

Otto-von-Guericke-Universität
Magdeburg
Institut für Theoretische Physik

Forschungsbeleg

Berechnungen des Grundzustandes eines wechselwirkenden
Fermionensystems

von
Jörg Schulenburg

(geb. am 10.02.69 in Rostock)
(Student in der Matrikel Physik 1/89)

Betreuer: Prof. J. Richter
beendet am 13.10.1993

Inhaltsverzeichnis

1	Einleitung	2
1.1	Supraleitung	2
1.2	Hubbard-Modell	2
1.3	t-J-Modell	2
1.4	Heisenberg-Modell	3
1.5	Gegenstand der Arbeit	3
2	Grundsätzliche Herangehensweise	4
3	Analytische Resultate	4
3.1	2-Platz-Heisenberg-System	4
3.2	2-Platz-t-J-Modell	5
3.3	2-Platz-Hubbard-Modell	6
3.4	3-Spin-Heisenberg-System	9
3.5	3-Platz-t-J-Modell	12
3.6	4-Spin- J_1 - J_2 -Modell	14
4	Darstellung der Spin- durch Fermioperatoren	16
5	Vertauschungen	17
5.1	Teilchenzahlerhaltung	18
5.2	S^z -Erhaltung	19
5.3	S^2 -Erhaltung	20
6	Der Basisraum	22
6.1	Kodierung	23
6.2	Hashing	23
6.3	Zweidimensionale Linkodierung	25
7	Die Hamiltonmatrix	26
8	Exakte Diagonalisierung	26
8.1	Berechnungen zum 16-Platz t- J_1 - J_2 -Modell	28
8.2	Berechnungen zum 8-Platz-Hubbard-Modell	36
9	Zusammenfassung	38
10	Anhang	39
10.1	Programmbeschreibung	39
10.2	Programm	42

1 Einleitung

1.1 Supraleitung

Im Jahre 1911 wurde durch H.K.Onnes die Supraleitung beim Quecksilber entdeckt. Sie tritt bei Abkühlung des Quecksilbers unter 4.2K auf. In den darauffolgenden Jahren wurde diese Erscheinung auch bei anderen Materialien entdeckt. Eine entsprechende Theorie für diese Supraleiter wurde erst kurz vor 1960 von Bardeen, Cooper und Schrieffer entwickelt, und ist unter dem Namen BCS-Theorie bekannt. Ein wichtiges Kriterium für die technische Anwendung von supraleitenden Materialien ist die Sprungtemperatur. Deshalb wurde nach Materialien mit höheren Sprungtemperaturen gesucht. Ein wesentlicher Schritt in dieser Entwicklung ist die Entdeckung der Hochtemperatursupraleitung bei oxidischen Materialien. Bei ihnen liegen die Sprungtemperaturen im Bereich von 100K. Bei den oxidischen Keramiken (La_2CuO_4 , $\text{YBa}_2\text{Cu}_3\text{O}_{6.5}$) ist die unmittelbare Anwendung der BCS-Theorie [2] fraglich. Eine Möglichkeit, verschiedene Eigenschaften von Hochtemperatursupraleitern zu beschreiben, geben das Heisenberg- und das Hubbard-Modell [5]. Beide Modelle sollen im folgenden kurz beschrieben werden.

1.2 Hubbard-Modell

Das einfachste Modell, das sowohl ein metallisches als auch ein nichtmetallisches Regime erwarten läßt, ist ein Ein-Band-Hubbard-Modell [3]. Dieses wird durch folgenden Hamiltonoperator beschrieben:

$$\hat{H}_H = \hat{H}_t + \hat{H}_U \quad (1)$$

$$\hat{H} = \sum_{\langle ij \rangle s} t_{ij} (\hat{c}_{is}^+ \hat{c}_{js} + \hat{c}_{js}^+ \hat{c}_{is}) + \sum_i U_i \hat{n}_{iu} \hat{n}_{id} \quad (2)$$

Im einfachsten Fall sind $t_{ij} = t$ ¹ und $U_i = U$. Der Term H_t stellt dabei den Hoppingterm (kinetische Energie) und H_U die Coulombabstoßung (potentielle Energie) dar. Die Elektronen können längs der Bonds $\langle ij \rangle$ (nächste Nachbarn) hüpfen. Im Hoppingterm wird über alle diese Bonds und die Spinausrichtung $s = (u, d)$ einfach summiert. U repräsentiert die Coulombabstoßung zweier Elektronen mit verschiedenem Spin im selbem Orbital. Befindet sich an jedem Platz ein Elektron, so wird verständlich, daß bei großem U viel Energie aufgewendet werden muß, um ein Elektron von einem Platz zum Nachbarplatz zu bewegen. Woraus wiederum nichtmetallisches Verhalten resultiert. Das reine La_2CuO_4 zeigt zum Beispiel eindeutig nichtmetallisches Verhalten [2]. Wesentlich für das Verhalten des Modells ist hier das Verhältnis der beiden Parameter t/U und die Elektronenkonzentration.

1.3 t-J-Modell

Das t-J-Modell ist herleitbar als Grenzfall des Ein-Band-Hubbard-Modells für große Coulombabstoßung ($U/t \gg 1$) [5].

$$\hat{H} = \hat{H}_t + \hat{H}_J \quad (3)$$

¹oft wird das Hoppingintegral t auch mit negativem Vorzeichen versehen

Dabei stellt der J-Term die antiferromagnetische Wechselwirkung zwischen den $\text{Cu}^{2+}d$ Elektronen (bzw. Löchern) innerhalb der Kupfer-Sauerstoff-Ebenen und der t-Term das Hüpfen der Elektronen bei vorhandenen Löchern dar.

$$\hat{H} = \sum_{ijs} t_{ij}(\hat{c}_{is}^+ \hat{c}_{js} + \hat{c}_{js}^+ \hat{c}_{is}) + \sum_{ij} J_{ij} \hat{S}_i \hat{S}_j \quad (4)$$

Summiert wird über alle Plätze i, j und die Spinausrichtung $s = (u, d)$. Da Zustände, bei denen sich zwei Elektronen an einem Platz befinden, ausgeschlossen werden sollen, werden die obigen Fermioperatoren \hat{c} und \hat{c}^+ durch neue Operatoren $\hat{\tilde{c}}$ und $\hat{\tilde{c}}^+$ ersetzt. In einem später folgenden Abschnitt wird dieser Schritt noch genauer erläutert. In einigen Arbeiten [5][7] [8][9] wird ein etwas anderer Hamiltonoperator verwendet.

$$\hat{H} = \sum_{ijs} t_{ij}(\hat{c}_{is}^+ \hat{c}_{js} + \hat{c}_{js}^+ \hat{c}_{is}) + \sum_{ij} J_{ij}(\hat{S}_i \hat{S}_j - \frac{1}{4} \hat{n}_i \hat{n}_j) \quad (5)$$

Der letzte Term liefert bei Modellen ohne und mit einem Loch jedoch nur eine Energieverschiebung [5]. In den meisten Fällen sind die Hoppingintegrale t_{ij} und die Austauschintegrale J_{ij} für nächste Nachbarn gleich und sonst Null. Dadurch vereinfacht sich der Hamiltonoperator erheblich.

1.4 Heisenberg-Modell

Das Heisenbergmodell ist ein Spezialfall des t-J-Modells. Es vereinfacht dieses durch Vernachlässigung des Hoppingtermes. Der allgemeine Heisenbergoperator hat dann folgendes Aussehen.

$$\hat{H} = \sum_{ij} J_{ij} \hat{S}_i \hat{S}_j \quad (6)$$

Berücksichtigt man nur nächste und übernächste Wechselwirkung, so bleibt letztendlich ein J_1 - J_2 -Modell übrig.

$$\hat{H} = J_1 \sum_{\langle ij \rangle} \hat{S}_i \hat{S}_j + J_2 \sum_{\langle\langle ij \rangle\rangle} \hat{S}_i \hat{S}_j \quad (7)$$

Dieses Modell kann zur Untersuchung magnetischer Eigenschaften von Hochtemperatursupraleitern herangezogen werden.

1.5 Gegenstand der Arbeit

In dieser Arbeit soll ein Verfahren beschrieben werden, mit dem sich die Grundzustände für kleine zweidimensionale² Systeme (4 mal 4 Gitter) und verschiedene Korrelationsfunktionen berechnen lassen. Im Ergebnis dieser Arbeit soll ein Programmpaket vorliegen, mit dem solche Systeme für die oben beschriebenen Modelle berechnet werden können. Dabei soll in dieser Arbeit im wesentlichen auf methodische Probleme eingegangen werden, die beim exakten Diagonalisieren großer Matrizen bestehen. Zur Diagonalisierung der Matrizen wird

²Vermutlich sind die CuO_2 -Ebenen das wesentliche gemeinsame Strukturmerkmal der Hochtemperatursupraleiter [2]

das Lanczos-Verfahren genutzt. Im Anschluß sollen mit dem entwickelten Programmpaket Ergebnisse verschiedener Veröffentlichungen [4][5][6][8][9] verifiziert und verschiedene Probleme, die bei der Nutzung des Programmpaketes auftreten, erörtert werden. Außerdem sollen die Grundzustände und Korrelationsfunktionen in Abhängigkeit vom Parameter J_2 im t - J_1 - J_2 -Modell für ein Loch berechnet werden. Diese werden dann mit denen ohne Loch [4] verglichen.

2 Grundsätzliche Herangehensweise

Es sollen die Eigenwerte und Eigenfunktionen zum Operator \hat{H} bestimmt werden. Wie später bewiesen wird, vertauschen die Operatoren $\hat{N} = \sum_{is} \hat{n}_{is}$, $\hat{S}^2 = (\sum_i \hat{S}_i^z)^2$, $\hat{S}^z = \sum_i \hat{S}_i^z$ und \hat{H} untereinander. Die Lösungen sind im Produktraum der Einzelspinfunktionen zu suchen.

$$\hat{H}|\Psi\rangle = E|\Psi\rangle \quad \text{mit} \quad |\Psi\rangle = \sum_n a_n |\psi_n\rangle \quad \Rightarrow \quad \sum_n \hat{H} a_n |\psi_n\rangle = \sum_n E a_n |\psi_n\rangle \quad (8)$$

Nach Multiplikation von links mit $\langle\psi_m|$ ergibt sich wegen $\langle\psi_n|\psi_m\rangle = \delta_{nm}$

$$\sum_n \langle\psi_m|\hat{H}|\psi_n\rangle a_n = E a_m \quad \text{mit} \quad H_{mn} = \langle\psi_m|\hat{H}|\psi_n\rangle \quad (9)$$

$$\underline{\underline{H}}\underline{\underline{a}} = E\underline{\underline{a}} \quad \text{bzw.} \quad (\underline{\underline{H}} - E\underline{\underline{I}})\underline{\underline{a}} = 0 \quad \text{mit} \quad \det(\underline{\underline{H}} - E\underline{\underline{I}}) \stackrel{!}{=} 0 \quad (10)$$

Um die Matrix $\underline{\underline{H}}$ aufzustellen, aus der durch Diagonalisierung die Eigenwerte E und die Koeffizienten $\underline{\underline{a}}$ bestimmt werden, muß eine geeignete Methode gefunden werden, die Basisvektoren $|\psi_n\rangle$ darzustellen. Bei einem 16 Platzsystem ohne Doppelbesetzung und Löcher kann jede Komponente des Basisvektors als 16-bit-Zahl dargestellt werden. Jedes Bit gibt dabei die Ausrichtung des Spins (0=up oder 1=down) an. Diese Darstellungsform ist besonders günstig. Wendet man die Operatoren \hat{S}_i^z , \hat{S}_i^+ oder \hat{S}_i^- auf diese Zustände an, brauchen nur das entsprechende Bit Nummer i geprüft bzw. geändert werden. Will man auch Löcher (und doppelbesetzte Plätze im Hubbardmodell) zulassen, benötigt man eine 32-bit-Zahl. Jedes Bit steht dann für einen Platz und eine der möglichen Ausrichtungsmöglichkeiten up oder down des Spins. Im Abschnitt 6 soll diese Vorgehensweise noch genauer erläutert werden. Da die meisten Rechner (Programmiersprachen) derzeit ganze Zahlen bis 32-bit verarbeiten, ist die Berechnung von 16-Platz-Systemen programmtechnisch relativ leicht handhabbar.

3 Analytische Resultate

Zur Einführung und Illustration der oben beschriebenen Herangehensweise sollen nun kleine Systeme analytisch berechnet werden. Besonders einfach ist die Berechnung eines 2-Spin-Heisenberg-Modells.

3.1 2-Platz-Heisenberg-System

Der Hamiltonoperator hat für diese System eine besonders einfache Form.

$$\hat{H} = J\hat{S}_1\hat{S}_2 \quad (11)$$

$$\hat{S}_i^z \hat{S}_j^z = \hat{S}_i^x \hat{S}_j^x + \hat{S}_i^y \hat{S}_j^y + \hat{S}_i^z \hat{S}_j^z = \frac{1}{2}(\hat{S}_i^+ \hat{S}_j^- + \hat{S}_i^- \hat{S}_j^+) + \hat{S}_i^z \hat{S}_j^z \quad (12)$$

$$\text{mit } \hat{S}^+ := \hat{S}^x + i\hat{S}^y \quad \text{und} \quad \hat{S}^- := \hat{S}^x - i\hat{S}^y$$

Im folgenden soll die Matrix $\underline{\underline{H}}$ aufgestellt werden, und daraus Eigenwerte E und die Koeffizienten $\underline{\underline{a}}$ bestimmt werden. $\Psi_i = \underline{\underline{a}}_i \underline{\underline{\psi}}$ sind die gesuchten Eigenfunktionen. Mit

$$|m_2 m_1\rangle \quad m_i \in (u, d) \quad (13)$$

existieren 2^2 Basiskonfigurationen.

$$|\underline{\underline{\psi}}\rangle = \begin{bmatrix} |dd\rangle \\ |du\rangle \\ |ud\rangle \\ |uu\rangle \end{bmatrix} \quad \underline{\underline{H}} = \frac{J}{4}\hbar^2 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Man erhält für die obige Matrix die 4 Eigenwerte $E_{1..3} = J\hbar^2/4$ und $E_4 = -3J\hbar^2/4$. Der Eigenvektor zu den 4 Eigenwerten ist leicht zu ermitteln. Er ist von 4 Parametern abhängig, die durch die Normierungsbedingungen bestimmt werden.

$$\underline{\underline{a}}_1 = [1, 0, 0, 0]^T \quad \underline{\underline{a}}_2 = [0, 1, 1, 0]^T/\sqrt{2} \quad \underline{\underline{a}}_3 = [0, 0, 0, 1]^T \quad \underline{\underline{a}}_4 = [0, 1, -1, 0]^T/\sqrt{2}$$

In der folgenden Tabelle sind zu den jeweiligen Eigenfunktionen die Spinquantenzahl s , die Magnetquantenzahl m und einige Erwartungswerte aufgeführt. \hbar wird, wenn nicht weiter aufgeführt, 1 gesetzt.

$$\langle \hat{S}^z \hat{S}^z \rangle = s(s+1)\hbar^2 \quad \langle \hat{S}^z \rangle = m\hbar$$

	EF	E	s	m	$\langle \hat{S}_i^z \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$
Ψ_1	$ dd\rangle$	$J/4$	1	-1	-1/2	1/4	1/4
Ψ_2	$(du\rangle + ud\rangle)/\sqrt{2}$	$J/4$	1	0	0	-1/4	1/4
Ψ_3	$ uu\rangle$	$J/4$	1	+1	1/2	1/4	1/4
Ψ_4	$(du\rangle - ud\rangle)/\sqrt{2}$	$-3J/4$	0	0	0	-1/4	-3/4

$$\langle \hat{S}_i^x \rangle = \langle \hat{S}_i^y \rangle = 0 \quad \langle \hat{S}_1^x \hat{S}_2^x \rangle = \langle \hat{S}_1^y \hat{S}_2^y \rangle = (\langle \hat{S}_1^z \hat{S}_2^z \rangle - \langle \hat{S}_1^z \hat{S}_2^z \rangle)/2$$

$$\langle \hat{S}_i^x \hat{S}_i^x \rangle = \langle \hat{S}_i^y \hat{S}_i^y \rangle = \langle \hat{S}_i^z \hat{S}_i^z \rangle = \hbar^2/4$$

Für $J > 0$ erhält man den antiferromagnetischen Grundzustand. Der ferromagnetische Grundzustand für $J < 0$ ist 3-fach entartet. Die möglichen Zustandsfunktionen sind unabhängig vom Parameter J .

3.2 2-Platz-t-J-Modell

Ebenfalls recht einfach ist das 2-Platz-t-J-Modell zu lösen. Doppeltbesetzte Plätze werden ausgeschlossen. Folglich existieren 3^2 Basiszustände. Diese werden auf folgende Art und Weise dargestellt.

$$|m_2 m_1\rangle \quad m_i = (., u, d) \quad (15)$$

Der Punkt steht für nichtbesetzten Platz. Der Hamiltonoperator wird wie folgt definiert.

$$\hat{H} = t(\hat{c}_{1u}^+ \hat{c}_{2u} + \hat{c}_{2u}^+ \hat{c}_{1u} + \hat{c}_{1d}^+ \hat{c}_{2d} + \hat{c}_{2d}^+ \hat{c}_{1d}) + J(\hat{S}_1^z \hat{S}_2^z) \quad (16)$$

Zur Bestimmung der Eigenfunktionen und Eigenwerte sind folgende Matrizen zu diagonalisieren. Die Basiszustände $|\cdot\rangle$, $|uu\rangle$, $|dd\rangle$ sind bereits Eigenzustände zu \hat{H} .

$$\begin{aligned} \begin{bmatrix} |u\rangle \\ |u\rangle \end{bmatrix}, \quad \begin{bmatrix} |d\rangle \\ |d\rangle \end{bmatrix} &\Rightarrow \begin{bmatrix} 0 & t \\ t & 0 \end{bmatrix} \\ \begin{bmatrix} |ud\rangle \\ |du\rangle \end{bmatrix} &\Rightarrow \frac{J}{4} \begin{bmatrix} -1 & 2 \\ 2 & -1 \end{bmatrix} \end{aligned}$$

Die Lösung führt zu folgenden Eigenzuständen und Eigenwerten von \hat{H} . Zur Vereinfachung wurde $\hbar = 1$ gesetzt.

	EF	E	N	s	m	$\langle \hat{c}_{1u}^+ \hat{c}_{2u} \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$
Ψ_1	$ \cdot\rangle$	0	0	0	0	0	0	0
Ψ_2	$(u\rangle + u\rangle)/\sqrt{2}$	t	1	1/2	1/2	1/2	0	0
Ψ_3	$(u\rangle - u\rangle)/\sqrt{2}$	$-t$	1	1/2	1/2	-1/2	0	0
Ψ_4	$(d\rangle + d\rangle)/\sqrt{2}$	t	1	1/2	-1/2	0	0	0
Ψ_5	$(d\rangle - d\rangle)/\sqrt{2}$	$-t$	1	1/2	-1/2	0	0	0
Ψ_6	$ uu\rangle$	$J/4$	2	1	1	0	1/4	1/4
Ψ_7	$ dd\rangle$	$J/4$	2	1	-1	0	1/4	1/4
Ψ_8	$(ud\rangle + du\rangle)/\sqrt{2}$	$J/4$	2	1	0	0	-1/4	1/4
Ψ_9	$(ud\rangle - du\rangle)/\sqrt{2}$	$-3J/4$	2	0	0	0	-1/4	-3/4

Bei Besetzung jedes Platzes mit einem Fermion ($N = 2$) ist bei $J > 0$ Ψ_9 Grundzustand dieses 2-Platz-Systems. Das Hopping kann vollständig vernachlässigt werden. Die Physik entspricht vollständig der des 2-Platz-Heisenberg-Systems. Erst wenn ein Loch $N = 1$ angenommen wird, kommt das Hopping zum tragen. Der Grundzustand ist dann für $t < 0$ und $m = 1/2$ Ψ_2 und für $m = -1/2$ Ψ_4 . Das Hopping erniedrigt die Energie des Systems. Die Physik ist bei diesem Modell unabhängig von J .

