

Transfermatrixrenormierung fern vom statistischen Gleichgewicht

Tilman Enß

Diplomarbeit am Lehrstuhl Prof. Wagner
Betreuer: PD Dr. Ulrich Schollwöck

Januar 2002

Sektion Physik
Ludwig-Maximilians-Universität München

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 2 | Stochastische Systeme | 3 |
| 2.1 | Mastergleichung | 3 |
| 2.1.1 | Kontinuierliche Zeit | 4 |
| 2.1.2 | Diskrete Zeit | 5 |
| 2.2 | TROTTER-SUZUKI-Zerlegung | 6 |
| 2.3 | <i>Exkurs</i> : Ein einfacher Reaktions-Diffusions-Prozeß | 8 |
| 2.4 | Zeitentwicklungsmatrix | 10 |
| 2.5 | Transfermatrix | 11 |
| 2.5.1 | Blockstruktur | 13 |
| 2.5.2 | Explizite Struktur der Transfermatrix | 13 |
| 2.5.3 | Zustandssumme und Erwartungswerte | 18 |
| 3 | Renormierung | 21 |
| 3.1 | Historische Einführung | 21 |
| 3.1.1 | Exakte Diagonalisierung | 22 |
| 3.1.2 | Grundidee der Renormierungsgruppe | 23 |
| 3.1.3 | Numerische Renormierungsgruppe | 24 |
| 3.2 | Dichtematrix-Renormierungsgruppe (DMRG) | 26 |
| 3.2.1 | Dichtematrix-Projektion | 26 |
| 3.2.2 | DMRG-Algorithmus | 30 |
| 3.2.3 | Stochastische DMRG | 31 |
| 3.3 | Transfermatrix-Renormierungsgruppe (TMRG) | 32 |
| 3.3.1 | Quanten-TMRG | 33 |
| 3.3.2 | Stochastische TMRG | 35 |
| 3.3.3 | Die asymmetrische Dichtematrix | 36 |
| 3.3.4 | Die symmetrische Dichtematrix | 37 |
| 3.3.5 | Alternativen | 40 |
| 4 | TMRG-Ergebnisse für Reaktions-Diffusions-Prozesse | 42 |
| 4.1 | Paar-Annihilation | 42 |

| | | |
|----------|--|-----------|
| 4.2 | <i>Branching-fusing</i> -Prozeß | 45 |
| 4.3 | Approximationsfehler | 46 |
| 4.4 | Spektrum der Dichtematrix | 47 |
| 4.5 | Spektrum der Transfermatrix | 48 |
| 4.6 | Methode der expliziten Eigenvektoren | 49 |
| 5 | Implementierung des TMRG-Algorithmus | 51 |
| 5.1 | Beschreibung des Algorithmus | 51 |
| 5.1.1 | Initialisierung | 52 |
| 5.1.2 | Eigenvektoren und Erwartungswerte | 54 |
| 5.1.3 | Wachstum des Gitters | 57 |
| 5.1.4 | Renormierung | 59 |
| 5.2 | TMRG-Programm | 62 |
| 5.2.1 | Die Programmiersprache Python | 62 |
| 5.2.2 | Matrizenrechnung in Python | 63 |
| 5.2.3 | Programmcode | 65 |
| 6 | Zusammenfassung und Ausblick | 72 |
| | Literaturverzeichnis | 75 |

1 Einleitung

In der statistischen Physik beschreibt man physikalische Systeme mit sehr vielen Freiheitsgraden. Ein Beispiel für Wechselwirkungen sind Reaktions-Diffusions-Prozesse: man betrachtet eine große Zahl von Teilchen, die nach vorgegebenen Regeln miteinander reagieren und diffundieren können. Eine wesentliche Frage ist, welches makroskopische (kollektive) Verhalten aus den mikroskopischen Reaktionsregeln folgt.

Eine Reihe von speziellen Systemen sind trotz ihrer vielen Freiheitsgrade exakt lösbar (z.B. mit dem BETHE-Ansatz), aber die meisten Reaktions-Diffusions-Prozesse entziehen sich einer solchen Behandlung. Um das Problem zumindest numerisch behandeln zu können, diskretisiert man zunächst den Raum, d.h. man betrachtet Teilchen auf einem räumlichen Gitter. Jeder Gitterplatz hat verschiedene Besetzungszustände, je nachdem, welche Sorte Teilchen diesen Platz einnimmt (Kapitel 2).

Wenn die Teilchen wechselwirken (reagieren), dann muß man alle Korrelationen zwischen Zuständen an unterschiedlichen Gitterplätzen berücksichtigen, um die Zeitentwicklung des Systems anzugeben. Der Zustandsraum ist das Tensorprodukt der n -dimensionalen Zustandsräume jedes Gitterplatzes, mit Dimension n^L . Diese Dimension wächst mit der Gittergröße L so schnell an, daß man nur sehr kleine Systeme exakt behandeln kann.

Um trotzdem das Verhalten großer Systeme untersuchen zu können, insbesondere den thermodynamischen Limes unendlich großer Systeme, haben WILSON, KADANOFF und andere die Idee der Renormierung entwickelt. Oft lassen sich Systeme mit vielen Freiheitsgraden durch eine effektive Theorie mit wenigen Freiheitsgraden beschreiben, wenn man sich nur für bestimmte Aspekte des Systems interessiert, z.B. für das skalenunabhängige Verhalten (kritische Phänomene). Damit hilft die Renormierung bei der Modellbildung, indem sie aufzeigt, welche der vielen Freiheitsgrade physikalisch bedeutsam sind für eine konkrete Fragestellung (Kapitel 3).

Diese Ideen wandte WHITE 1992 auf eindimensionale Quanten-Spinketten an: auch hier wächst die Dimension des Hilbertraumes exponentiell mit der Länge der Kette, der Grundzustand läßt sich also nur für sehr kurze Ketten exakt berechnen. Seine numerische Methode der *Dichtematrix-Renormierungsgruppe* (DMRG) liefert eine Vorschrift, wie man bei vorgegebener Dimension denjenigen Unterraum des Hilbertraumes findet, in dem sich der Grundzustand angemessen darstellen läßt. Indem man auf diese Weise die Dimension des Hilbertraumes beschränkt, bleibt das Problem numerisch handhab-

bar, und man kann das Gitter schrittweise immer weiter vergrößern. Damit erhält man Ergebnisse in nicht dagewesener Genauigkeit; die DMRG hat die numerische Methodik auf diesem Gebiet revolutioniert.

Die wesentliche Idee ist, daß man den gesuchten Grundzustand in Form einer Dichtematrix des Systems und seiner Umgebung darstellt. Die reduzierte Dichtematrix des Systems, die man durch eine partielle Spur über die Umgebung bildet, beschreibt das System als Teil des Universums in der Regel in einem gemischten Zustand. Wählt man aus diesem gemischten Zustand die m wichtigsten Bestandteile aus, so hat man den Zustandsraum effektiv auf m Dimensionen reduziert.

Das Ziel der vorliegenden Arbeit ist es nun, dieses sehr erfolgreiche Verfahren von Quantensystemen auf stochastische Systeme — wie die anfangs eingeführten Reaktions-Diffusions-Prozesse — zu übertragen. Anstelle der Dichtematrix verwenden wir jedoch die Transfermatrix, die es erlaubt, den thermodynamischen Limes exakt zu bilden. Beide Systeme haben gemeinsam, daß ihr Zustandsraum exponentiell mit der Gittergröße anwächst, unterscheiden sich aber wesentlich in ihren Randbedingungen — unter Randbedingungen verstehen wir hier sowohl die Bedingungen am linken und rechten Rand des Gitters als auch die Konfigurationen am Anfang und Ende der Zeitentwicklung. Daher kann man den DMRG-Algorithmus nicht einfach für die Renormierung der Transmatrix übernehmen, denn wie wir erstmals exakt bewiesen haben, wird dann die physikalische Wechselwirkung (Reaktion der Teilchen) weggemittelt. Stattdessen muß man überlegen, welche andere Form der Dichte- bzw. Transfermatrix den Zustand des stochastischen Systems mit seinen spezifischen Randbedingungen physikalisch sinnvoll repräsentiert. Ein zentrales Ergebnis der vorliegenden Arbeit ist die erstmalige explizite Konstruktion von Eigenvektoren der Transfermatrix, mit der sich unter gewissen Voraussetzungen beweisen läßt, welche Form der Dichtematrix für stochastische Systeme *optimal* ist.

Nach dieser fundierten theoretischen Begründung testen wir unsere Aussagen an zwei konkreten Reaktions-Diffusions-Prozessen (Kapitel 4). Im einen Fall, dem Paar-Annihulations-Prozeß, ist die Lösung exakt bekannt, so daß wir die Simulationsergebnisse mit verschiedenen Formen der Dichtematrix auch quantitativ vergleichen können. Im zweiten Fall, dem *branching-fusing*-Prozeß, untersuchen wir das Kurzzeitverhalten, das mit anderen Methoden wie Monte-Carlo-Rechnungen nur schwer zugänglich ist.

Schließlich gehen wir ausführlich auf die Details der Implementierung des Algorithmus für die Transfermatrix-Renormierungsgruppe (TMRG) und auf wesentliche Optimierungen ein (Kapitel 5). Ein komplett lauffähiges und schnelles Programm in nur 120 Zeilen Programmcode und mit ausführlichen Kommentaren demonstriert die Vorteile der Programmiersprache Python für wissenschaftliche Anwendungen: es ist sehr leicht erweiterbar und kann als Grundlage für künftige Projekte dienen. Abschließend fassen wir die wichtigsten Ergebnisse der vorliegenden Arbeit zusammen und geben einen Ausblick auf vielversprechende weitere Entwicklungen (Kapitel 6).

2 Stochastische Systeme

2.1 Mastergleichung

Wir betrachten stochastische Systeme auf einem räumlich eindimensionalen (periodischen) Gitter der Länge L . Jeder Gitterplatz habe n mögliche Zustände, z.B. $n = 2$, wenn jede Zelle entweder tot oder lebendig sein kann. Eine Konfiguration des Gitters beschreiben wir als Zahlenfolge $(\sigma_i)_{i=1,\dots,L}$, wobei i der Gitterplatz ist und $\sigma_i \in \{1, \dots, n\}$. Der Zustand des gesamten Gitters ist durch die Wahrscheinlichkeiten aller n^L möglichen Konfigurationen gegeben. Sie sind im n^L -dimensionalen Zustandsvektor $|P(t)\rangle = (p_1(t), \dots, p_{n^L}(t))$ zusammengefaßt, der zusätzlich von der Zeit abhängen kann. Für $n = 2$ und $L = 2$ gibt es z.B. $n^L = 4$ mögliche Konfigurationen des Gitters, $(\sigma_i) \in \{(11), (12), (21), (22)\}$, also ist $|P(t)\rangle = (p_{(11)}(t), p_{(12)}(t), p_{(21)}(t), p_{(22)}(t))$.

Jede der Wahrscheinlichkeiten liegt zwischen 0 und 1,

$$p_{(\sigma_i)} \in [0, 1]. \quad (2.1)$$

Die Summe dieser Wahrscheinlichkeiten, d.h. die Wahrscheinlichkeit, daß sich das System in *irgendeinem* Zustand befindet, ist für alle Zeiten auf Eins normiert:

$$\langle 1|P(t)\rangle = \sum_{(\sigma_i)} p_{(\sigma_i)}(t) = 1, \quad \text{mit } \langle 1| = (1, \dots, 1) \quad (\text{Summe über alle Konfig.}) \quad (2.2)$$

Diese Eigenschaft der Wahrscheinlichkeitserhaltung bei stochastischen Systemen bezeichnen wir im folgenden als „stochastische Randbedingungen“ (oder synonym als offene Randbedingungen): man kann sie verstehen als Bedingung am Ende der Zeitentwicklung. Analog gibt es eine Anfangsbedingung $|P(0)\rangle$, auf die wir später eingehen werden. Unter Randbedingungen verstehen wir allgemein sowohl die Bedingungen an den räumlichen Rändern des Gitters sowie an den *zeitlichen* Rändern seiner Zeitentwicklung.

2.1.1 Kontinuierliche Zeit

Da in der klassischen Physik Wahrscheinlichkeiten additiv sind, ist ihre zeitliche Entwicklung linear und läßt sich durch eine Mastergleichung beschreiben:

$$\frac{d}{dt}|P(t)\rangle = -\mathcal{H}|P(t)\rangle. \quad (2.3)$$

Sie besagt, daß die Änderung der Wahrscheinlichkeit einer Konfiguration gleich dem Wahrscheinlichkeitsstrom von anderen Konfigurationen in diese Konfiguration ist, ist also eine Kontinuitätsgleichung. Dabei gibt $-\mathcal{H}$ die Übergangsraten zwischen Konfigurationen an. Da die Wahrscheinlichkeit eine erhaltene Größe ist (Gleichung (2.2)), gilt

$$\langle\tau| - \mathcal{H}|\sigma\rangle = w(\sigma \mapsto \tau) \quad (\tau \neq \sigma) \quad (2.4a)$$

$$\langle\sigma| - \mathcal{H}|\sigma\rangle = - \sum_{\tau \neq \sigma} w(\sigma \mapsto \tau) \quad (2.4b)$$

mit den Mikrozuständen (Gitterkonfigurationen) $|\sigma\rangle$ und $|\tau\rangle$ und Übergangsraten $w(\sigma \mapsto \tau)$, die sich aus den Ratengleichungen ergeben. Schreibt man \mathcal{H} als Matrix auf dem Raum der Konfigurationen und $|P(t)\rangle$ als Spaltenvektor, so bedeutet die Bedingung (2.4), daß die Summe innerhalb jeder Spalte von \mathcal{H} verschwindet. Aus Gleichung (2.1) folgt ferner, daß dieser „stochastische Hamiltonoperator“ positiv semidefinit ist (bzw. bei komplexen Eigenwerten E_i ist der Realteil $\text{Re}(E_i) \geq 0$), denn sonst könnte die Wahrscheinlichkeit einer Konfiguration unbeschränkt anwachsen [vK97]. Eine solche Matrix $-\mathcal{H}$ wird als stochastische Matrix bezeichnet.

Wir betrachten hier nur zeitlich konstante \mathcal{H} , die Lösung der Mastergleichung (2.3) ist dann

$$|P(t)\rangle = e^{-t\mathcal{H}}|P(0)\rangle =: U(t, 0)|P(0)\rangle, \quad (2.5)$$

wobei $U(t, 0)$ den Zeitentwicklungsoperator bezeichnet. Im Limes langer Zeiten projiziert $U(t, 0)$ auf den stationären Eigenraum mit Eigenwert 0, da alle anderen Eigenräume von \mathcal{H} mit positiven Eigenwerten exponentiell mit der Zeit kontrahiert werden.

Damit die Wahrscheinlichkeit (2.2) in der Zeitentwicklung erhalten bleibt, muß $\langle 1|$ ein linker Eigenvektor von \mathcal{H} zum Eigenwert 0 sein:

$$\frac{d}{dt}\langle 1|P(t)\rangle = -\langle 1|\mathcal{H}|P(t)\rangle = 0 \quad \forall |P(t)\rangle \implies \langle 1|\mathcal{H} = 0. \quad (2.6)$$

Dies ist offensichtlich äquivalent zu Gleichung (2.4). Im Limes langer Zeiten gelangt das System in den stationären Zustand

$$|\text{stat}\rangle = \lim_{t \rightarrow \infty} |P(t)\rangle. \quad (2.7)$$

Dieser Zustand liegt im stationären Eigenraum, d.h. dem rechten Eigenraum zum Eigenwert 0, denn für jeden Zustand (Vektor) $|\text{stat}\rangle$ in diesem Eigenraum gilt $\frac{d}{dt}|\text{stat}\rangle = -\mathcal{H}|\text{stat}\rangle = 0$. Die Dimension des stationären Eigenraumes kann größer als Eins sein, dann hängt der stationäre Zustand $|\text{stat}\rangle$ vom Anfangszustand $|P(0)\rangle$ ab.

Die Energie-Eigenwerte bestimmen die zeitliche Annäherung an den stationären Zustand. Entscheidend ist die Energielücke

$$\Gamma = \inf_i \text{Re}(E_i) \quad (i \neq 0) \quad (2.8)$$

des ersten angeregten Zustands über dem Grundzustand ($E_0 = 0$). Ist $\Gamma > 0$, so nähert sich das System dem stationären Zustand exponentiell mit einer Relaxationszeit $\tau = 1/\Gamma$, ist dagegen $\Gamma = 0$, so kann der Abfall algebraisch sein.

Offensichtlich ist \mathcal{H} nicht symmetrisch, wenn $w(\sigma \mapsto \tau) \neq w(\tau \mapsto \sigma)$; dann gilt $|\text{stat}\rangle \neq \langle 1|^T$. Wenn dennoch die Bedingung der „*detailed balance*“ erfüllt ist,

$$w(\sigma \mapsto \tau) \cdot \langle \sigma | \text{stat} \rangle = w(\tau \mapsto \sigma) \cdot \langle \tau | \text{stat} \rangle, \quad (2.9)$$

dann kann man \mathcal{H} mit einer Ähnlichkeitstransformation in eine symmetrische Matrix überführen, die zwar nicht mehr stochastisch gemäß (2.4) ist, aber immer noch eine lokale Wechselwirkung beschreibt. *Detailed balance* ist z.B. gegeben, wenn es außer der Diffusion nur eine einzige weitere reversible Reaktion gibt, oder unter sehr speziellen Voraussetzungen auch bei mehreren reversiblen Reaktionen. Im Falle der *detailed balance* kann man den stationären Zustand $|\text{stat}\rangle$ leicht aus $\langle 1|$ berechnen. Wir beschränken uns hier allerdings nicht auf solche nahezu symmetrischen (zeitumkehrinvarianten) Hamiltonoperatoren, sondern betrachten echte Nichtgleichgewichtsprozesse.

Als Anfangsbedingung werden wir später hauptsächlich einen unkorrelierten Zustand brauchen,

$$|P(0)\rangle = (p_{(\sigma_i)}) \quad \text{mit} \quad p_{(\sigma_i)} = \prod_{i=1}^L w(\sigma_i) = w(\sigma_1) \cdots w(\sigma_L), \quad (2.10)$$

d.h. an jedem Gitterplatz ist die Besetzungswahrscheinlichkeit $w(\sigma)$ für den Zustand σ unabhängig vom Zustand der benachbarten Gitterplätze. Aus $\langle 1|P(0)\rangle = 1$ folgt

$$\sum_{\sigma=1}^n w(\sigma) = 1. \quad (2.11)$$

2.1.2 Diskrete Zeit

Im Gegensatz zu den betrachteten Systemen, die sich gemäß der Mastergleichung (2.3) in kontinuierlicher Zeit entwickeln, gibt es auch Systeme, die sich in diskreten Zeitschritten

entwickeln, z.B. zelluläre Automaten. Sie sind durch

$$|P(t + \Delta t)\rangle = u|P(t)\rangle, \quad (2.12)$$

definiert, wobei u der Zeitentwicklungsoperator (analog (2.5)) für einen diskreten Zeitschritt Δt ist: er enthält die *Übergangswahrscheinlichkeiten* von einer Konfiguration zur nächsten. Auch bei kontinuierlichen Systemen kann eine Betrachtung nur zu diskreter Zeit angemessen sein: wenn man z.B. eine Population von Tieren betrachtet, so sind Zeitschritte kleiner als die Tragzeit nicht sinnvoll. Zudem können diskrete Zeitschritte die numerische Behandlung vereinfachen, wie wir im nächsten Abschnitt sehen werden.

2.2 Trotter-Suzuki-Zerlegung

Vermittelt der Hamiltonoperator nur eine kurzreichweitige Wechselwirkung zwischen benachbarten Gitterplätzen, so können wir mit Hilfe der TROTTER-SUZUKI-Zerlegung die räumlich globale auf die lokale Zeitentwicklung zurückführen [Tro59, Suz85a, Suz85b]. Bei Wechselwirkung nur zwischen nächsten Nachbarn i und $i+1$ kann man \mathcal{H} als Summe lokaler 2-site-Hamiltonoperatoren $h_{i,i+1}$ schreiben. Diese Zerlegung gilt zwar allgemein für eindimensionale Gitter, da wir jedoch später nur periodische Gitter betrachten, beschränken wir uns schon jetzt auf diesen Fall, was die Notation vereinfacht (dann ist Gitterplatz $L+1$ identifiziert mit Gitterplatz 1). Diese lokalen $h_{i,i+1}$ fassen wir anschließend so zu $\mathcal{H}_{\text{even}}$ und \mathcal{H}_{odd} zusammen, daß die Zeitentwicklung innerhalb jedes der Teiloperatoren entkoppelt (siehe Gleichung (2.16) weiter unten):

$$\mathcal{H} = \sum_{i=1}^L h_{i,i+1} = \left(\sum_{i \text{ gerade}} h_{i,i+1} \right) + \left(\sum_{i \text{ ungerade}} h_{i,i+1} \right) = \mathcal{H}_{\text{even}} + \mathcal{H}_{\text{odd}}. \quad (2.13)$$

$\mathcal{H}_{\text{even}}$ und \mathcal{H}_{odd} kommutieren wegen der NN-Wechselwirkung nicht. Der Zeitentwicklungsoperator ist dann durch die TROTTER-Formel gegeben, die für die Exponentialfunktion einer Summe nichtkommutierender Matrizen gilt,

$$U(t, 0) = e^{-t(\mathcal{H}_{\text{even}} + \mathcal{H}_{\text{odd}})} = \lim_{M \rightarrow \infty} \left(\underbrace{e^{-(t/M)\mathcal{H}_{\text{odd}}}}_{u_{\text{odd}}} \times \underbrace{e^{-(t/M)\mathcal{H}_{\text{even}}}}_{u_{\text{even}}} \right)^M. \quad (2.14)$$

Bildet man den Limes nicht exakt, sondern wählt einendliches M , so macht man einen Fehler, der durch den Kommutatorterm nächsthöherer Ordnung, den man weggelassen hat, bestimmt ist:

$$[u_{\text{even}}, u_{\text{odd}}] = \mathcal{O}\left(\left(\frac{1}{M}\right)^2\right) \rightarrow 0 \quad \text{für } M \rightarrow \infty. \quad (2.15)$$

Die Idee ist, daß man die tatsächliche Zeitentwicklung am besten approximiert, wenn man die geraden und ungeraden Paare möglichst gleichberechtigt behandelt. Die Zeitentwicklung besteht daher abwechselnd aus einem Zeitschritt, in dem alle geraden Gitterplätze mit ihrem rechten Nachbarn wechselwirken (u_{even}), und anschließend einem Zeitschritt, in dem die dazwischenliegenden Paare wechselwirken (u_{odd}). Man erhält so ein „Schachbrettmuster“ lokaler Zeitentwicklungsoperatoren τ , das den ursprünglichen globalen Zeitentwicklungsoperator $U(t, 0)$ umso besser approximiert, je kleiner $\Delta t = t/M$ ist, je feiner also die Zerlegung ist (siehe Abbildung 2.1).

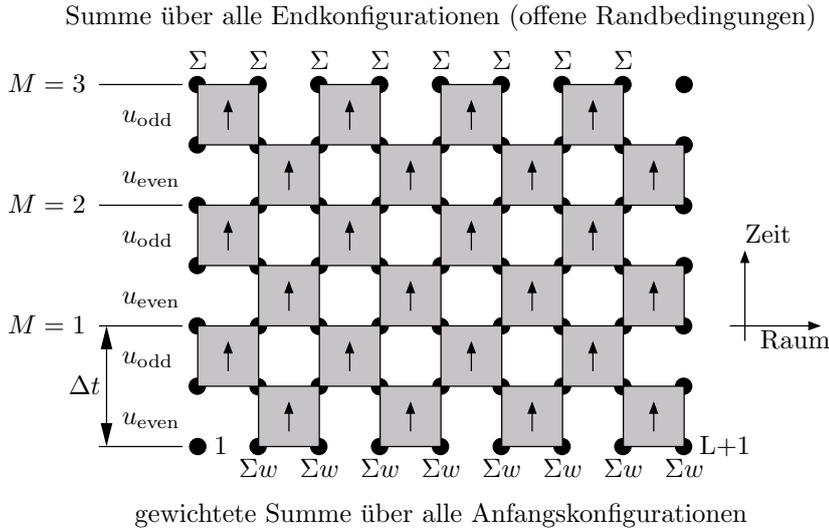


Abbildung 2.1: TROTTER-SUZUKI-Zerlegung des globalen Zeitentwicklungsoperators $U(t, 0)$ in lokale Transfermatrizen τ , die zu Zeilen u_{even} und u_{odd} zusammengefaßt sind. Es sind $M = 3$ Zeitschritte gezeigt.

Da die lokalen Hamiltonoperatoren $h_{i,i+1}$ in $\mathcal{H}_{\text{even}}$ unabhängig voneinander sind (und ebenso die $h_{i,i+1}$ in \mathcal{H}_{odd}), zerfallen die Zeitentwicklungsoperatoren u_{even} und u_{odd} für das gerade bzw. ungerade Untergitter jeweils in ein Tensorprodukt lokaler Zeitentwicklungsoperatoren $\tau_{i,i+1} = \exp(-t \cdot h_{i,i+1}/M)$:

$$u_{\text{even}} = \tau_{2,3} \otimes \tau_{4,5} \otimes \cdots \otimes \tau_{L,L+1} \quad (\text{für gerade } L) \quad (2.16a)$$

$$u_{\text{odd}} = \tau_{1,2} \otimes \tau_{3,4} \otimes \cdots \otimes \tau_{L-1,L}. \quad (2.16b)$$

Die Vorgehensweise ist also wie folgt: bei Systemen in diskreter Zeit sind die lokalen Übergangswahrscheinlichkeiten τ mit der Definition des Systems gegeben. Bei Systemen in kontinuierlicher Zeit wählt man dagegen ein Δt , das viel kleiner als die typische Reaktionszeit des stochastischen Prozesses ist, so daß $\|\Delta t \cdot h_{i,i+1}\| \ll 1$ in einer Matrixnorm erfüllt ist. Der lokale Hamiltonoperator $h_{i,i+1}$ wirkt auf allen Gitterplätzen auf

gleiche Weise, ist also unabhängig von i . Wir wählen nun ein festes Paar $i, i + 1$ von Gitterplätzen, die wir als „linken“ (l) und „rechten“ (r) bezeichnen. Beide haben jeweils n mögliche Zustände. Der Hamiltonoperator gibt die Übergangsrate von einer Anfangskonfiguration $l_1 r_1$ der beiden Gitterplätze (also von einem n^2 -dimensionalen Zustandsraum) in eine Endkonfiguration $l_2 r_2$ an. Die Matrixelemente der lokalen Transfermatrix τ in dieser Basis sind

$$(\tau)_{l_1 r_1}^{l_2 r_2} = [\exp(-\Delta t \cdot h_{i, i+1})]_{l_1 r_1}^{l_2 r_2} = \begin{array}{|c|c|} \hline l_2 & r_2 \\ \hline \uparrow \Delta t & \\ \hline l_1 & r_1 \\ \hline \end{array}. \quad (2.17)$$

Die stochastischen Randbedingungen für die stochastische Matrix $h_{i, i+1}$ drücken sich dadurch aus, daß man sicher in *irgendeinem* Zustand gelangt, also die Summe über alle Endkonfigurationen Eins ist (siehe Gleichung (2.2)):

$$\sum_{l_2 r_2} (\tau)_{l_1 r_1}^{l_2 r_2} = 1 \quad \forall l_1, r_1, \quad \begin{array}{|c|c|} \hline \Sigma_{l_2} & \Sigma_{r_2} \\ \hline \uparrow & \\ \hline l_1 & r_1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline \text{---} & \text{---} \\ \hline \uparrow & \\ \hline l_1 & r_1 \\ \hline \end{array} \quad (\text{trivial}). \quad (2.18)$$

Diese τ 's setzt man zu einem Schachbrettmuster zusammen, legt geeignete Randbedingungen fest, bildet die Spur über alle inneren Indizes und erhält so den Zeitentwicklungsoperator bis zur Zeit $t = M \cdot \Delta t$.

