

---

BAKKELAUREATSARBEIT  
Institut für Theoretische Physik

SIMULATION VON VIELTEILCHENSYSTEMEN  
MIT MATHEMATICA

Betreuer:  
Univ.-Prof. Dr.rer.nat. Enrico Arrigoni

CHRISTIAN GRUBER

Mat.Nr. 0430350

Sommersemester 2008

## Inhaltsverzeichnis

<b>1</b>	<b>Motivation</b>	<b>3</b>
<b>2</b>	<b>Fock-Raum</b>	<b>4</b>
2.1	Implementation . . . . .	5
<b>3</b>	<b>Hubbard-Modell</b>	<b>6</b>
3.1	Hamiltonian . . . . .	6
3.2	Eigenschaften . . . . .	8
3.2.1	spin-charge-separation . . . . .	8
3.2.2	reale Materialien . . . . .	8
3.3	Symmetrien . . . . .	9
3.3.1	Eichsymmetrie . . . . .	9
3.3.2	Spinsymmetrie . . . . .	9
3.4	Zeitentwicklung . . . . .	10
3.4.1	Dissipation . . . . .	10
<b>4</b>	<b>Simulation</b>	<b>11</b>
4.1	Systemgröße . . . . .	11
4.2	Simulationsergebnisse . . . . .	13
4.2.1	Spin-Charge-Separation . . . . .	13
4.2.2	Dissipation . . . . .	14
4.2.3	Korrelation . . . . .	17

## 1 Motivation

Zur Behandlung von quantenmechanischen Vielteilchenproblemen wird meist die 2. Quantisierung herangezogen. Hierbei erfolgt die Beschreibung durch Erzeugungs- und Vernichtungsoperatoren.

Die einfachen Transformationsregeln dieser Operatoren ermöglichen für sehr kleine Vielteilchensysteme die analytische Betrachtung mit symbolischer Programmiersoftware. Für diese Arbeit wird *Mathematica* verwendet.

Phänomene, wo die Korrelation zwischen Fermionen für das physikalische Verhalten wichtig sind, können nicht mit konventionellen Methoden, die auf einem effektiven Einteilchenproblem basieren (z.B. Local Density Approximation), behandelt werden. Bei solchen Vielteilchenproblemen liefern das Hubbard-Modell und seine Erweiterungen ein qualitatives Verständnis.

Für die Beschreibung von Festkörpern muss eine Vielzahl wechselwirkender Teilchen miteinbezogen werden. Das Hubbard-Modell enthält Terme, die zur Beschreibung wechselwirkender Fermionen benötigt werden: Einen kinetische Term, der die Bewegung der Elektronen wiedergibt (elektrische Leitfähigkeit), und einen Potentialterm, der deren Wechselwirkung enthält.

Trotz dieser Einfachheit kann das Hubbard-Modell nur für eine Dimension und periodischen Randbedingungen (Ring) mittels des Bethe-Ansatz exakt gelöst werden.

Das Hubbard-Modell und seine Erweiterungen werden in folgenden Bereichen eingesetzt:

- Hochtemperatur-Supraleitung
- Metall-Isolator-Übergang (Mott-Isolator)
- Bandmagnetismus (Fe, Ni, Co,...)
- Polymere
- Übergangsmetalle und deren Oxide
- Fullerene (z.B. C<sub>60</sub>)
- Seltene Erden - zB. CeAl<sub>3</sub> oder UPt<sub>3</sub>  
Schwere Fermionen Materialien (2-3 Mal größere Wärmekapazität bei niedrigen Temperaturen als üblich)
- Flüssiges Helium (<sup>3</sup>He ist Fermion)

## 2 Fock-Raum

Bei Vielteilchenproblemen erweist sich eine Darstellung mittels Erzeugungs- und Vernichtungsoperatoren als günstig, da die Symmetrien der Wellenfunktionen in den Vertauschungsrelationen enthalten sind [1].

Für Bosonen gelten folgende Kommutatorrelationen:

$$[a_i, a_j] = 0 \quad (1)$$

$$[a_i^\dagger, a_j^\dagger] = 0 \quad (2)$$

$$[a_i, a_j^\dagger] = \delta_{i,j} \quad (3)$$

Bei Fermionen wird aus dem Kommutator der Antikommutator. Zusätzlich wird jetzt auch der Spin  $\sigma$  mitberücksichtigt. Die Vertauschungsrelationen lauten:

$$\{c_{i\sigma}, c_{j\sigma'}\} = 0 \quad (4)$$

$$\{c_{i\sigma}^\dagger, c_{j\sigma'}^\dagger\} = 0 \quad (5)$$

$$\{c_{i\sigma}, c_{j\sigma'}^\dagger\} = \delta_{ij}\delta_{\sigma\sigma'} \quad (6)$$

Im Folgendem werden nur noch Fermionen betrachtet.

Jeder Zustand kann mit den Erzeugungsoperatoren aus dem Vakuumzustand erzeugt werden:

$$|i_{1\downarrow}, i_{1\uparrow}, i_{2\downarrow}, i_{2\uparrow}, \dots, i_{N\downarrow}, i_{N\uparrow}\rangle = c_{1\downarrow}^\dagger c_{1\uparrow}^\dagger c_{2\downarrow}^\dagger c_{2\uparrow}^\dagger \dots c_{N\downarrow}^\dagger c_{N\uparrow}^\dagger |\text{vak}\rangle$$

Die Anordnung der Operatoren wird einmal festgelegt und muss im weiteren Verlauf aufgrund der Antikommutatorrelationen (Gl. 4 - 6) beibehalten werden.

Die Anwendung des Vernichtungsoperators auf den Vakuumzustand ergibt 0:

$$c_{i\sigma} |\text{vak}\rangle = 0 \quad (7)$$

Beispielsweise folgt mit den Vertauschungsrelationen für die Anwendung des Vernichtungsoperators auf einen nichtbesetzten Zustand:

$$\begin{aligned} c_{1\downarrow} |i_{1\uparrow}, i_{2\downarrow}, i_{2\uparrow}, \dots, i_{N\downarrow}, i_{N\uparrow}\rangle &= \\ c_{1\downarrow} c_{1\uparrow}^\dagger c_{2\downarrow}^\dagger c_{2\uparrow}^\dagger \dots c_{N\downarrow}^\dagger c_{N\uparrow}^\dagger |\text{vak}\rangle &= \\ (-1)^{2N-1} \cdot c_{1\uparrow}^\dagger c_{2\downarrow}^\dagger c_{2\uparrow}^\dagger \dots c_{N\downarrow}^\dagger c_{N\uparrow}^\dagger c_{1\downarrow} |\text{vak}\rangle &= 0 \end{aligned}$$

## 2.1 Implementation

Das symbolische Rechnen mit Erzeugungs- und Vernichtungsoperatoren stellt das Ziel dieser Arbeit dar. Die Implementation erfolgt mittels *Mathematica*.

Die Operatoren werden übergeführt in Objekte mit denen in *Mathematica* Rechenoperationen durchgeführt werden können:

$$\begin{aligned} c_{i\sigma}^\dagger &\Longrightarrow F[1, i, \sigma] \\ c_{i\sigma} &\Longrightarrow F[-1, i, \sigma] \end{aligned}$$

Der erste Parameter von F kann nur die Werte 1 (Erzeuger) oder -1 (Vernichter) annehmen. Der zweite Parameter gibt den Platz an, wo der Operator wirken soll. Der Wert des dritten Parameters entspricht mit 1 Spin  $\uparrow$  und mit  $-1$  Spin  $\downarrow$  der Eigenschaft des jeweiligen Platzes.

Das Wirken der Operatoren wird im Programm mit nicht kommutativer Multiplikation (*Mathematica*-Funktion: NonCommutativeMultiply) realisiert.

Damit *Mathematica* auch wirklich mit den eben eingeführten Objekten rechnen kann, ist es erforderlich Regeln zu definieren:

- Erweiterung der Funktion 'NonCommutativeMultiply'  
Umgang mit Skalaren, Distributiv-Gesetz
- Vertauschungsrelationen (Gl. 4 - 6)
- Erzeugung einer Ordnung, da generell  $c_{i\sigma}^\dagger c_{j\sigma'}^\dagger \neq c_{j\sigma'}^\dagger c_{i\sigma}^\dagger$
- Pauliprinzip:  $c_{i\sigma}^\dagger c_{i\sigma}^\dagger |\text{vak}\rangle = 0$

Als Beispiel soll der Erwartungswert des Teilchenanzahloperators  $\hat{n}_{i\sigma} = c_{i\sigma}^\dagger c_{i\sigma}$  auf dem Platz  $i = 1$  eines Teilchens mit Spin  $\uparrow$  im Zustand  $|\downarrow\uparrow\rangle$  berechnet werden:

$$\overbrace{F[-1,1,1] ** F[-1,1,-1]}^{\langle\downarrow\uparrow|} ** \overbrace{F[1,1,1] ** F[-1,1,1]}^{\hat{n}_{1\uparrow}} ** \overbrace{F[1,1,-1] ** F[1,1,1]}^{|\downarrow\uparrow\rangle} ** |\text{vak}\rangle$$

wobei der Vakuumzustand  $|\text{vak}\rangle$  auf der rechten Seite stets implizit enthalten ist.

Mittels der definierten Regeln wird versucht die Vernichtungsoperatoren auf die rechte Seite zu bringen, da der gesamte Ausdruck auf den Vakuumzustand wirkt und ein Vernichtungsoperator auf diesen Zustand 0 ergibt.

Der Erwartungswert ergibt sich zu

$$\langle\downarrow\uparrow|\hat{n}_{1\uparrow}|\downarrow\uparrow\rangle = 1,$$

dh ein Teilchen mit Spin  $\uparrow$  befindet sich auf dem Platz  $i = 1$ .

### 3 Hubbard-Modell

Mit dem Formalismus der Erzeugungs- und Vernichtungsoperatoren kann der Hamilton-Operator für wechselwirkende Vielteilchensysteme ausgedrückt werden.

#### 3.1 Hamiltonian

Der Hamiltonoperator eines Festkörpers sollte im Prinzip eine Vielzahl von Beiträgen enthalten:

- Bewegung der Kerne
- Wechselwirkung zwischen den Kernen
- Bewegung der Elektronen
- Wechselwirkung zwischen den Elektronen
- Wechselwirkung zwischen Kernen und Elektronen

Die Schrödingergleichung, die alle diese Beiträge berücksichtigt, kann für mehrere Teilchen unmöglich gelöst werden.