3.3 2-Platz-Hubbard-Modell

Zur Darstellung der Zustandsfunktionen soll die "diskrete" Fock-Darstellung nach [14] verwendet werden. Jeder Platz kann a) leer, b) durch ein Elektron mit Spin-Up, c) ein Elektron mit Spin-Down oder d) durch 2 Elektronen mit Spin-Up und Spin-Down besetzt sein. Es existieren also 4^2 Zustände. Oder man betrachtet die 4 Einzelzustände n_1) Spin-Up an Platz 1, n_2) Spin-Down an Platz 1, n_3) Spin-Up an Platz 2 und n_4) Spin-Down an Platz 2 die jeweils durch ein Elektron besetzt (1) oder frei (0) sind, was 2^4 Basiszustände ergibt. Jeder Basiszustand kann also in folgender Besetzungszahldarstellung geschrieben werden.

$$|n_4 n_3 n_2 n_1\rangle \quad n_i = 0, 1 \quad (\text{Fermionen}) \quad (17)$$

Der Hamiltonoperator wird wie folgt definiert.

$$\hat{H} = t(\hat{c}_1^+ \hat{c}_3 + \hat{c}_3^+ \hat{c}_1 + \hat{c}_2^+ \hat{c}_4 + \hat{c}_4^+ \hat{c}_2) + U(\hat{n}_1 \hat{n}_2 + \hat{n}_3 \hat{n}_4) \quad (18)$$

Zur Bestimmung der Eigenfunktionen und Eigenwerte sind folgende Matrizen zu diagonalisieren. Die Basiszustände $|0000\rangle, |0101\rangle, |1010\rangle, |1111\rangle$ sind bereits Eigenzustände zu \hat{H} .

$$\begin{bmatrix} |1100\rangle \\ |0110\rangle \\ |1001\rangle \\ |0011\rangle \end{bmatrix} \Rightarrow \begin{bmatrix} U & -t & t & 0 \\ -t & 0 & 0 & -t \\ t & 0 & 0 & t \\ 0 & -t & t & U \end{bmatrix}$$

$$\begin{bmatrix} |1000\rangle \\ |0010\rangle \end{bmatrix}, \begin{bmatrix} |0100\rangle \\ |0001\rangle \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & t \\ t & 0 \end{bmatrix}$$

$$\begin{bmatrix} |1110\rangle \\ |1011\rangle \end{bmatrix}, \begin{bmatrix} |1101\rangle \\ |0111\rangle \end{bmatrix} \Rightarrow \begin{bmatrix} U & -t \\ -t & U \end{bmatrix}$$

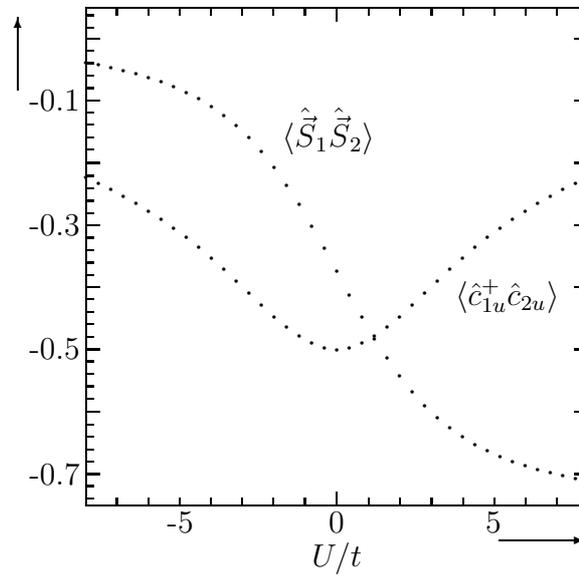
Die Lösung führt zu folgenden Eigenzuständen und Eigenwerten von \hat{H} . Zur Vereinfachung wurde $\hbar = 1$ gesetzt.

	EF	E	N	s	m	$\langle \hat{c}_{1u}^+ \hat{c}_{2u} \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$	$\langle \vec{\hat{S}}_1 \vec{\hat{S}}_2 \rangle$
Ψ_1	$ 0000\rangle$	0	0	0	0	0	0	0
Ψ_2	$(1000\rangle + 0010\rangle)/\sqrt{2}$	t	1	1/2	-1/2	0	0	0
Ψ_3	$(1000\rangle - 0010\rangle)/\sqrt{2}$	$-t$	1	1/2	-1/2	0	0	0
Ψ_4	$(0100\rangle + 0001\rangle)/\sqrt{2}$	t	1	1/2	1/2	1/2	0	0
Ψ_5	$(0100\rangle - 0001\rangle)/\sqrt{2}$	$-t$	1	1/2	1/2	-1/2	0	0
Ψ_6	$ 1010\rangle$	0	2	1	-1	0	1/4	1/4
Ψ_7	$ 0101\rangle$	0	2	1	1	0	1/4	1/4
Ψ_8	$(0110\rangle + 1001\rangle)/\sqrt{2}$	0	2	1	0	0	-1/4	1/4
Ψ_9	$(1100\rangle - 0011\rangle)/\sqrt{2}$	U	2	0	0	0	0	0
Ψ_{10}	$(1100\rangle + a 0110\rangle - a 1001\rangle + 0011\rangle)/\sqrt{2a^2 + 2}$	$2tb$	2	0	0	$-\frac{a}{a^2+1}$	$-\frac{1}{4}\frac{a^2}{a^2+1}$	$-\frac{3}{4}\frac{a^2}{a^2+1}$
Ψ_{11}	$(1100\rangle + b 0110\rangle - b 1001\rangle + 0011\rangle)/\sqrt{2b^2 + 2}$	$2ta$	2	0	0	$-\frac{b}{b^2+1}$	$-\frac{1}{4}\frac{b^2}{b^2+1}$	$-\frac{3}{4}\frac{b^2}{b^2+1}$
Ψ_{12}	$(1110\rangle + 1011\rangle)/\sqrt{2}$	$U - t$	3	1/2	-1/2	-1/2	0	0
Ψ_{13}	$(1110\rangle - 1011\rangle)/\sqrt{2}$	$U + t$	3	1/2	-1/2	1/2	0	0
Ψ_{14}	$(1101\rangle + 0111\rangle)/\sqrt{2}$	$U - t$	3	1/2	1/2	0	0	0
Ψ_{15}	$(1101\rangle - 0111\rangle)/\sqrt{2}$	$U + t$	3	1/2	1/2	0	0	0
Ψ_{16}	$ 1111\rangle$	$2U$	4	0	0	0	0	0

$$\text{mit } a = \frac{U + \sqrt{16t^2 + U^2}}{4t}, \quad \text{und } b = \frac{U - \sqrt{16t^2 + U^2}}{4t}$$

Physikalisch interessant sind die Zustände mit der Besetzungszahl $N = 2$. Für alle U und t ist Ψ_{10} Grundzustand des Systems. Dieser Zustand ist mit $\langle \vec{\hat{S}}_1 \vec{\hat{S}}_2 \rangle < 0$ antiferromagnetisch. Im

Grenzfall $U \rightarrow \infty$ stimmt diese Korrelationsfunktion mit der des 2-Platz-Heisenbergsystems ($-3/4$) überein. Ebenso verschwindet das Hopping $\langle \hat{c}_1^+ \hat{c}_2 \rangle$ mit $U \rightarrow \infty$. Im Diagramm sind 2 Korrelationsfunktionen des Grundzustandes Ψ_{10} für $t > 0$ grafisch dargestellt. Für $t < 0$ wird die Korrelationsfunktion $\langle \hat{c}_{1u}^+ \hat{c}_{2u} \rangle$ an der Geraden $y = 0$ gespiegelt.



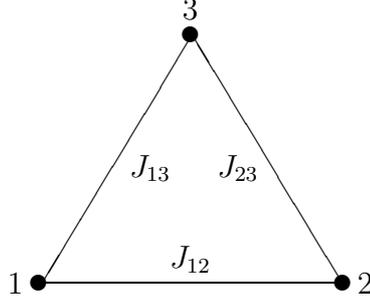


Abbildung 1: Darstellung des 3-Spin-Heisenberg-System

3.4 3-Spin-Heisenberg-System

Das 3-Spin-Modell liefert ein einfaches Beispiel für Frustration. Für dieses System wird folgender Hamiltonoperator verwendet:

$$\hat{H} = \sum_{\substack{i,j=1 \\ i < j}}^3 J_{ij} \hat{S}_i \hat{S}_j = J_{12} \hat{S}_1 \hat{S}_2 + J_{13} \hat{S}_1 \hat{S}_3 + J_{23} \hat{S}_2 \hat{S}_3 \quad (19)$$

Wieder wird wie im obigem Abschnitt erläutert vorgegangen. Basisvektor und Hamiltonmatrix haben dann folgendes Aussehen.

$$\underline{\psi} = \begin{bmatrix} |ddd\rangle \\ |ddu\rangle \\ |dud\rangle \\ |udd\rangle \\ |duu\rangle \\ |udu\rangle \\ |uud\rangle \\ |uuu\rangle \end{bmatrix} \quad \underline{H} = \frac{\hbar^2}{4} \begin{bmatrix} J_s & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2J_{12}-J_s & 2J_{23} & 2J_{13} & 0 & 0 & 0 & 0 \\ 0 & 2J_{23} & 2J_{13}-J_s & 2J_{12} & 0 & 0 & 0 & 0 \\ 0 & 2J_{13} & 2J_{12} & 2J_{23}-J_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2J_{23}-J_s & 2J_{12} & 2J_{13} & 0 \\ 0 & 0 & 0 & 0 & 2J_{12} & 2J_{13}-J_s & 2J_{23} & 0 \\ 0 & 0 & 0 & 0 & 2J_{13} & 2J_{23} & 2J_{12}-J_s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & J_s \end{bmatrix}$$

$$J_s := J_{12} + J_{13} + J_{23}, \quad J_q := \sqrt{J_{12}^2 + J_{13}^2 + J_{23}^2 - J_{12}J_{13} - J_{12}J_{23} - J_{13}J_{23}}$$

Man erhält die folgenden 8 Eigenwerte für die obige Matrix.

$$E_{1..4} = J_s \frac{\hbar^2}{4}, \quad E_{5,6} = (-J_s + 2J_q) \frac{\hbar^2}{4}, \quad E_{7,8} = (-J_s - 2J_q) \frac{\hbar^2}{4}$$

Der Eigenvektor zu den ersten 4 Eigenwerten ist leicht zu bestimmen. Er ist von 4 Parametern p abhängig, die durch die Normierungsbedingungen bestimmt werden.

$$\mathbf{a}_{1..4} = \begin{bmatrix} p_1 \\ p_2 \\ p_2 \\ p_2 \\ p_3 \\ p_3 \\ p_3 \\ p_4 \end{bmatrix} \Rightarrow \mathbf{a}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}_3 = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \mathbf{a}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Die Berechnung der anderen Eigenvektoren gestaltet sich wegen der komplizierteren Abhängigkeit der Eigenwerte von den J 's schwieriger. Insbesondere müssen bei den auftretenden Polynomdivisionen die Nulldivision beachtet werden. Dieser Fall tritt bei $J_{12} = J_{13} = J_{23}$ auf, und wird deshalb zunächst ausgeschlossen.

$$\begin{aligned} J_a &:= +J_{12}J_{23} - J_{13}^2 + J_{13}J_q & J_b &:= +J_{13}J_{23} - J_{12}^2 + J_{12}J_q \\ J_c &:= -J_{12}J_{13} + J_{23}^2 + J_{23}J_q & J_d &:= -J_{13}J_{23} + J_{12}^2 + J_{12}J_q \end{aligned}$$

$$\mathbf{a}'_5 = p_5 \begin{bmatrix} 0 \\ J_a \\ J_b \\ -J_a - J_b \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}'_6 = p_6 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ J_a \\ J_b \\ -J_a - J_b \\ 0 \end{bmatrix} \quad \mathbf{a}'_7 = p_7 \begin{bmatrix} 0 \\ J_c \\ -J_c - J_d \\ J_d \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}'_8 = p_8 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ J_c \\ -J_c - J_d \\ J_d \\ 0 \end{bmatrix}$$

$$p_{5,6} = \frac{1}{\sqrt{2(J_a^2 + J_a J_b + J_b^2)}}, \quad p_{7,8} = \frac{1}{\sqrt{2(J_c^2 + J_c J_d + J_d^2)}}$$

Im Fall $J_{12}=J_{13}=J_{23}=J$ ist $J_q = 0$ und die Eigenwerte $E_{5,6}$ und $E_{7,8}$ fallen zusammen. Zusätzlich vertauscht \hat{H} mit $\hat{E}_{123} = \hat{S}_1(\hat{S}_2 \times \hat{S}_3)$. Die Eigenvektoren wurden so gewählt, daß alle $\langle \hat{S}_i^z \rangle$ gleich sind. Für $J > 0$ tritt im Grundzustand zusätzliche Entartung bezüglich der Quantenzahlen s und m auf. Das System ist frustriert.

$$\mathbf{a}_5 = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 \\ 1 \\ k \\ k^* \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}_6 = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ k \\ k^* \\ 0 \end{bmatrix} \quad \mathbf{a}_7 = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 \\ 1 \\ k^* \\ k \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{a}_8 = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ k^* \\ k \\ 0 \end{bmatrix}$$

$$k := \exp\left(i\frac{2\pi}{3}\right) = \frac{-1 + \sqrt{3}i}{2} \quad k^* := \exp\left(-i\frac{2\pi}{3}\right) = \frac{-1 - \sqrt{3}i}{2}$$

In der folgenden Tabelle sind zu den jeweiligen Eigenfunktionen die Spinquantenzahl s , die Magnetquantenzahl m und diverse Erwartungswerte aufgeführt. Im folgendem gilt allgemein $i, j = 1..3$, $i < j$ und $S = \hat{S}$.

	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ_5	Ψ_6	Ψ_7	Ψ_8	
$\langle \hat{H} \rangle$	$3J$				$-3J$				$\hbar^2/4$
s	3/2	3/2	3/2	3/2	1/2	1/2	1/2	1/2	
m	-3/2	-1/2	1/2	3/2	-1/2	1/2	-1/2	1/2	
$\langle S_i^z \rangle$	-1	-1/3	1/3	1	-1/3	1/3	-1/3	1/3	$\hbar/2$
$\langle S_i^z S_j^z \rangle$	1	-1/3	-1/3	1	-1/3	-1/3	-1/3	-1/3	$\hbar^2/4$
$\langle S_i^x S_j^x \rangle$	0	2/3	2/3	0	-1/3	-1/3	-1/3	-1/3	$\hbar^2/4$
$\langle S_i^y S_j^y \rangle$	0	2/3	2/3	0	-1/3	-1/3	-1/3	-1/3	$\hbar^2/4$
$\langle \vec{S}_i \vec{S}_j \rangle$	1	1	1	1	-1	-1	-1	-1	$\hbar^2/4$
$\langle S_1^z S_2^z S_3^z \rangle$	-1	1	-1	1	1	-1	1	-1	$\hbar^3/8$
$\langle S_1^x S_2^y S_3^z \rangle$	0	0	0	0	$-1/\sqrt{3}$	$1/\sqrt{3}$	$1/\sqrt{3}$	$-1/\sqrt{3}$	$\hbar^3/8$
$\langle \hat{E}_{123} \rangle$	0	0	0	0	$-2\sqrt{3}$	$2\sqrt{3}$	$2\sqrt{3}$	$-2\sqrt{3}$	$\hbar^3/8$

$$\langle S_i^x \rangle = \langle S_i^y \rangle = \langle S_1^x S_2^x S_3^x \rangle = \langle S_1^y S_2^y S_3^y \rangle = 0 \quad \langle S_i^x S_i^x \rangle = \langle S_i^y S_i^y \rangle = \langle S_i^z S_i^z \rangle = \hbar^2/4$$

Die folgende Tabelle enthält die Erwartungswerte für den Fall verschiedener J 's. Für q kann x , y oder z eingesetzt werden.

	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ'_5	Ψ'_6	Ψ'_7	Ψ'_8	
$\langle H \rangle$	J_s				$-J_s + 2J_q$		$-J_s - 2J_q$		$\hbar^2/4$
s	3/2	3/2	3/2	3/2	1/2	1/2	1/2	1/2	
m	-3/2	-1/2	1/2	3/2	-1/2	1/2	-1/2	1/2	
$\langle S_1^z \rangle$	-1	-1/3	1/3	1	k_1	$-k_1$	$-k'_2$	k'_2	$\hbar/2$
$\langle S_2^z \rangle$	-1	-1/3	1/3	1	$-k_2$	k_2	k'_1	$-k'_1$	$\hbar/2$
$\langle S_3^z \rangle$	-1	-1/3	1/3	1	$-k_3$	k_3	$-k'_3$	k'_3	$\hbar/2$
$\langle S_1^q S_1^q \rangle$	1	1	1	1	k_1	k_3	$-k'_2$	k'_3	$\hbar^2/4$
$\langle S_2^q S_2^q \rangle$	1	1	1	1	$-k_2$	k_2	k'_1	$-k'_1$	$\hbar^2/4$
$\langle S_3^q S_3^q \rangle$	1	1	1	1	$-k_3$	$-k_1$	$-k'_3$	k'_2	$\hbar^2/4$
$\langle S_1^q S_2^q \rangle$					$-k_3$	k_1	$-k'_3$	$-k'_2$	$\hbar^2/4$
$\langle S_1^q S_3^q \rangle$	siehe obige				$-k_2$	$-k_2$	k'_1	k'_1	$\hbar^2/4$
$\langle S_2^q S_3^q \rangle$	Tabelle				k_1	$-k_3$	$-k'_2$	$-k'_3$	$\hbar^2/4$

$$k_1 := \frac{J_a J_b}{J_a^2 + J_a J_b + J_b^2} \quad k_2 := \frac{J_a^2 + J_a J_b}{J_a^2 + J_a J_b + J_b^2} \quad k_3 := \frac{J_b^2 + J_a J_b}{J_a^2 + J_a J_b + J_b^2}$$

$$k'_1 := \frac{J_c J_d}{J_c^2 + J_c J_d + J_d^2} \quad k'_2 := \frac{J_c^2 + J_c J_b}{J_c^2 + J_c J_d + J_d^2} \quad k'_3 := \frac{J_d^2 + J_c J_b}{J_c^2 + J_c J_d + J_d^2}$$

$$\langle S_i^x \rangle = \langle S_i^y \rangle = 0$$

Für das frustrierte System sind die Eigenzustände wieder unabhängig vom Parameter J . Dagegen sind die Eigenzustände des nichtfrustrierten Systems mit verschiedenen J_{ij} Funktionen dieser Parameter.

3.5 3-Platz-t-J-Modell

Nun soll das 3-Platzmodell auch für auftretende Hoppingterme berechnet werden.

$$\hat{H} = t \sum_{\substack{(i,j,s) \\ i < j}} (\hat{c}_{is}^+ \hat{c}_{js} + \hat{c}_{js}^+ \hat{c}_{is}) + J \sum_{\substack{(i,j,s) \\ i < j}} \hat{S}_i^z \hat{S}_j^z \quad (20)$$

Der Einfachheit halber soll nur der Fall gleicher Hoppingintegrale t und Austauschwechselwirkungen J untersucht werden. Für den Unterraum $N = 3$ ist das Modell identisch mit dem Heisenbergmodell. Dieser Fall wird daher nicht mehr betrachtet. Der Zustand $N = 0$ oder $|\dots\rangle$ ist bereits Eigenzustand. Für alle anderen Unterräume haben die Hamiltonmatrizen folgendes Aussehen:

$$\begin{aligned} \underline{\psi}_a &= \begin{bmatrix} |..u\rangle \\ |.u.\rangle \\ |u..\rangle \end{bmatrix}, \quad \begin{bmatrix} |..d\rangle \\ |.d.\rangle \\ |d..\rangle \end{bmatrix} \Rightarrow \underline{\mathbb{H}}_a = \hbar^2 \begin{bmatrix} 0 & t & t \\ t & 0 & t \\ t & t & 0 \end{bmatrix} \\ \underline{\psi}_b &= \begin{bmatrix} |uu\rangle \\ |u.u\rangle \\ |uu.\rangle \end{bmatrix}, \quad \begin{bmatrix} |.dd\rangle \\ |d.d\rangle \\ |dd.\rangle \end{bmatrix} \Rightarrow \underline{\mathbb{H}}_b = \hbar^2 \begin{bmatrix} J/4 & t & -t \\ t & J/4 & t \\ -t & t & J/4 \end{bmatrix} \\ \underline{\psi}_c &= \begin{bmatrix} |.ud\rangle \\ |.du\rangle \\ |u.d\rangle \\ |d.u\rangle \\ |ud.\rangle \\ |du.\rangle \end{bmatrix} \Rightarrow \underline{\mathbb{H}}_c = \hbar^2 \begin{bmatrix} -J/4 & J/2 & t & 0 & 0 & -t \\ J/2 & -J/4 & 0 & t & -t & 0 \\ t & 0 & -J/4 & J/2 & t & 0 \\ 0 & t & J/2 & -J/4 & 0 & t \\ 0 & -t & t & 0 & -J/4 & J/2 \\ -t & 0 & 0 & t & J/2 & -J/4 \end{bmatrix} \end{aligned}$$

Die Matrizen lassen sich relativ problemlos diagonalisieren. Die Lösung führt zu folgenden Eigenzuständen und Eigenwerten von \hat{H} . Zur Vereinfachung wurde $\hbar = 1$ gesetzt.