Ist zusätzlich keine räumliche Richtung ausgezeichnet, d.h. ist das Modell invariant unter räumlichen Spiegelungen, so nennen wir τ und das Modell *paritätsinvariant*,

$$(\tau)_{l_1 r_1}^{l_2 r_2} = (\tau)_{r_1 l_1}^{r_2 l_2}. \quad (2.19)$$

2.3 Exkurs: Ein einfacher Reaktions-Diffusions-Prozeß

Zur Verdeutlichung wollen wir den lokalen Hamiltonoperator h und die lokale Zeitentwicklungsmatrix τ eines einfachen Reaktions-Diffusions-Prozesses untersuchen. Als Beispiel nehmen wir Monomere A , die auf einem eindimensionalen Gitter einerseits wandern können und andererseits koagulieren, wodurch sie Dimere werden und aus der Betrachtung herausfallen (siehe auch Abschnitt 4.1). Zur Beschreibung reichen $n = 2$ Zustände: auf jedem Gitterplatz kann entweder ein solches Monomer sein (A), oder keines (\emptyset). Die Dynamik ist gegeben durch die folgenden Ratengleichungen für zwei benachbarte Gitterplätze:



Da höchstens zwei benachbarte A 's miteinander reagieren, haben wir eine NN-Wechselwirkung und können z.B. eine Ortsbasis wählen: $\{|\emptyset\emptyset\rangle, |AA\rangle, |A\emptyset\rangle, |\emptyset A\rangle\}$. Dann hat der

stochastische Hamiltonoperator $h_s^z = \langle z|h|s \rangle$ (Zeilen- und Spaltenindizes) die Gestalt

$$h = \begin{pmatrix} 0 & -2\alpha & & \\ & 2\alpha & & \\ & & D & -D \\ & & -D & D \end{pmatrix}. \quad (2.22)$$

Das ist wie folgt zu lesen: der Eintrag in der obersten Zeile, $\langle \emptyset\emptyset | -h | AA \rangle = 2\alpha$, gibt die Rate $|AA\rangle \rightarrow |\emptyset\emptyset\rangle$ an. Die Diagonalelemente sind so gesetzt, daß die Summe innerhalb jeder Spalte verschwindet (Gleichung (2.4b)) und h somit eine stochastische Matrix ist.

Wir beobachten, daß h blockdiagonal ist: Zustände mit einer (un-)geraden Anzahl A 's haben nach der Zeitentwicklung wieder eine (un-)gerade Anzahl A 's. Durch die Reaktionen bleibt also die *Parität eines Zustandes* (gerade/ungerade), definiert als $(-1)^{\#A}$, erhalten. Ferner ist der rechte untere Block (mit ungerader Parität) symmetrisch, d.h. die Diffusion ist reversibel (zeitumkehrinvariant). Dagegen ist der linke obere Block (mit gerader Parität) asymmetrisch, d.h. die Paar-Annihilation ist irreversibel.

Wir können h vollständig diagonalisieren, also in Produkte linker und rechter Eigenvektoren und ihre zugehörigen Eigenwerte zerlegen. Da h nicht symmetrisch ist, sind linke und rechte Eigenvektoren i.a. *nicht* transponiert zueinander, $|i\rangle \neq \langle i|^T$:

$$h = \sum_i \lambda_i |i\rangle \langle i| = \lambda_{e_1} |e_1\rangle \langle e_1| + \lambda_{e_2} |e_2\rangle \langle e_2| + \lambda_{o_1} |o_1\rangle \langle o_1| + \lambda_{o_2} |o_2\rangle \langle o_2| \quad (2.23)$$

$$\begin{aligned} &= 0 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes (1 \ 1 \ 0 \ 0) + 2\alpha \cdot \begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \otimes (0 \ 1 \ 0 \ 0) \\ &+ 0 \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} (0 \ 0 \ 1 \ 1) + 2D \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} (0 \ 0 \ 1 \ -1). \end{aligned} \quad (2.24)$$

Mit Hilfe der Eigenvektorzerlegung kann man den Zeitentwicklungsoperator τ in diesem Spezialfall bilden, indem man die Eigenwerte exponenziert (i.a. muß man aber noch den nilpotenten Anteil von τ berücksichtigen):

$$\tau = e^{-\Delta t h} = \sum_i e^{-\Delta t \lambda_i} |i\rangle \langle i| \quad (2.25a)$$

$$= 1 \cdot |e_1\rangle \langle e_1| + e^{-2\alpha \Delta t} \cdot |e_2\rangle \langle e_2| + 1 \cdot |o_1\rangle \langle o_1| + e^{-2D \Delta t} \cdot |o_2\rangle \langle o_2| \quad (2.25b)$$

$$= \begin{pmatrix} 1 & 1 - e^{-2\alpha \Delta t} & & \\ 0 & e^{-2\alpha \Delta t} & & \\ & & \frac{1}{2}(1 + e^{-2D \Delta t}) & \frac{1}{2}(1 - e^{-2D \Delta t}) \\ & & \frac{1}{2}(1 - e^{-2D \Delta t}) & \frac{1}{2}(1 + e^{-2D \Delta t}) \end{pmatrix}. \quad (2.25c)$$

Für kurze Zeiten $\Delta t \rightarrow 0$ geht τ gegen die Einheitsmatrix (der Zustand des Gitters bleibt im wesentlichen unverändert),

$$\lim_{\Delta t \rightarrow 0} \tau = |e_1\rangle \langle e_1| + |e_2\rangle \langle e_2| + |o_1\rangle \langle o_1| + |o_2\rangle \langle o_2| = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}. \quad (2.26)$$

Für lange Zeiten $\Delta t \rightarrow \infty$ projiziert τ auf den stationären Eigenraum:

$$\lim_{\Delta t \rightarrow \infty} \tau = |e_1\rangle\langle e_1| + |o_1\rangle\langle o_1| = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ & 1/2 & 1/2 \\ & 1/2 & 1/2 \end{pmatrix}. \quad (2.27)$$

Der erste Summand bedeutet, daß für lange Zeiten alle Konfigurationen AA in $\emptyset\emptyset$ übergehen, also die gemeinsame Wahrscheinlichkeit von AA und $\emptyset\emptyset$ in den Zustand $\emptyset\emptyset$ projiziert wird. Der zweite Term (Diffusion) mittelt räumlich, so daß linker und rechter Gitterplatz mit gleicher Wahrscheinlichkeit besetzt sind. Dagegen wird der Eigenraum o_2 , in dem die Differenz der Besetzungswahrscheinlichkeiten von linkem und rechtem Gitterplatz liegt, exponentiell mit der Diffusionsrate gedämpft.

Damit ergibt sich aus der Wahrscheinlichkeitserhaltung folgende Regel: nach der Wahrscheinlichkeitserhaltung ist $\langle 1|$, angewandt auf einen Zustand, 1. Also müssen alle Eigenräume i , in denen $\langle 1|$ eine Komponente hat ($\langle 1|i\rangle \neq 0$, hier für $|e_1\rangle$ und $|o_1\rangle$), erhalten bleiben, d.h. sie haben Eigenwerte $\lambda_i = 0$, und $|i\rangle$ ist ein Basisvektor des stationären Eigenraumes.

Alle anderen Eigenräume von h , für die also $\langle 1|i\rangle = 0$ gilt (d.h. sie treten in der Wahrscheinlichkeitsbilanz nicht auf), können unter der Wirkung von $-h$ kontrahiert werden, also positive Eigenwerte $\lambda_i > 0$ haben. Die positiven Eigenwerte kodieren die Übergangsraten der Reaktion. In diesen nichtstationären Eigenräumen spielt sich die ganze Dynamik des Systems ab, da der Hamiltonoperator für den stationären Eigenraum identisch verschwindet.

2.4 Zeitentwicklungsmatrix

Angenommen, wir wollen auf einem Gitter fester Länge L das Langzeitverhalten untersuchen, $t \rightarrow \infty$. Vorgegeben sei der Anfangszustand $|P(0)\rangle$ (ein n^L -dimensionaler Vektor) sowie die Randbedingung am linken und rechten Rand des Gitters. Dann können wir solange Zeitentwicklungsoperatoren u_{even} und u_{odd} hinzufügen, bis sich der Zustand $|P(t)\rangle$ nicht mehr ändert, wir haben $|P(t \rightarrow \infty)\rangle = |\text{stat}\rangle$ erreicht.

Einfacher geht es jedoch, wenn wir anstelle der Zustandsfolge die Zeitentwicklungsmatrix (siehe Gleichung (2.14)) $u = u_{\text{odd}} \times u_{\text{even}}$ betrachten und den Limes $\lim_{M \rightarrow \infty} U(M \cdot \Delta t, 0) = \lim_{M \rightarrow \infty} u^M$ bilden. u habe die Eigenwerte μ_i und zugehörige verallgemeinerte Eigenvektoren $|i\rangle$ und $\langle i|$, dann ist $u^M = \sum_i \mu_i^M |i\rangle\langle i|$. Aus dem Satz von Frobenius (siehe unten) folgt, daß U einen maximalen Eigenwert μ_0 hat, der wegen der Wahrscheinlichkeitserhaltung 1 ist. Also wird U für $M \rightarrow \infty$ von den Eigenwerten mit Betrag 1 dominiert:

$$\lim_{M \rightarrow \infty} u^M = \sum_{\{i:|\mu_i|=1\}} |i\rangle\langle i| = \text{Projektor auf stationären Eigenraum von } u. \quad (2.28)$$

Indem man also den Eigenraum von u zum Eigenwert 1 findet, kann man den Langzeitlimes *exakt* bilden.

Da U Übergangswahrscheinlichkeiten von Anfangs- in Endzustände enthält, sind alle Matrixelemente von U nichtnegativ (hier sogar zwischen 0 und 1). Ist U außerdem irreduzibel, dann besagt das Theorem von Frobenius, daß U einen maximalen positiven Eigenwert μ_0 der algebraischen Vielfachheit 1 und dazugehörige rechte und linke Eigenvektoren $|\mu_0\rangle$ und $\langle\mu_0|$ mit strikt positiven Einträgen hat. Maximal bedeutet hier, daß $|\mu_i| \leq \mu_0 \forall i > 0$ gilt.

2.5 Transfermatrix

Im Gegensatz zum Langzeitlimes eines endlichen Gitters (Abschnitt 2.4) vertauschen wir nun die Rollen von Raum und Zeit: wir interessieren uns hier für das Verhalten eines räumlich unendlich ausgedehnten Systems (thermodynamischer Limes) in endlicher Zeit.

Der Zeitentwicklungsoperator U für asymptotisch lange Zeiten ist einfach die Projektion auf den stationären Eigenraum. In diesem Limes wird das System komplett durch seinen stationären Zustand $|\text{stat}\rangle$ beschrieben. Analog führen wir die Transfermatrix T ein, die anstelle der zeitlichen die *räumliche* „Evolution“ des Systems beschreibt. Ein Zustand setzt sich nun nicht aus den Wahrscheinlichkeiten der räumlichen Konfigurationen zu einem bestimmten Zeitpunkt, sondern den Wahrscheinlichkeiten der *zeitlichen* Konfigurationen (möglichen Zeitentwicklungen) *eines* Gitterplatzes zusammen. Die Transfermatrix wirkt auf dem Raum der Zeitentwicklungen eines Gitterplatzes und liefert die möglichen Zeitentwicklungen des benachbarten Gitterplatzes. Analog zum Langzeitlimes bilden wir nun den thermodynamischen Limes, indem wir T wiederholt anwenden:

$$\lim_{N \rightarrow \infty} T^N = \text{Projektion auf stationären Eigenraum}, \quad (2.29)$$

wobei „stationär“ hier nicht die stationäre räumliche Konfiguration, sondern die stationäre Zeitentwicklung meint. Für stochastische Systeme werden wir gleich zeigen, daß der stationäre Eigenraum eindimensional ist, also

$$\lim_{N \rightarrow \infty} T^N = |\psi_R\rangle\langle\psi_L|, \quad (2.30)$$

$|\psi_R\rangle$ und $\langle\psi_L|$ sind rechter und linker Eigenvektor zum größten Eigenwert 1 (i.a. $\langle\psi_L| \neq |\psi_R\rangle^T$, da T asymmetrisch). Im Limes wird das System also wieder durch einen einzigen Zustand vollständig beschrieben.

Im Falle der TROTTER-SUZUKI-Zerlegung hat die Transfermatrix konkret folgende Gestalt: Der Zustand des Gitterplatzes i ist gegeben durch die Wahrscheinlichkeiten sämtlicher Zeitentwicklungen des Gitterplatzes zu den diskreten Zeiten $k/2 \cdot \Delta t$, $k = 1, \dots, 2M -$

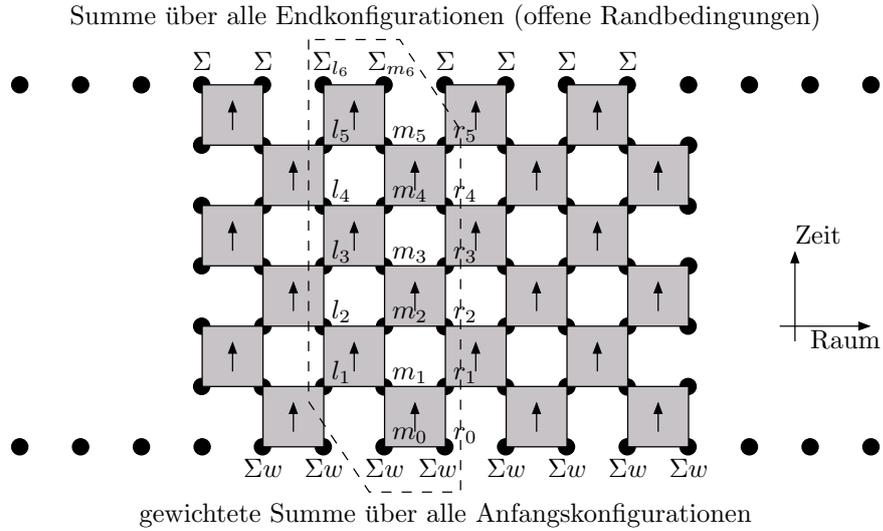


Abbildung 2.2: TROTTER-SUZUKI-Zerlegung der vollen Zeitentwicklung $U(t, 0)$ in lokale Transfermatrizen τ . Das System ist räumlich unendlich ausgedehnt, während es in zeitlicher Richtung auf $M = 3$ Zeitschritte begrenzt ist. Zwei benachbarte Spalten (umrandet) bilden die Transfermatrix $(T)_{r_1 \dots r_{2M-1}}^{l_1 \dots l_{2M-1}}$.

1. Wendet man die Transfermatrix auf den Zustand von i an, so erhält man den Zustand von $i + 2$, d.h. den Vektor der Wahrscheinlichkeiten, mit denen die verschiedenen Zeitentwicklungen von $i + 2$ auftreten. Zusätzlich müssen wir für die Zeiten 0 (Anfang) und t (Ende) die Randbedingungen festlegen, so daß die Transfermatrix am Anfang und Ende keine freien Indizes mehr hat. Bei stochastischen Systemen hat man offene Randbedingungen am Ende, es wird über alle möglichen Endkonfigurationen summiert (\sum_{σ_i} oder kurz Σ). Am Anfang seien alle Gitterplätze unkorreliert (Gleichung (2.10)), d.h. jeder Gitterplatz hat unabhängig von seinen Nachbarn die Besetzungswahrscheinlichkeit $w(\sigma)$ für die Zustände σ (im Diagramm $\sum_{\sigma_i} w(\sigma_i)$ oder kurz Σw). Mit der lokalen Transfermatrix τ aus Gleichung (2.17) ist die volle Transfermatrix (siehe Abbildung 2.2) definiert als

$$(T)_{r_1 \dots r_5}^{l_1 \dots l_5} = \sum_{m_0 r_0} w(m_0) w(r_0) \sum_{l_6 m_6} \sum_{m_1 \dots m_5} (\tau)_{l_5 m_5}^{l_6 m_6} (\tau)_{m_4 r_4}^{m_5 r_5} \dots (\tau)_{l_1 m_1}^{l_2 m_2} (\tau)_{m_0 r_0}^{m_1 r_1}. \quad (2.31)$$

Wir werden sehen, daß sich über die Struktur stochastischer Transfermatrizen einiges sagen läßt. Eine ausführliche Darstellung von ein- und zweidimensionalen Gittermodellen und verschiedenen Varianten der Transfermatrix findet der geneigte Leser z.B. in [Bax82].

2.5.1 Blockstruktur

Gibt es in der lokalen Transfermatrix τ erhaltene Größen (wie z.B. die Parität $(-1)^{\#A}$ im Exkurs Abschnitt 2.3), so zerfällt τ in Blöcke bzw. Unterräume, auf denen es unabhängig operiert.

Die mögliche Blockstruktur von τ legt die Frage nahe, ob auch die Zeitentwicklungsmatrix u oder die Transfermatrix T in Blöcke zerfallen, die man wegen der Linearität des Problems fortan getrennt behandeln kann, was Rechenzeit und Speicherbedarf stark reduzieren würde.

Im vorangegangenen Exkurs läßt sich die Zerlegung in gerade und ungerade Blöcke so formulieren (in der Notation von Gleichung (2.17)):

$$l_1 + r_1 \equiv l_2 + r_2 \pmod{2} \iff (-1)^{l_1 + r_1 + l_2 + r_2} = 1 \iff l_1 + l_2 \equiv r_1 + r_2 \pmod{2}. \quad (2.32)$$

Wir sehen also, daß in diesem Fall sowohl in zeitlicher wie in räumlicher Richtung die Parität, $(-1)^{\#A}$, erhalten bleibt.

Ob die gesamte Transfermatrix paritätserhaltend ist, hängt von den Randbedingungen ab: schließt man die Transfermatrix zeitlich periodisch (bzw. u räumliche periodisch), so ist der Beitrag der Randgitterplätze immer gerade: entweder auf beiden oder auf keinem der beiden identifizierten Randplätze sitzt ein Teilchen A . Damit erhält die Transfermatrix mit periodischen Randbedingungen (wie später bei Quantensystemen) eine (un-)gerade Teilchenzahl und zerfällt somit in Blöcke unterschiedlicher Parität. Bei offenen zeitlichen Randbedingungen hingegen, wie sie in stochastischen Systemen auftreten, bewirkt die Summe über alle Zustände der Anfangs- und Endgitterplätze, daß immer sowohl Zustände gerader als auch ungerader Teilchenzahl zusammen auftreten, T zerfällt also nicht in gerade/ungerade Blöcke. Physikalisch liegt dies daran, daß bei offenen Rändern A 's hinzukommen oder verschwinden können, was bei periodischen Randbedingungen nicht möglich ist.

2.5.2 Explizite Struktur der Transfermatrix

Der entscheidende Unterschied zu Quantensystemen mit periodischer Zeitachse ist, daß die Transfermatrix bei stochastischen Systemen in endlicher Zeit eine einfache Struktur hat, die man explizit angeben kann. Wie wir im folgenden zeigen werden, hat T einen Eigenwert 1, alle anderen Eigenwerte verschwinden. Es gibt also selbst im thermodynamischen Limes keine unendlich weiten Korrelationen, schon für *endliche* Systeme projiziert die Transfermatrix nur auf ihren stationären Zustand.

Betrachten wir zunächst den Rang von Potenzen der Transfermatrix, $\text{rk}(T^k)$. Durch die

stochastischen Randbedingungen wird die letzte lokale Transfermatrix τ von T (Abbildung 2.4a) trivial: ihr Wert ist 1 gemäß Gleichung (2.18), unabhängig von ihrer Anfangskonfiguration (l_3, m_3) . Damit bleiben nur noch $2 = 2M - 2$ Indizes (m_1, m_2) übrig, die den linken (l_i) und den rechten (r_i) Gitterplatz nichttrivial verbinden, also $\text{rk}(T) \leq n^{2M-2}$.

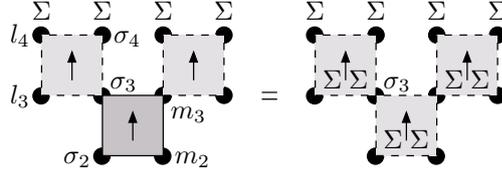


Abbildung 2.3: Offene Randbedingungen machen τ trivial (Gleichung (2.18)) und bewirken, daß auch über die Endindizes des vorangehenden τ *ungewichtet* summiert wird. So pflanzen sich die offenen Randbedingungen rückwärts in der Zeit fort.

Bei der nächsten Potenz T^2 (Abbildung 2.4b) ist die letzte Reihe von τ 's trivial (siehe Abbildung 2.3):

$$\sum_{l_4 \sigma_4 \sigma_3 \dots} \dots \underbrace{(\tau)_{l_3 \sigma_3}^{l_4 \sigma_4}}_1 \cdot (\tau)_{\sigma_2 m_2}^{\sigma_3 m_3} \dots = \sum_{\sigma_3 \dots} \dots (\tau)_{\sigma_2 m_2}^{\sigma_3 m_3} \dots \quad (2.33)$$

Weil das erste τ in Gleichung (2.33) trivial ist, wird über seinen Anfangsindex σ_3 *ungewichtet* summiert, also gelten für ein τ im vorletzten Zeitschritt ebenfalls offene (stochastische) Randbedingungen. Damit propagieren die Randbedingungen rückwärts in der Zeit und bilden einen Keil trivialer τ 's in dem Schachbrettmuster der Potenzen der Transfermatrix T , der nicht zu ihrem Wert beiträgt. Bei T^2 in Abbildung 2.4b bleibt als einziger verbindender Index m_1 übrig, also ist $\text{rk}(T^2) \leq n^{2M-3}$.

Schließlich gelangen wir zur $(2M - 1)$ -ten Potenz (Abbildung 2.4c): hier reicht der Keil trivialer τ 's gerade bis zum Anfang der Zeitentwicklung und trennt damit den linken und rechten Randgitterplatz von T^{2M-1} : sie sind statistisch unabhängig. Der Rang ist 1, die Transfermatrix zerfällt also in ein äußeres Produkt (Eigenvektorzzerlegung): $T^{2M-1} = |\psi_R\rangle \langle \psi_L|$, wobei sich $|\psi_R\rangle$ und $\langle \psi_L|$ aus den nichttrivialen τ 's ergeben, die links bzw. rechts vom Keil übrig bleiben. Daran ändert sich bei noch höheren Potenzen nichts mehr: es kommen nur triviale τ 's in der Mitte hinzu, die 1 sind gemäß Gleichung (2.18), also nichts am Wert der Transfermatrix ändern:

$$\text{rk}(T^k) = 1 \text{ und } T^k = |\psi_R\rangle \langle \psi_L| \quad \forall k \geq 2M - 1. \quad (2.34)$$

Daraus lesen wir sofort ab, daß T einen Eigenwert $\lambda_0 = 1$ hat und alle anderen Eigenwerte λ_i verschwinden. Aus $\text{Tr}(T^{2M-1}) = \sum_i \lambda_i^{2M-1} = 1 = \langle \psi_L | \psi_R \rangle$ folgt außerdem die Normierungsbedingung der Eigenvektoren.

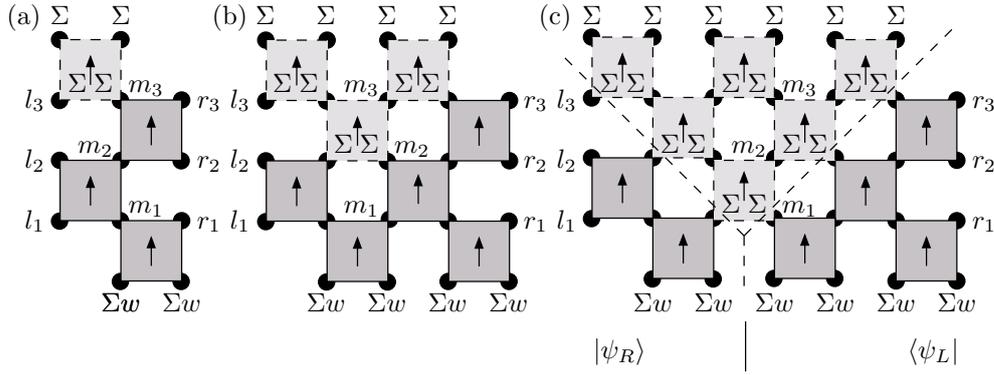


Abbildung 2.4: Potenzen der Transfermatrix T^k zerfallen in einen linken und rechten Teil abnehmenden Ranges, bis schließlich T^{2M-1} in zwei „Lichtkegel“ zerfällt, die während der M Zeitschritte nicht wechselwirken können (hier $2M = 4$):
 (a) $T = (\psi_R)^{l_1 l_2 l_3} \cdot (\psi_L)^{m_1 m_2}$ mit zwei verbindenden Indizes (m_1, m_2),
 (b) $T^2 = (\psi_R)^{l_1 l_2 l_3} \cdot (\psi_L)^{m_1}$ mit einem verbindenden Index (m_1), (c)
 $T^{2M-1} = (\psi_R)^{l_1 l_2 l_3} \cdot (\psi_L)^{r_1 r_2 r_3} = |\psi_R\rangle\langle\psi_L|$ ohne verbindende Indizes. Die weggelassenen Indizes verbinden ψ_L und ψ_R trivial, haben also keinen Einfluß auf den Wert von T^k .

All dies läßt sich auch explizit am Diagramm nachvollziehen: für $\langle\psi_L|\psi_R\rangle = 1$ setzen wir die beiden kleinen Dreiecke von nichttrivialen τ 's, die in Abbildung 2.4c unterhalb des trivialen Keils übriggeblieben sind, zusammen zu einem großen Dreieck; von dessen Spitze her pflanzen sich die stochastischen Randbedingungen bis an die Basis fort und machen dabei sämtliche τ 's trivial (siehe Abbildung 2.5).

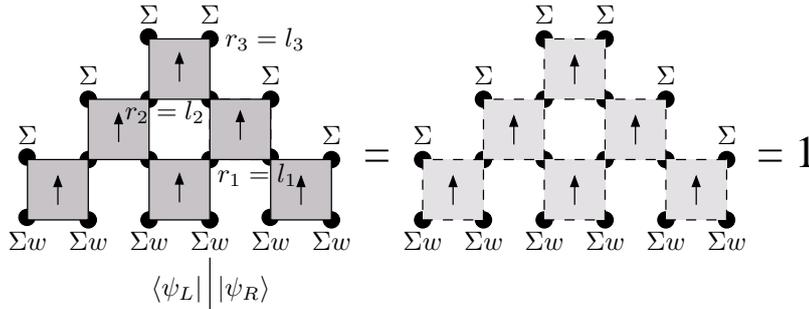


Abbildung 2.5: Die Normierungsbedingung der stationären rechten und linken Eigenvektoren $\langle\psi_L|\psi_R\rangle = 1$.

