf. Daß der Näherungswert \tilde{z}_n , siehe Tabelle 5.6, um den Faktor m_n über dem „wahren“, also kleinstmöglichen Nenner z_n liegt¹³⁶, vergleiche Tabelle 5.4, hat auf die rechenmaschinelle Umsetzung, vor allem die Verarbeitungsgeschwindigkeit keine Auswirkung: Die Ganzzahlarithmetik verlangt, angefangen mit der Störungsordnung U^{-15} , ohnehin eine Datenwortbreite, die über der üblichen Registerbreite von 64 Dualstellen liegt.

¹³⁶Zwar gilt im Allgemeinen nicht $\text{mod}(m_{n+1}, m_n) = 0$, aber die Eigenschaft $\text{mod}(z_{n+1}, z_n) = 0$ bleibt mit $\text{mod}(\tilde{z}_{n+1}, \tilde{z}_n) = 0$ erhalten.

n	k	d'	\tilde{z}_n	G.	D.	aif(\tilde{z}_n)
1	2	1	1	10^0	1	1,02120
3	3	1	1	10^0	1	1,02120
5	4	2	4	10^0	3	2,66056
7	5	2	32	10^1	6	4,18344
9	6	3	3456	10^3	13	6,80878
11	7	3	1119744	10^6	22	9,49297
13	8	4	644972544	10^8	31	12,1163
15	9	4	743008370688	10^{11}	41	14,7923
17	10	5	334353766809600000	10^{17}	60	19,3387
19	11	5	752295975321600000000000	10^{23}	81	24,0597
21	12	6	2708265511157760000000000000	10^{27}	93	26,5787

Tabelle 5.6. Genäherter Hauptnenner \tilde{z}_n nach Gleichung (5.9) bis einschließlich Störungsordnung $n = 21$. Orientierungshalber ist zusätzlich dessen Größenordnung (G.) vermerkt; überdies enthält die sechste Spalte die Anzahl der zur Speicherung nötigen Dualstellen (D.: $1 + \lceil \log_2(\tilde{z}_n) \rceil > 64$ für $n \geq 19$) und die siebente Spalte den Funktionswert der numerisch-approximativen inversen Fakultät, siehe Abschnitt 5.1.1, die das Anwachsen von \tilde{z}_n verdeutlicht ($\tilde{z}_n > n!$ für $n \geq 17$). Der nach Gleichung (5.8) berechnete „wahre“ Hauptnenner z_n ist in Tabelle 5.4 verzeichnet; die Primfaktoren von \tilde{z}_n ergeben sich aus den dortigen Spalten z_n und m_n .

Neben den „globalen“ Vorfaktoren aus Gleichung (3.101), die allen Einzelbeiträgen gemeinsamen sind, muß nun der genäherter Hauptnenner \tilde{z}_n im Ergebnis der gesamten Störungsrechnung, siehe Gleichung (3.1), berücksichtigt werden:

$$\begin{aligned}
 E_0(U) &= \sum_{k=2}^L \frac{1}{U^{2k-3}} E_0^{(2k-3)} + \mathcal{O}\left(\frac{1}{U^{2L-2}}\right) \\
 &= -\frac{1}{2^2} \frac{1}{2} \frac{a_1}{\tilde{z}_1} U^{-1} - \frac{1}{2^3} \frac{1}{4} \frac{a_3}{\tilde{z}_3} U^{-3} + \dots \\
 &\quad \dots - \frac{1}{2^L} \frac{1}{2L-2} \frac{a_{2L-3}}{\tilde{z}_{2L-3}} U^{-2L+3} + \mathcal{O}(U^{-2L+2}). \quad (5.13)
 \end{aligned}$$

Praktisch geschieht dies im Hauptprogramm, siehe Abbildung 3.1, zwischen den Elementen (3) und (4).

5.2.2. *Feinere Auftragsgranularisierung; Systemlastprofile.* Ein für das Laufzeitverhalten in höheren Ordnungen stark nachteiliger Effekt besteht in der Diskrepanz zwischen

dem exponentiellen Anwachsen der Ausgangszustandsanzahl und dem fakultativen Ansteigen der Rechenzeit pro Ausgangszustand, ausführlich behandelt Abschnitt 5.1.1. Die in Abbildung 5.3 (mit roter Linienfarbe) dargestellten Meßdaten für die in Kapitel 3 beschriebene Version I der Neuimplementierung verdeutlicht dieses durch ein „Leerlaufen“ des Parallelrechners.

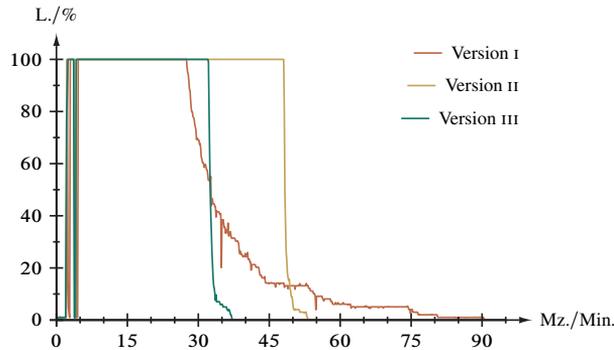


Abbildung 5.3. Systemlastprofil für die Berechnung des Reihenkoeffizienten der Ordnung U^{-13} des Hubbardmodelles. Aufgetragen ist die Auslastung eines EPIC-Hochleistungsrechners vom Typ SGI 4700 [89] („überladene“ Vollast: 1020 Prozeßfäden, zwei pro Kern) gegen die Realzeit. Die Lastschwankungen in den ersten fünf Minuten sind auf serielle Verarbeitungsstufen zurückzuführen. – Programmversion I: 64 Bit-Bruchzahlarithmetik, Verteilung der Ausgangszustände; Version II und III, letztere ohne Prüfausgaben: 128 Bit-Ganzzahlen mit Hauptnenner, Verteilung der Permutationsbaumäste.

Ursächlich für dieses Problem datentechnischer Art ist die Vorgehensweise des in Abschnitt 3.4.9 beschriebenen Rahmenalgorithmus'. Die im Programmschritt 4 beginnende Schleife arbeitet für einen jeden Butcherbaum $\mathcal{B}_{k,l,m} \in \mathcal{B}'_{k,l}$, also einen jeden Ausgangszustand, *parallel* sämtliche Meta-Permutationsbäume aus ${}^m\mathcal{P}_{2k-3}$ ab, deren Anzahl $|{}^m\mathcal{E}|$ beträgt; die damit verbundene Laufzeit eines jeden Rechenauftrags ist folglich von der Ordnung $(2k-3)!$. Für den Grundzustandsenergiebeitrag dreizehnter Ordnung, also $k=8$, kann sich der Zeitbedarf für *einen* der $||\mathcal{B}'_k|| = \sum_{l=1}^{\lambda_8} |\mathcal{B}'_{8,l}| = 6321$ Ausgangszustände bereits auf bis zu 12 Minuten von insgesamt rund 225 Stunden Prozessorzeit (beziehungsweise Realzeit bei serieller Verarbeitung) belaufen¹³⁷.

Die zu Skizze 5.1 gehörige Tabelle 5.7 zeigt nun beispielhaft das Programmverhalten für den „langsamsten Baum“ mit acht Gitterplätzen: Bearbeiten drei Rechenprozeßfäden (*threads*) auf einem Modellsystem mit drei Prozessoren (beziehungsweise Prozessorkernen) parallel die drei auftretenden Permutationsbäume ${}^m\mathcal{P}_{13,1}$, ${}^m\mathcal{P}_{13,2}$ und ${}^m\mathcal{P}_{13,4}$, so

¹³⁷Diese und folgende Angaben beziehen sich auf die Belastung eines der 496 Kerne des CISC-computeservers Typ SGI UV1000 [88].

Sequenzinitialisierungskante $r^{(1)}$	1	2	4	Summe
Laufzeit in Sekunden	722,7	638,4	730,5	2091,6
ausfallende Kanten	{2, 4}	{1, 4}	{1, 2}	
zu prüfende Permutationen	11!	11!	11!	

Tabelle 5.7. Für die Generierung sämtlicher Hüpfsequenzen des Butcherbaumes $\mathcal{B}_{8,95,26215}$ benötigte Zeit (Prozessorzeit beziehungsweise Realzeit bei serieller Verarbeitung), siehe Skizze 5.1. Die Anzahl der Initialisierungskanten und somit der Permutationsbäume ${}^m\mathcal{P}_{13,r^{(1)}}$ begrenzt die parallele Verarbeitung mit dem ursprünglichen Algorithmus auf $|{}^m\ddot{E}| = |{}^m\mathcal{P}'_{13}| = 3$ Prozessfäden.

befindet sich nach circa 638 Sekunden jener sämtliche Permutationen ab Sequenzinitialisierungskante $r^{(1)} = 2$ abarbeitende Prozessor im Leerlauf. Damit sinkt die Auslastung der gesamten Rechenanlage bis zum Abschluß des „nächstschnelleren“ Permutationsbaumes (mit $r^{(1)} = 1$) für rund 84 Sekunden auf zwei Drittel und sodann bis zur vollständigen Sequenzierung des gesamten Ausgangszustandes für weitere rund 8 Sekunden auf ein Drittel.

$$\mathcal{B}_{8,95,26215} = \begin{array}{c} \circ \\ \circ \end{array} \text{ mit } {}^m\ddot{E} = \{1, 2, 4\} \quad (\text{Skizze 5.1})$$

Zwar tritt unter realen Bedingungen der oben beschriebene gering erscheinende Fall nicht ein, da der Auftragsverteiler (*dispatcher*) unmittelbar dem abgeschlossenen Auftrag einen neuen, hier etwa den ersten Permutationsbaum ${}^m\mathcal{P}_{13,1}$ des nächsten Ausgangszustandes $\mathcal{B}_{8,95,26218}$, folgen läßt. Dennoch ergibt sich besagter Leerlaufeffekt zwangsläufig, sobald die Restanzahl der insgesamt 8122 Permutationsbäume für alle 6321 relevanten Butcherbäume, siehe dazu Tabelle 5.9, unter die bereitgestellte Prozessor(kern)anzahl sinkt. Entsprechend verstärkt sich diese mangelhafte Systemauslastung mit zunehmendem Parallelisierungsgrad der Verarbeitung und zunehmender Störungsordnung beziehungsweise Laufzeit der Permutationsbäume; neben Abbildung 5.3 dokumentiert auch Abbildung 5.4 dieses Verhalten durch die Gegenüberstellung zweier Auslastungsprofile (in blauer und roter Linienfarbe) einer kleineren Rechenanlage für verschiedene Entwicklungsstufen der Implementierung.

Zur praktischen Minderung dieses Effekts kann der entsprechende Teil des Graphenformalismus', vergleiche Abschnitt 3.4.5, folgendermaßen erweitert werden:

$${}^m\mathcal{P}'_{2k-4;\{r^{(1)}\}}: \text{ („Permutationsbaumzerlegung“) Geordnete Menge aller Hauptäste des Meta-Permutationsbaumes } {}^m\mathcal{P}_{2k-3,r^{(1)}} \in {}^m\mathcal{P}_{2k-3}, \text{ welche für einen gegebenen}$$

Ausgangszustand $\mathcal{B}_{k,l,m} \in B'_{k,l}$ nach Ausführung aller (unterscheidbaren) Sequenzinitialisierungssprünge über die Kanten $r^{(1)} \in {}^m\ddot{E}$ verbleiben und somit ihrerseits wieder Meta-Permutationsbäume der nun reduzierten Meta-Kantenmenge ${}^mG \setminus \{r^{(1)}\}$, schärfer formuliert ${}^mG \setminus {}^m\ddot{E}$, da der zweite Sprung nicht ebenfalls über eine Initialisierungskante erfolgen kann, sind.

$$\begin{aligned} \forall r^{(1)} \in {}^m\ddot{E} : {}^m\mathcal{P}'_{2k-4;\{r^{(1)}\}} &= \{ {}^m\mathcal{P}'_{2k-4,r^{(2)};\{r^{(1)}\}} \mid r^{(2)} \in {}^mG \setminus {}^m\ddot{E} \\ \Leftrightarrow F(a) \neq \chi_0 \wedge F(b) \neq \chi_3 \wedge F(b) \neq F(a) \mid a &= \llbracket G_{r^{(2)}} \rrbracket_1, b = \llbracket G_{r^{(2)}} \rrbracket_2 \} \quad (5.14) \end{aligned}$$

Darin ist $r^{(2)}$ die Kante des ersten Sprunges nach der „Sequenzöffnung“.

$T'_{k,l,m}$: („Ausgangszustandszerlegung“) Geordnete Menge aller (viergefärbten) Butcherbäume, welche nach Ausführung aller (unterscheidbaren) Sequenzinitialisierungssprünge über die Kanten $r^{(1)} \in {}^m\ddot{E}$ für einen gegebenen Ausgangszustand $\mathcal{B}_{k,l,m} \leftarrow |1M, 2M\rangle_{k,l}$ entsteht, im folgenden *Zwischenzustände erster Art* genannt. Es gilt dementsprechend $|T'_{k,l,m}| = |{}^m\ddot{E}(\mathcal{B}_{k,l,m})|$; überdies ist $\llbracket T'_{k,l,m} \rrbracket_i \in B_{k,l}$ für $i = 1, 2, \dots, |T'_{k,l,m}|$, also Element der Menge aller viergefärbten Butcherbäume k ter Ordnung und l ter Struktur.

$$\begin{aligned} T'_{k,l,m} &= \{ \check{c}_{b,1}^+ \check{c}_{a,1} |1M, 2M\rangle_{k,l} = \check{T}_{r^{(1)},1} |1M, 2M\rangle_{k,l} \mid \forall r^{(1)} \in {}^m\ddot{E} : a = \llbracket G_{r^{(1)}} \rrbracket_1, \\ &b = \llbracket G_{r^{(1)}} \rrbracket_2 \leftarrow \llbracket 1C \rrbracket_a = 0, \llbracket 1C \rrbracket_b = 1 \leftarrow F(a) = \chi_0, F(b) = \chi_3 \} \quad (5.15) \end{aligned}$$

Man beachte, daß tatsächlich alle Zwischenzustände erster Art Butcherbäume und nicht einmal teilweise deren Isochromismen sind, da die Festlegung der Initialisierungskante mit der Färbungsordnungsrelation beim Baumaufbau korrespondiert, vergleiche Abschnitt 3.2: In Skizze 3.4 wird beispielsweise nur der erste der beiden angegebenen Zwischenzustände erzeugt, da die Sprungkanten der selben Klasse angehören, siehe auch Skizze 3.6. In Abschnitt 6.2.3 wird nochmals auf diesen Sachverhalt zurückgegriffen.

Eine Verallgemeinerung der oben formulierten Zerlegungen und vorallem die Definition der Zwischenzustände höherer Art für Sprünge $j > 2$ ist keineswegs einfach, da mit dem jeweiligen Permutationsbaum ${}^m\mathcal{P}_{2k-3,r^{(1)}}$ für die Wahl der Hüpfkante $r^{(j)}$ auch stets der Binärbaum ${}^m\mathcal{D}_{2k-3,s^{(1)}}$ für die Wahl der Hüpfspins $s^{(j)}$ verbunden ist. Im Halbfüllungsfall ist dieser Sachverhalt jedoch für den auf die Sequenzinitialisierung folgenden Sprung $j = 2$ nicht relevant, da bei Doppelbesetzungsauflösung durch Sprung über die Gegenkante keine Binärbaumverzweigung zu beachten ist, vergleiche dazu Abschnitt 3.4.15.

Folgendes einfaches Beispiel aus der Ordnung $k = 3$, das noch eine übersichtliche graphische Darstellung der Permutationsbäume erlaubt, diene der Illustration der vorausgegangenen Zerlegungsdefinitionen:

$$\begin{aligned}
 {}^m\mathcal{P}_{3,1} &= \begin{array}{c} \textcircled{3} \quad \textcircled{4} \\ \textcircled{4} \quad \textcircled{2} \\ \textcircled{2} \quad \textcircled{1} \\ \textcircled{4} \quad \textcircled{3} \\ \textcircled{2} \quad \textcircled{3} \end{array} \longrightarrow {}^m\mathcal{P}'_{2;\{1\}} = \left\{ \begin{array}{c} \textcircled{4} \\ \textcircled{2} \\ \textcircled{3} \\ \textcircled{4} \\ \textcircled{2} \end{array}, \begin{array}{c} \textcircled{3} \\ \textcircled{2} \\ \textcircled{4} \\ \textcircled{3} \\ \textcircled{2} \end{array} \right\} = \{ {}^m\mathcal{P}'_{2,3;\{1\}}, {}^m\mathcal{P}'_{2,4;\{1\}} \} \\
 \mathcal{B}_{3,2,16} = \textcircled{\textcircled{2}} \longrightarrow \mathcal{T}'_{3,2,16} = \{ \check{\mathcal{T}}'_{1,1}(\mathcal{B}_{3,2,16}) = \textcircled{\textcircled{2}} = \mathcal{B}_{3,2,7} \}
 \end{aligned}$$

Dabei ist zu berücksichtigen, daß der Meta-Permutationsbaum ${}^m\mathcal{P}'_{2,2;\{1\}}$ gemäß den Kriterien in Gleichung (5.14) nicht Element von ${}^m\mathcal{P}'_{2;\{1\}}$ ist, da ein Hüpfen von der nach dem ersten Sprunge unbesetzten Wurzel, $F(1) = \chi_0$, nicht erfolgen kann.

Bezogen auf den eingangs vorgestellten Fall aus der dreizehnten Störungsrechenungsordnung (Skizze 5.1) ergeben sich folgende Permutationsbaum- und Ausgangszustandszerlegungen:

$$\begin{aligned}
 {}^m\mathcal{P}_{13,1} &\longrightarrow {}^m\mathcal{P}'_{12;\{1\}} = \{ {}^m\mathcal{P}'_{12,3;\{1\}}, {}^m\mathcal{P}'_{12,5;\{1\}}, {}^m\mathcal{P}'_{12,6;\{1\}}, \dots, {}^m\mathcal{P}'_{12,14;\{1\}} \}, \\
 {}^m\mathcal{P}_{13,2} &\longrightarrow {}^m\mathcal{P}'_{12;\{2\}} = \{ {}^m\mathcal{P}'_{12,r^{(2)};\{2\}} \mid r^{(2)} \in \{3, 5, 6, 7, 8, 9, 10, 11, 13, 14\} \}, \\
 {}^m\mathcal{P}_{13,4} &\longrightarrow {}^m\mathcal{P}'_{12;\{4\}} = \{ {}^m\mathcal{P}'_{12,r^{(2)};\{4\}} \mid r^{(2)} \in \{3, 5, 6, 7, 8, 9, 11, 12, 13, 14\} \}
 \end{aligned}$$

und

$$\begin{aligned}
 \mathcal{B}_{8,95,26215} &\longrightarrow \mathcal{T}'_{8,95,26215} = \{ \check{\mathcal{T}}'_{r^{(1)},1}(\mathcal{B}_{8,95,26215}) \mid r^{(1)} \in {}^m\check{E} \} \\
 &= \left\{ \begin{array}{c} \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \end{array}, \begin{array}{c} \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \end{array}, \begin{array}{c} \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \\ \textcircled{\textcircled{2}} \end{array} \right\} \\
 &= \{ \mathcal{B}_{8,95,9832}, \mathcal{B}_{8,95,9847}, \mathcal{B}_{8,95,13927} \}.
 \end{aligned}$$

Der Ausgangszustand $\mathcal{B}_{8,95,26215}$ zerfällt somit in $|\mathcal{T}'_{8,95,26215}| = 3$ Zwischenzustände erster Art, welche insgesamt $|{}^m\mathcal{P}'_{12;\{1\}}| + |{}^m\mathcal{P}'_{12;\{2\}}| + |{}^m\mathcal{P}'_{12;\{4\}}| = 31$ um einen Hüpf-schritt verkürzte Rechenaufträge nach sich ziehen; Tabelle 5.8 zeigt deren Laufzeiten und belegt eindrucksvoll, daß der Gesamtrechenzeitbedarf erwartungsgemäß konstant bleibt.

Sequenzinitialisierungskante $r^{(1)}$		1	2	4	Summe
m des Zwischenzustands $\mathcal{B}_{8,95,m}$		9832	9847	13927	
erste Sprungkante $r^{(2)}$	3	24,4	23,7	24,2	72,3
(Laufzeiten in Sekunden)	5	49,3	42,6	39,6	131,5
	6	70,7	23,3	75,0	169,0
	7	76,3	19,6	34,4	130,3
	8	37,8	51,6	70,9	160,3
	9	80,1	24,9	19,1	124,1
	10	67,6	79,1	—	146,7
	11	31,0	58,6	296,6	386,2
	12	48,0	—	55,9	103,9
	13	41,3	270,5	73,4	385,2
	14	196,2	44,5	41,4	282,1
Summe		722,7	638,4	730,5	2091,6
ausfallende Kanten		{2, 4}	{1, 4, 12}	{1, 2, 10}	
zu prüfende Permutationen		11!	10!	10!	

Tabelle 5.8. Für die Generierung sämtlicher Hüpfsequenzen des Butcherbaumes $\mathcal{B}_{8,95,26215}$ benötigte Zeit (Prozessorzeit beziehungsweise Realzeit bei serieller Verarbeitung), siehe Skizze 5.1. Die feinere Auftragsgranularisierung des modifizierten Algorithmus' erlaubt eine verteilte Verarbeitung durch $|{}^m\mathcal{P}'_{12;\{1\}}| + |{}^m\mathcal{P}'_{12;\{2\}}| + |{}^m\mathcal{P}'_{12;\{4\}}| = 31$ Prozeßfäden.

Die Implementierung dieses Ansatzes für eine feinere Auftragsgranularisierung durch die explizite Permutationsbaumzerlegung, die mittelbar auch die Ausgangszustandszerlegung nach sich zieht, erfolgt geradlinig im iterativen Rahmenalgorithmus, siehe Abschnitt 3.4.9 durch Einfügen einer Schleife über alle Zwischenzustände erster Art zwischen der Initialisierungskanten- (beginnend mit Programmschritt 4.3) und der eigentlichen Permutationsschleife (ab Schritt 4.3.4). Im Detail sind folgende Modifikationen bei Übernahme der Schritte 1 bis 4.2 und 5 bis 7 notwendig:

- 4.3 Für $q = 1, 2, \dots, |{}^m\ddot{E}(\mathcal{B}_{k,l',m'})|$: Schleife über alle Sequenzinitialisierungskanten:
- 4.3.1 [unverändert] $r^{(1)} = \llbracket {}^m\ddot{E} \rrbracket_q$ initialisieren und dabei voraussetzungsgemäß $s^{(1)} = 1$ durch $\chi_j = \chi_1$ festlegen, siehe Gleichung (3.89-1)

- 4.3.2 [unverändert] Formal ${}^m\mathcal{P} := {}^m\mathcal{P}_{2k-3, r^{(1)}}$ generieren, konkret zunächst initiale Liste der Hüpfkantenindices als Meta-Kantenmenge
- $${}^m\dot{G}_{r^{(1)}} = \{r^{(1)}, 1, 2, \dots, r^{(1)} - 1, r^{(1)} + 1, \dots, 2k - 3\}$$
- anlegen, siehe Flußdiagrammelement (5) in Abbildung 3.6
- 4.3.3 Sprungnummer $j = 2$ setzen
- 4.3.4 Für $t = 0, 1, \dots, 2k - 4$: Parallelisierte¹³⁸ Schleife über alle Sprünge, die möglicherweise¹³⁹ auf die Sequenzinitialisierung folgen:
- 4.3.5 (Erzeugung des Zwischenzustandes erster Art) Initialisierungssprung, $\bullet \rightarrow \bullet$, ausführen, entsprechend Programmschritt 4 im rekursiven Kern, Abschnitt 3.4.13
- 4.3.6 (optional) Falls Sprung über $r^{(2)} = \llbracket {}^mG \setminus {}^m\dot{E} \rrbracket_t$ durch $\circ \rightarrow$ oder $\rightarrow \bullet$ oder $\bullet \rightarrow \bullet$ oder $\bullet \rightarrow \bullet$ nicht ausführbar, zu Schritt 4.3.9 übergehen¹⁴⁰
- 4.3.7 Für $u = t(2k-4)!, \dots$: Schleife über die Permutationsbaumäste ${}^m\tilde{G}_u = \llbracket \mathbb{F}({}^m\mathcal{P}) \rrbracket_{u+1}$ beziehungsweise (indirekter Weg) über alle Hüpfkantenvertauschungen, beginnend mit ${}^m\tilde{G}_u = {}^m\tilde{G}_{t(2k-4)!}$, gegebenenfalls endend mit ${}^m\tilde{G}_{t(2k-4)!(2k-4)!-1} = \{r^{(1)}, r^{(2)}, 2k-4, 2k-5, \dots, r^{(1)} + 1, r^{(1)} - 1, \dots, r^{(2)} + 1, r^{(2)} - 1, \dots, 1\}$ mit $r^{(2)} = \llbracket {}^mG \setminus {}^m\dot{E} \rrbracket_t$:
- 4.3.7.1 [entspricht dem unveränderten Programmschritt 4.3.4.1]
- 4.3.7.2 Falls *flag* im vorausgehenden Programmschritt gesetzt, zum Schleifenende springen, Schritt 4.3.8
- 4.3.7.3 [entspricht dem unveränderten Schritt 4.3.4.3]
- 4.3.7.4 [entspricht dem unveränderten Schritt 4.3.4.4]
- 4.3.8 Formal $p' = (2k - 2 - p)!$ Äste auf ${}^m\mathcal{P}$ überspringen, konkret zu $u := u + p'$ übergehen, vergleiche Element (9), und *ute* Permutation der Kantenindices in einem Unterprogramm erzeugen, dann zu Schritt 4.3.7 rückspringen, falls $u < (t+1)(2k-4)!$; sonst Schleife beenden. Im Programm wird der Schleifenabbruch durch den Rückgabewert des „Permutierers“ ausgelöst, siehe Anhang, Abschnitt 2.2, sobald beim rechts in der Kantenindexmenge beginnenden Verfahren deren erstes Element, also der Index der Initialisierungskante erreicht wird. [Weiteres siehe Schritt 4.3.5]
- 4.3.9 Falls $t < 2k - 4$, zu $t := t + 1$ übergehen, Schritt 4.3.4; sonst Schleife beenden
- 4.4 [unverändert] Falls $q < |{}^m\dot{E}(\mathcal{B}_{k, l', m'})|$, zu $q := q + 1$ übergehen, Schritt 4.3; sonst Schleife beenden

¹³⁸Siehe Fußnote 83

¹³⁹Siehe Programmschritt 4.3.6

¹⁴⁰Dieser Test bewirkt zumindest bis $k = 7$ keinen meßbaren Leistungsvorteil, da offenbar im Mittel ebenso viel Zeit für die jeweilige Prüfung von Quell- und Zielgitterplatzbesetzung benötigt wird, wie für die verbleibenden Schritte bis zum Aufruf des rekursiven Kernes (Schritt 4.3.7.4) gebraucht werden, in welchem gegebenenfalls sonst der Abbruch erfolgt.

Die Wirksamkeit der modifizierten Version belegen die Systemlastprofile in den Abbildungen 5.3 und 5.4 (jeweils gelbe und grüne Linienfarbe) deutlich.¹⁴¹

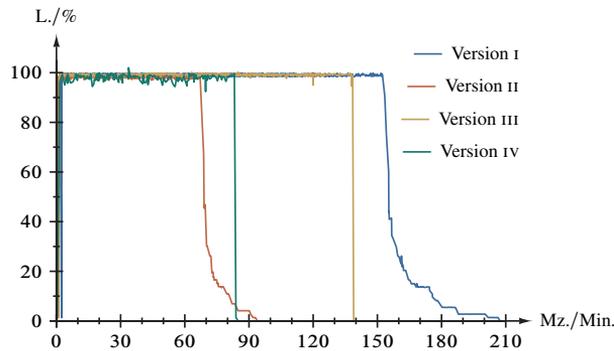


Abbildung 5.4. Systemlastprofil für die Berechnung des Reihenkoeffizienten der Ordnung U^{-13} des Hubbardmodelles. Aufgetragen ist die Auslastung eines CISC-Rechners vom Typ IBM 7233 [87] (Dreiviertellast: 72 Prozeßfäden, einer pro Kern) gegen die Realzeit. Die anfänglichen Lastschwankungen sind auf serielle Verarbeitungsstufen zurückzuführen. – Programmversion I: *multiple precision*-Bruchzahlarithmetik (GMP), Verteilung der Ausgangszustände; Version II: 64 Bit-Bruchzahlen, Verteilung der Ausgangszustände; Version III: *multiple precision*-Ganzzahlen (GMP), Verteilung der Permutationsbaumäste; Version IV: 128 Bit-Ganzzahlen mit Hauptnenner, Verteilung der Permutationsbaumäste.

Die Anzahl der vom rekursiven Sequenzierungsalgorithmus zu bearbeitenden Aufträge in Abhängigkeit von der Baumordnung k beziehungsweise der Störungsrechenungsordnung n ist in Tabelle 5.9 verzeichnet. Man berücksichtige, daß trotz wachsender Auftragsanzahl die Laufzeit pro Auftrag abnimmt (siehe obiges Beispiel für $k = 8$), nämlich im Idealfall der Ausführbarkeit aller Sprünge zwischen Stufe I und II um den Faktor $(k - 1)^{-1}$ (maximal die Hälfte aller Kanten kann der Sequenzinitialisierung dienen) und zwischen I und III entsprechend um den Faktor $(k - 1)^{-1}(2k - 3)^{-1} = (2k^2 - 5k + 3)^{-1}$.

Während der Ausarbeitung und Implementierung der insbesondere in den Abschnitten 3.4.15 und 5.2 vorgestellten Optimierungen wurden die Laufzeitverkürzungen für

¹⁴¹Man beachte, daß ein Vergleich der Rechenanlagen, auf welchen die Messungen durchgeführt wurden, untereinander aufgrund vielfältiger Unterschiede (insbesondere der Systemarchitekturen, Rechnergenerationszugehörigkeit und daher Taktzyklusdauer von Mikroprozessor und Arbeitsspeicher) nicht sinnvoll ist.

k	2	3	4	5	6	7	8	9	Stufe
$ \mathbb{B}'_k $	1	4	16	68	298	1356	6321	30642	I
$\sum_{\mathbb{B}'_k} {}^m\mathbb{P}_{2k-3} $	1	4	18	80	367	1708	8122	39729	II
$\sum_{\mathbb{B}'_k, r^{(1)}} {}^m\mathbb{P}'_{2k-4; \{r^{(1)}\}} $	1	8	54	320	1835	10248	56854	313632	III
$n = 2k - 3$ in U^{-n}	1	3	5	7	9	11	13	15	

Tabelle 5.9. Auftragsanzahl für unterschiedliche Granularisierungsstufen der Sequenzierung. – Ordnung der Laufzeitkomplexität Stufe I (ganze Ausgangszustände): $(k - 1)!$; Stufe II (Initialisierungskanten): $(2k - 3)!$; Stufe III (Hauptäste): $(2k - 4)!$.

Rechnungen hoher Ordnung mangels entsprechender Ressourcen nicht systematisch geprüft. Dennoch liegen einige Werte vor, welche die Beeinflussung des Gesamtzeitbedarfs dokumentieren (Pa. steht für die Prozessoranzahl, R. für den verwendeten Rechner (siehe Verzeichnis) und Mz. für die Maschinen- beziehungsweise Realzeit):

n	Version I			Version II		
	Pa.	R.	Mz.	Pa.	R.	Mz.
9	1	[85]	4 Sekunden			
11	1	[87]	800 Sekunden			
13	496	[88]	3000 Sekunden	368	[88]	1200 Sekunden
15				368	[88]	230 Stunden

Bei Version I handelt es sich um die Neuprogrammierung des Gesamtverfahrens gemäß Kapitel 3, welche folglich die Ausgangszustände parallel verarbeitet und mit 64 Bit-Bruchzahlen rechnet; Version II schließt die Optimierungen aus Kapitel 5 ein, verwendet also die Permutationsbaumzerlegung und 128 Bit-Ganzzahlen mit „globalem“ Hauptnenner. Unter der, wie Abschnitt 6.3.4 belegt, gerechtfertigten Annahme nahezu linearer Skalierung des Geschwindigkeitszuwachses mit der Prozessoranzahl ergibt sich für die Störungsrechnung dreizehnter Ordnung zwischen beiden Programmversionen eine Reduktion der Laufzeit auf knapp 30 %. Weitere Angaben zum Rechenzeitbedarf finden sich vornehmlich in den Abschnitten 5.1.2 und 6.3.1.