Für viele physikalische Eigenschaften der Festkörper sind nur die Elektronen der äußersten Schale entscheidend. Mit dieser Annahme können der Kern und die inneren Schalen zu einem Ionenrumpf zusammengefasst werden. Weiters kann die Bewegung der Kerne im Vergleich zur Bewegung der Elektronen aufgrund ihrer wesentlich höheren Masse vernachlässigt werden (Born-Oppenheimer-Näherung). Auch bei periodischer und fixer Anordnung der Ionenrümpfe ist man aufgrund der Vielzahl von Wechselwirkungen noch nicht in der Lage die Schrödingergleichung zu lösen. Erst die Näherungen, die auf ein effektives Ein-Elektronen-Problem führen, ermöglichen eine näherungsweise Lösung [1]:

- unabhängige Elektronen  
Hierbei wird die Wechselwirkung zwischen den Elektronen komplett vernachlässigt.
- Hartree-Fock-Näherung  
Diese Näherung beinhaltet die Wechselwirkung der Elektronen mit der Ladungsdichteverteilung der restlichen.
- Local Density Approximation  
Hier berücksichtigt man Austausch- und Korrelationseffekte zwischen den Elektronen

Das einfachste Hubbard-Modell eines Festkörpers reduziert sich auf die Beschreibung eines einzigen, lokalisierten Niveaus je Atom/Gitterplatz[4]. Je Atom/Gitterplatz ergeben sich daraus vier mögliche Konfigurationen:

- Niveau leer
- ein Elektron mit Spin  $\uparrow$
- ein Elektron mit Spin  $\downarrow$
- jeweils ein Elektron mit Spin  $\uparrow$  und Spin  $\downarrow$

Der Hamiltonoperator des Hubbard-Modells besteht aus nur 2 Termen:

- einem kinetischen Term (Hüpfterm), der Elektronen das Hüpfen auf nächstgelegene Nachbarplätze erlaubt, und
- einem Potentialterm, der die Coulomb-Abstoßung zweier Elektronen berücksichtigt, die sich auf dem selben Gitterplatz befinden.

Diese beiden Terme konkurrieren miteinander. Die Elektron-Elektron-Wechselwirkung (Coulomb-Wechselwirkung) möchte die Doppelbesetzung von Plätzen möglichst unterbinden, d.h. die Elektronen sind in ihrer Bewegung eingeschränkt. Der Hüpfterm hingegen versucht die Elektronen über das gesamte Gitter zu verteilen [2].

Der Hamiltonoperator ist nun wie folgt definiert:

$$\hat{H} = \hat{T} + \hat{U} = -t \sum_{\langle i,j \rangle} \sum_{\sigma} (c_{i\sigma}^{\dagger} c_{j\sigma} + c_{j\sigma}^{\dagger} c_{i\sigma}) + U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \quad (8)$$

$c_{i\sigma}^{\dagger}$  Erzeugungsoperator eines Elektrons mit Spin  $\sigma$  auf Platz  $i$

$c_{i\sigma}$  Vernichtungsoperator eines Elektrons mit Spin  $\sigma$  auf Platz  $i$

$\hat{n}_{i\sigma} = c_{i\sigma}^{\dagger} c_{i\sigma}$  Anzahloperator

Die Klammer  $\langle i, j \rangle$  bedeutet, dass über benachbarte Gitterplätzen summiert wird.

Die Hüpfamplitude  $t > 0$  entspricht dem Überlappintegral des Einteilchen-Hamiltonoperators zwischen den Wellenfunktionen zweier Gitterplätze, wobei nur nächste Nachbarn betrachtet werden.  $U > 0$  ist ein Maß für die Stärke der Coloumb-Wechselwirkung [2].

Dieses Modell besitzt offensichtlich nur zwei explizite Parameter:  $t$  und  $U$ . Implizit enthalten sind die Anzahl der Elektronen sowie die Temperatur.

## 3.2 Eigenschaften

Physikalisch interessant ist das Verhalten nahe bei Halbfüllung (Anzahl der Elektronen entspricht der Anzahl der Gitterplätze):

$U = 0$ : System ist leitend (Tight-Binding)  $\Rightarrow$  Metall

$U \neq 0$ : Ladungstransport durch Coloumb-Abstoßung gestört  $\Rightarrow$  für  $U \gg 0$  wird das System ein Isolator

### 3.2.1 spin-charge-separation

In einer Dimension ist es möglich, dass sich Spin und Ladung der Elektronen trennen [5]. Das Elektron "zerfällt" in zwei unabhängige Quasiteilchen (spinon und holon/chargon), die sich mit verschiedener Geschwindigkeit ausbreiten können. Das Spinon trägt den Spin des Elektrons, das holon/chargon die Ladung.

### 3.2.2 reale Materialien

Das Hubbard-Modell bildet die qualitative Grundlage für Systeme, wo Wechselwirkungen zwischen Elektronen eine entscheidene Rolle spielen.

Um das Verhalten vieler Materialien modellieren zu können, muss der Hamiltonoperator (Gl. 8) erweitert werden:

- $\Rightarrow$  Coloumb-Wechselwirkung nicht nur auf einem Gitterplatz
- $\Rightarrow$  Berücksichtigung mehrerer Orbitale
- $\Rightarrow$  Hüpfen auch auf andere Plätze als zum nächsten Nachbarn erlauben

### 3.3 Symmetrien

Im eindimensionalen Hubbard-Modell wird eine Vielzahl von Größen erhalten [3]. Auf die Erhaltung von Teilchenzahl und Gesamtspin wird nachfolgend genauer eingegangen.

#### 3.3.1 Eichsymmetrie

Der Hamiltonoperator ist unter folgender Transformation invariant:

$$c_{i\sigma}^\dagger \rightarrow \exp(i\alpha)c_{i\sigma}^\dagger, \quad c_{i\sigma} \rightarrow \exp(-i\alpha)c_{i\sigma} \quad (9)$$

Der Operator dieser unitären Transformation lautet  $\hat{U}(\alpha) = \exp(i\alpha\hat{N})$ . Daraus folgt, dass die Teilchenzahl  $\hat{N} = \sum_{i\sigma} c_{i\sigma}^\dagger c_{i\sigma}$  erhalten bleibt.

Die Erhaltung der Teilchenanzahl ist auch in den verschwindenden Kommutatoren mit  $\hat{H}$  ersichtlich:

$$[\hat{N}, \hat{H}] = 0 \quad (10)$$

$$[\hat{N}_\sigma, \hat{H}] = 0 \quad \text{mit } \hat{N}_\sigma = \sum_i \hat{n}_{i\sigma} \quad (11)$$

#### 3.3.2 Spinsymmetrie

Mit Hilfe der Pauli-Matrizen

$$\tau_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \tau_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \tau_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (12)$$

kann man lokale Spindichteoperatoren definieren:

$$\hat{S}_{\alpha,i} = \frac{\hbar}{2} \sum_{\sigma\sigma'} c_{i\sigma}^\dagger (\tau_\alpha)_{\sigma\sigma'} c_{i\sigma'} \quad (13)$$

mit  $\alpha = x, y, z$

Globale Spindichteoperatoren ergeben sich zu

$$\hat{S}_\alpha = \sum_i S_{\alpha,i}. \quad (14)$$

Für diese Operatoren gelten die Vertauschungsrelationen

$$[\hat{S}_{\alpha,i}, \hat{S}_{\beta,i}] = i\hbar\varepsilon_{\alpha\beta\gamma} \hat{S}_{\gamma,i} \quad (15)$$

$$[\hat{S}_\alpha, \hat{S}_\beta] = i\hbar\varepsilon_{\alpha\beta\gamma} \hat{S}_\gamma \quad (16)$$

Die Spindichteoperatoren vertauschen mit  $\hat{H}$ , somit können  $\hat{H}$ ,  $\hat{S}^2$ ,  $\hat{S}_\alpha$  gleichzeitig diagonalisiert werden können.

### 3.4 Zeitentwicklung

Da der Hamiltonoperator (Gl. 8) nicht explizit zeitabhängig ist, kann man zunächst die zeitunabhängige Schrödinger-Gleichung lösen. Sie lautet somit

$$\hat{H}|\psi_n\rangle = E_n|\psi_n\rangle \quad (17)$$

Jeder Eigenzustand  $|\psi_n\rangle$  entwickelt sich mit Zeit zu

$$|\psi_n(t)\rangle = e^{-\frac{i}{\hbar} E_n t} |\psi_n(0)\rangle \quad (18)$$

Ein beliebiger Zustand  $|\psi\rangle$  kann in Eigenzustände  $|\psi_n\rangle$  entwickelt werden:

$$|\psi\rangle = \sum_n c_n |\psi_n\rangle \quad (19)$$

mit  $c_n = \langle \psi_n | \psi \rangle$

Die Zeitentwicklung dieses Zustandes lautet:

$$|\psi(t)\rangle = \sum_n c_n |\psi_n(t)\rangle \quad (20)$$

#### 3.4.1 Dissipation

Der Grundzustand ist der Eigenzustand geringster Energie. Wird in der Zeitentwicklung (Gl. 18) ein Reibungsterm berücksichtigt, d.h. das System kann Energie "abbauen", strebt das System mit zeitlichen Verlauf zum Grundzustand.

$$|\psi(t)\rangle = N(t) \cdot \sum_n e^{-\eta E_n t} c_n |\psi_n(t)\rangle \quad (21)$$

$$\eta > 0,$$

$N(t)$  ... zeitabhängige Normierungskonstante

## 4 Simulation

Der Hauptteil des Programmes umfasst

1. Berechnung der Hamilton-Matrix
2. deren Diagonalisierung
3. Zeitentwicklung verschiedener Anfangszustände (Gl. 20)
4. deren grafische Darstellung

Die vorher erwähnten Symmetrien werden ausgenutzt um die Größe der zu diagonalisierenden Matrizen zu reduzieren. Aufgrund der Erhaltung der Anzahl der Teilchen mit Spin  $\uparrow$  als auch mit Spin  $\downarrow$  (Gl.11) kann  $\hat{H}$  in Blöcke gegliedert werden. Für die Bestimmung der Eigenenergien und -zustände genügt somit die Diagonalisierung des jeweiligen Blockes.

Simuliert werden Erwartungswerte

$$\langle \hat{H} \rangle \dots \text{Energie} \quad (22)$$

$$\langle \hat{n}_{i\sigma} \rangle \dots \text{Teilchendichte am Platz } i \text{ mit Spin } \sigma \quad (23)$$

und Korrelationen (on-site)

$$\langle \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \rangle - \langle \hat{n}_{i\uparrow} \rangle \langle \hat{n}_{i\downarrow} \rangle \quad (24)$$

### 4.1 Systemgröße

Die Anzahl der möglichen Zustände  $k$  und damit die Größe der Hilbertraumes steigt rasant mit zunehmender Systemgröße  $N$ . Mit  $N_{\uparrow}$  als Anzahl der Teilchen mit Spin  $\uparrow$  und  $N_{\downarrow}$  als Anzahl der Teilchen mit Spin  $\downarrow$  ergeben sich  $k$  mögliche Konfigurationen:

$$k = \binom{N}{N_{\uparrow}} \cdot \binom{N}{N_{\downarrow}} = \frac{N!}{N_{\uparrow}!(N - N_{\uparrow})!} \cdot \frac{N!}{N_{\downarrow}!(N - N_{\downarrow})!} \quad (25)$$

Die maximale Anzahl an Zuständen erhält man bei Halbfüllung ( $N_{\uparrow} = N_{\downarrow} = \frac{N}{2}$ ). In Tabelle 1 ist für unterschiedliche Systemgrößen die Größe des Hilbertraums bei Halbfüllung dargestellt. Wie man erkennt, ist bereits bei sehr kleinen Systemgrößen die derzeit mögliche Speicherkapazität erreicht.