Ebenso sieht man, daß $|\psi_R\rangle$ und $\langle\psi_L|$ tatsächlich Eigenvektoren von T (und nicht nur von T^{2M-1}) sind (siehe Abbildung 2.6).

Für paritätsinvariante τ kann man sich $\text{Tr}(T) = 1$ auch auf andere Weise veranschau-

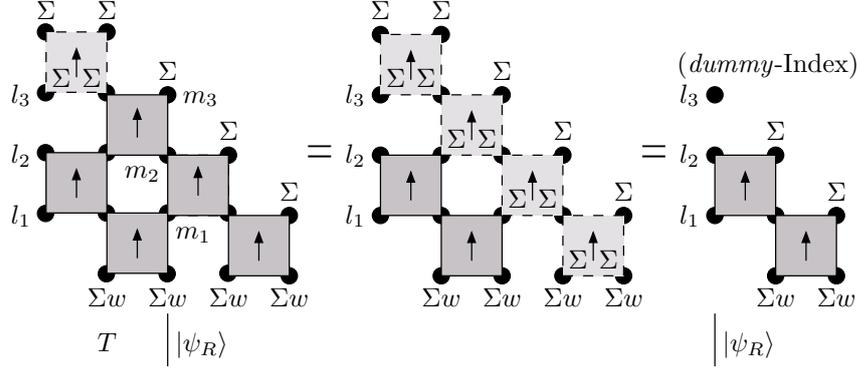


Abbildung 2.6: $|\psi_R\rangle$ ist tatsächlich der stationäre Eigenvektor der Transfermatrix T : es ist $T_{m_1 m_2 m_3}^{l_1 l_2 l_3} |\psi_R\rangle^{m_1 m_2 m_3} = |\psi_R\rangle^{l_1 l_2 l_3}$ (hier für $2M = 4$).

lichen: die Spur „rollt“ die Transfermatrix auf zu einem Zylinder, indem sie linke und rechte Randindizes identifiziert, während die Indizes in der Mitte auf der gegenüberliegenden Seite des Zylinders bleiben (siehe Abbildung 2.7). Das ist aber äquivalent zu einer rein lokalen Wechselwirkung zwischen dem linken bzw. rechten Gitterplatz auf der einen und dem mittleren auf der anderen Seite, für $2M$ Zeitschritte Δt , was exakt bekannt ist (z.B. für $2M = 6$):

$$\begin{aligned}
 \text{Tr}(T) &= \sum_{l_1 \dots l_5} (T)_{l_1 \dots l_5}^{l_1 \dots l_5} \\
 &= \sum_{m_0 r_0} w(m_0) w(r_0) \sum_{l_6 m_6} \sum_{l_1 \dots l_5} \sum_{m_1 \dots m_5} (\tau)_{l_5 m_5}^{l_6 m_6} (\tau)_{m_4 l_4}^{m_5 l_5} \dots (\tau)_{m_0 l_0}^{m_1 l_1} \\
 &= \sum_{m_0 r_0} w(m_0) w(r_0) \sum_{l_6 m_6} [\tau^{2M}(\Delta t)]_{r_0 m_0}^{l_6 m_6} \quad (\text{Paritätsinvarianz, Gl. (2.19)}) \\
 &= \sum_{m_0 r_0} w(m_0) w(r_0) \sum_{l_6 m_6} [\exp(-2M \Delta t \cdot h_{i, i+1})]_{r_0 m_0}^{l_6 m_6} \\
 &= \sum_{m_0 r_0} w(m_0) w(r_0) \sum_{l_6 m_6} [\tau(2M \Delta t)]_{r_0 m_0}^{l_6 m_6} \\
 &= \sum_{m_0 r_0} w(m_0) w(r_0) \cdot 1 = 1 \quad (\text{mit Gleichungen (2.18) und (2.11)}).
 \end{aligned}$$

Die statistische Unabhängigkeit von linkem und rechtem Randgitterplatz in T^{2M-1} mag zunächst erstaunen, hat aber ihre Ursache darin, daß wir mit der TROTTER-SUZUKI-Zerlegung eine endliche Wirkungsgeschwindigkeit („Lichtgeschwindigkeit“) eingeführt haben: pro Zeitintervall Δt kann sich die Wirkung nur um maximal zwei Gitterplätze ausbreiten. Man kann also $|\psi_R\rangle$ und $\langle\psi_L|$ als „Vergangenheitslichtkegel“ des linken bzw. rechten Randes auffassen. Für T^{2M-1} liegen $4M - 2$ Gitterplätze zwischen lin-

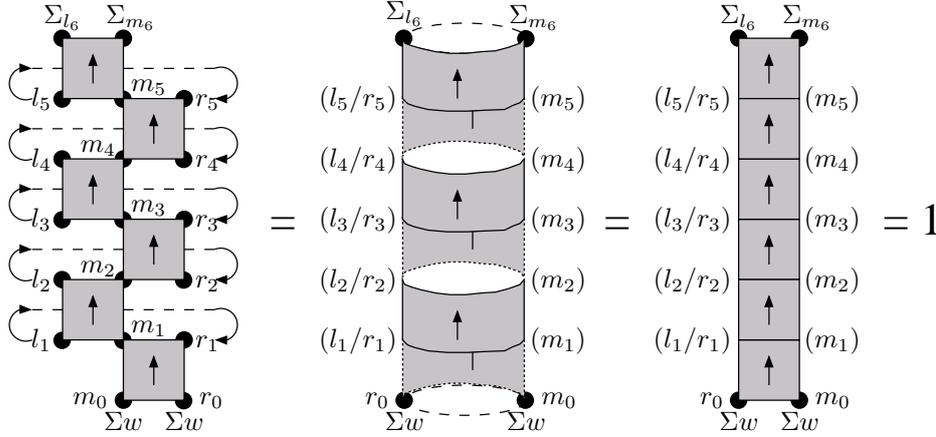


Abbildung 2.7: $\text{Tr}(T)_{r_1 \dots r_{2M-1}}^{l_1 \dots l_{2M-1}}$ ist äquivalent dazu, daß man T zu einem Zylinder mit rein lokaler Wechselwirkung aufrollt.

kem und rechtem Rand, ihre Vergangenheitslichtkegel überschneiden sich also gerade nicht während des Zeitintervalls $t = M \cdot \Delta t$.

Nun bleibt die Frage, woher der Rang größer Eins der niedrigen Potenzen von T kommt, wo doch alle Eigenwerte bis auf einen verschwinden. Hat τ eine NN-Wechselwirkung, dann ist τ als lokale Transfermatrix vom linken zum rechten Gitterplatz von maximalem Rang $\text{rk}(\tau)$, und der Rang von T^k ist durch die Anzahl verbindender Indizes m_i , $2M - 1 - k$, gegeben:

$$\text{rk}(T^k) = \begin{cases} n^{(2M-1)-k}, & k \leq 2M - 1 \\ 1, & k \geq 2M - 1 \end{cases} \quad (2.35)$$

Die Antwort ist, daß — anders als symmetrische Matrizen — asymmetrische Matrizen eine nichttriviale Jordanform haben können: jede Matrix läßt sich durch eine Ähnlichkeitstransformation in eine Matrix überführen, die auf der Diagonalen wiederum die Eigenwerte hat und auf einer der beiden Nebendiagonalen Einsen, die die *algebraische* Vielfachheit eines Eigenwertes angeben. Sie tragen zum Rang, nicht aber zum Spektrum der Matrix bei. In unserem Fall hat die Transfermatrix T^k folgende Jordanblöcke: einmal $[1]$, das ist der dominante Eigenwert, und jeweils $n_{2M-1-k-i}$ Blöcke mit Eigenwert 0 in der Diagonalen und i Einsen in der Nebendiagonalen:

$$T^k \sim 1 \times [1] \oplus \bigoplus_i n_{2M-1-k-i} \times \underbrace{\begin{bmatrix} 0 & 1 & & \\ & 0 & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix}}_{i \text{ Einsen}} \quad (2.36)$$

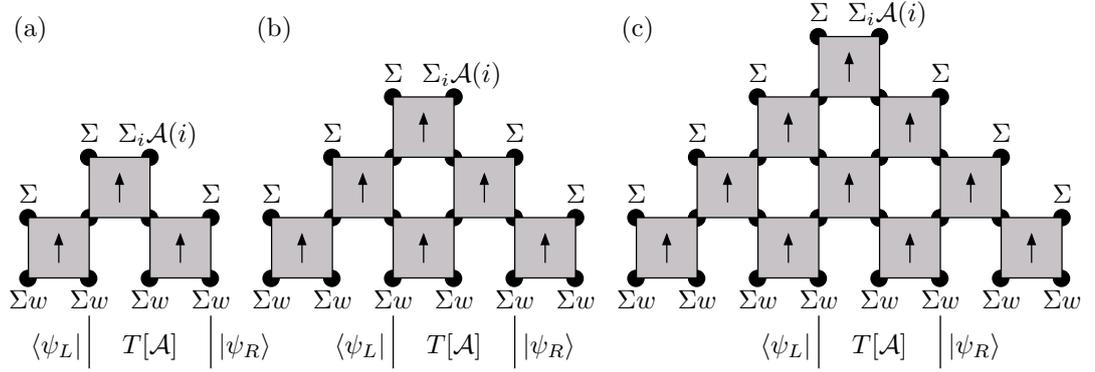


Abbildung 2.9: Der Erwartungswert eines lokalen Operators zum Zeitpunkt $t = M \cdot \Delta t$ wird bestimmt durch den Vergangenheitslichtkegel des Gitterplatzes, an dem der Operator gemessen wird. Der Operator verändert die Randbedingungen an dem Gitterplatz, an dem er eingefügt ist: $\sum_i \mathcal{A}(i)$ statt nur \sum_i . Der Vergangenheitslichtkegel ist für die Zeitschritte (a) $2M = 2$, (b) $2M = 3$ und (c) $2M = 4$ dargestellt.

Für die spätere Wahl der Dichtematrix ist die wichtige Beobachtung in Abbildung 2.9, daß bei Paritätsinvarianz (2.19) der linke Eigenvektor $\langle \psi_L |_{2M=3}$ im Trotterschritt $2M = 3$ und der rechte Eigenvektor $|\psi_R\rangle_{2M=4}$ im Trotterschritt $2M = 4$ identisch sind, wenn man zusätzlich einen *dummy*-Index (i_{2M-1}) hinzufügt, von dessen Wert $|\psi_R\rangle_{2M=4}$ nicht abhängt. Allgemein gilt

$$|\psi_R\rangle_{2M}^{i_1 \dots i_{2M-1}} = (\langle \psi_L |_{2M-1, i_1 \dots i_{(2M-1)-1}})^T =: |\psi_L\rangle_{2M-1}^{i_1 \dots i_{2M-2}}, \quad (2.41)$$

d.h. der rechte Eigenvektor zu einem gegebenen Zeitschritt ist identisch mit dem linken Eigenvektor im vorausgegangenen (halben) Zeitschritt.

Das Spektrum der Transfermatrix hat folgende Auswirkung auf die Korrelationslängen: die typische Korrelationslänge ξ für Zweipunkt-Korrelationsfunktionen ist definiert als die Länge, auf der die Korrelationsfunktion auf $1/e$ ihres Wertes abfällt,

$$e^{-2k\xi^{-1}} := \langle \rho(2k)\rho(0) \rangle \sim e^{-k \log \frac{\lambda_0}{\lambda_1}} = \left(\frac{\lambda_1}{\lambda_0} \right)^k = 0 \quad (k \geq 2M - 1), \quad (2.42)$$

wobei $\rho(2k)$ der Teilchenzahloperator am Gitterplatz $2k$ ist und $\langle \rho(2k)\rho(0) \rangle$ die Zweipunkt-Korrelationsfunktion. Weil der erste Eigenwert $\lambda_0 = 1$ ist und jeder andere Eigenwert $\lambda_i = 0$, folgt $\xi = 0$. Es gibt also in deutlichem Gegensatz zu den Quantensystemen mit periodischer Zeit keine exponentiell abklingenden und damit unendlich weitreichenden räumlichen Korrelationen; da der nilpotente Anteil schon bei endlichen Potenzen von T verschwindet, sind die Korrelationen strikt von endlicher Reichweite trotz des thermodynamischen Limes.

3 Renormierung

In Kapitel 2 haben wir gesehen, wie man mit der Transfermatrix den thermodynamischen Limes eines Systems nach endlicher Zeitentwicklung untersuchen kann. Wollen wir dieses Zeitintervall nun verlängern, so wächst der Zustandsraum (und damit die Dimension der Transfermatrix) exponentiell mit der Anzahl der Zeitschritte. Damit stoßen wir schnell an die Grenzen von Speicher und Rechenleistung jedes Computers.

Die Renormierung bietet einen Ausweg: es werden systematisch unwichtige Freiheitsgrade eliminiert; das reduziert die Dimension des Zustandsraumes. In der historischen Einführung (Abschnitt 3.1) schildern wir zunächst das Problem der vielen Freiheitsgrade am Beispiel wechselwirkender Quanten-Gittersysteme, um anschließend die Grundidee der Renormierungsgruppe darzustellen. Es folgt die Anwendung auf das KONDO-Problem (die WILSONSche Numerische Renormierungsgruppe) und eine Diskussion, warum sich diese Methode nur schlecht auf andere Quantensysteme übertragen läßt.

Diese Schwierigkeit hat WHITE mit der Dichtematrix-Renormierungsgruppe (DMRG) gelöst (Abschnitt 3.2): die DMRG ist eine numerische Methode, um den Grundzustand von stark gekoppelten Quanten-Gittersystemen mit hoher Genauigkeit zu berechnen. Diese ersten beiden Abschnitte sind als pädagogische Einführung gedacht, wir haben uns in der Darstellung eng an [NW99] gehalten.

Schließlich wenden wir die DMRG an, um den Zustandsraum nicht des Hamiltonoperators, sondern der *Transfermatrix* zu reduzieren (Abschnitt 3.3). Der Beweis, warum sich dieses für Quantensysteme erfolgreiche Verfahren wider Erwarten nicht gut auf stochastische Systeme übertragen läßt, ist ein zentrales Ergebnis dieser Diplomarbeit. Anschließend erläutern wir, wie man den Algorithmus ändern muß, um das richtige physikalische Verhalten optimal zu simulieren.

3.1 Historische Einführung

Wir beginnen mit einer Erläuterung der Grundlagen, auf denen die DMRG und später die Transfermatrix-DMRG aufbauen.

Angenommen, wir wollen ein stark gekoppeltes Quantenmodell auf einem Gitter untersuchen, z.B. das HEISENBERG-Modell

$$\mathcal{H} = J \sum_{\langle ij \rangle} \mathbf{S}_i \cdot \mathbf{S}_j, \quad (3.1)$$

wobei $\langle ij \rangle$ die Summe über nächste Nachbarn ist. Wir beschränken uns hier auf ein eindimensionales Gitter, um die DMRG möglichst klar zu veranschaulichen. Dann reduziert sich $\langle ij \rangle$ auf eine Summe über die Gitterplätze i und $j = i + 1$. Obwohl man den Hamiltonoperator in knapper Form schreiben kann, hat das HEISENBERG-Modell sehr viele Freiheitsgrade: für ein Spin- $\frac{1}{2}$ -System mit $n = 2$ Zuständen pro Gitterplatz hat das Modell 2^L Freiheitsgrade auf einem Gitter der Länge L .

Obwohl man das HEISENBERG-Modell mit Hilfe des BETHE-Ansatzes exakt lösen kann, ist es z.T. schwierig, daraus die physikalischen Eigenschaften zu extrahieren. Deshalb sind numerische Methoden — vor allem Quanten-Monte-Carlo-Rechnungen, exakte Diagonalisierung und die Numerische Renormierungsgruppe (NRG) — auch bei exakt lösbaeren Modellen von großem Nutzen. Gleichzeitig kann man die numerischen Verfahren an den exakten Lösungen prüfen.

3.1.1 Exakte Diagonalisierung

Der vielleicht einfachste numerische Ansatz ist, den Hamiltonoperator eines endlichen Gitters zu diagonalisieren. Der Hilbertraum des Gitters habe z.B. Zustände (Vektoren)

$$|m_1\rangle \otimes |m_2\rangle \otimes \cdots \otimes |m_L\rangle, \quad (3.2)$$

wobei $|m_i\rangle$ der Zustand des i -ten Gitterplatzes ist, also \uparrow oder \downarrow für Spin- $\frac{1}{2}$ -Systeme mit $n = 2$ Zuständen. Die Dimension des Hilbertraumes, n^L , wächst exponentiell mit der Gitterlänge L .

Welche Form hat nun der Hamiltonoperator in dieser Basis? Dazu betrachten wir einen lokalen Operator \mathcal{A} , der auf den Gitterplatz i wirkt:

$$\mathcal{A} = \text{id}_1 \otimes \cdots \otimes \text{id}_{i-1} \otimes \mathcal{A}_i \otimes \text{id}_{i+1} \otimes \cdots \otimes \text{id}_L \quad (3.3)$$

mit Matrixelementen

$$\langle m_L | \cdots \langle m_1 | \mathcal{A} | m_{1'} \rangle \cdots | m_{L'} \rangle = \delta_{m_1 m_{1'}} \cdots \delta_{m_{i-1}, m_{i-1'}} \mathcal{A}_{m_i m_{i'}} \delta_{m_{i+1}, m_{i+1'}} \cdots \delta_{m_L m_{L'}}. \quad (3.4)$$

Daraus kann man sich den Hamiltonoperator als Summe von Produkten lokaler Operatoren \mathbf{S}_i zusammensetzen. Seine Matrixdarstellung ist durch die vielen δ -Funktionen extrem dünn besetzt (*sparse*). Deshalb sind gewöhnliche Algorithmen zur Diagonalisierung (JACOBI, HOUSEHOLDER) nicht mehr effizient, stattdessen eignet sich z.B. der

LANCZOS-Algorithmus [Lan50] gut, um wenige Eigenvektoren einer dünn besetzten Matrix zu finden. Dabei speichert man \mathcal{H} typischerweise nicht vollständig, sondern erzeugt die wenigen von Null verschiedenen Elemente nach Bedarf.

Das Wachstum des Hilbertraumes und damit des Speicherbedarfs für den Eigenvektor limitieren jedoch die Gittergröße, für die man diesen Algorithmus praktisch einsetzen kann, auf etwa 36 Gitterplätze. Als Ausweg kann man versuchen, sich auf einen Unterraum des Hilbertraumes zu beschränken. Eine Vorschrift dafür liefert die Numerische Renormierungsgruppe (NRG, Abschnitt 3.1.3). Zuvor wollen wir jedoch kurz auf die Grundidee der Renormierung eingehen.

3.1.2 Grundidee der Renormierungsgruppe

Die Idee der Renormierungsgruppe (RG) ist es, aus einem physikalischen System mit vielen Freiheitsgraden schrittweise Freiheitsgrade zu eliminieren, die für die konkrete Fragestellung unwichtig sind. Man sagt, Freiheitsgrade werden *ausintegriert*, wenn man das System durch ein Pfadintegral über alle Freiheitsgrade $[\phi, \psi]$ beschreibt und die Integration über die „unwichtigen“ Freiheitsgrade $[\psi]$ durchführt. Übrig bleibt dann eine effektive Theorie in den verbleibenden Freiheitsgraden $[\phi]$:

$$Z = \int \mathcal{D}\phi \mathcal{D}\psi e^{iS[\phi, \psi]} = \int \mathcal{D}\phi e^{iS_{\text{eff}}[\phi]} \quad \text{mit} \quad S_{\text{eff}}[\phi] = -i \ln \int \mathcal{D}\psi e^{iS[\phi, \psi]}. \quad (3.5)$$

Dies ist offensichtlich eine irreversible Operation, die ausintegrierten Freiheitsgrade lassen sich aus der effektiven Theorie nicht mehr wiederherstellen. Deshalb ist die Renormierungsgruppe im mathematischen Sinne eine Semigruppe (d.h. ohne inverses Element (Umkehroperation)). Der entscheidende Vorteil ist, daß man auf diese Weise die Zahl der Freiheitsgrade (also die Dimension des Zustandsraumes) reduziert, aber dabei die restlichen Freiheitsgrade nicht einfach vernachlässigt, sondern durch das Ausintegrieren *vollständig* mit allen physikalischen Auswirkungen berücksichtigt. Oft hat die neue effektive Theorie formal eine ähnliche Gestalt wie die ursprüngliche, jedoch mit anderen Werten für die Kopplungskonstanten. Werden iterativ nur (infinitesimal) wenige Freiheitsgrade ausintegriert, so induziert die Renormierungsgruppe einen Fluß im Raum der Kopplungskonstanten (Raum der verschiedenen Theorien); in diesem Fall interessiert man sich für Fixpunkte des Renormierungsgruppen-Flusses.

Integriert man z.B. in einem System mit hoch- und niederfrequenten Freiheitsgraden (Biomolekül mit longitudinalen und transversalen Schwingungsmoden) die hochfrequenten Moden aus, so bekommt man eine effektive Theorie für die niederfrequenten Moden. Diese vereinfachte Theorie berücksichtigt die hochfrequenten Effekte und ihre Kopplung an die niederfrequenten Moden vollständig, aber zu dem Preis, daß man nun keine Aussagen mehr über das hochfrequente Verhalten machen kann.

3.1.3 Numerische Renormierungsgruppe

Um das Problem der vielen Freiheitsgrade zu lösen, beginnen wir zunächst mit einer Einführung in die grundlegenden Ideen der WILSONschen Methode der Numerischen Renormierungsgruppe (NRG) [Wil75]. Auf die algorithmischen Details, die denen der DMRG gleichen, werden wir später eingehen.

WILSON hat die Idee der Renormierungsgruppe angewandt, um das KONDO-Problem mit einfacher Verunreinigung [Wil75] *numerisch* zu untersuchen. Der Hamiltonoperator sei als Matrix in einer bestimmten Basis des Gitters gegeben. Dann fügt man Gitterplätze hinzu, wodurch sich die Basis vergrößert, und integriert mit der Renormierungsgruppen-Transformation unwichtige Basisvektoren aus, indem man den Hamiltonoperator numerisch auf eine reduzierte Basis projiziert.

Wir wollen diese Methode nun konkret für ein eindimensionales Quanten-Gittermodell entwickeln. WILSON hat das sphärisch symmetrische KONDO-Problem (eine einzige Verunreinigung in einem Metall) im Impulsraum auf ein halbumendliches eindimensionales Gitter abgebildet: die Verunreinigung befindet sich auf dem ersten Gitterplatz, und die sphärisch symmetrischen Impuls-Schalen entsprechen den folgenden Gitterplätzen mit nächster-Nachbar-Wechselwirkung. Dann verfährt man wie folgt:

1. Betrachte einen Ausschnitt des Systems mit L Gitterplätzen. L sollte klein genug sein, daß man den Hamiltonoperator \mathcal{H}_L noch exakt diagonalisieren kann.
2. Diagonalisiere \mathcal{H}_L numerisch für die m niedrigsten Eigenwerte (Energiewerte) und ihre zugehörigen Eigenvektoren (Wellenfunktionen) und speichere sie in den Spalten der unitären Matrix \mathcal{U}_L .
3. Transformiere \mathcal{H}_L und möglicherweise andere Operatoren \mathcal{A}_L , an deren Erwartungswerten wir interessiert sind, in diese neue m -dimensionale Basis, $\bar{\mathcal{H}}_L = \mathcal{U}_L^\dagger \mathcal{H}_L \mathcal{U}_L$, $\bar{\mathcal{A}}_L = \mathcal{U}_L^\dagger \mathcal{A}_L \mathcal{U}_L$ etc. Anschließend ist $\bar{\mathcal{H}}_L$ eine diagonale Matrix mit m Elementen.
4. Füge einen neuen Gitterplatz $L + 1$ an und bilde mit $\bar{\mathcal{H}}_L$ den neuen gemeinsamen Hamiltonoperator \mathcal{H}_{L+1} , indem man die Wechselwirkung zwischen dem neuen Gitterplatz und dem Block L in der neuen, reduzierten Basis rekonstruiert (s.u.).
5. Wiederhole ab Schritt 2 mit \mathcal{H}_{L+1} anstelle von \mathcal{H}_L .

Dieser Algorithmus ist als Diagramm in Abbildung 3.1 dargestellt. Ist der Hamiltonoperator in einer Ortsbasis (*real-space*) dargestellt, so faßt die Renormierungsgruppe mehrere Spins zu Blöcken zusammen. Typischerweise hält man die Anzahl m der erhaltenen Basisvektoren konstant, so daß Zeit- und Speicheraufwand für jeden Diagonalisierungsschritt konstant bleiben, und die Rechenzeit linear mit L zunimmt.

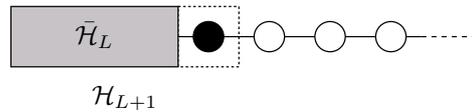


Abbildung 3.1: Diagramm zur Numerischen Renormierungsgruppe (NRG) von WILSON.

Es sei bemerkt, daß man in jedem Schritt die Eigenzustände *niedrigster* Energie für ein System der Größe L auswählt, unter der Annahme, daß diese Zustände die wichtigsten sind, um die Zustände niedrigster Energie für $L + 1$ Gitterplätze darzustellen. Inwieweit diese Annahme zutrifft, hängt entscheidend von den Randbedingungen des Blocks ab.

Am einfachsten ist es, die Wechselwirkung mit den benachbarten Gitterplätzen zu vernachlässigen (offene Randbedingungen). Für das KONDO-Problem mit einfacher Verunreinigung, das WILSON untersucht hat, ist das gerechtfertigt, da jeder darauffolgende Gitterplatz einer niedrigeren Energieskala entspricht, die exponentiell schwächer an die vorige gekoppelt ist. Das HEISENBERG-Modell hat dagegen keine intrinsische Trennung der Energieskalen; bei solchen Modellen wird die Numerische Renormierungsgruppe schon nach wenigen Iterationen recht ungenau. WHITE hat daraufhin für nichtwechselwirkende Modelle eine Methode entwickelt, in der alle Kombinationen offener und freier Randbedingungen (Ableitungen der Wellenfunktion verschwindet am Rand) verwendet werden, um eine optimale Basis zu konstruieren. Dies funktioniert für die untersuchten Systeme sehr gut, läßt sich aber schlecht auf wechselwirkende Vielteilchen-Quantensysteme erweitern.