6. Zusammenfassung und Ausblick

Mit ihrer kombinatorisch-graphentheoretischen Störungsrechnung [15–17] hat EVA KALINOWSKI ein von der Kato-Takahashi-Störungstheorie ausgehendes allgemeines maschinell-analytisches Verfahren für die Untersuchung des Hubbardmodells in den Grenzfällen hoher Dimension (Bethegitter mit unendlicher Koordinationszahl) und starker Elektronenkorrelation (Mottische Isolatorphase) vorgelegt, welches *systematisch* den Reihenkoeffizienten für die Energie Ordnung für Ordnung in einer $1/U$ -Entwicklung der Kopplung *exakt* bestimmt, also keiner über den Modellrahmen hinausgehenden Näherungen bedarf. Wie ihre Studien [18, 21] mit ihrer ersten Implementierung dieser Methode, beschränkt sich auch die vorliegende Arbeit auf Grundzustandseigenschaften, erreicht indes jene hohen Ordnungen, die für die Beurteilung von Befunden numerischer Näherungslösungen, vornehmlich der Dynamischen *mean-field*-Theorie entscheidend sind – besonders die Werte für die kritische Wechselwirkung, unterhalb welcher die Mottphase instabil wird. Dies wird trotz faktorieller Laufzeitkomplexität durch einen effizienzoptimierten und parallelisierten Algorithmus des *greedy*-Typs erreicht, welcher sich zudem, über die Vorarbeiten hinausgehend, für die bereits ursprünglich angedachte [19] Anwendung auf viele weiterführende Fragestellungen eignet. Letztlich wird damit in diesem Kontext ein Maßstab-setzendes Gegenstück zur DMFT etabliert, welche sich einschließlich ihrer Verallgemeinerungen generell zum führenden Ansatz für das Studium korrelierter Fermionensysteme entwickelt hat [26].

Die Fülle der Erkenntnisse und Ergebnisse in den vorausgegangenen Kapiteln legt zur besseren Strukturierung eine Aufteilung der Zusammenfassung und der Ausblicknahme auf die beteiligten Fächer nahe, wobei – nach Natur der Sache – eine scharfe Trennung kaum erfolgen kann: Abgesehen von in ähnlicher Weise computeralgebraisch zu behandelnden künftigen Störungsreihen werfen allein viele Verallgemeinerungen dieses Rechenverfahrens Probleme auf, die nur in einer interdisziplinären Zusammenarbeit mit erwiesenermaßen allseitigem Gewinn effizient gelöst werden können. Weiterentwicklungen vorbereitend befaßt sich dieses zusammenfassende Kapitel zunächst im Schwerpunkt mit der Physik (Rückgriff auf die Kapitel 1 und 4), geht dann auf der Mathematik zugehörige Gesichtspunkte ein (Kapitel 2) und schließt mit Fragen der Informatik und Datenverarbeitungstechnik ab (Kapitel 3 und 5).

6.1. Physikalische Aspekte.

6.1.1. Derzeitiger Stand. Die vorgelegte neue Implementierung hat, einzig durch die verfügbaren, derzeit leistungsfähigsten Ressourcen beschränkt, die Berechnung der Störungsreihenoeffizienten der Grundzustandsenergie und der daraus abgeleiteten Doppelbesetzungszahl, Gleichungen (4.2) und (4.4), des Hubbardmodells bis einschließlich der Ordnung U^{-15} ermöglicht. Absichernd steht zudem für das Falicov-Kimball-Modell das Resultat der Ordnung U^{-17} , Gleichung (4.3) respektive (4.5), im Einklang mit dem analytischen Ergebnis, Gleichung (1.3). Auf dieser Grundlage konnten die numerischen Befunde für die kritische Wechselwirkung, Gleichung (4.10) beziehungsweise (4.11), und den kritischen Exponenten, Gleichung (4.13), der Quanten-Monte-Carlo-Methode

als *impurity solver* für die Gleichungen der Dynamischen *mean-field*-Theorie bestätigt werden. Entscheidend war dafür die bereits in den Referenzen [18, 19] vorgestellte extrapolierte Störungstheorie auf Basis einer Domb-Sykes-Analyse der Asymptotik der Störungsreihe, wobei sich das dort angewendete ($s = n + 1$)-Schema für die Bestimmung des singulären Verhaltens nahe U_c bewährt hat.

6.1.2. *Allgemeine Aussichten.* Bei allen Überlegungen zu weiterführenden Anwendungen der in dieser Arbeit betrachteten und gegebenenfalls zu modifizierenden kombinatorisch-graphentheoretischen Störungsrechnung ist die Einhaltung einer entscheidenden Bedingung zu beachten: Die Korrelationsfunktion

$$\langle \phi_0 | \prod_{j=1}^{2k-2} \hat{c}_{u_j, \sigma_j}^+ \hat{c}_{u_j, \sigma_j'} | \phi_0 \rangle \quad (6.1)$$

kann nur dann, wie in Abschnitten 1.3 bis 1.5 erläutert, einfach ausgewertet werden, wenn die Singulett-Grundzustandsentartung in allen Ordnungen der Reihenentwicklung erhalten bleibt¹⁴². Ist diese nicht erfüllt, kann eine exakte Berechnung der relevanten Korrelationsfunktion komplizierter oder sogar unmöglich werden. Falls kein anderer Zugang gefunden wird, lassen sich aber die Reihenkoeffizienten auf numerischem Wege approximativ bestimmen, und auch wenn die Störungsrechnung damit ihre Exaktheit einbüßt, bleibt im Gegensatz zu vielen anderen Näherungsverfahren die Approximation auf obigen Erwartungswert beschränkt, da die Methode weiterhin *alle* Prozesse und Sequenzen in der betrachteten Ordnung berücksichtigt. Da diese Entartung bei endlicher Wechselwirkung offensichtlich eine Eigenschaft des Modells ist, muß die Übertragbarkeit der Methode von Fall zu Fall kontrolliert werden.

Einen erheblichen Zuwachs an Rechenleistung vorausgesetzt, lassen sich mit der Neuprogrammierung nächstliegend prinzipiell noch *höhere Ordnungen* der Kato-Takahashi-Störungsrechnung erreichen¹⁴³. Weil Datenverarbeitungsanlagen solcher Größe zur Zeit nur als *cluster* aus *SMP-servers* zur Verfügung stehen, ist allerdings, wie in den Abschnitten 5.1.2 und 6.3.2 behandelt wird, eine „hybride“ Parallelisierung unter abermaliger Verfeinerung der Auftragsgranularität¹⁴⁴ notwendig. Damit stünde ein nochmals verbesserter Referenzmaßstab für die mit numerischen Methoden (beispielsweise DDMRG, DFT/LDA und QMC) berechneten Grundzustandseigenschaften zur Verfügung.

Ein ebenfalls in Veröffentlichung [19] vorgezeichneter Ansatz besteht in der Reduktion des Verfahrens auf die *dominierenden*, sich weit über das Gitter erstreckenden Ausgangszustände, welche aus Sicht des Metall-Isolator-Übergangs die signifikanten Beiträge liefern. Die damit verbundene Aufwandsminderung erlaubte das Erreichen weitaus

¹⁴²Siehe dazu auch Fußnote 7

¹⁴³Die Breite der *general-purpose*-Register heutiger Mikroprozessoren von 64 Bit schränkt bei Fermionen auf 15 Gitterplätze ein, sofern die genutzte Binärkodierung aller Teilchennummern in nur einer Ganzzahl beibehalten wird.

¹⁴⁴Also eine Verallgemeinerung des in Abschnitt 5.2.2 diskutierten Verfahrens

höherer Ordnungen und somit vor allem eine präzisere Bestimmung der kritischen Wechselwirkung.¹⁴⁵ Eine Kombination mit numerischen Methoden wiederum könnte zu einem (gegebenenfalls stochastischen) *Hybridverfahren* führen, dessen „Ergebnisqualität“ über jener der reinen Numerik liegt.

Mit Verfahrensmodifikationen bei der Ausgangszustandssequenzierung hinsichtlich der Beitragsgewichtung ist eine kontinuierliche *Interpolation* des Hüpfamplitudenverhältnisses zwischen \uparrow - und \downarrow -Elektronen, also dem Hubbard- ($t_{\downarrow}/t_{\uparrow} = 1$) und dem Falicov-Kimball-Modell ($t_{\downarrow}/t_{\uparrow} = 0$) möglich. Auch mit der Absicht, die Ergebnisse dieses Rechenverfahrens als Bewertungsmaßstab den Resultaten von Näherungsmethoden gegenüberzustellen, könnten über die hier behandelten Modelle hinausgehende Erweiterungen etwa die Betrachtung *mehrerer Bänder* [55] oder interne $SU(n)$ -Symmetrien [56] betreffen.

6.1.3. Beliebige Gitterfüllungen und Anregungen; niedrigdimensionale Systeme. Untersuchungen mit dieser Rechenmethode am Hubbardmodell mit *beliebiger Bandfüllung* und – noch allgemeiner – an metallischen Phasen im Starkwechselwirkungsregime steht vor allem das im vorigen Abschnitt 6.1.2 ausgeführte ernste Problem der Singulett-Grundzustandsentartung entgegen; tatsächlich ist diese bei $U = \infty$ jenseits der Halbfüllung bereits in der ersten Ordnung des Hüpfterms partiell aufgehoben. Ein Ausweg kann wiederum die numerische Bestimmung des „Erwartungswertes“ (6.1) und folglich des Reihenkoeffizienten sein. Seitens der Datentechnik ist im neuen Programm an nahezu¹⁴⁶ allen wesentlichen Stellen die Ausweitung des Verfahrens auf beliebig gefüllte und *angeregte Ausgangszustände* und Hüpfprozesse darauf berücksichtigt; eine solche wird für die Entwicklung der Greenschen Funktion, siehe Abschnitt 6.1.4, unabdingbar. Grundsätzlich ist somit auch eine Untersuchung von Systemen aus *Spin-Bosonen* möglich, vergleiche Abschnitt 6.1.5, für welche neben dem Kernalgorithmus für die Ausgangszustandssequenzierung auch das Baumerzeugungsverfahren bezüglich der Farbanzahl leicht angepaßt werden muß, siehe auch Abschnitt 3.4.1.

Zumindest als Näherungsverfahren läßt sich diese Störungsrechnung für das Hubbardmodell überdies, wenngleich mit höherem Entwicklungsaufwand, an andere, vom Bethegitter abweichende Geometrien sowie für endliche Dimensionen¹⁴⁷ und Koordinationszahlen adaptieren [11, 58]. Soweit schleifenfrei, sind *niedrigdimensionale Gitter*, insbesondere Kettenstrukturen, leicht erschließbar, da die Ausgangszustände eine Untermenge der im hiesigen Kontext erzeugten sind. Die wesentliche Änderung besteht hierbei in der Gewichtung der generierten Hüpfsequenzen. Durch eine allgemeinere, mathematisch indes entsprechend komplexere und folglich auch technisch aufwendigere Graphendefinition lassen sich *Mehrfachsprünge*¹⁴⁸ erfassen; die Modellierung *ionischer Gitter* durch

¹⁴⁵Inwieweit diese Näherung *kontrollierbar* ist, bliebe Gegenstand weiterführender Überlegungen.

¹⁴⁶Aus Leistungsgründen erfolgen, wie im Kapitel 3 dargelegt, bestimmte Variablen-, Iterations- und Rekursionsinitialisierungen in einer nur für den Grundzustand gültigen Weise.

¹⁴⁷Als approximative Methode für die Behandlung starker Elektronenkorrelation wird bereits die DMFT beispielsweise für *ab-initio-Rechnungen* in drei Dimensionen eingesetzt [25, 26, 57].

¹⁴⁸Durch Kantenmultimengen, siehe Fußnote 19

verschiedene Hüpfamplituden in Form von „Kantengewichten“¹⁴⁹ ist auf diesem Wege ebenfalls erreichbar.

6.1.4. Zustandsdichte und Greensche Funktion. Über die Grundzustandseigenschaften und die damit verbundenen kritischen Parameter hinausgehend, gehört die Form der Hubbardbänder in der Ein-Teilchen-Zustandsdichte (*density of states*, DOS) des Hubbardmodells und die daraus zugängliche Bandlücke zu den großen offenen Fragen der Theorie des Mottisolators. Im Besonderen lassen DMFT-Rechnungen sowohl in der isolierenden [21], als auch der metallischen [29] Phase im kritischen Bereich bislang unerklärte scharfe Strukturen an den Bandkanten bei niedriger Energie erkennen, siehe Abbildung 6.1. Einerseits wurde bislang die Zustandsdichte der Mottischen Isolatorphase störungstheoretisch bis $\mathcal{O}(U^{-2})$ direkt aus dem Hubbardmodell berechnet [17], andererseits liegt das Ergebnis bis $\mathcal{O}(U^{-3})$ aus der Lösung der DMFT-Selbstkonsistenzgleichung vor [31]¹⁵⁰; diese Resultate erklären aber nicht vollständig die beobachteten Strukturen. Weil zudem die DMFT-Resultate für die Zustandsdichte empfindlich vom genutzten Lösungsschema für das *self-consistent impurity problem* abhängen [29], sind genauere Daten, besonders durch Störungsreihenterme hoher Ordnung unabdingbar.

Die Zustandsdichte und die Mott-Hubbard-Bandlücke¹⁵¹ sind aus der lokalen Einteilchen-Greenfunktion zugänglich, für deren Entwicklung sich die in dieser Studie umfassend behandelte kombinatorisch-graphentheoretische Herangehensweise ebenfalls eignet. Von EVA KALINOWSKI wurde in ihrer Dissertation [15–17] die „manuelle“ analytische Berechnung der Grundzustandseigenschaften bis einschließlich vierter Ordnung und der Zustandsdichte auf Grundlage der Greenschen Funktion bis einschließlich zweiter Ordnung in W/U , worin W die Bandbreite des nicht-wechselwirkenden Systemes ist, durchgeführt, und die Ergebnisse stimmen mit DMRG-gelösten DMFT-Simulationen für $U \geq 1,5W$ gut überein. Gegenstand laufender Arbeiten ist nun, den hier verwendeten Formalismus für die Greensche Funktion bis $\mathcal{O}(U^{-4})$ zunächst von Hand zu zeigen, wodurch insbesondere der Weg zu einer „maschinellen“ Fortführung zu hohen und prinzipiell beliebigen Ordnungen geebnet und die Arbeiten für die notwendigen Modifikationen der vorgestellten hochoptimierten und parallelisierten Implementierung des Algorithmus’ abgeschlossen werden können [59].

Die Reihenentwicklung der Greenfunktion bei endlichem U kann als selbstkonsistente Integralgleichung formuliert werden, wobei in dieser Gleichung Polynome der Funktion selbst für $U \rightarrow \infty$ mit steigenden Ordnungen enthalten sind. Bei der Berechnung der Koeffizienten dieser Polynome kann im Wesentlichen wie bei der Berechnung der Reihenoeffizienten der Grundzustandsenergie in dieser Abhandlung vorgegangen werden, wiewohl der Rechenzeitbedarf vermutlich bedeutend höher sein wird. Unklar ist

¹⁴⁹Beispielsweise mittels *Multigraphen*

¹⁵⁰In Referenz [31] wurde zudem aus der Bandlücke ein Wert von $U_c \approx 4,406$ für die kritische Wechselwirkung gefolgert.

¹⁵¹Denkbar ist auch eine Bestimmung der Bandlückenbreite aus der Differenz der Grundzustandsenergien des Quantensystems bei Halbfüllung und mit einem Loch, wobei allerdings die Einwände aus Abschnitt 6.1.2 zu berücksichtigen sind.

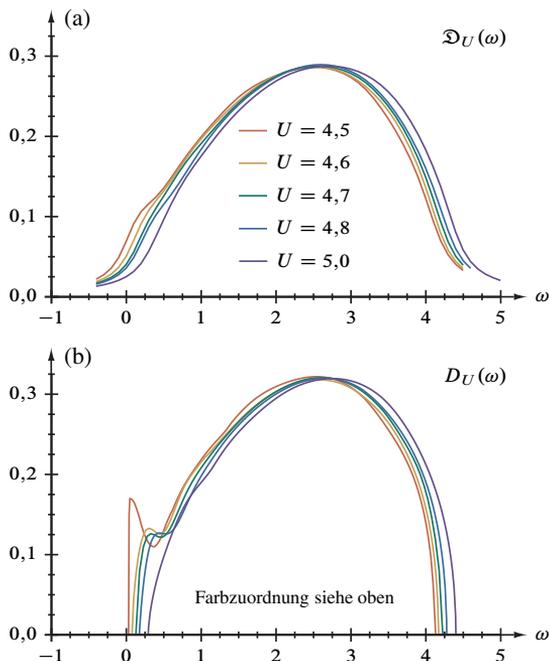


Abbildung 6.1. DDMRG-Ergebnisse [21] für die Zustandsdichte des Hubbardmodells (oberes Band) für verschiedene Wechselwirkungsstärken U . In Abbildungsteil (a) sind die mit einem Lorentzkern frequenzabhängiger Breite gefalteten „Rohdaten“ $\mathcal{D}_U(\omega)$ dargestellt, Teil (b) zeigt im gleichen Maßstab die entfaltenen Resultate $D_U(\omega)$ – siehe Teil 1 dieser Arbeit. Man beachte, daß die Werte für U ($4,4 \lesssim U_c \lesssim 4,5$ für DDMRG-DMFT) gegenüber den Ergebnissen dieser Arbeit ($4,76 \lesssim U_c \lesssim 4,78$ für EPT) zu kleinerer Wechselwirkung verschoben sind, vergleiche Abschnitt 4.3.3.

allerdings noch, ob – zumindest im ersten Ansatz – nicht neben den diskret-mathematischen Techniken auch numerische Analysis zur Lösung der selbstkonsistenten Integralgleichung benötigt wird, ob also das Ergebnis exakt oder nur näherungsweise angegeben werden kann. Trotz dieser Unsicherheit wären bereits einige ordnungshöhere Beiträge zur Greenschen Funktion von beträchtlichem Wert: Mit genauerer Kenntnis der Zustandsdichte und der Bandlücke ist auch die Bestimmung der kritischen Wechselwirkung U_c und des kritischen Exponenten τ präziser möglich, was ein besseres Verständnis des paramagnetischen Mottüberganges zwischen Metall und Isolator zu gewinnen erlaubt. Darüber hinaus bieten diese Resultate einen unmittelbaren und vollständigeren *benchmark* für numerische DMFT-Methoden, weil diese derzeit auf Selbstkonsistenz-Rechnungen der Greenfunktion beruhen.

6.1.5. Bose-Einstein-Kondensate in optischen Gittern. Eine paramagnetische Mottisolatorphase ist nicht nur bei stark wechselwirkenden fermionischen, sondern auch bei bosonischen Quantensystemen und Mischungen beider prinzipiell möglich. Neben organischen Schichtstrukturen finden sich diese experimentell an Quantengasen in *optischen Gittern* [60]. Während der Phasenübergang beim „elektronischen“ System zum Metall erfolgt, entsteht bei Bosonen das *Suprafluid*. TOSIO KATOS Störungsreihe wurde in der Vergangenheit bereits zur Untersuchung der Phase des Mottisolators am entsprechenden *Bose-Hubbard-Modell* mit spinlosen Bosonen im Limes starker Wechselwirkung verwendet [61]; sofern der ungestörte Grundzustand nicht entartet ist, kann dieser Ansatz leicht zur Berechnung hoher Ordnungen genutzt werden. Für Bosonen mit Spin $S \neq 0$ erschwert allerdings die Grundzustandsentartung die Behandlung immens, und entsprechende Arbeiten wurden bislang – soweit bekannt – nicht durchgeführt. Da das Entartungsproblem jenem am elektronischen Hubbardmodell in der Kato-Takahashi-Störungstheorie und somit dem in dieser Arbeit behandelten Fall direkt verwandt ist, scheint eine künftige Anwendung des Graphenalgorithmus’ sowohl auf die Mottphase des Hubbardmodells für ($S \neq 0$)-Bosonen bei Spinunordnung [62], als auch die Mottphase von anderen Fermionen-Systemem [11] oder Boson-Fermion-Mixturen [63] in optischen Gittern durchaus realistisch [9].

6.1.6. Konzeptionelle Übertragung auf andere Reihenentwicklungen. Ein interessanter Aspekt künftiger Forschung könnte die gewinnbringende Übertragung der Konzepte der hier vorgestellten Großrechnerimplementierung der Kato- beziehungsweise Kato-Takahashi-Störungsrechnung auf andere Reihenentwicklungen [64, 65] sein. Ein Beispiel dafür ist die Methode *kontinuierlicher unitärer Transformationen (continuous unitary transformations, CUT)* [66], die das Hubbardmodell im Starkwechselwirkungslimes auf ein *effektives Modell* abbildet, in welchem die Doppelbesetzungszahl erhalten bleibt. Unter Verwendung einer passenden Trunkierung lassen sich die Flußgleichungen des effektiven Hamiltonoperators schließen und somit lösen [67]. Dies führt zu einer systematischen Entwicklung dieses Hamiltonoperators und anderer Observablen in Potenzen von $1/U$ – sehr ähnlich der hier verwendeten Reihenentwicklung KATOS. Ein möglicher Ansatz besteht nun darin, die Gleichungen „auf Störungsart“ zu trunkieren und dabei in einer Reihe zu entwickeln [68].¹⁵²

6.2. Mathematische Aspekte.

6.2.1. Derzeitiger Stand. Reihenentwicklungen gehören zu den grundlegenden mathematischen „Werkzeugen“ der Theoretischen Physik und werden auch oft bei der Untersuchung stark korrelierter Gittermodelle wie dem Hubbardmodell genutzt, siehe beispielsweise Referenz [64]. Eine attraktive Eigenschaft dieser Methoden besteht in deren oftmals guter Eignung für die Verwendung von gegebenenfalls spezialisierten¹⁵³ Computeralgebraprogrammen. Überdies kann – wie der vorliegende Fall eindrucksvoll belegt –

¹⁵²Ein kürzlich in Referenz [69] vorgestelltes Verfahren, das jedoch nicht auf einer Störungsrechnung beruht, bedient sich ebenfalls graphentheoretischer Techniken.

¹⁵³Beispielsweise FORM [83]

unter bestimmten Umständen eine optimierte Großrechnerimplementierung im Sinne einer *Hochleistungs-Computeralgebra* erfolgen, die dadurch zu einem Gegenstück der überwiegend genutzten *Hochleistungs-Computernumerik* wird. Auf diesem Wege läßt sich das enorme Potential moderner wissenschaftlicher Datenverarbeitungsanlagen nutzen, ohne auf die Strenge analytischen Rechnens verzichten zu müssen, da vom schwerwiegenden Nachteil endlicher Genauigkeit der numerischen Algorithmen und in Besonderheit der Fließkomma-Arithmetik die Ganzzahloperationen nicht betroffen sind, ob schon auch Ordnung, also die Anzahl der Reihenglieder endlich bleibt.

Aus Perspektive der Mathematik ist die hier gewählte Darstellung der Störungsrechnung zunächst „nur“ der Gebrauch graphentheoretischer Definitionen, welche eine in sich geschlossene Formulierung des Verfahrens erlaubt, denn der Reihenkoeffizient wird nicht im eigentlichen Sinne anhand von Graphen *berechnet*. Indes ist die formale Beschreibung und Analyse Voraussetzung für ein tieferes Verständnis und eine Fülle von algorithmischen Verbesserungen, insbesondere durch das „Aufspüren“ und Ausnutzen von Symmetrien geworden, welche teils mit dem Computeralgebrasystem *Mathematica*, vor allem aber in Kooperation mit der Theoretischen Informatik unter Verwendung der funktionalen Programmiersprache *Haskell* ausgearbeitet wurden. Dabei hat sich gezeigt, wie sehr eine enge Kooperation allen beteiligten Disziplinen zugutekommt: Vom Standpunkt der Physik sind hochgenaue Ergebnisse für zentrale Modelleigenschaften zugänglich geworden, für die Informatik liegt ein bereits sehr komplexes Problem aus der Anwendung vor, das unmittelbar die laufende Forschung berührt, gar herausfordert [33], und seitens der Mathematik werden beispielsweise über die Nutzung¹⁵⁴ und Ergänzung¹⁵⁵ der Ganzzahldatenbank OEIS hinaus Zahlenfolgen aus bislang unverbundenen Fragestellungen verknüpft [70].

6.2.2. *Allgemeine Aussichten.* Kraft der in der Begriffswelt der Diskreten Mathematik verankerten Darstellung dieser Rechenmethode – Kombinatorik auf Graphenstrukturen mit zahlentheoretischen Aspekten – ist der Grund für die künftige weitergehende wahrhaft „mathematische“ Behandlung gelegt, welche möglicherweise die Angabe der Reihe nicht bloß durch *Zählung* von gewichteten Hüpfsequenzen, sondern durch *Berechnung* deren Anzahl erlaubt. Das später im Abschnitt 6.3.3 skizzierte Verfahren¹⁵⁶ könnte dabei ein Ansatz sein.

¹⁵⁴Primär wurde auf folgende Sequenzen zurückgegriffen: A000055, A000081 und A038055 für den Aufbau der Ausgangszustände, A001006, A002426, A005773 und A025565 für die Berechnung der Hüpfprozesse. Details finden sich in den jeweiligen Abschnitten.

¹⁵⁵Beigetragen wurden folgende Sequenzen: Die Ausgangszustandsgesamtzahl A198760 und hilfweise dazu A242353 und A242354, die Hüpfsequenzgesamtzahlen A198761 (Hubbardmodell) und A225823 (Falicov-Kimball-Modell), die Doppelbesetzungsverlaufsgesamtzahlen A240605 und die Reihenkoeffizienten, A226584 (Zähler) und A226585 (Nenner). Die Anzahl der Hüpfprozesse und zugehörige Zahlenfolgen, siehe vornehmlich Tabelle 3.4, sind Gegenstand späterer Veröffentlichungen in der OEIS.

¹⁵⁶Siehe insbesondere Gleichung (1.3) im Anhang

Eine interessante Beobachtung, welche sich aus der Analyse der (bereits nach in Referenz [19] vorgestelltem Verfahren) erzeugten Hüpfsequenzen ergibt, besteht in der „Verwandschaft“ von Bäumen, folglich auch (gegebenenfalls färbungsinversen) Ausgangszuständen, die der selben Topologiekategorie angehören. Deren Anzahl γ_k ist, wie aus Tabelle 2.1 zu ersehen, in zunehmender Ordnung beträchtlich kleiner, als die Anzahl der Strukturen λ_k ; ließe sich ein modifizierter Algorithmus für die Reduktion auf Topologieklassen mit entsprechenden Multiplizitäten für die einzelnen Sequenzen finden, ginge damit ein beträchtlicher Geschwindigkeitszuwachs einher. Konkret findet sich in der Ordnung U^{-3} bei den diesbezüglich zusammengehörigen¹⁵⁷ Ausgangszuständen $\mathcal{B}_{3,1,26}$ und $\mathcal{B}_{3,2,18}$ die Sequenzpaare $((3.103-1), (3.103-7))$, $((3.103-2), (3.103-8))$, $((3.103-3), (3.103-10))$, sowie $((3.103-4), (3.103-9))$ und bei $\mathcal{B}_{3,1,27}$ und $\mathcal{B}_{3,2,16}$ das Paar $((3.103-5), (3.103-6))$. Entsprechendes gilt für alle Ordnungen beispielsweise für alle Sequenzen in der „Zusammengehörigkeit“ $B_{k,\lambda_{k-1}} \bowtie B_{k,\lambda_k}$, womit der Sonderfall $a_{k,\lambda_{k-1}} = a_{k,\lambda_k}$ verbunden ist. Weitere vertiefende Untersuchungen¹⁵⁸ fänden, wie

¹⁵⁷Siehe Fußnote 100

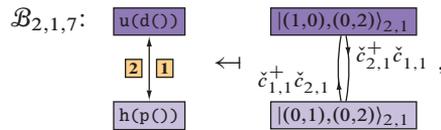
¹⁵⁸Es ergibt sich beispielsweise für die Störungsrechnungsordnung n beziehungsweise die Graphenordnung k mit der Graphentopologiekategorie $g = 1, 2, \dots, \gamma_k$ und der Baumstrukturnummer $l \in [1, \lambda_k]$, vergleiche Tabelle 2.1, sowie $\mathbb{h}_k = \sum_{l=1}^{\lambda_k} \mathbb{h}_{k,l} = \mathbb{h}_{(n+3)/2}$ und $a_k = \sum_{l=1}^{\lambda_k} a_{k,l} = a_{(n+3)/2}$, siehe Tabelle 3.8 und Gleichungen (4.2) und (4.3) (respektive Tabelle 4.1), folgende Aufschlüsselung der Hüpfsequenzanzahlen und Reihenkoeffiziententeile:

n	k	g	l	Hubbardmodell				Falicov-Kimball-Modell			
				$\mathbb{h}_{k,l}/2$	$\mathbb{h}_k/2$	$a_{k,l}$	a_k	$\mathbb{h}_{k,l}$	\mathbb{h}_k	$a_{k,l}$	a_k
1	2	1	1	1	1	$-\frac{1}{2}$	$-\frac{1}{2}$	1	1	$-\frac{1}{4}$	$-\frac{1}{4}$
3	3	1	1	5	10	$-\frac{1}{4}$	$-\frac{1}{2}$	2	4	0	0
			2	5		$-\frac{1}{4}$		2		0	
5	4	1	1	84	324	$\frac{3}{8}$	$-\frac{19}{8}$	14	54	$\frac{1}{32}$	0
			3	168		$-\frac{27}{8}$		28		$-\frac{5}{32}$	
		2	36	$\frac{5}{16}$		6		$\frac{1}{16}$			
		4	36	$\frac{5}{16}$		6		$\frac{1}{16}$			
7	5	1	1	2962	22736	$\frac{487}{32}$	$-\frac{593}{32}$	204	1568	$\frac{123}{256}$	0
			5	5924		$-\frac{43}{4}$		408		$-\frac{1}{8}$	
			7	2962		$-\frac{831}{32}$		204		$-\frac{155}{256}$	
			2	1278		$-\frac{529}{256}$		88		$-\frac{85}{256}$	
		3	2556	$\frac{4397}{256}$		176		$\frac{201}{256}$			
		6	2556	$-\frac{4153}{256}$		176		$-\frac{153}{256}$			
		8	3834	$\frac{773}{256}$		264		$\frac{133}{256}$			
		3	4	332		$\frac{17}{32}$		24		$-\frac{1}{16}$	
9	332	$\frac{17}{32}$	24	$-\frac{1}{16}$							

am Beispiel im folgenden Abschnitt belegt, idealerweise Platz in einer fächerübergreifenden Zusammenarbeit.

6.2.3. *Aufbau und Wegsuche auf Zustandsgraphen.* Hüpfsequenzen $H^{(n)}$, definiert in Gleichung (3.87), lassen sich, wenn auch in dieser Studie linear in Art einer „Reaktionsgleichung“ geschrieben, ihrerseits als Graph¹⁵⁹ auffassen und notieren. Werden alle diese Sequenzgraphen eines gegebenen Ausgangszustandes vereinigt, entsteht naturgemäß wiederum ein Graph, siehe Abschnitt 2.1.1. Dieser umfaßt mit seiner Eckenmenge sämtliche Zustände, die das Quantensystem einnehmen kann, und mit seiner Kantenmenge alle möglichen Übergänge zwischen diesen; folglich kann er als *Zustandsgraph* bezeichnet werden. Darin sind entsprechend die einzelnen Hüpfsequenzen als Wege enthalten, vergleiche Abschnitt 2.1.3, welche stets durch (abgesehen von Teilchenvertauschungen) identischen Ausgangs- und Endzustand eine „Hauptschleife“ bilden. Weitere Schleifen können auftreten, sofern die Ausgangs- oder Zwischenzustände mehrfach erreicht werden.