	EF	E	N	s	m	$\langle \hat{c}_{1u}^+ \hat{c}_{2u} \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$
Ψ_1	$ \dots\rangle$	0	0	0	0	0	0	0
Ψ_2	$(..u\rangle + .u.\rangle + u..\rangle)/\sqrt{3}$	$2t$	1	1/2	1/2	1/3	0	0
Ψ_3	$(..u\rangle - u..\rangle)/\sqrt{2}$	$-t$	1	1/2	1/2	0	0	0
Ψ_4	$(.u.\rangle - u..\rangle)/\sqrt{2}$	$-t$	1	1/2	1/2	0	0	0
Ψ_5	$(.d.\rangle + .d.\rangle + d..\rangle)/\sqrt{3}$	$2t$	1	1/2	-1/2	0	0	0
Ψ_6	$(.d.\rangle - d..\rangle)/\sqrt{2}$	$-t$	1	1/2	-1/2	0	0	0
Ψ_7	$(.d.\rangle - d..\rangle)/\sqrt{2}$	$-t$	1	1/2	-1/2	0	0	0

	EF	E	N	s	m	$\langle \hat{c}_{1u}^+ \hat{c}_{2u} \rangle$	$\langle \hat{S}_1^z \hat{S}_2^z \rangle$	$\langle \vec{\tilde{S}}_1 \vec{\tilde{S}}_2 \rangle$
Ψ_8	$(.uu\rangle - u.u\rangle + uu.\rangle)/\sqrt{3}$	$J/4 - 2t$	2	1	1	$-1/3$	$1/12$	$1/12$
Ψ_9	$(.uu\rangle - uu.\rangle)/\sqrt{2}$	$J/4 + t$	2	1	1	0	$1/12$	$1/12$
Ψ_{10}	$(u.u\rangle + uu.\rangle)/\sqrt{2}$	$J/4 + t$	2	1	1	$1/3$	0	0
Ψ_{11}	$(.dd\rangle - d.d\rangle + dd.\rangle)/\sqrt{3}$	$J/4 - 2t$	2	1	-1	0	$1/12$	$1/12$
Ψ_{12}	$(.dd\rangle - dd.\rangle)/\sqrt{2}$	$J/4 + t$	2	1	-1	0	$1/12$	$1/12$
Ψ_{13}	$(d.d\rangle + dd.\rangle)/\sqrt{2}$	$J/4 + t$	2	1	-1	0	0	0
Ψ_{14}	$(.ud\rangle - du\rangle + u.d\rangle - d.u\rangle + ud.\rangle - du.\rangle)/\sqrt{6}$	$-3J/4 + 2t$	2	0	0	$1/6$	$-1/12$	$-1/4$
Ψ_{15}	$(.ud\rangle + du\rangle - u.d\rangle - d.u\rangle + ud.\rangle + du.\rangle)/\sqrt{6}$	$J/4 - 2t$	2	1	0	$-1/6$	$-1/12$	$1/12$
Ψ_{16}	$(.ud\rangle - du\rangle - ud.\rangle + du.\rangle)/2$	$-3J/4 - t$	2	0	0	0	$-1/12$	$-1/4$
Ψ_{17}	$(u.d\rangle - d.u\rangle - ud.\rangle + du.\rangle)/2$	$-3J/4 - t$	2	0	0	$-1/6$	0	0
Ψ_{18}	$(.ud\rangle + du\rangle - ud.\rangle - du.\rangle)/2$	$J/4 + t$	2	1	0	0	$-1/12$	$1/12$
Ψ_{19}	$(u.d\rangle + d.u\rangle + ud.\rangle + du.\rangle)/2$	$J/4 + t$	2	1	0	$1/6$	0	0

Die folgenden Betrachtungen gelten für $t < 0$. Für 1 Loch $N = 2$ und $m = 0$ ist der Grundzustand bei $J > -t$ Ψ_{14} und bei $J < -t$ ($a\Psi_{18} + b\Psi_{19}$). Das heißt, im Fall $J < -t$ ist der Grundzustand 2-fach entartet. Bei $J = -t$ ist das System sogar 3-fach entartet. Für $N = 2$ und $m = 1$ ist ($a\Psi_9 + b\Psi_{10}$) der entartete Grundzustand. Analog für $m = -1$ ist dieser ($a\Psi_{12} + b\Psi_{13}$). Für $N = 1$ ist der Grundzustand nicht entartet.

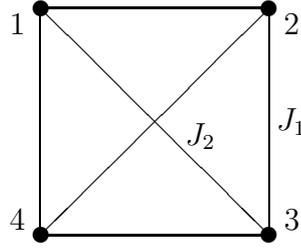


Abbildung 2: Darstellung des 4-Spin-Heisenberg-Modells

3.6 4-Spin- J_1 - J_2 -Modell

Jetzt soll ein etwas größeres kleines Heisenbergmodell mit nächster und übernächster Nachbarwechselwirkung berechnet werden. Der Hamiltonoperator für dieses System sieht wie folgt aus.

$$\hat{H} = J_1(\hat{S}_1\hat{S}_2 + \hat{S}_2\hat{S}_3 + \hat{S}_3\hat{S}_4 + \hat{S}_1\hat{S}_4) + J_2(\hat{S}_1\hat{S}_3 + \hat{S}_2\hat{S}_4) \quad (21)$$

Man kann sich dieses Modell auch als eindimensionale Kette von 4 Spins mit zyklischen Randbedingungen vorstellen. Die Heisenbergmatrix $\underline{\underline{H}} = (H_{mn}) = (\langle \psi_m | \hat{H} | \psi_n \rangle)$ hat bei gegebener Basis $|\underline{\psi}\rangle$ folgendes Aussehen.

$$\begin{aligned}
 |\psi_{a1}\rangle &= |uuuu\rangle, & |\psi_{a2}\rangle &= |dddd\rangle & \Rightarrow & H_a = \frac{\hbar^2}{2}(2J_1 + J_2) \\
 |\psi_{b1}\rangle &= \begin{bmatrix} |uudd\rangle \\ |uudu\rangle \\ |uduu\rangle \\ |duuu\rangle \end{bmatrix}, & |\psi_{b2}\rangle &= \begin{bmatrix} |uudd\rangle \\ |dudd\rangle \\ |ddud\rangle \\ |dddu\rangle \end{bmatrix} & \Rightarrow & \underline{\underline{H}}_b = \frac{\hbar^2}{2} \begin{bmatrix} 0 & J_1 & J_2 & J_1 \\ J_1 & 0 & J_1 & J_2 \\ J_2 & J_1 & 0 & J_1 \\ J_1 & J_2 & J_1 & 0 \end{bmatrix} \\
 |\psi_c\rangle &= \begin{bmatrix} |uudd\rangle \\ |udud\rangle \\ |uddu\rangle \\ |duud\rangle \\ |dudu\rangle \\ |dduu\rangle \end{bmatrix} & \Rightarrow & \underline{\underline{H}}_c = \frac{\hbar^2}{2} \begin{bmatrix} -J_2 & J_1 & J_2 & J_2 & J_1 & 0 \\ J_1 & J_2 - 2J_1 & J_1 & J_1 & 0 & J_1 \\ J_2 & J_1 & -J_2 & 0 & J_1 & J_2 \\ J_2 & J_1 & 0 & -J_2 & J_1 & J_2 \\ J_1 & 0 & J_1 & J_1 & J_2 - 2J_1 & J_1 \\ 0 & J_1 & J_2 & J_2 & J_1 & -J_2 \end{bmatrix}
 \end{aligned}$$

Zu den Untermatrizen gehören folgende Eigenwerte E .

$$\begin{aligned}
 \underline{\underline{H}}_a &\rightarrow 2J_1 + J_2 \\
 \underline{\underline{H}}_b &\rightarrow 2J_1 + J_2, \quad -2J_1 + J_2, \quad -J_2(2\times) \\
 \underline{\underline{H}}_c &\rightarrow 2J_1 + J_2, \quad -2J_1 + J_2, \quad -J_2(2\times), \quad -3J_2, \quad -4J_1 + J_2
 \end{aligned}$$

Zu den Eigenwerten gehören die normierten Eigenzustände:

$$\begin{aligned}
\Psi_1 &= |uuuu\rangle \\
\Psi_2 &= (|uud\bar{u}\bar{d}\rangle + |uudu\rangle + |udu\bar{u}\bar{d}\rangle + |duuu\rangle)/2 \\
\Psi_3 &= (|uudd\rangle + |udud\rangle + |uddu\rangle + |duud\rangle + |dudu\rangle + |dduu\rangle)/\sqrt{6} \\
\Psi_4 &= (|udd\bar{d}\bar{d}\rangle + |dudd\rangle + |ddud\rangle + |ddd\bar{u}\bar{u}\rangle)/2 \\
\Psi_5 &= |dddd\rangle \\
\Psi_6 &= (|uud\bar{u}\bar{d}\rangle - |uudu\rangle + |udu\bar{u}\bar{d}\rangle - |duuu\rangle)/2 \\
\Psi_7 &= (|udud\rangle - |dudu\rangle)/\sqrt{2} \\
\Psi_8 &= (|udd\bar{d}\bar{d}\rangle - |dudd\rangle + |ddud\rangle - |ddd\bar{u}\bar{u}\rangle)/2 \\
\Psi_9 &= (|uud\bar{u}\bar{d}\rangle + i|uudu\rangle - |udu\bar{u}\bar{d}\rangle - i|duuu\rangle)/2 \\
\Psi_{10} &= (|uud\bar{u}\bar{d}\rangle - i|uudu\rangle - |udu\bar{u}\bar{d}\rangle + i|duuu\rangle)/2 \\
\Psi_{11} &= (|uudd\rangle + i|uddu\rangle - i|duud\rangle - |dduu\rangle)/2 \\
\Psi_{12} &= (|uudd\rangle - i|uddu\rangle + i|duud\rangle - |dduu\rangle)/2 \\
\Psi_{13} &= (|udd\bar{d}\bar{d}\rangle - i|dudd\rangle - |ddud\rangle + i|ddd\bar{u}\bar{u}\rangle)/2 \\
\Psi_{14} &= (|udd\bar{d}\bar{d}\rangle + i|dudd\rangle - |ddud\rangle - i|ddd\bar{u}\bar{u}\rangle)/2 \\
\Psi_{15} &= (|uudd\rangle - |uddu\rangle - |duud\rangle + |dduu\rangle)/2 \\
\Psi_{16} &= (|uudd\rangle - 2|udud\rangle + |uddu\rangle + |duud\rangle - 2|dudu\rangle + |dduu\rangle)/\sqrt{12}
\end{aligned}$$

Die entarteten Eigenfunktionen sind so gewählt worden, daß die Erwartungswerte zu \hat{S}_i^z mit $i = 1..4$ gleich sind. In der folgenden Tabelle sind zu den jeweiligen Eigenfunktionen die Spinquantenzahl s , Magnetquantenzahl m und verschiedene Erwartungswerte dargestellt. Im folgenden gilt $i < j$ und $i, j = 1..3$. Alle Erwartungswerte beziehen sich auf $(\hbar/2)^n$.

	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ_5	Ψ_6	Ψ_7	Ψ_8	Ψ_9	Ψ_{10}	Ψ_{11}	Ψ_{12}	Ψ_{13}	Ψ_{14}	Ψ_{15}	Ψ_{16}
$\langle \hat{H} \rangle$	E _{1..5}					E _{6..8}			E _{9..14}						E ₁₅	E ₁₆
s	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0
m	2	1	0	-1	-2	1	0	-1	1	1	0	0	-1	-1	0	0
$\langle \hat{S}_i^z \rangle$	1	$\frac{1}{2}$	0	$-\frac{1}{2}$	-1	$\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0
$\langle \hat{S}_i^x \hat{S}_j^z \rangle$	1	0	$-\frac{1}{3}$	0	1	0	$2k_1$	0	0	0	$2k_2$	$2k_2$	0	0	$2k_2$	k_3
$\langle \hat{S}_i^x \hat{S}_j^x \rangle$	0	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{2}$	0	k_1	0	k_1	k_2	k_2	0	0	k_2	k_2	$2k_2$	k_3
$\langle \hat{S}_i^y \hat{S}_j^y \rangle$	0	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{2}$	0	k_1	0	k_1	k_2	k_2	0	0	k_2	k_2	$2k_2$	k_3
$\langle \hat{S}_1^x \hat{S}_2^y \hat{S}_3^z \rangle$	0	0	0	0	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0
$\langle \hat{S}_1^y \hat{S}_2^z \hat{S}_3^x \rangle$	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0
$\langle \hat{S}_1^z \hat{S}_2^x \hat{S}_3^y \rangle$	0	0	0	0	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0
$\langle \hat{E}_{123} \rangle$	0	0	0	0	0	0	0	0	-2	2	-2	2	-2	2	0	0

(i, j)	k_1	k_2	k_3	
(1, 2)	-1/2	0	-2/3	E _{1..5} = $(2J_1 + J_2)\hbar^2/2$
(1, 3)	1/2	-1/2	1/3	E _{6..8} = $(-2J_1 + J_2)\hbar^2/2$
(1, 4)	-1/2	0	-2/3	E _{9..14} = $-J_2\hbar^2/2$
(2, 3)	-1/2	0	-2/3	E ₁₅ = $-3J_2\hbar^2/2$
(2, 4)	1/2	-1/2	1/3	E ₁₆ = $(-4J_1 + J_2)\hbar^2/2$
(3, 4)	-1/2	0	-2/3	

$$\hat{E}_{123} = \hat{S}_1(\hat{S}_2 \times \hat{S}_3)$$

$$\langle S_i^x \rangle = \langle S_i^y \rangle = 0 \quad \langle S_i^x S_i^x \rangle = \langle S_i^y S_i^y \rangle = \langle S_i^z S_i^z \rangle = \frac{\hbar^2}{4}$$

Auch beim 4-Spin-System sind die Eigenzustände unabhängig von den Parametern J_1 und J_2 . Für $J_2 < J_1$ und $J_1 > 0$ ist der Grundzustand Ψ_{16} antiferromagnetisch und nicht entartet. Für $J_2 > J_1$ und $J > 0$ ist der Grundzustand Ψ_{15} .

4 Darstellung der Spin- durch Fermioperatoren

Die üblichen Spinoperatoren $\hat{S} = \frac{\hbar}{2} \hat{\sigma}$ können durch fermionische Operatoren dargestellt werden. Diese Darstellungsmöglichkeit soll in diesem Abschnitt diskutiert werden. Werden keine Einheiten angegeben, so sind entsprechende Potenzen von \hbar zu verwenden. Dabei werden die folgenden üblichen fermionischen Operatoren verwendet.

$$\begin{aligned} \hat{c}_i^+ | \dots n_i \dots \rangle &= (-1)^{N_i} \delta_{n_i,0} | \dots n_i + 1 \dots \rangle \\ \hat{c}_i | \dots n_i \dots \rangle &= (-1)^{N_i} \delta_{n_i,1} | \dots n_i - 1 \dots \rangle, \quad \text{mit} \quad N_i = \sum_{j=1}^{i-1} n_j \\ \hat{n}_i | \dots n_i \dots \rangle &= \hat{c}_i^+ \hat{c}_i | \dots n_i \dots \rangle = n_i | \dots n_i \dots \rangle \end{aligned} \quad (22)$$

Vergleicht man die Wirkungen der Operatoren $\hat{\sigma}_i^q$ mit denen der fermionischen Operatoren so finden sich leicht folgende Beziehungen:

$$\begin{aligned} \hat{\sigma}_i^z &= \hat{n}_{iu} - \hat{n}_{id} = \hat{c}_{iu}^+ \hat{c}_{iu} - \hat{c}_{id}^+ \hat{c}_{id} \\ \hat{\sigma}_i^+ &= 2 \hat{c}_{iu}^+ \hat{c}_{id} \\ \hat{\sigma}_i^- &= (\hat{\sigma}_i^+)^+ = 2 \hat{c}_{id}^+ \hat{c}_{iu} \\ \hat{\sigma}_i^x &= \frac{1}{2} (\hat{\sigma}_i^+ + \hat{\sigma}_i^-) = \hat{c}_{iu}^+ \hat{c}_{id} + \hat{c}_{id}^+ \hat{c}_{iu} \\ \hat{\sigma}_i^y &= \frac{1}{2} i (\hat{\sigma}_i^- - \hat{\sigma}_i^+) = i \hat{c}_{id}^+ \hat{c}_{iu} - i \hat{c}_{iu}^+ \hat{c}_{id} \end{aligned} \quad (23)$$

Zusammenfassend:

$$\hat{S}_i = \frac{1}{2} \sum_{ss'} \vec{\sigma}_{ss'} \hat{c}_{is}^+ \hat{c}_{is'} \quad \vec{\sigma}_{ss'} = \begin{cases} \sigma_{ud}^x = 1 & \sigma_{du}^x = 1 \\ \sigma_{ud}^y = -i & \sigma_{du}^y = i \\ \sigma_{uu}^z = 1 & \sigma_{dd}^z = -1 \end{cases} \quad \text{Rest} = 0 \quad (24)$$

Diese Darstellung ist korrekt, wenn die Vertauschungsrelation $[\hat{\sigma}^x, \hat{\sigma}^y]_- = 2i\hat{\sigma}^z$ erfüllt ist. Mit $[\hat{c}_i, \hat{c}_j]_+ = [\hat{c}_i^+, \hat{c}_j^+]_+ = 0$ und $[\hat{c}_i, \hat{c}_j^+]_+ = \delta_{ij}$ folgt:

$$\begin{aligned} [\hat{\sigma}^x, \hat{\sigma}^y]_- &= i[\hat{c}_u^+ \hat{c}_d + \hat{c}_d^+ \hat{c}_u, \hat{c}_d^+ \hat{c}_u - \hat{c}_u^+ \hat{c}_d]_- \\ &= 2i(\hat{c}_u^+ \hat{c}_d \hat{c}_d^+ \hat{c}_u - \hat{c}_d^+ \hat{c}_u \hat{c}_u^+ \hat{c}_d) = 2i[\hat{c}_u^+ \hat{c}_d, \hat{c}_d^+ \hat{c}_u]_- \\ &= 2i(\hat{c}_u^+ \hat{c}_u \hat{c}_d \hat{c}_d^+ - \hat{c}_u \hat{c}_u^+ \hat{c}_d^+ \hat{c}_d) \\ &= 2i(\hat{c}_u^+ \hat{c}_u (1 - \hat{c}_d^+ \hat{c}_d) - (1 - \hat{c}_u^+ \hat{c}_u) \hat{c}_d^+ \hat{c}_d) \\ &= 2i(\hat{c}_u^+ \hat{c}_u - \hat{c}_d^+ \hat{c}_d) \\ &= 2i\hat{\sigma}^z \end{aligned} \quad (25)$$

Die anderen Vertauschungen lassen sich analog zeigen. Im t-J-Modell werden die Fermiooperatoren durch folgende Operatoren ersetzt.

$$\begin{aligned}
\hat{c}_{is}^+ &= \hat{c}_{is}^+ \hat{p}_{is} = \hat{c}_{is}^+ (1 - \hat{n}_{i\bar{s}}) \\
\hat{c}_{is} &= \hat{c}_{is} \hat{p}_{is} = \hat{c}_{is} (1 - \hat{n}_{i\bar{s}}) \\
\hat{n}_{is} &= \hat{c}_{is}^+ \hat{c}_{is} = \hat{c}_{is}^+ \hat{p}_{is} \hat{c}_{is} \hat{p}_{is} = \hat{c}_{is}^+ \hat{c}_{is} (1 - 2\hat{n}_{i\bar{s}} + \hat{n}_{i\bar{s}}^2) \\
&= \hat{c}_{is}^+ \hat{c}_{is} \hat{p}_{is} = \hat{n}_{is} (1 - \hat{n}_{i\bar{s}})
\end{aligned} \tag{26}$$

\bar{s} ist der zu s entgegengesetzte Spinzustand ($u \leftrightarrow d$). $\hat{p}_{is} = (1 - \hat{n}_{i\bar{s}})$ ist ein Projektionsoperator, der Doppelbesetzungen verhindert. Der Operator \hat{n}_i wird dann folgendermaßen dargestellt:

$$\begin{aligned}
\hat{n}_i &= \hat{n}_{iu} \hat{p}_{iu} + \hat{n}_{id} \hat{p}_{id} \\
&= \hat{n}_{iu} (1 - \hat{n}_{id}) + \hat{n}_{id} (1 - \hat{n}_{iu}) \\
&= \hat{n}_i - 2\hat{n}_{iu} \hat{n}_{id} = \delta_{n_{iu}, 1-n_{id}}
\end{aligned} \tag{27}$$

Die Spinoperatoren nach (23) wirken auf doppeltbesetzte Plätze wie auf leere Plätze. Das heißt, Doppelbesetzungen haben den Spin 0.

$$\sigma^\alpha |du\rangle = 0 \quad \alpha = (x, y, z) \tag{28}$$

5 Vertauschungen

Mit Hilfe guter Quantenzahlen kann der zu untersuchende Hilbertraum reduziert werden. Aufgabe dieses Abschnittes soll es nun sein, solche Quantenzahlen für beide verwendete Modelle zu finden. Es müssen Operatoren gefunden werden, die mit dem Hamiltonoperator vertauschen. Dabei geht man von den fermionischen Vertauschungsrelationen aus.

$$[c_{is}, c_{js'}]_+ = 0 \quad [c_{is}^+, c_{js'}^+]_+ = 0 \quad [c_{is}, c_{js'}^+]_+ = \delta_{ij} \delta_{ss'} \tag{29}$$

Die Indizes beinhalten hier sowohl Platznummer i, j als auch Spinrichtung s, s' . Alle anderen Relationen müssen sich auf diese rückführen lassen. Gelten diese Vertauschungen auch für die Operatoren \tilde{c}^+ und \tilde{c} , so gelten die in den folgenden Abschnitten gezeigten Vertauschungen auch für diese projizierten Fermioperatoren. Um das zu zeigen, werden neben den obigen Relationen folgende verwendet.

$$\begin{aligned}
[AB, CD]_+ &= [AB, C]_+ D + C[D, AB]_- \\
&= ([C, A]_+ B + A[B, C]_-) D + C([D, A]_- B + A[D, B]_-)
\end{aligned} \tag{30}$$

$$[p_{is}, c_{js'}]_- = +\delta_{ij} \delta_{\bar{s}s'} c_{js'}, \quad [p_{is}, c_{js'}^+]_- = -\delta_{ij} \delta_{\bar{s}s'} c_{js'}^+ \tag{31}$$

$$\bar{s} \equiv -s, \quad \delta_{\bar{s}'s} \equiv \delta_{s'\bar{s}}, \quad c_{is}^+ c_{is}^+ = c_{is} c_{is} \equiv 0 \tag{32}$$

$$\begin{aligned}
[\tilde{c}_{is}, \tilde{c}_{js'}]_+ &= \underbrace{[c_{js'}, c_{is}]_+}_{0} p_{is} + c_{is} \underbrace{[p_{is}, c_{js'}]_-}_{\delta_{ij} \delta_{\bar{s}s'} c_{js'}} p_{js'} + c_{js'} \underbrace{([p_{js'}, c_{is}]_- p_{is} + c_{is} [p_{js'}, p_{is}]_-)}_{\delta_{ij} \delta_{\bar{s}s'} c_{is}} \\
&= \delta_{ij} \delta_{\bar{s}s'} c_{is} c_{i\bar{s}} (p_{i\bar{s}} + p_{is}) \quad \text{mit} \quad c_{is} p_{i\bar{s}} = 0 \\
&= 0
\end{aligned} \tag{33}$$

$$\begin{aligned}
[\tilde{c}_{is}^+, \tilde{c}_{j's'}^+]_+ &= \underbrace{([c_{j's'}^+, c_{is}^+]_+ p_{is} + c_{is}^+ [p_{is}, c_{j's'}^+]_-)}_0 p_{j's'} + c_{j's'}^+ \underbrace{([p_{j's'}, c_{is}^+]_- p_{is} + c_{is}^+ [p_{j's'}, p_{is}]_-)}_{-\delta_{ij}\delta_{ss'}c_{is}^+} \\
&= -\delta_{ij}\delta_{ss'}c_{is}^+c_{is}^+(p_{i\bar{s}} + p_{is}) \\
&= \delta_{ij}\delta_{ss'}c_{is}^+c_{i\bar{s}}^+(c_{is}^+c_{is} + c_{i\bar{s}}^+c_{i\bar{s}}) \\
&= 0
\end{aligned} \tag{34}$$

$$\begin{aligned}
[\tilde{c}_{is}, \tilde{c}_{j's'}^+]_+ &= \underbrace{([c_{j's'}^+, c_{is}]_+ p_{is} + c_{is} [p_{is}, c_{j's'}^+]_-)}_{\delta_{ij}\delta_{ss'}} p_{j's'} + c_{j's'}^+ \underbrace{([c_{is}, p_{j's'}]_- p_{is} + c_{is} [p_{is}, p_{j's'}]_-)}_{\delta_{ij}\delta_{ss'}c_{is}^+} \\
&= \delta_{ij}(\delta_{ss'}(p_{is} + c_{i\bar{s}}^+ \underbrace{c_{is}p_{i\bar{s}}}_0) + \delta_{ss'} \underbrace{c_{i\bar{s}}^+c_{is}p_{is}}_{c_{i\bar{s}}^+c_{is}}) \\
&= \delta_{ij}(\delta_{ss'}p_{is} + \delta_{ss'}c_{i\bar{s}}^+c_{is}) \neq [c_{is}, c_{j's'}^+]_+
\end{aligned} \tag{35}$$

Wird in den folgenden Ableitungen auf die letzte Vertauschung zurückgegriffen, so müssen diese gesondert betrachtet werden!