Das Problem läßt sich anhand der Grundzustands-Wellenfunktion eines Teilchens in einem Kasten (mit offenen Randbedingungen) illustrieren: in einem Kasten der Länge L hat die Grundzustands-Wellenfunktion die Form $\sin(\frac{\pi}{L}x)$, $x \in [0, L]$ (siehe Abbildung 3.2). Im Renormierungsschritt möchte man daraus die Grundzustands-Wellenfunktion eines größeren Kastens linear kombinieren (hier für ein *nicht* wechselwirkendes System mit der Länge $2L$), die die Form $\sin(\frac{\pi}{2L}x)$, $x \in [0, 2L]$, hat. Diese hat aber ein Maximum bei $x = L$, während dort alle Wellenfunktionen des kleineren Kastens verschwinden, es gibt also keine gute Approximation.

Die Idee von WHITE war nun, die Randbedingungen (offen oder frei) nicht fest vorzuschreiben, sondern den Block der Länge L in einen größeren Block („Superblock“) einzubetten. Die Fluktuationen im Rest des Superblocks bewirken, daß der ursprüngliche Systemblock der Länge L „allgemeine“ Randbedingungen an seinen Rändern sieht, bzw. daß sich die Randbedingungen, die man natürlich auch im Superblock annehmen muß, im Systemblock nicht direkt bemerkbar machen: an den Rändern des Systemblocks gelten genau die Randbedingungen, die er als Teil eines größeren Systems hat. Im Limes eines unendlich großen Superblocks wird diese Konstruktion exakt. Aber schon für einen Superblock, der nur doppelt so groß ist wie der untersuchte Systemblock der Länge L ,

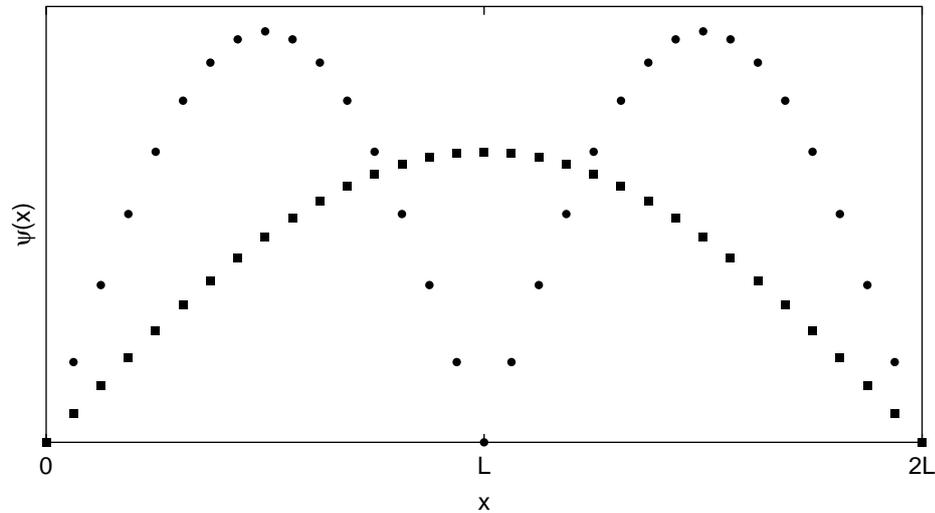


Abbildung 3.2: Grundzustands-Wellenfunktionen für ein Teilchen in einem Kasten der Länge L (●) bzw. $2L$ (■).

kann man sehr genaue Ergebnisse erhalten. Voraussetzung dafür ist eine gute Vorschrift, wie die Wellenfunktion des Superblocks auf die Wellenfunktion des Systemblocks projiziert wird, was nicht eindeutig ist. Die Whitesche Dichtematrix-Renormierungsgruppe (DMRG) liefert dafür sogar eine optimale Projektions-Vorschrift.

3.2 Dichtematrix-Renormierungsgruppe (DMRG)

Um das Problem der Randbedingungen und der optimalen Projektion zu lösen, hat S. WHITE die Numerische Renormierungsgruppe mit der Superblock-Methode [WN92] zur Dichtematrix-Renormierungsgruppe [Whi92, Whi93] kombiniert.

3.2.1 Dichtematrix-Projektion

Die Idee der Dichtematrix-Projektion ist, daß man den gesuchten Grundzustand der Spinkette nicht als Wellenfunktion $|\psi\rangle$, sondern dazu äquivalent als Dichtematrix eines reinen Zustandes, $\rho = |\psi\rangle\langle\psi|$, beschreibt. Interessiert man sich nur für einen Teil der Spinkette, so bildet man die Spur über die restlichen Spins und erhält so eine reduzierte Dichtematrix für den untersuchten Teil, die in der Regel einen gemischten Zustand beschreibt [Fey72].

In unserem Fall bildet die gesamte Spinkette den „Superblock“, der uns interessierende

Ausschnitt das „System“ und der Rest des Superblocks die „Umgebung“ (*environment*), siehe Abbildung 3.3.

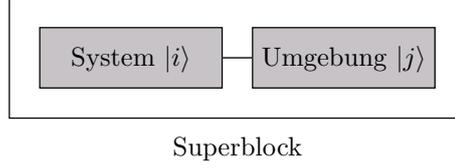


Abbildung 3.3: Der Superblock setzt sich aus System- und Umgebungsblock zusammen.

Sei $|i\rangle$ eine Basis des Systemblocks und $|j\rangle$ eine Basis der Umgebung. Dann kann man einen Zustand des Superblocks darstellen als

$$|\psi\rangle = \sum_{ij} \psi_{ij} |i\rangle \otimes |j\rangle. \quad (3.6)$$

Die reduzierte Dichtematrix des Systems erhält man als Spur über die Umgebung,

$$(\rho_{\text{sys}})_{ii'} = (\text{Tr}_{\text{env}} |\psi\rangle\langle\psi|)_{ii'} = \sum_j \psi_{ij} \psi_{i'j}^*. \quad (3.7)$$

Wegen der Normierung $\langle\psi|\psi\rangle = 1$ ist $\text{Tr}(\rho_{\text{sys}}) = 1$. Die reduzierte Dichtematrix enthält alle Eigenschaften des Systemblocks als Teil des Superblocks, z.B. ist der Erwartungswert eines Operators \mathcal{A} , der nur auf dem Systemblock wirkt,

$$\langle\mathcal{A}\rangle = \text{Tr}(\rho_{\text{sys}}\mathcal{A}) = \sum_{ii'} (\rho_{\text{sys}})_{ii'} \mathcal{A}_{i'i}. \quad (3.8)$$

ρ_{sys} ist per Konstruktion hermitesch, läßt sich also diagonalisieren mit reellen Eigenwerten $w_\alpha \geq 0$, $\sum_\alpha w_\alpha = 1$:

$$\rho_{\text{sys}} = \sum_\alpha w_\alpha |u^\alpha\rangle\langle u^\alpha|, \quad (3.9)$$

also gilt für den Systemblock-Operator \mathcal{A}

$$\langle\mathcal{A}\rangle = \sum_\alpha w_\alpha \langle u^\alpha | \mathcal{A} | u^\alpha \rangle. \quad (3.10)$$

Damit kann man beurteilen, wie wichtig ein einzelner Basisvektor $|u^\alpha\rangle$ des Systemblocks ist: je größer w_α ist, desto wichtiger sein Beitrag zum Erwartungswert von Operatoren. Wenn man einen Basisvektor mit $w_\alpha \approx 0$ wegläßt, macht man für kein \mathcal{A} einen nennenswerten Fehler bei der Berechnung von $\langle\mathcal{A}\rangle$.

Diese Argumentation wollen wir noch etwas präzisieren, auch um sie später besser verallgemeinern zu können. Wir werden zeigen, daß man die genaueste Darstellung des Zustandes des Superblocks erhält, wenn man die *wahrscheinlichsten* Eigenvektoren der reduzierten Dichtematrix als Basis verwendet.

Gesucht sind also m neue Basisvektoren für das System, $|u^\alpha\rangle = \sum_i u_i^\alpha |i\rangle$, $\alpha = 1, \dots, m$. Typischerweise ist $m < \ell_{\text{sys}}$ (die Dimension der Systembasis), also können wir den Superblock-Zustand $|\psi\rangle$ nicht exakt darstellen, sondern müssen ihn möglichst gut annähern:

$$|\psi\rangle \approx |\tilde{\psi}\rangle = \sum_{\alpha j} a_{\alpha,j} |u^\alpha\rangle \otimes |j\rangle. \quad (3.11)$$

Bedingung für eine optimale Darstellung sei, daß der Approximations-Fehler

$$S(|\tilde{\psi}\rangle) = \||\psi\rangle - |\tilde{\psi}\rangle\|^2 \quad (3.12)$$

minimal ist, wobei man bei gegebenem m über $a_{\alpha,j}$ und orthonormale $|u^\alpha\rangle$ variiert. Mit $\sum_j a_{\alpha,j} |j\rangle = \sum_j a_\alpha v_j^\alpha |j\rangle = a_\alpha |v^\alpha\rangle$, $|v^\alpha\rangle$ orthonormal, können wir $|\tilde{\psi}\rangle$ o.B.d.A. auch schreiben als

$$|\tilde{\psi}\rangle = \sum_\alpha a_\alpha |u^\alpha\rangle \otimes |v^\alpha\rangle = \sum_{\alpha ij} a_\alpha u_i^\alpha v_j^\alpha |i\rangle \otimes |j\rangle. \quad (3.13)$$

Damit ist in der $|i\rangle \otimes |j\rangle$ -Basis

$$S(|\tilde{\psi}\rangle) = \sum_{ij} \left(\psi_{ij} - \sum_{\alpha=1}^m a_\alpha u_i^\alpha v_j^\alpha \right)^2, \quad (3.14)$$

wobei wir nun ψ_{ij} als rechteckige Matrix auffassen und über a_α sowie orthonormale Basen u^α und v^α variieren. Dieses Minimierungsproblem ist ein bekanntes Problem der linearen Algebra, seine Lösung ist gegeben durch die Singulärwertzerlegung (*singular value decomposition*, SVD) der Matrix ψ_{ij} :

$$\psi = U D V^\dagger, \quad (3.15)$$

wobei U eine unitäre $\ell_{\text{sys}} \times \ell_{\text{sys}}$ -Matrix ist, V^\dagger eine unitäre $\ell_{\text{env}} \times \ell_{\text{env}}$ -Matrix (ℓ_{env} ist die Dimension des Umgebungsblocks) und D eine diagonale $\ell_{\text{sys}} \times \ell_{\text{env}}$ -Matrix. Die Diagonalelemente von D , $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_\ell \geq 0$, sind reell, man bezeichnet sie als die *Singulärwerte* von ψ . Sie sind eine Verallgemeinerung der Eigenwerte für nicht quadratische Matrizen. Da U und V^\dagger maximalen Rang haben, ist der Rang von ψ durch die Anzahl *positiver* Singulärwerte $\sigma_i > 0$ gegeben (alle anderen Matrixelemente von D sind Null). Weil $|\tilde{\psi}\rangle$ maximalen Rang m haben soll, sind die optimalen a_α durch die größten m Singulärwerte von ψ , und die u^α und v^α durch die dazugehörigen Spalten von U und V gegeben.

Das Minimierungsproblem kann man auch so formulieren: gesucht ist die Matrix $\tilde{\psi}$ von Rang $\text{rk}(\tilde{\psi}) \leq m$, die

$$S(\tilde{\psi}) = \text{Tr}_{\text{sys}} \left[(\psi - \tilde{\psi})(\psi - \tilde{\psi})^\dagger \right] \quad (3.16)$$

minimiert, wobei das Matrixprodukt die Umgebungsindizes und die Spur die Systemindizes kontrahiert. Nun setzen wir die SVD $\psi = UDV^\dagger$ ein und definieren \tilde{D} durch $\tilde{\psi} = U\tilde{D}V^\dagger$,

$$S(\tilde{D}) = \text{Tr}_{\text{env}} \left[U(D - \tilde{D})V^\dagger \cdot V(D - \tilde{D})^\dagger U^\dagger \right] \quad (3.17)$$

$$= \text{Tr}_{\text{env}} \left[U(D - \tilde{D})(D - \tilde{D})^\dagger U^\dagger \right] \quad (3.18)$$

$$= \text{Tr} \left[(D - \tilde{D})(D - \tilde{D})^\dagger \right]. \quad (3.19)$$

Gesucht ist also eine Matrix \tilde{D} von Rang $\text{rk}(\tilde{D}) \leq m$, die die Diagonalmatrix D der Singulärwerte optimal approximiert. Das ist offensichtlich erfüllt, wenn \tilde{D} ihrerseits eine Diagonalmatrix ist, die die m größten Singulärwerte von ψ enthält. Dann ist

$$|\tilde{\psi}_{\text{opt}}\rangle = \sum_{ij} (U\tilde{D}V^\dagger)_{ij} |i\rangle \otimes |j\rangle = \sum_{\alpha\beta ij} \tilde{D}_{\alpha\beta} (u_i^\alpha |i\rangle) \otimes (v_j^\alpha |j\rangle) = \sum_{\alpha=1}^m \sigma_\alpha |u^\alpha\rangle \otimes |v^\alpha\rangle \quad (3.20)$$

die beste Approximation an $|\psi\rangle$ mit Rang m .

Die reduzierte Dichtematrix des Systems ist

$$\rho_{\text{sys}} = \text{Tr}_{\text{env}} |\psi\rangle\langle\psi| = \psi\psi^\dagger = UDV^\dagger V D^\dagger U^\dagger = U D D^\dagger U^\dagger = U D^2 U^\dagger, \quad (3.21)$$

d.h. U diagonalisiert auch ρ_{sys} . Also sind die optimalen Basisvektoren des Systems, $|u^\alpha\rangle$, gleichzeitig die Eigenvektoren zu den größten Eigenwerten $w_\alpha = \sigma_\alpha^2 = a_\alpha^2$ der reduzierten Dichtematrix. Die w_α sind die Wahrscheinlichkeiten, daß sich der Systemblock im Zustand $|u^\alpha\rangle$ befindet, wenn der gesamte Superblock im Zustand $|\psi\rangle$ ist. Ist mehr als ein $w_\alpha > 0$, so befindet sich der Systemblock in einem gemischten Zustand.

Der Approximationsfehler Δ ist die Summe der Wahrscheinlichkeiten der vernachlässigten Zustände des Systemblocks,

$$\Delta = S(|\tilde{\psi}_{\text{opt}}\rangle) = \sum_{\alpha=m+1}^{\ell} w_\alpha = 1 - \sum_{\alpha=1}^m w_\alpha. \quad (3.22)$$

Offenbar ist die Approximation umso besser, je kleiner die trunkierten Eigenwerte w_α , $\alpha > m$, sind. Wir haben also gezeigt, daß man einen reinen Zustand des Superblocks am besten approximiert, indem man die m wichtigsten Eigenzustände der reduzierten

Dichtematrix des Systemblocks, die man aus einer Wellenfunktion des Superblocks gewonnen hat, verwendet. Das ist ein wichtiger Unterschied zur NRG, wo man als Basis die Zustände niedrigster Energie genommen hat.

Analog kann man auch eine reduzierte Basis für den Umgebungsblock konstruieren, indem man die reduzierte Dichtematrix durch eine Spur über den Systemblock bildet,

$$\rho_{\text{env}} = \text{Tr}_{\text{sys}} |\psi\rangle\langle\psi| = \psi^\dagger \psi = VD^\dagger U^\dagger U D V^\dagger = VD^2 V^\dagger, \quad (3.23)$$

und ihre wichtigsten Eigenvektoren $|v^\alpha\rangle$ als neue Umgebungsbasis verwendet. Dieselbe Singulärwertzerlegung von ψ liefert also gleichzeitig die optimalen Basen $|u^\alpha\rangle$ (U) für das System und $|v^\alpha\rangle$ (V) für die Umgebung. Es sei aber betont, daß die SVD nur zur theoretischen Begründung der Methode dient, numerisch gewinnt man die Eigenvektoren $|u^\alpha\rangle$ und $|v^\alpha\rangle$ durch Diagonalisierung der hermiteschen Matrizen ρ_{sys} und ρ_{env} .

3.2.2 DMRG-Algorithmus

Um die Dichtematrix-Renormierungsgruppe durchzuführen, müssen wir zunächst definieren, wie wir den Systemblock vergrößern wollen, wie der Superblock (und der Umgebungsblock) aussehen und welchen Zustand des Superblocks wir approximieren wollen.

Bei wechselwirkenden Systemen wächst die Dimension des Zustandsraumes exponentiell mit der Anzahl der Gitterplätze, andererseits ist die Approximation am besten, wenn m nicht viel kleiner als ℓ_{sys} , die Dimension des vergrößerten Systemblocks, ist. Daher empfiehlt es sich, in jedem Renormierungsschritt möglichst wenige Gitterplätze, also einen, hinzuzufügen.

Für die Wahl des Umgebungsblocks gibt es mehrere sinnvolle Möglichkeiten; wir beschreiben hier den sogenannten *infinite-system*-Algorithmus, der die beste Näherung an den thermodynamischen Limes (unendliches System) erlaubt. Dazu spiegelt man den Systemblock am rechten Rand (wo er wächst), siehe Abbildung 3.4, und erhält so den Umgebungsblock.

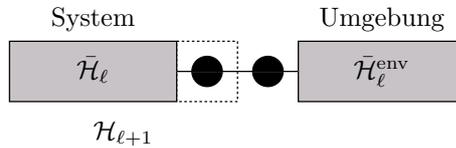


Abbildung 3.4: Der Superblock des *infinite-system*-Algorithmus, in der Mitte zwei neu hinzugefügte Gitterplätze.

Die reduzierte Dichtematrix liefert, wie im vorigen Abschnitt beschrieben, eine Basis für denjenigen Superblock-Zustand, aus dem sie gebildet wurde. Diesen Zustand bezeichnet

man als Ziel- (*target*-) Zustand. Typischerweise interessiert man sich für den Grundzustand oder den niedrigsten angeregten Zustand.

Der *infinite-system*-Algorithmus geht nun wie folgt vor (für ein spiegelsymmetrisches System):

1. Bilde einen Superblock mit $L = 2\ell + 2$ Gitterplätzen, der klein genug ist, um ihn noch exakt diagonalisieren zu können.
2. Finde numerisch die Energie und die Wellenfunktion $|\psi\rangle$ des *target*-Zustandes des Superblock-Hamiltonoperators $\mathcal{H}_L^{\text{super}}$. Dazu verwendet man meist den LANCZOS- oder DAVIDSON-Algorithmus für dünn besetzte Matrizen.
3. Bilde aus $|\psi\rangle$ die reduzierte Dichtematrix ρ_{sys} des Systems, d.h. für die $\ell + 1$ linken Gitterplätze.
4. Diagonalisiere ρ_{sys} mit einer Routine für dichte, hermitesche Matrizen. Die m Eigenvektoren zu den größten Eigenwerten werden in der unitären Matrix \mathcal{U}_L zusammengefaßt.
5. Erweitere $\bar{\mathcal{H}}_\ell$ um einen Gitterplatz in der Mitte zu $\mathcal{H}_{\ell+1}$ (und analog andere Operatoren \mathcal{A} im Systemblock). Transformiere $\mathcal{H}_{\ell+1}$, $\mathcal{A}_{\ell+1}$ auf die neue Basis: $\bar{\mathcal{H}}_{\ell+1} = \mathcal{U}_L^\dagger \mathcal{H}_{\ell+1} \mathcal{U}_L$, $\bar{\mathcal{A}}_{\ell+1} = \mathcal{U}_L^\dagger \mathcal{A}_{\ell+1} \mathcal{U}_L$.
6. Bilde den neuen Superblock der Größe $L + 2 = 2\ell + 4$ mit $\bar{\mathcal{H}}_{\ell+1}$, zwei neuen Gitterplätzen in der Mitte und dem gespiegelten Hamiltonoperator für die Umgebung, $\bar{\mathcal{H}}_{\ell+1}^{\text{env}}$.
7. Wiederhole ab Schritt 2 mit $\ell + 1$ anstelle von ℓ .

Mit der vorgestellten DMRG-Methode und *finite-size*-Korrekturen lassen sich Grundzustands-Wellenfunktionen von bis zu mehreren tausend Gitterplätzen berechnen. Damit kann man kritische Exponenten z.T. genauer abschätzen als mit alternativen Methoden wie Quanten-Monte-Carlo (QMC) oder Reihenentwicklungen.

3.2.3 Stochastische DMRG

Bisher haben wir die Renormierung in Quantensystemen beschrieben, aber unser Ziel ist es, stochastische Systeme zu behandeln. Ein Ansatz, die Methode der Renormierung auf Reaktions-Diffusions-Prozesse zu übertragen, stammt von CARLON, HENKEL und SCHOLLWÖCK [CHS99]: sie haben die DMRG-Methode auf kritische Phänomene bei stochastischen Prozessen im Nichtgleichgewicht angewandt, am Beispiel des Paar-Annihilations-Prozesses (Abschnitt 4.1) und des *branching-fusing*-Prozesses (Abschnitt

4.2) im thermodynamischen Limes. Aus der Energielücke Γ (Gleichung (2.8)) bei einem Gitter fester Länge L kann man auf den dynamischen Exponenten $\theta = \nu_{\parallel}/\nu_{\perp}$ schließen, den man berechnet durch $\Gamma \sim L^{-\theta}$. Um die Energielücke zu bestimmen, braucht man auch die Energie des ersten angeregten Zustandes, d.h. bei den DMRG-Iterationen muß man nicht nur eine gute Basis für die Wellenfunktion des Grundzustandes, sondern auch des ersten angeregten Zustandes finden. Es hat sich gezeigt, daß dies am genauesten geht, wenn man die reduzierte Dichtematrix *nur* aus dem ersten angeregten Zustand konstruiert.

Die Autoren kamen zu folgenden Ergebnissen:

- Um von der endlichen Gitterlänge $L < 100$ auf den thermodynamischen Limes schließen zu können, ist es zunächst erforderlich, mit dem *finite-system*-Algorithmus [Whi93] (einer Variante des in Abschnitt 3.2.2 beschriebenen Algorithmus) sehr genau Ergebnisse für verschiedene Gittergrößen L zu berechnen, die sich dann mit *finite-size*-Korrekturen auf $L \rightarrow \infty$ extrapolieren lassen.
- Die Wahl der reduzierten Dichtematrix $\rho_{\text{sys}} = \text{Tr} |\psi\rangle\langle\psi|$ ist hier nicht eindeutig, da der Hamiltonoperator nicht symmetrisch (zeitumkehrinvariant) ist und unterschiedliche rechte und linke Eigenvektoren hat, $|\psi_R\rangle$ und $\langle\psi_L|$. Empirisch zeigt sich, daß die Konvergenz des Algorithmus mit der symmetrischen Kombination $\rho_{\text{sys}}^{\text{symm}} = \frac{1}{2} \text{Tr}(|\psi_R\rangle\langle\psi_R| + |\psi_L\rangle\langle\psi_L|)$ mit Abstand am besten ist (siehe auch Abschnitt 3.3.4 weiter unten).
- Die Eigenwerte der reduzierten Dichtematrix fallen so stark ab, daß nur die größten $m \leq 32$ Eigenwerte oberhalb der Fließkomma-Genauigkeit ϵ liegen (also numerisch nicht-singulär sind). Um die Genauigkeit zu verbessern, hilft es also nicht, bei der Dichtematrix-Projektion einen größeren Unterraum des Zustandsraumes zuzulassen ($m > 32$), sondern nur, die Fließkomma-Genauigkeit über die üblichen 53 Mantissenbits (*double precision*) hinaus zu erhöhen.

Obschon die DMRG für stochastische Prozesse noch nicht die Genauigkeit der Quanten-Monte-Carlo-Methode (QMC) erreicht, hat sie gegenüber dieser einige Vorteile: die Konvergenz verlangsamt sich nicht am kritischen Punkt (*critical slow-down*), die DMRG braucht keine Zufallszahlen, und man muß keine Annahme über den grundlegenden Zustand machen, um den herum man entwickelt. Seitdem ist sie auf eine Vielzahl von Reaktions-Diffusions-Prozessen angewandt worden [Hie98, KP98, CHS01, CDvL01, HS01].

3.3 Transfermatrix-Renormierungsgruppe (TMRG)

Während sie die DMRG gut eignet, das Verhalten von Quantensystemen bei $T = 0$ bzw. das Langzeitverhalten von stochastischen Systemen auf *endlichen* Gittern zu untersu-

chen, ist die Extrapolation auf den thermodynamischen Limes mit Hilfe der *finite-size*-Korrekturen (Abschnitt 3.2.3) mühsam. In Kapitel 2 haben wir jedoch gesehen, daß sich mit der Transfermatrix für stochastische Systeme der thermodynamische Limes exakt bilden läßt, allerdings nur für ein kurzes Zeitintervall, da der Zustandsraum exponentiell mit der Anzahl der Zeitschritte wächst. Analog läßt sich für Quantensysteme eine Quanten-Transfermatrix konstruieren, die ein System im thermodynamischen Limes, aber bei endlicher Temperatur $T > 0$ beschreibt (Abschnitt 3.3.1). Diese Quanten-Transfermatrix läßt sich sehr erfolgreich renormieren, was die Vermutung nahelegt, daß man auch die stochastische Transfermatrix gut renormieren kann. Wie dies geschehen kann, insbesondere mit welcher Form der Dichtematrix, diskutieren wir in den darauffolgenden Abschnitten.

3.3.1 Quanten-TMRG

Von NISHINO kam 1995 die Idee [Nis95], daß man die DMRG auch benutzen kann, um die Transfermatrix eines zweidimensionalen klassischen Gittermodells zu renormieren, da die thermodynamischen Eigenschaften allein durch den größten Eigenwert und den dazugehörigen Eigenzustand der Transfermatrix bestimmt sind.

BURSILL *et al.* [BXG96] haben daraufhin die Transfermatrix einer eindimensionalen Quanten-Spinkette renormiert, indem sie diese zuerst mit Hilfe der TROTTER-SUZUKI-Zerlegung auf ein zweidimensionales klassisches Modell abgebildet haben. Das hat es ihnen ermöglicht, die thermodynamischen Eigenschaften der Spinkette bis zu sehr niedrigen Temperaturen zu untersuchen. Dazu geht man wie folgt vor: die Zustandssumme der eindimensionalen Quanten-Spinkette mit Temperatur T und Spektralparameter $\beta = 1/k_B T$ ist

$$Z(\beta) = \text{Tr} (e^{-\beta\mathcal{H}}) = \sum_{\{\sigma\}} \langle \sigma | e^{-\beta\mathcal{H}} | \sigma \rangle, \quad (3.24)$$

wobei die Spur über alle Konfigurationen der Spinkette gebildet wird. Den Operator $\exp(-\beta\mathcal{H})$ kann man aber auch auffassen als „Zeitentwicklungsoperator“ (Lösung der Schrödingergleichung) über das imaginäre Zeitintervall $i\beta$. Damit kommt die imaginäre Zeit als zweite Dimension hinzu. Die Schrödingergleichung in imaginärer Zeit hat formal die gleiche Gestalt wie die Mastergleichung (2.3). Anders als bei der Zeitentwicklung in der Mastergleichung erzwingt die Spur jedoch *periodische* Randbedingungen in Zeitrichtung, indem sie Anfangs- und Endkonfigurationen des Gitters identifiziert. Es wird sich herausstellen, daß dies der entscheidende Unterschied ist in der Anwendbarkeit der TMRG auf stochastische und Quanten-Systeme.