In erster Störungsrechnungsordnung, also der Baumordnung $k = 2$, ergibt sich aus der einzigen Hüpfsequenz, siehe Gleichung (1.27), des (bis auf Färbungsinversion bezüglich der Wurzel) einzigen Ausgangszustandes der Zustandsgraph



worin die dunkelviolette Ecke den Ausgangszustand und die hellviolette Ecke den ersten (und hier einzigen) Zwischenzustand kennzeichnet¹⁶⁰. Da der Übergang zwischen zwei verbundenen Zuständen sich stets eindeutig aus deren „Differenz“ rekonstruieren läßt, benötigen die Kanten außer dem Richtungspfeil kein Attribut. Der Übersichtlichkeit halber vermerken wir aber unter Anwendung der in dieser Arbeit üblichen Farbkodierung ($\boxed{\blacksquare} \mapsto \uparrow$ -Spin, $\boxed{\blacksquare} \mapsto \downarrow$ -Spin; blässere Druckfarben für bessere Lesbarkeit von Beschriftungen) die propagierende Elektronensorte sowie die Kantenummer. Weitere Details zum

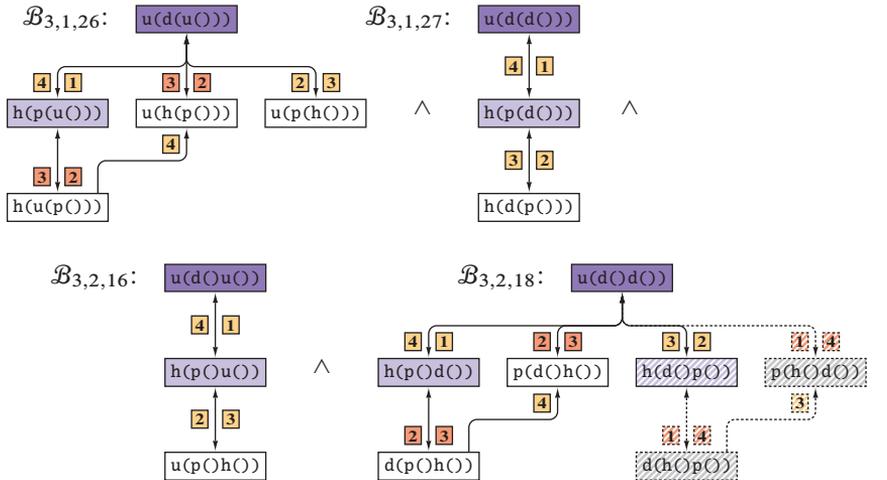
¹⁵⁹Verglichen mit den Permutations- und Binärbäumen, welche hinsichtlich der Bethegitterausschnitte formal Metagraphen sind, handelt es sich dabei ebenfalls um einen „Graphen zweiter Art“.

¹⁶⁰Wir reduzieren unter Aufgabe der „Teilchenpersonalisierung“, siehe Abschnitt 3.4.2, die Schreibweise zum Beispiel für $\mathcal{B}_{3,1,26}$ auf

$$\begin{aligned} & |(1, 0, 3), (0, 2, 0)\rangle_{3,1} \mapsto \text{u(d(uO))} \quad \text{sowie} \\ & \hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow} \mapsto \hat{c}_{2,1}^+ \hat{c}_{1,1} \mapsto \boxed{\hat{c}_2^+ \hat{c}_1} \mapsto \boxed{(1, 2)} \mapsto \boxed{1} \quad \text{und} \\ & \hat{c}_{3,\downarrow}^+ \hat{c}_{2,\downarrow} \mapsto \hat{c}_{3,2}^+ \hat{c}_{2,2} \mapsto \boxed{\hat{c}_3^+ \hat{c}_2} \mapsto \boxed{(2, 3)} \mapsto \boxed{2}. \end{aligned}$$

Hintergrund und der Notation werden insbesondere im Abschnitt 3.4.3 und folgendem umfassend erläutert.

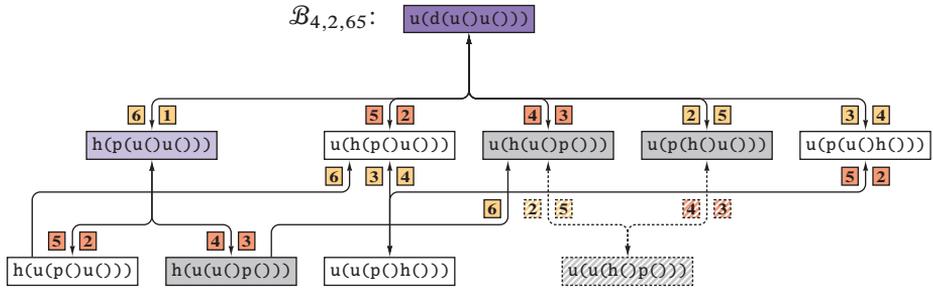
In dritter Ordnung, entsprechend $k = 3$, treten zu den bereits benannten Ecken weiße hinzu, die solche Zwischenzustände im Sequenzverlauf markieren, welche sich an deren Initialisierung anschließen:



Nach den Gleichungen (3.103) ergeben sich aus den vier Ausgangszuständen insgesamt $4 + 1 + 1 + 4 = 10$ Hüpfsequenzen, welche die obigen vier Graphen bilden. Der vierte und letzte Ausgangszustand $u(d()d())$ führt dabei – bedingt durch seine Symmetrie – erstmals auf eine Besonderheit in den Zustandsgraphen: Da die Kanten 1 und 2 der selben Klasse angehören¹⁶¹, ${}^m E_{\text{CNL}}(\mathcal{B}_{3,2,18}) = \{\{1, 2\}\}$, darf nur ein Repräsentant zur Sequenzinitialisierung genutzt werden (konventionsgemäß, siehe Abschnitt 3.4.5, der erste); daher stellt der auf schraffierter Fläche befindliche Baum $h(d()p())$ einen Zustand dar, der im hiesigen Fall nicht bereits nach dem ersten Hüpfschritt über Kante 2, sondern erst nach den drei Schritten über 1, 4 und 2 erreicht werden kann und im folgenden letzten Sprung über Kante 3 zum Ausgangszustand aufgelöst werden muß. Dieser Weg entspricht der Sequenz in Gleichung (3.103-9), dem die Kantenpermutation ${}^m \tilde{G}_4 = \{1, 4, 2, 3\}$ zugrundeliegt. Folglich können – im Gegensatz zu ihren „Spiegelbildern“ $p(d()h())$ und $d(p()h())$ – die gestrichelt umrahmten Zustände $p(h()d())$ und $d(h()p())$ nicht entstehen.

¹⁶¹Vereinfachend werden hier nur die Indices der Kanten innerhalb der Hüpfkantenmenge G , das heißt Elemente der zugehörigen Meta-Menge ${}^m G$ angegeben, siehe Fußnote 160 und verdeutlichend in den Skizzen durch Fettdruck hervorgehoben.

Die folgende Skizze zeigt die zunehmende Komplexität der Zustandsgraphen anhand eines zur fünften Störungsrechnungsordnung, somit zu $k = 4$ gehörenden Baumes:



Dieser Graph besteht aus 16 Sequenzen. Weil, wie oben bereits beim Initialisierungssprung von $u(d()d())$ erwähnt und eingehend in Abschnitt 3.4.5 dargelegt, Permutationen mehrerer Kanten einer Klasse (hier 2 und 3 aus ${}^mE_{\text{CNL}}(\mathcal{B}_{4,2,65}) = \{\{1\}, \{2, 3\}\}$) nicht auftreten dürfen, wird überdies zwar der Zustand $u(u(p)h())$, nicht jedoch der (zu diesem isochrome) Zustand $u(u(h)p())$ realisiert.

Schon in der Anfangsphase der parallelisierten Neuimplementierung von EVA KALINOWSKIS Rechenverfahren ist in Zusammenarbeit mit der Arbeitsgruppe für Theoretische Informatik von RITA LOOGEN die Idee entstanden,

- (1) zunächst den Zustandsgraphen für den Baum $\mathcal{B}_{k,l,m} \in B'_{k,l}$ direkt zu generieren,
- (2) sodann darauf alle Hüpfsequenzen in Form aller die Sequenzkriterien (im Rahmen der Problemstellung: ausschließlich Sprünge zu nächsten Nachbarn, Initialisierung von der Baumwurzel, einmalige „Erfassung“ einer jeden Kantennummer, Rückkehr zum Ausgangszustand, Einhaltung des Pauliprinzips bei Fermionen) erfüllenden Wege der Länge $2(k - 1)$ zu isolieren, und
- (3) abschließend die das Vorzeichen des Sequenzbeitrages festlegende Teilchenvertauschungsanzahl b zu bestimmen, im einfachsten Falle durch Verfolgung der Teilchennummern im Sequenzverlauf.

Wesentlich für die Konstruktion der Zustandsgraphen ist dabei die Erkenntnis, daß deren Ecken für den Ausgangs- und (zumindest) den ersten Zwischenzustand der Menge der viergefärbten Butcherbäume entstammen müssen, überdies im Allgemeinen aber auch Isochromismen derselben sein können. Ein Beispiel dafür ist $h(d)p() \cong h(p)d()$ für $\mathcal{B}_{3,2,18}$; alle Weiteren¹⁶² sind in den Skizzen der Zustandsgraphen dieses Abschnittes grau hinterlegt.

Diese Lösung des rechenzeitbestimmenden Hüpfproblems wurde erstmalig von GUNTHER GRUBER in funktionaler Formulierung mit Haskell umgesetzt. Dabei wurden, wie in Abschnitt 3.4.1 angesprochen und in allen Details in Referenz [33] behandelt,

¹⁶² $h(d)p()$ ist ein spezieller Fall: Dieser Zustand, nach unserer Definition farbordnungsbedingt kein „echter“ gefärbter Butcherbaum, tritt zwar, wie zuvor behandelt, innerhalb einer Sequenz auf, kann aber als Isochromismus nicht an deren Anfang stehen.

jeweils¹⁶³ ein Produktgraph, dort auch Hüpfschrittgraph genannt, gebildet, welcher den eigentlichen Zustandsgraphen enthält. Inwieweit sich auf dieser Grundlage der (dem in vorliegender Arbeit beschriebenen Algorithmus wesenseigenen) fakultativen Zeitkomplexität „entrinnen“ läßt, bedarf neben der Aufwandabschätzung für eine Programmierung, besonders auf Parallelrechenanlagen, noch weitergehender Untersuchung. Möglicherweise bietet die Graphentheorie für ein solches Verfahren mehr und effizientere Ansatzmöglichkeiten, als in der ursprünglichen Methode, zumal *Routenplanungs- und Rundreiseprobleme*, hier mit fester Weglänge und gegebenenfalls mehrfachem Eckenbesuch, intensiv erforscht sind [37–40, 43, 71, 72]. Inkaufgenommen werden muß dabei allerdings der vermutlich bedeutend höhere Schwierigkeitsgrad der Zustandsgraphenerzeugung, welcher mit nicht nur halbgefüllten und zudem angeregten Ausgangszuständen nochmals anwachsen könnte.

6.3. Datenverarbeitungstechnische Aspekte.

6.3.1. Derzeitiger Stand. Die Neuimplementierung ermöglicht die Berechnung der für die Bewertung numerischer Ergebnisse entscheidenden Ordnungen unter Verwendung im wissenschaftlichen Umfeld üblicher Datenverarbeitungssysteme in der Größenordnung von Tagen, siehe Abschnitt 5.1.2. Bereits auf der für die Entwicklung genutzten „klassischen“ *UNIX-workstation* [85] liegt die Rechenzeit des Hubbardmodells für $\mathcal{O}(U^{-9})$ um 10 Sekunden und für $\mathcal{O}(U^{-11})$ bei etwa 37 Minuten auf einem Prozessor; bei Nutzung einer zweiten CPU sinkt diese auf 5,3 Sekunden respektive 19 Minuten. Auch für die in Abschnitt 6.1.2 erwähnten Erweiterungen des Verfahrens sind somit erste orientierende Ergebnisse von bereits beachtlicher Genauigkeit ohne die Nutzung größerer *computeserver* zugänglich. Aufgrund der idealen Datenparallelität ist darüberhinaus für „Produktionsrechnungen“ exzellentes Skalierungsverhalten zu erwarten; bestätigt wurde dies für bis zu 510 Prozessorkerne beziehungsweise 1020 Prozeßfäden [89]. Eine vertiefte Analyse mit bis zu 256 *threads* wird in Abschnitt 6.3.4 vorgestellt.

Neben vielen Ansatzpunkten für die Verfahrensanpassung auf andere Fragestellungen eignet sich die neue Implementierung – naheliegenderweise – im Prinzip auch für höhere Ordnungen, wobei auf absehbare Zeit für das Hubbardmodell nur $\mathcal{O}(U^{-17})$ und für das Falicov-Kimball-Modell nur $\mathcal{O}(U^{-19})$ realistisch erscheint: Eine das Konzept des „globalen Zählers“, siehe Abschnitt 5.2.1, prüfende Rechnung am Falicov-Kimball-Modell siebzehnter Ordnung benötigt auf einem *EPIC-server* vom Typ HP AH353A [86] knapp 191 Stunden, etwa 2000 Prozesstagen entsprechend, bei einem Speicherbedarf von 16,4 GWorten, also 131 GBytes; auf Grundlage der Betrachtungen in Abschnitt 5.1.2

¹⁶³GUNTHER GRUBER geht in einer ersten Implementierungsmöglichkeit zunächst von unverwurzelten Bäumen aus. Da beispielsweise in dritter Störungsrechnungsordnung nur $\gamma_3 = 1$ Topologie mit zwei relevanten Färbungen existiert – die Butcherbäume $\mathcal{B}_{3,1,26}$ und $\mathcal{B}_{3,2,16}$ sowie $\mathcal{B}_{3,1,27}$ und $\mathcal{B}_{3,2,18}$ gehören paarweise zusammen –, existieren nur zwei Produkt- beziehungsweise Hüpfschrittgraphen. Die Festlegung der Wurzel erfolgt, vereinfacht gesagt, erst im Zuge der „Extraktion“ der Hüpfsequenzen. Eine zweite, ebenfalls in Referenz [33] diskutierte Herangehensweise beruht unmittelbar auf den (verwurzelten) Bäumen, erzeugt also implizit die oben skizzierten vier Zustandsgraphen.

lassen sich als Größenordnung des Zeitaufwandes für das Hubbardmodell gleicher Ordnung 750000 Prozessortage für die aktuelle Maschinengeneration abschätzen. Immerhin kann dieser Bedarf – zulasten des Hauptspeichers – durch das im Abschnitt 6.3.3 beschriebene Verfahren mindestens auf ein Drittel vermindert werden.

6.3.2. *Allgemeine Aussichten.* Obschon durch diese Arbeit eine parallelisierte und bereits stark effizienzoptimierte *greedy*-Implementierung des Algorithmus' existiert, liegen – neben den im vorausgegangenen Abschnitt 6.2.2 angesprochenen grundlegenden Ansatzmöglichkeiten auf Ebene der Mathematik – zahlreiche weitere Modifikationen mit teils erheblichem Einfluß auf den Rechenzeitbedarf nahe. Zu den zentralen *algorithmischen Verfeinerungen* gehören in der Reihenfolge des Programmablaufs:

Bereich Ausgangszustände: *Parallelisierte Baumerzeugung* innerhalb einer Graphenordnung k . Bei $k = 10$ und $c = 4$, also Vierfärbung, beläuft sich die serielle Rechenzeit für die ${}^c\mathcal{B}_{k,l,m}$ bereits, wie in Tabelle 3.1 verzeichnet, auf etwa 12 Stunden. Wird beispielsweise die zuvor angesprochene Mischung aus Spin-1/2- und Spin-3/2-Fermionen, welche die Farbanzahl auf $c = |X| = 16$ erhöht, verallgemeinert Gegenstand der Betrachtung in Abschnitt 3.4.1, werden durch die Komplexität $\mathcal{O}(c^k \lambda_k)$ schon in deutlich niedrigeren Graphen- beziehungsweise Störungsrechnungsordnungen¹⁶⁴ ähnliche Laufzeiten erreicht, sofern sämtliche Anregungen mitgeneriert werden. Selbst bei mäßiger Parallelisierungseffizienz entstünde im Sinne eines flexiblen Einsatzes der Methode durch diese Änderung ein großer Vorteil; Ansatzpunkt ist beispielsweise die Schleife 7 innerhalb des in Abschnitt 3.2.1 beschriebenen Algorithmus', wobei (abhängig von der Umsetzung) ${}^c\mu_{k,l}$ im voraus berechnet und die Behandlung des Färbungsnummernzählers m angepaßt werden muß.

Dualkodierung der Butcherbäume in Struktur- und Färbungszahl. Ein bedeutender Verarbeitungsaufwand bei der Baumerzeugung wird durch die Zeichenkettenkodierung der Butcherbäume verursacht, da *string*-Manipulationen wie das Aneinanderhängen von Ästen an eine Wurzel, siehe Gleichung (3.19), im Allgemeinen zeichenweise ausgeführt werden und somit auf eine verhältnismäßig lange Befehlsabfolge zurückgehen. Neben der hinsichtlich Registeranzahl und Pufferspeichergrößen bedeutsamen Platzersparnis – ein Datenwort zu 64 Bit pro Struktur, ein Datenwort zu 64 Bit pro Färbung¹⁶⁵ – besitzt hingegen die binäre Ganzzahlkodierung der Bäume den Vorzug, nur weniger hocheffizienter *integer*-Operationen zu bedürfen, im Wesentlichen der *bit*-Verschiebung und der Addition, siehe

¹⁶⁴Die Baumordnung $k = 6$, folglich die Störungsrechnungsordnung $n = 9$, liegt bei knapp der Hälfte des genannten Aufwandes, $k = 7$ bereits eine Größenordnung darüber.

¹⁶⁵Jeweils acht Zeichen entsprechend. Diese bei Vierfärbung bis $k = 16$ konstante Datenmenge wird in Zeichenkettendarstellung bereits bei $k = 6$ überschritten beziehungsweise, sofern Bäume gemäß Abschnitt 2.2.3 „komprimiert“ gespeichert werden, bei $k = 4$ erreicht. Eine weitere Reduktion tritt ein, wenn die Struktur ${}^S[{}^4\mathcal{B}_{k,l}]_2$ für alle ${}^4\mu_{k,l}$ Färbungen ${}^F[{}^4\mathcal{B}_{k,l,m}]_2$ nur einmal gespeichert wird; diese beläuft sich im Grenzfall hoher Ordnungen und somit Färbungszahlen auf ein Datenwort, also acht *bytes*.

Gleichungen (2.35) und (2.36). Im Zuge einer solchen Programmodifikation muß jedoch auch die auf Zeichenketten basierende rekursive Vorbereitungsstufe, siehe Abschnitt 3.4.11, angepaßt werden, was separat gespeicherte Astmultiplizitäten erfordert.

Bereich Hüpfprozesse: *Verminderung der Prozeßanzahl* durch Festlegung der neben Grundzustandsprojektoren \hat{P}_0 befindlichen \hat{S} -Exponenten auf $\kappa = 1$ bereits auf Stufe der \mathcal{T} - und \mathcal{D} -Produkte. Diese Methode zur „Kompression“ der $\hat{P}^{(n)}$ wurde in Abschnitt 3.3.4 für den Halbfüllungsfall entwickelt, ist jedoch auf Prozesse auf beliebigen Ausgangszuständen ausweitbar. Im Unterbereich Zählerprodukte innerhalb des in Abschnitt 3.3.2 beschriebenen Algorithmus' kommen dafür die Programmschritte 6.1 und 6.4 in Frage; die Verringerung der Zählerproduktanzahl wirkt sich gemäß Gleichung (3.44) auch auf Schritt 8.3 im Unterbereich Nennerprodukte günstig aus, vergleiche Tabelle 3.3.

Bereich Hüpfsequenzen: *Vorabberechnung des Energiebeitrages* eines jeden Hüpfsequenztyps innerhalb der Störungsrechnungsordnung n . Da jede Sequenz eindeutig einem der d_n möglichen Doppelbesetzungsverläufe zugehörig ist, läßt sich, wie im folgenden Abschnitt 6.3.3 eingeführt und umfassend im Abschnitt 1 des Anhangs ausgearbeitet, deren Anteil am Reihenkoeffizienten und folglich der Grundzustandsenergie im voraus bestimmen. Diese Verbesserung des Kernalgorithmus' in Programmschritt 8, siehe Abschnitt 3.4.13, und darin im Wesentlichen in Schritt 5 der Betragsermittlung, siehe Abschnitt 3.5.1, bewirkt einen bedeutenden nochmaligen Leistungsgewinn, da die Rechenzeit fast vollständig durch den Programmteil zur Ausgangszustandssequenzierung bestimmt wird.

Weitere Verfeinerung der Auftragsgranularität. Da beispielsweise für $k_1 = 10$ ($n = 17$) gegenüber $k_2 = 9$ ($n = 15$) der Kantenpermutationsaufwand um die Ordnung $((k_1 - 1)(2k_1 - 3)!)/((k_2 - 1)(2k_2 - 3)!) = 18 \cdot 17 = 306$ anwächst, die Ausgangszustandsanzahl jedoch nur um den Faktor $145920/30117 \approx 4,85$ zunimmt, wird ohne entsprechende Zunahme der Leistung eines einzelnen Prozessors das in Abschnitt 5.2.2 dargelegte Systemauslastungsproblem fortgesetzt. Zwar kann, wie dargelegt, der hüpfende Spin des dritten Schrittes bei einem Doppelbesetzungsloch-Sprung über eine zuvor noch nicht „übersprungene“ Kante nicht mehr eindeutig festgelegt werden (Binärbaumverzweigung), aber der Zwischenzustand zweiter Art ist forderungsgemäß eindeutig erzeugbar. Damit läßt sich das Konzept der Permutationsbaumzerlegung noch um eine weitere Stufe unverändert fortführen.

Begünstigend für die Berücksichtigung dieser Punkte, insbesondere aber für die in Abschnitt 6.1.2 diskutierte Anwendung des Verfahrens auf weitergehende Fragestellungen seitens der Physik erscheint eine *Modularisierung* des Programmes sinnvoll. Dadurch können „Zwischenergebnisse“, siehe beispielsweise Ablaufschema 3.1, wie die Tafeln

der Butcherbäume, Flußdiagrammelement (1)¹⁶⁶, und der Hüpfprozesse, Element (2), beziehungsweise der im folgenden Abschnitt 6.3.3 diskutierten Sequenztypbeiträge permanent auf Massenspeicher ausgelagert und zeitsparend erst vor der Sequenzgenerierung und -auswertung, Element (3), eingelesen werden.¹⁶⁷

Zu den wesentlichen *technischen Verbesserungen* gehört aufgrund der größenbegrenzten SMP-Systeme vorrangig die Portierung des Sequenzierungsalgorithmus' auf Verbundanlagen (*cluster*) mehrerer identisch konfigurierter Einzelrechner (*nodes*), die ihrerseits heutzutage symmetrische Multiprozessormaschinen sind und sich in ihrer Gesamtheit als Massivparallelrechner darstellen. Typischerweise sind die „Knoten“ mit bis zu zwei CISC-Mehrkernprozessoren bestückt¹⁶⁸ und über eine schnelle Netzwerkinfrastruktur verschaltet. Wiewohl sich mit dieser Technik Rechner mit weit über 10^4 Knoten aufbauen lassen, liegt darin der entscheidende Nachteil: Einerseits wird der Knoten-lokale Hauptspeicher der MPP-Architektur im Gegensatz zum oft Prozessor-lokal organisierten Speicher der SMP-Architektur nicht „global“ verwaltet, ist somit nicht zusammenhängend und bildet dadurch keinen linearen Adreßraum. Folglich entsteht innerhalb einer parallelierten Anwendung die Notwendigkeit, nicht nur Speicheradressen wie zwischen dem „*master thread*“ und den „*slave threads*“ eines OpenMP-Programmes auszutauschen, sondern sämtliche Daten (neben den Kontroll- und Steuerinformationen) zwischen den Knoten im *client-server*-Modell zu übertragen, was eine entsprechend optimierte Bandbreite voraussetzt. Andererseits bewirkt die zusätzliche Kommunikationsebene in der *software* höhere Latenzen¹⁶⁹, als diese durch die „reine“ *hardware*-Lösung in Multiprozessorrechnern auftreten.

¹⁶⁶Besonders dieser derzeit seriell ausgeführte Programmschritt führt mit zunehmender Größe der genutzten Parallelrechner zu signifikanten Leerlaufzeiten, siehe Abbildung 5.3.

¹⁶⁷In Vorbereitung von Nachfolgeprojekten aus Abschnitt 6.1.2 bis 6.1.4 sowie für *benchmark*-Zwecke, siehe Abschnitt 6.3.4, wurde dieses Modularisierungskonzept bereits umgesetzt.

¹⁶⁸Die ranghöchsten *supercomputer* [73] mit *cluster*-Aufbau bestehen derzeit zumeist aus SMP-Knoten, welche ihrerseits als *hosts* spezielle Koprozessorkarten mit hoher Kernanzahl verwalten, siehe Fußnote 121; dadurch ergibt sich im Allgemeinen mit dem Steckkarten-lokalen Arbeitsspeicher eine weitere Parallelisierungsebene – sowie in Konsequenz gegebenenfalls eine weitere Verengung des Anwendungsbereiches. Nicht in dieses Konzept fällt beispielsweise der aktuelle MPP-Höchstleistungsrechner des Forschungszentrums Jülich vom Typ IBM BlueGene/Q, welcher aus 28672 einheitlichen Knoten zu jeweils 16 nutzbaren RISC-Prozessorkernen (à vier *threads*) besteht.

¹⁶⁹Eine Ausnahme stellt in diesem Zusammenhang der im Entwicklungsstadium genutzte CISC-*server* vom Typ IBM 7233 [87] dar, bei welchem vier sogenannte *hypernodes* mit jeweils vier Sechskern-Prozessoren in Tetraedertopologie verschaltet sind und sich bereits auf unterer *software*-Ebene (*firmware*) äußerlich wie ein einziger SMP-Rechner mit $4 \cdot 4 \cdot 6$ Prozessorkernen verhalten. Dennoch weist dieses Konzept gegenüber „klassischen“ CISC-*servern*, etwa solchen vom Typ HP AM447A [84] mit $8 \cdot 10$ Kernen, bereits merklich erhöhte Antwortzeiten bei Speicherzugriffen außerhalb des Knotensegments auf; verstärkt gilt dies auch für andere Ansätze zur anwendungstransparenten Bereitstellung eines *cluster*-weiten Adreßraumes.

Aus Sicht des Algorithmus' der graphentheoretischen Störungsrechnung, wie dieser bereits in Referenz [19] beschrieben ist, überwiegen jedoch aufgrund der hochgradigen Datenunabhängigkeit und der geringen Interprozeßkommunikation die Vorteile eines MPP-Systems mit universellen Prozessoren – nämlich bis zu drei Größenordnungen mehr Rechenleistung – die Schwächen weitaus. Es ist zu erwarten, daß das in Abschnitt 6.3.4 vertieft betrachtete nahezu lineare Anwachsen der Verarbeitungsgeschwindigkeit mit der Größe symmetrischer Multiprozessorrechner auch zu einer außerordentlich guten Skalierung auf *clustern* führt. Daher erscheint die „Übersetzung“ der Parallelisierung von OpenMP nach MPI oder ein hybrider, zweistufiger Ansatz, welcher *grobgranulär* mit MPI „Auftragspakete“ an die Knoten verteilt und jeweils innerhalb der SMP-Knoten *feingranulär* mit OpenMP die Aufträge den *threads* zuweist, als vordringlicher Schritt, zumal erst dadurch, siehe Abschnitt 5.1.2, der Reihenkoeffizient der Ordnung U^{-17} für das Hubbardmodell zugänglich werden dürfte.

Weitere Laufzeitverkürzungen können durch die verstärkte Auslastung des Arbeitsspeichers kraft *thread*-eigener Kopien zentraler Datenstrukturen erreicht werden. Eine bereits oben erwähnte algorithmische Änderung mit beträchtlichen Auswirkungen auf den Ressourcenbedarf, die Verwendung von Hüpfsequenztypzählern nach Berechnung der Doppelbesetzungstypbeiträge, wird im folgenden Abschnitt behandelt. Obwohl SMP-Hochleistungsrechner, etwa der intensiv genutzte HP Superdome 2 [86] (Modelle AH353A und AH337A), aktuell bis zu 2048 MWords Hauptspeicher pro Prozessor-kern bereitstellen und die kommende Generation dieses Angebot um den Faktor 3 bis 6 übertreffen wird, und auch *cluster*-Knoten zwischen 128 und 512 MWords pro Kern anbieten, muß eine „Verlagerung von Rechen- in Speicheraufwand“ planvoll geschehen: Eine Überlastung der Kapazitäten der Puffer zwischen dem Hauptspeicher und den Prozessorregistern, welche typischerweise in speichernächster Instanz (*level-II* bzw. *level-III cache*) nur circa 4096 kWords pro Prozessor vorhalten können, kann durch *cache misses* und relativ lange Nachladezeiten zu deutlichen Leistungseinbußen führen.

Diese Überlegungen zur *Datenlokalität* gelten auch für Feinheiten der vorliegenden Implementierung: Um zeitintensive Speicherbereitstellungen zu vermeiden, werden beispielsweise für die Störungsrechnungen ab siebzehnter Ordnung¹⁷⁰ die dynamischen *integer*-Variablen der GMP, basierend auf ganzzahligen Vielfachen der Registerbreite (128, 192, ... Bit), mit einer stark überschätzten Anfangsbreite initialisiert; dieser Optimierung kann allerdings infolge einer auf etwa 2048 Worte beschränkten Kapazität der registernahen Puffer (*level-I cache*) starke Maschinenabhängigkeit entgegenstehen.¹⁷¹

¹⁷⁰Bis zur dreizehnten Ordnung sind 64 Bit-Ganzzahlen ausreichend, für die fünfzehnte Ordnung genügen 128 Bit-Ganzzahlen fester Breite, siehe Fußnote 123.

¹⁷¹Ein ähnlicher Effekt tritt prinzipiell auch bei Datentypumwandlungen (*type casts*) auf, wenn „kleine“ Ganzzahlen zwecks Speicherplatzersparnis zunächst in Variablen halber Registerbreite abgelegt, im Verfahrensablauf aber mit Zahlen voller Breite „verrechnet“ werden.

Aus diesem Grunde sollten alle Verbesserungen rechentechnischer Art stets von detaillierten Laufzeituntersuchungen mit entsprechenden Entwicklungswerkzeugen begleitet werden.

6.3.3. *Vorabberechnung der Energiebeiträge.* Bereits ohne vertiefte Analysen durch einen *profiler* ist aus der zusammenfassenden Betrachtung der Teilalgorithmen in den Abschnitten 3.4.9, 3.4.13 und 3.5.1 einsichtig, daß der wesentliche Anteil der Rechenzeit mit der Gewichtung einer erzeugten Hüpfsequenz durch die Prozesse der zugehörigen Klasse anhand Gleichung (3.100) oder (unter Verwendung der auf Ganzzahlen erweiterten Beiträge) Gleichung (5.10) verbraucht wird. Die „Systematik“ hinter den Doppelbesetzungsverläufen $\mathfrak{d}^{(n)}$, die bereits in Tabelle 3.9 angedeutet wird, legt nun die dem eigentlichen Hüpfen vorausgehende *typweise* Berechnung der Sequenzbeiträge zur Grundzustandsenergie nahe: Das Vorliegen sämtlicher \hat{S} -Exponentenverläufe $\rho^{(n)}$ und der zugehörigen Koeffizienten e in den Hüpfprozessen $\mathcal{P}^{(n)}$, siehe Abschnitt 3.3, erlaubt die vorgezogene Ausführung des mit einem Stern bezeichneten Teiles aus Gleichung (4.1),

$$a_n = -\frac{\gamma}{2^{(n+3)/2}} \frac{1}{n+1} \sum_{g=1}^{2^{(n-1)/2}} \sum_{h=1}^{|\mathfrak{H}_{g-1}^{(n)}|} s_h \sum_{i=1}^{|\hat{\mathfrak{P}}_{g-1}^{(n)}|} e_i \prod_{j=2}^{n+1} \frac{1}{d_j^{\kappa_j}} \quad ,$$

wodurch sich die Auswertung einer später ermittelten Hüpfsequenz $\mathfrak{H}^{(n)}$ auf die Bestimmung ihres Typs und die Erhöhung oder Erniedrigung des Typzählers anhand des Sequenzvorzeichens s reduziert.