5.1 Teilchenzahlerhaltung

Eine solche gesuchte gute Quantenzahl ist zum Beispiel die Teilchenzahlerhaltung.

$$N = \sum_{is} n_{is} \tag{36}$$

n_{is} vertauscht mit den fermionischen Operatoren wie folgt:

$$[n_{is}, c_{j's'}^+]_- = \delta_{ij}\delta_{ss'}c_{j's'}^+ \quad [n_{is}, c_{j's'}]_- = -\delta_{ij}\delta_{ss'}c_{j's'} \tag{37}$$

Daraus folgt für N

$$[N, c_{j's'}^+]_- = c_{j's'}^+ \quad [N, c_{j's'}]_- = -c_{j's'} \tag{38}$$

$$\begin{aligned}
[N, c_{is}^+c_{j's'}]_- &= \underbrace{[N, c_{is}^+]_-}_{c_{is}^+} c_{j's'} + c_{is}^+ \underbrace{[N, c_{j's'}]_-}_{-c_{j's'}} \\
&= 0
\end{aligned} \tag{39}$$

Und für $i = j$ und $s = s'$ folgt:

$$[N, n_{is}]_- = [N, c_{is}^+c_{is}]_- = 0 \tag{40}$$

Für die projizierten Fermioperatoren gilt ähnliches.

$$\begin{aligned}
[n_{is}, \tilde{c}_{j's'}^+]_- &= [n_{is}, c_{j's'}^+p_{j's'}]_- \\
&= c_{j's'}^+ \underbrace{[n_{is}, p_{j's'}]_-}_0 + \underbrace{[n_{is}, c_{j's'}^+]_-}_{\delta_{ij}\delta_{ss'}c_{j's'}^+} p_{j's'} \\
&= \delta_{ij}\delta_{ss'}\tilde{c}_{j's'}^+
\end{aligned} \tag{41}$$

$$\begin{aligned}
[n_{is}, \tilde{c}_{js'}]_- &= [n_{is}, c_{js'} p_{js'}]_- \\
&= c_{js'} \underbrace{[n_{is}, p_{js'}]_-}_0 + \underbrace{[n_{is}, c_{js'}]_-}_{-\delta_{ij} \delta_{ss'} c_{js'}} p_{js'} \\
&= -\delta_{ij} \delta_{ss'} \tilde{c}_{js'}
\end{aligned} \tag{42}$$

$$\begin{aligned}
[n_{is}, \tilde{c}_{js'}^+ \tilde{c}_{ks'}]_- &= [n_{is}, \tilde{c}_{js'}^+]_- \tilde{c}_{ks'} + \tilde{c}_{js'}^+ [n_{is}, \tilde{c}_{ks'}]_- \\
&= \delta_{ss'} (\delta_{ij} \tilde{c}_{js}^+ \tilde{c}_{ks} - \delta_{ik} \tilde{c}_{js}^+ \tilde{c}_{ks})
\end{aligned} \tag{43}$$

$$\begin{aligned}
[N_s, \tilde{c}_{js'}^+ \tilde{c}_{ks'}]_- &= \delta_{ss'} (\tilde{c}_{js}^+ \tilde{c}_{ks} - \tilde{c}_{js}^+ \tilde{c}_{ks}) = 0 \\
[N, \tilde{c}_{js}^+ \tilde{c}_{ks}]_- &= 0 \quad \Rightarrow [N, H_t]_- = 0
\end{aligned} \tag{44}$$

Weshalb dann auch

$$[N, \tilde{n}_{is}]_- = [N, \tilde{c}_{is}^+ \tilde{c}_{is}]_- = 0 \tag{45}$$

gilt. Damit vertauscht N mit H_t , H_j und H_U .

5.2 S^z -Erhaltung

Als nächstes soll die Vertauschung mit S^z bestimmt werden.

$$S^z = \sum_i S_i^z = \frac{\hbar}{2} \sum_i (n_{iu} - n_{id}) \tag{46}$$

$$\begin{aligned}
[c_{is_1}^+ c_{is_2}, c_{ks_3}^+]_- &= c_{is_1}^+ c_{is_2} c_{ks_3}^+ - c_{ks_3}^+ c_{is_1}^+ c_{is_2} \\
&= +c_{is_1}^+ [c_{is_2}, c_{ks_3}^+]_+ = +c_{is_1}^+ \delta_{ik} \delta_{s_2 s_3} \\
[c_{is_1}^+ c_{is_2}, c_{ks_3}]_- &= -[c_{is_1}^+, c_{ks_3}]_+ c_{is_2} = -c_{is_2} \delta_{ik} \delta_{s_1 s_3}
\end{aligned} \tag{47}$$

$$\begin{aligned}
2[S_i^z, c_{ks}^+]_- &= [c_{iu}^+ c_{iu}, c_{ks}^+]_- - [c_{id}^+ c_{id}, c_{ks}^+]_- \\
&= c_{iu}^+ \delta_{ik} \delta_{us} - c_{id}^+ \delta_{ik} \delta_{ds} \\
&= (-1)^{\delta_{sd}} c_{iu}^+ \delta_{ik} \\
2[S_i^z, c_{ks}]_- &= (-1)^{\delta_{su}} c_{is} \delta_{ik}
\end{aligned} \tag{48}$$

$$\begin{aligned}
2[S^z, c_{is}^+]_- &= (-1)^{\delta_{sd}} c_{is}^+ \\
2[S^z, c_{is}]_- &= (-1)^{\delta_{su}} c_{is}
\end{aligned} \tag{49}$$

Und mit $[A, BC]_- = B[A, C]_- + [A, B]_- C$ folgt:

$$\begin{aligned}
2[S^z, c_{is}^+ c_{js}]_- &= ((-1)^{\delta_{su}} + (-1)^{\delta_{sd}}) c_{is}^+ c_{js} \\
&= 0 \quad \Rightarrow [S^z, H_t]_- = 0
\end{aligned} \tag{50}$$

Für die projizierten Operatoren gilt:

$$\begin{aligned}
2[S^z, \tilde{c}_{is}^+ \tilde{c}_{js}]_- &= [N_u, \tilde{c}_{is}^+ \tilde{c}_{js}]_- - [N_d, \tilde{c}_{is}^+ \tilde{c}_{js}]_- \\
&= 0 \quad \Rightarrow [S^z, H_t]_- = 0
\end{aligned} \tag{51}$$

Das läßt sich auch anschaulich leicht verstehen, da ein hüpfendes Elektron seinen Spin s^z nicht ändert.

$$\begin{aligned} [S^z, S_i^x S_j^x]_- &= +i\hbar(S_i^x S_j^y + S_i^y S_j^x) \\ [S^z, S_i^y S_j^y]_- &= -i\hbar(S_i^y S_j^x + S_i^x S_j^y) \\ [S^z, S_i^z S_j^z]_- &= 0 \end{aligned} \quad (52)$$

$$\begin{aligned} [S^z, S_i^x S_j^x + S_i^y S_j^y + S_i^z S_j^z]_- &= 0 \\ [S^z, H_J]_- &= 0 \end{aligned} \quad (53)$$

$$\begin{aligned} [S^z, n_{is}]_- &= [S^z, c_{is}^+ c_{is}]_- = 0 \\ [S^z, n_{iu} n_{id}]_- &= 0 \\ [S^z, H_U]_- &= 0 \end{aligned} \quad (54)$$

S^z vertauscht somit mit N , H_t , H_J und H_U . Statt dieser Quantenzahl können auch die folgenden gleichwertigen Quantenzahlen verwendet werden.

$$N_u = \sum_i n_{iu} = \frac{1}{2}N + S^z \quad (55)$$

$$N_d = \sum_i n_{id} = \frac{1}{2}N - S^z \quad (56)$$

5.3 S^2 -Erhaltung

Ausgangspunkt ist wieder die Darstellung der Spinoperatoren durch fermionische Operatoren.

$$\begin{aligned} 2[S_i^x, c_{ks}^+]_- &= [c_{iu}^+ c_{id}, c_{ks}^+]_- + [c_{id}^+ c_{iu}, c_{ks}^+]_- \\ &= c_{iu}^+ \delta_{ik} \delta_{sd} + c_{id}^+ \delta_{ik} \delta_{su} \\ &= +c_{i\bar{s}}^+ \delta_{ik} \\ 2[S_i^x, c_{ks}]_- &= -c_{i\bar{s}} \delta_{ik} \end{aligned} \quad (57)$$

$$\begin{aligned} 2[S^x, c_{is}^+]_- &= +c_{i\bar{s}}^+ \\ 2[S^x, c_{is}]_- &= -c_{i\bar{s}} \end{aligned} \quad (58)$$

$$\begin{aligned} 2[S^x, c_{is}^+ c_{js}]_- &= c_{i\bar{s}}^+ c_{js} - c_{is}^+ c_{j\bar{s}} \\ 2[S^x, c_{iu}^+ c_{ju} + c_{id}^+ c_{jd}]_- &= 0 \quad \Rightarrow [S^x, H_t]_- = 0 \end{aligned} \quad (59)$$

$$\begin{aligned} 2[S_i^y, c_{ks}^+]_- &= -i[c_{iu}^+ c_{id}, c_{ks}^+]_- + i[c_{id}^+ c_{iu}, c_{ks}^+]_- \\ &= -i c_{iu}^+ \delta_{ik} \delta_{sd} + i c_{id}^+ \delta_{ik} \delta_{su} \\ &= (-1)^{\delta_{sd}} i c_{i\bar{s}}^+ \delta_{ik} \\ 2[S_i^y, c_{ks}]_- &= (-1)^{\delta_{sd}} i c_{i\bar{s}} \delta_{ik} \end{aligned} \quad (60)$$

$$\begin{aligned} 2[S^y, c_{is}^+]_- &= (-1)^{\delta_{sd}} i c_{i\bar{s}}^+ \\ 2[S^y, c_{is}]_- &= (-1)^{\delta_{sd}} i c_{i\bar{s}} \end{aligned} \quad (61)$$

$$\begin{aligned} 2[S^y, c_{is}^+ c_{js}]_- &= (-1)^{\delta_{sd}} i (c_{is}^+ c_{j\bar{s}} + c_{i\bar{s}}^+ c_{js}) \\ 2[S^y, c_{iu}^+ c_{ju} + c_{id}^+ c_{jd}]_- &= 0 \quad \Rightarrow [S^y, H_t]_- = 0 \end{aligned} \quad (62)$$

Genauso lassen sich analoge Beziehungen zu den projizierten Operatoren erstellen.

$$\begin{aligned}
[\tilde{c}_{is_1}^+ \tilde{c}_{is_2}, \tilde{c}_{ks_3}^+]_- &= \tilde{c}_{is_1}^+ \tilde{c}_{is_2} \tilde{c}_{ks_3}^+ - \tilde{c}_{ks_3}^+ \tilde{c}_{is_1}^+ \tilde{c}_{is_2} \\
&= +\tilde{c}_{is_1}^+ [\tilde{c}_{is_2}, \tilde{c}_{ks_3}^+]_+ \\
&= +\tilde{c}_{is_1}^+ \delta_{ik} (\delta_{s_2 s_3} p_{is_3} + \delta_{\bar{s}_2 s_3} c_{i\bar{s}_2}^+ c_{is_3}) \\
[\tilde{c}_{is_1}^+ \tilde{c}_{is_2}, \tilde{c}_{ks_3}]_- &= \tilde{c}_{is_1}^+ \tilde{c}_{is_2} \tilde{c}_{ks_3} - \tilde{c}_{ks_3} \tilde{c}_{is_1}^+ \tilde{c}_{is_2} \\
&= -[\tilde{c}_{is_1}^+, \tilde{c}_{ks_3}]_+ \tilde{c}_{is_2} \\
&= -\delta_{ik} (\delta_{s_1 s_3} p_{is_3} + \delta_{\bar{s}_1 s_3} c_{i\bar{s}_1}^+ c_{is_3}) \tilde{c}_{is_2}
\end{aligned} \tag{63}$$

$$\begin{aligned}
2[S_i^x, \tilde{c}_{ks}^+]_- &= [\tilde{c}_{iu}^+ \tilde{c}_{id}, \tilde{c}_{ks}^+]_- + [\tilde{c}_{id}^+ \tilde{c}_{iu}, \tilde{c}_{ks}^+]_- \\
&= \tilde{c}_{iu}^+ \delta_{ik} (\delta_{ds} p_{id} + \delta_{us} c_{iu}^+ c_{iu}) + \tilde{c}_{id}^+ \delta_{ik} (\delta_{us} p_{iu} + \delta_{ds} c_{id}^+ c_{id}) \\
&= \tilde{c}_{i\bar{s}}^+ \delta_{ik} (p_{i\bar{s}} + n_{is}) \\
&= \tilde{c}_{i\bar{s}}^+ \delta_{ik}
\end{aligned} \tag{64}$$

$$\begin{aligned}
2[S_i^x, \tilde{c}_{ks}]_- &= [\tilde{c}_{iu}^+ \tilde{c}_{id}, \tilde{c}_{ks}]_- + [\tilde{c}_{id}^+ \tilde{c}_{iu}, \tilde{c}_{ks}]_- \\
&= -\delta_{ik} (\delta_{us} p_{iu} + \delta_{ds} c_{id}^+ c_{id}) \tilde{c}_{id} - \delta_{ik} (\delta_{ds} p_{id} + \delta_{us} c_{iu}^+ c_{iu}) \tilde{c}_{iu} \\
&= -\delta_{ik} (p_{i\bar{s}} + n_{is}) \tilde{c}_{id} \\
&= -\delta_{ik} \tilde{c}_{i\bar{s}}
\end{aligned} \tag{65}$$

$$\begin{aligned}
2[S^x, \tilde{c}_{is}^+]_- &= \tilde{c}_{i\bar{s}}^+ \\
2[S^x, \tilde{c}_{is}]_- &= -\tilde{c}_{i\bar{s}} \\
2[S^x, \tilde{c}_{is}^+ \tilde{c}_{js}]_- &= \tilde{c}_{i\bar{s}}^+ \tilde{c}_{js} - \tilde{c}_{i\bar{s}}^+ \tilde{c}_{j\bar{s}} \\
2[S^x, \tilde{c}_{iu}^+ \tilde{c}_{ju} + \tilde{c}_{id}^+ \tilde{c}_{jd}]_- &= \tilde{c}_{id}^+ \tilde{c}_{ju} - \tilde{c}_{iu}^+ \tilde{c}_{jd} + \tilde{c}_{iu}^+ \tilde{c}_{jd} - \tilde{c}_{id}^+ \tilde{c}_{ju} \\
&= 0 \quad \Rightarrow [S^x, H_t]_- = 0
\end{aligned} \tag{66}$$

Für S^y läßt sich das gleiche zeigen.

$$2[S^y, \tilde{c}_{iu}^+ \tilde{c}_{ju} + \tilde{c}_{id}^+ \tilde{c}_{jd}]_- = 0 \quad \Rightarrow [S^y, H_t]_- = 0 \tag{67}$$

$$\begin{aligned}
[S^x S^x, S_i^x]_- &= 0 \\
[S^y S^y, S_i^x]_- &= -i\hbar (S^y S_i^z + S_i^z S^y) \\
[S^z S^z, S_i^x]_- &= +i\hbar (S^z S_i^y + S_i^y S^z) \\
[S^2, S_i^x]_- &= +i\hbar (S^z S_i^y + S_i^y S^z - S^y S_i^z - S_i^z S^y) \\
&= +i\hbar (-i\hbar S_i^x + i\hbar S_i^x) = 0
\end{aligned} \tag{68}$$

Analog lassen sich die Vertauschungen S^2 mit S_i^y und S_i^z finden. Aus diesen folgt dann:

$$\begin{aligned}
[S^2, S_i S_j]_- &= [S^2, S_i^x S_j^x + S_i^y S_j^y + S_i^z S_j^z]_- = 0 \\
[S^2, H_J]_- &= 0
\end{aligned} \tag{69}$$

$$\begin{aligned}
[S^x, n_{is}]_- &= [S^x, c_{is}^+ c_{is}]_- = (-1)^{\delta_{sd} i} S_i^y \\
[S^y, n_{is}]_- &= [S^y, c_{is}^+ c_{is}]_- = (-1)^{\delta_{su} i} S_i^x \\
[S^z, n_{is}]_- &= [S^z, c_{is}^+ c_{is}]_- = 0
\end{aligned} \tag{70}$$

$$\begin{aligned}
[S^x S^x, n_{is}]_- &= (-1)^{\delta_{sd} i} (S^x S_i^y + S_i^y S^x) \\
[S^y S^y, n_{is}]_- &= (-1)^{\delta_{su} i} (S^y S_i^x + S_i^x S^y) \\
[S^z S^z, n_{is}]_- &= 0
\end{aligned} \tag{71}$$

$$\begin{aligned}
[S^2, n_{is}]_- &= 0 \\
[S^2, n_{iu} n_{id}]_- &= 0 \quad \Rightarrow [S^2, H_U]_- = 0
\end{aligned} \tag{72}$$

Damit vertauscht $S^2 = S^x S^x + S^y S^y + S^z S^z$ mit H_t, H_J, H_U, N und selbstverständlich auch mit S^z .

6 Der Basisraum

Die gesuchten Zustandsfunktionen werden als Linearkombinationen der Basisvektoren dargestellt.

$$|\Psi\rangle = \sum_{n=1}^{d1} a_n |\psi_n\rangle \tag{73}$$

Hier sollen Möglichkeiten gezeigt werden, die Basisvektoren $|\psi_n\rangle$ darzustellen beziehungsweise dem Index n zuzuordnen. Wenn nicht anders erwähnt, beziehen sich die Zahlenbeispiele auf ein 16-Platz-System. Im Allgemeinen kann jeder Platz durch zwei Fermionen, die sich durch ihren Spin unterscheiden, besetzt sein. Ein Basisvektor sieht dann beispielsweise folgendermaßen aus:

$$\begin{aligned}
|\psi_i\rangle &= |.. .. du d. .u .. .u .. .u d. .. .uu\rangle \\
&= |00 00 00 11 10 01 00 01 00 01 10 00 01 00 00 01\rangle
\end{aligned} \tag{74}$$

In diesem Beispiel sitzen up-Spins an den Plätzen 1, 4, 7, 9, 11 und 13. Auf den Plätzen 6, 12 und 13 sitzen down-Spins. Platz 13 ist durch 2 Fermionen besetzt. Für jeden Platz werden 2 bit benötigt, um alle möglichen Besetzungen darzustellen. Im Heisenbergmodell [15][17] wären pro Platz nur 1 bit nötig. Ohne Berücksichtigung der Vertauschungen benötigt man zur Darstellung eines Zustandes $d1 = 4^{16} = 4\,294\,967\,296$ Basisvektoren. Beim t-J-Modell werden doppelt besetzte Plätze ausgeschlossen. Damit reduziert sich die Anzahl der Basisvektoren bzw. der möglichen Spin-Konfigurationen für ein 16-Platz-System auf $d1 = 3^{16} = 43\,046\,721$. Zur Speicherung aller Vektorkomponenten a_n benötigt man bei Verwendung von 8-Byte-Realzahlen 328.5 Megabyte Speicherplatz. Berücksichtigt man nun die guten Quantenzahlen, so verringert sich die Anzahl der Basisvektoren $d1$ zum Teil erheblich. In Tabelle 1 ist die Abhängigkeit der Vektorgröße $d1$ bei verschiedenen Quantenzahlen dargestellt. Sie wächst sehr schnell mit zunehmender Zahl der Löcher ($16 - N$). Man kann eine festgelegte Zahl von up- und down-Spins (und Leerstellen) auf $d1$ verschiedene Art und Weise auf die 16 Plätze verteilen. Die möglichen Basiszustände sind so als Permutation mit Wiederholung aufzufassen. Dementsprechend berechnet sich $d1$ wie folgt.

$$d1(N) = \frac{16!}{(16 - N)! N!} 2^N \tag{75}$$

N	$d1(N)$	$ S^z _{\max}$	$d1(N, S^z _{\max})$	$ S^z _{\min}$	$d1(N, S^z _{\min})$
0	1	0/2	1	0	1
1	32	1/2	16	1/2	16
2	480	2/2	120	0	240
3	4 480	3/2	560	1/2	1 680
4	29 120	4/2	1 820	0	10 920
5	139 776	5/2	4 368	1/2	43 680
6	512 512	6/2	8 008	0	160 160
7	1 464 320	7/2	11 440	1/2	400 400
8	3 294 720	8/2	12 870	0	900 900
9	5 857 280	9/2	11 440	1/2	1 441 440
10	8 200 192	10/2	8 008	0	2 018 016
11	8 945 664	11/2	4 368	1/2	2 018 016
12	7 454 720	12/2	1 820	0	1 681 680
13	4 587 520	13/2	560	1/2	960 960
14	1 966 080	14/2	120	0	411 840
15	524 288	15/2	16	1/2	102 960
16	65 536	16/2	1	0	12 870

Tabelle 1: Anzahl der Vektorkomponenten $d1$ in Abhängigkeit von den Quantenzahlen N und S^z beim 16-Platz-Fermionensystem ohne doppelbesetzte Plätze für das t-J-Modell

$$d1(N, S^z) = \frac{16!}{(16-N)!N_u!N_d!} = \frac{16!}{(16-N)!(\frac{N}{2}-S^z)!(\frac{N}{2}+S^z)!} \quad (76)$$

Bei der Verwendung von 32-bit-Integerzahlen zur Darstellung der Basiskonfiguration darf das 32-te Bit nicht als Voreichenbit interpretiert werden. Das heißt, auf Rechenoperationen wie *div* und *mod* ist zu verzichten. Statt dessen werden im Programm nur die *shift*-, *or*- und *and*-Operationen verwendet. Diese behandeln das Vorzeichenbit wie alle anderen Bits. Fehlinterpretationen werden so von vornherein vermieden.