Dennoch besteht die Möglichkeit auf größere Systeme zu extrapolieren. Das System wird dabei in eine Vielzahl kleiner, lösbarer Cluster unterteilt und darin gelöst. Mittels Störungsrechnung erfolgt eine Kopplung zwischen den verschiedenen Clustern (Cluster Perturbation Theory) [6].

Tabelle 1: Größe des Hilbertraums bei Halbfüllung

Systemgröße $N$	Anzahl Zustände $k$
2	4
4	36
6	400
8	4900
10	63504
20	34134779536

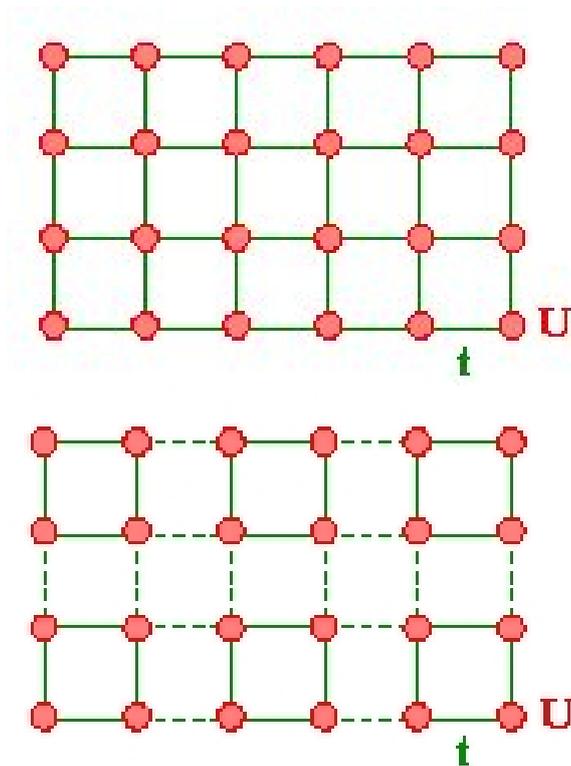


Abbildung 1: Aufteilung eines System in kleinere, lösbare Cluster  
die strichlierten Linien im unteren Bild stellen die Kopplungen zwischen den Clustern dar

## 4.2 Simulationsergebnisse

### 4.2.1 Spin-Charge-Separation

Für diese Simulation wird auf 7 Gitterplätzen zuerst der Grundzustand von einem 2-Teilchen-System (1 Teilchen mit Spin  $\uparrow$ , 1 Teilchen mit Spin  $\downarrow$ ) bei periodischen Randbedingungen (Ring) berechnet. Zu diesem Grundzustand wird ein weiteres Teilchen mit Spin  $\uparrow$  hinzugefügt. Im Anschluss erfolgt die Entwicklung dieses neuen Zustandes mit der Zeit.

Abbildung 2 zeigt den unterschiedlichen Verlauf der Erwartungswerte von Spin (durchgezogen) und Ladung (strichliert) an vier verschiedenen Zeitpunkten, wobei hier nur die Änderungen der Erwartungswerte bezogen auf die Erwartungswerte bei  $t = 0$  dargestellt werden.

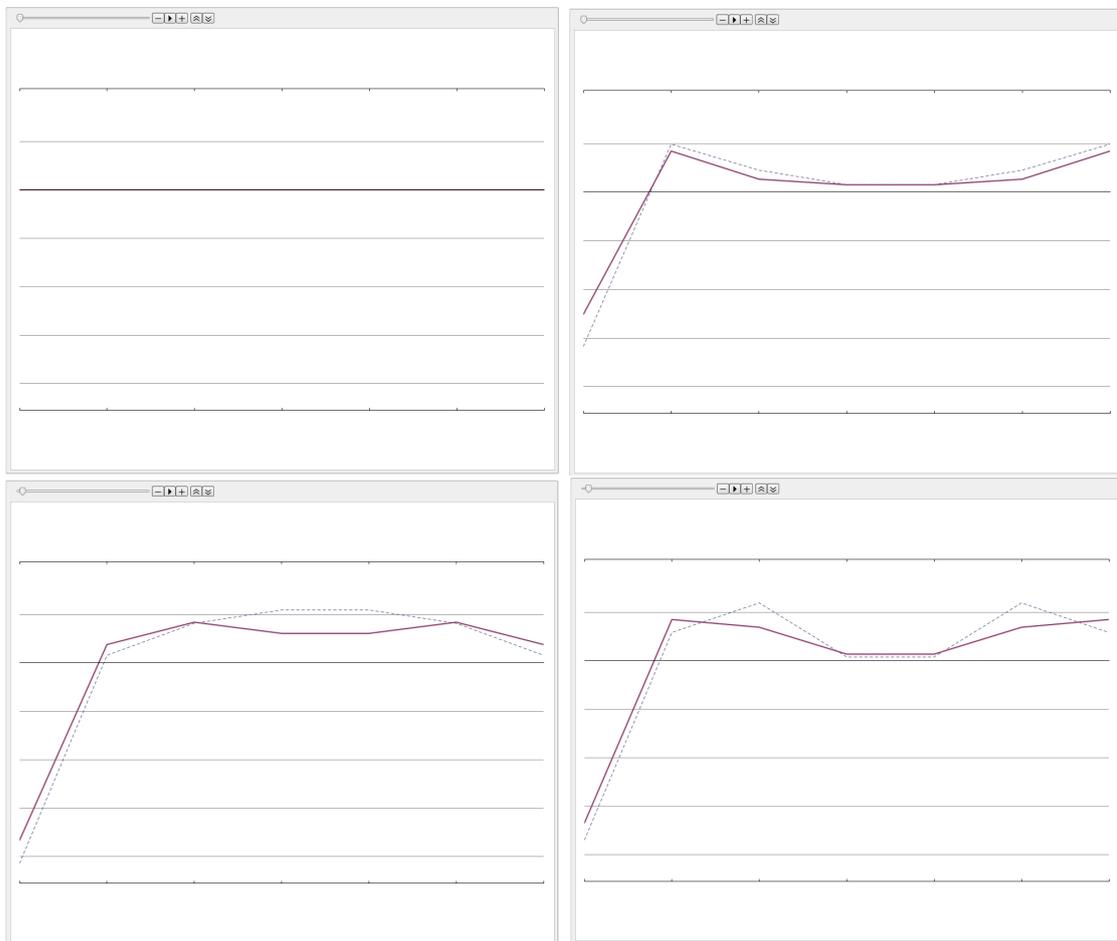


Abbildung 2: Trennung von Spin (durchgezogen) und Ladung (strichliert) - dargestellt an verschiedenen Zeitpunkten

### 4.2.2 Dissipation

Der Ausgangszustand dieser Simulation ist ein System von vier Teilchen auf acht Gitterplätzen bei nicht-periodischen Randbedingungen (kein Ring). Zwei Teilchen mit Spin  $\uparrow$  befinden sich auf den Gitterplätzen 1 und 5, zwei Teilchen mit Spin  $\downarrow$  auf den Gitterplätzen 3 und 6.

Die Abbildungen 3 bis 6 zeigen die zeitliche Entwicklung des Anfangszustandes zu verschiedenen Zeitpunkten ( $t_0 < t_1 < t_2 < t_3$ ), wobei die Farbe orange den Teilchen mit Spin  $\uparrow$  und die Farbe lila den Teilchen mit Spin  $\downarrow$  entspricht.

Durch die Berücksichtigung eines Reibungsterm in der Zeitentwicklung (Gl. 21) strebt das System zum Grundzustand. Jede der nachfolgenden Abbildungen enthält unterschiedliche Stärken der Coloumb-Wechselwirkung:  $U = 0$  (keine Wechselwirkung),  $U = 1$  (schwache Wechselwirkung) und  $U = 100$  (starke Wechselwirkung).

Jede Zeile einer Abbildung besteht aus der aktuellen Konfiguration (Teilchendichte an den verschiedenen Gitterplätzen für beide Spinarten) auf der linken Seite, aus einer Darstellung der Erwartungswerte von kinetischer (schwarz) sowie potentieller Energie (rot) der jeweiligen Konfiguration (Mitte) sowie aus dem Grundzustand (rechts).