Um den Erwartungswert eines lokalen Operators \mathcal{A} zu berechnen, setzt man diesen Operator an beliebiger zeitlicher Stelle (da die Zeit periodisch geschlossen ist) in die

und rechte Eigenvektoren $\langle\psi_L|$ und $|\psi_R\rangle$ hat. Damit stellt sich — wie auch bei der stochastischen DMRG (Abschnitt 3.2.3) — die Frage nach der korrekten reduzierten Dichtematrix. BURSILL *et al.* wählten die symmetrische Dichtematrix

$$\rho_{\text{sys}} = \text{Tr}_{\text{env}} |\psi_R\rangle\langle\psi_R|. \quad (3.28)$$

Später konnten WANG und XIANG [WX97] zeigen, daß die asymmetrische Wahl

$$\rho_{\text{sys}}^{\text{asym}} = \text{Tr}_{\text{env}} |\psi_R\rangle\langle\psi_L| \quad (3.29)$$

physikalisch sinnvoller ist und überdies zu besseren numerischen Resultaten führt. Diese Wahl der Dichtematrix hat sich inzwischen für die Quanten-TMRG durchgesetzt [Shi97, MS98, NS99].

3.3.2 Stochastische TMRG

Die DMRG dezimiert den Hilbertraum der räumlichen Gitter-Konfigurationen, auf dem der Hamiltonoperator wirkt. Für stochastische Systeme im thermodynamischen Limes betrachten wir stattdessen die Transfermatrix auf dem Raum möglicher Zeitentwicklungen einzelner Gitterplätze, der reduziert werden muß, um numerisch handhabbar zu sein: das ist die Idee der TMRG. Die Wahrscheinlichkeit einer Zeitentwicklung in einem Zeitintervall hängt nicht nur von dessen Vergangenheit, sondern auch von dessen Weiterentwicklung in der Zukunft ab. Da die unendliche Zukunft aber noch nicht bekannt ist, gibt man sich mit einem „*look-ahead*“ von mehreren Zeitschritten zufrieden („Umgebung“ oder *environment*), während die vergangene Zeitentwicklung, deren Zustandsraum reduziert wird, das „System“ bildet. Aufgrund der nichtperiodischen Zeitachse (zeitliche Randbedingungen) und der Tatsache, daß das System nicht invariant unter Zeitumkehr ist, sind System und Umgebung nicht spiegelsymmetrisch (anders als bei der DMRG und der Quanten-TMRG), siehe Abbildung 3.6.

Mit expliziten Eigenvektoren (siehe Abschnitt 2.5.2) ist es schwierig, zwischen System und Umgebung die neuen Gitterplätze (Zeitschritte) einzufügen wie bei der DMRG. Stattdessen kann man sie nur am Ende der Zeitentwicklung anfügen und die Grenze zwischen System und Umgebung mit nach hinten verschieben, so daß die Länge des *look-ahead* konstant bleibt (siehe Abschnitt 4.6). Das führt aber zu numerischen Problemen, deshalb sind die expliziten Eigenvektoren im wesentlichen für analytische Argumentationen von Nutzen.

Als numerisch günstigere Alternative kann man auch die Transfermatrix getrennt nach System und Umgebung speichern, so lassen sich an beide Teile problemlos vorne und hinten zusätzliche Zeitschritte anfügen, allerdings muß man dann mit jedem Schritt die dominanten Eigenvektoren neu bestimmen. Das geschieht z.B. mit dem *power*-Verfahren: auf einen zufällig gewählten Startvektor wird solange eine Matrix multipliziert, bis der

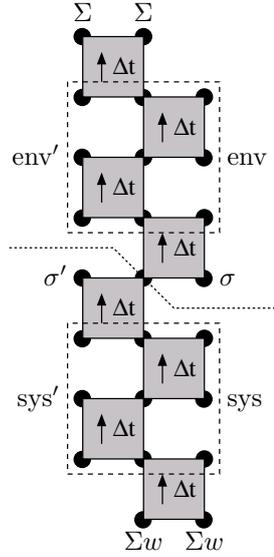


Abbildung 3.6: System- und Umgebungsblock der Transfermatrix in der TMRG (hier für $2M = 8$).

Vektor zum Eigenvektor des größten Eigenwertes der Matrix konvergiert ist. Diese Methode ist eindeutig, da es selbst nach der Renormierung noch einen einzigen Eigenwert 1 gibt, der durch eine endliche Lücke vom nächsten Eigenwert getrennt ist. Das LANZOS-Verfahren eignet sich hier nicht gut, weil die Transfermatrix nicht dünn besetzt ist.

3.3.3 Die asymmetrische Dichtematrix

Zunächst liegt es nahe zu versuchen, die gleiche Vorschrift für die reduzierte Dichtematrix zu benutzen, die sich in der Quanten-TMRG bewährt hat (Gleichung (3.29)):

$$\rho_{\text{sys}}^{\text{asym}} = \text{Tr}_{\text{env}} |\psi_R\rangle\langle\psi_L|. \quad (3.30)$$

Wir haben jedoch bereits in Abbildung 2.5 gesehen, wie sich die stochastischen Randbedingungen vom Ende zum Anfang hin fortpflanzen, so daß $\text{Tr}_{\text{sys}}(\rho_{\text{sys}}) = \text{Tr}_{\text{sys,env}} |\psi_R\rangle\langle\psi_L| = \langle\psi_L|\psi_R\rangle = 1$ ist. Wenn wir die partielle Spur nur über die Umgebung bilden, werden bereits alle τ 's mit Umgebungsindizes trivial, und es bleibt nur der dominante Zustand des *Systems* übrig (siehe Abbildung 3.7):

$$\rho_{\text{sys}}^{\text{asym}} = \text{Tr}_{\text{env}} \left(T_{[2M]}^{2M-1} \right) = T_{[M]}^{M-1} = |\psi_R\rangle_{\text{sys}} \langle\psi_L|_{\text{sys}}. \quad (3.31)$$

Das ist aber nichts anderes als der dominante Zustand des Gesamtsystems auf der Hälfte der Zeit, als System und Umgebung zusammen nur so groß waren wie das System jetzt.

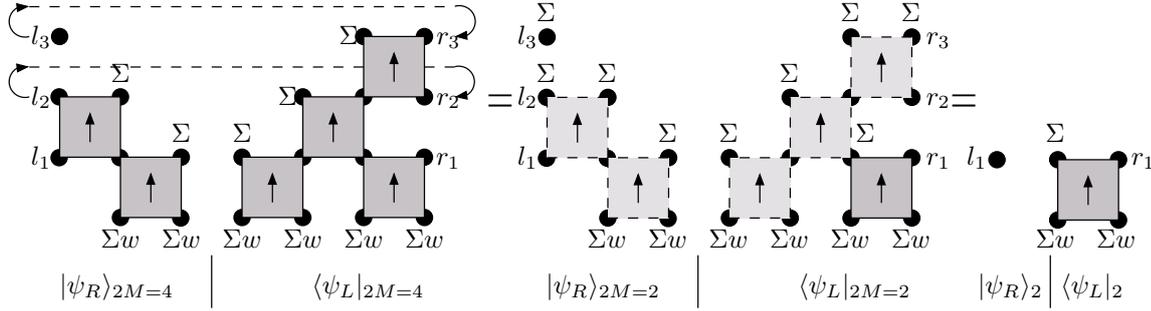


Abbildung 3.7: Bei der asymmetrischen reduzierten Dichtematrix für das System, $\rho_{\text{sys}}^{\text{asym}}$, liefert die Spur über den Umgebungsblock die reduzierte Dichtematrix eines reinen Zustandes: $\text{Tr}_{\text{env}} |\psi_R\rangle\langle\psi_L| = |\psi_R\rangle_{\text{sys}}\langle\psi_L|_{\text{sys}}$.

Das Spektrum ist folglich

$$\text{Spec}(\rho_{\text{sys}}^{\text{asym}}) = \{1, 0, \dots, 0\}. \quad (3.32)$$

Der Renormierungsschritt projiziert also auf einen nur eindimensionalen Unterraum, was offensichtlich nicht genügt: die *stochastischen Randbedingungen am Ende mitteln die physikalische Wechselwirkung der Umgebung weg* (vgl. Gleichung (3.31)), da es nun nicht mehr darauf ankommt, wie sich das System in der Zukunft verhält. In den Randbedingungen liegt also der entscheidende Unterschied zur Quanten-TMRG.

Die Projektion auf einen einzigen Basisvektor entspricht einer räumlichen Mittelung, so daß alle räumlichen Korrelationen mit der Renormierung verloren gehen, man bekommt im wesentlichen nur *mean-field*-Verhalten. Um einen Ausweg zu finden, betrachten wir noch einmal die genaue Gestalt von rechtem und linkem Eigenvektor.

3.3.4 Die symmetrische Dichtematrix

Betrachten wir den dominanten linken Eigenvektor der Transfermatrix in aufeinanderfolgenden Zeitschritten (Abbildung 3.8). Angenommen, wir wollen eine Basis finden, um $\langle\psi_L|$ bei $2M = 5$ zu renormieren. Die ersten beiden Indizes (l_1 und l_2) seien das System, die folgenden (l_3 und l_4) die Umgebung. Nachdem man die Spur über die Umgebung gebildet hat, haben die neuen Basisvektoren nur noch die Indizes l_1 und l_2 und haben Ähnlichkeit mit $\langle\psi_L|_{2M=3}$. Die beiden Systemindizes zeichnen sich dadurch aus, daß sie stark korreliert sind, denn sie sind durch eine Kante von τ verbunden, und in kurzen Zeitintervallen finden in der Regel wenige Reaktionen statt, also haben l_1 und l_2 höchstwahrscheinlich die gleiche Belegung.

Um nun zum nächstgrößeren Eigenvektor $\langle\psi_L|_{2M=6}$ zu kommen, müssen wir $\langle\psi_L|_{2M=5}$ mit der Matrix C_6 multiplizieren, die durch eine Spalte von drei τ 's gegeben ist. Haben

Eigenvektoren im selben Zeitschritt gut repräsentiert.

Für die asymmetrische Dichtematrix des vorigen Abschnitts hatten wir das Minimierungsproblem für den Approximationsfehler der DMRG (Abschnitt 3.2) so erweitert, daß wir nach biorthonormalen Basen gesucht haben, in denen $|\psi_R\rangle$ und $\langle\psi_L|$ getrennt in der linken bzw. rechten Basis optimal repräsentiert werden:

$$\min_{\{\text{biorth. Basen}\}} \text{Tr}_{\text{sys}} \left[(\psi_L - \tilde{\psi}_L)(\psi_L - \tilde{\psi}_L)^\dagger + (\psi_R - \tilde{\psi}_R)(\psi_R - \tilde{\psi}_R)^\dagger \right]. \quad (3.34)$$

Stattdessen suchen wir nun eine *gemeinsame*, orthonormale Basis für $|\psi_R\rangle$ und $\langle\psi_L|$:

$$\min_{\{\text{orth. Basen}\}} \text{Tr}_{\text{sys}} \left[(\psi_L - \tilde{\psi}_L)(\psi_L - \tilde{\psi}_L)^\dagger \right] + \text{Tr}_{\text{sys}} \left[(\psi_R - \tilde{\psi}_R)(\psi_R - \tilde{\psi}_R)^\dagger \right]. \quad (3.35)$$

Über die SVD erhalten wir folgende Vorschrift für die *symmetrisierte reduzierte Dichtematrix*:

$$\rho_{\text{sys}}^{\text{symm}} = \text{Tr}_{\text{env}} (|\psi_R\rangle\langle\psi_R| + |\psi_L\rangle\langle\psi_L|) \quad (3.36)$$

(mit geeigneter Normierung, so daß $\text{Tr}_{\text{sys}}(\rho_{\text{sys}}) = 1$). Die Eigenvektoren der m größten Eigenwerte bilden die Basis des Unterraumes, in dem sich sowohl $|\psi_R\rangle$ als auch $\langle\psi_L|$ gut darstellen lassen.

An einem einfachen stochastischen Modell, dessen exakte Lösung bekannt ist, können wir die Renormierung mit der symmetrisierten Dichtematrix testen (Kapitel 4): die Übereinstimmung ist sehr gut.

Die Idee der symmetrisierten Dichtematrix scheint also die Erwartung zu erfüllen, das physikalisch richtige Verhalten zu simulieren. Allerdings führt sie zu numerischen Problemen: die Simulation bricht nach rund 100 Zeitschritten ab, weil linker und rechter Eigenvektor einen extrem kleinen Überlapp verglichen mit ihrer euklidischen Norm haben. Wählen wir sehr kleine Zeitschritte, $\Delta t \rightarrow 0$, dann ist τ eine Einheitsmatrix (Gleichung (2.26)),

$$\lim_{\Delta t \rightarrow 0} (\tau)_{l_1 r_1}^{l_2 r_2} = \begin{array}{c} l_2 \bullet \\ | \\ l_1 \bullet \end{array} \uparrow \begin{array}{c} \bullet r_2 \\ | \\ \bullet r_1 \end{array} = \delta_{l_1 l_2} \cdot \delta_{r_1 r_2}, \quad (3.37)$$

und die linken und rechten Eigenvektoren der Transfermatrix lassen sich exakt angeben ($2M$ gerade):

$$\psi_L = w(\sigma_1) \cdot \delta_{\sigma_2 \sigma_3} \cdots \delta_{\sigma_{2M-2}, \sigma_{2M-1}} \quad (3.38a)$$

$$\psi_R = \delta_{\sigma_1 \sigma_2} \cdot \delta_{\sigma_3 \sigma_4} \cdots \delta_{\sigma_{2M-3}, \sigma_{2M-2}} \quad (3.38b)$$

Den *dummy*-Index σ_{2M-1} von ψ_R haben wir deshalb so genannt, weil er gar nicht vorkommt, der Wert von ψ_R also nicht vom seinem letzten Index abhängt. Offensichtlich ist

$\langle \psi_L | \psi_R \rangle = \sum_{\{\sigma_i\}} \psi_L \psi_R = \sum_{\sigma_1} w(\sigma_1) = 1$ erfüllt. Jedoch haben ψ_L und ψ_R exponentiell wachsende euklidische Norm:

$$\langle \psi_L | \psi_L \rangle = \sum_{\sigma_1} w^2(\sigma_1) \cdot \delta_{\sigma_2 \sigma_3}^2 \cdots \delta_{\sigma_{2M-2}, \sigma_{2M-1}}^2 = \sum_{\sigma_1} w^2(\sigma_1) \cdot n^{M-1} \quad (3.39a)$$

$$\langle \psi_R | \psi_R \rangle = \sum_{\sigma_1} \delta_{\sigma_1 \sigma_2}^2 \cdots \delta_{\sigma_{2M-3}, \sigma_{2M-2}}^2 = n^{M-1} \quad (3.39b)$$

Die Renormierung ist eine orthogonale Transformation auf dem Unterraum des Zustandsraumes, in dem sich das System mit größter Wahrscheinlichkeit aufhält, ist also normerhaltend. Die Normen von ψ_L und ψ_R wachsen also weiter exponentiell mit M an, während die Zahl der Dimensionen durch die Renormierung auf m beschränkt bleibt, d.h. auch die Komponenten von ψ_L und ψ_R müssen exponentiell anwachsen. Gleichzeitig erfüllen sie die Normierungsbedingung $\langle \psi_L | \psi_R \rangle = 1$, d.h. die großen Einträge müssen sich fast perfekt kompensieren. Dazu ist eine Fließkomma-Genauigkeit von $\mathcal{O}(M)$ Binärstellen für $n = 2$ notwendig, oder allgemein $\mathcal{O}(M \cdot \log_2 n)$. Verdoppelt man die Zahl der Binärstellen, so lassen sich rund doppelt so viele Zeitschritte simulieren.

3.3.5 Alternativen

Um zu überprüfen, ob die symmetrisierte reduzierte Dichtematrix (3.36) wirklich optimal ist, haben wir viele andere asymmetrische und symmetrische Dichtematrizen untersucht, aber keine hatte ein nur annähernd so gutes Konvergenzverhalten:

$$\text{Tr}_{\text{env}}(|\psi_R\rangle\langle\psi_L| + |\psi_L\rangle\langle\psi_R|) \quad (3.40a)$$

$$\text{Tr}_{\text{env}}|\psi_R + \psi_L\rangle\langle\psi_R + \psi_L| \quad (3.40b)$$

$$\text{Tr}_{\text{env}}|\psi_R - \psi_L\rangle\langle\psi_R - \psi_L| \quad (3.40c)$$

$$\text{Tr}_{\text{env}}|\sqrt{\psi_R \bullet \psi_L}\rangle\langle\sqrt{\psi_R \bullet \psi_L}| \quad (\text{komponentenweises Produkt und Wurzel}) \quad (3.40d)$$

Ebenso haben wir versucht, mit dem Schmidtschen Orthogonalisierungsverfahren aus den getrennten System- und Umgebungs-Singulärvektoren von ψ_L und ψ_R , $|u^\alpha\rangle$ und $|v^\alpha\rangle$, direkt eine biorthonormale Basis zu erzeugen.

Um das Problem der großen Zahlen in ψ zu umgehen, haben wir das Gitter um einen halben Platz verschoben:

$$T = \sqrt{T_1} \cdot T_2 \cdot \sqrt{T_1}, \quad (3.41)$$

wobei T_1 und T_2 die beiden versetzten Spalten von T sind. Wenn τ paritätsinvariant ist, dann ist die so definierte Transfermatrix T symmetrisch, $\langle\psi_L| = |\psi_R\rangle^T =: \langle\psi|$ und $\langle\psi|\psi\rangle = \langle\psi_L|\psi_R\rangle = 1$, und auch $\rho_{\text{sys}} = \text{Tr}_{\text{env}}|\psi\rangle\langle\psi|$ ist symmetrisch. Wie schon bei der

asymmetrischen Dichtematrix ist aber das Spektrum von ρ_{sys} trivial wegen der stochastischen Randbedingungen.

Da bei $\Delta t \rightarrow 0$ Probleme auftreten, wenn die Übergangswahrscheinlichkeiten klein sind, haben wir auch eine Störungsentwicklung nach Ordnungen von Δt in Erwägung gezogen. Das geht, sobald wir mit der expliziten Darstellung der Eigenvektoren das Problem linearisiert haben (statt das nichtlineare Eigenwertproblem für T lösen zu müssen), so daß wir verschiedene Beiträge zu ψ getrennt in der Zeit verfolgen können. Allerdings wächst der kombinatorische Vorfaktor so schnell, daß der Aufwand für die Buchhaltung der verschiedenen Terme exponentiell anwächst, was die Behandlung ebenso aufwendig wie die des exponentiellen Wachstums des Zustandsraumes macht.

4 TMRG-Ergebnisse für Reaktions-Diffusions-Prozesse

Die Motivation, die TMRG auf stochastische Modelle anzuwenden, ist, den Zusammenhang zwischen mikroskopischem und makroskopischem Verhalten am Beispiel von Reaktions-Diffusions-Prozessen zu untersuchen.

Für manche Systeme kann man die mikroskopische Dynamik realistisch modellieren, z.B. chemische Reaktionen, Bevölkerungsentwicklung oder die Ausbreitung von Epidemien. Daraus lassen sich typischerweise nichtlineare Differentialgleichungen für das makroskopische Verhalten ableiten. In anderen Fällen kennt man nichtlineare makroskopische Gleichungen, die aber praktisch nicht lösbar sind, und erfindet dafür ein stochastisches mikroskopisches Modell, das man auf einem Gitter simulieren kann.

DE MASI, FERRARI und LEBOWITZ [DFL85] haben im Detail untersucht, unter welchen mikroskopischen Bedingungen das makroskopische System *mean-field*-Verhalten zeigt. Im wesentlichen muß die Diffusionsrate viel größer als die Reaktionsrate sein, dann trennen sich mikro- und makroskopische Skalen, und das Gesetz der großen Zahlen garantiert deterministische makroskopische Gleichungen trotz mikroskopischer Fluktuationen.

4.1 Paar-Annihilation

Dieser exakten analytischen Behandlung entziehen sich jedoch häufig Systeme, bei denen Reaktions- und Diffusionsraten von derselben Größenordnung sind: hier kann das makroskopische Verhalten deutlich vom *mean-field*-Verhalten abweichen. Ein Beispiel ist der bereits in Abschnitt 2.3 vorgestellte Paar-Annihilations-Prozeß. Er beschreibt z.B. die thermische Wechselwirkung von Soliton-Antisoliton-Paaren nach der Photoanregung von Trans-Polyacetylen [Rác85]. Als „Medium“ nehmen wir das eindimensionale Gitter der ganzen Zahlen, \mathbb{Z} . Anfänglich sind Monomere auf diesem Gitter verteilt, jeder Gitterplatz hat unabhängig die Besetzungswahrscheinlichkeit w (Gleichung (2.10)). Danach diffundieren die Monomere identisch und unabhängig mit der Rate D . Wenn zwei Monomere aufeinandertreffen, bilden sie irreversibel Paare (Dimere), die aus der Betrachtung

der Monomere herausfallen, mit der Rate 2α :



Ist D sehr groß, so beobachten wir makroskopisch *mean-field*-Verhalten: die mittlere Konzentration von A folgt der Gleichung

$$\dot{a}(t) = -4\alpha \cdot a^2(t), \quad a(0) = a_0 \quad (4.3)$$

mit der Lösung, daß sich das Langzeitverhalten wie $1/t$ verhält:

$$a(t) = \langle \mathcal{A}(t) \rangle = \frac{a_0}{1 + 4\alpha a_0 t} \longrightarrow \frac{1}{4\alpha t} \quad (t \rightarrow \infty). \quad (4.4)$$

Dies ist die kinetische Gleichung, sie ist unabhängig von der Dimension des Ortsraumes.

Ist dagegen $D = \alpha$, d.h. die Reaktionsrate genau doppelt so groß wie die Diffusionsrate, dann ist das System stark korreliert, und das *mean-field*-Ergebnis gilt nicht mehr (hier für eine Dimension, [Spo88]):

$$a(t) = a_0 e^{-4Dt} \left[I_0(4Dt) + 2(1 - a_0) \sum_{k=1}^{\infty} (1 - 2a_0)^{k-1} I_k(4Dt) \right] \quad (4.5)$$

$$\longrightarrow \frac{1}{\sqrt{8\pi Dt}} \quad (t \rightarrow \infty), \quad (4.6)$$

wobei $I_k(x)$ die modifizierte Besselfunktionen bezeichnet [Abr72]. Dieses Ergebnis unterscheidet sich von dem *mean-field*-Ergebnis dadurch, daß man hier räumliche Korrelationen berücksichtigt hat. Die Annihilation ist limitiert durch die Diffusion: der Abstand zum nächsten Reaktionspartner ist im Mittel $\sim t^{d/2}$ in d Dimensionen.

Die anfängliche Dichte $a_0 = a(0)$ der Teilchensorte A berechnet sich aus der anfänglichen lokalen Wahrscheinlichkeit $w(\sigma)$ der Konfigurationen σ jedes Gitterplatzes:

$$a_0 = \sum_{\sigma=1}^n w(\sigma) \mathcal{A}(\sigma). \quad (4.7)$$

Es gibt zwei Spezialfälle, in denen sich Gleichung (4.5) vereinfacht (mit $\mathcal{A}(\emptyset) = 0$ und $\mathcal{A}(A) = 1$):

1. $w(A) = 1$ und $w(\emptyset) = 0$, d.h. $a_0 = 1$:

$$a(t) = e^{-4Dt} I_0(4Dt) \sim t^{-1/2}. \quad (4.8)$$

2. $w(A) = w(\emptyset) = \frac{1}{2}$, d.h. alle Gitterplätze sind unabhängig mit Wahrscheinlichkeit $\frac{1}{2}$ besetzt, und $a_0 = \frac{1}{2}$:

$$a(t) = \frac{1}{2}e^{-4Dt}[I_0(4Dt) + I_1(4Dt)] \sim t^{-1/2}. \quad (4.9)$$

In Abschnitt 3.3 haben wir gezeigt, daß die Renormierung mit der asymmetrischen Dichtematrix (3.30) bei jedem Renormierungsschritt räumlich mittelt, also zu *mean-field*-Verhalten führt, während die symmetrisierte Dichtematrix (3.36) dank der räumlichen Korrelationen das physikalisch richtige Verhalten liefert.

Damit haben wir einen guten Testfall für die Qualität unserer TMRG-Simulation (siehe Abbildung 4.1). Verwenden wir die symmetrische Dichtematrix, so reproduzieren wir den erwarteten $1/\sqrt{t}$ -Verlauf (Gleichung (4.5)). Mit der asymmetrischen Dichtematrix bekommen wir einen $1/t$ -Verlauf (Gleichung (4.4)).

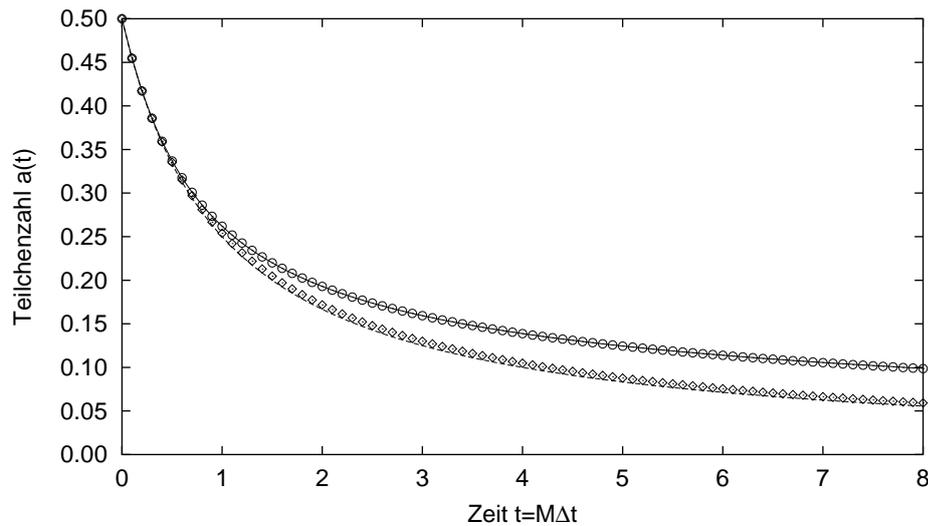
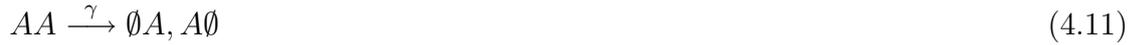


Abbildung 4.1: TMRG-Simulation des Paar-Annihilations-Prozesses mit symmetrischer und asymmetrischer Dichtematrix. (—) exakte Lösung (4.9), (O) Simulation mit der symmetrischen Dichtematrix, (---) *mean-field*-Näherung (4.4), (◇) Simulation mit der asymmetrischen Dichtematrix ($a_0 = 1/2$, $\alpha = D = 0.5$ und $\Delta t = 0.1$).