Die Anzahl \mathfrak{d}_n der möglichen Abfolgen von Doppelbesetzungszahlen wurde in Abschnitt 3.4.7 anhand der innerhalb der Störungsrechnung n ter Ordnung generierten Sequenzen ermittelt und tabelliert, wodurch empirisch $\mathfrak{d}_n^{\text{HM}} = \mathfrak{d}_n^{\text{FKM}}$ gefunden wurde. Ein beispielsweise induktiver Beweis für beliebige n auf Grundlage der Hüpfsequenzen setzte allerdings voraus, die $\mathfrak{H}^{(n)}$ des jeweiligen Modells vollständig etwa aus den $\mathfrak{H}^{(n-i)}$ für $i = 2, 4, \dots, n-1$ erzeugen zu können; das entspräche jedoch einer Lösung des Gesamtproblems, da das hier beschriebene ordnungsweise systematische Ausprobieren unnötig würde. Aus den Hamiltonoperatoren (1.1) und (1.2) der beiden Modelle ist jedenfalls $\mathfrak{d}_n^{\text{HM}} \geq \mathfrak{d}_n^{\text{FKM}}$ ebenso wie $\mathfrak{H}^{(n),\text{HM}} \supseteq \mathfrak{H}^{(n),\text{FKM}}$ erkennbar.

Daß tatsächlich für das Hubbard- und das Falicov-Kimball-Modell innerhalb einer Ordnung die gleichen Doppelbesetzungsverläufe existieren, wird aber dann sofort einsichtig, wenn auf die *modellunabhängigen* Operatoren \hat{T}_0 , \hat{T}_{+1} und \hat{T}_{-1} aus Gleichungen (1.29) zurückgegriffen wird. So muß sich für jedes im Rahmen der diesbezüglichen Randbedingungen, die ausführlich im Abschnitt 1.1 des Anhanges behandelt werden, mögliche Operatortupel aus $n+1$ Elementen auf einer beliebigen Baumstruktur der

Ordnung $k = (n + 3)/2$ mindestens eine Sequenz mit nur einer mobile Spinspezies realisieren lassen. Folgende Untersuchung soll dies kurz für zwei Fälle anhand von Ausgangszuständen $\mathcal{B}_{k,1,m'} \in \mathcal{B}'_{k,1}$ zeigen, die auch für andere Untersuchungen in einer Dimension interessant werden können; dazu betrachten wir in Anlehnung an Abschnitt 5.2.1 zunächst den „minimalen“ und den „maximalen“ \hat{T} -Operatorverlauf.

Die kleinstmögliche Anzahl an \hat{T}_{+1} und \hat{T}_{-1} , welche die Doppelbesetzungszahl erhöhen und erniedrigen, folglich die größtmögliche an \hat{T}_0 , welche diese konstant halten, ergibt sich stets für die zur Operatorklasse $f = 0$ gehörigen Verläufe, siehe Tabelle 3.5, allgemein

$$\begin{aligned} \mathfrak{d} &= (0) \circ (1 \mid i = 1, 2, \dots, n) \circ (0) \\ &\mapsto (\hat{T}_{+1}) \circ (\hat{T}_0 \mid i = 1, 2, \dots, n - 1) \circ (\hat{T}_{-1}) . \end{aligned} \quad (6.2)$$

Zum größtmöglichen Auftreten der Operatoren \hat{T}_{+1} und \hat{T}_{-1} , gleichbedeutend mit der höchstmöglichen Doppelbesetzungszahl $\lfloor k/2 \rfloor$, sei auf die Gleichungen (5.7) verweisen; diesen führen für gerade Werte von k auf

$$\begin{aligned} &(\hat{T}_{+1} \mid i = 1, 2, \dots, (n + 3)/4) \circ ((\hat{T}_{-1}, \hat{T}_{+1}) \mid i = 1, 2, \dots, (n - 1)/4) \circ \\ &(\hat{T}_{-1} \mid i = 1, 2, \dots, (n + 3)/4) , \end{aligned} \quad (6.3)$$

für ungerade auf

$$\begin{aligned} &(\hat{T}_{+1} \mid i = 1, 2, \dots, (n + 1)/4) \circ (\hat{T}_0 \mid i = 1, 2, \dots, (n + 1)/2) \circ \\ &(\hat{T}_{-1} \mid i = 1, 2, \dots, (n + 1)/4) . \end{aligned} \quad (6.4)$$

Aus diesen Überlegungen lassen sich für die Extremfälle leicht passende Färbungen, also Spinbelegungen der Kette konstruieren und Hüpfsequenzen darauf für das Falicov-Kimball-Model herleiten: Fall (6.2) legt $F(\mathcal{B}_{k,1,m'}) = (1) \circ (2 \mid 1, 2, \dots, k - 1)$ nahe, wie analog zu Gleichungszeile (3.103-5) beispielsweise für $k = 4$ mit $\mathcal{B}_{4,1,107}$

$$\begin{array}{cccccccc} \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{3,\uparrow}^+ \hat{c}_{2,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{4,\uparrow}^+ \hat{c}_{3,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{3,\uparrow}^+ \hat{c}_{4,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{2,\uparrow}^+ \hat{c}_{3,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{1,\uparrow}^+ \hat{c}_{2,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} \\ \hat{T}_{+1} & & \hat{T}_0 & & \hat{T}_{-1} & \end{array} \quad (6.5)$$

zeigt; die Fälle (6.3) und (6.4) führen auf $F(\mathcal{B}_{k,1,m'}) = (1) \circ ((2, 1) \mid 1, 2, \dots, (k - 2)/2) \circ (2)$ beziehungsweise $F(\mathcal{B}_{k,1,m'}) = (1) \circ ((2, 1) \mid 1, 2, \dots, (k - 1)/2)$, wobei ersterer analog zu Gleichungszeile (3.103-4) etwa mit $\mathcal{B}_{4,1,103}$ durch

$$\begin{array}{cccccccc} \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{2,\uparrow}^+ \hat{c}_{1,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{4,\uparrow}^+ \hat{c}_{3,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{3,\uparrow}^+ \hat{c}_{2,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{2,\uparrow}^+ \hat{c}_{3,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{3,\uparrow}^+ \hat{c}_{4,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} & \xrightarrow{\hat{c}_{1,\uparrow}^+ \hat{c}_{2,\uparrow}} & \begin{array}{|c|} \hline \bullet \\ \bullet \\ \bullet \\ \bullet \\ \hline \end{array} \\ \hat{T}_{+1} & & \hat{T}_{+1} & & \hat{T}_{-1} & & \hat{T}_{+1} & & \hat{T}_{-1} & & \hat{T}_{-1} & & \hat{T}_{-1} & \end{array} \quad (6.6)$$

verdeutlicht wird.¹⁷² Leicht ist zu überprüfen, daß die Sequenz in Gleichung (6.5) die einzig mögliche für den gegebenen Ausgangszustand ist; für den Butcherbaum in Gleichung (6.6) hingegen existieren im Falicov-Kimball- beziehungsweise Hubbardmodell 6 respektive 56, wovon 2 respektive 4 dem Hüpfsequenztyp des Beispiels angehören.

n	1	3	5	7	9	11	13	15	17	19
d'_n	1	2	10	59	397	2878	21266	162732	1253128	9839212
d''_n	1	2	8	49	335	2473	18452	142297	*	*
Sb.	1	16	256	$\underbrace{4 \quad 64}_{\cdot 2^{10}}$		$\underbrace{1 \quad 16 \quad 256}_{\cdot 2^{20}}$			$\underbrace{4 \quad 64}_{\cdot 2^{30}}$	

Tabelle 6.1. Berechnete Gesamtzahl d'_n der möglichen Doppelbesetzungsverläufe aller Hüpfsequenzklassen [24:A240605] bis einschließlich Störungsordnung $n = 19$. Die Übereinstimmung mit den bei Programmausführung „gezählten“ Werten d_n aus Tabelle 3.10 ist bis einschließlich $n = 15$ gegeben. In der Zeile Sb. ist mit 4^{n-1} der (ungefähre) Speicherbedarf in Worten der jeweiligen Datentypbreite (2^{10} : kWorte, 2^{20} : MWorte, 2^{30} : GWorte) für die zugehörigen Hüpfsequenztypbeträge und -zähler verzeichnet, weiteres siehe Text. Zeile d''_n gibt, vorgehend auf Abschnitt 1.1 des Anhanges, die Anzahl jener Doppelbesetzungsverläufe an, deren Typbeitrag zur Grundzustandsenergie von null verschieden ist. Ein Stern vertritt dabei Werte, welche aufgrund der außerordentlich hohen Rechenzeitanforderungen noch nicht ermittelt werden konnten.

Ein Ausweiten dieser Untersuchung auf die anderen möglichen Doppelbesetzungsverläufe bestätigt in niedriger Ordnung schnell, daß sich im Sinne der Modellunabhängigkeit der \hat{T} -Operatorabfolgen stets Färbungen für Falicov-Kimball-Sequenzen finden.

¹⁷²Die Färbungsnummern m' dieser zweifarbigen Butcherbäume der Struktur $l = 1$ innerhalb der ${}^4\mathcal{B}_{k,l}$ kann für beliebiges k jeweils berechnet werden. Für den zu Gleichung (6.2) gehörigen Fall einer bis auf die \uparrow -Spin-Wurzel nur mit \downarrow -Spins besetzten Kette gilt

$$m' = \frac{4^k}{2} - x_{k-1} = \frac{5 \cdot 4^k + 4}{12},$$

worin $x_{k'}$ das k' te Folgeglied der Ganzzahlfolge [24:A002450] ist. Die zu den Gleichungen (6.3) und (6.4) gehörenden Ausgangszustände aus einer bezüglich des „Gitterplatzspins“ alternierenden Kette mit einem \uparrow -Spin auf der Wurzel haben den Index

$$m' = \frac{4^k}{2} - \psi_{k-1} = \frac{4^{k+1} + (-1)^k + 5}{10};$$

dabei ist $\psi_{k'}$ als k' tes Folgeglied der Ganzzahlfolge [24:A037481] gegeben. Diese Möglichkeit, innerhalb der Ausgangszustände auf die beiden Spezialfälle direkt zugreifen zu können, stellt auch einen kleinen datentechnischen Vorteil bei einer künftigen Anwendung auf das eindimensionale Hubbardmodell dar.

Während den Rahmen dieser Arbeit eine mathematische Klärung der Frage im allgemeinen übersteigt, kann nun auf Grundlage dieser Betrachtung und der „Gesetzmäßigkeiten“ hinter der gesamten graphentheoretischen Störungsrechnung jedoch auch dafür argumentiert werden, daß zu jedem möglichen \hat{T} -Operatorverlauf eine Hüpfsequenz mit nur einer mobilen Spinspezies existiert, somit die empirisch ermittelte Doppelbesetzungsverlaufsanzahl d_n und die berechnete, d'_n , siehe Tabelle 6.1 [24:A240605], übereinstimmen.

Abschnitt 1.1 des Anhangs behandelt umfassend ein im Sinne der eingangs durchgeführten Betrachtung modifiziertes Verfahren zur Berechnung des Reihenkoeffizienten. Aus rechentechnischer Sicht wird dabei zugunsten bedeutend verkürzter Laufzeit ein erheblicher Teil des Aufwandes „vom Prozessor in den Arbeitsspeicher verschoben“: Die in der Liste \mathcal{W} zentral vorgehaltenen Sequenztypbeiträge benötigen, siehe auch Zeile Sb. in Tabelle 6.1, in n ter Ordnung 4^{n-1} Datenworte mit zur Anzahl der Dualstellen der Ganzzahlen passender Breite; die im Idealfall für jeden Rechenprozeßfaden¹⁷³ lokal angelegten Sequenztypzähler in der Liste \mathcal{U} verlangen ebenfalls 4^{n-1} Worte mit Adreßbusbreite und zudem pro berechnetem Modell d'_n Worte mit Registerbreite. Werden für die Typbeiträge \mathcal{W} GMP-Ganzzahlen mit (initial) 256 Bit genutzt, ergibt sich folglich in fünfzehnter beziehungsweise siebzehnter Störungsrechnungsordnung ein Bedarf von mindestens 1 respektive 16 GWorte zu 64 Bit, folglich 8 respektive 128 GByte Hauptspeicher. Bedeutend größer sind aber beispielsweise bei 240 *threads* die Anforderungen für die Zähler in \mathcal{U} , weil unter Verwendung einer vorzeichenbehafteten Ganzzahl zu 63 Bit für jeden Doppelbesetzungsverlaufstyp und jedes der beiden Modell bei einer Adreß- und Registerbreite von 64 Bit insgesamt etwa 60,1 respektive 961 GWorte, entsprechend 481 GByte respektive 7,51 TByte notwendig werden. Dennoch ist die Umsetzung eines solchen Verfahrens durchaus realistisch, da die derzeit vor Marktverfügbarkeit stehende Generation der *SMP-server* prinzipiell einen Speicherausbau bis 3072 GWorte (12,8 GWorte/Kern), also 24 TByte zulassen [90].

Eine Konzeptprüfung dieses Ansatzes ist bereits unter anderem auf einem EPIC-Großrechner des Typs HP AH337A (128 Prozessorkerne, 192 GWorte Hauptspeicher, sonst identisch mit HP AH353A [86]) erfolgt; dem Prinzip nach entsprechend erstrecken sich (Schreib)operationen auf der großen Datenstruktur \mathcal{U} nun fast über den gesamten Hauptspeicher, wodurch der Vorteil der kurzen Latenzen Register-naher Puffer entfällt. Ohne weitergehende Bemühungen zum Ausgleich dieses Nachteils zeigt sich eine Laufzeitverkürzung für die Berechnung des Störungsreihenkoeffizienten des Hubbardmodells der Ordnung U^{-13} auf nur 32,7 % und des Falicov-Kimball-Modells der Ordnung U^{-15}

¹⁷³Teilen sich alle *threads* eine Sequenztypzählerliste, ist der Synchronisationsaufwand vor dem nur seriell möglichen Schreiben darauf zu berücksichtigen. Dieses mit der *thread*-Anzahl nicht-linear anwachsende Problem läßt sich in Abwägung mit dem Speicherbedarf durch die gemeinsame Nutzung von „semi-lokalen“ Listen durch eine Rechenprozeßfaden- oder Prozessorgruppe optimieren – allerdings unter Inkaufnahme des deutlich höheren Programmieraufwandes für eine Hybridparallelisierung. Im Falle einer Portierung des Verfahrens auf *cluster* aus kleineren *SMP*-Rechnern (mit beispielsweise jeweils zwei Prozessoren à 8 bis 15 Kernen) entstünde dieser Aufwand durch einen gemischten Ansatz mit *OpenMP* und *MPI* jedoch ohnehin.

auf 33,3 % des bisherigen Zeitaufwandes. Ursächlich ist dafür die Reduktion der Komplexität des in Abschnitt 3.5 erläuterten Verfahrens für die Sequenzbeitragsermittlung von $\mathcal{O}(e^n)$ auf $\mathcal{O}(n^2)$; dieser „Rest“ ist durch das nun dominierende Verfahren zur Ermittlung des Beitragsvorzeichens, Flußdiagrammelement (3) in Abbildung 3.10 bedingt.

Zu beachten sind bei der Bewertung dieses Ergebnisses Details der Systemarchitektur, die sich hinsichtlich der Speicherzugriffe vorteilhaft auswirken: Im Gegensatz zu ähnlichen Maschinen mit einem reinen *non-uniform memory access*, bei welchen jeder Prozessor lokal einen Teil des gesamten Hauptspeichers verwaltet, diesen gemäß *shared-memory*-Konzept aber auch allen andere Prozessoren mit entsprechend verlängerten Latenzen und verringerten Bandbreiten bereitstellt, erlaubt dieser Rechnertyp ergänzend die bevorzugte Nutzung von 7/8 des Prozessor-lokalen Arbeitsspeichers für *threads* auf den Prozessor-eignen Kernen. Damit ergibt sich für dieses Verfahren eine adäquate Auslastung der verfügbaren 1344 MWords eines Kernes durch zwei Rechenprozeßfäden¹⁷⁴, welche zusammen 513 MWords beanspruchen.

6.3.4. Skalierungsverhalten und Amdahlsches Gesetz. Dank der Bereitstellung der im Jahre 2013 größten SMP-server der EPIC-Architektur¹⁷⁵ durch Hewlett-Packard mit dem schnellsten verfügbaren Prozessor, Typ Intel Itanium 9560, konnten abschließend die erforderlichen umfangreichen und folglich zeitaufwendigen Untersuchungen zum Skalierungsverhalten des Algorithmus' in Abhängigkeit von der Prozessoranzahl durchgeführt werden. Aufgrund der üppigen Ressourcen war dabei die vergleichende Prüfung zweier Entwicklungsstufen möglich: Die Programmversion der „Produktionsrechnungen“ (I), welche sämtliche Optimierungen aus Abschnitt 5.2 einschließt, sowie jene Fassung (II), die zudem auch den Entwicklungsschritt aus Abschnitt 6.3.3 berücksichtigt. Tabelle 6.2 enthält die Meßdaten (jeweils bester Wert aus drei Läufen) für den rechenzeitbestimmenden Hauptabschnitt des Verfahrens, die Ausgangszustandssequenzierung; in Abbildung 6.2 sind diese veranschaulicht.

Neben deren Nutzen für die Aufwands- und Anforderungsabschätzung für künftige Projekte, berühren diese Ergebnisse unmittelbar das sogenannte *Amdahlsche Gesetz*. Diese von GENE AMDAHL 1967 vorgestellte Überlegung [74] besagt, daß bei einem Anteil p parallelisierbaren Aufwandes am Gesamtaufwand eines Datenverarbeitungsproblems bei Verwendung von n Prozessoren allenfalls der Leistungszuwachs (*speed-up*)

$$s_p(n) = \left(1 - p + \frac{p}{n}\right)^{-1} \quad (6.7)$$

¹⁷⁴Nutzung der in Fußnote 130 erläuterten HyperThreading-Technologie, welche sich hier im Gegensatz zu den Ausführungen in Abschnitt 5.1.3 mit bis zu 6,1 % Leistungsgewinn vorteilhaft bemerkbar macht

¹⁷⁵Auf Grundlage der CISc-Architektur existieren zwar SMP-Parallelrechner mit insgesamt mehr *Prozessorkernen* vergleichbarer Verarbeitungsgeschwindigkeit und Speicherbandbreite, jedoch leistungsschwächeren Verbindungen zwischen den *Prozessorsockeln* einerseits und zum (generell verteilt organisierten) Hauptspeicher andererseits, beides mit der Folge höherer Antwortzeiten.

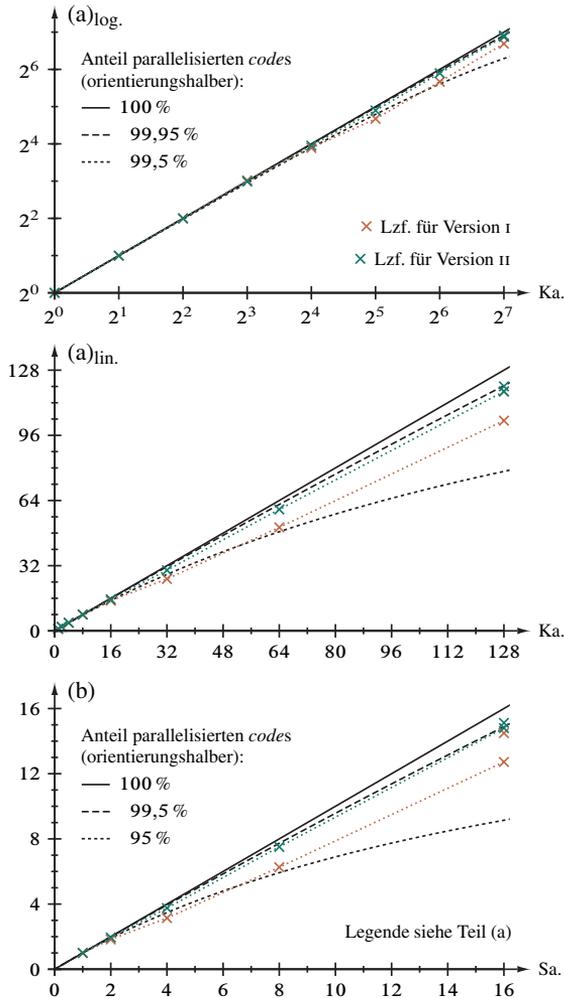


Abbildung 6.2. Leistungsanzwachsungs faktor (Lzf.) bei der Sequenzierung der Ausgangszustände, siehe Abschnitt 3.4.9, in Abhängigkeit von der Prozessoranzahl im Vergleich mit dem Amdahlschen Gesetz. Die schwarzen Referenzlinien (durchgezogen und gestrichelt) ergeben sich aus Gleichung (6.7) für die angegebenen Werte für p . Die dortige Prozessoranzahl n bezieht sich in Abbildungsteil (a) auf Prozessorkerne (Ka.) und in Teil (b) auf Prozessorsocket (Sa.). Die Datengrundlage stammt konsistent von einem EPIC-server Typ HP AH337A [86] anhand der Berechnung des Falicov-Kimball-Modells zu $\mathcal{O}(U^{-15})$. Der obere der jeweils zwei Datenpunkte gleicher Farbe für 128 Kerne beziehungsweise 16 Sockel ergibt sich aus der Verwendung von HyperThreading.

Pa.	Version I			Version II			Q.
	Mz./s	Lzf.		Mz./s	Lzf.		
		Kerne	Socket		Kerne	Socket	
1	314000	1		107000	1		0,340
2	158000	1,99		53400	2,00		0,339
4	78200	4,01		26800	3,98		0,343
8	38700	8,11	1	13400	7,94	1	0,348
16	21100	14,8	1,83	6900	15,5	1,95	0,326
32	12300	25,5	3,14	3580	29,8	3,75	0,291
64	6170	50,8	6,26	1790	59,6	7,51	0,290
128	3040	103	12,7	908	118	14,8	0,299
256	2670	118	14,5	889	120	15,1	0,333
p		0,9979	0,981		0,9993	0,994	

Tabelle 6.2. Leistungszuwachsfaktor (Lzf.) bei der Sequenzierung der Ausgangszustände, siehe Abschnitt 3.4.9, in Abhängigkeit von der Prozeßfadenanzahl (Pa.). Sämtliche Laufzeiten („Maschinenzeit“ (Mz.), bestes Ergebnis aus drei Läufen) stammen von einem EPIC-server des Typs HP AH337A [86] mit 16 Prozessorsockeln zu jeweils acht Kernen (siehe Text) und wurden anhand der Berechnung des Falicov-Kimball-Modells zu $\mathcal{O}(U^{-15})$ ermittelt. Der Wert für 256 Prozeßfäden (zwei pro Prozessorkern) ergibt sich unter Verwendung der HyperThreading-Technik. Die Angaben für den Parameter p in der letzten Tabellenzeile wurden mit bis zu 128 Fäden durch lineare Ausgleichsrechnung anhand von Gleichung (6.7) gewonnen. Der Quotient (Q.) in der letzten Spalte gibt das Laufzeitverhältnis von Version II zu I an. Durch den Maschinenaufbau aus insgesamt acht *blades* ist die sprunghafte „Verschlechterung“ des Leistungszuwachsfaktors – geringer zunächst ab 16 (zwei Prozessoren), stärker dann ab 32 Prozeßfäden (zwei Doppelprozessorsteckkarten) – bedingt.

erzielt werden kann. Dabei wird noch kein zusätzlich dämpfender Synchronisationsaufwand berücksichtigt, welcher sich in problemabhängigem Maße zwischen den Rechenprozessen (*threads*) ergibt. Bereits im Jahrzehnt nach seiner Einführung galt dennoch das Amdahlsche Gesetz als pessimistische Abschätzung, weil einige die Skalierung begünstigende Konstruktionsmerkmale neuerer Prozessoren [47, 53], insbesondere Pufferinstanzen zwischen den Registern des Rechenwerks und dem Daten- und Anweisungsspeicher nicht in Gleichung (6.7) eingehen.

AMDAHLS Gesetz selbst und seine Verbesserungen werden bis heute in sehr vielen Veröffentlichungen diskutiert, und das Interesse daran hat sich insbesondere durch die leichte Verfügbarkeit „kleiner Parallelrechner“ in Form von Prozessoren, welche auf einem *Prozessorsockel* mehrere *Prozessorkerne* tragen, verstärkt. Zwar wirken nämlich oberhalb der *hardware*-Abstraktionsebene alle Kerne wie eigenständige Prozessoren, wodurch sich bei Mehrprozessorsystemen die Prozessorgesamtzahl als Produkt der Steckfassungsanzahl und der Kernanzahl pro Fassung ergibt; auf Schaltkreisebene jedoch teilen sich mehrere physische Kerne Pufferspeicherinstanzen und den Zugang zum Hauptspeicher oder im Falle der logischen Kerne der HyperThreading-Technik darüberhinaus die Recheneinheiten. Gleiches gilt für den Hauptspeicher, der aus Sicht des Betriebssystems einheitlich und linear adressierbar scheint, in *hardware* sich aber als Summe des von jedem Sockel lokal verwalteten Speichers ergibt.¹⁷⁶ Die Konsequenz für das Amdahlsche Gesetz besteht nun wiederum in einer „Störung“ der vorausgesetzten Unabhängigkeit der Rechenprozesse durch gemeinsam genutzte, aber zum Teil getrennt organisierte Schaltungslogik, wobei die Fülle der technischen Verbesserungen insgesamt auf die parallele Ausführung begünstigend wirkt.

Im Einklang mit Leistungsmessungen durch andere parallelisierte *benchmarks* in diesem Zusammenhang läßt sich nun am zentralen Teilalgorithmus der graphentheoretischen Störungsrechnung beobachten, daß eine Beurteilung der Parallelisierungseffizienz auf Basis der Kernanzahl den Anteil p an parallel ausführbarem *code* überschätzt, vergleiche Tabelle 6.2 und Teil (a) in Abbildung 6.2: Zwar ist eine Abschätzung für dieses im Prinzip ideal-datenparallele Problem außerordentlich schwierig, aber vor allem durch die sukzessive Auftragsvergabe an die *threads* durch den *scheduler*¹⁷⁷ verbleiben für das Falicov-Kimball-Modell der Ordnung U^{-15} vermutlich 0,5...5% reihenfolgegebundene Instruktionen.¹⁷⁸ Wird hingegen die Sockelanzahl zugrundegelegt, siehe Teil (b), erscheint der Gesamteindruck realistischer; dadurch kann anhand dieser Implementierung ebenfalls nachgewiesen werden, daß GENE AMDAHL'S Gesetz – trotz seiner Unzulänglichkeiten und nur durch Kompensationen – nun für n jeweils aus mehreren Prozessorkernen und diversen *cache*-Instanzen zusammengesetzte Logikeinheiten, also

¹⁷⁶Auf die resultierenden Wartezeiten (Latenzen) beim Zugriff auf „fremden“, also nicht am eigenen Sockel eines Prozessorkernes befindlichen Arbeitspeicher, welcher ja *kohärent* zum Inhalt der *caches* anderer Prozessorkerne gehalten muß (CC-NUMA), wurde bereits im Abschnitt 5.1 eingegangen.

¹⁷⁷Einige explizit serielle Protokollausgaben auf dem Standardkanal und entsprechende Synchronisationen wirken sich hier nicht signifikant aus und können bedarfsweise bereits bei der Kompilierung deaktiviert werden.

¹⁷⁸Dieser Anteil liegt beim Hubbardmodell niedriger und sinkt generell mit zunehmender Störungsrechenungsordnung, siehe Abschnitt 5.2.2. Man beachte auch, daß bei einer Hybridparallelisierung mit OpenMP und MPI für *cluster*-basierte Anlagen die Verhältnisse noch günstiger werden, weil die Verteilung der gegebenenfalls zuvor randomisierten Ausgangszustände (im Sinne von Abschnitt 5.2.2) von einem „*master dispatcher*“ nur zu Beginn der Rechnung an die Rechenknoten erfolgen muß und im weiteren Verlauf der jeweils dort lokalisierte „*slave dispatcher*“ die (dann relativ wenigen) lokalen Prozeßfäden mit Aufträgen beschickt. Ein solcher Ansatz ließe sich mit zwei OpenMP-Ebenen gleichfalls, sofern künftig verfügbar, auf SMP-Systeme mit mehreren tausend Prozessorkernen übertragen.

Prozessoren in neuerem Sinne, noch immer gilt. Kraft der nahezu vollständigen Systembeanspruchung bezüglich der Prozessoren und Datenbusse und insbesondere die hohe Auslastung des Hauptspeichers durch die typweise Vorabberechnung der Energiebeiträge (Version II), wird dieses Rechenverfahren derweil selbst als *benchmark* für Anlagen der neuen Generation genutzt, wofür sich aufgrund der oftmals relativ geringen Speichernutzung durch polynomielle Rechenzeitkomplexitäten numerischer Methoden nur wenige konkrete wissenschaftliche Anwendungen eignen.

Anhang: Ergänzungen zum Rechenverfahren

I. Höhere Algorithmen

Bereits im Rahmen dieser Arbeit wurden für künftige Anwendungen und weitergehende Optimierungen einige Verfahren entwickelt, die als Hilfsfunktionen den in Kapitel 3 vorgestellten Hauptprogrammteilen untergeordnet sind. Exemplarisch werden im folgenden zwei miteinander verbundene Algorithmen näher erläutert, welche schon im Ausblick, Kapitel 6, Erwähnung gefunden haben.

I.1. Erzeugung aller Doppelbesetzungsverlaufstypen. Den zuerst in Abschnitt 3.4.7 beschriebenen und erneut umfassend in Abschnitt 5.2.1 behandelten Doppelbesetzungsverläufen liegen nach ihrer an der oben befindlichen „Nullinie“ gespiegelten graphischen Darstellung, in $\mathcal{O}(U^{-5})$ beispielsweise¹



folgende Eigenschaften zugrunde:

- (1) Jeder Doppelbesetzungsverlaufspfad hat in n ter Störungsrechnungsordnung beziehungsweise k ter Baumordnung eine Länge von $n + 1 = 2k - 2$ Segmenten oder Schritten, die der Anzahl der in einer Hüpfsequenz zusammengefaßten Zustände abzüglich des Ausgangszustandes entspricht.
- (2) Jeder Pfad beginnt auf der Nullebene, da dem hier behandelten Problem *anregungsfreie Ausgangszustände* mit der Doppelbesetzungsanzahl $d = 0$ zugrundeliegen.
- (3) Die Ebene ändert sich nach einem Schritt höchstes um eine Einheit – korrespondierend mit den in Gleichungen (1.29) definierten Operatoren \hat{T}_0 , \hat{T}_{+1} und \hat{T}_{-1} .
- (4) Die maximale Ebene ist $d' = \lfloor k/2 \rfloor$, was aus der *Gitterhalbfüllung*, demgemäß einem Elektron pro Butcherbaumecke, sowie Punkt (2) folgt.
- (5) Zur minimalen Ebene, also $d = 0$ kann nur nach einer geraden Anzahl an Schritten zurückgekehrt werden. Auch diese Forderung ergibt sich aus der *Gitterhalbfüllung* und Punkt (2).
- (6) Falls k gerade ist, können infolge der *Halbfüllung* nicht hintereinander zwei oder mehr Schritte auf Ebene d' verlaufen, es existieren also keine Plateaux. Ebenso

¹Verlauf $(0, 1, 1, 1, 0, 1, 0) = d^{(5)} = \llbracket \mathcal{H}^{(5)} \rrbracket_2$, folglich $(\hat{T}_{+1}, \hat{T}_0, \hat{T}_0, \hat{T}_{-1}, \hat{T}_{+1}, \hat{T}_{-1}) \leftarrow \mathfrak{x}^{(5)}$; Hüpfsequenzvorzeichen $\llbracket \mathcal{H}^{(5)} \rrbracket_1 = s \in \{-1, +1\}$ beliebig

müssen zwischen zwei- oder mehrmaligem Erreichen der maximalen Ebene d' eine gerade Anzahl an Schritten liegen. Folglich gilt für $k = 6$ beispielsweise:



Bei graphentheoretisch-abstrahierter Behandlung der Pfade² kommt die physikalisch „triviale“ Anforderung $d \in \mathbb{N}$ hinzu, sodaß niemals das Nullniveau unterschritten werden darf.

Mit dieser Betrachtung der Doppelbesetzungsverläufe lenken wir den Blick von der jeweils erreichten Ebene im Pfadgraphen auf die Richtung des Schrittes, sozusagen „die erste Ableitung der Ebene nach dem Schritt“, folglich auf die darunterliegenden \hat{T} -Operatoren. Neben den vielfältigen Möglichkeiten der Darstellung des Verlaufes der Doppelbesetzungsanzahl durch Graphen, Referenz [48] gibt dazu eine Übersicht, lassen sich nun die Sequenzen aus \hat{T}_0 , \hat{T}_{+1} und \hat{T}_{-1} naheliegend als Zeichenketten kodieren³, worauf auch das folgende Verfahren indirekt zurückgreift.