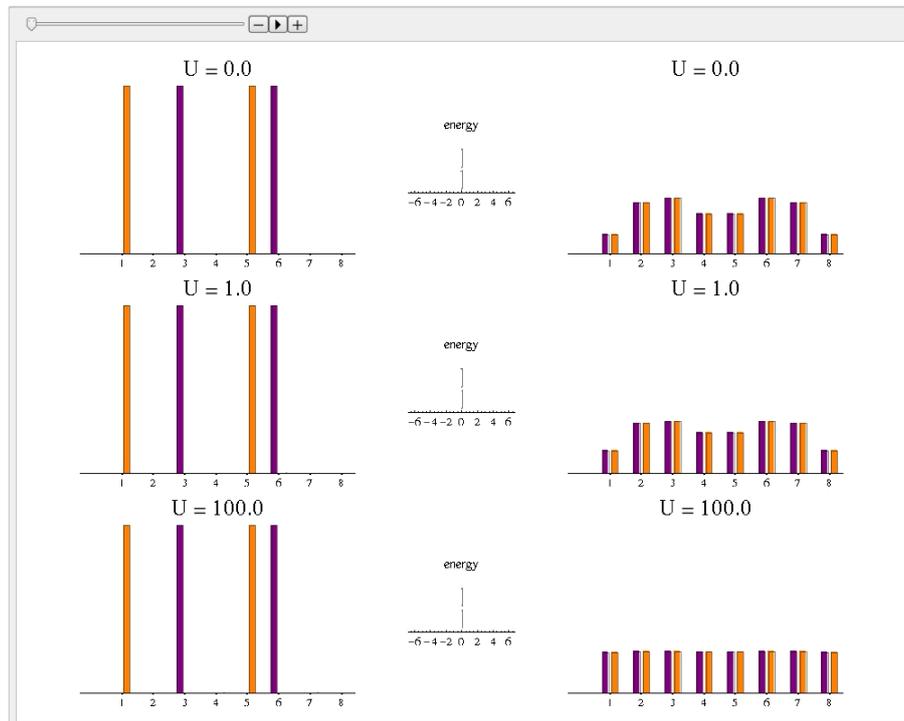


Abbildung 3: Konfigurationen zum Zeitpunkt  $t_0 = 0$

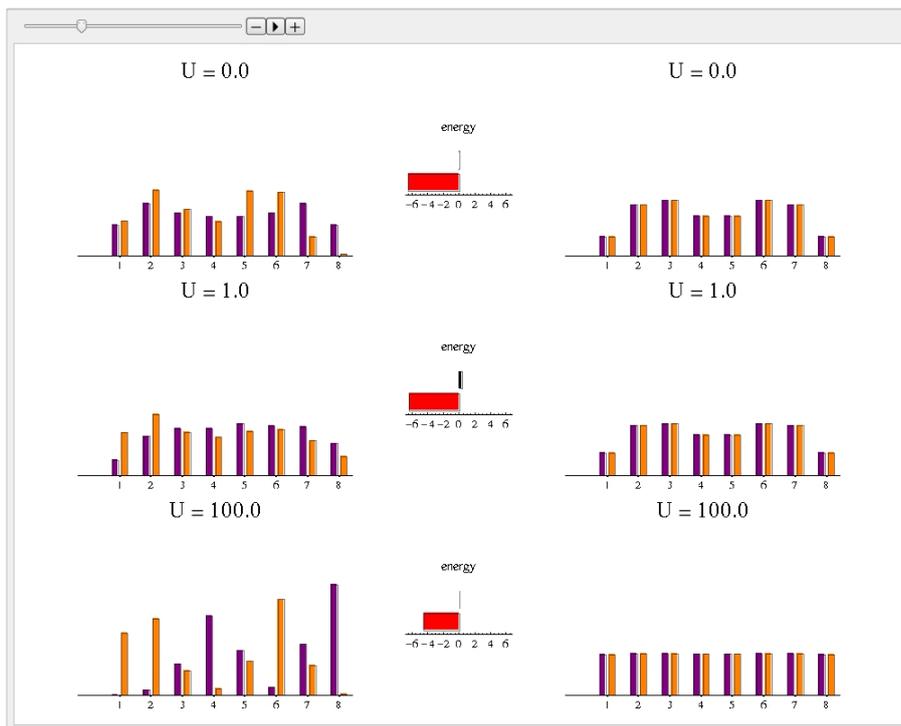


Abbildung 4: Konfigurationen zum Zeitpunkt  $t_1$

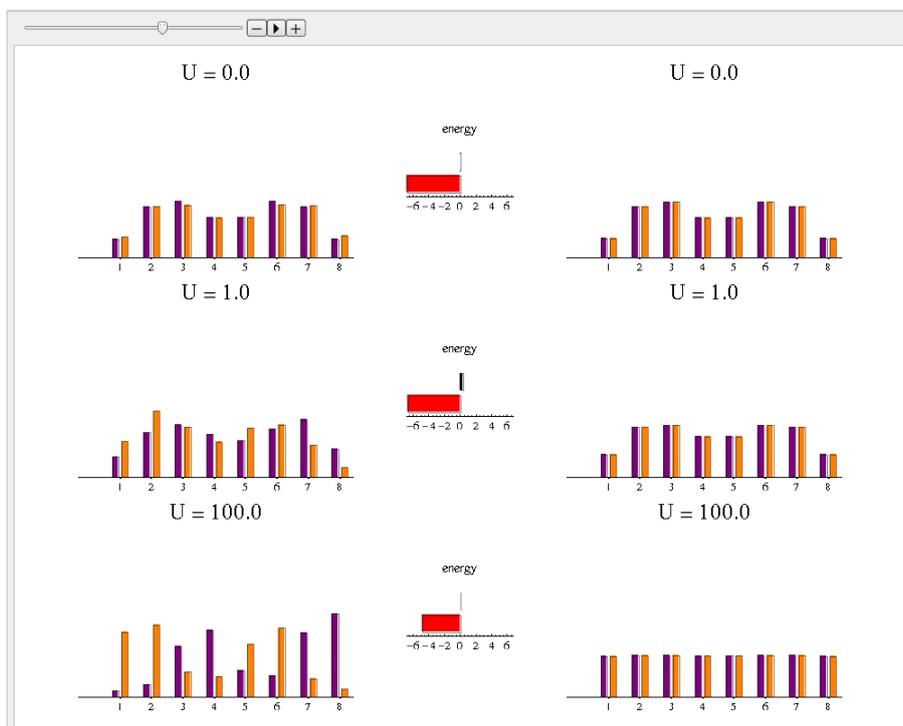
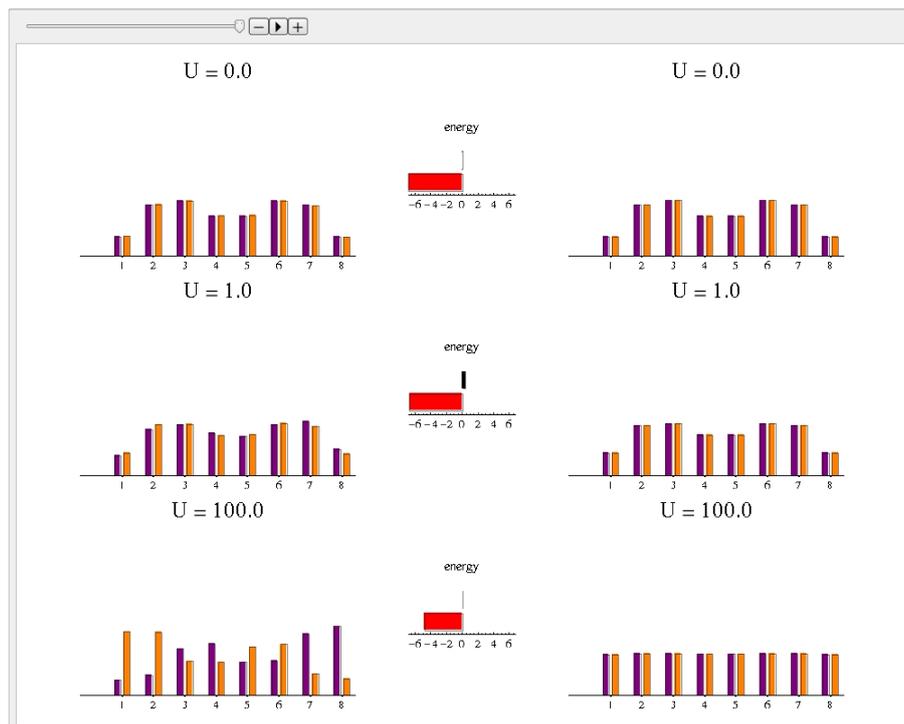


Abbildung 5: Konfigurationen zum Zeitpunkt  $t_2$

Abbildung 6: Konfigurationen zum Zeitpunkt  $t_3$ 

Die Abbildungen 3 bis 6 demonstrieren das zeitliche Verhalten des selben Anfangszustandes bei unterschiedlicher Stärke der Coloumb-Wechselwirkung. Das Streben des Systems zum Grundzustandes wird durch die Wechselwirkung gestört. Je stärker die Coloumb-Abstoßung, desto länger dauert es, bis der Grundzustand erreicht wird, da je nach Stärke der Wechselwirkung die Teilchen in ihrer Bewegung beeinträchtigt werden.

### 4.2.3 Korrelation

Diese Simulation passiert auf 8 Gitterplätzen mit zwei Teilchen bei nicht-periodischer Randbedingung - Ring nicht geschlossen. Der Anfangszustand lautet: ein Teilchen mit Spin  $\uparrow$  befindet sich auf Gitterplatz 8 das zweite Teilchen mit Spin  $\downarrow$  auf Gitterplatz 1.

Die Abbildungen 7 bis 10 zeigen für verschiedenen Zeitpunkte ( $t_0 < t_1 < t_2 < t_3$ ) jeweils im oberen Teilbild die aktuelle Konfiguration (Teilchendichte am jeweiligen Gitterplatz für beide Spinarten) und im unteren Teilbild die on-site Korrelation (Gl. 24) an jedem Gitterplatz. Zusätzlich enthalten sie auch unterschiedliche Coloumb-Wechselwirkungen (links:  $U = 1$ , rechts:  $U = 100$ ).

Die Farbe orange entspricht wiederum den Teilchen mit Spin  $\uparrow$ , die Farbe lila den Teilchen mit Spin  $\downarrow$ .