4.2 Branching-fusing-Prozeß

Als Anwendung, deren exakte Lösung nicht bekannt ist, haben wir den *branching-fusing*-Prozeß [CHS99] gewählt:



Zwischen den Raten besteht folgender Zusammenhang:

$$D = 2\alpha = \gamma = \delta = 1 - p, \quad \beta = p. \quad (4.15)$$

Bei $p_c \approx 0.84036$ (der exakte Wert ist nicht bekannt) gibt es einen kritischen Punkt in der Universalitätsklasse der gerichteten Perkolation (*directed percolation*, DP). Zunächst testen wir die Abhängigkeit der TMRG-Ergebnisse von der Anzahl der mitgenommenen Zustände m (siehe Abbildung 4.2) am kritischen Punkt, an dem die TMRG am langsamsten konvergiert.

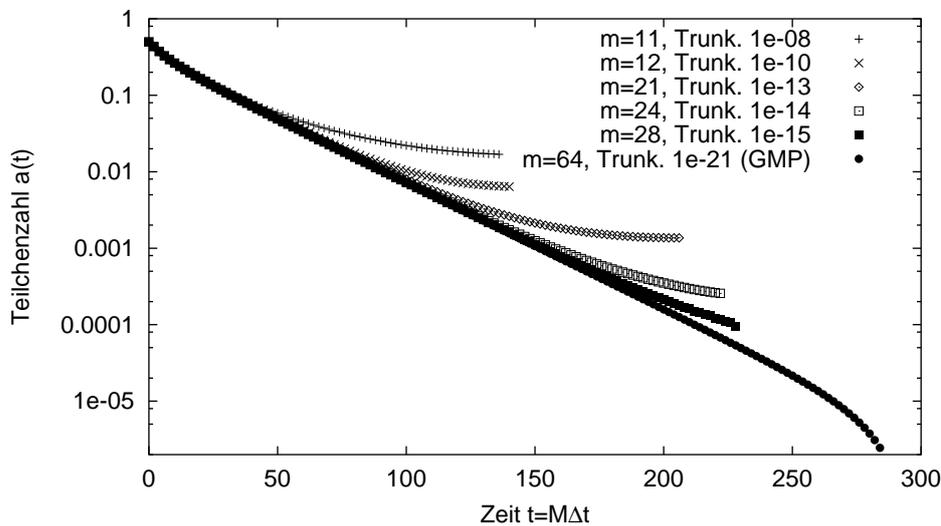


Abbildung 4.2: *Branching-fusing*-Prozeß [CHS99] am kritischen Punkt $p = 0.84036$ für unterschiedliche Anzahl erhaltener Zustände (m) bzw. Trunkierung des Spektrums von ρ_{sys} .

Wir beobachten, daß die Teilchenzahl exponentiell abfällt, obwohl wir am kritischen Punkt einen algebraischen Abfall erwartet hätten. Das liegt daran, daß wir nur sehr kurze Zeiten simuliert haben, zu denen der algebraische Abfall noch vom exponentiellen Kurzzeitverhalten überdeckt wird. Das Ergebnis wird immer genauer, je größer die Anzahl der Zustände m ist, d.h. je kleiner der Eigenwert der Dichtematrix ist, an dem das Spektrum trunziert wird. Bei $m = 28$ erreichen wir eine Grenze, weil die Trunkierung bei 10^{-15} etwa der Fließkomma-Genauigkeit der *double-precision*-Zahlen (53 Mantissenbits) entspricht. Um eine höhere Genauigkeit zu erreichen und mehr Zeitschritte simulieren zu können, muß man die Anzahl der Mantissenbits erhöhen: dies geht z.B. mit der GNU *multi-precision* (GMP)-Bibliothek, die Fließkommazahlen beliebiger Genauigkeit zur Verfügung stellt, was sehr flexibel, aber langsam ist.

4.3 Approximationsfehler

Es ist zu erwarten, daß sich die Abweichung vom exakt bekannten Resultat wie der Trotterfehler (2.15) verhält, also proportional zu $(\Delta t)^2$. Das ist genau, was die Simulation für kurze Zeiten ergibt (siehe Abbildung 4.3): bei $t = 0.4$ erkennt man deutlich, daß der absolute Fehler für doppelt so große Zeitschritte etwa viermal so groß ist.

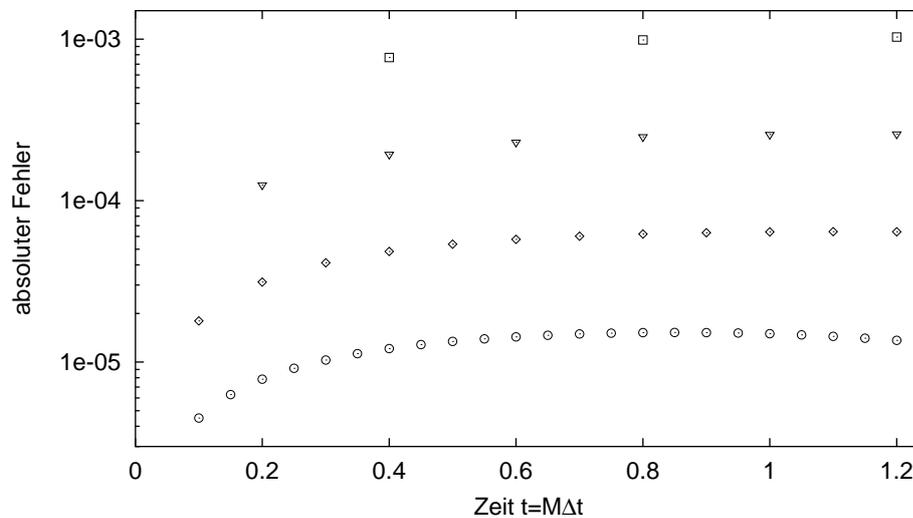


Abbildung 4.3: Abweichung der TMRG-Simulation des Paar-Annihilations-Prozesses vom exakten Ergebnis für kurze Zeiten mit verschiedenen Zeitschritten Δt : (○) $\Delta t = 0.05$, (◇) $\Delta t = 0.1$, (▽) $\Delta t = 0.2$, (□) $\Delta t = 0.4$.

Für längere Zeiten ändert sich das Verhalten des Fehlers jedoch entscheidend (siehe Abbildung 4.4): während für kleine Zeitschritte $\Delta t \leq 0.05$ der Fehler stärker als linear mit

der Trotterzahl M ansteigt, steigt er für $\Delta t > 0.05$ schwächer als linear an. Somit lassen sich für dieses spezielle System durch größere Zeitschritte Δt nicht nur die Dynamik über längere Zeiten verfolgen, sondern zusätzlich noch genauere Ergebnisse erzielen.

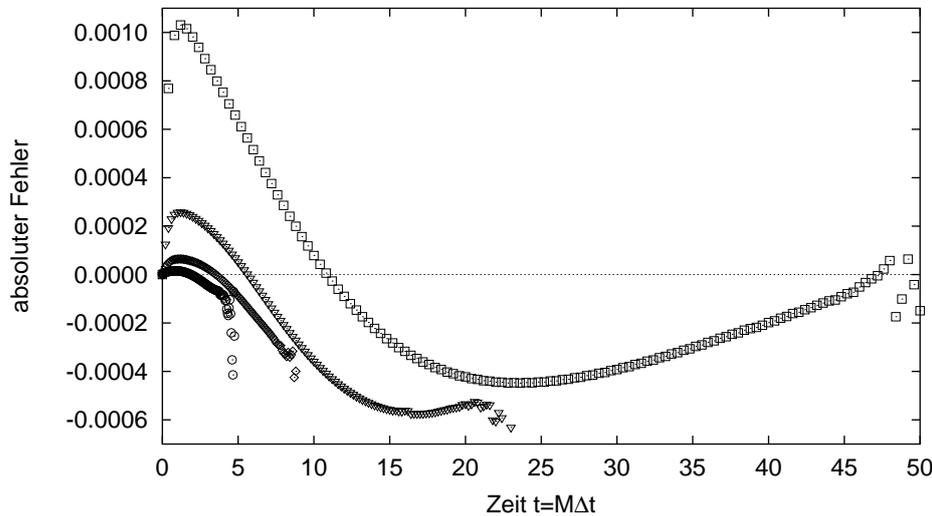


Abbildung 4.4: Abweichung der TMRG-Simulation des Paar-Annihilations-Prozesses vom exakten Ergebnis für längere Zeiten mit verschiedenen Zeitschritten Δt : (\circ) $\Delta t = 0.05$, (\diamond) $\Delta t = 0.1$, (∇) $\Delta t = 0.2$, (\square) $\Delta t = 0.4$.

4.4 Spektrum der Dichtematrix

Nachdem die verschiedenen Dichtematrizen so unterschiedliche Spektren haben, stellt sich die Frage, welche Form des Spektrums am geeignetsten ist. Offenbar müssen wir zwei gegensätzliche Forderungen stellen:

1. Je mehr nichtsinguläre Eigenwerte $\lambda_i > 0$ das Spektrum hat, desto mehr Information über die räumlichen Korrelationen und über die zukünftige Entwicklung im Umgebungsblock bleiben bei der DMRG-Projektion erhalten.
2. Je kleiner die trunkierten Eigenwerte λ_i , $i > m$, sind, desto kleiner ist der Approximationsfehler Δ .

Die asymmetrische Dichtematrix mit nur einem Eigenwert $\lambda_0 = 1$ erfüllt 2. perfekt, $\Delta = 0$, aber erfüllt 1. sehr schlecht. Bei der symmetrischen Dichtematrix (für den *branching-fusing*-Prozeß am kritischen Punkt) hat das Spektrum dagegen folgende Gestalt

(siehe Abbildung 4.5): in den ersten Renormierungsschritten fallen die Eigenwerte ungefähr exponentiell ab, $\lambda_i \sim \alpha^{-i}$ für ein konstantes α . Nach vielen Schritten fallen die Eigenwerte ungefähr algebraisch ab, $\lambda_i \sim i^{-\beta}$ mit großem β (≈ 13). In beiden Fällen erfüllt die symmetrische Dichtematrix die beiden Forderungen exzellent.

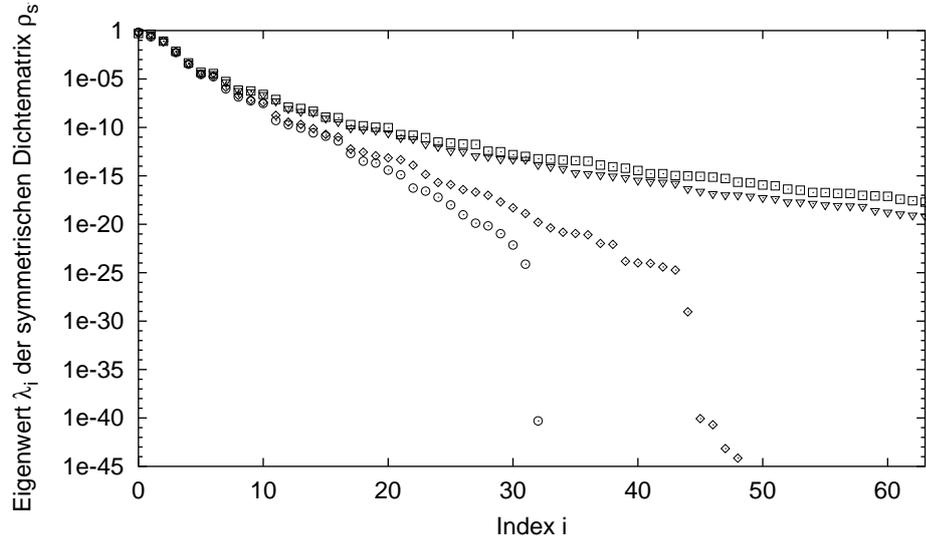


Abbildung 4.5: Spektrum der symmetrischen reduzierten System-Dichtematrix $\rho_{\text{sys}}^{\text{symm}}$ für den *branching-fusing*-Prozeß am kritischen Punkt zu bestimmten Trotterschritten $2M$ ($m = 64$). (\circ) $2M = 18$ (trunk. 10^{-40}), (\diamond) $2M = 20$ (trunk. 10^{-29}), (∇) $2M = 60$ (trunk. 10^{-21}), (\square) $2M = 280$ (trunk. 10^{-21}).

4.5 Spektrum der Transfermatrix

Wie bereits in Abschnitt 2.5 gezeigt, ist das Spektrum der Transfermatrix, $\text{Spec}(T) = \{1, 0, \dots, 0\}$, trivial. Nach dem ersten Renormierungsschritt beobachten wir jedoch, daß der zweite Eigenwert von Null verschieden ist und sich dem ersten Eigenwert in den darauffolgenden Iterationen immer weiter nähert. Das liegt daran, daß eine Projektion, wie es die Dichtematrix-Projektion ist, nicht das Spektrum einer Matrix (hier der Transfermatrix) erhält, wie man an folgendem Beispiel sieht:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \Rightarrow T' = B^T T B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix} \quad (4.16)$$

T ist die typische Jordanform der Transfermatrix mit nilpotentem Anteil und Spektrum $\{1, 0, 0\}$, T' hat offensichtlich dasselbe Spektrum, aber wenn man die letzte Zeile und Spalte von T' wegläßt (Projektion), dann hat T' die Eigenwerte $\{1, 1/2\}$. Die wesentliche

Beobachtung ist, daß durch eine Projektion auch der nilpotente Anteil von T Einfluß auf das Spektrum von T' nimmt. Mit diesem zweiten nichtsingulären Eigenwert, $\lambda_1 > 0$, hat das System nach Gleichung (2.42) eine endliche Korrelationslänge und exponentiell abklingende räumliche Korrelationen.

4.6 Methode der expliziten Eigenvektoren

Schließlich wollen wir die numerische Stabilität des gewöhnlichen TMRG-Algorithmus, bei dem sowohl System als auch Umgebung renormiert werden, mit der erstmals aufgezeigten Möglichkeit vergleichen, die stationären Eigenvektoren explizit zu konstruieren. Weil man hier nicht mehr das Eigenwert-Problem der Transfermatrix lösen muß, spart man die Iterationen zur Diagonalisierung der Transfermatrix, und somit über die Hälfte der Rechenzeit. Allerdings kann man so nur das System renormieren und behält die Umgebung unrenormiert als *look-ahead* konstanter endlicher Länge: weil wir neue Gitterplätze nur am *Ende* der Zeitentwicklung einfügen können (siehe Abbildung 3.8), müssen wir zur Vergrößerung des Systems ehemalige Umgebungsgitterplätze verwenden. Damit wir den ersten Umgebungsgitterplatz abseparieren und zum System hinzufügen können, muß die Umgebung in einer unrenormierten Ortsbasis gespeichert bleiben.

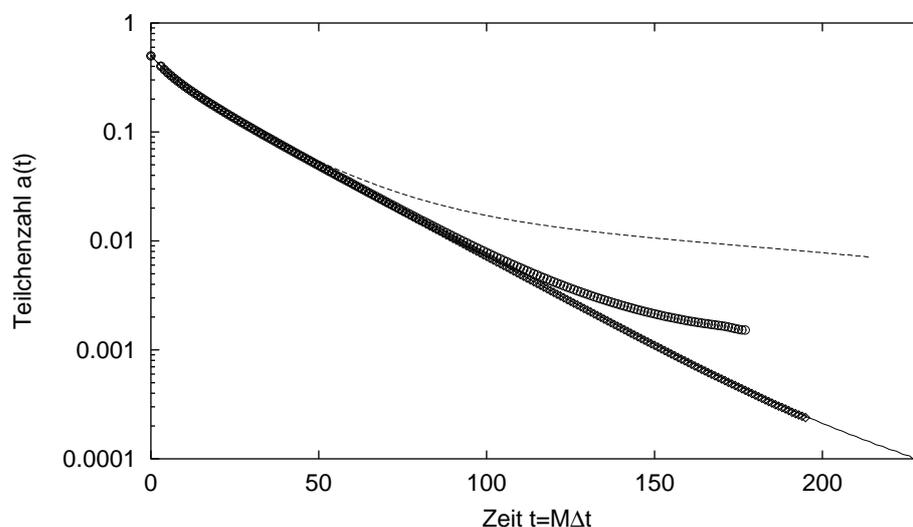


Abbildung 4.6: *Branching-fusing*-Prozeß am kritischen Punkt mit unterschiedlichen TMRG-Methoden. (—) herkömmliche Methode, $m = 28$; (---) dito mit $m = 11$; (○) explizite Eigenvektoren, $m = 8$ (4 Gitterplätze *look-ahead*); (◇) dito mit $m = 11$ (6 Gitterplätze *look-ahead*).

Das Ergebnis für den *branching-fusing*-Prozeß ist in Abbildung 4.6 dargestellt: für eine

mäßige Genauigkeit von $\epsilon = 10^{-8}$ ($m = 11$ Zustände und 6 Gitterplätze *look-ahead*) ist die Methode der expliziten Eigenvektoren sowohl schneller als auch genauer als die gewöhnliche Methode. Nimmt man allerdings mehr Zustände, so verbessert sich das explizite Ergebnis nicht weiter, während es sich bei der herkömmlichen Methode bis $\epsilon = 10^{-15}$ (Maschinengenauigkeit) verbessert ($m = 28$). Das mag daran liegen, daß die renormierte Umgebung zum Schluß $M \approx 100$ Gitterplätze repräsentiert, während bei festem *look-ahead* wegen des exponentiellen Wachstums des Zustandsraumes nur etwa 12 Gitterplätze praktikabel sind (4096 Zustände bei $n = 2$). Solange sich nicht eine Möglichkeit findet, neue Gitterplätze in die Mitte der expliziten Eigenvektoren einzufügen und beide Blöcke zu renormieren, bleibt diese Methode vorerst eher von theoretischem Interesse für analytische Untersuchungen der Struktur der Transfermatrix.

5 Implementierung des TMRG-Algorithmus

Der TMRG-Algorithmus für Quantensysteme, in denen die Zeitachse periodisch geschlossen ist, ist bereits in [Nis95, WX97, MS98] detailliert beschrieben. Die besten Ergebnisse wurden hier mit der asymmetrischen Dichtematrix erzielt. KEMPER *et al.* [Kem00, KSZ01] beschreiben die Änderungen für stochastische Systeme: am Ende der Zeitentwicklung gelten offene Randbedingungen, und empirisch wurde das beste Konvergenzverhalten mit der symmetrischen Dichtematrix beobachtet.

Da der Algorithmus also nicht neu ist, beschränken wir uns auf eine kurze Darstellung. Falls die Leserin oder der Leser im Zweifel über ein Detail der Implementierung ist, kann der abgedruckte Programmcode in Abschnitt 5.2 eine definitive Antwort geben.

5.1 Beschreibung des Algorithmus

Da die thermodynamischen Eigenschaften des Systems im stationären Eigenraum der Transfermatrix kodiert sind, basiert der TMRG-Algorithmus auf der exakten Diagonalisierung der Transfermatrix. Da jedoch der Zustandsraum und damit die Dimension der Transfermatrix zu stark anwächst, müssen wir uns auf einen Unterraum des Zustandsraumes beschränken. Die Auswahl des optimalen Unterraumes geschieht mit Hilfe der Dichtematrix-Renormierung.

Man hat den folgenden Ablauf:

1. Initialisiere τ und setze die anfänglichen System- und Umgebungsblocks (S , E und $E[\mathcal{A}]$), indem auf τ geeignete Randbedingungen angewandt werden.
2. Finde die Eigenvektoren $|\psi_R\rangle$ und $\langle\psi_L|$ zum größten Eigenwert der Transfermatrix $T = \text{Tr}(S \otimes E)$ (exakte Diagonalisierung). Berechne den Erwartungswert $\langle\mathcal{A}\rangle = \langle\psi_L|T[\mathcal{A}]|\psi_R\rangle$.
3. Vergrößere S , E etc. in der Mitte der Zeitentwicklung um jeweils ein τ .

4. Renormiere, wenn der Zustandsraum von S , E etc. zu groß wird:
 - a) Bilde die reduzierten Dichtematrizen ρ_{sys} und ρ_{env} .
 - b) Zerlege ρ_{sys} in Eigenvektoren und wähle die Eigenvektoren zu den größten Eigenwerten als Basis für den optimalen Unterraum, U_{sys} , aus. Analog für ρ_{env} .
 - c) Projiziere S , E etc. auf diese Unterräume.
5. Gehe zu Schritt 2.

5.1.1 Initialisierung

Bei Prozessen in diskreter Zeit, wie z.B. den DOMANI-KINZEL-Zellulärautomaten [KSZ01], ist die lokale Transfermatrix τ direkt gegeben. Andernfalls bestimmt man sie aus dem stochastischen Hamiltonoperator, der sich aus den Ratengleichungen zusammensetzt, indem man

$$\tau = e^{-\Delta t \cdot h} \quad (5.1)$$

bildet. Dazu überführt man zunächst h mit einer Ähnlichkeitstransformation in seine Jordanform J . Die Diagonalelemente (Eigenwerte) von J exponenziert man komponentenweise und bildet explizit die endlichen Potenzen des nilpotenten Anteils von J , bis er verschwindet. Die so gewonnene Matrix transformiert man mit der Ähnlichkeitstransformation wieder zurück in die ursprüngliche Basis und erhält

$$(\tau)_{r_1 r_2}^{l_1 l_2} = [\exp(-\Delta t \cdot h)]_{r_1 r_2}^{l_1 l_2} = \begin{array}{c} l_2 \bullet \\ \uparrow \Delta t \\ l_1 \bullet \end{array} \begin{array}{c} r_2 \\ \bullet \\ r_1 \end{array} \quad (5.2)$$

(mit der Konvention, daß die oberen Indizes einer Matrix in der bildlichen Darstellung links und die unteren Indizes rechts sind). Den Systemblock $S_s^{s' \sigma'}$ erhält man daraus, indem man die Randbedingungen für den Anfangszustand auf τ anwendet:

$$(S_{\text{init}})_{s \sigma}^{s' \sigma'} = \sum_{l_1 r_1} w(l_1) w(r_1) \cdot (\tau)_{r_1 \sigma}^{l_1 \sigma'} = \begin{array}{c} \sigma' \bullet \\ \uparrow \Delta t \\ \sigma \bullet \end{array} \begin{array}{c} \sigma \\ \bullet \\ \sigma \end{array} \cdot \quad (5.3)$$

Der Umgebungsblock $E_{\sigma e}^{\sigma' e'}$ ist noch trivial wegen (2.18):

$$(E_{\text{init}})_{\sigma e}^{\sigma' e'} = \sum_{l_2 r_2} (\tau)_{\sigma r_2}^{\sigma' l_2} = 1 = \begin{array}{c} \Sigma_{l_2} \bullet \\ \uparrow \Delta t \\ \sigma' \bullet \end{array} \begin{array}{c} \Sigma_{r_2} \\ \bullet \\ \sigma \end{array} \quad (5.4)$$

Die Systemindizes s , s' und die Umgebungsindizes e , e' sind für $M = 0$ noch irrelevant, da sie nur $m = n^M = 1$ Wert annehmen können.

Um später Erwartungswerte berechnen zu können, die sich in veränderten Randbedingungen am Ende der Zeitentwicklung ausdrücken, müssen wir auch die so modifizierte Transfermatrix $T[\mathcal{A}]$ berechnen. Sie setzt sich zusammen aus dem unveränderten Systemblock S und einem modifizierten Umgebungsblock $E[\mathcal{A}]$:

$$(E[\mathcal{A}]_{\text{init}})_{\sigma e}^{\sigma' e'} = \sum_{l_2 r_2} \mathcal{A}(l_2)(\tau)_{\sigma r_2}^{\sigma' l_2} = \begin{array}{c} \Sigma_{l_2} \mathcal{A}(l_2) \\ \sigma' \quad \uparrow \Delta t \quad \sigma \\ \Sigma_{r_2} \end{array}, \quad (5.5)$$

wobei es unerheblich ist, ob man \mathcal{A} am Gitterplatz l_2 oder r_2 mißt.

In unserem Programmbeispiel beschränken wir uns auf paritätsinvariante Systeme (invariant unter räumlicher Spiegelung), d.h. es ist keine Raumrichtung gegenüber der anderen ausgezeichnet. Dann ist τ symmetrisch als Transfermatrix von links nach rechts bzw. von rechts nach links (Gleichung (2.19)):

$$(\tau)_{r_1 r_2}^{l_1 l_2} = (\tau)_{l_1 l_2}^{r_1 r_2}. \quad (5.6)$$

Wie wir bereits in Abschnitt 3.3.4 gesehen haben, ist der linke Eigenvektor $\langle \psi_L |$ gleich dem gespiegelten rechten Eigenvektor $|\psi_R\rangle$, an den noch eine Spalte von τ 's heranzumultipliziert wird. Das ist eine einfache Multiplikation (in der jeweiligen renormierten Basis), die wesentlich schneller geht, als auch den linken Eigenvektor (wie schon den rechten) iterativ zu bestimmen. Bei hoher Rechengenauigkeit (vielen Zuständen) werden leicht mehr als 100 Iterationen benötigt, um den rechten Eigenvektor zu finden. Das dominiert die Laufzeit des Programms. Da man sich mit dieser Optimierung einen nochmal ebenso großen Aufwand für den linken Eigenvektor spart, wird das Programm in diesem Fall fast um einen Faktor zwei schneller.

Die Spalte von τ 's, die man an $|\psi_R\rangle$ heranzumultipliziert, ist zwar ein Tensorprodukt von τ 's, man sollte also idealerweise die τ 's einzeln nacheinander heranzumultiplizieren, allerdings ist $|\psi_R\rangle$ schon in der renormierten Basis, so daß es am zweckmäßigsten ist, wenn auch die Spalte τ 's in derselben Basis gespeichert ist: die eine Hälfte in der Systembasis (CS), die andere in der Umgebungsbasis (CE). Dann kann man sie ganz analog dem System- und Umgebungsblock erweitern und renormieren. Man initialisiert CS und CE also wie folgt (CE_{init} hat keine σ -Indizes, da immer nur eines von beiden auf den mittleren Index σ wirkt):

$$(CS_{\text{init}})_{s \sigma}^{s' \sigma'} := (S_{\text{init}})_{s \sigma}^{s' \sigma'} \quad (n \times n) \quad (5.7)$$

$$(CE_{\text{init}})_e^{e'} := 1 \quad (1 \times 1). \quad (5.8)$$

Nach diesen Initialisierungen beginnen wir im folgenden Schritt die Iteration über die Trotterzahl mit $M = 1$.

5.1.2 Eigenvektoren und Erwartungswerte

Aufgrund der asymmetrischen Anordnung der τ 's in der TROTTER-SUZUKI-Zerlegung müssen wir zwischen geraden und ungeraden Schritten unterscheiden.