Zunächst ersetzen wir dazu einen jeden der Operatoren \hat{T}_i durch die Ganzzahl seines Index⁴ aus dem Intervall $\{i \in \mathbb{Z} \mid -1 \leq i \leq +1\}$, entsprechend $\hat{T}_{-1} \mapsto -1$, $\hat{T}_0 \mapsto 0$ und $\hat{T}_{+1} \mapsto +1$; diese können wir wiederum durch $\lceil \log_2 |\{-1, 0, +1\}| \rceil = 2$ Dualstellen darstellen, der rechten für die Zahl $|i| \in \{0, 1\}$ selbst, der linken für deren Vorzeichen $\text{sgn}(i) \in \{-1, +1\}$. Damit erhalten wir $\hat{T}_{-1} \mapsto 11_2 = 3_4$, $\hat{T}_0 \mapsto 00_2 = 0_4$ und $\hat{T}_{+1} \mapsto 01_2 = 1_4$, wobei die verbleibenden Kombination $10_2 = 2_4$, formal -0 , datentechnisch dem Konzept „keine Zahl“ entspreche. Auf dieser Grundlage ist es uns nun möglich, eine Sequenz aus $n + 1$ \hat{T} -Operatoren für die Ordnung U^{-n} in $n + 1$ Zweierblöcken, somit insgesamt $2^{2(n+1)} = 4^{n+1}$ Binärziffern abzulegen. Berücksichtigen wir ausdrücklich den Halbfüllungsfall, in welchem jede Operatorabfolge mit \hat{T}_{+1} beginnt und mit \hat{T}_{-1} endet, reduziert sich der Platzbedarf auf 4^{n-1} Binärziffern. Alle möglichen Sequenzen sind dementsprechend im Ganzzahlintervall $[0, 4^{n-1} - 1] \subset \mathbb{N}$ enthalten und nur noch anhand der oben angeführten Kriterien einschließlich der undefiniertheit von 10_2 „herauszusieben“.

²Deren Anzahl wurde mit Hilfe *dynamischer Programmierung* berechnet [75] und als Folge A240605 in Referenz [24] publiziert; bis einschließlich $k = 17$ lautet diese

1, 2, 10, 59, 397, 2878, 21266, 162732, 1253128, 9839212, 77644825,
620377508, 4981522538, 40351448045, 328421827064, 2690586461296 .

Die Berechnung dieser Ganzzahlsequenz erfolgt durch Zählen der gültigen Doppelbesetzungsverlaufspfade jeweiliger Ordnung, diese werden also explizit erzeugt. Dafür wurde eine spezielle Parallelisierung der *greedy*-Variante III des in diesem Abschnitt beschriebenen Algorithmus⁵, siehe Fußnote 7, auf einem EPIC-server des Typs HP AH337A verwendet. Weiteren Gliedern steht neben dem immensen Zeitbedarf (bereits 32,5 Stunden auf 128 Prozessorkernen für $k = 17$) die *hardware*-seitige Beschränkung auf eine Datenwortbreite von 64 Bit entgegen, die aus technischen Gründen mit dem letzten angegebenen Wert ausgeschöpft ist.

³Auf die Darstellung als *strings* greifen beispielsweise viele Einträge zu den Catalan- und Motzkinzahlen in der OEIS-Datenbank [24] zurück.

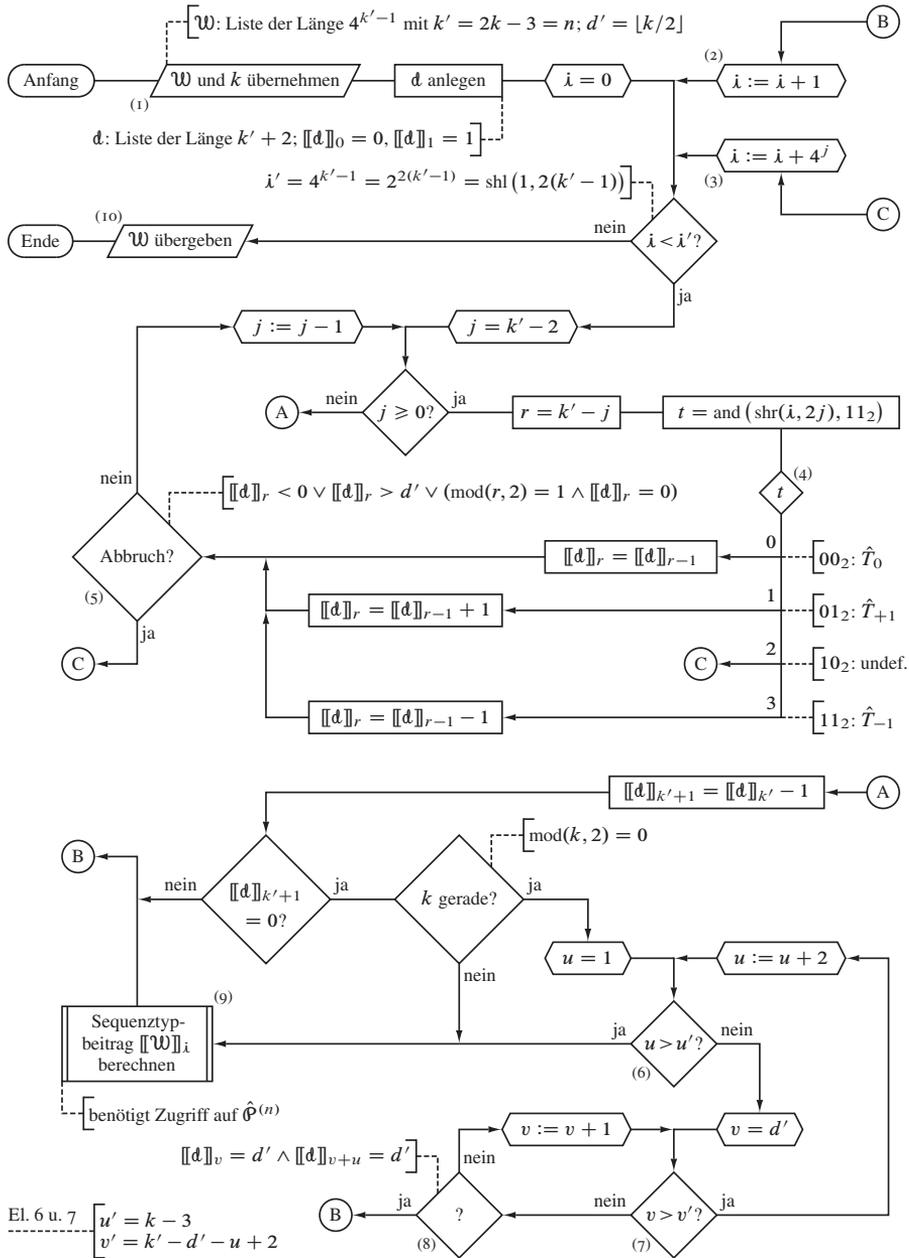


Abbildung 1.1. Ablaufplan eines Unterprogramms für die Generierung der Doppelbesetzungsverlaufstypen. Die Schleifenabbrüche mit nachfolgenden Übergängen von C nach C führen auf den Algorithmustyp "greedy"; wird von C nach B gesprungen, entsteht der Typ "brute force".

Mit der in Abschnitt 6.3.3 beschriebenen Zielsetzung, die möglichen verschiedenen Beiträge zum Störungsreihenkoeffizienten a_n nach dem *Typ der jeweils zugrundeliegenden Hüpfsequenz* vorab zu berechnen, muß ein Unterprogramm im Hauptablaufschema in Abbildung 3.1 unmittelbar nach dem Element (2) eingefügt werden. Voraussetzung für den nachfolgend formuliertem Algorithmus mit der Laufzeit- und Speicherkomplexität $\mathcal{O}(4^n)$ ist die Bereitstellung einer Liste \mathcal{W} mit der Länge 4^{n-1} für die Typbeiträge⁴; „Adresse“ eines jeden Eintrages ist die sich aus dem *bit*-Muster der zum Sequenztyp gehörigen \hat{T} -Operatorabfolge ergebende ganze Zahl. Dieses Vorgehen hat zwar zur Konsequenz, daß die meisten Listenplätze von \mathcal{W} , wie in der vierten Zeile von Tabelle 1.1 quantifiziert wird, unbesetzt bleiben, ermöglicht aber einen schnellen direkten⁵ Zugriff durch die anhand des im folgenden Abschnitt 1.2 vorgestellten Verfahrens *berechenbare Adresse*⁶, vergleiche dazu auch Abschnitt 3.3 in diesem Anhang.

- 1 Graphenordnung k als Ausführungsparameter und Liste für \mathcal{W} vom aufrufenden Hauptprogramm übernehmen, dabei Hilfsgrößen $k' = 2k - 3$ (entspricht der Störungsrechnungsordnung n) und $d' = \lfloor k/2 \rfloor$ berechnen, siehe Element (1) des Flußdiagrammes in Abbildung 1.1
- 2 Hilfsliste d für die Doppelbesetzungszahlen entsprechend eingangs definiertem Kriterium (1) anlegen; $\llbracket d \rrbracket_0 = 0$ und $\llbracket d \rrbracket_1 = 1$ gemäß Kriterium (2) initialisieren
- 3 Schleife für $i = 0, 1, \dots, 4^{k'-1} - 1$:
 - 3.1 Schleife für $j = k' - 2, k' - 3, \dots, 0$:
 - 3.1.1 Abkürzend Adresse der aktuellen Doppelbesetzungszahl $r = k' - j$ berechnen
 - 3.1.2 \hat{T} -Operatoräquivalent $t = \text{and}(\text{shr}(i, 2j), 11_2)$ durch „Extraktion“ des j ten zweistelligen Blockes aus i bestimmen
 - 3.1.3 \hat{T} -Operator dekodieren, Flußdiagrammelement (4), und nach Kriterium (3) ausführen:

$$t = 00_2 = 0_4, \text{ entspricht } \hat{T}_0: \text{ unveränderte Übernahme der Doppelbesetzungszahl des vorherigen Zustandes durch } \llbracket d \rrbracket_r = \llbracket d \rrbracket_{r-1}$$

$$t = 01_2 = 1_4, \text{ entspricht } \hat{T}_{+1}: \text{ Übernahme der Zahl der Doppelbesetzungen des vorherigen Zustandes und Inkrement um eins durch } \llbracket d \rrbracket_r = \llbracket d \rrbracket_{r-1} + 1$$

$$t = 10_2 = 2_4, \text{ undefiniert: Sprung zu Programmschritt 5}$$

⁴Formal eine *look-up*-Tabelle

⁵Zeitaufwand $\mathcal{O}(1)$, im Gegensatz zum indirekten Wege, beispielsweise über einen Suchbaum mit $\mathcal{O}(\log m)$ bei m Einträgen

⁶Aus technischer Sicht kann dieser Ansatz als Sonderfall eines *kollisionsfreien hash-Verfahrens* angesehen werden, da zum einen der Suchschlüssel eindeutig dem *hash*-Wert, also dem Listenindex entspricht (injektive *hash*-Funktion) oder, anders formuliert, ein jeder Zielwert in einem eigenen *bucket* abgelegt werden kann, also mit jedem Index nur ein Sequenztypbeitrag verbunden ist.

- $t = 112 = 34$, entspricht \hat{T}_{-1} : Übernahme der Zahl der Doppelbesetzungen des vorherigen Zustandes und Dekrement um eins durch $\llbracket d \rrbracket_r = \llbracket d \rrbracket_{r-1} - 1$
- 3.1.4 Doppelbesetzungszahl prüfen, Element (5): Falls $\llbracket d \rrbracket_r < 0$ oder nach Kriterium (4) $\llbracket d \rrbracket_r > d'$ oder nach Kriterium (5) $\text{mod}(r, 2) = 1$ und zugleich $\llbracket d \rrbracket_r = 0$ gilt, zu Programmschritt 5 springen
 - 3.2 Zu $j := j - 1$ übergehen und, falls $j \geq 0$, zu Schritt 3.1.1 zurückkehren; sonst Schleife beenden
 - 3.3 In d abschließende Doppelbesetzungszahl $\llbracket d \rrbracket_{k'+1} = \llbracket d \rrbracket_{k'} - 1$ eintragen
 - 3.4 Falls problemstellungsgemäß durch Wiederherstellung des Ausgangszustandes $\llbracket d \rrbracket_{k'+1} = 0 = \llbracket d \rrbracket_0$ ist, Ausführung fortsetzen; anderenfalls unmittelbar zu Schritt 4 übergehen
 - 3.5 Falls k eine gerade Graphenordnung, also $\text{mod}(k, 2) = 0$ ist, Ausführung fortsetzen; anderenfalls unmittelbar zu Schritt 3.8 übergehen
 - 3.6 Schleife für $u = 1, 3, \dots, k - 3$:
 - 3.6.1 Schleife für $v = d', d' + 1, \dots, k' - d' - u + 2$:
 - 3.6.1.1 d auf geradzahlige Schrittzahl zwischen dem Auftreten der maximalen Doppelbesetzungszahl d' überprüfen, Element (8): Falls nach Kriterium (6) $\llbracket d \rrbracket_v = d'$ und zugleich $\llbracket d \rrbracket_{v+u} = d'$ gilt, zu Programmschritt 4 springen
 - 3.6.2 Zu $v := v + 1$ übergehen und, falls $v \leq k' - d' - u + 2$, zu Schritt 3.6.1.1 zurückkehren; sonst Schleife beenden
 - 3.7 Zu $u := u + 2$ übergehen und, falls $u \leq k - 3$, zu Schritt 3.6.1 zurückkehren; sonst Schleife beenden
 - 3.8 In einem analog zu Programmablaufschemata 3.10, siehe dazu Abschnitt 3.5, formuliertem Verfahren nach Bestimmung der Prozeßklasse f aus d den Beitrag $\llbracket \mathcal{W} \rrbracket_i \in \mathcal{Q}$ des Sequenztyps i zum Reihenkoeffizienten berechnen, Element (9). Dazu ist gemäß Gleichung (5) (für die Ganzzahlbehandlung, folglich $\llbracket \mathcal{W} \rrbracket_i \in \mathbb{Z}$, siehe Abschnitt 5.2.1) Zugriff auf die Hüpfprozesse $\hat{\rho}^{(n)}$ vonnöten.
 - 4 Zu $i := i + 1$, Element (2), übergehen und, falls $i \leq 4^{k'-1} - 1$, zu Schritt 3.1 zurückkehren; sonst Schleife beenden und zu Schritt 6 springen
 - 5 Zu $i := i + 4^j$, Element (3), übergehen und, falls $i \leq 4^{k'-1} - 1$, zu Schritt 3.1 zurückkehren; sonst Schleife beenden
 - 6 \mathcal{W} dem aufrufenden Hauptprogramm übergeben, Element (10), und Verfahren beenden

Dieses Verfahren kann auch als eigenständiges Programm zur Berechnung der Zahlenfolge in Tabelle 6.1 genutzt werden. Zu diesem Zweck entfällt die Übernahme, Bearbeitung und Übergabe von \mathcal{W} ; statt Flußdiagrammelement (9) wird ein entsprechender Zähler inkrementiert, der nach Element (1) angelegt und vor Element (10) ausgegeben wird.

Der oben beschriebene Algorithmus stellt bereits, wie nun verdeutlicht werden soll, eine über die naheliegende Parallelisierung der Hauptschleife 3 hinausgehende Optimierung dar. Als Grundlage und Maßstab für die Laufzeit betrachten wir zunächst den *brute-force*-Ansatz, Programmvariante I: Beim undefinierten Wert $t = 10_2 = 2_4$ für die Zuordnung und Ausführung des \hat{T} -Operators in Programmschritt 3.1.3 oder bei ungültiger Doppelbesetzungszahl in Schritt 3.1.4 wird dabei im Programmablaufschema 1.1 generell von C nach B zurückgesprungen, also schlicht in Schritt 4 zum nächsten *bit*-Muster λ übergegangen. Eine erste, den Hauptschleifenablauf beschleunigende Verbesserung, Variante II, macht sich die Überlegung zunutze, daß nach erstmaligem Auftreten des Musters 10_2 im (von rechts gezählt) j ten *bit*-Block weitere $4^j - 1$ *bit*-Kombinationen folgen werden, welche gleichfalls den Abbruch in Schritt 3.1.3 auslösen. Daher kann die Schleifenvariable λ durch den dortigen Übergang von C nach C direkt um 4^j erhöht werden, wodurch im j ten Block nun ein neues, nämlich das definierte Muster $t = 11_2 = 3_4 \leftarrow \hat{T}_{-1}$ entsteht. Dieses *greedy*-Konzept weiterführend wird, wie oben formuliert, in Variante III der Rücksprung von C nach C auch gegebenenfalls nach Prüfung der Doppelbesetzungszahl in Schritt 3.1.4 veranlaßt; eine nochmalige, jedoch aufwendigere Verfeinerung stellte die Anwendung dieser Technik auf den Schleifenabbruch durch Schritt 3.6.1.1 dar.⁷

Die Laufzeiten für die Beispielimplementierungen der drei Versionen in ANSI-C auf der auch für die Entwicklung des optimierten Störungsrechneralgorithmus' verwendeten *workstation* vom Typ HP A9638A [85] finden sich ebenfalls in Tabelle 1.1. Deutlich wird darin – neben der immensen Beschleunigung des Verfahrens kraft der Optimierungen – durch Programmvariante I der mit der vierten Potenz skalierende Zeitaufwand bestätigt.

⁷Man beachte, daß sich im Allgemeinen ein den Schleifenvariablenwert beeinflussender *greedy*-Ansatz nicht mit der Parallelisierung der Schleife vereinbaren läßt, da gegebenenfalls ein *thread* die Obergrenze des ihm zugewiesenen Intervalls „überspringt“ und im Intervall des „benachbarten“ *threads* – mit diesem überschneidend – weiterarbeitet. Im Gegensatz zum C-Kompiler von Hewlett-Packard, welcher strikt dem OpenMP-Standard folgt, erlaubt der GNU-Kompiler dennoch die parallele Verarbeitung der Schleife 3, wobei korrekte Ergebnisse nur bei einer *thread*-Anzahl erzielt werden, die einer Zweipotenz entspricht, wodurch Intervalle gleicher Länge entstehen. Erwartungsgemäß führt jedoch dieses kombinierte Verfahren zu einer schlechten Systemauslastung: Bei vier „Rechenprozeßfäden“ erhält beispielsweise der dritte den Wertebereich $\lambda = 2 \cdot 4^{k'-2}, 2 \cdot 4^{k'-2} + 1, \dots, 3 \cdot 4^{k'-2} - 1$, welcher stets an der linken Position $k' - 2$ mit dem Blockwert $t = 2_4 = 10_2$ beginnt. Da dieser undefiniert ist, wird das Intervall nach $\lambda := 2 \cdot 4^{k'-2} + 4^{k'-2}$ durch $\lambda > 3 \cdot 4^{k'-2} - 1$ verlassen, und der betreffende *thread* ist bis zum Abschluß des parallelen Abschnittes durch alle übrigen *threads* auftragslos.

Die in das Hauptprogramm integrierte Fassung des Algorithmus' entspricht der parallelisierbaren und parallelisierten *brute-force*-Variante I mit selbstanpassender Lastverteilung: Einerseits ist diese (portierbare) Version zumindest in den relevanten Störungsrechnungsordnungen $n \leq 17$ auf den dafür erforderlichen Parallelrechnern schneller als die *greedy*-Variante III, andererseits kann damit der wesentliche und geschwindigkeitsbestimmende Programmschritt 3.8, die Berechnung der Sequenztypbeiträge, auf alle Prozessoren verteilt werden. Gegensätzlich wird in Abschnitt 1.3 vorgegangen, siehe Fußnote 10.

k	5	6	7	8	9	10	11
n	7	9	11	13	15	17	19
d'_n	59	397	2878	21266	162732	1253128	9839212
$d'_n/4^{n-1}/10^{-4}$	144	60,6	27,4	12,7	6,06	2,92	1,43
Pz./s für Variante I	≤ 0,06			1,00	15,9	260	4170
Pz./s für Variante II	≤ 0,02			0,37	5,59	82,3	1290
Pz./s für Variante III	≤ 0,01			0,06	0,45	5,59	45,6

Tabelle 1.1. Zeitaufwand für die Erzeugung der möglichen Doppelbesetzungsverläufe der Graphenordnung $k = 5, 6, \dots, 11$. Angegeben ist neben deren Anzahl d'_n jeweils die auf einer RISC-workstation (Typ HP A9638A [85]) benötigte Prozessorzeit (Pz.), welche bei seriellem Vorgehen identisch mit der Maschinen- beziehungsweise Realzeit ist. Erläuterungen zu den drei Varianten (I: *brute force*, II: und III: *greedy*) finden sich im Text. Ergänzend ist in der vierten Zeile der „Füllgrad“ von \mathcal{W} mit gültigen Sequenztypen in Zehntelpro-mille angegeben.

1.2. Bestimmung eines Doppelbesetzungsverlaufstyps. Die vollständige Umsetzung des in Abbildung 1.1 dargestellten und im vorausgegangenen Abschnitt beschriebenen Algorithmus' liefert für jede Störungsrechnungsordnung n mit der („dünn“ besetzten) Liste \mathcal{W} den Beitrag für jeden Doppelbesetzungsverlaufs- beziehungsweise Hüpfsequenztyp zum Reihenkoeffizienten a_n . In Tabelle 1.2 sind beispielhaft die Werte bis einschließlich $n = 5$ aufgeführt.

Notwendiger Schritt für die vollständige Umsetzung dieses Verfahrens ist nun die Zuordnung und Zählung einer jeden im Programmkern, siehe Abschnitt 3.4.13, generierten Hüpfsequenz $H^{(n)} \mapsto \mathcal{H}^{(n)} = (s, d^{(n)})$ zu deren jeweiligem Verlaufstyp im Flußdiagrammelement (16) des Ablaufschemas 3.9 – an jener Stelle also, wo bislang letztlich durch Element (4) in Schema 3.10 der Beitrag einer jeden Sequenz einzeln berechnet wurde. Dazu ist bereits im Hauptprogrammschema 3.1 unmittelbar vor Element (3) eine \mathcal{W} entsprechende Liste \mathcal{U} der Länge 4^{n-1} anzulegen, welche an jeder besetzten Position i von \mathcal{W} im Falle einer modellunabhängigen Veranlagung des Hauptprogrammes zwei Sequenztypzähler $\llbracket \mathcal{U}^{\text{HM}} \rrbracket_i$ und $\llbracket \mathcal{U}^{\text{FKM}} \rrbracket_i$ enthält.⁸ Während die Adresse i des Sequenztyps anhand von d bestimmt wird, wie untenstehender Algorithmus ebenfalls beschreibt, kann das Sequenzvorzeichen s unmittelbar „verrechnet“ werden, indem die Zähler auch datentypseitig negative Werte zulassen.

⁸Man beachte, daß bei konsequenter Verwendung dieser Vorgehensweise im parallelisierten Rahmenprogramm, siehe Abschnitt 3.4.13, bedeutend mehr Arbeitsspeicher als bei unmittelbarer Berechnung des Sequenzbeitrages zum Störungsreihenkoeffizienten erforderlich ist.

n	$(\mathfrak{d} \mapsto \mathfrak{t} \mapsto \mathfrak{i}, -\llbracket \mathcal{W} \rrbracket_{\mathfrak{i}})$
1	$(\begin{array}{ c } \hline \nabla \\ \hline \end{array} \mapsto (\hat{T}_{+1}, \hat{T}_{-1}) \mapsto 0_{10}, -2)$ (Sonderfall)
3	$(\begin{array}{ c c } \hline \nabla & \nabla \\ \hline \end{array} \mapsto (\hat{T}_{+1}, \underline{\hat{T}_0}, \underline{\hat{T}_0}, \hat{T}_{-1}) \mapsto 00_4 = 0_{10}, -4) \wedge$ $(\begin{array}{ c c } \hline \nabla & \nabla \\ \hline \end{array} \mapsto (\hat{T}_{+1}, \underline{\hat{T}_{-1}}, \underline{\hat{T}_{+1}}, \hat{T}_{-1}) \mapsto 31_4 = 13_{10}, 4)$
5	$(\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 0000_4 = 0_{10}, -24) \wedge (\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 0013_4 = 7_{10}, -12) \wedge$ $(\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 0130_4 = 28_{10}, -12) \wedge (\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 1300_4 = 112_{10}, -12) \wedge$ $(\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 1313_4 = 119_{10}, -6) \wedge (\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 0031_4 = 13_{10}, 0) \wedge$ $(\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 1331_4 = 125_{10}, 0) \wedge (\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 3100_4 = 208_{10}, 96) \wedge$ $(\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 3113_4 = 215_{10}, 42) \wedge (\begin{array}{ c c c } \hline \nabla & \nabla & \nabla \\ \hline \end{array} \mapsto 3131_4 = 221_{10}, -48)$

Tabelle 1.2. Doppelbesetzungsverlaufstypen \mathfrak{i} und -typbeiträge \mathcal{W} bis einschließlich Störungsordnung $n = 5$, siehe auch Tabelle 3.9. Für die ersten beiden Ordnungen sind zusätzlich die \hat{T} -Operatorsequenzen $\mathfrak{t}^{(n)}$ vermerkt, vergleiche Abschnitt 3.5.2, wobei für $n = 3$ die in \mathfrak{i} berücksichtigten unterstrichen wurden. Da im Falle $n = 1$ nur zwei Hüpf Schritte, folglich außer \hat{T}_{+1} am Anfang und \hat{T}_{-1} am Ende keine weiteren möglich sind, ist ebenfalls formal $\mathfrak{i} = 0$ und $\llbracket \mathcal{W} \rrbracket_0$ Typbeitrag der einzigen Hüpfsequenz. Man beachte das von Gleichung (3.56) herrührende Vorzeichen von $\llbracket \mathcal{W} \rrbracket_{\mathfrak{i}}$.

Die in Abschnitt 3.5.1 im Detail beschriebene Programmroutine zur Berechnung des Hüpfsequenzbeitrages zum Reihenkoeffizienten wird durch folgendes Verfahren ersetzt:

- 1 Neben der Rechnungsordnung k der Graphen beziehungsweise $n = 2k - 3$ der Störung Ausführungsparameter vom aufrufenden Kern, siehe Abschnitt 3.4.13, und vom Rahmenprogramm, siehe Abschnitt 3.4.9, übernehmen: Hüpfsequenz $H^{(n)}$ und daraus abgeleitet \mathfrak{d} und \mathfrak{z} , für den rechnenden Prozeßfaden im Parallelbetrieb Sequenztypzähler \mathcal{U}^{HM} (Hubbardmodell) und \mathcal{U}^{FKM} (Falicov-Kimball-Modell) sowie zugehörige Sequenzzähler $\mathfrak{h}_n^{\text{HM}}$ und $\mathfrak{h}_n^{\text{FKM}}$, siehe Flußdiagrammelement (1) in Abbildung 1.2
- 2 Aus dem Endzustand $|{}_1M, {}_2M\rangle^{(2k-2)}$ die Anzahl der vertauschten Elektronen innerhalb einer jeden der zwei Spinspezies bestimmen und summieren, Element (2). (Dieser Programmschritt entspricht exakt dem in Punkt 3 in Abschnitt 3.5.1 beschriebenen.)
- 3 Berechnung des Sequenzvorzeichens mit $s = (-1)^b$

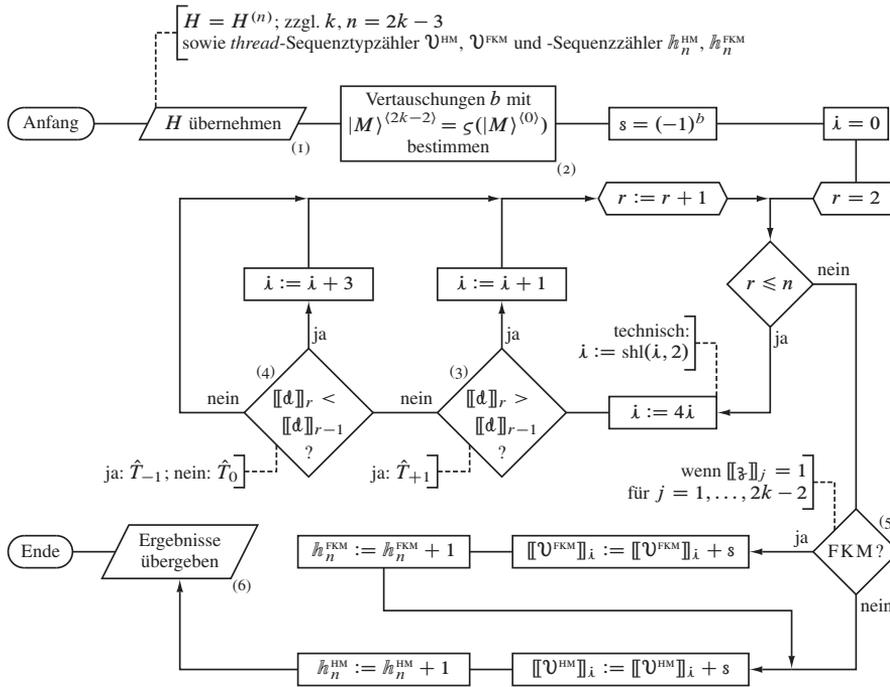


Abbildung 1.2. Ablaufplan eines Unterprogramms für die Bestimmung des Doppelbesetzungsverlaufstyps der vom Kern, Abbildung 3.9, generierten Hüpfsequenz; Ersatz für das Unterprogramm in Abschnitt 3.5.1.

- 4 Adresse des Doppelbesetzungsverlaufstyps $\lambda = 0$ initialisieren
- 5 Schleife über die Zwischenzustände $r = 2, 3, \dots, 2k - 3$:
 - 5.1 *bit*-Muster von λ um zwei Dualstellen durch $\lambda := \text{shl}(\lambda, 2)$ nach links verschieben
 - 5.2 Falls die Doppelbesetzungsanzahl gegenüber dem vorherigen Zustand durch das Einwirken des Operators \hat{T}_{+1} zugenommen hat, Element (3), Adresse durch Addition des Äquivalentkodes $01_2 = 1_4$ auf $\lambda := \lambda + 1$ setzen und zu Schritt 6 übergehen, sonst fortsetzen
 - 5.3 Falls die Doppelbesetzungsanzahl gegenüber dem vorherigen Zustand durch das Einwirken des Operators \hat{T}_{-1} abgenommen hat, Element (4), Adresse durch Addition des Äquivalentkodes $11_2 = 3_4$ auf $\lambda := \lambda + 3$ setzen und zu Schritt 6 übergehen; anderenfalls muß die Doppelbesetzungszahl durch \hat{T}_0 unverändert geblieben sein (formal ist der zum Operator äquivalente Kode $00_2 = 0_4$ zur Adresse zu addieren)

- 6 Zu $r := r + 1$ übergehen und, falls $r \leq n$, zu Schritt 5.1 zurückkehren; sonst Schleife beenden
- 7 Falls es sich bei der Hüpfsequenz um eine zum Falicov-Kimball-Modell gehöri-
ge handelt, Element (5), also nur – gemäß unserer Wahl – Elektronen mit \uparrow -Spin
propagiert sind und entsprechend $\llbracket \mathfrak{z} \rrbracket_j = 1$ für $j = 1, \dots, 2k - 2$ gilt, „Vorzei-
chen“ $s \in \{-1, +1\}$ zum Sequenztypzähler addieren, $\llbracket \mathcal{V}^{\text{FKM}} \rrbracket_i := \llbracket \mathcal{V}^{\text{FKM}} \rrbracket_i + s$,
und $\mathfrak{h}_n^{\text{FKM}}$ um eins erhöhen
- 8 Den Sequenztypzähler des Hubbardmodells dem Vorzeichen $s \in \{-1, +1\}$ ent-
sprechend um eins erhöhen oder erniedrigen, $\llbracket \mathcal{V}^{\text{HM}} \rrbracket_i := \llbracket \mathcal{V}^{\text{HM}} \rrbracket_i + s$, und $\mathfrak{h}_n^{\text{HM}}$ um
eins erhöhen. Die Berücksichtigung der Symmetrie zwischen beiden Spinpezies
im Hubbardmodell durch den Vorfaktor 2 in beiden Zählern wird in das Hauptpro-
gramm verlagert.
- 9 Unterprogramm beenden und Sequenz(typ)zähler für jedes Modell an das aufrufen-
de Rahmenprogramm zurückgeben, Element (6)

Die Anwendung dieses Verfahrens liefert die bis einschließlich Störungsrechnungs-
ordnung $n = 5$ in Tabelle 1.3 aufgeführten Werte der Typzähler für das Falicov-Kimball-
und für das Hubbardmodell. Ergänzend sind dort auch die Hüpfsequenzanzahlen nach
Vorzeichen getrennt aufgeführt, wobei für die Modelle jeweils⁹

$$\mathfrak{h}_n = \sum_{i=0}^{4n-1-1} (\llbracket \mathcal{V}^+ \rrbracket_i + (-1)^i \llbracket \mathcal{V}^- \rrbracket_i) \quad (\text{I.1})$$

gilt.