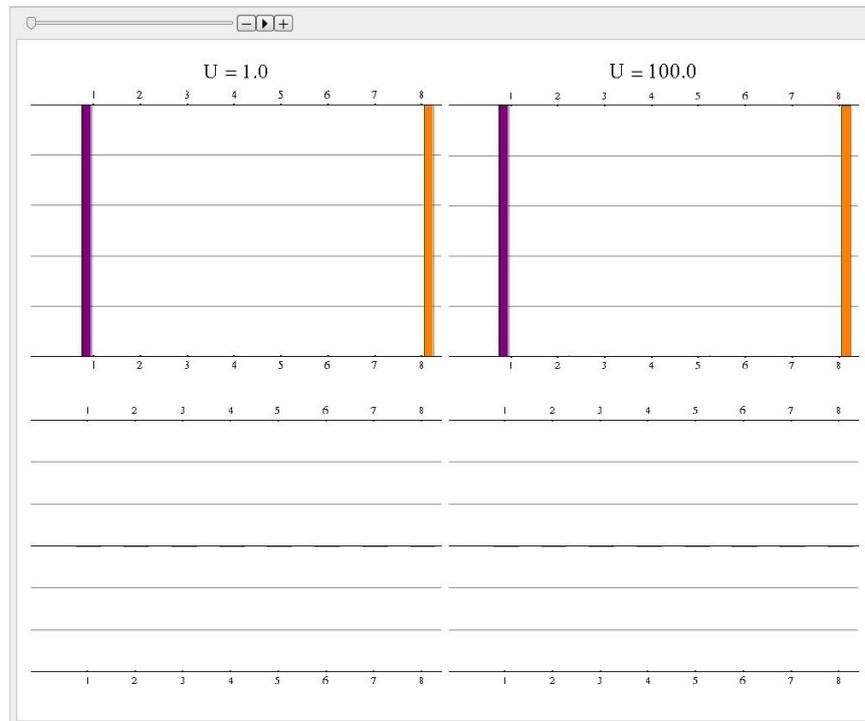


Abbildung 7: Konfigurationen zum Zeitpunkt  $t_0$

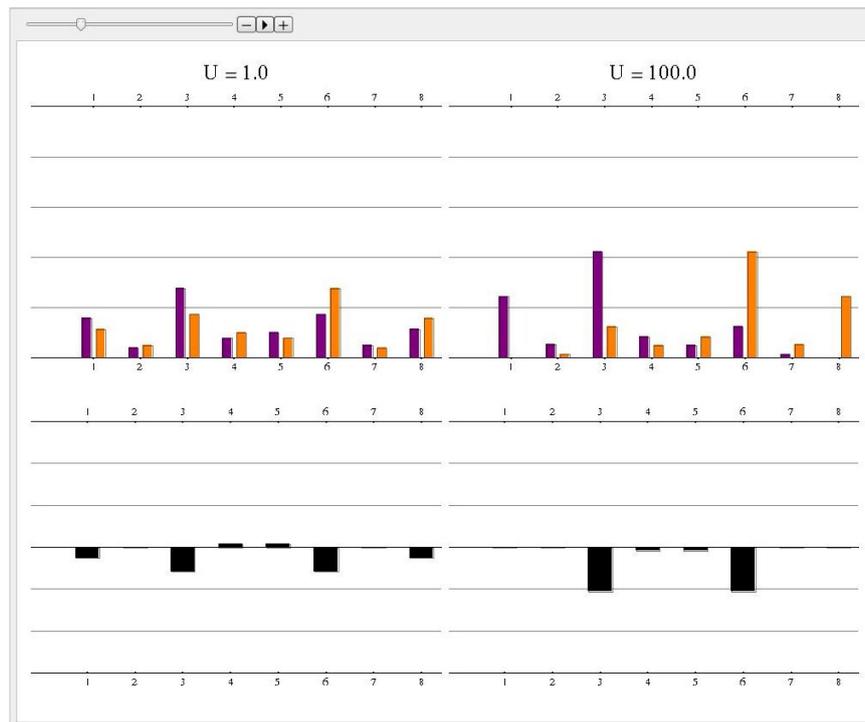


Abbildung 8: Konfigurationen zum Zeitpunkt  $t_1$

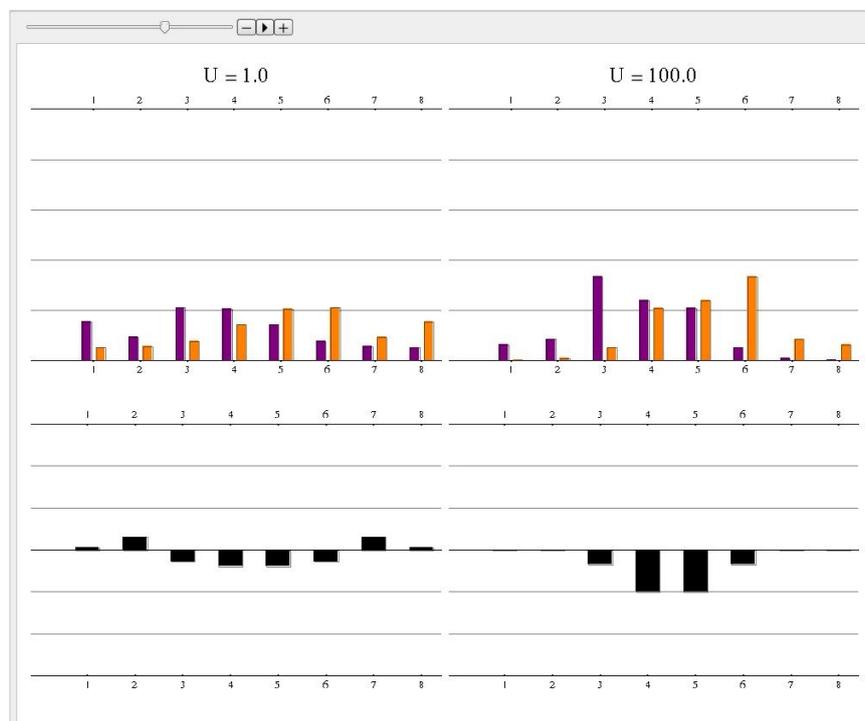
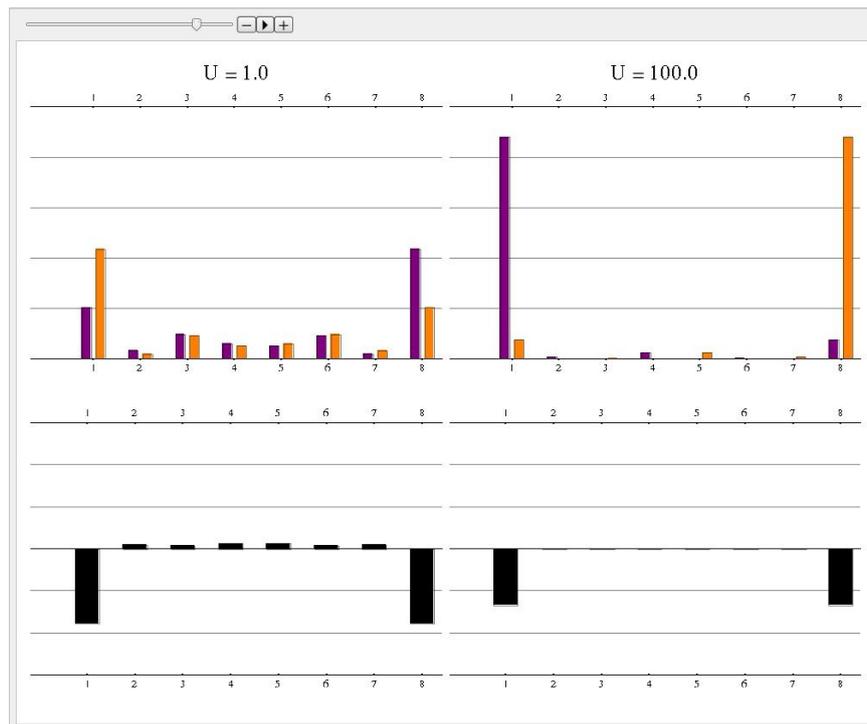


Abbildung 9: Konfigurationen zum Zeitpunkt  $t_2$

Abbildung 10: Konfigurationen zum Zeitpunkt  $t_3$ 

Aus den Abbildungen 7 bis 10 erkennt man, dass sich bei unterschiedlich starker Coloumb-Wechselwirkung die Teilchen mit fortschreitender Zeitentwicklung andersartig verhalten. Bei  $U = 100$  ist die Wahrscheinlichkeit ein Teilchen am anderen Ende der Kette zu finden äußerst gering - in allen 4 Abbildungen bleibt das Teilchen mit Spin  $\uparrow$  (orange) auf der rechten Seite der Kette während das sich das Teilchen mit Spin  $\downarrow$  (lila) grobteils auf der linken Seite aufhält.

## Literatur

- [1] F. Gebhard: *The Mott Metal-Insulator Transition*, 1997, Springer
- [2] P. Fazekas: *Lecture notes on electron correlation and magnetism* Series in modern condensed matter physics, 1999, World Scientific
- [3] Siehe z.B. A. Mielke: Skript zur Vorlesung *Starke Korrelationen und Magnetismus*, 2008, Universität Heidelberg
- [4] N. W. Ashcroft, D. N. Merim: *Festkörperphysik*, 3. Auflage, 2007, Oldenbourg
- [5] E. A. Jagla and K. Hallberg and C. A. Balseiro: *Numerical study of charge and spin separation in low dimensional systems* in Physical Review B, Volume 47, Number 10, 1993  
Für ein Review siehe z.B. auch J. Voit: *one-dimensional Fermi liquids* in Reports on Progress in Physics, Volume 58, Number 9 (977-1116), 1995
- [6] Siehe z.B. M. Aichhorn, *Ordering Phenomena in Strongly-Correlated Systems: Cluster Perturbation Theory Approaches*, Dissertation, TU Graz, 2004, sowie den darin enthaltenen Referenzen

---

## GRUNDLAGEN

```
(* Erweiterung "*" *)
Unprotect[NonCommutativeMultiply];
NonCommutativeMultiply[A_, (B_ + C_)] := A ** B + A ** C;
NonCommutativeMultiply[(B_ + C_), A_] := B ** A + C ** A;
NonCommutativeMultiply[A_, (-B_)] := -A ** B;
NonCommutativeMultiply[(-A_), B_] := -A ** B;
Protect[NonCommutativeMultiply];

(* Regeln für "*" *)
subsnoncomm := {NonCommutativeMultiply[A_, r_] => r A /; CnumQ[r],
  NonCommutativeMultiply[r_, A_] => r A /; CnumQ[r],
  NonCommutativeMultiply[A_, r_B_] => r A ** B /; CnumQ[r],
  NonCommutativeMultiply[r_A_, B_] => r A ** B /; CnumQ[r],
  NonCommutativeMultiply[Times[A_, B_]] => Times[A, B]}

subsnonpower := {A_^n_Integer => noncommpower[A, n] /; (! FreeQ[A, NonCommutativeMultiply])}
noncommpower[A_, n_] := Apply[NonCommutativeMultiply, Table[A, {i, 1, n}]]

(* Definition Buchstaben *)
cnumbers = Map[ToExpression, CharacterRange["a", "z"]];
fermions = {F, G, H};
bosons = {A, B, C};
CnumQ[xx_] := NumericQ[xx] || MemberQ[cnumbers, xx] || MemberQ[cnumbers, Head[xx]]
FermionQ[xx_] := MemberQ[fermions, xx] || MemberQ[fermions, Head[xx]]
BosonQ[xx_] := MemberQ[bosons, xx] || MemberQ[bosons, Head[xx]]

(* Vorzeichen Bosonen/Fermionen *)
FermiSign[xx_, yy_] := -1 /; FermionQ[xx] && FermionQ[yy];
FermiSign[xx_, yy_] := 1 /; (! FermionQ[xx]) || (! FermionQ[yy])

(* Definition Kronecker-Delta *)
KD[{m___}, {n___}] := If[Apply[And, Table[{m}[[i]] == {n}[[i]], {i, 1, Length[{m}]}]], 1, 0]

(* Abfrage Operator *)
OperatorQ[xx_] := FermionQ[xx] || BosonQ[xx]
```

### ■ ERZEUGER / VERNICHTER

erster Index:Erzeuger+1,Vernichter-1

zweiter Index:Teilchennummer

dritter Index:Spin (up=+1;down=-1)

```

ToNormalOrder := {F[-1, m___] ** F[1, n___] => -F[1, n] ** F[-1, m] + cac[F[-1, m], F[1, n]]}
ToOrder := {F[i_, m___] ** F[i_, n___] => -F[i, n] ** F[i, m] /; (! OrderedQ[{{m}, {n}}])}
subscac :=
  {cac[F[-1, m___], F[1, n___]] => KD[{m}, {n}],
   cac[F[1, m___], F[-1, n___]] => KD[{m}, {n]}}
similaroperator := {F[i_, m___] ** F[i_, m___] => 0 /; ((i == 1) || (i == -1))}
sub0 := {NonCommutativeMultiply[xx___, 0, yy___] => 0}
sub1 := {NonCommutativeMultiply[1, xx_] => xx, NonCommutativeMultiply[xx_, 1] => xx}
arrange := Join[ToNormalOrder, ToOrder, subscac, sub0, sub1]
rules := Join[ToNormalOrder, ToOrder, subscac, similaroperator, subnoncomm]
annihilation := {F[-1, n___] => 0}
ToMatrix := Join[annihilation, sub0]
bra[k_] := NonCommutativeMultiply@@
  Reverse[Table[k[[i]], {i, 1, Length[k]}] /. F[i_, m___] -> F[-i, m]]
vernicht := {F[m___] :> 0}
ApplyRules[x_] :=
  {x //. subnoncomm //. arrange /. similaroperator /. sub0 //. arrange /. similaroperator /.
   sub0 //. ToMatrix}

```

## ■ Anwendung der Regeln (Beispiele)

```

F[-1, 1, 1] ** F[1, 1, 1] /. ToNormalOrder /. subscac
1 - F[1, 1, 1] ** F[-1, 1, 1]

(* beliebiger Zustand *)
uz = F[1, 1, -1] ** F[1, 1, 1] ** F[1, 3, -1];

(* Anzahloperator *)
AZ = F[1, 1, 1] ** F[-1, 1, 1]
F[1, 1, 1] ** F[-1, 1, 1]

(* Erwartungswert *)
bra[uz] ** AZ ** uz //. rules //. ToMatrix
1

```

## ■ KOMMUTATORREGEL

```
commregel := {comm[xx_, yy_] => xx ** yy - yy ** xx}
```

## ■ TEILCHENANZAHLOPERATOR

```
Na[xx_, i_] := xx[1, i] ** xx[-1, i]
```

## ■ SPINDICHTOPERATOR (PAULI-MATRIZEN)

erster Index:Ausrichtung (x=1,y=2,z=3)

zweiter Index:Teilchennummer

```

Tau = {{{0, 1}, {1, 0}}, {{0, -I}, {I, 0}}, {{1, 0}, {0, -1}}};
S[dir_, num_] :=
  1/2 Sum[Sum[Tau[[dir, i, j]] F[1, num, If[i == 1, -1, 1]] ** F[-1, num, If[j == 1, -1, 1]],
    {i, 1, 2}], {j, 1, 2}]