M ungerade:

In den ersten Trotterschritten mit $m = n^M \leq m_{\max}$ hat der Systemblock noch eine Einzelbasis, nach der DMRG-Projektion dann eine renormierte m -dimensionale Basis, die wir allgemein wie folgt darstellen:

$$(S_{\text{odd}})_{s\sigma}^{s'\sigma'} = \begin{array}{c} \begin{array}{c} \sigma' \\ [n] \end{array} \begin{array}{c} \uparrow \Delta t \\ [n] \end{array} \begin{array}{c} \sigma \\ [n] \end{array} \\ \begin{array}{c} s' \\ [m] \end{array} \begin{array}{c} \text{[hatched]} \\ [m] \end{array} \begin{array}{c} s \\ [m] \end{array} \end{array} . \quad (5.9)$$

Die obersten Indizes σ und σ' sind noch in der Einzelbasis der zuletzt angefügten τ und werden erst im nächsten Schritt mit renormiert. Analog haben E und $E[\mathcal{A}]$ die folgende Gestalt:

$$(E_{\text{odd}})_{\sigma e}^{\sigma' e'} = \begin{array}{c} \begin{array}{c} e' \\ [m] \end{array} \begin{array}{c} \text{[hatched]} \\ [m] \end{array} \begin{array}{c} e \\ [m] \end{array} \\ \begin{array}{c} \sigma' \\ [n] \end{array} \begin{array}{c} \uparrow \Delta t \\ [n] \end{array} \begin{array}{c} \sigma \\ [n] \end{array} \end{array} . \quad (5.10)$$

Daraus setzt man T_{odd} wie folgt durch eine Spurbildung zusammen:

$$(T_{\text{odd}})_{s\sigma e}^{s'\sigma' e'} = \sum_{\tilde{\sigma}} (E_{\text{odd}})_{\tilde{\sigma} e}^{\sigma' e'} \cdot (S_{\text{odd}})_{s\sigma}^{s' \tilde{\sigma}} = \begin{array}{c} \begin{array}{c} e' \\ [m] \end{array} \begin{array}{c} \text{[hatched]} \\ [m] \end{array} \begin{array}{c} e \\ [m] \end{array} \\ \begin{array}{c} \sigma' \\ [n] \end{array} \begin{array}{c} \uparrow \Delta t \\ [n] \end{array} \begin{array}{c} \tilde{\sigma} \\ [n] \end{array} \begin{array}{c} \uparrow \Delta t \\ [n] \end{array} \begin{array}{c} \sigma \\ [n] \end{array} \\ \begin{array}{c} s' \\ [m] \end{array} \begin{array}{c} \text{[hatched]} \\ [m] \end{array} \begin{array}{c} s \\ [m] \end{array} \end{array} . \quad (5.11)$$

Den Eigenvektor von T_{odd} zum größten Eigenwert 1 finden wir nun mit Hilfe des *power*-Verfahrens: wir beginnen mit einem zufälligen Startvektor $|\psi_R\rangle$ und multiplizieren T so

oft darauf, bis der Vektor $|\psi_R\rangle$ konvergiert ist. Da der $|\psi_R\rangle$ der einzige Eigenvektor der Transfermatrix mit maximalem Eigenwert ist, ist das Ergebnis eindeutig.

$$(T_{\text{odd}} \cdot |\psi_R\rangle)^{s'e'e'} = \sum_{s\sigma e} (T_{\text{odd}})_{s'\sigma'e'}^{s\sigma e} \cdot (|\psi_R\rangle)^{s\sigma e} =$$

$$(5.12a)$$

$$= \sum_{\tilde{\sigma}e} (E_{\text{odd}})_{\tilde{\sigma}e}^{s'e'} \cdot \left(\sum_{s\sigma} (S_{\text{odd}})_{s\sigma}^{s'\tilde{\sigma}} \cdot (|\psi_R\rangle)^{s\sigma e} \right) =$$

$$(5.12b)$$

Gleichung (5.12) zeigt eine weitere wesentliche Optimierung: um $|\psi_R\rangle$ mit T_{odd} zu multiplizieren, multiplizieren wir $|\psi_R\rangle$ zunächst mit S_{odd} und den resultierenden Vektor anschließend mit E_{odd} . Rechenzeit und Speicherbedarf für beide Varianten sind in Tab. 5.1 verglichen. Man spart durch die Optimierung also einen Faktor $m/2$ in der Rechenzeit, so daß man wesentlich größere Unterräume des Zustandsraumes berechnen kann.

| | Rechenzeit | Speicherbedarf |
|--------------------------|------------------|------------------|
| T | $m^4 \cdot n^2$ | $m^4 \cdot n^2$ |
| $\text{Tr}(S \otimes E)$ | $2m^3 \cdot n^2$ | $2m^2 \cdot n^2$ |
| Gewinn | $m/2$ | $m^2/2$ |

Tabelle 5.1: Rechenzeit und Speicherbedarf je nachdem, ob man T explizit berechnet und speichert oder nur S und E .

Ist τ nicht paritätsinvariant, so wird entsprechend auch der linke Eigenvektor $\langle\psi_L|$ von T iterativ bestimmt. Andernfalls können wir $\langle\psi_L|$ schnell aus $|\psi_R\rangle$ gewinnen, indem wir

eine Spalte von τ 's an $|\psi_R\rangle$ heranmultiplizieren:

$$\langle\psi_L|^T = (CS \otimes CE) |\psi_R\rangle = \begin{array}{c} \begin{array}{|c|} \hline \text{e}' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{e} \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \cdot \begin{array}{c} \begin{array}{|c|} \hline \sigma' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{s}' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{s} \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \uparrow \Delta t \\ \hline \end{array} \end{array} . \quad (5.13)$$

Bei der Eigenvektorbestimmung mit dem Iterationsverfahren geht die natürliche Norm von $|\psi_R\rangle$ verloren, also reskalieren wir nun $\langle\psi_L|$ und $|\psi_R\rangle$, so daß

$$Z = \langle\psi_L|\psi_R\rangle = 1 \quad (5.14)$$

gilt. Damit können wir nun den Erwartungswert des lokalen Operators \mathcal{A} ,

$$\langle\mathcal{A}\rangle = \langle\psi_L| \text{Tr} (S \otimes E[\mathcal{A}]) |\psi_R\rangle \quad (5.15)$$

berechnen.

M gerade:

Analog sehen in den geraden Trotterschritten S und E wie folgt aus:

$$(S_{\text{even}})_{s\sigma}^{s'\sigma'} = \begin{array}{c} \begin{array}{|c|} \hline \sigma' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{s}' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{s} \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \uparrow \Delta t \\ \hline \end{array} , \quad (E_{\text{even}})_{\sigma e}^{\sigma' e'} = \begin{array}{c} \begin{array}{|c|} \hline \text{e}' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{e} \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \sigma' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \uparrow \Delta t \\ \hline \end{array} \quad (5.16)$$

und

$$(T_{\text{even}})_{s\sigma e}^{s'\sigma' e'} = \sum_{\tilde{\sigma}} (S_{\text{even}})_{s\tilde{\sigma}}^{s'\tilde{\sigma}} \cdot (E_{\text{even}})_{\tilde{\sigma} e}^{\sigma' e'} = \begin{array}{c} \begin{array}{|c|} \hline \text{e}' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{e} \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \tilde{\sigma} \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline [n] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{s}' \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \begin{array}{|c|} \hline \text{s} \\ \hline \end{array} \\ \begin{array}{|c|} \hline [m] \\ \hline \end{array} \end{array} \begin{array}{c} \uparrow \Delta t \\ \hline \end{array} \begin{array}{c} \uparrow \Delta t \\ \hline \end{array} \quad (5.17)$$

$$(T_{\text{even}} \cdot |\psi_R\rangle)^{s'\sigma'e'} = \sum_{s\sigma e} (T_{\text{even}})_{s\sigma e}^{s'\sigma'e'} \cdot (|\psi_R\rangle)^{s\sigma e} = \sigma' \begin{array}{c} \begin{array}{|c|} \hline e' \\ \hline \end{array} \\ \begin{array}{|c|} \hline \uparrow \Delta t \\ \hline \end{array} \\ \begin{array}{|c|} \hline \tilde{\sigma} \\ \hline \end{array} \\ \begin{array}{|c|} \hline \uparrow \Delta t \\ \hline \end{array} \\ \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline s' \\ \hline \end{array} \end{array} \cdot \begin{array}{c} \begin{array}{|c|} \hline e \\ \hline \end{array} \\ \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline s \\ \hline \end{array} \end{array} \quad (5.18a)$$

$$= \sum_{s\tilde{\sigma}} (S_{\text{even}})_{s\tilde{\sigma}}^{s'\tilde{\sigma}} \cdot \left(\sum_{\sigma e} (E_{\text{even}})_{\tilde{\sigma}e}^{\sigma'e'} \cdot (|\psi_R\rangle)^{s\sigma e} \right) = \sigma' \begin{array}{c} \begin{array}{|c|} \hline e' \\ \hline \end{array} \\ \begin{array}{|c|} \hline \uparrow \Delta t \\ \hline \end{array} \\ \begin{array}{|c|} \hline \tilde{\sigma} \\ \hline \end{array} \\ \begin{array}{|c|} \hline \uparrow \Delta t \\ \hline \end{array} \\ \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline s' \\ \hline \end{array} \end{array} \cdot \begin{array}{c} \begin{array}{|c|} \hline e \\ \hline \end{array} \\ \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline s \\ \hline \end{array} \end{array} \quad (5.18b)$$

5.1.3 Wachstum des Gitters

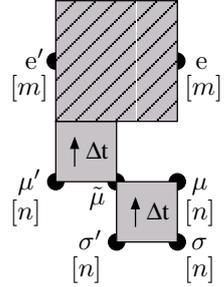
Für den nächsten Zeitschritt wird das Gitter vergrößert, indem man an das Ende von S und an den Anfang von E und $E[\mathcal{A}]$ jeweils ein lokales τ anfügt. Wiederum unterscheidet sich die Geometrie bei gerader und ungerader Trotterzahl.

M ungerade:

Der Systemblock wird wie folgt vergrößert:

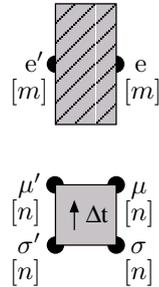
$$(S_{\text{even}})_{s\mu\sigma}^{s'\mu'\sigma'} = \sum_{\tilde{\mu}} (\tau)_{\tilde{\mu}\sigma}^{\mu'\sigma'} \cdot (S_{\text{odd}})_{s\tilde{\mu}}^{s'\tilde{\mu}} = \begin{array}{c} \begin{array}{|c|} \hline \sigma' \\ \hline \end{array} \\ \begin{array}{|c|} \hline \uparrow \Delta t \\ \hline \end{array} \\ \begin{array}{|c|} \hline \sigma \\ \hline \end{array} \\ \begin{array}{|c|} \hline \mu' \\ \hline \end{array} \\ \begin{array}{|c|} \hline \tilde{\mu} \\ \hline \end{array} \\ \begin{array}{|c|} \hline \uparrow \Delta t \\ \hline \end{array} \\ \begin{array}{|c|} \hline \mu \\ \hline \end{array} \\ \begin{array}{|c|} \hline s' \\ \hline \end{array} \\ \begin{array}{|c|} \hline s \\ \hline \end{array} \end{array} \quad (5.19)$$

Entsprechend werden auch E bzw. $E[\mathcal{A}]$ vergrößert:

$$(E_{\text{even}})_{\sigma \mu e}^{\sigma' \mu' e'} = \sum_{\tilde{\mu}} (E_{\text{odd}})_{\tilde{\mu} e}^{\mu' e'} \cdot (\tau)_{\sigma \mu}^{\sigma' \tilde{\mu}} =$$


$$(5.20)$$

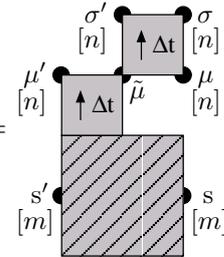
Damit haben S und E jeweils $(m \cdot n)^2 \cdot n^2$ Komponenten. Im ungeraden Trotterschritt bleibt CS unverändert, nur CE wird vergrößert durch ein Tensorprodukt mit τ :

$$(CE)_{\sigma \mu e}^{\sigma' \mu' e'} = (\tau)_{\sigma \mu}^{\sigma' \mu'} \cdot (CE)_e^{e'} =$$


$$(5.21)$$

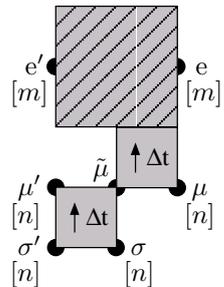
M gerade:

In einem geraden Trotterschritt ist die Geometrie von S und E genau umgekehrt:

$$(S_{\text{odd}})_{s \mu \sigma}^{s' \mu' \sigma'} = \sum_{\tilde{\mu}} (S_{\text{even}})_{s \tilde{\mu}}^{s' \mu'} \cdot (\tau)_{\mu \sigma}^{\tilde{\mu} \sigma'} =$$


$$(5.22)$$

und

$$(E_{\text{odd}})_{\sigma \mu e}^{\sigma' \mu' e'} = \sum_{\tilde{\mu}} (\tau)_{\sigma \tilde{\mu}}^{\sigma' \mu'} \cdot (E_{\text{even}})_{\tilde{\mu} e}^{\mu' e'} =$$


$$(5.23)$$

Nun bleibt CE unverändert, während CS durch ein Tensorprodukt mit τ vergrößert wird:

$$(CS)_{s\mu\sigma}^{s'\mu'\sigma'} = (CS)_s^{s'} \cdot (\tau)_{\mu\sigma}^{\mu'\sigma'} = \begin{array}{c} \begin{array}{ccc} \sigma' & \uparrow \Delta t & \sigma \\ [n] & & [n] \\ \mu' & & \mu \\ [n] & & [n] \end{array} \\ \cdot \\ \begin{array}{ccc} & & \\ & & \\ s' & & s \\ [m] & & [m] \end{array} \end{array} . \quad (5.24)$$

5.1.4 Renormierung

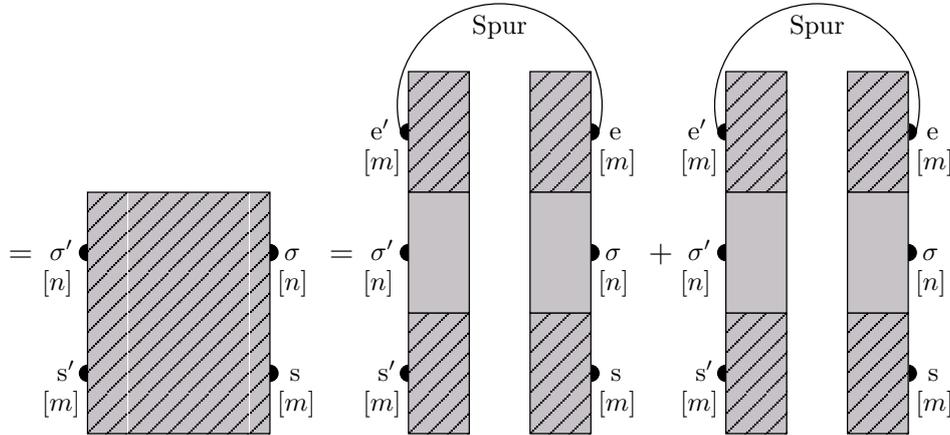
System- und Umgebungsblock sind nun $(m \cdot n)$ -dimensional. Falls

$$\dim \mathcal{H}_S = \dim \mathcal{H}_E = m \cdot n > m_{\max}, \quad (5.25)$$

dann ist der Zustandsraum zu groß, also muß man renormieren. Dazu bildet man zunächst die reduzierte Dichtematrix für den Systemblock,

$$\rho_{\text{sys}} = \frac{1}{2} \text{Tr}_{\text{env}} (|\psi_R\rangle\langle\psi_R| + |\psi_L\rangle\langle\psi_L|) : \quad (5.26)$$

$$(\rho_{\text{sys}})_{s\sigma}^{s'\sigma'} = \frac{1}{2} \left(\sum_e (|\psi_R\rangle)^{s'\sigma'e} (\langle\psi_R|)_{s\sigma e} + \sum_e (|\psi_L\rangle)^{s'\sigma'e} (\langle\psi_L|)_{s\sigma e} \right) \quad (5.27)$$



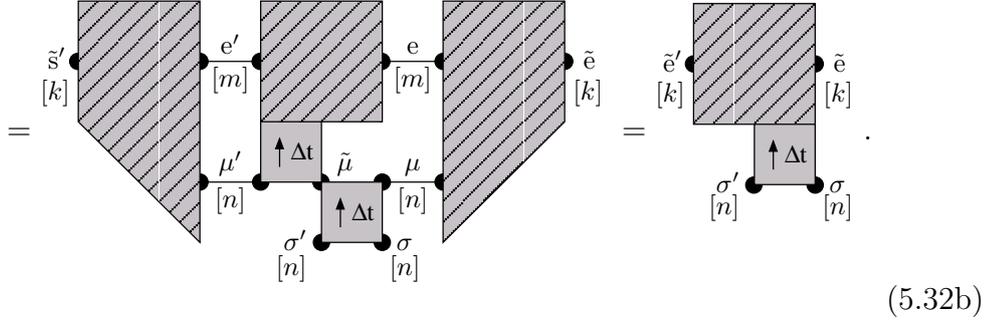
Analog bildet man

$$\rho_{\text{env}} = \frac{1}{2} \text{Tr}_{\text{sys}} (|\psi_R\rangle\langle\psi_R| + |\psi_L\rangle\langle\psi_L|) : \quad (5.28)$$

$$(\rho_{\text{env}})_{\sigma e}^{\sigma' e'} = \frac{1}{2} \left(\sum_s (|\psi_R\rangle)^{s\sigma'e'} (\langle\psi_R|)_{s\sigma e} + \sum_s (|\psi_L\rangle)^{s\sigma'e'} (\langle\psi_L|)_{s\sigma e} \right) \quad (5.29)$$

und

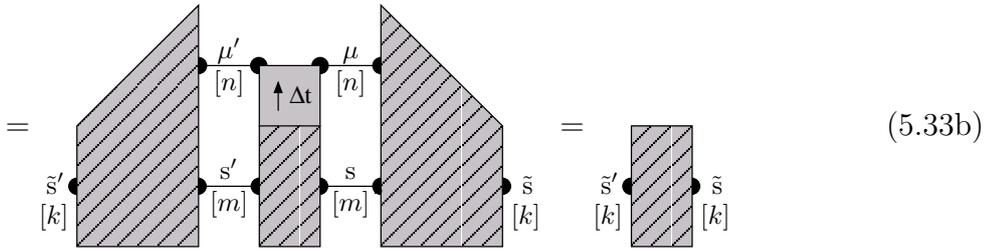
$$(\tilde{E}_{\text{even}})_{\sigma \tilde{e}}^{\sigma' \tilde{e}'} = \sum_{\mu' e' \mu e} (\mathcal{O}_E^T)_{\mu' e'}^{\tilde{e}'} \cdot (E_{\text{even}})_{\sigma \mu e}^{\sigma' \mu' e'} \cdot (\mathcal{O}_E)_{\tilde{e}}^{\mu e} \quad (5.32a)$$



(5.32b)

S und E haben also $(k \cdot n)^2$ Komponenten; k ist das m im nächsten Trotterschritt. Außerdem muß man CS und CE renormieren, damit sie in derselben Basis wie S und E sind:

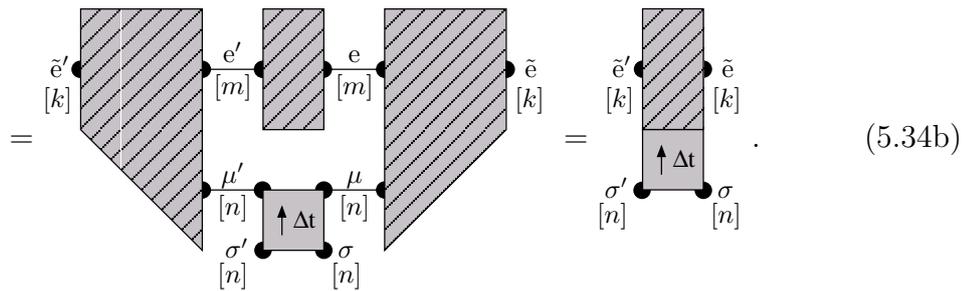
$$(\tilde{CS}_{\text{even}})_{\tilde{s}}^{\tilde{s}'} = \sum_{s' \mu' s \mu} (\mathcal{O}_S^T)_{s' \mu'}^{\tilde{s}'} \cdot (CS_{\text{even}})_{s \mu}^{s' \mu'} \cdot (\mathcal{O}_S)_{\tilde{s}}^{s \mu} \quad (5.33a)$$



(5.33b)

und

$$(\tilde{CE}_{\text{even}})_{\sigma \tilde{e}}^{\sigma' \tilde{e}'} = \sum_{\mu' e' \mu e} (\mathcal{O}_E^T)_{\mu' e'}^{\tilde{e}'} \cdot (CE_{\text{even}})_{\sigma \mu e}^{\sigma' \mu' e'} \cdot (\mathcal{O}_E)_{\tilde{e}}^{\mu e} \quad (5.34a)$$



(5.34b)

M gerade:

Die Projektion bei gerader Trotterzahl geht analog.

Nun fährt man mit dem 2. Schritt fort und berechnet die Eigenvektoren der Transfermatrix und die Erwartungswerte.

5.2 TMRG-Programm

5.2.1 Die Programmiersprache Python

Python ist eine interpretierte Sprache, die sehr bequeme Matrixoperationen (ähnlich Matlab[®]) bereitstellt und leicht durch bestehende C, C++ oder Fortran-Bibliotheken erweiterbar ist.

Das hat folgende Vorteile:

- Der TMRG-Algorithmus, der im wesentlichen lineare Algebra ist, läßt sich sehr knapp und intuitiv schreiben.
- Die Matrixdimensionen werden zur Laufzeit überprüft, damit bekommt man bei Programmierfehlern gleich eine spezifische Fehlermeldung und während der gesamten Entwicklungsphase keinen einzigen *segmentation fault*.
- Durch die Einbindung der Lapack-Bibliothek stehen schnelle Routinen für die Matrix-Diagonalisierung direkt in Python zur Verfügung.
- Ergibt das *profiling*, daß eine Python-Routine zu langsam ist, dann kann man sie in C/C++ oder Fortran implementieren und automatisch von Python aus darauf zugreifen.

In unserem Fall wird jedoch typischerweise mehr als 80% der Laufzeit in der Matrixmultiplikation (intern C) und in der Matrix-Diagonalisierung (Lapack) verbraucht, so daß eine Beschleunigung des restlichen Codes einen minimalen Geschwindigkeitsvorteil brächte.

- Da Python interpretiert wird, kann man direkt in den Programmablauf eingreifen, z.B. um sich das Spektrum der reduzierten Dichtematrix ausgeben zu lassen. Änderungen des Algorithmus lassen sich in wenigen Sekunden (ohne Neukompilierung) testen.
- Flexibilität: da der Algorithmus in abstrakter linearer Algebra beschrieben ist, lassen sich bequem die Datentypen ändern: soll z.B. die Fließkomma-Genauigkeit erhöht werden, so braucht man in der Initialisierung von τ bloß den Float-Datentyp aus der GNU Multiprecision Library (GMP) zu verwenden, und der gesamte Algorithmus wird ohne eine Änderung im Quelltext in beliebiger höherer Genauigkeit gerechnet (bloß die verwendete Matrix-Diagonalisierung aus Lapack muß man ersetzen, da Lapack diesen Datentyp nicht unterstützt). Das ist nicht immer numerisch effizient, hat aber den großen Vorteil, daß man ohne wesentliche Programmänderungen die Auswirkung höherer numerischer Genauigkeit untersuchen kann.

5.2.2 Matrizenrechnung in Python

In Zeile 17 des Python-Programms wird das `Numeric`-Modul geladen, es stellt die Matrixoperationen bereit. Arrays beliebiger Stufe, d.h. mit beliebig vielen Indizes, werden in Python in der Regel flach (wie in Fortran und C) gespeichert. Das erleichtert die Einbindung von Fortran-Bibliotheken wie Lapack.

In der Routine `setup_tau()` wird die lokale Transfermatrix initialisiert. Exemplarisch wollen wir hier den Paar-Annihilations-Prozeß simulieren (siehe Abschnitt 2.3), also deklarieren wir τ zunächst als Array mit vier Indizes (l_1, l_2, r_1, r_2) , die jeweils $n = 2$ Werte annehmen (Zeile 40). Dann initialisieren wir τ mit den Werten aus (2.25), wobei der Index 0 für einen leeren (\emptyset) und 1 für einen besetzten Gitterplatz (A) steht. Diese Routine ist der einzige Teil des Programms, der vom untersuchten physikalischen Modell abhängt.

In der Routine `init()` werden die System- und Umgebungsblöcke für den ersten Trotterschritt initialisiert. In Zeile (58) wird

$$(S_{\text{init}})_{s\sigma}^{s'\sigma'} = \sum_{l_1 r_1} w(l_1) w(r_1) \cdot (\tau)_{r_1\sigma}^{l_1\sigma'} \quad (5.35)$$

initialisiert: mit `transpose()` geht das Indexbild von τ in (l_2, r_2, l_1, r_1) über, und `dot()` kontrahiert den letzten Index des ersten Arguments mit dem ersten Index des zweiten Arguments (Wahrscheinlichkeit $w(i) = \text{iweight}[i]$). Die Matrixelemente des Umgebungsblocks sind anfänglich alle

$$(E_{\text{init}})_{\sigma e}^{\sigma' e'} = \sum_{l_2 r_2} (\tau)_{\sigma r_2}^{\sigma' l_2} = 1 \quad (n \times n) \quad (5.36)$$

(Zeile 60). Der Umgebungsblock $E[\mathcal{A}]$ mit den Randbedingungen eines lokalen Operators \mathcal{A} wird gebildet, indem über einen Endindex summiert wird und der andere mit dem Erwartungswert des Operators $\mathcal{A}(i) = \text{fweight}[i]$ kontrahiert wird (Zeile 63).

Die grundlegende Matrixmultiplikation in Python ist `innerproduct()`: sie kontrahiert den letzten Index des ersten Arguments mit dem *letzten* Index des zweiten Arguments und gibt das resultierende Array mit den verbleibenden Indizes beider Argumente zurück, z.B.

```
a = ones((k,l,m))
b = ones((r,s,t,m))
c = innerproduct(a,b)
```

liefert ein fünfstufiges Array mit Dimensionen `c[k,l,r,s,t]`. `innerproduct()` ist vollständig in C implementiert und sehr schnell, deshalb basieren die meisten Matrixoperationen im Programm auf dieser Routine. Dazu müssen zuerst die Indizes, die kontrahiert

werden sollen, mit `reshape()` und `transpose()` an die letzte Stelle gebracht werden. Die Funktion `reshape()` faßt die Indizes eines Arrays anders zusammen: das zweite Argument ist das neue Indexbild, über das der Zugriff auf das flach gespeicherte Array erfolgen soll. Eine -1 im Indexbild bedeutet, daß dieser Index die Dimension übernehmen soll, die von den anderen Indizes übriggelassen wird. Ein Beispiel ist `reshape(psi, (Sdim, -1))` in Zeile 83: `psi[s σ e]` ist $m \cdot n \cdot m$ -dimensional und `Sdim` ist die gemeinsame Dimension von `s` und `σ` , $m \cdot n$. Also steht der verbleibende Index -1 für $(m \cdot n \cdot m)/(m \cdot n) = m$ Dimensionen. Damit haben wir aus dem Vektor `psi` ein zweistufiges Array (Matrix) gemacht. Diese beiden Indizes werden nun mit `transpose()` vertauscht: das zweite Argument `(1, 0)` gibt an, in welcher neuen Reihenfolge die alten Indizes auftreten sollen.