6.1 Kodierung

Dieser Abschnitt soll verschiedene Methoden erörtern, jedem Basiszustand $|\psi_n\rangle$ einen Index n zuzuordnen. Der umgekehrte Fall kann durch eine einfache Tabelle realisiert werden. Eine Kodierung der genannten Art ist notwendig, da eine Tabelle, die direkt über den Code (74) indiziert wird, zu groß ausfallen würde. Im folgenden sollen 2 Verfahren beschrieben werden, die eine Kodierung ermöglichen, die mit wenig Speicherplatz und Rechenzeit auskommt. Zum einen ist das die Hashing-Technik [16], zum anderen die Lin-Kodierung [17].

6.2 Hashing

Beim Hashing wird versucht, eine Menge beliebiger Schlüssel k_i auf die Menge $0 \dots m-1$ über Quasizufallsfunktionen abzubilden. Die Schlüssel stellen im hiesigen Fall die Spinkonfigura-

Konfiguration	k	$h(k)$
0011	3	3
0101	5	5
0110	6	6
1001	9	2
1010	10	3
1100	12	5

Tabelle 2: Hashkodierung für ein Spin-1/2 System mit 4 Plätzen, $S^z = 0$ und $m=7$

tionen der Basisvektoren dar. Die Menge $0 \dots m - 1$ ist identisch mit den oben erwähnten Indizes. Im einfachsten Fall sieht die Zufallsfunktion wie folgt aus:

$$h(k) = k \bmod m \quad (77)$$

m ist eine Primzahl, die größer als die Anzahl der zu kodierenden Schlüssel $d1$ ist. Diese Funktion ist sehr leicht zu realisieren und benötigt nur minimale Rechenzeit. Die Hashfunktion läßt sich auch sehr gut für sehr lange Schlüssel k verwenden. Sie sieht dann folgendermaßen aus:

$$h := k[1]; \text{ for } j := 2 \text{ to } ksize \text{ do } h := ((h * 256) + k[j]) \bmod m; \quad (78)$$

$ksize$ ist die Länge des Schlüssels in Byte. Tabelle 2 zeigt ein Beispiel für eine Hashkodierung eines 4-Platz-Heisenbergmodells. Jeder Platz ist durch ein Spin-1/2-Teilchen mit 2 möglichen Ausrichtungen besetzt. Die Tabelle zeigt auch einen wesentlichen Nachteil des Hashens. Es ist das Auftreten von Kollisionen. Unter Kollision versteht man, daß 2 verschiedene Schlüssel k den gleichen Hashwert h erhalten. Dieses Verhalten ist recht unerwünscht. Die Anzahl der Kollisionen hängt von der Größe der Schlüssel, deren Anzahl und der gewählten Primzahl m ab. Hier sollen 3 Verfahren gezeigt werden, wie das Problem der Kollision behoben werden kann. Alle Verfahren erhöhen jedoch die Suchzeit und den Speicherplatz. Als erstes soll das Verfahren der *getrennten Verkettung* erläutert werden. Hier wird für jeden Hashwert eine Liste (Kette) von Schlüsseln mit diesem Hashwert angelegt, die nach Berechnung des Hashwertes abgesucht werden muss. Da die Listen in der Regel sehr kurz sind, hält sich die Suchzeit in Grenzen. Eine weitere Möglichkeit ist das *lineare Austesten*. Es wird eine Tabelle mit m Einträgen angelegt. Nachdem zum Schlüssel k der Hashwert h gebildet wurde, wird zunächst der h -te Tabelleneintrag darauf untersucht, ob bereits ein Schlüssel eingetragen ist oder nicht. Ist keiner eingetragen so wird der Schlüssel k eingefügt, ansonsten wird der nächste freie Eintrag gesucht. Dabei können im ungünstigen Fall, noch nicht belegte Hashwerte anderer Schlüssel belegt werden, so daß sich der Suchprozeß verlängert. Eine Verbesserung liefert die Technik des *doppelten Hashing*. Dieses funktioniert im wesentlichen wie das vorhergehende. Im Kollisionsfall wird jedoch nicht der nächste freie Tabelleneintrag gesucht. Statt dessen verwendet man eine zweite Hashfunktion, die einen neuen Hashwert für die Tabelle liefert. Diese Verfahren sind ausführlich in [16] beschrieben.

h	k	Konfiguration	Bemerkung
0			leer
1			leer
2	9	1001	
3	3	0011	
4	10	1010	nach 1er Kollision
5	5	0101	
6	6	0110	
7	12	1100	nach 2 Kollisionen

Tabelle 3: Hashkodierung für ein Spin-1/2 System mit 4 Plätzen, $S^z = 0$ und $m = 7$ mit linearem Austesten

Konfiguration	a-Konf.	I_a	$J_a(I_a)$	b-Konf.	I_b	$J_b(I_b)$	$J = J_a + J_b$
0011	00	0	0	11	3	1	1
0101	01	1	1	01	1	1	2
0110	01	1	1	10	2	2	3
1001	10	2	3	01	1	1	4
1010	10	2	3	10	2	2	5
1100	11	3	5	00	0	1	6

Tabelle 4: 2D-Linkodierung für ein Spin-1/2 System mit 4 Plätzen, $S^z = 0$

6.3 Zweidimensionale Linkodierung

Für die Kodierung von Spinkonfigurationen im Heisenbergmodell hat Lin [17] eine günstigere Möglichkeit verwendet. Diese läßt sich ebenfalls für die in dieser Arbeit betrachteten Modelle verwenden. Bei dieser Kodierung wird wie folgt vorgegangen. Die Spinkonfiguration wird in 2 Teile zerlegt. Bei 16 Plätzen enthalten die beiden so erhaltenen Teile jeweils noch 8 Plätze. Enthält I die 16-bit Zahl für die Konfiguration $|\phi_n\rangle$. So enthält I_a die oberen 8 bit und I_b die unteren 8 bit. Den Werten I_a und I_b werden nun 2 Zahlen J_a und J_b zugeordnet, so daß $J = J_a + J_b$ für jede Konfiguration den gesuchten Index liefert. Zur Kodierung benötigt man jetzt nur noch 2 Tabellen $J_a(I_a)$ und $J_b(I_b)$ geringerer Größe. Tabelle 4 zeigt diese Kodierung wieder für das bereits oben erwähnte 4-Platz Modell. Für die Kodierung aller Basisvektoren des 16-Platz Hubbard-Modells werden so statt 4^{16} Speicherplätzen nur noch $2 \cdot 4^8$ Speichplätze benötigt. Für mehr Plätze bzw. längere Codes für jede Konfigurationen lassen sich auch ähnliche höherdimensionale Kodierungsformen finden. Eine solche n -dimensionale Kodierungsform stellt die Prozedur *ll2* im Anhang dar. Sie benötigt nur noch 16^3 Speicherplätze. Deren Umkehrfunktion *ll1* wurde ebenfalls im Programm aufgenommen.

7 Die Hamiltonmatrix

Wie schon in Abschnitt 2 erwähnt, wird der Hamiltonoperator in Matrixform dargestellt.

$$\underline{\underline{H}} = (H_{mn}) = (\langle \psi_m | \hat{H} | \psi_n \rangle) \quad (79)$$

Diese Matrix enthält $d1^2$ Elemente. Das heißt, sie läßt sich nur bei sehr kleinen Systemen komplett abspeichern. Da in der Regel nur nächste oder übernächste Nachbarwechselwirkungen betrachtet werden, verschwinden viele Elemente der Hamiltonmatrix. Es brauchen also nur relativ wenige Elemente der Matrix berechnet werden. In dem entwickelten Programmpaket werden die jeweils benötigten Elemente der Matrix immer wieder neu berechnet, wenn sie zur Diagonalisierung benötigt werden. Die Rechenzeit wächst damit sehr schnell mit zunehmender Anzahl von Plätzen. Das ist auch der wichtigste Grund dafür, das bisher nur relativ kleine Systeme berechnet wurden.

8 Exakte Diagonalisierung

Um die gesuchten Eigenwerte und Eigenfunktionen zu den entsprechenden Operatoren zu finden, muß die oben erwähnte Matrix diagonalisiert werden. Da nur die Grundzustände gesucht sind, erweist sich das Lanczos-Verfahren als besonders günstig. Lanczosprozeduren zur Berechnung von Eigenwerten und Eigenvektoren basieren auf der Basic-Lanczos-Rekursion zur Tridiagonalisierung von reellen symmetrischen Matrizen [18]. Die Diagonalisierung der Matrix wird also auf das einfachere Problem der Diagonalisierung einer tridiagonalen Matrix zurückgeführt. Diese Vorgehensweise liefert noch einige später zu erwähnende Vorteile. Zunächst soll die Basic Lanczos Rekursion erläutert werden. Ausgangspunkt der Rekursion ist eine n-reihige und n-spaltige quadratische Matrix A und ein zufällig generierter Startvektor v_1 . Daraus soll durch schrittweise Iteration die tridiagonale Lanczos Matrix erzeugt werden.

$$T_j = \begin{bmatrix} a_1 & b_2 & & & \\ b_2 & a_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_j \\ & & & b_j & a_j \end{bmatrix} \quad \text{mit} \quad j = 1, 2, \dots, m \quad (80)$$

Die Transformation sieht in Matrixschreibweise wie folgt aus:

$$T = V^{-1}AV \quad \text{oder} \quad VT = AV \quad (81)$$

V ist die Transformationsmatrix mit v_1, v_2, \dots als m-zeilige Spaltenvektoren oder auch Lanczos-Vektoren. Formel 81 sieht spaltenweise aufgeschrieben folgendermaßen aus.

$$\begin{aligned} Av_1 &= a_1v_1 + b_2v_2 \\ Av_2 &= b_2v_1 + a_2v_2 + b_3v_3 \\ Av_3 &= b_3v_2 + a_3v_3 + b_4v_4 \\ &\vdots \end{aligned}$$

$$\begin{aligned} Av_{m-1} &= b_{m-1}v_{m-2} + a_{m-1}v_{m-1} + b_mv_m \\ Av_m &= b_mv_{m-1} + a_mv_m \end{aligned} \quad (82)$$

Dieses Gleichungssystem läßt sich durch Auflösen jeder Zeile nach dem letzten Lanczosvektor in eine iterative Form überführen.

$$\begin{aligned} v_2 &= (Av_1 - a_1v_1)/b_2 \\ v_3 &= (Av_2 - a_2v_2 - b_2v_1)/b_3 \\ &\vdots \\ v_m &= (Av_{m-1} - a_{m-1}v_{m-1} - b_{m-1}v_{m-2})/b_m \end{aligned} \quad (83)$$

Die Koeffizienten a_i und b_i werden so gewählt, daß jeder Lanczosvektor v_i orthogonal bezüglich seiner Vorgängervektoren v_{i-1} und v_{i-2} ist.

$$a_i \stackrel{\text{def}}{=} v_i^T Av_i \quad i = 1, 2, \dots, m \quad (84)$$

$$b_{i+1} \stackrel{\text{def}}{=} v_{i+1}^T Av_i \quad b_1 = 0 \quad (85)$$

$$b_{i+1}v_{i+1} = Av_i - a_iv_i - b_iv_{i-1} \quad (86)$$

Schreibt man Gleichung (86) in Matrixform um, so erhält man für jedes j folgende Matrixgleichung

$$AV_j = V_jT_j + b_{j+1}v_{j+1}e_j^T \quad (87)$$

mit $V_j = [v_1, \dots, v_j]$ als $n \cdot j$ Matrix. e_j ist ein Koordinatenvektor dessen j -te Komponente 1 alle anderen Komponenten 0 sind. Die Lanczosprozedur generiert also in jedem Schritt zu einer reellen symmetrischen Matrix A eine tridiagonale symmetrische Matrix T . Diese benötigt wesentlich weniger Speicherplatz als die Ausgangsmatrix und ist bedeutend einfacher strukturiert. Im Programm wird sie in 2 (Quasi-)Vektoren gespeichert. Zu dieser Matrix können nun die Eigenwerte μ berechnet werden. Für $m \leq M$ (M ist die Größe von T) stellt der Eigenwert von T eine Näherung des Eigenwertes für A dar. Mittels inverser Iteration können dann die zu μ korrespondierenden Eigenvektoren u berechnet werden.

$$T_mu = \mu u \quad (88)$$

Der Vektor

$$y = V_M u \quad (89)$$

stellt dann eine Näherung für den gesuchten Eigenvektor der Matrix A dar. Das Lanczosverfahren hat nun zwei wichtige Eigenschaften, die es für die Berechnung großer Systeme besonders nützlich machen.

1. A geht nur über Av_i in die Berechnung ein, bei schwach besetzten Matrizen bedeutet das eine zur Matrixgröße proportionale Rechenzeit
2. jede Iteration benötigt nur die zwei letzten berechneten Eigenvektoren

Ist v_1 ein Startvektor mit von Null verschiedener Projektion in jedem Eigenvektor von A , so gilt [18] für jedes $j \leq n$ bei exakter Arithmetik

$$V_j^T V_j = I_j \quad \text{und} \quad T_j = V_j^T AV_j. \quad (90)$$

Das heißt, T repräsentiert A im Raum der v_j .

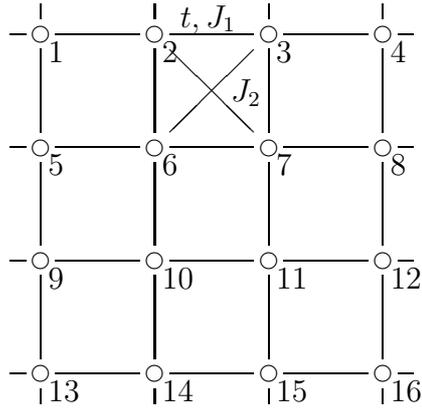


Abbildung 3: 4x4 Gitter mit periodischen Randbedingungen, nächster und übernächster Nachbarwechselwirkung

N_u	N_d	matrix size $d1$	Speicherbedarf	Zeitbedarf (100 It.)
8	8	12 870	1 MB	7 min 30 s
7	8	102 960	4 MB	37 min 20 s
7	7	411 840	13 MB	2 h 40 min
6	6	1 681 680	47 MB	32 h 44 min

Tabelle 5: Für ein 16-Platz System benötigte Rechenressourcen (IBM RISC-6000)

8.1 Berechnungen zum 16-Platz t - J_1 - J_2 -Modell

Mit zunehmender Größe können solche Modelle nur selten analytisch gelöst werden. Man ist vielmehr auf numerische Lösungstechniken angewiesen. In diesem Abschnitt sollen nun einige Ergebnisse für ein 16-Platz Modell ermittelt und Probleme erörtert werden, die bei der Behandlung solcher Systeme auftreten. Ausgangspunkt ist ein zweidimensionales Gitter mit nächster und übernächster Nachbarwechselwirkung. Der Hamiltonoperator hat für diese System folgendes Aussehen.

$$\hat{H} = t \sum_{\langle ij \rangle_s} (\hat{c}_{is}^+ \hat{c}_{js} + \hat{c}_{js}^+ \hat{c}_{is}) + J_1 \sum_{\langle ij \rangle} \hat{S}_i \hat{S}_j + J_2 \sum_{\langle\langle ij \rangle\rangle} \hat{S}_i \hat{S}_j \quad (91)$$

Dabei bedeuten $\langle ij \rangle$ nächste Nachbarn, und $\langle\langle ij \rangle\rangle$ übernächste (diagonale) Nachbarn. Ohne Loch verschwindet der Hoppingterm, da keine Hüpfprozesse stattfinden können. Das Modell ist dann identisch zum J_1 - J_2 -Heisenbergmodell. Dagotto [6] und Barnes [5] berechneten bereits einige Grundzustände für dieses Modell bei verschwindendem Diagonalterm J_2 . Barnes [5] verwendet wie in der Einleitung bereits erwähnt einen etwas modifizierten Heisenbergterm. Dadurch entsteht wie gesagt jedoch nur eine Energieverschiebung von $-8J_1$ bei Systemen ohne Loch und eine Verschiebung von $-7J_1$ bei Systemen mit einem Loch. In Tabelle 6 sind einige Werte aufgeführt, die mit dem erstellten Programm berechnet wurden.

N_u	N_d	t	J	E_0	Bemerkungen
6	6	-1.0	0.8	-16.931952	
8	8	-1.0	1.0	-11.228483	entspricht Heisenberg
8	8	-1.0	0.2	-2.2456966	$\langle \vec{S}\vec{S} \rangle = 0.00$
7	8	-1.0	0.2	-4.5441010	$\langle \vec{S}\vec{S} \rangle = 0.75$
7	8	-1.0	0.5	-7.0805320	.
7	8	-1.0	0.7	-8.8527059	.
7	8	-1.0	1.0	-11.573680	.
7	8	-1.0	2.0	-20.929656	
7	8	-1.0	5.0	-50.116749	
7	8	-1.0	20.0	-198.20394	
7	8	-1.0	50.0	-494.59538	

Tabelle 6: Energie des Grundzustandes des 16-Platz-Systems für ausgewählte N_u , N_d , t und J

Die Energie der Grundzustände wurde mit denen in verschiedenen Literaturquellen [8] [9] [5] [6] vorgestellten Ergebnisse verglichen, und bestätigen die Funktionstüchtigkeit des Programmes. Der Parameter J ist mit J_1 aus obiger Formel identisch. Bei der Berechnung des Grundzustandes traten folgende Probleme auf. Zunächst wurde ein Startvektor gewählt, der nur eine Basiskomponente enthielt. Alle anderen Komponenten wurden 0.0 gesetzt. Ohne Loch führt die Iteration zum korrekten Grundzustand. Mit Loch erhält man nicht die Energie des Grundzustandes, sondern höhere Anregungen. In diesen Fällen verschwindet das Skalarprodukt zwischen Grundzustand und Startvektor. Die Iteration verläuft dann in einem Unterraum des Hilbertraums, der den Grundzustand nicht enthält. Wählt man einen zufälligen Startvektor, so daß jede Komponente durch eine Zufallszahl gesetzt wird, vermeidet man solche Fälle. Das Skalarprodukt zwischen Grundzustand und Startvektor verschwindet in der Regel nicht. Weiterhin ist beim System mit Loch eine von den Parametern abhängige Entartung festzustellen. Der Grad der Entartung kann mit dem Programmpaket auf folgende Art und Weise festgestellt werden. Man berechnet mit dem Programm aus verschiedenen Startvektoren die Grundzustände und orthogonalisiert diese zueinander, bis keine orthogonale Komponenten mehr auftreten.