```

```

S[1, 1] (* x-Richtung *)
S[2, 1] (* y-Richtung *)
S[3, 1] (* z-Richtung *)

1
-- (F[1, 1, -1] ** F[-1, 1, 1] + F[1, 1, 1] ** F[-1, 1, -1])
2

1
-- (-i F[1, 1, -1] ** F[-1, 1, 1] + i F[1, 1, 1] ** F[-1, 1, -1])
2

1
-- (F[1, 1, -1] ** F[-1, 1, -1] - F[1, 1, 1] ** F[-1, 1, 1])
2

S[2, 1] ** F[1, 1, -1] //. rules //. annihilation //. rules

1
-- i F[1, 1, 1]
2

(* Vertauschung Spinoperatoren *)
(* gerade Permutation *)
Table[comm[S[i, 1], S[Mod[i, 3] + 1, 1]] - I S[Mod[i + 1, 3] + 1, 1], {i, 1, 3}] /. commregel //.
  rules // Simplify
(* ungerade Permutation *)
Table[comm[S[i, 1], S[Mod[i + 1, 3] + 1, 1]] + I S[Mod[i, 3] + 1, 1], {i, 1, 3}] /. commregel //.
  rules // Simplify

{0, 0, 0}

{0, 0, 0}

S2 = S[1, 1] ** S[1, 1] + S[2, 1] ** S[2, 1] + S[3, 1] ** S[3, 1] //. rules // Simplify

3
-- (F[1, 1, -1] ** F[-1, 1, -1] + F[1, 1, 1] ** F[-1, 1, 1] +
4
  2 F[1, 1, -1] ** F[1, 1, 1] ** F[-1, 1, -1] ** F[-1, 1, 1])

test := {NonCommutativeMultiply[F[i_, m___], F[j_, m___]] =>
  1 /; ((i == 1 && j == -1) || (i == -1 && j == 1))}

S2 ** F[1, 1, 1] ** F[1, 2, 1] //. rules /. test /. sub1 /. annihilation /. sub0 // Simplify

3
-- F[1, 1, 1] ** F[1, 2, 1]
4

Splus = (S[1, 1] + I S[2, 1]) //. rules // Simplify
Sminus = (S[1, 1] - I S[2, 1]) //. rules // Simplify

F[1, 1, -1] ** F[-1, 1, 1]

F[1, 1, 1] ** F[-1, 1, -1]

(Splus + Sminus) / 2 // Simplify
(Splus - Sminus) / (2 I) // Simplify

1
-- (F[1, 1, -1] ** F[-1, 1, 1] + F[1, 1, 1] ** F[-1, 1, -1])
2

1
-- i (F[1, 1, -1] ** F[-1, 1, 1] - F[1, 1, 1] ** F[-1, 1, -1])
2

```

---

## HUBBARD-MODELL auf 2 Plaetzen

```

(* Hamiltonian *)
H = -t Sum[F[1, 1, i] ** F[-1, 2, i] + F[1, 2, i] ** F[-1, 1, i], {i, -1, 1, 2}] +
  u Sum[Na[F, i, 1] ** Na[F, i, -1], {i, 1, 2}];

```



## HUBBARD-MODELL auf mehreren Plaetzen

```
num = 4;
```

### ■ Hamiltonian

```
(* Hopping *)
hopping := F[1, j, i] ** F[-1, j+1, i] + F[1, j+1, i] ** F[-1, j, i]
periodicHopping := F[1, 1, i] ** F[-1, num, i] + F[1, num, i] ** F[-1, 1, i]

(* Hamiltonian *)
T = -t Sum[hopping, {i, -1, 1, 2}, {j, 1, num-1}] - y Sum[periodicHopping, {i, -1, 1, 2}];
U = u Sum[Na[F, i, 1] ** Na[F, i, -1], {i, 1, num}];
H = T + U;

(* S_tot in einer Richtung *)
Sges[aus_] := Sum[S[aus, i], {i, 1, num}]

(* [S_tot, H] = 0 ! *)
Table[comm[Sges[i], H /. y -> t], {i, 1, 3}] /. commregel //. rules // Simplify
{0, 0, 0}

(* [S^2, H] = 0 ! *)
S2 = Sum[Sges[aus] ** Sges[aus], {aus, 1, 3}] //. rules // Simplify;
comm[S2, H /. y -> t] /. commregel //. rules // Simplify
0
```

### ■ Zustände

```
zustand[i_] := {1, F[1, i, -1], F[1, i, 1], F[1, i, -1] ** F[1, i, 1]}

(* alle möglichen Zustände bei num Plaetzen *)
c1 = Flatten[Outer[NonCommutativeMultiply,
  Sequence@@Table[zustand[i], {i, 1, num}]]] //. sub1;

(* Anzahl der Zustände(Erzeuger) mit Spin up oder Spin down *)
npart[j_, x_] := Count[Flatten[{x /. NonCommutativeMultiply -> List}], F[1, _, j]]

(* Ersetzung in SparseArray *)
ReplaceSparse[ff_, rr_Rule] :=
  (ff /. rr) /. HoldPattern[SparseArray[xx_]] -> SparseArray@@({xx} /. rr)

(* return alle Zustände(Erzeuger) mit Anzahl up und Anzahl down *)
kzu[up_, down_] := Select[c1, npart[1, #] == up && npart[-1, #] == down &]
```

### ■ Matricelemente von H

```
Table[Length[kzu[i, j]], {i, 0, num}, {j, 0, num}]
{{1, 2, 1}, {2, 4, 2}, {1, 2, 1}}
```

```

up = 0; down = 2;
Timing[
  (* Zustand rechts *)
  k1 = kzu[up, down];
  (* Zustand links *)
  k2 = Table[Nc[{k1[[i]]}], {i, 1, Length[k1]} /. Nc[xx___] :=
    Nc @@ Reverse[Flatten[xx /. NonCommutativeMultiply -> List]] /. F[i_, m___] -> F[-i, m];
  (* Hamiltonian anwenden *)
  l1 = ApplyRules[Table[H ** k1[[i]], {i, 1, Length[k1]}]][[1]];
  l2 = k2 /. Nc -> NonCommutativeMultiply /. sub1;
  (* Matrix ausgeben *)
  ho = ApplyRules[Outer[NonCommutativeMultiply, l2, l1]][[1]] //. vernicht //. subnoncomm //
    Simplify;
  h[up, down] = SparseArray[ho];
{0.024001, Null}

Save["christian/mathematica/theo-physik/data/2_hamiltonian.dat", h]

<< "christian/mathematica/theo-physik/data/2_hamiltonian.dat";

ReplaceSparse[h[1, 1], y -> t] // MatrixForm

$$\begin{pmatrix} u & -2t & 2t & 0 \\ -2t & 0 & 0 & -2t \\ 2t & 0 & 0 & 2t \\ 0 & -2t & 2t & u \end{pmatrix}$$

ev2 = Eigenvalues[ReplaceSparse[h[1, 1], y -> t]]

$$\left\{0, u, \frac{1}{2} \left(u - \sqrt{64t^2 + u^2}\right), \frac{1}{2} \left(u + \sqrt{64t^2 + u^2}\right)\right\}$$


```

## ■ Matricelemente von $S^2$

```

{k1, k2 /. Nc -> NonCommutativeMultiply //. rules} // Transpose // MatrixForm;

d1 = Table[S2 ** k1[[i]], {i, 1, Length[k1]}] //. rules //. ToMatrix //. rules // Simplify;
d2 = k2 /. Nc -> NonCommutativeMultiply //. rules;

s2 = Outer[NonCommutativeMultiply, d2, d1] //. rules //. ToMatrix //. vernicht //.
  subnoncomm // Simplify;
s2 // MatrixForm
s2 = SparseArray[%];
Eigenvalues[s2]

```

## ■ Zeitentwicklung

```

(* Hamilton-Matrix *)
Clear[h];
<< "christian/mathematica/theo-physik/data/4_hamiltonian.dat";

up = 1; down = 1;

kzu[up, down][[10]]

F[1, 1, -1] ** F[1, 4, 1]

(* Anfangszustand *)
anfang = 10;
psi0 = Table[0, {i, 1, Length[kzu[up, down]}]];
psi0[[anfang]] = 1;

```

```

(* potentielle/kinetische Energie in MatrixForm*)
matH = ReplaceSparse[h[up, down], t → 1.];
matU = DiagonalMatrix[Diagonal[matH]];
matT = matH - matU;

ew[period_, ww_] := Eigenvalues[ReplaceSparse[ReplaceSparse[matH, y → period], u → ww]];
ev[period_, ww_] := Eigenvectors[ReplaceSparse[ReplaceSparse[matH, y → period], u → ww]];

periodic = 0.;
pot = {1., 0., -100.};
Table[eigw[i] = ew[periodic, pot[[i]]], {i, 1, 3}];
Table[eigv[i] = ev[periodic, pot[[i]]], {i, 1, 3}];

(* Zeitentwicklungsvektor *)
eps = 0.0;
psil[t_, w_] := Sum[psi0.eigv[w][[i]] Exp[-I eigw[w][[i]] t (1 - I eps)] eigv[w][[i]],
  {i, 1, Length[kzu[up, down]]}];
psi[t_, w_] := #/ Sqrt[Conjugate[#].#] &[psil[t, w]]

```

---

## spin-charge separation

```
num = 7;
```

### ■ Hamiltonian

```
(* Hopping *)
hopping := F[1, j, i] ** F[-1, j + 1, i] + F[1, j + 1, i] ** F[-1, j, i]
periodicHopping := F[1, 1, i] ** F[-1, num, i] + F[1, num, i] ** F[-1, 1, i]

(* Hamiltonian *)
T = -t Sum[hopping, {i, -1, 1, 2}, {j, 1, num - 1}] - y Sum[periodicHopping, {i, -1, 1, 2}];
U = u Sum[Na[F, i, 1] ** Na[F, i, -1], {i, 1, num}];
H = T + U;

(* S_tot in einer Richtung *)
Sges[aus_] := Sum[S[aus, i], {i, 1, num}]
```

### ■ Zustände

```
zustand[i_] := {1, F[1, i, -1], F[1, i, 1], F[1, i, -1] ** F[1, i, 1]}

(* alle möglichen Zustände bei num Plätzen *)
c1 = Flatten[Outer[NonCommutativeMultiply,
  Sequence@@Table[zustand[i], {i, 1, num}]]] //. sub1;

(* Anzahl der Zustände (Erzeuger) mit Spin up oder Spin down *)
npart[j_, x_] := Count[Flatten[{x /. NonCommutativeMultiply -> List}], F[1, _, j]]

(* Ersetzung in SparseArray *)
ReplaceSparse[ff_, rr_Rule] :=
  (ff /. rr) /. HoldPattern[SparseArray[xx_]] -> SparseArray@({xx} /. rr)

(* return alle Zustände (Erzeuger) mit Anzahl up und Anzahl down *)
kzu[up_, down_] := Select[c1, npart[1, #] == up && npart[-1, #] == down &]
```