Ein typisches Beispiel ist die Multiplikation der Transfermatrix mit einem Vektor (`multSE()`, Zeile 79):

$$(T_{\text{odd}} \cdot |\psi_R\rangle)^{s'\sigma'e} = \sum_{\tilde{\sigma}e} (E_{\text{odd}})_{\tilde{\sigma}e}^{\sigma'e'} \cdot \left(\sum_{s\sigma} (S_{\text{odd}})_{s\sigma}^{s'\tilde{\sigma}} \cdot (|\psi_R\rangle)^{s\sigma e} \right) \quad (5.37)$$

- zuerst werden die Indizes `s` und `σ` kontrahiert, bringe also `s` und `σ` von `|\psi_R\rangle` mit `reshape()` und `transpose()` als einen gemeinsamen Index an die letzte Stelle: `psi[e, s σ] := psiR[s σ e]`; der `s σ` -Index von `S[s' σ' , s σ]` ist bereits der letzte Index (Zeile 83)
- bilde das Produkt `psi[s' $\tilde{\sigma}$, e] := innerproduct(S, psi)` (Zeile 84)
- fasse die Indizes von `psi` mit `reshape()` anders zusammen, `psi[s', $\tilde{\sigma}$ e]`, und bilde das Produkt mit `E[s' σ' , $\tilde{\sigma}$ e]`: `psi[s', σ' e'] := innerproduct(psi, E)` (Zeile 85)
- die Funktion `ravel()` gibt das resultierende Array flach (als Vektor `psi[s' σ' e']`) zurück (Zeile 92).

Ähnlich verfährt man in der Routine `renorm_basis()` (Zeile 139) bei der Bildung der reduzierten Dichtematrix des Systems:

- wir wollen den Umgebungsindex `e` kontrahieren, also isolieren wir ihn zunächst mit `reshape()` als letzten Index: `l[s σ , e] = psiL[s σ e]` (Zeile 141), ebenso `r[s σ , e] = psiR[s σ e]` (Zeile 142)
- die Kontraktion der letzten Indizes (`innerproduct()`) liefert eine Matrix der Form `[s σ , s σ]` zurück, wobei die vorderen Indizes zur Unterscheidung gestrichen sind: `rhosys[s' σ' , s σ]` (Zeile 143).

Diese wird durch ihre Spur geteilt (normiert) und diagonalisiert (Zeile 144). Ebenso bilden wir die reduzierte Dichtematrix der Umgebung, bloß daß man nun den Systemindex `s` isoliert und nach hinten bringt (Zeilen 149–151).

In Zeile 146 liefert `greater(Sval,epsilon)` ein Array von der Form von `Sval` zurück, dessen Elemente i Eins sind, falls `Sval[i] > epsilon`, und Null sonst. Also zählt

$$\text{sum}(\text{greater}(\text{Sval},\text{epsilon})) \quad (5.38)$$

die Anzahl der nichtsingulären Eigenwerte. In keinem Fall nimmt man jedoch mehr als m_{\max} Eigenwerte (Zeilen 146 und 154), denn der Zustandsraum soll ja reduziert werden. Der Befehl `sum(Sval[keep:])` ist die Summe der Eigenwerte *oberhalb* von `keep`, die trunziert werden, also der Approximationsfehler Δ .

5.2.3 Programmcode

```
1  #
2  # stochastic TMRG program
3  #   (optimized for systems invariant under spatial reflection)
4  #
5  # (C) 2001 by Tilman Enss (Tilman.Enss@physik.uni-muenchen.de)
6  #
7
8  # Set to 1 for debugging output (density matrix spectra).
9  debug = 0
10
11 # Maximum error for the convergence of the power method
12 # ('power()') and for the selection of nonsingular density matrix
13 # eigenvalues; should be approximately machine accuracy.
14 epsilon = 1e-14
15
16 # Load Numeric module for matrix functions.
17 from Numeric import *
18
19 # Much time is spent in the diagonalization of the reduced density
20 # matrix; try to use a fast routine (e.g. Lapack DSYEVR).
21 # symeigenvec() returns _sorted_ eigenvalues/-vectors.
22 try:
23     from linalg import dsyevr
24     def symeigenvec(A): return dsyevr(A,0)
25 except:
26     from LinearAlgebra import eigenvectors
27     def symeigenvec(A):
28         eval, evec = eigenvectors(A)
29         # real eigenvalues, since A is symmetric
30         try: eval = eval.real; evec = evec.real
```

```
31         except: pass
32         return eval, evec
33
34 # Setup the local transfer matrix tau for a simple reaction-
35 # diffusion model: monomers aggregate with rate 'twoalpha' and
36 # diffuse with rate 'D'; the time step is 'dt' units.
37
38 def setup_tau(dt=0.1, alpha=0.5, D=0.5):
39     n = 2          # number of states per site: 0=empty, 1=A
40     tau = zeros((n,n,n,n), Float) # indices tau[l1,l2,r1,r2]
41     expalpha = exp(-2*alpha*dt)
42     expD = exp(-2*D*dt)
43     tau[0,0,0,0] = 1.0
44     tau[1,0,1,0] = 1.0-expalpha
45     tau[1,1,1,1] = expalpha
46     tau[0,0,1,1] = tau[1,1,0,0] = 0.5*(1.0+expD)
47     tau[0,1,1,0] = tau[1,0,0,1] = 0.5*(1.0-expD)
48     return n, tau
49
50 # Initialize the system (S) and environment (E) blocks for the
51 # first Trotter step (M=1). 'iweight' is the initial weight
52 # (bias) of each state; 'EA' is the environment block with
53 # boundary conditions corresponding to the local operator given by
54 # the expectation values for each final local state, 'fweight'.
55
56 def init(n, tau, iweight, fweight):
57     # S[l2,r2]=sum(l1,r1) iweight[l1]*iweight[r1]*tau[l2,r2,l1,r1]
58     S = CS = dot(dot(transpose(tau, (1,3,0,2)), iweight), iweight)
59     # E[l1,r1] = 1
60     E = ones((n,n), Float)
61     CE = ones((1,1), Float)
62     # EA[l1,r1] = sum(l2,r2) tau[l1,l2,r1,r2]*fweight[r2]
63     EA = sum(dot(tau, fweight), 1)
64     return S, E, EA, CS, CE
65
66 # Normalize the vector v: first compute its norm, if it is nonzero
67 # then divide by it and return the normalized vector and its norm.
68
69 def normalize(v):
70     norm = sqrt(innerproduct(v,v))
71     if norm > epsilon: v = v/norm
72     return v, norm
```

```

73
74 # Multiply a vector 'psi' with the transfer matrix, stored sepa-
75 # rately as system ('S') and environment ('E'). The order of the
76 # multiplications with 'S' and 'E' depends on the exact geometry
77 # (whether the Trotter number M is odd or even).
78
79 def multSE(S,E,odd,psi):
80     Sdim,Edim = S.shape[1],E.shape[1] # dimensions to contract
81     if odd:
82         # if M is odd, first multiply with S, then with E
83         psi = transpose(reshape(psi,(Sdim,-1)),(1,0))
84         psi = innerproduct(S,psi)
85         psi = innerproduct(reshape(psi,(-1,Edim)),E)
86     else:
87         # if M is even, first multiply with E, then with S
88         psi = innerproduct(reshape(psi,(-1,Edim)),E)
89         psi = transpose(reshape(psi,(Sdim,-1)),(1,0))
90         psi = innerproduct(S,psi)
91     # return flat vector
92     return ravel(psi)
93
94 # The power method for finding the right eigenvector corresponding
95 # to the largest eigenvalue of a dense matrix: begin with some
96 # start vector 'psi' and continue to multiply the transfer matrix
97 # onto it until the eigenvector converges within an error of
98 # 'epsilon'.
99
100 MAXITER = 200
101 def power(S,E,odd,psi):
102     for i in range(MAXITER): # iterate at most MAXITER times
103         psinew,lamb = normalize(multSE(S,E,odd,psi))
104         err = max(abs((psinew-psi)))/max(abs(psinew))
105         if err < epsilon: return lamb,psinew
106         psi = psinew
107     print "power: largest eigenvector DID NOT converge!"
108     return 0,psinew
109
110 # Add local tau to S, E, CS, or CE. 'first' comes first in the
111 # time evolution (system block) and is a square matrix of size
112 # [a*b,a*b]. 'second' comes afterwards (environment block) and is
113 # a square matrix of size [b*c,b*c]. The order of the multipli-
114 # cation is determined by the odd or even geometry. The 'b' index

```

```

115 # is contracted. Finally, the grown matrix is returned as a
116 # square matrix.
117
118 def grow(first,second,odd,a,b,c):
119     if odd:
120         first = transpose(reshape(first,(a,b,a*b)),(0,2,1))
121         second = transpose(reshape(second,(b*c,b,c)),(0,2,1))
122     else:
123         first = reshape(first,(a*b,a,b))
124         second = transpose(reshape(second,(b,c,b*c)),(1,2,0))
125     # combine both parts by contracting the 'b' index
126     combined = innerproduct(first,second)
127     # return square matrix
128     return reshape(transpose(combined,(0,2,1,3)),(a*b*c,a*b*c))
129
130 # Find a truncated basis for renormalization by constructing the
131 # symmetric reduced density matrix for the system/environment,
132 # tracing over the resp. other indices and taking the eigenvectors
133 # corresponding to the largest eigenvalues. A maximum of 'mmax'
134 # eigenvectors are retained (such that the renormalization reduces
135 # the state space sufficiently), but only those corresponding to
136 # _nonsingular_ eigenvalues: 'sum(greater(Sval,epsilon))' is the
137 # number of eigenvalues greater than 'epsilon'.
138
139 def renorm_basis(n,m,mmax,psiL,psiR):
140     # find reduced system basis
141     l = reshape(psiL,(m*n,m))
142     r = reshape(psiR,(m*n,m))
143     rhosys = innerproduct(l,l) + innerproduct(r,r)
144     Sval,Svec = symeigenvec(rhosys/trace(rhosys))
145     if debug: print "Sys. eigenvalues:\n",Sval # print eigenvals
146     keepsys = min(mmax,sum(greater(Sval,epsilon)))
147
148     # find reduced environment basis
149     l = transpose(reshape(psiL,(m,n*m)),(1,0))
150     r = transpose(reshape(psiR,(m,n*m)),(1,0))
151     rhoenv = innerproduct(l,l) + innerproduct(r,r)
152     Eval,Evec = symeigenvec(rhoenv/trace(rhoenv))
153     if debug: print "Env. eigenvalues:\n",Eval # print eigenvals
154     keepenv = min(mmax,sum(greater(Eval,epsilon)))
155     keep = max(keepsys,keepenv)
156

```

```

157     # compute approximation error and return truncated basis
158     if debug: print "keep m=%i, approx.error %.2e(S), %.2e(E)" % \
159         (keep,sum(Sval[keep:]),sum(Eval[keep:]))
160     return Svec[:keep],Evec[:keep],keep
161
162 # Project a square matrix 'M' into a new basis, given by the array
163 # of basis vectors 'basis'. 'l' ('r') dimensions on the left
164 # (right) are left unchanged.
165
166 def project(M,basis,l,r):
167     newdim,olddim = basis.shape
168     # project right side of matrix 'M'
169     M = transpose(reshape(M,(-1,olddim,r)),(0,2,1))
170     M = transpose(innerproduct(M,basis),(0,2,1))
171     # project left side of matrix 'M'
172     M = transpose(reshape(M,(l,olddim,-1)),(0,2,1))
173     M = transpose(innerproduct(M,basis),(0,2,1))
174     # return projected square matrix
175     return reshape(M,(l*newdim*r,l*newdim*r))
176
177 # The main Trotter loop: simulate 'Mmax' Trotter steps, with
178 # S(E)-state space dimension of 'mmax' states maximum. The local
179 # operator to be measured at the end of the time evolution is
180 # given by 'Aweight', its expectation values for the local states.
181 # 'dt' is the time step, the initial weight 'iweight' is unbiased
182 # by default.
183
184 def main(Mmax=200,mmax=16,dt=0.1,Aweight=[0.0,1.0],iweight=None):
185     # setup the local transfer matrix tau
186     n,tau = setup_tau(dt)
187
188     # if not defined, unbiased initial weight
189     if iweight is None: iweight = [1.0/n] * n
190
191     # initialize system and environment blocks
192     S,E,EA,CS,CE = init(n,tau,iweight,Aweight)
193     m = 1
194
195     # write out initial density
196     print "M = %3i:  t = %8.4f, Erwartungswert = %.8f" % \
197         (0,0,dot(iweight,Aweight))
198     outfile = open("density","w")

```

```
199     outfile.write("%8.4f %.8f\n" % (0,dot(iweight,Aweight)))
200
201     # loop over Trotter number
202     for M in range(1,Mmax+1):
203         # geometry of S and E alternates with M
204         odd = M % 2
205
206         # compute largest right eigenvalue/-vector lmax,psiR of
207         # the transfer matrix and multiply psiR by one column of
208         # tau's to compute left eigenvector psiL
209         psiR = ones((m*n*m,)) # start vector for power method
210         lmax,psiR = power(S,E,odd,psiR)
211         psiL = multSE(CS,CE,odd,psiR)
212
213         # max. eigenvalue should be approx. 1
214         if abs(lmax-1.0) > 0.1:
215             print "*** numerical instability ***"
216             return
217
218         # compute and print expectation value of local operator
219         id = lmax*innerproduct(psiL,psiR) # expect. of identity op
220         density = innerproduct(psiL,multSE(S,EA,odd,psiR))/id
221         print "M = %3i:  t = %8.4f, Erwartungswert = %.8f" % \
222             (M,M*dt,density)
223         outfile.write("%8.4f %.8f\n" % (M*dt,density))
224         outfile.flush()
225
226         # grow S, E, EA, CS, and CE by one tau each
227         S = grow(S,tau,odd,m,n,n)
228         E = grow(tau,E,odd,n,n,m)
229         EA = grow(tau,EA,odd,n,n,m)
230         if odd: CE = grow(tau,CE,odd,n*n,1,m)
231         else:   CS = grow(CS,tau,odd,m,1,n*n)
232
233         # if state space is too large, renormalize:
234         if m*n > mmax:
235             # compute optimal reduced basis
236             psiL,norm = normalize(psiL)
237             Sbasis,Ebasis,m = renorm_basis(n,m,mmax,psiL,psiR)
238
239         # project down to a smaller state space
240         S = project(S,Sbasis,1,n)
```

```
241         E = project(E,Ebasis,n,1)
242         EA = project(EA,Ebasis,n,1)
243         CS = project(CS,Sbasis,1,n**(1-odd))
244         CE = project(CE,Ebasis,n**odd,1)
245     else:
246         m = m*n # grow state space dimension
247     outfile.close()
```

tmrilight.py: TMRG-Programm in Python¹

¹Vorübergehend verfügbar unter der folgenden URL:
<http://www.theorie.physik.uni-muenchen.de/~tilman/tmrilight.py>

6 Zusammenfassung und Ausblick

Die Transfermatrix-Renormierungsgruppe (TMRG) für Quantenspinketten ermöglicht die Untersuchung thermodynamischer Größen in eindimensionalen Quantensystemen bei sehr tiefen Temperaturen. In der vorliegenden Arbeit haben wir diskutiert, wie sich diese Methode auch eignet, um das Kurzzeitverhalten eindimensionaler stochastischer Systeme (Reaktions-Diffusions-Prozesse) fern vom statistischen Gleichgewicht zu untersuchen.

Die stochastische TMRG hat gegenüber anderen Methoden eine Reihe klarer Vorteile:

- Der thermodynamische Limes ist exakt, man bekommt so wesentlich verlässlichere Ergebnisse als z.B. mit der stochastischen DMRG, die von einer endlichen Gittergröße auf ein unendliches Gitter extrapolieren und den *target*-Zustand aufwendig auswählen muß.
- Da die TMRG im Pfadintegral *sämtliche* Konfigurationen in allen diskreten Zeitschritten berücksichtigt, ist sie vor allem in niederdimensionalen Systemen mit starken Fluktuationen der Monte-Carlo-Methode überlegen, die nur eine Vielzahl exemplarischer Konfigurationen simuliert.

Die Wahl der Dichtematrix ist für die TMRG von zentraler Bedeutung. Bei der Anwendung auf nichthermitesche (nicht zeitumkehrinvariante) Probleme ist sie jedoch vom Algorithmus nicht eindeutig vorgeschrieben. In der Quanten-TMRG verwendet man die asymmetrische Dichtematrix mit großem Erfolg. Für die kürzlich eingeführte stochastische TMRG konnten wir dagegen erstmals beweisen, daß die offenen Randbedingungen in Verbindung mit der Wahrscheinlichkeitserhaltung die asymmetrische Dichtematrix unbrauchbar machen. Das grundsätzliche Problem ist hier, daß bei offenen Randbedingungen die physikalische Wechselwirkung weggemittelt wird, was bei Quantensystemen mit periodischen Randbedingungen nicht der Fall ist. Dies ist ein zentrales Ergebnis der vorliegenden Arbeit.

Wir haben anschließend eine neue Betrachtungsweise der TMRG für stochastische Systeme aufgezeigt, indem wir die dominanten linken und rechten Eigenvektoren der vollen Transfermatrix erstmals explizit konstruiert haben. Das hat es uns ermöglicht, die physikalische Bedeutung unterschiedlicher Dichtematrizen und Varianten des Algorithmus

sehr viel fundierter als bisher zu untersuchen. Eine kurze Darstellung dieser neuen Ergebnisse findet sich in [ES01]. Für die numerische Simulation ergibt sich damit folgende Alternative:

- Die asymmetrische Dichtematrix — wie in der Quanten-TMRG — hat angewandt auf stochastische Systeme den Nachteil, daß man bei der DMRG-Projektion alle räumlichen Korrelationen verliert, also das *mean-field*-Ergebnis berechnet. Andererseits hat sie den numerischen Vorteil, daß die Normen von ψ_L und ψ_R beschränkt bleiben.
- Bei der symmetrischen Dichtematrix macht das exponentielle Wachstum der Matrixelemente die Simulation numerisch sehr aufwendig. Die Anzahl der möglichen Trotterschritte ist durch die Fließkomma-Genauigkeit begrenzt und skaliert linear mit der Zahl der Mantissenbits. Dies ermöglicht uns theoretisch die Simulation beliebig vieler Rechenschritte, stößt aber bisher noch schnell an die Grenzen der uns zur Verfügung stehenden Rechenleistung, so ließ sich z.B. das Kurzzeitverhalten interessanter Nichtgleichgewichts-Prozesse (SELKOV-Prozeß [GS94], *branching-fusing*-Prozeß) bisher nur ansatzweise untersuchen. Der entscheidende Vorteil ist jedoch, ein stochastisches System über mehr als 100 Zeitschritte physikalisch sehr genau mit *allen* räumlichen Korrelationen simulieren zu können.

Als Anwendung haben wir folgende Reaktions-Diffusions-Prozesse betrachtet: die durch Diffusion begrenzte Paar-Annihilation und den *branching-fusing*-Prozeß. Da die Lösung des Paar-Annihilations-Prozesses exakt bekannt ist, konnten wir das Konvergenzverhalten und die Genauigkeit der Simulation abschätzen. Die Abweichung ist sehr gering und liegt sogar deutlich unter dem Trotterfehler, der durch die diskreten Zeitschritte zustande kommt.

Schließlich haben wir eine komplette und schnelle Implementierung des TMRG-Algorithmus in nur 120 Zeilen Programmcode und mit ausführlichen Kommentaren präsentiert, die leicht erweiterbar ist und als Grundlage für künftige Projekte dienen mag.

Eine erfolgversprechende Erweiterung könnte sein, anstelle eines stochastischen Systems in kontinuierlicher Zeit ein anderes in diskreter Zeit zu suchen, das derselben Universalitätsklasse angehört [Kem00]. Typischerweise haben diskrete Systeme größere Übergangswahrscheinlichkeiten als kontinuierliche Systeme mit feiner TROTTER-SUZUKI-Zerlegung, also ist ihre lokale Transfermatrix τ weiter entfernt von der Einheitsmatrix. Damit ist das exponentielle Wachstum der Matrixelemente langsamer, man kann z.B. bei den diskreten DOMANI-KINZEL-Automaten selbst bei doppelter Rechengenauigkeit schon bis zu etwa 170 Trotterschritten simulieren.

Literaturverzeichnis

- [Abr72] M. ABRAMOWITZ (Hrsg.), *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, Dover, New York, 1972.
- [Bax82] R. J. BAXTER, *Exactly solved models in statistical mechanics*, Academic Press, London, 1982.
- [BXG96] R. J. BURSILL, T. XIANG, und G. A. GEHRING, *The density matrix renormalization group for a quantum spin chain at non-zero temperature*, J. Phys.: Condens. Matter **8**, L583–90 (1996), `cond-mat/9609001`.
- [CDvL01] E. CARLON, A. DRZEWIŃSKI, und J. M. J. VAN LEEUWEN, *Crossover behavior for long reptating polymers*, Phys. Rev. E **64**, 010801 (2001), `cond-mat/0010177`.
- [CHS99] E. CARLON, M. HENKEL, und U. SCHOLLWÖCK, *Density matrix renormalization group and reaction-diffusion processes*, Eur. Phys. J. B **12**, 99 (1999), `cond-mat/9902041`.
- [CHS01] ———, *Critical properties of the reaction-diffusion model $2A$ to $3A$, $2A$ to 0* , Phys. Rev. E **63**, 036101 (2001), `cond-mat/9912347`.
- [DFL85] A. DE MASI, P. A. FERRARI, und J. L. LEBOWITZ, *Rigorous derivation of reaction-diffusion equations with fluctuations*, Phys. Rev. Lett. **55**, 1947–9 (1985).
- [ES01] T. ENSS und U. SCHOLLWÖCK, *On the choice of the density matrix in the stochastic TMRG*, J. Phys. A: Math. Gen. **34**, 7769–82 (2001), `cond-mat/0106194`.
- [Fey72] R. P. FEYNMAN, *Statistical mechanics. A set of lectures*, Benjamin, Reading, Mass., 1972.
- [GS94] P. GRAY und S. K. SCOTT, *Chemical oscillations and instabilities*, The international series of monographs on chemistry 21, Clarendon Press, Oxford, 1994.

- [Hie98] Y. HIEIDA, *Application of the density matrix renormalization group method to a non-equilibrium problem*, J. Phys. Soc. Jpn. **67**, 369–72 (1998), cond-mat/9711072.
- [HS01] M. HENKEL und U. SCHOLLWÖCK, *Universal finite-size scaling amplitudes in anisotropic scaling*, J. Phys. A: Math. Gen. **34**, 3333–50 (2001), cond-mat/0010061.
- [Kem00] A. KEMPER, *Dichtematrix-Renormierungsgruppe für zweidimensionale klassische Modelle der statistischen Mechanik*, Diplomarbeit, Institut für Theoretische Physik der Universität zu Köln, Köln, 2000.
- [KP98] M. KAULKE und I. PESCHEL, *A DMRG study of the q -symmetric Heisenberg chain*, Eur. J. Phys. B **5**, 727–34 (1998), cond-mat/9802175.
- [KSZ01] A. KEMPER, A. SCHADSCHNEIDER, und J. ZITTARTZ, *Transfer-matrix density-matrix renormalization group for stochastic models: the Domany-Kinzel cellular automaton*, J. Phys. A: Math. Gen. **34**, L279 (2001), cond-mat/0102395.
- [Lan50] C. LANCZOS, J. Res. Natl. Bur. Stand. **45**, 225 (1950).
- [MS98] K. MAISINGER und U. SCHOLLWÖCK, *Thermodynamics of frustrated quantum spin chains*, Phys. Rev. Lett. **81**, 445–8 (1998), cond-mat/9803122.
- [Nis95] T. NISHINO, *Density matrix renormalization group method for 2D classical models*, J. Phys. Soc. Jpn. **64**, 3598–601 (1995), cond-mat/9508111.
- [NS99] T. NISHINO und N. SHIBATA, *Efficiency of asymmetric targeting for finite- T DMRG*, J. Phys. Soc. Jpn. **68**, 3501–4 (1999), cond-mat/9905229.
- [NW99] R. M. NOACK und S. R. WHITE, *The density matrix renormalization group*, in PESCHEL *et al.* [PWKH99], pp. 27–66.
- [PWKH99] I. PESCHEL, X. WANG, M. KAULKE, und K. HALLBERG (Hrsg.), *Density matrix renormalization*, Lecture notes in physics 528, Berlin, Springer, 1999.
- [Rác85] Z. RÁCZ, *Diffusion-controlled annihilation in the presence of particle sources: Exact results in one dimension*, Phys. Rev. Lett. **55**, 1707–10 (1985).
- [Shi97] N. SHIBATA, *Thermodynamics of the anisotropic Heisenberg chain calculated by the density matrix renormalization group method*, J. Phys. Soc. Jpn. **66**, 2221–3 (1997), cond-mat/9706152.

- [Spo88] J. L. SPOUGE, *Exact solutions for a diffusion-reaction process in one dimension*, Phys. Rev. Lett. **60**, 871–4 (1988).
- [Suz85a] M. SUZUKI, *Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics*, J. Math. Phys. **26**, 601 (1985).
- [Suz85b] ———, *Transfer-matrix method and Monte Carlo simulation in quantum spin systems*, Phys. Rev. B **31**, 2957 (1985).
- [Tro59] H. F. TROTTER, Proc. Am. Math. Soc. **10**, 545 (1959).
- [vK97] N. G. VAN KAMPEN, *Stochastic processes in Physics and Chemistry*, 2. ed., North Holland, Amsterdam, 1997.
- [Whi92] S. R. WHITE, *Density matrix formulation for quantum renormalization groups*, Phys. Rev. Lett. **69**, 2863–6 (1992).
- [Whi93] ———, *Density-matrix algorithms for quantum renormalisation groups*, Phys. Rev. B **48**, 10345–56 (1993).
- [Wil75] K. G. WILSON, *The renormalization group: Critical phenomena and the Kondo problem*, Rev. Mod. Phys. **47**, 773–840 (1975).
- [WN92] S. R. WHITE und R. M. NOACK, *Real-space quantum renormalization groups*, Phys. Rev. Lett. **68**, 3487 (1992).
- [WX97] X. WANG und T. XIANG, *Transfer-matrix density-matrix renormalization-group theory for thermodynamics of one-dimensional quantum systems*, Phys. Rev. B **56**, 5061–4 (1997), cond-mat/9705301.

Danksagung

An dieser Stelle möchte ich allen danken, die zum Gelingen der Diplomarbeit beigetragen haben: zuvorderst PD Dr. Ulrich Schollwöck für die Auswahl des Themas, die ausgezeichnete Betreuung und die Zeit, die er sich immer für meine Fragen genommen hat. Weiterhin danke ich Prof. Dr. Malte Henkel, Prof. Dr. Herbert Spohn, Andreas Kemper und PD Dr. Andreas Schadschneider für fruchtbare Diskussionen.

Johannes Martin und Dominique Gobert danke ich für die Mühe des Korrekturlesens, Dr. Jens Schmalzing für unermüdliche Computerhilfe und der Arbeitsgruppe von Prof. Dr. Herbert Wagner für die anregende Atmosphäre, daß mir das Diplomarbeiten auch nach einem Jahr noch so viel Freude macht wie zu Beginn.

Erklärung

Ich habe die vorliegende Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt.

München, den 9. Januar 2002