1.3. Berechnung des Reihenkoeffizienten. Um von den in \mathcal{W} und \mathcal{V} „gelisteten“ Se-
quenztypbeiträgen und Sequenztypzählern zum Störungsreihenkoeffizienten a_n der be-
trachteten Ordnung n zu gelangen, kann mit drei Änderungen auf den in Abschnitt 1.1
des Anhangs beschriebenen Algorithmus zur Erzeugung aller Typen der Doppelbeset-
zungsverläufe beziehungsweise Sequenzen zurückgegriffen werden.¹⁰ Dieser Verarbei-
tungsschritt muß generell zwischen den Flußdiagrammelementen (3) und (4) im Haupt-
programmablaufplan 1.1 eingefügt werden. Bei parallelisiertem Betrieb ist dies im Rah-
menprogramm, Schema 3.6, im entsprechenden Bereich nach dem Ende der Ausgangs-
zustandsschleife direkt vor Element (11) nötig, wobei vor dem Abschluß noch die *thread*-
eigenen Variablen $a'_n = 0$ aufzusummieren und, wie unten erläutert, zu a_n zu skalieren
sind.

⁹Siehe Fußnote 98

¹⁰Im Gegensatz zur dort verwendeten Fassung 1, siehe Fußnote 7, wird hier jedoch auf die *greedy*-Varian-
te III zurückgegriffen: Während \mathcal{W} (parallelisiert) „Programm-global“ für alle *threads* angelegt wird, muß die
Berechnung des Reihenkoeffizientenbeitrages aus \mathcal{W} und den (im Parallelbetrieb) „Prozessfaden-lokalen“ \mathcal{V}
zunächst innerhalb eines jeden *threads* seriell mit abschließender Summation zum Endergebnis durch das
Hauptprogramm erfolgen. In diesem Falle kann eine Beschleunigung nur durch die effiziente Behandlung der
Hauptschleife durch Übergänge von C nach C im Programmablaufschema 1.1 erreicht werden.

n	i	Modell					
		Falicov-Kimball			Hubbard		
		$\llbracket \mathcal{V}^+ \rrbracket_i$	$\llbracket \mathcal{V}^- \rrbracket_i$	$\llbracket \mathcal{V} \rrbracket_i$	$\llbracket \mathcal{V}^+ \rrbracket_i$	$\llbracket \mathcal{V}^- \rrbracket_i$	$\llbracket \mathcal{V} \rrbracket_i$
1	0_{10}	1	0	1	1	0	1
3	$00_4 = 0_{10}$	2	0	2	6	0	6
	$31_4 = 13_{10}$	2	0	2	4	0	4
5	$0000_4 = 0_{10}$	10	0	10	78	0	78
	$0013_4 = 7_{10}$	4	0	4	36	0	36
	$0031_4 = 13_{10}$	6	0	6	36	0	36
	$0130_4 = 28_{10}$	2	-2	0	20	-2	18
	$1300_4 = 112_{10}$	4	0	4	36	0	36
	$1313_4 = 119_{10}$	4	0	4	16	0	16
	$1331_4 = 125_{10}$	4	0	4	16	0	16
	$3100_4 = 208_{10}$	6	0	6	36	0	36
	$3113_4 = 215_{10}$	4	0	4	16	0	16
	$3131_4 = 221_{10}$	8	0	8	32	0	32

Tabelle 1.3. Anzahl der Hüpfsequenzen nach Doppelbesetzungsverlaufstyp, Modell und Vorzeichen bis einschließlich Störungsordnung $n = 5$. Die beiden Summanden von $\llbracket \mathcal{V} \rrbracket_i = \llbracket \mathcal{V}^+ \rrbracket_i + \llbracket \mathcal{V}^- \rrbracket_i$ geben jeweils die Hüpfsequenzanzahl ohne und mit Teilchenvertauschungen an; man beachte, daß – wie hier belegt – die Möglichkeit dazu gemäß Abschnitt 3.4.15 erst ab $n = 5$ besteht.

Anzupassen ist in Abbildung 1.1 zunächst Flußdiagrammelement (1), folglich Programmschritt 1:

- 1 Zur Ausführung durch einen jeden *thread* bei Parallelbetrieb: Von der aufrufenden (Haupt)programmebene die Graphenordnung k als Ausführungsparameter und die Liste \mathcal{W} , zudem die Sequenztypzähler \mathcal{V}^{HM} (Hubbardmodell) und \mathcal{V}^{FKM} (Falicov-Kimball-Modell) und für jedes Modell die Zwischensummen $a'_n = 0$ des Reihen­koeffizienten des jeweiligen Prozeßfadens übernehmen, dabei Hilfsgrößen $k' = 2k - 3$ (entspricht der Störungsrechnungsordnung n) und $d' = \lfloor k/2 \rfloor$ berechnen

Im weiteren Verlauf ist Element (9) und entsprechend Programmschritt 3.8 durch folgenden zu ersetzen:

- 3.8 Für jedes der untersuchten Modelle: Addition der Beträge $\llbracket \mathcal{W} \rrbracket_i$ aller $\llbracket \mathcal{V} \rrbracket_i$ Sequenzen des Typs i zur Reihen­koeffizientenzwischensumme:

$$a'_n := a'_n + \llbracket \mathcal{V} \rrbracket_i \cdot \llbracket \mathcal{W} \rrbracket_i \quad (1.2)$$

Abschließend ist Element (10) und somit Programmschritt 6 zu ändern durch:

6 a'_n für jedes Modell an die aufrufende (Haupt)programmebene übergeben und Verfahren beenden

Die Einbeziehung dieses Unterprogrammes, das effektiv einer optimierten Summationschleife entspricht, kann die verfahrensunabhängige Koeffizientengleichung (4.1) unter Berücksichtigung des genäherten Hauptnenners \tilde{z}_n zu

$$a_n = -\frac{r}{2^{(n+3)/2}} \frac{1}{n+1} \frac{1}{\tilde{z}_n} \sum_{i=0}^{4^n-1} \llbracket \mathcal{V} \rrbracket_i \cdot \llbracket \mathcal{W} \rrbracket_i \quad (1.3)$$

umformuliert werden. Damit besteht auch Konsistenz zur Reihe (4.2) für die Ergebnisdiskussion in Kapitel 4.

An die Abschnitte 1.4.2 und 3.5.2 anknüpfend, in welchen die Reihenkoeffizienten erster und dritter Störungsrechnungsordnung als Grundzustandsenergiebeiträge für U^{-1} und U^{-3} explizit berechnet wurden, können wir mit diesem effizienten Verfahren nun auch das Ergebnis fünfter Ordnung übersichtlich aufschreiben. Für die 10 Doppelbesetzungsverlaufstypen in U^{-5} ergeben sich mit den Tabellen 1.2 und 1.3 folgende Werte:

Modell	i	0000 ₄	0013 ₄	0031 ₄	0130 ₄	...		
HM	$-\llbracket \mathcal{V} \rrbracket_i \cdot \llbracket \mathcal{W} \rrbracket_i$	-1872	-432	0	-216	...		
FKM	$-\llbracket \mathcal{V} \rrbracket_i \cdot \llbracket \mathcal{W} \rrbracket_i$	-240	-48	0	0	...		
	...	1300 ₄	1313 ₄	1331 ₄	3100 ₄	3113 ₄	3131 ₄	$-a'_n$
	...	-432	-96	0	3456	672	-1536	-456
	...	-48	-24	0	576	168	-384	0

Darin ist mit einem Vorzeichen, welches von Gleichung (3.56), siehe Abschnitt 3.3.4, wieder auf die allgemeine Form (4.1) beziehungsweise obige Gleichung (1.3) zurückführt,

$$a'_n = \sum_{i=0}^{4^n-1} \llbracket \mathcal{V} \rrbracket_i \cdot \llbracket \mathcal{W} \rrbracket_i . \quad (1.4)$$

Auf die Gleichungen (1.27) und (1.28) respektive (3.104) und (3.105) folgt somit aus a'_n durch Gleichung (1.3)

$$E_0^{(5),\text{HM}} = -\frac{19}{8} \quad (1.5)$$

und erwartungsgemäß

$$E_0^{(5),\text{FKM}} = 0 . \quad (1.6)$$

2. Hilfsfunktionen

Im folgenden Abschnitt werden neben einem einfachen Verfahren zur Berechnung der Mangoldtzahlen kurz jene „kleineren“ Prozeduren beschrieben, welche für die gesamte in Kapitel 3 eingehend beschriebene Rechenmethode an verschiedenen Stellen wesentlich sind. Diese sind oder stützen sich ihrerseits auf Standardalgorithmen der Kombinatorik, etwa zur Erzeugung von Partitionen und Permutationen, oder auf übliche Sortier Routinen. Soweit nicht spezielle Anpassungen vonnöten sind – beispielsweise für die Berechnung der Parität vertauschter Listenelemente –, finden sich die Grundlagen in DONALD KNUTHS *Art of Computer Programming* [43] auch aus mathematischer Sicht erschöpfend beschrieben; eine eher zur Anwendung hin orientierte Darstellung bieten die Lehrbücher von ROBERT SEDGEWICK [71] oder THOMAS CORMEN [72].

2.1. Sortierverfahren. Das Ordnen von Mengen oder Tupeln ist in der maschinellen Ausführung der Störungsrechnung stets dann vonnöten, wenn diese aus kartesischen Produkten hervorgehen und nicht *alle Vertauschungen*, sondern nur *ein Repräsentant* daraus benötigt wird, wovon vornehmlich die Isomorphismen-freie Generierung von Graphen betroffen ist. Die „Theorie“ dazu findet sich in Kapitel 2, eingeführt in den Abschnitten 2.1.14 und insbesondere 2.2.6; die „Praxis“ zeigt sich im Kapitel 3 vor allem in den Abschnitten 3.1.1 und 3.2.1 beim Aufbau der (gefärbten) Butcherbäume als erstem Schritt des Hauptprogrammes.

Die vielfältigen elementaren und hochentwickelten Sortieralgorithmen werden beispielsweise umfassend im dritten Band von Referenz [43] behandelt. Um den Aufwand der Neuimplementierung nicht durch eine weitergehende Detailanalyse zu erhöhen, haben der Wahl der hier angewendeten Verfahren nur drei einfache Forderungen zugrunde gelegen. Unter Berücksichtigung der vergleichsweise kleinen Datenmengen¹¹ – die maximale Listenlänge ist von der Ordnung n der Störungsrechnung – waren dies

- (1) die Ausführbarkeit ohne Hilfsdatenstrukturen (*in-place*), um Speicherbereitstellungen und -zugriffe zu minimieren und Platz in den *caches* zu sparen,
- (2) die „Stabilität“, also die Beibehaltung bestehender (Teil)ordnung bei Verwendung von Sortierschlüsseln¹² und
- (3) die einfache Modifizierbarkeit für eine datentechnisch vorteilhafte Behandlung von Tabellenzeilen.

¹¹Dadurch sind prinzipiell stark optimierte Verfahren wie *heap sort* unnötig.

¹²Betroffen davon sind derzeit noch „experimentelle“ Programmteile, beispielsweise zur Bestimmung von Ecken- und (daraus ableitbar) Kantenklassen durch Baumrotationen, vorbereitet in Abschnitt 2.1.7, und die hier nicht behandelte topologiekonsistente Sortierung von Ästen gemäß ihres Färbungsgewichtes, angedeutet in Abschnitt 2.2.3.

Da Sortierschritte für die Lösung des Gesamtproblems vom Zeitbedarf her nicht kritisch sind, wurde vorzugsweise das einfache *insertion sort* in einer die Vergleichsanzahl reduzierenden Variante benutzt, welches im ungünstigen Fall kein schlechteres Laufzeitverhalten als etwa das sonst schnellere, aber instabile¹³ *quick sort* aufweist und im Gegensatz zum generell schnelleren und stabilen *merge sort* keine Hilfsliste benötigt. Folglich wurden nicht gegebenenfalls auf Maschinenebene leistungsoptimierte Bibliotheksfunktionen, zum Beispiel der *standard template library* (STL) [76], eingesetzt, sondern die Verfahren unmittelbar im Quelltext bereitgestellt.

Neben der Anwendung bei der Baumerzeugung bedarf auch der in der Neuimplementierung gewählte Ansatz zur Bestimmung des Beitragsvorzeichens einer jeden generierten Hüpfsequenz eines Sortierverfahrens, da bei der Behandlung fermionischer Systeme festgestellt werden muß, siehe Programmpunkt 3 in Abschnitt 3.5.1 beziehungsweise Punkt 2 in Abschnitt 1.2 des Anhanges, ob zwischen Ausgangs- und Endzustand eine gerade oder ungerade Anzahl an Teilchenvertauschungen auf dem Gitterausschnitt stattgefunden hat. Zu diesem Zweck wurde der Einfachheit halber auf *bubble sort*, Algorithmus B im Kapitel 5.2.2 des Bandes 3 von Referenz [43], dem wohl einfachst denkbaren überhaupt, zurückgegriffen und dessen Eigenschaft, stets nur benachbarte Elemente zu vertauschen, nutzbar gemacht¹⁴. Interesse besteht folglich nicht an der abschließend sortiert vorliegenden Liste der Teilchennummern für die „Elektronenfarben“ $s \in \{1, 2\}$, genauer gesagt dem (End)zustandsvektor $|_s M\rangle^{(2k-2)}$, welcher somit wieder dem Ausgangszustand $|_s M\rangle^{(0)}$ entspricht¹⁵, sondern am Wert eines mit einer beliebigen geraden Zahl initialisierten Zählers $_s b$, welcher nach jedem Vertauschungsschritt B3 um eins inkrementiert wird.

2.2. Permutationen. Das zum vorausgegangenen Abschnitt gegenläufige Problem besteht im Aufsuchen aller Vertauschungsmöglichkeiten der Elemente einer (geordnet) erzeugten Menge oder eines Tupels und tritt im neuimplementierten Störungsrechnungsverfahren, Kapitel 3, zunächst in der $1/U$ -Entwicklung des Hamiltonoperators auf, dem zweiten Schritt des Hauptprogrammes, vergleiche Abschnitt 3.3.2. Auch für die (lexikographische) Generierung von Permutationen existieren viele, teils auf Sonderfälle spezialisierte Ansätze, welche ebenfalls in Referenz [43] eingehend diskutiert werden. Im Falle der Herleitung der Hüpfprozesse durch die Elementarfaktorbildung und die Nennerentwicklung wurde dabei mit „*next permutation*“ ein allgemeines Verfahren aus der STL modifiziert, welches auch Listen mit klassenweise äquivalenten Elementen behandelt.

Diese auf Algorithmus L aus Kapitel 7.2.1.2 des Bandes 4 von Referenz [43] fußende Methode ist ebenfalls im Zentralteil der Störungsrechnung, nämlich der Erzeugung aller

¹³Eine stabile Version ist zulasten des Aufwandes durch entsprechend konstruierte Schlüssel realisierbar.

¹⁴Gravierender Nachteil des *bubble-sort*-Verfahrens ist generell dessen stets quadratische Zeitkomplexität; dieser wirkt sich allerdings mit einem Anteil von circa 1,3% beispielsweise für $\mathcal{O}(U^{-11})$ kaum auf die Ausführungsdauer der Störungsrechnung aus.

¹⁵Auf Vertauschungen wird ohnehin nur geprüft, wenn Ausgangs- und Endzustand teilchennummernbezogen nicht übereinstimmen.

Hüpfsequenzen für jede mögliche Hüpfkantenvertauschung, siehe Abschnitt 3.4.9 (iterativer Rahmen), verwendet worden. Durch das *greedy*-Konzept des Ausgangszustandssequenzierers wird dabei jedoch nicht stets die nächste Permutation der Sprungabfolge der aktuell verwendeten auf Durchführbarkeit geprüft, sondern zunächst anhand eines Rückgabewertes der Hüpfoutine, siehe Abschnitt 3.4.13 (rekursiver Kern), eine bestimmte Anzahl an Permutationen übergangen beziehungsweise – bildlich gesprochen – Äste aus dem Permutationsbaum „ausgebrochen“. Dazu ist der primitive Ansatz nahelegend, über eine den Permutationsgenerator der STL umgebende Schleife mit dem Anfangswert 1 und einem Endwert aus $\{1!, 2!, \dots\}$ die „uninteressanten“ Vertauschungen „ins Leere hinein“ erzeugen zu lassen, bis die nächste prüfenswerte erhalten wird. Tatsächlich wird im Programm trotz fakultativer Laufzeit bis zu – empirisch ermittelten – $4! = 24$ Kantenvertauschungen aufgrund der Geschwindigkeit oben genannter Methode derart vorgegangen; darüberhinaus wird jedoch ein Algorithmus benötigt, der gezielt eine beliebige (und somit die nächste potentiell zu Sequenzen führende) Permutation eines n' -Tupels ($i \mid i = 1, 2, \dots, n'$) zu erzeugen erlaubt.

Eine elegante Lösung¹⁶ dieses keineswegs einfachen Problemes wurde von JAMES McCAFFREY 2003 veröffentlicht [77], wobei die Verwendung von *Faktoraden* (*factoradics*) zur Indizierung von Permutationen anhand des Lehmer-Kodes auf eine Empfehlung des Mathematikers PETER CAMERON zurückgeht. Zur Erläuterung des Vorgehens führen wir zunächst die allgemeingültige Zerlegung

$$\forall u \in \mathbb{N} : u = \sum_{i=0}^{\infty} c_i i! \quad \text{mit} \quad c_i \in [0, i] \subset \mathbb{N} \tag{2.1}$$

ein; dieser Ausdruck reduziert sich, wenn wir $u \in [0, k'! - 1]$ einschränken, auf die endliche Summe

$$u = \sum_{i=0}^{k'-1} c_i i! \tag{2.2}$$

Das Koeffiziententupel (c_0, c_1, \dots) notieren wir als *faktorielles Stellenwertsystem* in umgekehrter, also bezüglich i absteigender Reihenfolge und nennen es Faktorade. Für gegebenes k' hat diese folgendes Aussehen:

$$\mathfrak{F} = (c_{k'-1}, c_{k'-2}, \dots, c_1, c_0) . \tag{2.3}$$

Beispiel für die ersten Werte von $u = 0, 1, \dots, k'! - 1$ mit $k' = 4$:

u	0	1	2	3	4	5	...
\mathfrak{F}_u	(0, 0, 0, 0)	(0, 0, 1, 0)	(0, 1, 0, 0)	(0, 1, 1, 0)	(0, 2, 0, 0)	(0, 2, 1, 0)	...

¹⁶Wie eine mit dem Kern von Algorithmus L vergleichende Betrachtung zeigt, ist der Zeitbedarf dieses Verfahrens höher; darin liegt der Grund für den Rückfall auf die gegebenenfalls mehrfach erfolgende sukzessive Permutationserzeugung, sobald der Gesamtaufwand auf diesem Wege geringer ist.

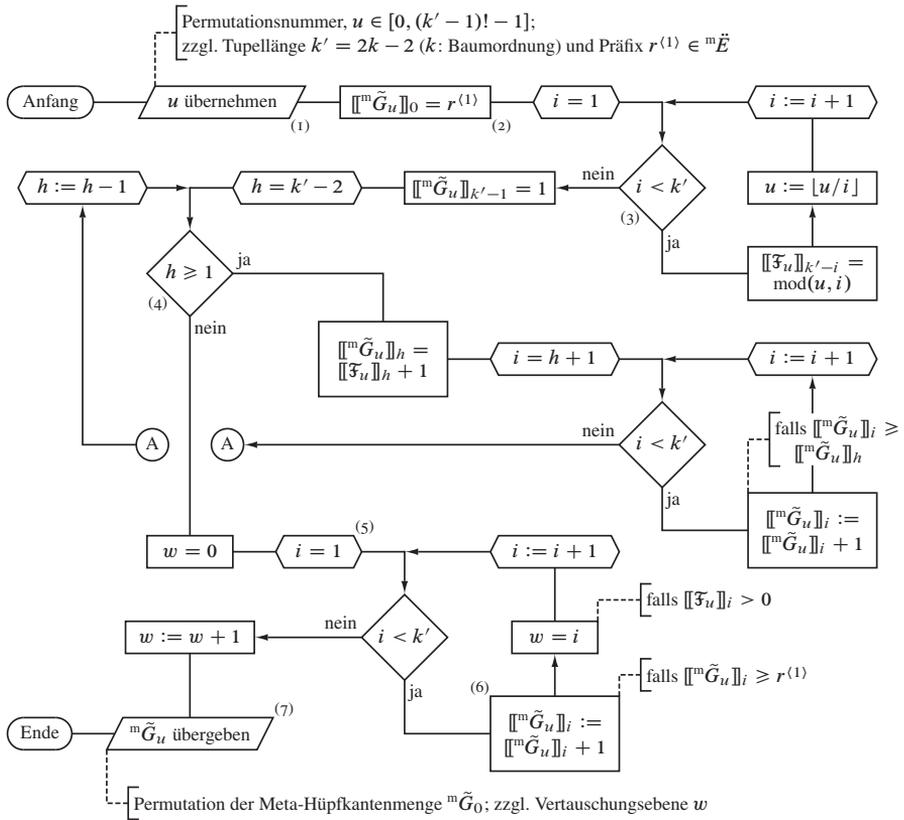


Abbildung 2.1. Ablaufplan eines Unterprogramms zur direkten Erzeugung der u ten lexikographischen Permutation des k' -Tupels ${}^m\tilde{G}$ (Meta-Hüpfkantenmenge), siehe Referenz [77], unter Berücksichtigung des Präfix' $r^{(1)}$ (Sequenzinitialisierungskante). Die Bezeichnungen sind Abschnitt 3.4.9 entsprechend gewählt; man beachte jedoch, daß die Elemente von ${}^m\tilde{G}_u$ hier analog zur Faktorate \mathcal{F}_u mit $i = 0, 1, \dots, k' - 1$ numeriert sind.

Die „Konstruktion“ eines der $k'!$ Vertauschungen der $k' = 2(k - 1)$ Elemente des Tupels ${}^m\tilde{G}$, welches der geordneten Meta-Hüpfkantenmenge entspricht und für das unter Einbeziehung von Gleichung (3.94) ${}^m\tilde{G}_{u=0} = {}^m\tilde{G}_{r^{(1)}}$ gelte, aus \mathcal{F}_u dient nun folgender Algorithmus, worin zunächst aus gegebenem u die Faktorate selbst aufgebaut wird. Problemstellungsgemäß, siehe insbesondere Abschnitt 3.4.9, betrachten wir dabei die Einschränkung $u \in [0, (k' - 1)! - 1]$, da das erste Tupelelement mit dem „Präfix“ $r^{(1)} \in {}^m\tilde{E}$, nämlich der Sequenzinitialisierungskante, vorgegeben ist. Eine explizite Berücksichtigung der durch Festlegung von $r^{(2)} \in {}^mG \setminus {}^m\tilde{E}$ gegebenenfalls bewirkten

Permutationsbaumzerlegung, siehe Abschnitt 5.2.2, ist nicht erforderlich, da u entsprechend vorgegeben wird.¹⁷

- 1 Neben der Nummer u der zu erzeugenden Permutation die Tupellänge k' und den Präfix $r^{(1)}$ beispielsweise vom in Schritt 4.3.5 aufrufenden Rahmenprogramm (vergleiche Abschnitt 3.4.9) übernehmen, siehe Flußdiagrammelement (1) in Abbildung 2.1
- 2 Erstes, nicht von der Vertauschung betroffenes Element $\llbracket \tilde{G}_u \rrbracket_0 = r^{(1)}$ festlegen
- 3 (Aufbau der Faktorate) Schleife über $i = 1, 2, \dots, k' - 1$:
 - 3.1 Element $k' - i$ der Faktorate auf den Rest der Ganzzahldivision u/i , also $\text{mod}(u, i)$ setzen
 - 3.2 u auf den ganzzahligen Teil des Quotienten u/i setzen. Da $u \in \mathbb{N}$ und $i \in \mathbb{N} \setminus \{0\}$ gilt, ist hier $u := \lfloor u/i \rfloor$.
- 4 Zu $i := i + 1$ übergehen und, falls $i < k'$, zu Schritt 3.1 zurückkehren; sonst Schleife beenden
- 5 Das „rechte äußere“ Element $\llbracket \tilde{G}_u \rrbracket_{k'-1} = 1$ festlegen. Man beachte dabei und im Folgenden, daß aufwandverringern für die Elemente $\llbracket \tilde{G}_u \rrbracket_i$ nicht der Indexbereich $i = 1, 2, \dots, k'$, sondern analog zu den Stellen der Faktorate $i = 0, 1, \dots, k' - 1$ verwendet wird.
- 6 (Aufbau der Permutation) Schleife über $h = k' - 2, k' - 3, \dots, 1$:
 - 6.1 $\llbracket \tilde{G}_u \rrbracket_h = \llbracket \mathcal{F}_u \rrbracket_h + 1$ setzen
 - 6.2 Schleife über $i = h + 1, h + 2, \dots, k' - 1$:
 - 6.2.1 Falls $\llbracket \tilde{G}_u \rrbracket_i \geq \llbracket \tilde{G}_u \rrbracket_h$ ist, $\llbracket \tilde{G}_u \rrbracket_i := \llbracket \tilde{G}_u \rrbracket_i + 1$ setzen
 - 6.3 Zu $i := i + 1$ übergehen und, falls $i < k'$, zu Schritt 6.2.1 zurückkehren; sonst Schleife beenden
- 7 Zu $h := h - 1$ übergehen und, falls $h > 0$, zu Schritt 6.1 zurückkehren; sonst Schleife beenden
- 8 $w = 0$ initialisieren
- 9 (Berücksichtigung des Präfix') Schleife über $i = 1, 2, \dots, k' - 1$:
 - 9.1 Falls $\llbracket \tilde{G}_u \rrbracket_i \geq r^{(1)}$ ist, $\llbracket \tilde{G}_u \rrbracket_i := \llbracket \tilde{G}_u \rrbracket_i + 1$ setzen
 - 9.2 (Bestimmung der Vertauschungsebene) Falls $\llbracket \mathcal{F}_u \rrbracket_i > 0$ ist, $w = i$ setzen
- 10 Zu $i := i + 1$ übergehen und, falls $i < k'$, zu Schritt 9.1 zurückkehren; sonst Schleife beenden
- 11 w zwecks Angleichung an den unter anderem für \tilde{G} geltenden Index-Wertebereich $\mathbb{N} \setminus \{0\}$ des Rahmenprogrammes um eins erhöhen
- 12 u te Permutation von \tilde{G}_0 sowie „Vertauschungsebene“¹⁸ w , welche der Stelle des „von links gesehen“ ersten vertauschten Tupelelementes entspricht, beispielsweise vom aufrufenden Rahmenprogramm übergeben, Element (7), und Verfahren beenden

¹⁷Prinzipiell kann kraft einer analogen Modifizierung der aufrufenden Programminstanz sogar auf die ausdrückliche Verwendung des Präfixes verzichtet werden.

¹⁸Die Begriffswahl ist Hinweis auf den formal zugrundeliegenden Permutationsbaum ${}^m\mathcal{P}_{2k-3, r^{(1)}}$.

Um die Wirkungsweise dieses Verfahrens zu demonstrieren, greifen wir auf das zweite Beispiel in Abschnitt 3.4.10 zurück und ergänzen die dortige Tabelle um eine Spalte für die Faktorade \mathfrak{F}_u , wodurch das „Konstruktionsschema“ der Permutation der Meta-Hüpfkantenmenge ${}^m\tilde{G}_u$ unmittelbar einsichtig wird:

Nr.	u	\mathfrak{F}_u	w	${}^m\tilde{G}_u$	j	j'	p	p'	v	flag?
1	0	(0, 0, 0, 0, 0, 0)	1	{1, 2, 3, 4, 5, 6}	1	–	3	6	–	nein
2	6	(0, 0, 1, 0, 0, 0)	3	{1, 2, 4, 3, 5, 6}	3	–	3	6	–	nein
3	12	(0, 0, 2, 0, 0, 0)	3	{1, 2, 5, 3, 4, 6}	3	6	5	1	–	ja
4	13	(0, 0, 2, 0, 1, 0)	5	{1, 2, 5, 3, 6, 4}	5	6	5	1	4	ja
5	14	(0, 0, 2, 1, 0, 0)	4	{1, 2, 5, 4, 3, 6}	4	–	4	2	–	nein
6	16	(0, 0, 2, 2, 0, 0)	4	{1, 2, 5, 6, 3, 4}	4	4	5	1	–	ja
7	17	(0, 0, 2, 2, 1, 0)	5	{1, 2, 5, 6, 4, 3}	4	4	5	1	3	ja
8	18	(0, 0, 3, 0, 0, 0)	3	{1, 2, 6, 3, 4, 5}	3	–	4	2	–	nein
9	20	(0, 0, 3, 1, 0, 0)	4	{1, 2, 6, 4, 3, 5}	4	6	5	1	–	ja
10	21	(0, 0, 3, 1, 1, 0)	5	{1, 2, 6, 4, 5, 3}	5	6	5	1	4	ja
11	22	(0, 0, 3, 2, 0, 0)	4	{1, 2, 6, 5, 3, 4}	4	4	5	1	–	ja
12	23	(0, 0, 3, 2, 1, 0)	5	{1, 2, 6, 5, 4, 3}	4	4	5	1	–	ja
13	24	(0, 1, 0, 0, 0, 0)	2	{1, 3, 2, 4, 5, 6}	–	–	2	24	–	—
14	48	(0, 2, 0, 0, 0, 0)	2	{1, 4, 2, 3, 5, 6}	2	–	2	24	–	nein
15	72	(0, 3, 0, 0, 0, 0)	2	{1, 5, 2, 3, 4, 6}	2	–	2	24	–	nein
16	96	(0, 4, 0, 0, 0, 0)	2	{1, 6, 2, 3, 4, 5}	2	2	4	2	–	ja
17	98	(0, 4, 0, 1, 0, 0)	4	{1, 6, 2, 4, 3, 5}	2	2	5	1	–	ja
18	99	(0, 4, 0, 1, 1, 0)	5	{1, 6, 2, 4, 5, 3}	2	2	5	1	–	ja
19	100	(0, 4, 0, 2, 0, 0)	4	{1, 6, 2, 5, 3, 4}	2	2	5	1	–	ja
20	101	(0, 4, 0, 2, 1, 0)	5	{1, 6, 2, 5, 4, 3}	2	2	5	1	–	ja
21	102	(0, 4, 1, 0, 0, 0)	3	{1, 6, 3, 2, 4, 5}	–	–	3	6	–	—
22	108	(0, 4, 2, 0, 0, 0)	3	{1, 6, 4, 2, 3, 5}	2	2	5	1	–	ja
23	109	(0, 4, 2, 0, 1, 0)	5	{1, 6, 4, 2, 5, 3}	2	2	5	1	–	ja
24	110	(0, 4, 2, 1, 0, 0)	4	{1, 6, 4, 3, 2, 5}	–	–	4	2	–	—
25	112	(0, 4, 2, 2, 0, 0)	4	{1, 6, 4, 5, 2, 3}	2	2	4	2	–	ja
26	114	(0, 4, 3, 0, 0, 0)	3	{1, 6, 5, 2, 3, 4}	2	2	5	1	–	ja
27	115	(0, 4, 3, 0, 1, 0)	5	{1, 6, 5, 2, 4, 3}	2	2	5	1	–	ja
28	116	(0, 4, 3, 1, 0, 0)	4	{1, 6, 5, 3, 2, 4}	–	–	4	2	–	—

Fortsetzung auf der nächsten Seite

Nr.	u	\mathfrak{F}_u	w	${}^m\tilde{G}_u$	j	j'	p	p'	v	$flag?$
<i>Fortsetzung</i>										
29	118	(0, 4, 3, 2, 0, 0)	4	{1, 6, 5, 4, 2, 3}	2	2	4	2	-	ja
-	120	(1, 0, 0, 0, 0, 0)	1	{2, 1, 3, 4, 5, 6}	-	-	-	-	-	—

Mit wenigen Modifikationen läßt sich obige Vorgehensweise zur Erzeugung jeder beliebigen Permutation $u \in [0, k'! - 1]$ des k' elementigen Ausgangstupels erweitern. Zunächst entfallen dabei erwartungsgemäß die Schritte 2 und 9.1 beziehungsweise die Flußdiagrammelemente (2) und (6), welche die Berücksichtigung des Präfixes bewirken. Damit auch $\llbracket \mathfrak{F}_u \rrbracket_0$, also der Koeffizient $c_{k'-1}$ aus Gleichung (2.3) einbezogen wird, ist zunächst folgende Änderung notwendig, siehe Flußdiagrammelement (3):

3 (Aufbau der Faktorade) Schleife über $i = 1, 2, \dots, k'$: [...]

sowie entsprechend

4 Zu $i := i + 1$ übergehen und, falls $i \leq k'$, zu Schritt 3.1 zurückkehren; sonst Schleife beenden

Die „Übersetzung“ der nun erweiterten Faktorade erfordert die Anpassung einer weiteren Schleifengrenze, siehe Flußdiagrammelement (4):

6 (Aufbau der Permutation) Schleife über $h = k' - 2, k' - 3, \dots, 0$: [...]

sowie entsprechend

7 Zu $h := h - 1$ übergehen und, falls $h \geq 0$, zu Schritt 6.1 zurückkehren; sonst Schleife beenden

In gleicher Weise muß abschließend für $\llbracket \mathfrak{F}_u \rrbracket_0$ bei der Ermittlung des (bezüglich des Index i in $\llbracket {}^m\tilde{G}_u \rrbracket_i$) ersten vertauschten Elementes im Permutationstupel verfahren werden, siehe Flußdiagrammelement (5):

9 (Berücksichtigung des Präfix') Schleife über $i = 0, 1, \dots, k' - 1$: [...]