### ■ Grundzustand

```
up = 1; down = 1;

(* Hamilton-Matrix *)
Clear[h];
<< "mathematica/theo-physik/data/7_hamiltonian.dat";

(* potentielle/kinetische Energie in MatrixForm*)
matH = ReplaceSparse[h[up, down], t -> 1.];
matU = DiagonalMatrix[Diagonal[matH]];
matT = matH - matU;

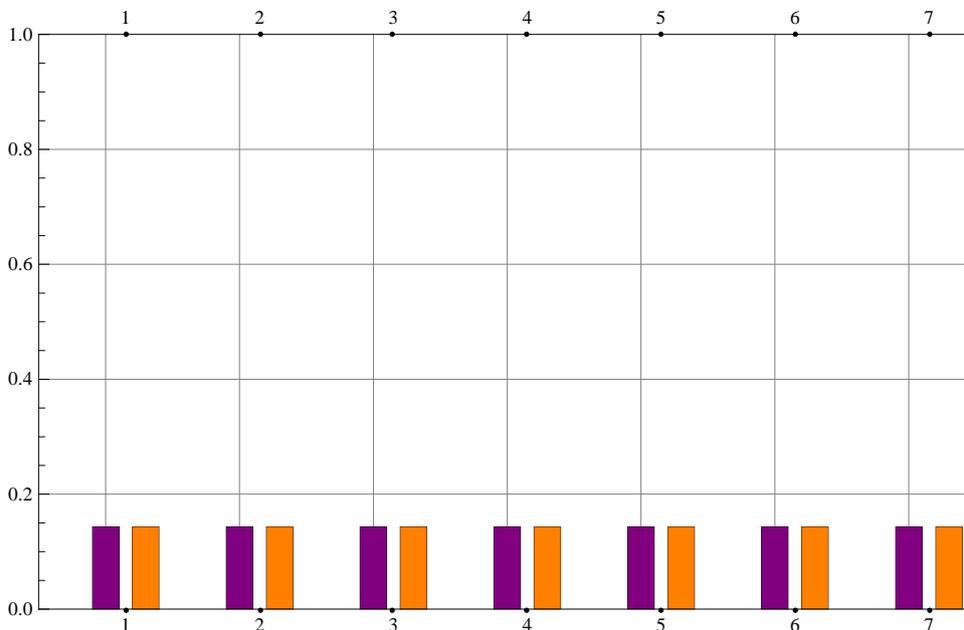
(* Eigenwerte und Eigenzustände *)
ew[period_, ww_] := Eigenvalues[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]];
ev[period_, ww_] := Eigenvectors[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]];

periodic = 1.;
pot = {1., 0., -8.};
Table[eigw[i] = ew[periodic, pot[[i]]], {i, 1, 3}];
Table[eigv[i] = ev[periodic, pot[[i]]], {i, 1, 3}];

(* Matrixdarstellung des Anzahloperators *)
Clear[nup, ndown];
<< "mathematica/theo-physik/data/7_nup.dat";
<< "mathematica/theo-physik/data/7_ndown.dat";

Needs["BarCharts`"]
```

```
(* Grundzustand: Zustand minimaler Energie *)
co = 1;
mi = Position[eigw[co], Min[eigw[co]]][[1, 1]];
BarChart[{Table[Conjugate[eigv[co][[mi]]].nup[i, up, down].eigv[co][[mi]], {i, 1, num}],
  Table[Conjugate[eigv[co][[mi]]].ndown[i, up, down].eigv[co][[mi]], {i, 1, num}]},
  PlotRange -> {{0.5, num + 0.5}, {0, 1}}, Axes -> {True, False},
  BarStyle -> {Purple, Orange}, BarSpacing -> 0.1, BarGroupSpacing -> 0.4,
  ImageSize -> 450, GridLines -> Automatic, Frame -> True]
```



```
Clear[grund];
grund = Chop[eigv[co][[mi]] * kzu[up, down];
grund = Plus@@grund;

(* Hinzufuegen eines Teilchens mit Spin up auf Platz 1 *)
startzustand = F[1, 1, 1] ** grund /. rules;
up = 2; down = 1;

basisneu = kzu[up, down];
startzustand = startzustand /. Plus -> List /. Times -> List;

koeffneu = Table[0, {i, 1, Length[basisneu]}];
Do[koeffneu[[Position[basisneu, startzustand[[i, 2]]][[1]]] = startzustand[[i, 1]],
  {i, 1, Length[startzustand]}];
koeffneu = koeffneu / Norm[koeffneu];
```

## ■ Zeitentwicklung

```
(* potentielle/kinetische Energie in MatrixForm*)
matH = ReplaceSparse[h[up, down], t -> 1.];
matU = DiagonalMatrix[Diagonal[matH]];
matT = matH - matU;

(* Eigenwerte und Eigenzustaende *)
ew[period_, ww_] := Eigenvalues[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]];
ev[period_, ww_] := Eigenvectors[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]];

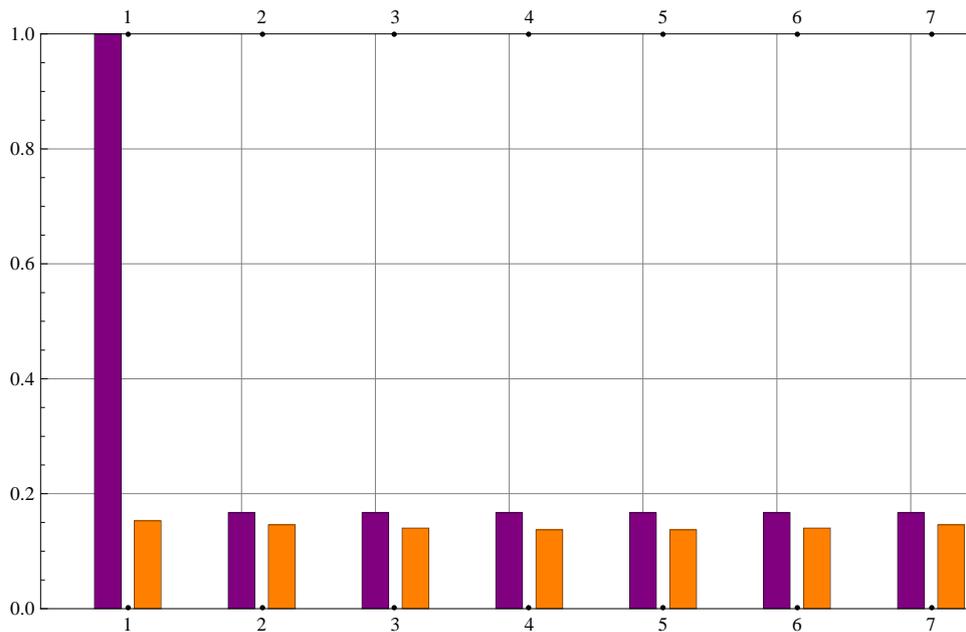
periodic = 1.;
pot = {1., 0., -8.};
Quiet[Table[eigw[i] = ew[periodic, pot[[i]]], {i, 1, 3}]];
Quiet[Table[eigv[i] = ev[periodic, pot[[i]]], {i, 1, 3}]];

```

```
(* Zeitentwicklungsvektor *)
eps = 0.0;
psil[t_, w_] := Sum[koeffneu.eigv[w][[i]] Exp[-I eigw[w][[i]] t (1 - I eps)] eigv[w][[i]],
  {i, 1, Length[kzu[up, down]]}
psi[t_, w_] := #/ Sqrt[Conjugate[#].#] &[psil[t, w]]

(* Dichte bzw. Energie je Platz *)
dup[t_, w_] := Table[Conjugate[psi[t, w]].nup[platz, up, down].psi[t, w], {platz, 1, num}]
ddown[t_, w_] :=
  Table[Conjugate[psi[t, w]].ndown[platz, up, down].psi[t, w], {platz, 1, num}]

BarChart[{Table[Conjugate[koeffneu].nup[i, up, down].koeffneu, {i, 1, num}],
  Table[Conjugate[koeffneu].ndown[i, up, down].koeffneu, {i, 1, num}]},
  PlotRange -> {{0.5, num + 0.5}, {0, 1}}, Axes -> {True, False},
  BarStyle -> {Purple, Orange}, BarSpacing -> 0.1, BarGroupSpacing -> 0.4,
  ImageSize -> 450, GridLines -> Automatic, Frame -> True]
```



```
tend = 10;
steps = 1000;

Timing[
  tabup = Table[Chop[dup[t, 1]], {t, 0, tend, tend/steps}];
  tabdown = Table[Chop[ddown[t, 1]], {t, 0, tend, tend/steps}];
  ladung = tabup + tabdown;
  spin = tabup - tabdown;]

{352.914, Null}

(*Save["ladung.dat",ladung];
Save["spin.dat",spin];*)
```

## ■ Animation

```
Clear[ladung, spin, ld, sp]
<< "mathematica/theo-physik/anim/7_ladung10-1000.dat";
<< "mathematica/theo-physik/anim/7_spin10-1000.dat";

(* Abzug Offset *)
ld = Chop[Table[ladung[[i]] - ladung[[1]], {i, 1, steps + 1}]];
sp = Chop[Table[spin[[i]] - spin[[1]], {i, 1, steps + 1}]];

(* Max/Min von Ladung und Spin wegen Darstellung*)
maxy = Max[sp, ld];
miny = Min[sp, ld];
```

```
ListAnimate[Table[ListLinePlot[{ld[[j]], sp[[j]]},  
  PlotStyle → {Dashed, Thick}, PlotRange → {{1, 7}, {miny, maxy}},  
  Axes → {True, False}, ImageSize → {800, 650}, GridLines → {None, Automatic},  
  FrameTicks → {Table[i, {i, 1, num}], {}}, Frame → {True, False}], {j, 1, steps + 1}],  
  AnimatorElements → {"ProgressSlider", "PlayPauseButton",  
    "StepLeftButton", "StepRightButton", "FasterSlowerButtons"},  
  AnimationRunning → False, AnimationRepetitions → 2]
```

---

## Dissipation

```
num = 8;
```

### ■ Zeitentwicklung

```
(* Hamilton-Matrix *)
Clear[h];
<< "christian/mathematica/theo-physik/data/8_hamiltonian.dat";

up = 2; down = 2;

(* Anfangszustand *)
anfang = 513;
psi0 = Table[0, {i, 1, Length[h[up, down]]}];
psi0[[anfang]] = 1;

(* Ersetzung in SparseArray *)
ReplaceSparse[ff_, rr_Rule] :=
  (ff /. rr) /. HoldPattern[SparseArray[xx_]] => SparseArray@@ ({xx} /. rr)

(* potentielle/kinetische Energie in MatrixForm*)
matH = ReplaceSparse[h[up, down], t -> 1.];
matU = DiagonalMatrix[Diagonal[matH]];
matT = matH - matU;

ew[period_, ww_] := Eigenvalues[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]]
ev[period_, ww_] := Eigenvectors[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]]

periodic = 0.;
pot = {0., 1., 100.};
Quiet[Table[eigw[i] = ew[periodic, pot[[i]]], {i, 1, 3}]];
Quiet[Table[eigv[i] = ev[periodic, pot[[i]]], {i, 1, 3}]];

(* Zeitentwicklungsvektor *)
eps = 0.1;
psil[t_, w_] := Sum[psi0.eigv[w][[i]] Exp[-I eigw[w][[i]] t (1 - I eps)] eigv[w][[i]],
  {i, 1, Length[h[up, down]]}
psi[t_, w_] := #/ Sqrt[Conjugate[#].#] &[psil[t, w]]
```