Im folgenden sollen einige Ergebnisse dargestellt werden. Es wurden die Grundzustände in Abhängigkeit vom Parameter J_2 berechnet. In Tabelle 7 ist der Grad der Entartung bei den entsprechenden Parametern und im t-J-Modell mit einem Loch aufgeführt. Zu den Grundzuständen wurden verschiedene Korrelationsfunktionen ermittelt. Berücksichtigt man die Symmetrien des Systems, kommt man zu folgenden Relationen.

$$\langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(1,0)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(0,1)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(3,0)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(0,3)} \rangle = \langle \vec{S}_1 \vec{S}_2 \rangle \quad (92)$$

$$\langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(2,0)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(0,2)} \rangle = \langle \vec{S}_1 \vec{S}_3 \rangle \quad (93)$$

$$\langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(1,1)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(1,3)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(3,1)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(3,3)} \rangle = \langle \vec{S}_1 \vec{S}_6 \rangle \quad (94)$$

J_2	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
$t=-0.2$	4	4	4	2	2	2	2	2	2	1	1	1
$t=-5.0$	6	2	2	2	2	2	4	4	4	4	4	4

Tabelle 7: Entartungsgrad des Grundzustandes beim t - J -Modell mit 16 Plätzen, einem Loch und verschiedenen Parametern t

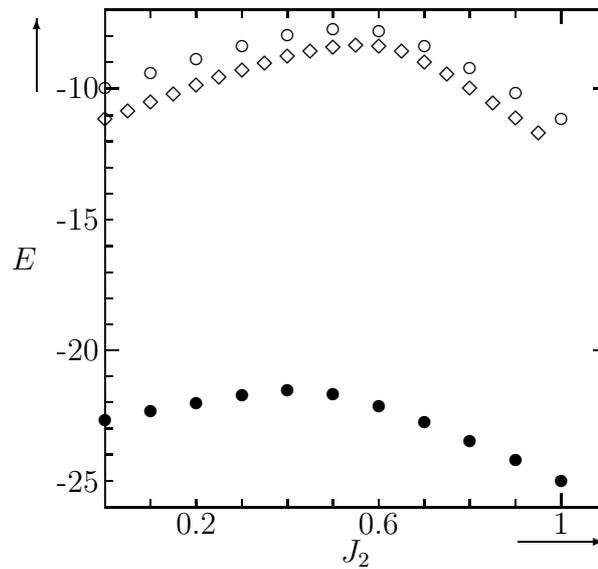


Abbildung 4: Energie des Grundzustandes beim zweidimensionalen t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

$$\langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(1,2)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(2,1)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(3,2)} \rangle = \langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(2,3)} \rangle = \langle \vec{S}_1 \vec{S}_7 \rangle \quad (95)$$

$$\langle \vec{S}_{\vec{R}} \vec{S}_{\vec{R}+(2,2)} \rangle = \langle \vec{S}_1 \vec{S}_{11} \rangle \quad (96)$$

Das Quadrat der Untergittermagnetisierung kann unter Berücksichtigung aller Symmetrien wie folgt berechnet werden:

$$(M_s)^2 = \langle \left(\frac{1}{N} \sum_{i=1}^N \tau_i \vec{S}_i \right)^2 \rangle = \frac{1}{N} \sum_{i=1}^N \tau_i \langle \vec{S}_1 \vec{S}_i \rangle \quad (97)$$

$$\tau_{i \in A} = +1 \quad \tau_{i \in B} = -1$$

A und B sind die 2 Untergitter des Systems. Zu A gehören die Gitterpunkte 1, 3, 6, 8, 9, 11, 14 und 16. Zum Untergitter B gehören die Punkte 2, 4, 5, 7, 10, 12, 13 und 15.

$$|M_s|^2 = \frac{1}{N} \sum_{i=1}^N \tau_i |\langle \vec{S}_1 \vec{S}_i \rangle| \quad (98)$$

$$(M_{s,a})^2 = \langle \left(\frac{1}{N/2} \sum_{i_a=1}^{N/2} \tau_{i,a} \vec{S}_i \right)^2 \rangle = \frac{1}{N} \sum_{i_a=1}^{N/2} \tau_{i,a} \langle \vec{S}_1 \vec{S}_i \rangle \quad (99)$$

$$\tau_{i,a} = \pm 1, \quad a = A, B,$$

Summiert wird hier jeweils nur über ein Untergitter. Der Abbildung 4 kann man entnehmen, das sich das Energiemaximum des Grundzustandes über den Parameter J_2 mit wachsendem Betrag des Hoppingintegrals zu kleineren J_2 verschiebt. Eine Verschiebung der Extremas ist auch bei den Korrelationen zu erkennen. Für betragsmäßig größere Hoppingintegrale scheint sich bei bestimmten J_2/J_1 ein Level-Crossing zu befinden. Das heißt, der Zustand tiefster Energie wechselt sprunghaft in einen anderen Zustand. Durch weitere Rechnungen kann dieser Sachverhalt genauer untersucht werden.

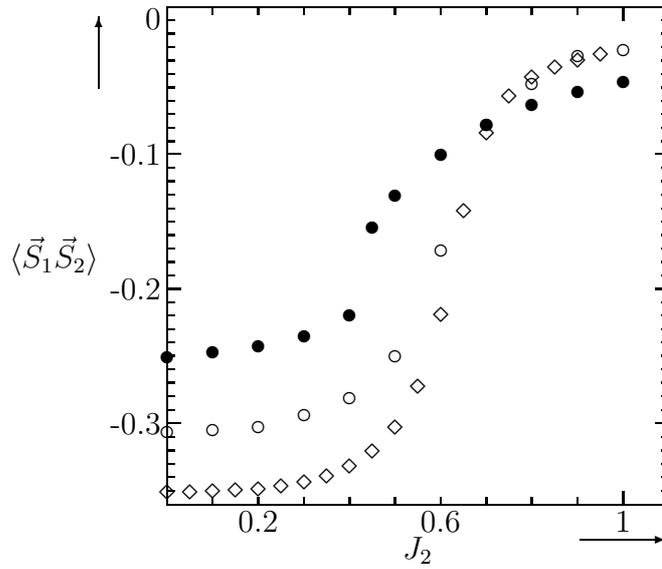


Abbildung 5: Korrelationsfunktion $\langle \vec{S}_1 \vec{S}_2 \rangle$ im Grundzustand beim t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

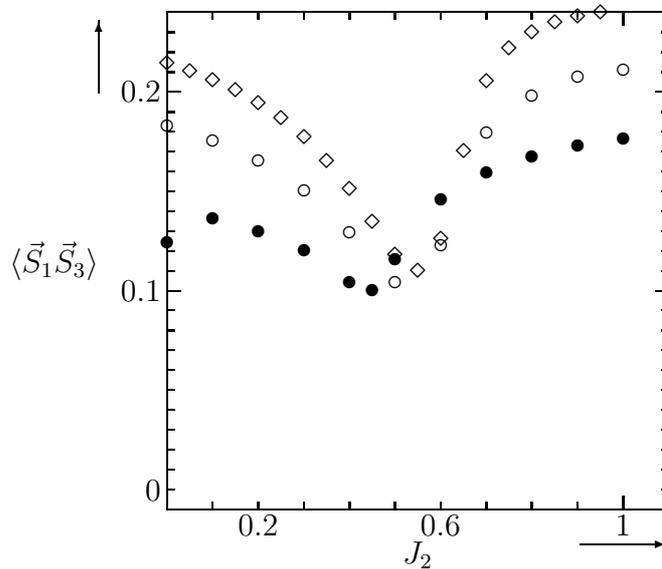


Abbildung 6: Korrelation $\langle \vec{S}_1 \vec{S}_3 \rangle$ im Grundzustand beim t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

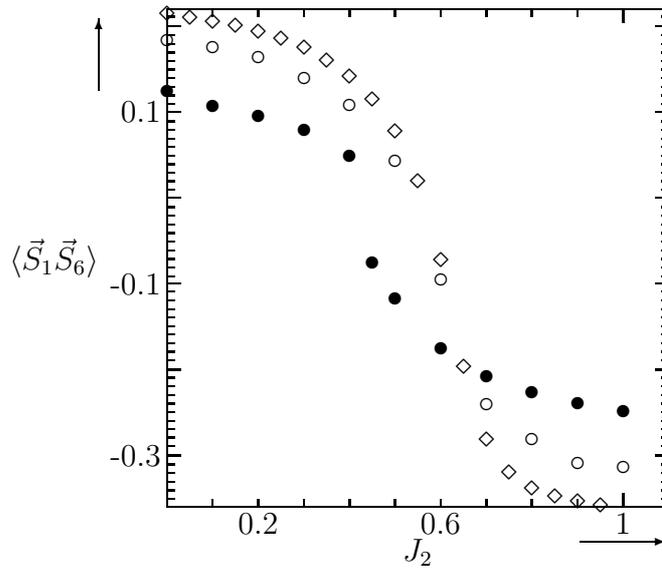


Abbildung 7: Korrelation $\langle \vec{S}_1 \vec{S}_6 \rangle$ im Grundzustand beim zweidimensionalen t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

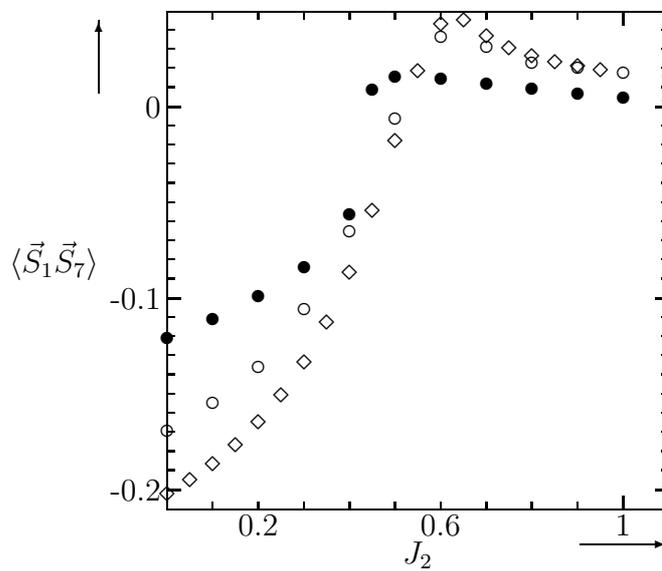


Abbildung 8: Korrelation $\langle \vec{S}_1 \vec{S}_7 \rangle$ im Grundzustand beim zweidimensionalen t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

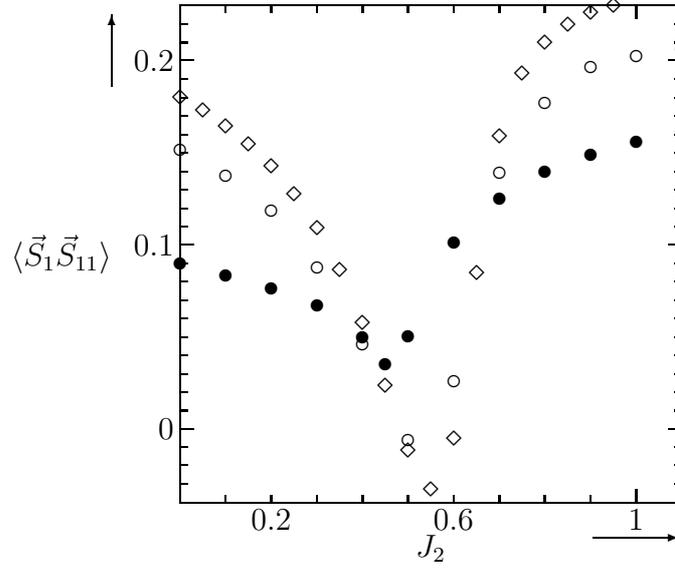


Abbildung 9: Korrelation $\langle \vec{S}_1 \vec{S}_{11} \rangle$ im Grundzustand beim zweidimensionalen t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

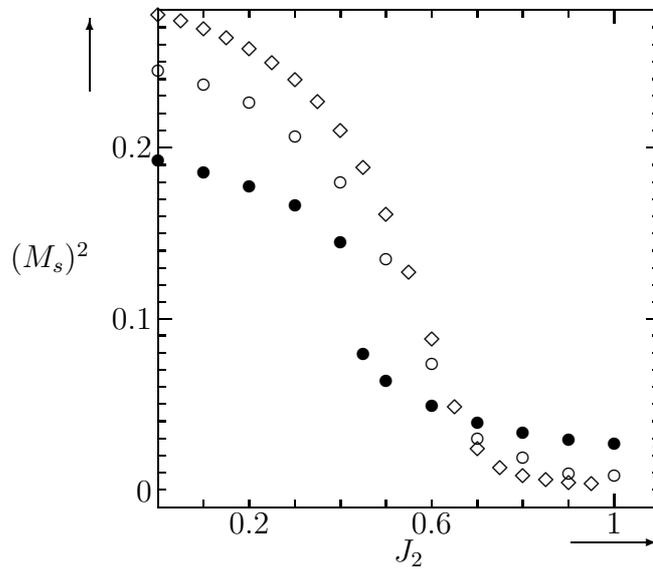


Abbildung 10: Untergittermagnetisierung $(M_s)^2$ im Grundzustand beim t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

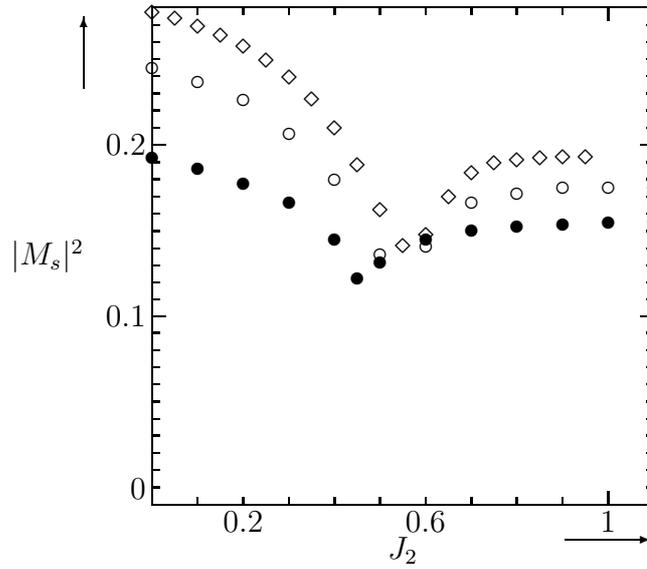


Abbildung 11: Untergittermagnetisierung $|M_s|^2$ im Grundzustand beim t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

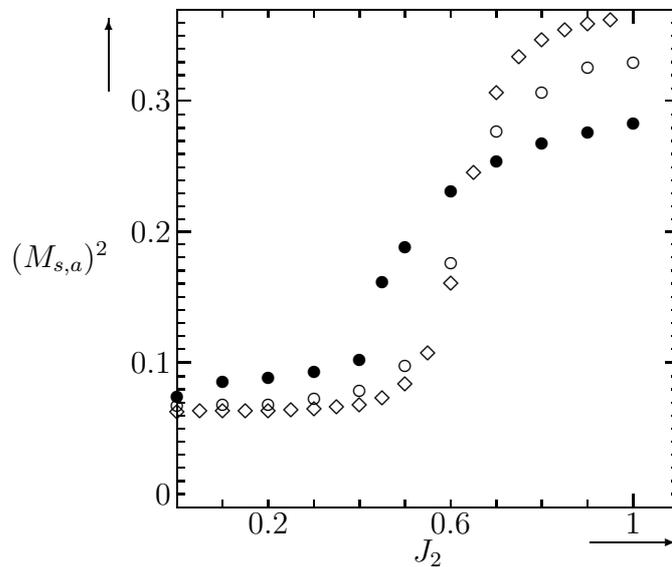


Abbildung 12: Untergittermagnetisierung $(M_{s,a})^2$ im Grundzustand beim t - J_1 - J_2 -Modell mit 16 Plätzen ohne Loch (\diamond) und mit einem Loch und $t = -0.2$ (\circ) bzw. $t = -5.0$ (\bullet). Alle Größen beziehen sich auf $J_1 = 1$.

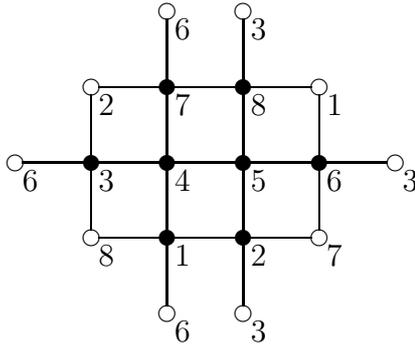


Abbildung 13: Geometrie des 2-dimensionalen 8-Platz-Gitters für das Hubbardmodell

U	E_a	E_b	E_c	Bemerkungen
1	-8.492203	-8.739323	-9.182626	$s = 0$, nicht entartet
2	-9.246918	-9.713176	-10.47214	...
5	-12.96283	-13.71674	-14.85484	...
10	-21.57773	-22.20890	-23.26870	...

Tabelle 8: Berechnete Grundenergien des Hubbardmodells mit 8 Plätzen ohne Loch und $S^z = 0$ für verschiedene Startvektoren a) $|\dots du du du du\rangle$, b) $|\dots u d. du du du\rangle$ und c) zufällig und bei verschiedenen U , bezogen auf $t = -1$

8.2 Berechnungen zum 8-Platz-Hubbard-Modell

Um zu zeigen, daß das Programm auch für das Hubbardmodell korrekte Ergebnisse liefert, werden in diesem Abschnitt die Energien des Grundzustandes für verschiedene Parameter berechnet. Diese werden dann mit Dagottos [6] Ergebnissen verglichen. Zunächst soll der Hamiltonoperator für das ein-Band-Hubbard-Modell aufgeschrieben werden.

$$\hat{H} = t \sum_{\langle ij \rangle, s} (\hat{c}_{is}^+ \hat{c}_{js} + \hat{c}_{js}^+ \hat{c}_{is}) + U \sum_i (\hat{n}_{iu} - \frac{1}{2})(\hat{n}_{id} - \frac{1}{2}) \quad (100)$$

Summiert wird über alle nächste Nachbarn $\langle ij \rangle$ und Spins $s = (u, d)$ beziehungsweise über alle Plätze i . Zur Berechnung eines Grundzustandes sind auf einer IBM RISC System/6000 580 etwa 0.3 Megabyte Speicher und bis zu 30s Rechenzeit nötig. Die Wahl des richtigen Startvektors ist hier unbedingt zu beachten. In Tabelle 8 wird dieser Sachverhalt deutlich. Es wurde die Iteration mit zwei definierten Startvektoren und einem Zufallsstartvektor durchgeführt. Dabei liegen die ersten beiden Startvektoren ungünstig. Am Ende der Iteration erhält man somit immer angeregte Zustände. Bei Nutzung des Zufallsvektors erhält man dagegen immer den Grundzustand. Die darauf folgende Abbildung zeigt die Energie des Grundzustandes für Systeme ohne und mit einem Loch. Die Ergebnisse stimmen mit Dagottos [6] überein. Der Grundzustand ist für $N_u = 4$ und $N_d = 4$ für $t = -1$ und $U > 0.1..100$ nicht entartet. Der Erwartungswert $\langle \vec{S} \vec{S} \rangle$ ist 0. Für $U = 0.01$ ist der Grundzustand 4-fach

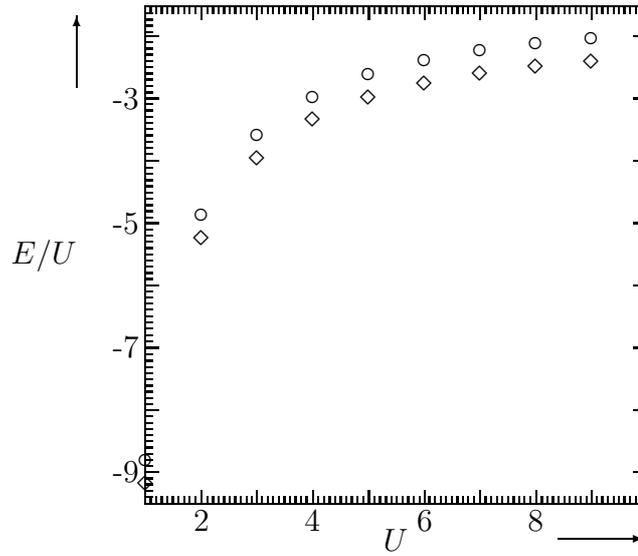


Abbildung 14: E/U (E =Grundenergie) des Hubbardmodells mit 8 Plätzen ohne Loch (\diamond) und $S^z = 0$ und mit einem Loch (\circ) und $S^z = -1/2$ bei verschiedenen U , bezogen auf $t = -1$

entartet. Mit Loch ($N_u = 3$, $N_d = 4$) ist der Grundzustand für $t = -1$ und $U = 1..10$ 4-fach entartet. Der Erwartungswert $\langle \vec{S}\vec{S} \rangle$ ist hier 3.75 ($s = 1.5$). Bei $U = 20$ ist der Grundzustand 6-fach entartet und $s = 2.5$. Bei $U = 40$ ist $s = 3.5$ maximal und der Grundzustand nicht mehr entartet.

9 Zusammenfassung

In dieser Arbeit wurde gezeigt, wie kleine fermionische Systeme prinzipiell gelöst werden können. Im ersten Teil wurden sehr kleine Systeme analytisch berechnet. Diese sollen den Leser mit der Materie vertraut machen. Der zweite Teil dieser Arbeit beschäftigt sich mit der Berechnung größerer, analytisch nicht mehr zu berechnender Modelle. Der Schwerpunkt liegt hier auf der Beschreibung der zur Berechnung dieser Modelle benutzten Lanczosprozedur. Auf diese stützt sich das im Anhang befindliche Programmpaket. Das Funktionieren des Programmes wurde mit Hilfe des Vergleiches erhaltener Eigenwerte und Korrelationen mit denen aus verschiedenen Literaturquellen und den analytisch berechneten Modellen bestätigt. Zukünftige Erweiterungen sollen die Zuordnung verschiedener entarteter Grundzustände zu bestimmten Vektoren \vec{k} des reziproken Raumes ermöglichen. Desweiteren ist eine Verkürzung der Programmlaufzeit (weniger Iterationen) durch günstigere Wahl des Startvektors zum Beispiel als Neélzustand möglich und auch wünschenswert.