Damit ist der in Referenz [77] beschriebene Algorithmus für künftige Anwendungen vollständig umgesetzt.

2.3. Partitionen. Die Notwendigkeit, alle Partitionen¹⁹ einer Ganzzahl zu erzeugen, ergibt sich in zwei zentralen Abschnitten des Gesamtverfahrens: zum einen beim Aufbau der Butcherbäume in den Abschnitten 3.1.1 und 3.2.1, wie in Abbildung 3.2 für ungefärbte illustriert, zum anderen bei der Generierung der \hat{S} -Exponenten-Tupel als „Bausteine“ der Hüpfprozesse im Abschnitt 3.3.2 in Umsetzung von Gleichung (1.16), siehe Abschnitt 1.4. Mit Algorithmus P wird im Kapitel 7.2.1.4 des Bandes 4 in Referenz [43] ein Verfahren vorgestellt, das unverändert übernommen wurde. Die Implementierung ist entsprechend auch nicht als eigenständige Funktion erfolgt; vielmehr bildet der Algorithmus eine äußere, umrahmende Schleife, welche als Schritt P3 den eigentlichen „Verarbeitungskern“ enthält.

¹⁹Deren Gesamtzahl ν_p für $p \in \mathbb{N}$ ist in Tabelle 2.5 angegeben, siehe Abschnitt 2.2.2.

2.4. Mangoldtzahlen. Die in Abschnitt 5.2.1 eingeführten Mangoldtschen Zahlen \mathcal{M} , katalogisiert und erläutert als Ganzzahlfolge A014963 in Referenz [24], lassen sich folgendermaßen definieren:

- (1) $\mathcal{M}_1 := 1$,
- (2) $\mathcal{M}_i := 1$, wenn sich \mathcal{M}_i als Produkt aus beliebig vielen, aber voneinander *verschiedenen* $\mathcal{M}_j \in \{\mathcal{M}_2, \mathcal{M}_3, \dots, \mathcal{M}_{i-1}\}$ darstellen läßt,
- (3) $\mathcal{M}_i := i$, wenn i prim ist; anderenfalls wird
- (4) \mathcal{M}_i jene Primzahl zugewiesen, welche als Faktor im „Primfaktoraufbau“, Punkt (2), noch fehlt.

Beispiele, vergleiche Tabelle 5.5: $\mathcal{M}_4 = 4/\mathcal{M}_2 = 2$, $\mathcal{M}_6 = 6/(\mathcal{M}_2 \cdot \mathcal{M}_3) = 1$, $\mathcal{M}_8 = 8/(\mathcal{M}_2 \cdot \mathcal{M}_4) = 2$

Zu den für die Zahlentheorie bedeutsamen Eigenschaften gehört der mit Punkt (2) in obiger Definition verknüpfte Zusammenhang

$$\mathcal{M}_i = \frac{\text{kgV}(1, 2, \dots, i)}{\text{kgV}(1, 2, \dots, i-1)}. \quad (2.4)$$

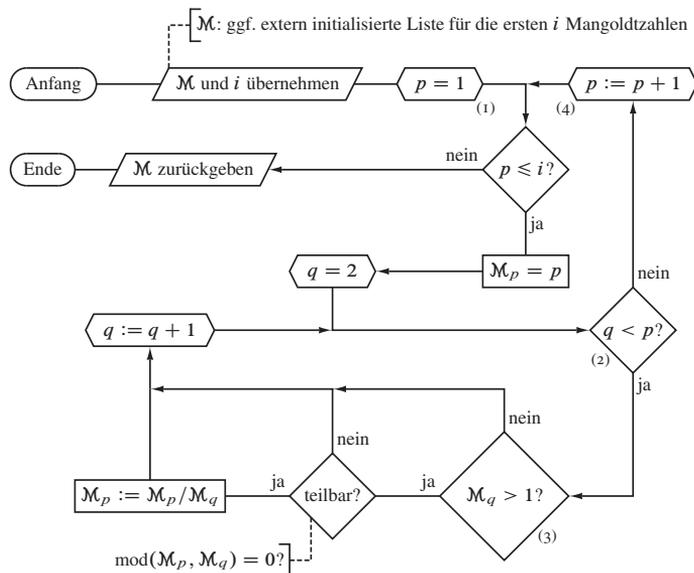


Abbildung 2.2. Einfacher Algorithmus zur Berechnung der Folge der Mangoldtzahlen \mathcal{M} . Flußdiagrammelement (3), welches die Division durch eins verhindert, stellt bereits eine Schleifenoptimierung dar, siehe auch Tabelle 2.1.

Quasi „aus dem Nichts“, also angefangen mit \mathcal{M}_1 lassen sich die ersten i Mangoldtzahlen mit folgendem einfachen *brute-force*-Verfahren mit der Laufzeit $\mathcal{O}(i^2)$ erzeugen.²⁰ Dabei wird, passend für die Verwendung im Gesamtalgorithmus der graphentheoretischen Störungsrechnung, davon ausgegangen, daß eine übergeordnete aufrufende Programminstanz eine leere Liste für die angeforderten i Ergebniswerte an das Unterprogramm übergibt.

- 1 Ausführungsparameter i und Liste für \mathcal{M} übernehmen
- 2 Schleife für $p = 1, 2, \dots, i$:
 - 2.1 \mathcal{M}_p mit p initialisieren
 - 2.2 Schleife für $q = 2, 3, \dots, p - 1$:
 - 2.2.1 Wenn \mathcal{M}_p durch \mathcal{M}_q restlos teilbar ist, also $\text{mod}(\mathcal{M}_p, \mathcal{M}_q) = 0$ gilt, dann $\mathcal{M}_p := \mathcal{M}_p / \mathcal{M}_q$ ausführen
 - 2.3 Zu $q := q + 1$ übergehen und, falls $q < p$, zu Schritt 2.2.1 zurückkehren; sonst Schleife beenden
- 3 Zu $p := p + 1$ übergehen und, falls $p \leq i$, zu Schritt 2.1 zurückkehren; sonst Schleife beenden
- 4 Verfahren beenden und \mathcal{M} übergeben

Die Ausführung von Programmschritt 2.2.1 sollte effizienterweise davon abhängig gemacht werden, ob $\mathcal{M}_q > 1$ ist, siehe Flußdiagrammelement (3) in Abbildung 2.2; anderenfalls kann nämlich sofort zum nächsten potentiellen Divisor \mathcal{M}_{q+1} übergegangen werden, ohne Zeit auf die Ganzzahldivision durch eins verwenden zu müssen.

i	25000	50000	100000	200000	400000	800000
Pz./s für Variante I	12,3	47,1	186	736	2940	12800
Pz./s für Variante II	6,20	24,6	97,4	387	1550	7040

Tabelle 2.1. Zeitaufwand für die *brute-force*-Berechnung der Mangoldtzahlen von \mathcal{M}_1 bis einschließlich \mathcal{M}_i . Angegeben ist jeweils die auf einer RISC-workstation (Typ HP A9638A [85]) benötigte Prozessorzeit (Pz.), welche bei hiesigem Verfahren identisch mit der Maschinen- beziehungsweise Realzeit ist. Variante II nutzt dabei im Gegensatz zu Variante I die Laufzeitoptimierung durch Element (3) im Programmablaufschema 2.2; beide Versionen operieren auf hierfür ausreichenden 32 Bit-Ganzzahlen.

Tabelle 2.1 zeigt die Laufzeiten einer Beispielimplementierung in ANSI-C auf der für die gesamte Studie verwendeten Entwicklungsumgebung [85], einer workstation vom Typ HP A9638A. Abgesehen von der für diesen Zweck ausreichenden Verwendung von

²⁰Unter Ausnutzung der Assoziativität des *kleinsten gemeinsamen Vielfachen* in Gleichung (2.4) und dessen „Verwandtschaft“ mit dem *größten gemeinsamen Teiler* sind die Mangoldtzahlen beispielsweise auch über den Euklidischen Algorithmus und seine Optimierungen in besserer Laufzeit zugänglich.

32 Bit langen Ganzzahlen²¹, gelten dabei die gleichen Übersetzungs- und Ausführungsbedingungen wie für die Störungsrechnung, vergleiche Abschnitt 5.1.3. Die Umsetzung der vorausgegangenen allgemeinen Beschreibung als eigenständiges Programm erfordert, die Liste \mathcal{M} für die Mangoldtzahlen zu Beginn der Hauptfunktion im Programmschritt 1 anzulegen, also vor Flußdiagrammelement (1). Die Ergebnisausgabe, welche gegenüber dem eigentlichen Rechenaufwand vernachlässigbaren Zeitbedarf hat, erfolgt für jeden Wert \mathcal{M}_p im Schritt 3 beziehungsweise zwischen den Elementen (2) und (4).

Die Betrachtung der gemessenen Zeiten bestätigt für beide Programmversionen eindrucksvoll den quadratisch skalierenden Aufwand. Der einzige zu höherer Rechenzeit abweichende Wert für $i = 800000$ ist darauf zurückzuführen, daß der Datenpuffer von 1536 kBytes (entspricht 393216 Datenworten zu 32 Bit) zwischen den Registern des Mikroprozessors PA-8700+ und dem Arbeitsspeicher des Rechners nicht mehr für die Mangoldtzahlliste ausreichend ist; daher entstehen merkliche Verzögerungen durch das Nachladen derselben beim Zugriff auf \mathcal{M}_q innerhalb von Schleife 2.2.

3. Tafeln

Die Tabellen dieses Kapitels dienen – neben der über Abschnitt 5.1.1 hinausgehenden graphischen Veranschaulichung der drastisch anwachsenden Problemkomplexität – vornehmlich zwei Zielen: Zum einen wird die Überprüfung einer möglichen Neuprogrammierung anhand des Vorliegens einiger Zwischenergebnisse aus dem Gesamtalgorithmus vereinfacht, zum anderen kann insbesondere durch die viergefärbten Butcherbäume im hiesigen Abschnitt 3.2 die Erweiterung des Rechenverfahrens auf Probleme jenseits der Halbfüllung zumindest in niedriger Ordnung unterstützt werden.

3.1. Zweigefärbte Butcherbäume. Die zweigefärbten Butcherbäume treten in der vorgelegten Implementierung der Störungsrechnung nur mittelbar in den Ausgangszuständen auf. In folgender Tabelle sind der Vollständigkeit halber sämtliche ${}^2\mathcal{B}_{k,l,m}$ bis einschließlich $k = 5$ aufgeführt; dabei liegt die in der Produktionsfassung des Algorithmus für die viergefärbten Bäume verwendete Sortierung und die übliche Farbkonvention ($\blacksquare \mapsto \uparrow$ -Spin und $\blacksquare \mapsto \downarrow$ -Spin mit $\blacksquare < \blacksquare$) zugrunde. Die Anzahl ${}^2\mu_{k,l}$ für gegebene Ordnung k und Struktur l ist als Ganzzahlfolge A242353 in Referenz [24] verzeichnet. Die für das Hubbardmodell bei anregungsfreier Halbfüllung relevanten Ausgangszustände sind mit einem Rahmen versehen; deren Anzahl ist unter dem Kürzel Az. in der letzten Spalte angegeben.

k	l	${}^2\mathcal{B}_{k,l,m} \mid m = 1, 2, \dots, {}^2\mu_{k,l}$	${}^2\mu_{k,l}$	Az.
1	1		2	
2	1		4	2

Fortsetzung auf der nächsten Seite

²¹Werden Ganzzahlen von der Breite der Prozessorregister (64 Bit) genutzt, verringert sich der Zeitaufwand auf etwa zwei Drittel.

k	l	${}^2\mathcal{B}_{k,l,m} \mid m = 1, 2, \dots, {}^2\mu_{k,l}$	${}^2\mu_{k,l}$	Az.
<i>Fortsetzung</i>				
3	1		8	4
	2		6	4
4	1		16	8
	2		12	6
	3		16	12
	4		8	6
5	1		32	16
	2		24	12
	3		32	16
	4		16	8
	5		32	24
	6		24	18

Fortsetzung auf der nächsten Seite

k	l	${}^2\mathcal{B}_{k,l,m} \mid m = 1, 2, \dots, {}^2\mu_{k,l}$	${}^2\mu_{k,l}$	Az.
<i>Fortsetzung</i>				
7			20	14
8			24	20
9			10	8

3.2. Viergefärbte Butcherbäume. Die folgende Tabelle enthält sämtliche viergefärbte Butcherbäume ${}^4\mathcal{B}_{k,l,m}$ gemäß der Sortierung im neuimplementierten Rechenverfahren und der üblichen Farbkonvention ($\square \mapsto \emptyset$ (Loch), $\blacksquare \mapsto \uparrow$ -Spin, $\blacksquare \mapsto \downarrow$ -Spin und $\blacksquare \mapsto \uparrow \wedge \downarrow = \updownarrow$ (Doppelbesetzung) mit $\square < \blacksquare < \blacksquare < \blacksquare$) bis einschließlich $k = 4$. Die Anzahl ${}^4\mu_{k,l}$ für gegebene Ordnung k und Strukturnummer l ist als Ganzzahlfolge A242354 in Referenz [24] verzeichnet. Auch in dieser Tabelle sind die Ausgangszustände des Hubbardmodells bei anregungsfreier Halbfüllung (mit zu dieser Arbeit konsistenten Färbungsnummern m) eingerahmt und deren Anzahl in der Spalte Az. angegeben.

k	l	${}^4\mathcal{B}_{k,l,m} \mid m = 1, 2, \dots, {}^4\mu_{k,l}$	${}^4\mu_{k,l}$	Az.
1	1		4	
2	1		16	2
3	1		64	4
3	2		40	4

Fortsetzung auf der nächsten Seite

k	l	${}^4\mathcal{B}_{k,l,m} \mid m = 1, 2, \dots, {}^4\mu_{k,l}$	${}^4\mu_{k,l}$	Az.
-----	-----	---------------------------------------------------------------	-----------------	-----

Fortsetzung

4	1		256	8
	2			

Fortsetzung auf der nächsten Seite

k	l	${}^4\mathcal{B}_{k,l,m} \mid m = 1, 2, \dots, {}^4\mu_{k,l}$	${}^4\mu_{k,l}$	Az.
-----	-----	---------------------------------------------------------------	-----------------	-----

Fortsetzung

3		160	6
---	--	-----	---

Fortsetzung auf der nächsten Seite

k	l	${}^4\mathcal{B}_{k,l,m} \mid m = 1, 2, \dots, {}^4\mu_{k,l}$	${}^4\mu_{k,l}$	Az.
-----	-----	---------------------------------------------------------------	-----------------	-----

Fortsetzung

4		256	12
	80	6	

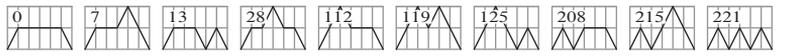
3.3. Doppelbesetzungsverlaufstypen. In Ergänzung zu dem in Abschnitt 1.1 dieses Anhangs beschriebenen Verfahren enthält die folgende Tabelle sämtliche Doppelbesetzungsverlaufstypen $\mathfrak{d} = \tilde{\mathfrak{d}}_{k,j}$ in deren Darstellung als „Höhenprofil“ bis einschließlich $k = 6$. In jedem Verlauf ist zusätzlich das Dezimalzahläquivalent der zugehörigen Binärkodierung $\mathfrak{i} = \tilde{\mathfrak{i}}_{k,j}$ vermerkt; es gilt für gegebene Ordnung k stets $\tilde{\mathfrak{i}}_{k,j} < \tilde{\mathfrak{i}}_{k,j+1}$ und somit $\tilde{\mathfrak{d}}_{k,i} < \tilde{\mathfrak{d}}_{k,i+1}$. Die jeweilige Anzahl $\tilde{\mathfrak{d}}_k$ (letzte Spalte) findet sich als Ganzzahlfolge A240605 in Referenz [24].

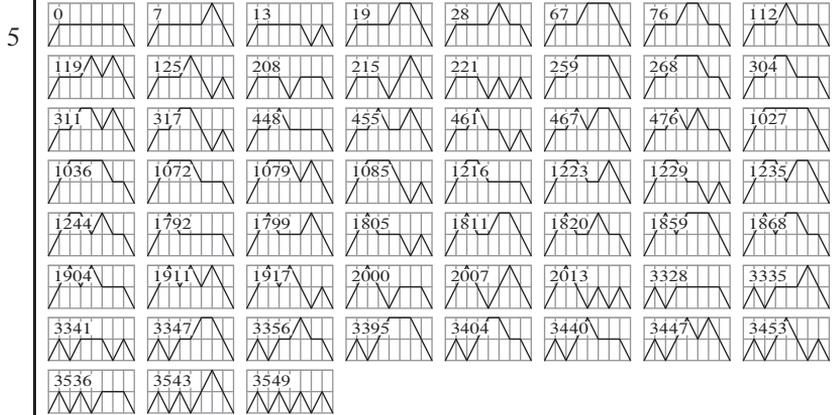
k	$\tilde{\mathfrak{d}}_{k,j} \mid j = 1, 2, \dots, \tilde{\mathfrak{d}}_k$	$\tilde{\mathfrak{d}}_k$
2		1
3		2

Fortsetzung auf der nächsten Seite

$$k \mid \tilde{a}_{k,j} \mid j = 1, 2, \dots, \tilde{a}_k \mid \tilde{a}_k$$

Fortsetzung

4  10



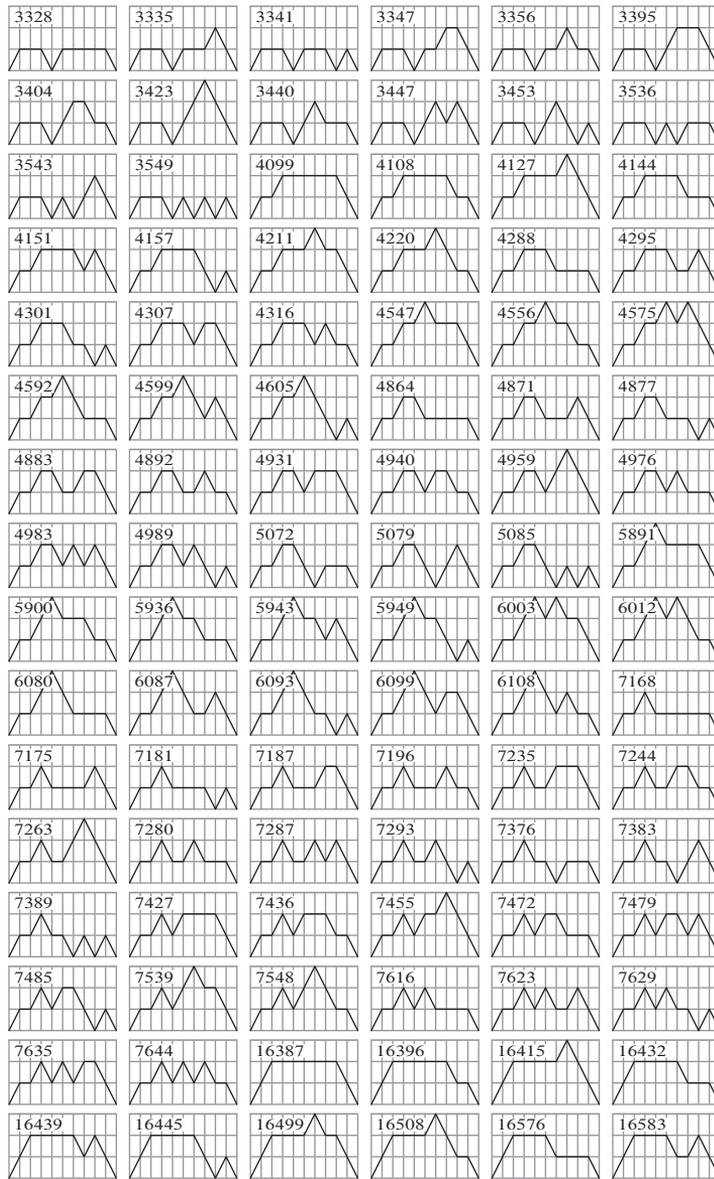
10

59

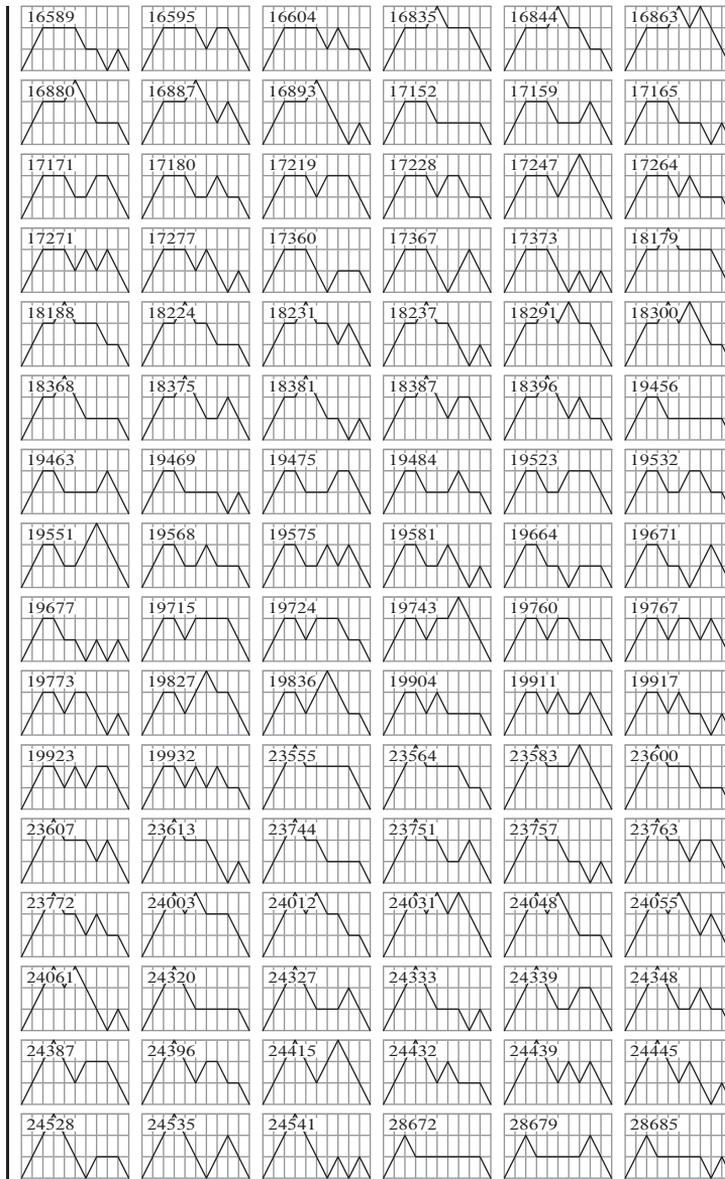
Fortsetzung auf der nächsten Seite

$k \mid \tilde{d}_{k,j} \mid j = 1, 2, \dots, \tilde{d}_k$
 \tilde{d}_k

Fortsetzung

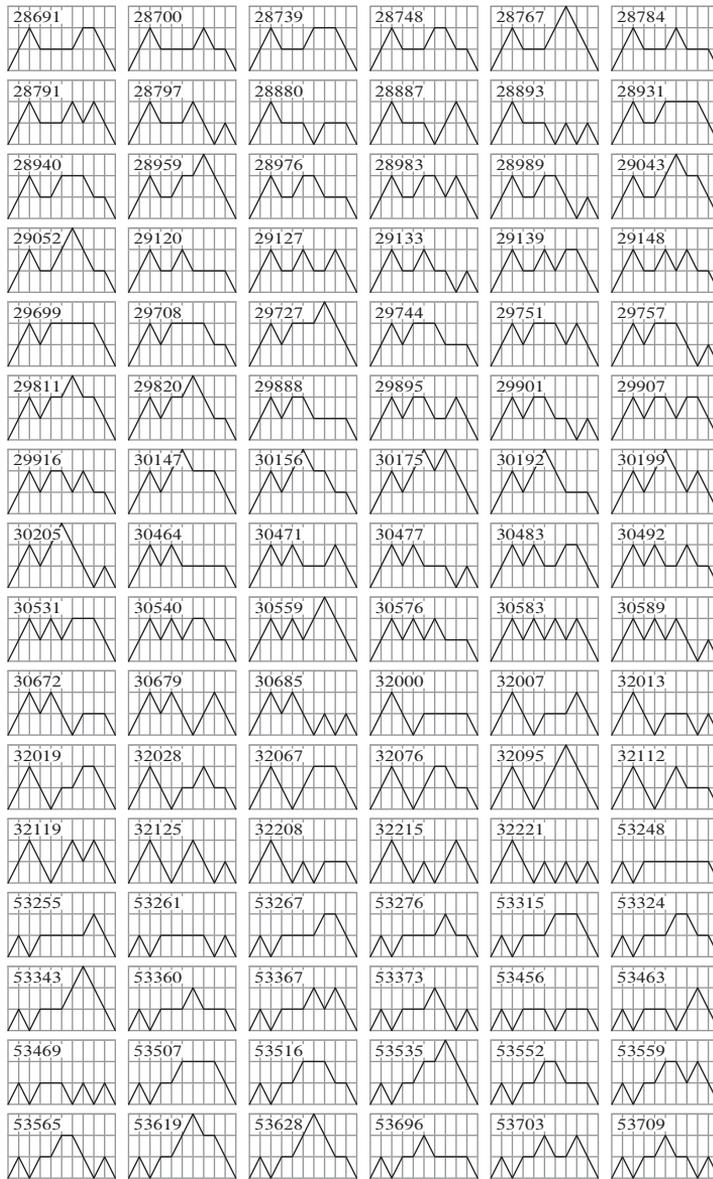


Fortsetzung auf der nächsten Seite

$k \mid \tilde{a}_{k,j} \mid j = 1, 2, \dots, \tilde{a}_k$
 \tilde{a}_k
Fortsetzung

Fortsetzung auf der nächsten Seite

$k \mid \tilde{d}_{k,j} \mid j = 1, 2, \dots, \tilde{d}_k$
 \tilde{d}_k

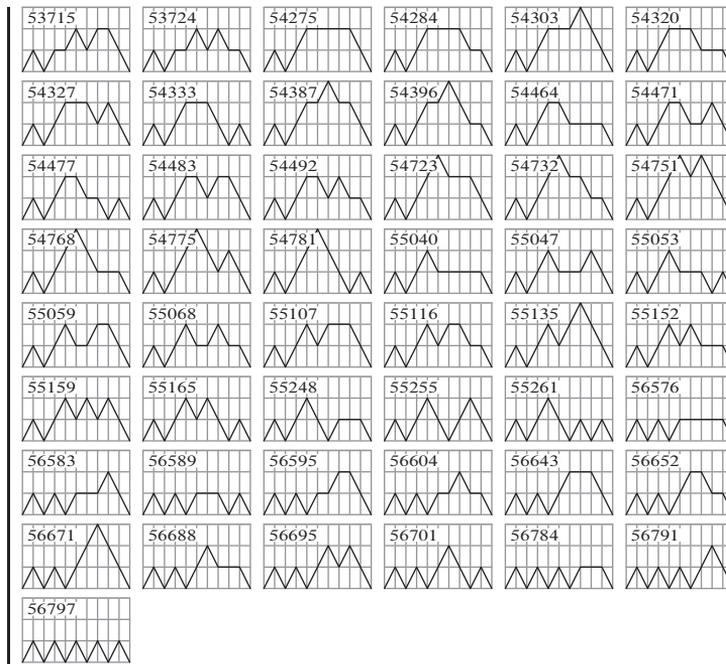
Fortsetzung



Fortsetzung auf der nächsten Seite

$k \mid \tilde{a}_{k,j} \mid j = 1, 2, \dots, \tilde{a}_k$
 \tilde{a}_k

Fortsetzung



Literatur- und Quellenverzeichnis

- [1] JOHN HUBBARD: Electron Correlations in Narrow Energy Bands. – *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 276 (1365), 238–257 (1963).
- [2] MARTIN C. GUTZWILLER: Effect of Correlation on the Ferromagnetism of Transition Metals. – *Physical Review Letters* 10 (5), 159–162 (1963); JUNJIRO KANAMORI: Electron Correlation and Ferromagnetism of Transition Metals. – *Progress of Theoretical Physics* 30 (3), 275–289 (1963).
- [3] JENŐ SÓLYOM: *Fundamentals of the Physics of Solids*. Volume 3 (Normal, Broken-Symmetry, and Correlated Systems). Berlin: Springer, 2011.
- [4] ELLIOTT H. LIEB AND FA-YUEH WU: Absence of Mott Transition in an Exact Solution of the Short-Range, One-Band Model in One Dimension. – *Physical Review Letters* 20 (25), 1445–1448 (1968); Erratum. – *ibid.* 21 (3), 192 (1968); The one-dimensional Hubbard model: a reminiscence. – *Physica A: Statistical Mechanics and its Applications* 321 (1–2), 1–27 (2003).
- [5] PATRICK FAZEKAS: *Lecture Notes on Electron Correlation and Magnetism*. Series: *Modern Condensed Matter Physics* 5. Singapore: World Scientific, 1999.
- [6] DOUGLAS J. SCALAPINO: A common thread: The pairing interaction for unconventional superconductors. – *Reviews of Modern Physics* 84 (4), 1383–1417 (2012).
- [7] SIR NEVILL FRANCIS MOTT: *Metal-Insulator Transitions*. Second Edition. London: Taylor & Francis, 1990; FLORIAN GEBHARD: *The Mott Metal-Insulator Transition. Models and Methods*. Series: *Springer Tracts in Modern Physics* 137. Berlin: Springer, 1997.
- [8] YUZURU KUROSAKI, YOSHIMITSU SHIMIZU, KAZUYA MIYAGAWA, KAZUSHI KANODA, AND GUNZI SAITO: Mott Transition from a Spin Liquid to a Fermi Liquid in the Spin-Frustrated Organic Conductor κ -(ET)₂Cu₂(CN)₃. – *Physical Review Letters* 95 (17), 177001, 4 pages (2005).
- [9] IMMANUEL BLOCH, JEAN DALIBARD, AND WILHELM ZWERGER: Many-body physics with ultracold gases. – *Reviews of Modern Physics* 80 (3), 885–964 (2008).
- [10] MANUEL ENDRES et al.: Observation of Correlated Particle-Hole Pairs and String Order in Low-Dimensional Mott Insulators. – *Science* 334 (6053), 200–203 (2011).