### ■ Animation

```
num = 8;
up = 2; down = 2;

(* Matrixdarstellung des Anzahloperators *)
Clear[nup, ndown]
<< "christian/mathematica/theo-physik/data/8_nup.dat";
<< "christian/mathematica/theo-physik/data/8_ndown.dat";

(* Dichte bzw. Energie je Platz *)
dup[t_, w_] := Table[Conjugate[psi[t, w]].nup[platz, up, down].psi[t, w], {platz, 1, num}]
ddown[t_, w_] :=
  Table[Conjugate[psi[t, w]].ndown[platz, up, down].psi[t, w], {platz, 1, num}]
dboth[t_, w_] := Table[Conjugate[psi[t, w]].nup[platz, up, down].
  ndown[platz, up, down].psi[t, w], {platz, 1, num}]
ekin[t_, w_] := Conjugate[psi[t, w]].ReplaceSparse[matT, y -> periodic].psi[t, w]
epot[t_, w_] := Conjugate[psi[t, w]].ReplaceSparse[matU, u -> pot[[w]]].psi[t, w]

(* Einstellungen fuer Animation *)
Needs["BarCharts`"]
labvec = {"U = 0.0", "U = 1.0", "U = 100.0"};
tend = 50;
steps = 500;
```

```

(* Berechnung von Dichte und Energie in Abhängigkeit von Platz,
Zeit und Wechselwirkung *)
Clear[density, corr, energy]
density[up, down, tend, steps] =
  Chop[Table[{dup[t, u], ddown[t, u]}, {u, 1, 3}, {t, 0, tend, tend/steps}]];
Save["8_density513dnp.dat", density];
corr[up, down, tend, steps] =
  Chop[Table[dboth[t, u] - dup[t, u] * ddown[t, u], {u, 1, 3}, {t, 0, tend, tend/steps}]];
Save["8_corr513dnp.dat", corr];
energy[up, down, tend, steps] =
  Chop[Table[{ekin[t, r], epot[t, r]}, {r, 1, 3}, {t, 0, tend, tend/steps}]];
Save["8_energy513dnp.dat", energy];

Clear[density, energy, corr]
<< "christian/mathematica/theo-physik/anim/8_density513d.dat";
<< "christian/mathematica/theo-physik/anim/8_energy513d.dat";
<< "christian/mathematica/theo-physik/anim/8_corr513d.dat";

mi = Table[Position[eigw[co], Min[eigw[co]]][[1, 1]], {co, 1, 3}];
minen = Min[energy[up, down, tend, steps]];

ListAnimate[
  Table[Column[Table[Row[{BarChart[density[up, down, tend, steps][[i, j]], PlotLabel →
    Style[labvec[[i]], Black, 20], PlotRange → {0, 1}, Axes → {True, False},
    BarStyle → {Purple, Orange}, BarSpacing → 0.1, BarGroupSpacing → 0.4,
    ImageSize → {350, 200}], BarChart[energy[up, down, tend, steps][[i, j]],
    PlotLabel → "energy", PlotRange → {minen, -minen}, {0, 3}, Axes → {True, False},
    BarStyle → {Red, Black}, ImageSize → {100, 100}, BarOrientation → Horizontal],
    BarChart[{Table[Conjugate[eigv[i][[mi[[i]]]]].nup[k, up, down].eigv[i][[mi[[i]]]],
      {k, 1, num}], Table[Conjugate[eigv[i][[mi[[i]]]].ndown[k, up, down].
      eigv[i][[mi[[i]]]], {k, 1, num}], PlotLabel → Style[labvec[[i]], Black, 20],
    PlotRange → {0, 1}, Axes → {True, False}, BarStyle → {Purple, Orange},
    BarSpacing → 0.1, BarGroupSpacing → 0.4, ImageSize → {350, 200}]], {i, 1, 3}],
  {j, 1, steps + 1}], AnimatorElements → {"ProgressSlider",
  "PlayPauseButton",
  "StepLeftButton",
  "StepRightButton"},
  AnimationRunning → False, AnimationRepetitions →
  2]

```

---

## on-site Korrelation

```
num = 8;
```

### ■ Zeitentwicklung

```
(* Hamilton-Matrix *)
Clear[h];
<< "christian/mathematica/theo-physik/data/8_hamiltonian.dat";

up = 1; down = 1;

(* Anfangszustand *)
anfang = 57;
psi0 = Table[0, {i, 1, Length[h[up, down]]}];
psi0[[anfang]] = 1;

(* Ersetzung in SparseArray *)
ReplaceSparse[ff_, rr_Rule] :=
  (ff /. rr) /. HoldPattern[SparseArray[xx_]] => SparseArray@@ ({xx} /. rr)

(* potentielle/kinetische Energie in MatrixForm*)
matH = ReplaceSparse[h[up, down], t -> 1.];
matU = DiagonalMatrix[Diagonal[matH]];
matT = matH - matU;

ew[period_, ww_] := Eigenvalues[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]]
ev[period_, ww_] := Eigenvectors[ReplaceSparse[ReplaceSparse[matH, y -> period], u -> ww]]

periodic = 0.;
pot = {0., 1., 100.};
Quiet[Table[eigw[i] = ew[periodic, pot[[i]]], {i, 1, 3}]];
Quiet[Table[eigv[i] = ev[periodic, pot[[i]]], {i, 1, 3}]];

(* Zeitentwicklungsvektor *)
eps = 0.0;
psil[t_, w_] := Sum[psi0.eigv[w][[i]] Exp[-I eigw[w][[i]] t (1 - I eps)] eigv[w][[i]],
  {i, 1, Length[h[up, down]]}
psi[t_, w_] := #/ Sqrt[Conjugate[#].#] &[psil[t, w]]
```

### ■ Animation

```
up = 1; down = 1;

(* Matrixdarstellung des Anzahloperators *)
Clear[nup, ndown]
<< "mathematica/theo-physik/data/8_nup.dat";
<< "mathematica/theo-physik/data/8_ndown.dat";

(* Dichte bzw. Energie je Platz *)
dup[t_, w_] := Table[Conjugate[psi[t, w]].nup[platz, up, down].psi[t, w], {platz, 1, num}]
ddown[t_, w_] :=
  Table[Conjugate[psi[t, w]].ndown[platz, up, down].psi[t, w], {platz, 1, num}]
dboth[t_, w_] := Table[Conjugate[psi[t, w]].nup[platz, up, down].
  ndown[platz, up, down].psi[t, w], {platz, 1, num}]
ekin[t_, w_] := Conjugate[psi[t, w]].ReplaceSparse[matT, y -> periodic].psi[t, w]
epot[t_, w_] := Conjugate[psi[t, w]].ReplaceSparse[matU, u -> pot[[w]]].psi[t, w]

(* Einstellungen fuer Animation *)
tend = 50;
steps = 500;
Needs["BarCharts`"]
labvec = {"U = 0.0", "U = 1.0", "U = 100.0"};
```

```

(* Berechnung von Dichte und Energie in Abhängigkeit von Platz,
Zeit und Wechselwirkung *)
Clear[density, energy, corr]
density[up, down, tend, steps] =
  Chop[Table[{dup[t, u], ddown[t, u]}, {u, 1, 3}, {t, 0, tend, tend/steps}]];
Save["8_density57.dat", density];
corr[up, down, tend, steps] =
  Chop[Table[dboth[t, u] - dup[t, u] * ddown[t, u], {u, 1, 3}, {t, 0, tend, tend/steps}]];
Save["8_corr57.dat", corr];
energy[up, down, tend, steps] =
  Chop[Table[{ekin[t, r], epot[t, r]}, {r, 1, 3}, {t, 0, tend, tend/steps}]];
Save["8_energy57.dat", energy];

Clear[density, energy, corr]
<< "mathematica/theo-physik/anim/8_density57.dat";
<< "mathematica/theo-physik/anim/8_energy57.dat";
<< "mathematica/theo-physik/anim/8_corr57.dat";

maxen = Max[energy[up, down, tend, steps]];
ListAnimate[
  Table[Column[Table[Row[{BarChart[density[up, down, tend, steps][[i, j]], PlotLabel →
    Style[labvec[[i]], Black, 20], PlotRange → {0, 1}, Axes → {True, False}, BarStyle →
    {Purple, Orange}, BarSpacing → 0.1, BarGroupSpacing → 0.4, ImageSize → {400, 215}],
    BarChart[energy[up, down, tend, steps][[i, j]], PlotLabel → "energy",
    PlotRange → {{-maxen, maxen}, {0, 3}}, Axes → {True, False}, BarStyle → {Red, Black},
    ImageSize → {200, 100}, BarOrientation → Horizontal]], {i, 1, 3}]],
  {j, 1, steps + 1}], AnimatorElements → {"ProgressSlider", "PlayPauseButton",
  "StepLeftButton", "StepRightButton"},
  AnimationRunning → False, AnimationRepetitions → 2]

```

## ■ Korrelationen

```

inww2 = 2;
inww3 = 3;
ListAnimate[
  Table[Row[{Column[{BarChart[density[up, down, tend, steps][[inww2, j]], PlotLabel →
    Style[labvec[[inww2]], Black, 20], PlotRange → {0, 1}, Axes → {True, False},
    BarStyle → {Purple, Orange}, BarSpacing → 0.1, BarGroupSpacing → 0.4,
    ImageSize → {400, 320}, GridLines → {None, Automatic}, Frame → {True, False}],
    BarChart[corr[up, down, tend, steps][[inww2, j]], PlotRange → {-0.15, 0.15},
    Axes → {True, False}, BarStyle → Black, BarSpacing → 0.1, BarGroupSpacing → 0.4,
    ImageSize → {400, 320}, GridLines → {None, Automatic}, Frame → {True, False}]}],
  Column[{BarChart[density[up, down, tend, steps][[inww3, j]], PlotLabel →
    Style[labvec[[inww3]], Black, 20], PlotRange → {0, 1}, Axes → {True, False},
    BarStyle → {Purple, Orange}, BarSpacing → 0.1, BarGroupSpacing → 0.4,
    ImageSize → {400, 320}, GridLines → {None, Automatic}, Frame → {True, False}],
    BarChart[corr[up, down, tend, steps][[inww3, j]], PlotRange → {-0.15, 0.15},
    Axes → {True, False}, BarStyle → Black, BarSpacing → 0.1, BarGroupSpacing → 0.4,
    ImageSize → {400, 320}, GridLines → {None, Automatic}, Frame → {True, False}]}]}],
  {j, 1, steps + 1}], AnimatorElements → {"ProgressSlider",
  "PlayPauseButton",
  "StepLeftButton", "StepRightButton"},
  AnimationRunning → False, AnimationRepetitions →
  2]

```