10 Anhang

10.1 Programmbeschreibung

Wichtige Variablen:

- nw: Anzahl der Bindungspaare (zum Beispiel nächster Nachbarn)
- iw: Liste der Bindungspartner, (0,3 = Bindung Platz 1 mit Platz 3)
- wt: Liste der Hoppingintegrale
- wj: Liste der Austauschwechselwirkungen
- n: Anzahl der Plätze
- nu: Quantenzahl N_u des zu untersuchenden Unterraums
- nd: Quantenzahl N_d des zu untersuchenden Unterraums
- n1: Anzahl der Basiskonfigurationen bzw. Größe der Matrix \underline{H}
- n2: Länge der 2 benötigten LIN-Tabellen
- nev: Maximale Anzahl zu bestimmender Eigenvektoren zu Eigenwerten
- nam: Filename für Gruppe orthonormierter Eigenvektoren
- ew: Platz für kleinste Eigenwerte
- ev: Platz für Eigenvektoren
- itr: Anzahl benötigter Iterationen
- v0..v2: Zwischenvariablen für Zustandsvektoren
- l1,l2: Tabelle der Basiskonfigurationen und LIN-Tabellen
- io: Nummer des Hamiltonoperators, es sind 3 vordefiniert

Hauptprogramm (EW=Eigenwert, EV=Eigenvektor)

PAINI, SZ - Lintabelle anlegen, Basisraum ermitteln	
	RANDVEC - zufaelligen Startvektor waehlen
	INC1,INCV1 - EW und EV bestimmen
	IORTHO - EV zu anderen orthonormieren
	SAVEVEC - EV sichern
	CHECK1 - diverse Erwartungswerte berechnen
	Entartungsgrad+1
	RETURN bei Fehler in IORTHO

Programm INC1 (mit v_1 als Startvektor)

CLRVEC - $v_0 = (0, 0, ..)$	
$i = 1..150$	
MLTPLY - $a_i = \langle v_1 H v_1 \rangle$, $v_0 = H v_1 + v_0$	
$b_i = \ v_0 - a_i v_1\ $	
BISEC - Eigenwerte aus Tridiagonalmatrix	
Konvergenz oder $i=150$? (ja,nein)	
VEC12 - EVs der Tridiagonalmatrix	
RETURN	
$v_2 = v_1$, $v_1 = (v_0 - a_i v_1)/b_i$, $v_0 = -b_i v_2$	

Liste aller verwendeten Prozeduren:

tjModell:	Hauptprogramm zur Bestimmung des Grundzustandes + Entartung
itime:	liefert Laufzeit in Sekunden
iortho:	Orthonormierung zweier Vektoren
randvec:	Erzeugung eines zufälligen normierten (Start-)Vektors
normiere:	Normierung eines Vektors
outHmatrix:	(teilweise) Ausgabe der Basis und der Hamiltonmatrix
fak:	Berechnung der Fakultät
np:	Berechnung der Permutation mit Wiederholung von 3 Elementen
paini:	Belegt Array mit Permutationswerten (spart Neuberechnungen)
iplatz:	Selektiert Besetzung eines Platzes (leer,up,down)
ll1:	Zuordnung Zahl zu Spinkonfiguration
ll2:	Zuordnung Spinkonfiguration zu Zahl
ckfg:	Spinkonfiguration darstellen
sz:	Lintabelle anlegen
lin:	Linkodierung wie <i>ll2</i>
hamilton:	Hamiltonoperator, berechnet $v_1 = Hv_0$
mltply:	Berechnung $pd = \langle v_1 H v_1 \rangle$ und $v_0 = H v_1 + v_0$
clrvec:	0-Vektor erzeugen, eventuell <i>iv</i> -te Komponente 1.0 setzen
savevec:	Vektor speichern
loadvec:	Vektor laden
lanz1:	Eigenvektor und Eigenwert ohne explizite Tridiagonalisierung
incl:	Tridiagonalisierung und Eigenwerte bestimmen
bisec:	kleinste Eigenwerte einer tridiagonalen symmetrischen Matrix
vec12:	Eigenvektoren einer tridiagonalen symmetrischen Matrix
incv1:	Transformation der Eigenvektoren über inverse Iteration
check1:	Berechnung diverser Erwartungswerte und Korrelationen

10.2 Programm

```

c Fortranprogramm zur Berechnung kleiner fermionischer Systeme
c hier speziell: T-J-Modell
c Kodierung: | . d . . u d u .>
c           | 0 2 0 0 1 2 1 0>
c           |00 10 00 00 01 10 01 00> => 8096+64+32+4=8196
c Platznummer: 7 6 5 4 3 2 1 0
c max:          n1 = 2 018 016
c              n2 = 65 536
c size: nev*n1*8 + n1*4+n2*8 + 24*n1 = n1(8nev+28)+530k
c-----
      BLOCK DATA
      implicit real*8 (a-h,o-z),integer*4 (i-n)
      common/bw /nw,iw(2,256) /brw/wt(256),wj(256) ! nw=0..256
c --- 2D-4*4-Platzsystem(32ww) ----
c --- iw(1,i) muss kleiner als iw(2,i) sein !!!
c --- iw -> Wechselwirkungen zwischen je 2 Plaetzen (Nr=PlatzNr-1)
      data nw/64/
      data iw/0, 1, 1, 2, 2, 3, 0, 3, 4, 5, 5, 6, 6, 7, 4, 7,
+      8, 9, 9,10, 10,11, 8,11, 12,13, 13,14, 14,15, 12,15,
+      0, 4, 4, 8, 8,12, 0,12, 1, 5, 5, 9, 9,13, 1,13,
+      2, 6, 6,10, 10,14, 2,14, 3, 7, 7,11, 11,15, 3,15,
+      0, 5, 5,10, 10,15, 15, 0, 1, 6, 6,11, 11,12, 1,12,
+      2, 7, 7, 8, 8,13, 2,13, 3, 4, 4, 9, 9,14, 3,14,
+      0, 7, 7,10, 10,13, 0,13, 1, 4, 4,11, 11,14, 1,14,
+      2, 5, 5, 8, 8,15, 2,15, 3, 6, 6, 9, 9,12, 3,12,
+      384*0/
      data wt/32*-5.0d0, 224*0.0d0/ ! 0.5,1.0,2.0,5.0
      data wj/32* 1.0d0,32* 0.0d0,192*0.0d0/
      ! ----- 4-Platz -----
c      data nw/6/
c      data iw/0,1, 1,2, 2,3, 0,3, 0,2, 1,3, 500*0/
c      data wt/4*-1.0d0, 252* 0.0d0/
c      data wj/4* 1.0d0,2* 0.0d0, 250* 0.0d0/
      ! ----- 8-Platz -----
c      data nw/32/
c      ! 2-Platz-Wechselwirkungen: NR-0 entspricht hier Platz 1 etc.
c      ! NR-1.Platz muss kleiner NR-2.Platz sein -> proc_hamilton
c      ! im 8er alle geraden/ungeraden Plaetze je ein Untergitter
c      data iw/5,6, 6,7, 2,7, 0,7, 1,2, 2,3, 3,6, 3,4, ! nn
c      +      4,7, 4,5, 0,5, 2,5, 0,3, 0,1, 1,4, 1,6, ! nn
c      +      2,6, 4,6, 3,7, 5,7, 0,2, 2,6, 1,3, 3,7, ! nnn (diagonalen)
c      +      0,4, 4,6, 1,5, 5,7, 0,2, 0,4, 1,3, 1,5, ! nnn alle 2fach im 8er
c      +      448*0/
c      data wt/16*-1.0d0, 240*0.0d0/
c      data wj/16* 1.0d0, 240*0.5d0/
      END

```

```

c-----
program tjModell
implicit real*8 (a-h,o-z),integer*4 (i-n)          ! nev=1..4
c   parameter (n=16,nu=5,nd=5,n1=2 018 016,n2=2**16-1,nev=1)
c   parameter (n=16,nu=7,nd=8,n1= 102 960,n2=2**16-1,nev=1)
c   parameter (n=16,nu=8,nd=8,n1= 12 870,n2=2**16-1,nev=1)
c   parameter (n=16,nu=6,nd=6,n1=1 681 680,n2=2**16-1,nev=1)
c   parameter (n= 8,nu=3,nd=4,n1= 280,n2=2** 8-1,nev=1)
c   parameter (n= 4,nu=2,nd=2,n1= 6,n2=2** 4-1,nev=1)
c   parameter (n= 4,nu=1,nd=3,n1= 4,n2=2** 4-1,nev=1)
c   parameter (n= 4,nu=1,nd=2,n1= 12,n2=2** 4-1,nev=1)
c   parameter (n= 4,nu=0,nd=3,n1= 4,n2=2** 4-1,nev=1)
c   parameter (n= 4,nu=1,nd=1,n1= 12,n2=2** 4-1,nev=1)
c   parameter (n= 4,nu=0,nd=2,n1= 6,n2=2** 4-1,nev=1)
c   parameter (n= 4,nu=0,nd=1,n1= 4,n2=2** 4-1,nev=1)
character*12 nam
integer*1 inam(12)
equivalence (nam,inam(1))
common/be /ew(4),iv,itr,ev(n1,nev)
common/bv /v0(n1),v1(n1),v2(n1)
common/bl /l1(n1),l2(2,0:n2)
common/bw /nw,iw(2,256) /brw/wt(256),wj(256) /operator/io
nam='s161f.tj1 ' ! Dateiname fuer Vektoren
call srand(0.5) ! Zufallsgenerator initialisieren
if(n1.ne.np(n-nu-nd,nu,nd))stop 'Fehler n1<>n!/(no!nu!nd!)'
if(n2.ne.2**n-1) stop 'Fehler n2<>2**n-1'
c --- io=0 Op=t(i,j)(cp(i)c(j)+h.c.)+j(i,j)(S(i)S(j)-ninj/4)
c --- io=1 Op=t(i,j)(cp(i)c(j)+h.c.)+j(i,j)(S(i)S(j))
c --- io=2 Op=S*S
call paini(n) ! beschleunigt SZ um das 4fache
call sz(n,nu,nd,n1,n2,l1,l2) ! LINcodetabelle anlegen
10 io=1 ! Operator-nr.
iv=1 ! DB: 1..n1 -> Startvector, Entartung
write(*,'(5(A4,I3),4(A4,F5.2))')
+ 'iv=',iv,'io=',io,'n=',n,'nu=',nu,'nd=',nd,
+ 't1=',wt(1),'t2=',wt(nw),'j1=',wj(1),'j2=',wj(nw)
! call outHmatrix(v0,v1,n,n1,n2,l1,l2) ! H-Matrix und Basis
75 write(*,'(A,I8)') 'iv=',iv ! Nr. Startvektor
! call clrvec(n1,v0,iv) ! oder LOADVEC
! call lanz1(n,n1,n2,nev,v0,v1,v2,ew,iv,itr,l1,l2) ! EW SAVEVEC
! call clrvec(n1,v1,iv) ! definierter Startvektor
call randvec(v1,n1) ! zufaelliger Startvektor
call savevec(n1,v1,'startvec ') ! Startvektor speichern
call incl (n,n1,n2,nev,v0,v1,v2,ew,iv,itr,l1,l2) ! EW
! call clrvec(n1,v1,iv) ! definierter Startvektor
call loadvec(n1,v1,'startvec ') ! Startvektor laden
call incv1(n,n1,n2,nev,v0,v1,v2,iv,itr,ev,l1,l2) ! EV

```

```

do 69,j=1,n1                ! nev=1 nur Grundzustand!
69  v0(j)=ev(j,1)           ! nev=2 fuer 1. Anregung
do 85,i2=1,iv-1            ! zu anderen EVs orthonormieren
  inam(4)=48+i2             ! Dateinamen aendern
  call loadvec(n1,v1,nam)   ! anderen EV laden
  if(iortho(v0,v1,n1).eq.1) goto 67 ! vermutlich letzter entarteter EV
85  continue                ! Ende Schleife i2
  inam(4)=48+iv            ! Dateinamen aendern
  call savevec(n1,v0,nam)   ! 7,8s16->800K
  call check1(n,n1,n2,v0,v1,l1,l2) ! diverse Erwartungswerte und Fehler
  iv=iv+1                  ! Entartungsgrad + 1
  goto 75                   ! Ende Schleife iv (anderer EV)
67  continue                ! Entartungsgrad gefunden
  write(*,'(A,I3,A,I6,A)') ' Entartung=',iv-1,' t=',itime(),'s'
  inam(5)=inam(5)+1        ! neuer Filename
  do 71,i=33,64            ! j2 aendern
71  wj(i)=wj(i)+0.1        ! Schrittweite j2/j1
  if(wj(nw).lt.1.2d0) goto 10 ! Lintab nur beim 1. Lauf!
  end
c --- ---- Zeit in s (1a=31557600s) -----
integer*4 function itime() ! liefert Laufzeit in Sekunden
itime = mclock()/100      ! je nach FORTRAN-Version
end

c --- ---- orthonormiere v0 zu v1 -----
integer*4 function iortho(v0,v1,n1) ! 0=ok,1=parallel
implicit real*8 (a-h,o-z),integer*4 (i-n)
dimension v0(n1),v1(n1)
r1=0.0d0
do 10,i=1,n1
10  r1=r1+v0(i)*v1(i)
  if(abs(r1).lt.0.99999d0) goto 20
  write(*,'(A)') ' Fehler proc_iortho, v0 und v1 parallel'
  iortho=1 ! returncode fehler
  return
20  if(abs(r1).lt.1.0d-20) goto 50 ! ist bereits orthogonal
  r2=0.0d0 ! gleich normieren, geht schneller!
  do 30,i=1,n1
    v0(i)=v0(i)-r1*v1(i)
    r2=r2+v0(i)*v0(i)
30  continue
  r2=1.0d0/sqrt(r2)
  do 40,i=1,n1
40  v0(i)=v0(i)*r2
50  iortho=0 ! returncode ok.
  end
c --- ---- Zufallsvector erzeugen -----
subroutine randvec(v0,n1)

```

```

        implicit real*8 (a-h,o-z),integer*4 (i-n)
        dimension v0(n1)
        do 10,i=1,n1
10      v0(i)=2.0d0*rand()-1.0d0 ! zwischen -1 und +1
        call normiere(v0,n1)
        end
c --- ---- normiere v0 -----
        subroutine normiere(v0,n1) ! 0-Vector wird nicht normiert
        implicit real*8 (a-h,o-z),integer*4 (i-n)
        dimension v0(n1)
        r=0.0d0
        do 10,i=1,n1
10      r=r+v0(i)*v0(i)
        if(abs(r).gt.1.0d-20) goto 20
        write(*,'(A)') ' Fehler bei proc_normiere, 0-Vector'
        return
20      r=1.0d0/sqrt(r)
        do 30,i=1,n1
30      v0(i)=v0(i)*r
        end
c --- ---- H-Matrix ausgeben -----
        subroutine outHmatrix(v0,v1,n,n1,n2,l1,l2)
        implicit real*8 (a-h,o-z),integer*4 (i-n)
        character*16 CKFg
        dimension v0(n1),v1(n1),l1(n1),l2(2,0:n2)
        if (n.gt.4) goto 30 ! ausfuehrlich bei kleinen Matrizen
        write(*,'(3H B=,12A16)') (ckfg(n,l1(i)),i=1,n1)
        write(*,'(3H H=)')
        do 20 i=1,n1
            call clrvec(n1,v0,i)
            call hamilton(v0,v1,n,n1,n2,l1,l2)
            write(*,'(3H ,8F10.6)') (v1(ii),ii=1,n1)
20      continue
30      end
c --- ---- Fakultaet -----
        real*8 function fak(i)
        implicit real*8 (a-h,o-z),integer*4 (i-n)
        a=1.0 ! wegen Groesse als REAL-Zahl
        do 10, j=2,i
10      a=a*j
        fak=a
        end
c --- Permutation mit Wiederholung (3 Elemente) -----
        integer*4 function np(no,nu,nd)
        implicit real*8 (a-h,o-z),integer*4 (i-n)
        np=NINT(fak(no+nu+nd)/fak(no)/fak(nu)/fak(nd))
        end

```

```

c --- Permutationsmatrix belegen, spart Neuberechnungen -----
      subroutine paini(n)
      implicit real*8 (a-h,o-z),integer*4 (i-n)
      common/bpa/npa(0:16,0:16,0:16)
      do 20 i1=0,n,1
      do 20 i2=0,n-i1,1
      do 20 i3=0,n-i1-i2,1
20    npa(i1,i2,i3)=np(i1,i2,i3)
      end
c --- Besetzung (0..3) am Platz i(0..n-1) in code j(0..4**n-1)
      integer*4 function iplatz(i,j)
      implicit real*8 (a-h,o-z),integer*4 (i-n)
      iplatz=iand(ishft(j,-2*i),3)
      end

c --- -----
c --- Inp: Index des Basiszustandes, Out: Bitmuster des Basiszustandes
      integer*4 function ll1(ii,n,nu,nd)
      implicit real*8 (a-h,o-z),integer*4 (i-n)
      common/bpa/npa(0:16,0:16,0:16)
      j=0
      io=n-nu-nd
      iu=nu
      id=nd
      i=ii
30    if (io.le.0) goto 32
      if (i.gt.npa(io-1,iu,id)) goto 31
      j=j*4
      io=io-1
      goto 30
31    i=i-mpa(io-1,iu,id)
32    if (iu.le.0) goto 34
      if (i.gt.npa(io,iu-1,id)) goto 33
      j=ior(j*4,1)
      iu=iu-1
      goto 30
33    i=i-mpa(io,iu-1,id)
34    if (id.le.0) goto 36
      if (i.gt.npa(io,iu,id-1)) goto 35
      j=ior(j*4,2)
      id=id-1
      goto 30
35    i=i-mpa(io,iu,id-1)
36    ll1=j
      end

c --- -----
c --- Inp: Bitmuster des Basiszustandes, Out: Index des Basiszustandes

```

```

integer*4 function ll2(ii,n,nu,nd)
implicit real*8 (a-h,o-z),integer*4 (i-n)
common/bpa/npa(0:16,0:16,0:16)
io=n-nu-nd
iu=nu
id=nd
j=1
do 37, i=1,n
  i1=iplatz(n-i,ii)
  if ((i1.gt.0).and.(io.gt.0)) j=j+npa(io-1,iu,id)
  if ((i1.gt.1).and.(iu.gt.0)) j=j+npa(io,iu-1,id)
  if (i1.eq.0) io=io-1
  if (i1.eq.1) iu=iu-1
  if (i1.eq.2) id=id-1
37  continue
  ll2=j
end

c --- -----
c --- Ausgabe der Spinkonfiguration als Textstring zB: ...u.dudu
character*16 function ckfg(n,ic)
implicit real*8 (a-h,o-z),integer*4 (i-n)
character str*16,s2*1(4)
data s2 /'.','u','d','3'/          ! . = Platz leer
str='          '                   ! u = up-spin
do 40, i=0,n-1                     ! d = down-spin
40  str(16-i:16-i)=s2(iplatz(i,ic)+1) ! 3 = up+down-spins
  ckfg=str
end

c --- -----
c --- Berechnung der Umsetzungstabellen
c --- Out: Tabellen l1,l2 fuer 2D-Lincodierung
subroutine sz(n,nu,nd,n1,n2,l1,l2)
implicit real*8 (a-h,o-z),integer*4 (i-n)
character*16 ckfg
common/bpa/npa(0:16,0:16,0:16)
dimension l1(n1),l2(2,0:n2)
ja=0
jb=0
ibx=0
do 10,i=0,n2
  l2(1,i)=-1
  l2(2,i)=-1
10  continue
c  write(*,'(A17,5A6)') 'code','ib','ia','jb','ja','jb+ja'
do 50 icnt=1,n1
  i=ll1(icnt,n,nu,nd)
  ia=iand(i,2**n-1)

```

```

        ib=ishft(i,-n)
        if(ib.eq.ibx)then
            ja=ja+1
        else
            ibx=ib
            ja=1
            jb=icnt-1
        endif
        ierr=0
        if(ja+jb.ne.icnt)          ierr=1      ! Kontrolle zur Sicherheit!
c      if(l12(i,n,nu,nd).ne.icnt) ierr=2      ! Vergleich mit nD-Lincodierung
        if((l2(1,ia).ne.-1).and.(l2(1,ia).ne.ja)) ierr=3
        if((l2(2,ib).ne.-1).and.(l2(2,ib).ne.jb)) ierr=4
c      if((icnt.lt.20).or.
        if((ierr.ne.0))
+ write(*,'(A17,5I6,I2)') ckfg(n,i),ib,ia,jb,ja,icnt,ierr
        if(ierr.ne.0) stop 'Fehler in Lincodierung'
        l1(icnt)=i
        l2(1,ia)=ja
        l2(2,ib)=jb
50  continue
    end
c --- Umrechnung Index <-> Spinkonfiguration -----
integer*4 function lin(i,n,n2,l2)
implicit real*8 (a-h,o-z),integer*4 (i-n)
dimension l2(2,0:n2)
ia=iand(i,ishft(1,n)-1)
ib=ishft(i,-n)
lin=l2(1,ia)+l2(2,ib)
end
c -----
c --- Hamiltonoperator H(index)          |v1>=H*|v0>
c --- inp: wn=Anzahl_Bounds,v0, out: v1
subroutine hamilton(v0,v1,n,n1,n2,l1,l2)
implicit real*8 (a-h,o-z),integer*4 (i-n)
dimension l1(n1),l2(2,0:n2),v0(n1),v1(n1)
common/bw /nw,iw(2,256)/brw/wt(256),wj(256)
common/operator /io
call clrvec(n1,v1,0)
do 999,i=1,n1
if (io.ne.1.and.io.ne.0) goto 112
do 100, i1=1,nw
    ib1=l1(i)
    iw1=iw(1,i1)
    iw2=iw(2,i1)
    is=1
    do 10,i2=iw(1,i1)+1,iw(2,i1)-1