- [11] LORENZO DE LEO, JEAN-SÉBASTIEN BERNIER, CORINNA KOLLATH, ANTOINE GEORGES, AND VITO W. SCAROLA: Thermodynamics of the three-dimensional Hubbard model: Implications for cooling cold atomic gases in optical lattices. – *Physical Review A: Atomic, molecular, and optical physics* 83 (2), 023606, 17 pages (2011).
- [12] TOSIO KATO: On the Convergence of the Perturbation Method. I. – *Progress of Theoretical Physics* 4 (4), 514–523 (1949).
- [13] CLAUDE BLOCH: Sur la théorie des perturbations des états liés. – *Nuclear Physics* 6, 329–347 (1958); ALBERT MESSIAH: *Quantenmechanik*. Band 2: Dritte Auflage. Berlin: Walter de Gruyter, 1990.
- [14] MINORU TAKAHASHI: Half-filled Hubbard model at low temperature. – *Journal of Physics C: Solid State Physics* 10 (8), 1289–1301 (1977).
- [15] EVA KALINOWSKI: *Mott-Hubbard-Isolator in hoher Dimension*. Dissertation. Marburg: Fachbereich Physik der Philipps-Universität, 2002.
- [16] EVA KALINOWSKI AND FLORIAN GEBHARD: Mott-Hubbard Insulator in Infinite Dimensions. – *Journal of Low Temperature Physics* 126 (3–4), 979–1007 (2002).
- [17] MICHAEL P. EASTWOOD, FLORIAN GEBHARD, EVA KALINOWSKI, SATOSHI NISHIMOTO, REINHARD M. NOACK: Analytical and numerical treatment of the Mott-Hubbard insulator in infinite dimensions. – *The European Physics Journal B: Condensed Matter and Complex Systems* 35 (2), 155–175 (2003).
- [18] NILS BLÜMER AND EVA KALINOWSKI: Mott insulator: Tenth-order perturbation theory extended to infinite order using a quantum Monte Carlo scheme. – *Physical Review B: Condensed matter and materials physics* 71 (19), 195102, 5 pages (2005); Ground state of the frustrated Hubbard model within DMFT: energetics of Mott insulator and metal from EPT and QMC. – *Physica B: Condensed Matter* 359–361, 648–650 (2005).
- [19] EVA KALINOWSKI AND WŁADYSŁAW GLUZA: Evaluation of high-order terms for the Hubbard model in the strong-coupling limit. – *Physical Review B: Condensed matter and materials physics* 85 (4), 045105, 8 pages (2012).
- [20] GABRIEL KOTLIAR AND DIETER VOLLHARDT: Strongly Correlated Materials: Insights From Dynamical Mean-Field Theory. – *Physics Today* 57 (3), 53–59 (2004).
- [21] SATOSHI NISHIMOTO, FLORIAN GEBHARD, AND ERIC JECKELMANN: Dynamical density-matrix renormalization group for the Mott-Hubbard insulator in high dimensions. – *Journal of Physics: Condensed Matter* 16 (39), 7063–7083 (2004).
- [22] ERWIN MÜLLER-HARTMANN: The Hubbard model at high dimensions: some exact results and weak coupling theory. – *Zeitschrift für Physik B: Condensed Matter* 76 (2), 211–217 (1989); SANDRA MAHLERT: *Störungstheorie in vierter Ordnung für das halbgefüllte Hubbard-Modell im Limes hoher Dimensionen*. Dissertation. Marburg: Fachbereich Physik der Philipps-Universität, 2003.
- [23] FLORIAN GEBHARD, ERIC JECKELMANN, SANDRA MAHLERT, SATOSHI NISHIMOTO, AND REINHARD M. NOACK: Fourth-order perturbation theory for the half-filled Hubbard model in infinite dimensions. – *The European Physical Journal B: Condensed Matter and Complex Systems* 36 (4), 491–509 (2003).

- [24] NEIL J. A. SLOANE: *The On-Line Encyclopedia of Integer Sequences*. Elektronisch publiziert unter <http://oeis.org>, 2010. (Die auf diese Referenz folgende Nummer, abgetrennt durch einen Doppelpunkt, bezeichnet die zitierte Ganzzahlfolge.)
- [25] NAN LIN, CHRIS A. MARIANETTI, ANDREW J. MILLIS, AND DAVID R. REICHMAN: Dynamical Mean-Field Theory for Quantum Chemistry. – *Physical Review Letters* 106 (9), 096402, 4 pages (2011).
- [26] DIETER VOLLHARDT: Dynamical mean-field theory for correlated electrons. Einstein Lecture. – *Annalen der Physik (Berlin)* 524 (1), 1–19 (2012); DIETER VOLLHARDT, KRZYSZTOF BYCZUK, AND MARCUS KOLLAR: Dynamical Mean-Field Theory. Chapter 7 in: ADOLFO AVELLA AND FERDINANDO MANCINI (Editors): *Strongly Correlated Systems*. Series: *Springer Series in Solid-State Sciences* 171. Berlin, Heidelberg: Springer, 2012.
- [27] RALF BULLA, THEO A. COSTI, AND DIETER VOLLHARDT: Finite-temperature numerical renormalization group study of the Mott transition. – *Physical Review B: Condensed matter and materials physics* 64 (4), 045103, 9 pages (2001).
- [28] DANIEL J. GARCÍA, KAREN HALLBERG, AND MARCELO J. ROZENBERG: Dynamical Mean Field Theory with the Density Matrix Renormalization Group. – *Physical Review Letters* 93 (24), 246403, 4 pages (2004).
- [29] MICHAŁ KARSKI, CARSTEN RAAS, AND GÖTZ S. UHRIG: Electron spectra close to a metal-to-insulator transition. – *Physical Review B: Condensed matter and materials physics* 72 (11), 113110, 4 pages (2005); Single-particle dynamics in the vicinity of the Mott-Hubbard metal-to-insulator transition. – *ibid.* 77 (7), 075116, 20 pages (2008); ROK ŽITKO AND THOMAS PRUSCHKE: Energy resolution and discretization artifacts in the numerical renormalization group. – *ibid.* 79 (8), 085106, 17 pages (2009).
- [30] NILS BLÜMER: *Mott-Hubbard Metal-Insulator Transition and Optical Conductivity in High Dimensions*. Dissertation. Herzogenrath: Shaker, 2003.
- [31] DANIEL RUHL AND FLORIAN GEBHARD: Strong-coupling approach to the Mott-Hubbard insulator on a Bethe lattice in dynamical mean-field theory. – *Physical Review B: Condensed matter and materials physics* 83 (3), 035120, 12 pages (2011).
- [32] INGO PESCHEL et al.: The Density Matrix Renormalization Group: Concepts and Applications. Part IX in: HOLGER FEHSKE, RALF SCHNEIDER, AND ALEXANDER WEISSE (Editors): *Computational Many-Particle Physics*. Series: *Lecture Notes in Physics* 739. Berlin, Heidelberg: Springer, 2008.
- [33] GUNTHER GRUBER: *Entwicklung einer graphbasierten Methode zur Analyse von Hüpfsequenzen auf Butcherbäumen und deren Implementierung in Haskell*. Diplomarbeit. Marburg: Fachbereich Mathematik und Informatik der Philipps-Universität, 2011; persönl. Mitteilungen von RITA LOOGEN und GUNTHER GRUBER.
- [34] LEOPOLDO M. FALICOV AND JOHN C. KIMBALL: Simple Model for Semiconductor-Metal Transitions: SmB_6 and Transition-Metal Oxides. – *Physical Review Letters* 22 (19), 997–999 (1969); PETER G. J. VAN DONGEN: Exact mean-field theory of the extended simplified Hubbard model. – *Physical Review B: Condensed matter and materials physics* 45 (5), 2267–2281 (1992).

- [35] HANS BETHE: Statistical Theory of Superlattices. – *Proceedings of the Royal Society of London, Series A 150 (871)*, 552–575 (1935).
- [36] RODNEY J. BAXTER: *Exactly Solved Models in Statistical Mechanics*. London: Academic Press, 1982.
- [37] REINHARD DIESTEL: *Graphentheorie*. Vierte Auflage. Heidelberg: Springer, 2010.
- [38] JIŘÍ MATOUŠEK UND JAROSLAV NEŠETŘIL: *Diskrete Mathematik. Eine Entdeckungsreise*. Zweite Auflage. Berlin, Heidelberg: Springer, 2007.
- [39] WALTER OBERSCHELP UND DETLEF WILLE: *Mathematischer Einführungskurs für Informatiker. Diskrete Strukturen*. Stuttgart: Teubner, 1976.
- [40] WILLIBALD DÖRFLER UND JÖRG MÜHLBACHER: *Graphentheorie für Informatiker*. Berlin: Walter de Gruyter, 1973.
- [41] JOHN C. BUTCHER: *The Numerical Analysis of Ordinary Differential Equations*. Chichester: Wiley, 1976.
- [42] ARTHUR CAYLEY: On the theory of the analytical forms called trees. – *Philosophical Magazine XIII*, 172–176 (1857). In: *The collected mathematical papers of Arthur Cayley, Volume III, Paper 203*. Cambridge: At the University Press, 1890.
- [43] DONALD E. KNUTH: *The Art of Computer Programming*. Volumes 3 (Sorting and Searching) and 4A (Combinatorial Algorithms). Upper Saddle River (New Jersey): Addison-Wesley, 2011. Verschiedene Algorithmen daraus wurden von GLENN C. RHOADS in der Programmiersprache C implementiert, siehe <http://mysite.verizon.net/vze16nctz/Code/code.html>, und mit entsprechenden Anpassungen in der vorgelegten Arbeit verwendet.
- [44] EVA KALINOWSKI: A route-determination algorithm for Butcher trees: Delphi implementation. – *The Archives of Theoretical and Applied Informatics*, 12 pages (2005).
- [45] EDWARD R. TUFTE: *Beautiful evidence*. Third printing. Cheshire (Connecticut): Graphics Press, 2010.
- [46] *Intel Itanium Architecture Software Developers Manual. Volume 3: Instruction Set Reference*. Santa Clara (California): Intel Corporation, 2010.
- [47] DANIEL P. SIEWIOREK, C. GORDON BELL, AND ALLEN NEWELL (EDS.): *Computer Structures: Principles and Examples*. New York: McGraw-Hill, 1982.
- [48] ROBERT DONAGHEY AND LOUIS W. SHAPIRO: Motzkin Numbers. – *Journal of Combinatorial Theory, Series A 23*, 291–301 (1977).
- [49] CYRIL DOMB AND MARTIN F. SYKES: On the Susceptibility of a Ferromagnetic above the Curie Point. – *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 240 (1221)*, 214–228 (1957); CYRIL DOMB AND MARTIN F. SYKES: Use of Series Expansions for the Ising Model Susceptibility and Excluded Volume Problem. – *Journal of Mathematical Physics 2 (1)*, 63–67 (1961).
- [50] CHRISTOPHER HUNTER AND BRUNO GUERRIERI: Deducing the Properties of Singularities of Functions from their Taylor Series Coefficients. – *SIAM Journal on Applied Mathematics 39 (2)*, 248–263 (1980).

- [51] DAVID W. CANTRELL: *Inverse gamma function*. Philadelphia (Pennsylvania): The Math Forum, 2001. Elektronisch publiziert unter <http://mathforum.org/kb/message.jspa?messageID=342551>.
- [52] YEDIDYA HILEWITZ AND RUBY B. LEE: A New Basis for Shifters in General-Purpose Processors for Existing and Advanced Bit Manipulations. – *IEEE Transactions on Computers* 58 (8), 1035–1048 (2009).
- [53] THOMAS FLIK: *Mikroprozessortechnik und Rechnerstrukturen*. Siebente Auflage. Berlin: Springer, 2005.
- [54] *Intel Itanium Processor 9500 Series Reference Manual*. Santa Clara (California): Intel Corporation, 2012.
- [55] EBERHARD JAKOBI, NILS BLÜMER, AND PETER VAN DONGEN: Orbital-selective Mott transitions in a doped two-band Hubbard model. – *Physical Review B: Condensed matter and materials physics* 80 (11), 115109, 11 pages (2009); MARKUS GREGER, MARCUS KOLLAR, AND DIETER VOLLHARDT: Emergence of a Common Energy Scale Close to the Orbital-Selective Mott Transition. – *Physical Review Letters* 110 (4), 046403, 5 pages (2013).
- [56] LARS BONNES, KADEN R. A. HAZZARD, SALVATORE R. MANMANA, ANA MARIA REY, AND STEFAN WESSEL: Adiabatic Loading of One-Dimensional $SU(N)$ Alkaline-Earth-Atom Fermions in Optical Lattices. – *Physical Review Letters* 109 (20), 205305, 5 pages (2012); NILS BLÜMER AND ELENA V. GORELIK: Mott transitions in the half-filled $SU(2M)$ symmetric Hubbard model. – *Physical Review B: Condensed matter and materials physics* 87 (8), 085115, 9 pages (2013).
- [57] EVA PAVARINI, ERIK KOCH, DIETER VOLLHARDT, AND ALEXANDER LICHTENSTEIN (Editors): *The LDA+DMFT approach to strongly correlated materials*. Series: *Lecture Notes*. Jülich: Forschungszentrum, 2011.
- [58] GEORG ROHRINGER, ALESSANDRO TOSCHI, ANDREY A. KATANIN, AND KARSTEN HELD: Critical Properties of the Half-Filled Hubbard Model in Three Dimensions. – *Physical Review Letters* 107 (25), 256402, 5 pages (2011).
- [59] Persönliche Mitteilungen von EVA KALINOWSKI.
- [60] MARKUS GREINER, OLAF MANDEL, TILMAN ESSLINGER, THEODOR W. HÄNSCH, AND IMMANUEL BLOCH: Quantum phase transition from a superfluid to a Mott insulator in a gas of ultracold atoms. – *Nature* 415, 39–44 (2002).
- [61] NIKLAS TEICHMANN, DENNIS HINRICHs, MARTIN HOLTHAUS, AND ANDRÉ ECKARDT: Bose-Hubbard phase diagram with arbitrary integer filling. – *Physical Review B: Condensed matter and materials physics* 79 (10), 100503, 4 pages (2009); Process-chain approach to the Bose-Hubbard model: Ground-state properties and phase diagram. – *ibid.* 79 (22), 224515, 14 pages (2009); ANDRÉ ECKARDT: Process-chain approach to high-order perturbation calculus for quantum lattice models. – *ibid.* 79 (19), 195131, 10 pages (2009); CHRISTOPH HEIL AND WOLFGANG VON DER LINDEN: Strong coupling expansion for the Bose-Hubbard and Jaynes-Cummings lattice models. – *Journal of Physics: Condensed Matter* 24 (29), 295601, 14 pages (2012); DENNIS HINRICHs, AXEL PELSTER, AND

- MARTIN HOLTHAUS: Perturbative calculation of critical exponents for the Bose-Hubbard model. – *Applied Physics B: Lasers and Optics* 113 (1), 57–67 (2013).
- [62] MATTHEW P. A. FISHER, PETER B. WEICHMAN, GEOFFREY GRINSTEIN, AND DANIEL S. FISHER: Boson localization and the superfluid-insulator transition. – *Physical Review B: Condensed matter and materials physics* 40 (1), 546–570 (1989).
- [63] EHUD ALTMAN, EUGENE DEMLER, AND ACHIM ROSCH: Mott Criticality and Pseudogap in Bose-Fermi Mixtures. – *Physical Review Letters* 109 (23), 235304, 5 pages (2012).
- [64] JAAN OITMAA, CHRIS HAMER, AND WEIHONG ZHENG: *Series Expansion Methods for Strongly Interacting Lattice Models*. Cambridge: Cambridge University Press, 2010.
- [65] SIVINEE SAWATDIAREE: *Thermodynamics of the Fractional Quantum Hall System: The High-Temperature Expansion Method*. Dissertation. Hannover: Fachbereich Physik der Universität Hannover, 2001; SIVINEE SAWATDIAREE AND WALTER APEL: On the high-temperature expansion for a partially filled lowest Landau level. – *Physica E: Low-dimensional Systems and Nanostructures* 6 (1–4), 75–78 (2000); ALEXANDER L. CHERNYSHEV, DIMITRIOS GALANAKIS, PHILIP PHILLIPS, ALEXANDER V. ROZHKOVA, AND ANDRÉ-MARIE S. TREMBLAY: Higher order corrections to effective low-energy theories for strongly correlated electron systems. – *Physical Review B: Condensed matter and materials physics* 70 (23), 235111, 12 pages (2004); GEORG HAGER, ALEXANDER WEISSE, GERHARD WELLEIN, ERIC JECKELMANN, AND HOLGER FEHSKE: The spin-Peierls chain revisited. – *Journal of Magnetism and Magnetic Materials* 310 (2.2), 1380–1382 (2007); Erratum. – *ibid.* 316 (1), 43 (2007); DANIEL KLAGGES AND KAI PHILLIP SCHMIDT: Constraints on Measurement-Based Quantum Computation in Effective Cluster States. – *Physical Review Letters* 108 (23), 230508, 5 pages (2012).
- [66] JÜRGEN STEIN: Flow equations and the strong-coupling expansion for the Hubbard model. – *Journal of Statistical Physics* 88 (1–2), 487–511 (1997); ALEXANDER REISCHL, ERWIN MÜLLER-HARTMANN, AND GÖTZ S. UHRIG: Systematic mapping of the Hubbard model to the generalized t - J model. – *Physical Review B: Condensed matter and materials physics* 70 (24), 245124, 15 pages (2004); SIMONE A. HAMERLA, SEBASTIAN DUFFE, AND GÖTZ S. UHRIG: Derivation of the t - J model for finite doping. – *ibid.* 82 (23), 235117, 20 pages (2010).
- [67] STEFAN KEHREIN: *The Flow Equation Approach to Many-Particle Systems*. Series: *Series: Springer Tracts in Modern Physics* 217. Berlin: Springer, 2006.
- [68] HONG-YU YANG, ANDREAS M. LÄUCHLI, FRÉDÉRIC MILA, AND KAI PHILLIP SCHMIDT: Effective Spin Model for the Spin-Liquid Phase of the Hubbard Model on the Triangular Lattice. – *Physical Review Letters* 105 (26), 267204, 4 pages (2010).
- [69] HONG-YU YANG AND KAI P. SCHMIDT: Effective models for gapped phases of strongly correlated quantum lattice models. – *Europhysics Letters* 94 (1), 17004, 6 pages (2011).

- [70] Persönliche Mitteilung von NEIL SLOANE.
- [71] ROBERT SEDGEWICK AND KEVIN WAYNE: *Algorithms*. Fourth edition. Upper Saddle River (New Jersey): Addison-Wesley, 2011.
- [72] THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST, AND CLIFFORD STEIN: *Introduction to Algorithms*. Second Edition. Cambridge (Massachusetts): MIT Press, 2001.
- [73] <http://www.top500.org/list/2014/06>.
- [74] GENE M. AMDAHL: Validity of the single processor approach to achieving large scale computing capabilities. – *American Federation of Information Processing Societies (AFIPS): Proceedings of the spring joint computer conference 30*, 483–485 (1967); GEORG HAGER AND GERHARD WELLEIN: *Introduction to High Performance Computing for Scientists and Engineers*. Series: HORST D. SIMON (Editor): *Chapman & Hall / CRC Computational Science*. Boca Raton (Florida): CRC Press, 2010.
- [75] Persönliche Mitteilung von ALOIS P. HEINZ.
- [76] ALEXANDER STEPANOV AND MING LEE: *The Standard Template Library*. Technical report. Hewlett-Packard Laboratories, 1995.
- [77] JAMES MCCAFFREY: *Using Permutations in .NET for Improved Systems Security*. Redmond (Washington): Microsoft Corporation, 2003. Elektronisch publiziert unter <http://msdn.microsoft.com/library/Aa302371>.
- [78] MARTIN PAECH, WALTER APEL, EVA KALINOWSKI, AND ERIC JECKELMANN: Comparison of computer-algebra strong-coupling perturbation theory and dynamical mean-field theory for the Mott-Hubbard insulator in high dimensions. – *Physical Review B: Condensed matter and materials physics 90 (24)*, 245147, 10 pages (2014).

Datenverarbeitungsressourcen

- [79] *The Haskell Programming Language*. <http://www.haskell.org>.
- [80] *Mathematica. Version 9.0*. Champaign (Illinois): Wolfram Research, Inc., 2012.
- [81] *Splint – annotation-assisted static program checker. Version 3.1.2*. © 1994–2007 by the Secure Programming Group, University of Virginia and Massachusetts Institute of Technology. <http://www.splint.org>.
- [82] TORBJÖRN GRANLUND and the GMP development team: *The GNU Multiple Precision Arithmetic Library. Version 5.0.5*. 2012. <http://gmplib.org>

- [83] JOS VERMASEREN et al.: *FORM – Symbolic Manipulation System*.
<http://www.nikhef.nl/~form>;
 persönliche Mitteilung von ALEXANDER WEISSE.

- [84] *CISC-server*, Typ HP AM447A, bereitgestellt von Hewlett-Packard:

Rechner	Hewlett-Packard ProLiant DL980 G7
Prozessoren	acht zu je 10 Kernen, Typ Intel Xeon E7-4870
Hauptspeicher	128 GWorte (1024 GBytes)
Betriebssystem	Red Hat Enterprise Linux Version 6.4
Kompiler	Intel C++ Composer XE 2013 Version 13.1.2
Stand	Juni 2013

- [85] *RISC-workstation*, Typ HP A9638A:

Rechner	Hewlett-Packard System 9000 J6750
Prozessoren	zwei, Typ PA-8700+
Hauptspeicher	2 GWorte (16 GBytes)
Betriebssystem	HP-UX Version B.11.11
Kompiler	ANSI-C B3899BA/B3901BA Version B.11.11.16
Stand	März 2009

- [86] *EPIC-server*, Typ HP AH353A, bereitgestellt von Hewlett-Packard:

Rechner	Hewlett-Packard Integrity Superdome 2-32s
Prozessoren	32 zu je 8 Kernen, Typ Intel Itanium 9560
Hauptspeicher	512 GWorte (4096 GBytes)
Betriebssystem	HP-UX Version B.11.31
Kompiler	ANSI-C B9007AA Version C.11.31.07
Stand	Juli 2013

Bedingt durch die besonders auf niedrige Speicherantwortzeiten und hohe Speicherbandbreiten optimierte Architektur hat dieser Rechner die etwa 2,8fache Leistung einer halben Partition (248 Kerne) von [88] mit *memory interleaving* und reinem *non-uniform memory access*.

Für Skalierungstests und die Berechnung der Glieder diverser Ganzzahlfolgen wurde zudem ein *software*-seitig identisch konfigurierter *computeserver* vom Typ HP AH337A verwendet:

Rechner	Hewlett-Packard Integrity Superdome 2-16s
Prozessoren	16 zu je 8 Kernen, Typ Intel Itanium 9560
Hauptspeicher	192 GWorte (1536 GBytes)
Betriebssystem	HP-UX Version B.11.31
Kompiler	ANSI-C B9007AA Version C.11.31.07
Stand	April 2014

[87] *cisc-server*, Typ IBM 7233, bereitgestellt vom Institut für Theoretische Physik der Universität Hannover:

Rechner	IBM System x3950 M2, vier Knoten mit ScaleXpander
Prozessoren	16 zu je 6 Kernen, Typ Intel Xeon X7460
Hauptspeicher	64 GWorte (512 GBytes)
Betriebssystem	Debian GNU/Linux Version 6.0
Kompiler	Intel C++ Composer XE 2011 Version 12.0.0
Stand	März 2011

[88] *cisc-server*, Typ SGI UV1000, bereitgestellt vom Norddeutschen Verbund für Hoch- und Höchstleistungsrechnen (HLRN-II):

Rechner	SGI Altix UltraViolet 1000 (eine Partition)
Prozessoren	64 zu je 8 Kernen, Typ Intel Xeon X7560 62 (496 Kerne) für Anwendungen nutzbar
Hauptspeicher	256 GWorte (2048 GBytes)
Betriebssystem	SUSE Linux Enterprise Server Version 11 SP1
Kompiler	Intel C++ Composer XE 2011 Version 12.0.0
Stand	Juli 2011

- [89] EPIC-server; Typ SGI 4700, testhalber bereitgestellt vom Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften („Höchstleistungsrechner Bayern II“, HLRB-II):

Rechner	SGI Altix 4700 (eine Partition)
Prozessoren	256 zu je 2 Kernen, Typ Intel Itanium 9040 255 (510 Kerne) für Anwendungen nutzbar
Hauptspeicher	257 GWorte (2056 GBytes)
Betriebssystem	SUSE Linux Enterprise Server Version 11 SP1
Kompiler	Intel C++ Composer XE 2011 Version 12.0.0
Stand	Juni 2011

- [90] CISC-server; Typ HP AT147A, bereitgestellt von Hewlett-Packard:

Rechner	Hewlett-Packard Integrity Superdome X
Prozessoren	16 zu je 15 Kernen, Typ Intel Xeon E7-2890v2
Hauptspeicher	1536 GWorte (12288 GBytes)
Betriebssystem	Red Hat Enterprise Linux Version 6.5
Kompiler	GCC Version 4.9.1
Stand	August/September 2014

Danksagung

Herrn Prof. Dr. ERIC JECKELMANN, meinem Doktorvater, möchte ich herzlichen Dank aussprechen, daß er mir die Möglichkeit zur Promotion gegeben, den Rahmen dafür geschaffen und mir vielfältige Unterstützung bei der Durchführung meiner Projekte gewährt hat. Auch sein Vertrauen zu mir als sein Assistent in manch verantwortungsvoller Aufgabe empfinde ich als Ehre. Herrn Priv.-Doz. Dr. WALTER APEL, meinem Koreferenten, gebührt nicht minder Dank für seine ebenso unermüdliche wie zeitaufwendige Betreuung insbesondere des zweiten Teiles dieser Arbeit und sein lebhaftes Interesse an diesem Thema. Frau Prof. Dr. RITA LOOGEN hat nicht nur durch zahlreiche Beratungen über die zur Diskreten Mathematik und Theoretischen Informatik gehörenden Aspekte den zweiten Teil gefördert, sondern auch diese Anwendung in ihre eigene Forschung einbezogen, und ich danke ihr für beides sehr. Neben ihrer kreativen Begabung für die Methodenentwicklung und ihrem wissenschaftlichen Elan hat Frau Dr. EVA KALINOWSKI durch Erläuterung der anspruchsvollen Details ihres Verfahrens diese Arbeit umfassend unterstützt, wofür ich ihr sehr dankbar bin. Daß Herr Prof. Dr. HERBERT PFNÜR spontan meiner Bitte um Übernahme des Vorsitzes der Promotionskommission entsprochen hat, war eine erhebliche Entlastung bei der Organisation des Verfahrens. Vielmaligen Dank schulde ich ferner *Hewlett-Packard* für die besondere Gunst, daß ich mehr als ein Jahr lang mit großen Freiheiten über einige der neuesten und leistungsfähigsten symmetrischen Multiprozessorrechner verfügen durfte.

Zu einer gediegenen Ausdrucksweise hinsichtlich der Formulierung mathematischer Gedanken hat mich Herr Dr. ALEXANDER SEEL kraft seines breiten fachlichen Wissens und hohen didaktischen Geschickes angehalten; bezüglich der Deutschen Sprache hat dafür mit großem Können Frau Pfarrerin ASTRID-MARIA RÄDER gesorgt, und der Stilfragen im Englischen hat sich zu meiner Freude Frau VIRGINIA APEL angenommen. Allen dreien danke ich herzlich für ihre nie fehlende Freundlichkeit und ihre Geduld mit dem zum Teil mehrfachen Korrekturlesen dieser umfangreichen Arbeit. Mein Dank richtet sich überdies an meinen Patenonkel EBERHARD BEDKE für seine großzügige Unterstützung des Druckes dieser Dissertation.

Von Herzen danke ich auch Herrn Prof. Dr. WOLFGANG BESTGEN, der mir bereits in Marburg einen heutzutage einzigartigen Zugang zu Wesen und Reichtum der Theoretischen Physik vermittelt hat und mir ein prägendes Vorbild für die nüchterne wie präzise Herangehensweise der „Titanen“ dieses Faches geworden ist. Dankbar bin ich zudem

Herrn Prof. Dr. HANS-ULRICH EVERTS, der in Hannover meine Arbeit in ungezählten ermutigenden Gesprächen mit hoher Aufmerksamkeit verfolgt und oft meinen Blick auf die erfrischende Vielfalt der modernen Physik gelenkt hat. Besonders würdigen möchte ich Herrn Dr. MARIO PUCHNER, der mir seit dem Anfang unseres Chemiestudiums in Marburg und somit auch auf den schwierigen Abschnitten meiner Doktorandenzeit stets seine Freundschaft erwiesen hat.

Meinen Eltern verdanke ich jedoch hinsichtlich der Grundlagen vor allen anderen, diese Arbeit vorlegen zu können. Ohne deren nicht zu ermessende Fürsorge und unfaßbare Geduld wäre ein derart an Erkenntnissen und Erfahrungen reicher Weg durch zwei Naturwissenschaften unmöglich gewesen.

Wissenschaftlicher Werdegang

MARTIN PAECH, Diplom-Chemiker, geboren am 4. September 1974 in Pasewalk in Vorpommern, deutscher Staatsangehöriger; **1974** Übersiedlung nach Westdeutschland; **1995** Allgemeine Hochschulreife mit Latinum am Gymnasium der August-Hermann-Francke-Schule in Gießen; **1995 – 2000** Chemiestudium an der Philipps-Universität Marburg; **1997** Grundpraktikum in Kernchemie, Diplom-Vorprüfung, **1998** Forschungspraktikum in Anorganischer Chemie an der Eidgenössischen Technische Hochschule Zürich (Prof. Dr. REINHARD NESPER, *Synthese und Strukturaufklärung neuer Zintl-Phasen im ternären System Barium bzw. Strontium, Silizium, Jod*; Priv.-Doz. Dr. THOMAS F. FÄSSLER, *Synthese und NMR-spektroskopische Untersuchung anionischer Zinn- und Germanium-Cluster*), **1998/1999** Forschungspraktikum in Theoretischer Chemie an der Universität Zürich (Prof. Dr. WALTER THIEL, *Evaluation einer GUGA-CI-Implementierung in MNDO durch Berechnung elektronisch angeregter Zustände einfacher organischer Moleküle*); **2000 – 2003** Studium der Physik an der Université de Fribourg und an der Universität Marburg; **2000** Diplomarbeit in Experimenteller und Theoretischer Physikalischer Chemie in Fribourg (*Dissoziative Elektronenanlagerung an Phenylazid*, betreut von Prof. Dr. MICHAEL ALLAN), **2001 – 2004** Gast der Arbeitsgruppen Vielteilchentheorie (Prof. Dr. FLORIAN GEBHARD) und Didaktik der Physik (Prof. Dr. WOLFGANG BESTGEN) in Marburg, **2003** Diplom in Chemie mit Hauptfach Physikalische Chemie und Nebenfach Theoretische Physik an der Université de Fribourg; **2006 – 2014** Doktorarbeit in Theoretischer Physik an der Gottfried Wilhelm Leibniz Universität Hannover (*Numerische und algebraisch-graphentheoretische Algorithmen für korrelierte Quantensysteme*, betreut von Prof. Dr. ERIC JECKELMANN).

Anstellungen: **1997/1998** Studentische Hilfskraft am Fachbereich Chemie der Universität Marburg (Chemisches Praktikum für Studierende der Biologie, Prof. Dr. HANS GÜNTHER AURICH); **1998/1999** Hilfsassistent im Rahmen eines Forschungsprojekts am Departement Chemie der ETH Zürich (Prof. Dr. REINHARD NESPER, *CPMD-Rechnungen zur Oxidation von Wolframcarbid-Oberflächen*); **2003/2004** Wissenschaftliche Hilfskraft mit Abschluß am Fachbereich Physik der Universität Marburg (Lehrstuhl Physik komplexer Systeme, Prof. Dr. BRUNO ECKHARDT); **2006 – 2012** Wissenschaftlicher Angestellter in der Fakultät für Mathematik und Physik der Universität Hannover (Arbeitsgruppe Theorie Kondensierter Materie, Assistent von Prof. Dr. ERIC JECKELMANN); seit **2009** Wissenschaftlicher Berater des neugegründeten Rechenzentrums der Hochschule für Informatik und Management, Bielsko-Biała (Polen).