```

```

10  if(btest(ib1,i2*2).or.btest(ib1,i2*2+1)) is=-is
    ip1=iplatz(iw1,ib1)
    ip2=iplatz(iw2,ib1)
    ib1=ib1-ishft(ip1,2*iw1)-ishft(ip2,2*iw2)
    ii=ieor(ip1,ip2)
    if ((ii.eq.2).or.(ii.eq.1)) then ! --- hopping ou,od,uo,do
        ib2=ib1+ishft(ip2,2*iw1)+ishft(ip1,2*iw2)
        i2=lin(ib2,n,n2,l2)
        v1(i2)=v1(i2)+is*wt(i1)*v0(i)
    endif
    if (io.ne.1) goto 20 ! --- Heisenberg ohne -1/4ninj
    if (ip1.eq.ip2.and.ip1.gt.0) then ! --- uu,dd
        v1(i)=v1(i)+wj(i1)*0.25*v0(i)
    endif
    if (ii.eq.3) then ! --- ud,du
        ib2=ib1+ishft(ip2,2*iw1)+ishft(ip1,2*iw2)
        i2=lin(ib2,n,n2,l2)
        v1(i2)=v1(i2)+wj(i1)*0.50*v0(i)
        v1(i )=v1(i )-wj(i1)*0.25*v0(i)
    endif
20  if (io.ne.0) goto 100 ! --- Heisenberg mit -1/4ninj
    if (ii.eq.3) then ! --- ud,du
        ib2=ib1+ishft(ip2,2*iw1)+ishft(ip1,2*iw2)
        i2=lin(ib2,n,n2,l2)
        v1(i2)=v1(i2)+wj(i1)*0.5*v0(i)
        v1(i )=v1(i )-wj(i1)*0.5*v0(i)
    endif
100 continue
112 if (io.ne.2) goto 999 ! --- SS=Sum(SiSi)+2Sum(SiSj)
    ib1=l1(i)
    do 80,iw1=0,n-1
        ip1=iplatz(iw1,ib1)
        if(ip1.eq.1.or.ip1.eq.2) v1(i)=v1(i)+0.75*v0(i)
80  continue
    do 200,iw1=0,n-2
    do 90,iw2=iw1+1,n-1
        ip1=iplatz(iw1,ib1)
        ip2=iplatz(iw2,ib1)
        if(ip1.gt.0.and.ip1.lt.3.and.ip2.gt.0.and.ip2.lt.3) then
            if(ip1.eq.ip2) v1(i)=v1(i)+0.5*v0(i) ! SizSjz
            if(ip1.ne.ip2) v1(i)=v1(i)-0.5*v0(i) ! SizSjz
            if(ip1.ne.ip2) then ! SixSjx+SiySjy
                ib2=ieor(ib1,ishft(3,2*iw1)+ishft(3,2*iw2))
                i2 =lin(ib2,n,n2,l2)
                v1(i2)=v1(i2)+v0(i)
            endif
        endif
    endif
endif

```

```

90  continue
200 continue
999 continue
    end

c -----
c --- Operator SiSj(index)
subroutine opsisj(iw1,iw2,v0,v1,n,n1,n2,l1,l2)
implicit real*8 (a-h,o-z),integer*4 (i-n)
dimension l1(n1),l2(2,0:n2),v0(n1),v1(n1)
call clrvec(n1,v1,0)
do 999,i=1,n1
  ib1=l1(i)
  ip1=iplatz(iw1,ib1)
  ip2=iplatz(iw2,ib1)
  if(iw1.eq.iw2) then
    if(ip1.gt.0)v1(i)=v1(i)+0.75*v0(i)          ! extra, sonst fehler
  else
    if(ip1.gt.0.and.ip1.lt.3.and.ip2.gt.0.and.ip2.lt.3) then
      if(ip1.eq.ip2) v1(i)=v1(i)+0.25*v0(i)    ! --- SizSjz
      if(ip1.ne.ip2) v1(i)=v1(i)-0.25*v0(i)    ! --- SizSjz
      if(ip1.ne.ip2) then                       ! --- SixSjx+SiySjy
        ib2=ieor(ib1,ishft(3,2*iw1)+ishft(3,2*iw2))
        i2 =lin(ib2,n,n2,l2)
        v1(i2)=v1(i2)+0.5*v0(i)
      endif
    endif
  endif
endif
999  continue
    end

c -----
c --- Multiplikationen pd=<v1*H*v1>,v0=H*v1+v0
c --- Inp: v0,v1 Out:v0,pd Use:v2
subroutine mltply(v1,v0,v2,pd,n,n1,n2,l1,l2)
implicit real*8 (a-h,o-z),integer*4 (i-n)
dimension v0(n1),v1(n1),v2(n1)
call hamilton(v1,v2,n,n1,n2,l1,l2)
pd=0.d0
do 20,j=1,n1
  v0(j)=v0(j)+v2(j)
  pd=pd+v1(j)*v2(j)
20  continue
    end

c --- --- Clear Vector -----
subroutine clrvec(n1,vx,iv)
implicit real*8 (a-h,o-z), integer*4 (i-n)
dimension vx(n1)
do 10,i=1,n1

```

```

10  vx(i)=0.d0
    if(iv.gt.0) vx(iv)=1.0d0
    end
c --- --- Save Vector -----
subroutine savevec(n1,vx,nam)
implicit real*8 (a-h,o-z), integer*4 (i-n)
dimension vx(n1)
character*12 nam
open(2,FILE=nam,FORM='UNFORMATTED')
write(2) (vx(j),j=1,n1)
close(2)
end
c --- --- Load Vector -----
subroutine loadvec(n1,vx,nam)
implicit real*8 (a-h,o-z), integer*4 (i-n)
dimension vx(n1)
character*12 nam
open(2,FILE=nam,FORM='UNFORMATTED')
read(2) (vx(j),j=1,n1)
close(2)
end
c --- --- Berechnung des kleinsten EW mit Lanczos (Gagliano) -----
c --- Inp: iv Out: v0,itr,ew Wrk: v1
subroutine lanz1(n,n1,n2,nev,v0,v1,v2,ew,iv,itr,l1,l2)
implicit real*8 (a-h,o-z),integer*4 (i-n)
dimension v0(n1),v1(n1),v2(n1),ew(4),l1(n1),l2(2,0:n2)
eps=1d-10 ! EPS=0.00005 WAR STANDARD FUER 12 SPINS !!
write(*,'(A,e8.1)') ' subroutine lanz1 eps=',EPS
itr=nev+iv ! no operation
Itr=0
2  ew2=ew1
   itr=itr+1
c   if(n1.lt.9) write(*,'(3H i=,I3,5H   v=,8f10.6)') itr,v0
   call hamilton(v0,v1,n,n1,n2,l1,l2)
   call hamilton(v1,v2,n,n1,n2,l1,l2)
   H =0.
   HH =0.
   HHH=0.
   DO4I=1,N1
     R =v0(I)
     H =H+R*v1(I)
     HH =HH+R*v2(I)
     HHH=HHH+v1(I)*v2(I)
4  CONTINUE
   B =DSQRT(HH-H*H)
   IF(B.LT.1.E-6) GOTO 6
   F =(HHH-3.*H*HH+2.*H*H*H)/(2.*B*B*B)

```

```

ALF=F-SQRT(1.+F*F)
EW1=H+B*ALF          ! ????????
A  =1./SQRT(1.+ALF*ALF)
AK  =A*ALF/B
DO5,I=1,N1
5  v0(I)=(A-AK*H)*v0(I)+AK*v1(I)
6  ew1=h
c   write(*,'(3H I=,I3,5H  ew=,F12.7,3H t=,I5)') itr,h,itime()
c   write(*,'(3H I=,I3,12H H,HH,HHH,B=,4E10.3)') itr,h,hh,hhh,b
      if(itr.lt.2) goto 2
      er =ew1
      if(er.eq.0) er=1.0
      IF(ABS((ew2-ew1)/er).GT.EPS) GOTO 2  ! Iteration beenden ?
c   if(n1.lt.9) write(*,'(3H i=,I3,5H  v=,8f10.6)') itr,v0
      ew(1)=ew1
      END
c --- --- Berechnung der kleinsten EW mit Lanczos -----
c --- Inp: iv  Out: a,b,itr,ew  Wrk: v0,v1
      subroutine inc1(n,n1,n2,nev,v0,v1,v2,ew,iv,itr,l1,l2)
      implicit real*8 (a-h,o-z),integer*4 (i-n)
      dimension v0(n1),v1(n1),v2(n1),ew(4),et(4),l1(n1),l2(2,0:n2)
      common/vecdat/a(150),b(150),c(150,5)
      write(*,'(A)') ' subroutine inc1'
      i=iv+nev          ! NO OPERATION (dummy)
      call clrvec(n1,v0,0)      ! Initialisierung
      eps=1.d-10             ! Genauigkeit fuer bisec
      do 100,i=1,150         ! Iteration
      call mltply(v1,v0,v2,pd,n,n1,n2,l1,l2) ! pd=<v1*H*v1>,v0=H*v1+v0,v2=H*v1
      a1=pd
      a(i)=a1
      b1=0.0d0
      do 50,j=1,n1
50  b1=b1+(v0(j)-a1*v1(j))**2
      b1=sqrt(b1)
      b(i)=b1
      itr=i
      if(n1.lt.9.or.(i.ge.20.and.mod(i,5).eq.0))then ! Konvergenztest
      call bisec(a,b,i,4,eps,ew)
      write(*,'(4H ew=,4d14.7,3H i=,I3,3H t=,I6,1Hs)') ew,i,itime()
      if(i.eq.20.or.(n1.lt.9.and.i.eq.1)) goto 20
      r1=ew(1)
      if(abs(r1).lt.0.5d-10) r1=1.0
      if(abs((et(2)-ew(2))/r1).gt.1.0d-10) goto 20  ! oder ew(1,3,4)
18  if(nev.gt.0) call vec12(i,nev,ew)
      return
20  do 22,j=1,4
22  et(j)=ew(j)

```

```

    if(i.ge.150) then
      write(*,'(A)') ' keine Konvergenz nach 150 Schritten'
      goto 18                                ! Abbruch (goto 100)
    endif
  endif
endif
if (b1.le.0.5d-20) then                      ! 0.5d-30 division durch 0
  write(*,'(A,I3)') ' b(i) zu klein bei i=',i
  call bisec(a,b,i,1,eps,ew)
  goto 18                                    ! stop 'Fehler bei Iter.'
endif
do 110 j=1,n1
  t1=v1(j)
  t2=(v0(j)-a1*v1(j))/b1
  v0(j)=-b1*t1
  v1(j)=t2
110 continue
100 continue
end

c ---- -----
c --- Inp: a,b=Tridiagonalmatrix(TM),nn=dimension der matrix
c ---      ne=anzahl zu berechnender ew,ep=genauigkeit
c --- Out: ew  Eigenwerte der TM
      subroutine bisec(a,b,nn,ne,ep,ew)
      implicit real*8 (a-h,o-z),integer*4 (i-n)
      dimension a(nn),b(nn),b2(2000),ew(4)
      if(nn.gt.2000) stop 'Fehler bisec.nn>2000'
c      if(ne.gt.nn) stop 'Fehler bisec.nn<ne'
      range=abs(a(1))+abs(b(1))             ! bound initiieren
      do 10,k=2,nn-1
10     range=max(range,abs(b(k-1))+abs(a(k))+abs(b(k)))
      range=max(range,abs(b(nn-1))+abs(a(nn)))
      range=-range
      b2(1)=0.d0
      do 20,i=2,nn
20     b2(i)=b(i-1)**2
      epa=abs(range)*ep
      do 30,i=1,ne
30     ew(i)=-range
      rb=range
      do 100,k=1,ne                          ! bisection methode
      ra=ew(k)
      do 110,j=1,100
      rc=(ra+rb)/2.d0
      if(abs(ra-rb).lt.epa)goto 100
      numneg=0
      g=1.d0
      ipass=0

```

```

do 120,i=1,nn
  if(ipass.eq.0)then
    g=rc-a(i)-b2(i)/g
  else if(ipass.eq.1)then
    ipass=2
  else
    g=rc-a(i)
    ipass=0
  endif
  if(ipass.eq.0)then
    if(g.le.0.d0)numneg=numneg+1
    if(abs(g).le.abs(b2(i)*epa*ep))ipass=1
  endif
120  continue
  numneg=nn-numneg
  if(numneg.lt.k)then
    rb=rc
  else
    ra=rc
    do 130,i=k,min(numneg,ne)
130  ew(i)=rc
    endif
110  continue
100  continue
end

c --- -----
c --- EVs einer Tridiagonalen Matrix mit inverser Iteration
c --- nn i Matrixgroesse
subroutine vec12(nn,nev,ew)
implicit real*8 (a-h,o-z),integer*4 (i-n)
dimension ew(4)
common/vecdat/a(150),b(150),c(150,5)
common/wrk/di(2000),bl(2000),bu(2000),bv(2000),cm(2000),lex(2000)
if(nn.gt.2000) stop 'Fehler vec12.nn>2000'
do 10,k=1,nev
  do 100,j=1,nn
    di(j)=ew(k)-a(j)
    bl(j)=-b(j)
    bu(j)=-b(j)
100  continue
  do 110 j=1,nn-1          ! LU decomposition
    if(abs(di(j)).gt.abs(bl(j)))then
      lex(j)=0          ! non pivoting
      if(abs(di(j)).lt.1.d-40)di(j)=1.d-40
      cm(j+1)=bl(j)/di(j)
      di(j+1)=di(j+1)-cm(j+1)*bu(j)
      bv(j)=0.d0

```

```

else                                     ! pivoting
  lex(j)=1
  cm(j+1)=di(j)/bl(j)
  di(j)=bl(j)
  s=bu(j)
  bu(j)=di(j+1)
  bv(j)=bu(j+1)
  di(j+1)=s-cm(j+1)*bu(j)
  bu(j+1)= -cm(j+1)*bv(j)
end if
110 continue
if(abs(di(nn)).lt.1.d-40)di(nn)=1.d-40
do 120 j=1,nn                           ! initial vector
120 c(j,k)=1.d0/(j*5.d0)
if(k.eq.1)then                          ! degeneracy check up
  km=k                                  ! -> Entartung?
else if(abs(EW(k)-EW(km)).gt.1.d-13)then
  km=k
else
  do 130 i=km,k-1
    prd=0.d0
    do 140 j=1,nn
140     prd=prd+c(j,i)*c(j,k)
    do 150 j=1,nn
150     c(j,k)=c(j,k)-prd*c(j,i)
130     continue
end if
do 160 l=1,k-km+3                       ! inverse iteration
  if((l.ne.1).or.(k.ne.km))then
    do 170 j=1,nn-1                     ! forward substitution
      if(lex(j).eq.0)then
        c(j+1,k)=c(j+1,k)-cm(j+1)*c(j,k)
      else
        s=c(j,k)
        c(j,k)=c(j+1,k)
        c(j+1,k)=s-cm(j+1)*c(j,k)
      end if
170     continue
end if
do 180 j=nn,1,-1                        ! backward substitution
  s=c(j,k)
  if(j.le.nn-1)s=s-bu(j)*c(j+1,k)
  if(j.le.nn-2)s=s-bv(j)*c(j+2,k)
  c(j,k)=s/di(j)
180 continue
dnorm=0.d0                              ! normalisation
do 190 j=1,nn

```

```

190     dnorm=dnorm+c(j,k)**2
        if(dnorm.gt.1.d-50)dnorm=1./sqrt(dnorm)
        do 200 j=1,nn
200     c(j,k)=c(j,k)*dnorm
160     continue
10     continue
        end
c --- -----
c --- eigenvector by the Lanczos method
c --- Inp: a,b,c
c --- Out: v0,v1,ev
        subroutine incv1(n,n1,n2,nev,v0,v1,v2,iv,itrv,ev,l1,l2)
        implicit real*8 (a-h,o-z),integer*4 (i-n)
        dimension v0(n1),v1(n1),ev(n1,nev),l1(n1),l2(2,0:n2)
        common/vecdat/a(150),b(150),c(150,5)
        write(*,'(A)') ' subroutine incv1'
        i=iv+nev ! NOP
        call clrvec(n1,v0,0) ! Initialisierung
        do 20 k=1,nev
        do 20 j=1,n1
20     ev(j,k)=c(1,k)*v1(j)
        do 100,i=1,itrv-1 ! Iteration
            call mltply(v1,v0,v2,pd,n,n1,n2,l1,l2) ! alpha,beta
            a1=a(i)
            b1=b(i)
            do 40 k=1,nev
            do 40 j=1,n1
40     ev(j,k)=ev(j,k)+c(i+1,k)*(v0(j)-a1*v1(j))/b1
            if(i.eq.itrv-1) goto 100
            do 50 j=1,n1
                t1=v1(j)
                t2=(v0(j)-a1*v1(j))/b1
                v0(j)=-b1*t1
                v1(j)=t2
50     continue
100    continue
            do 200 k=1,nev ! Normalisierung
                dnorm=0.d0
                do 210 j=1,n1
210     dnorm=dnorm+ev(j,k)**2
                dnorm=1/sqrt(dnorm)
                do 220 j=1,n1
220     ev(j,k)=ev(j,k)*dnorm
200    continue
        end
c --- -----
c --- ***** check of the eigenvector and eigenvalue *****

```

```

      subroutine check1(n,n1,n2,v0,v1,l1,l2)
c --- Inp: v0  eigenvector to be checked
c --- Out: ... rq  (H*v0-h*v0)**2
      implicit real*8 (a-h,o-z),integer*4 (i-n)
      dimension v1(n1),v0(n1),l1(n1),l2(2,0:n2)
      dimension ru(16),rd(16),rs(16)
      common/vecdat/a(150),b(150),c(150,5) /operator/io
      write(*,'(A)') ' subroutine check1'
      write(*,'(A8,6d10.3)') 'x=',(v0(j),j=1,min(n1,6))
c --- <H> -----
      rh=0.0
      rq=0.0
      call hamilton(v0,v1,n,n1,n2,l1,l2)
      do 30,i=1,n1
30    rh=rh+v1(i)*v0(i)
      do 40,i=1,n1
40    rq=rq+(v1(i)-rh*v0(i))**2
      write(*,'(2(A8,1pd16.8))') '<H>=',rh,'rq=',rq ! Fehler=(H*x-h*x)^2
c --- <SS> -----
      rh=0.0
      rq=0.0
      i1=io
      io=2
      call hamilton(v0,v1,n,n1,n2,l1,l2)
      io=i1
      do 130,i=1,n1
130   rh=rh+v1(i)*v0(i)
      do 140,i=1,n1
140   rq=rq+(v1(i)-rh*v0(i))**2
      write(*,'(2(A8,1pd16.8))') '<SS>=',rh,'rq=',rq
c --- <SiSj> -----
      do 280,i2=1,1          ! n (eine zeile genuegt bei symmetrie)
      do 290,i1=1,n
      rs(i1)=0.0
      call opsisj(i2-1,i1-1,v0,v1,n,n1,n2,l1,l2)
      do 230,i=1,n1
230   rs(i1)=rs(i1)+v1(i)*v0(i)
290   continue
      write(*,'(A8,16f8.5)') ' <SiSj>=',(rs(i1),i1=1,n)
280   continue
c --- <n> Aufenthaltswahrscheinlichkeiten -----
      do 200,k=1,n
      ru(k)=0.0
      rd(k)=0.0
      do 190,i=1,n1
      if(btest(l1(i),2*k-2)) ru(k)=ru(k)+v0(i)**2
      if(btest(l1(i),2*k-1)) rd(k)=rd(k)+v0(i)**2

```

```
190  continue
200  continue
    write(*,'(A8,16f8.5)') '<nu>=',(ru(i),i=1,n)
    write(*,'(A8,16f8.5)') '<nd>=',(rd(i),i=1,n)
    end
```

c -----

Literatur

- [1] H.Ibach, H.Lüth: Festkörperphysik, Einführung in die Grundlagen, Springer-Verlag, Berlin Heidelberg 1990
- [2] E.Müller-Hartmann: Überlegungen zur Theorie der Hochtemperatursupraleiter, Institut für Theoretische Physik, Universität zu Köln, aus 19. IFF-Ferienkurs in der Kernforschungsanlage Jülich 1988
- [3] J.Hubbard: Proc. Roy. Soc. A276 (1963) 238
- [4] J.Richter: Zero-temperature magnetic ordering in the inhomogeneously frustrated quantum Heisenberg antiferromagnet on a square lattice, PRB Vol.47, No.10, pp.5794-5804 (1993)
- [5] T.Barnes, A.E.Jacobs, M.D.Kovarik, W.G.Macready: Detailed Lanczos study of one- and two-hole band structure and finite-size effects in the t-J model, Physical Review B (PRB) Vol.45, No.1, pp.256-264 (1992)
- [6] E.Dagotto, R.Joynt, A.Moreo, S.Bacci, E.Gagliano: Strongly correlated electronic systems with one hole: Dynamical properties, PRB Vol.41, No.13, pp.9049-9072 (1990)
- [7] D.Poilblanc, E.Dagotto, J.Riera: Cluster study of the t-J model: Chiral fluctuations, PRB Vol.43, No.10, pp.7899-7911 (1991)
- [8] J.Bonca, P.Prelovsek, I.Sega: High-Doping Regime in a Planar Model for Strongly Correlated Electrons, Europhys. Lett., 10(1), pp.87-92 (1989)
- [9] J.Bonca, P.Prelovsek: Next neighbour hopping of holes in the planar effective model for strongly correlated systems, Solid State Communications, Vol.71, No.9, pp.775-759 (1989)
- [10] A.Mistriotis, H.Büttner, W.Pesch: Holes in finite electron systems with strong correlation, Phys.: Condens. Matter 1(1989)891-900 UK
- [11] A.Mistriotis, H.Büttner, W.Pesch: Ground-state properties of strongly correlated electrons on finite clusters, Phys. C: Solid State Phys. 21(1988) L1021-L1026, UK
- [12] H.Fehske, V.Waas, H.Röder, H.Büttner: On the possibility of phase separation in the extended hubbard model, Solid State Communications, Vol.76, No.12, pp.1333-1336 (1990) Graet Britain
- [13] H.Röder, V.Waas, H.Fehske, H.Büttner: Holes in a two-dimensional Hubbard antiferromagnet, PRB Vol.43, No.7, pp.6284-6287 (1991)
- [14] Nolting: Viel-Teilchen-Theorie, Grundkurs theoretische Physik, Verlag Zimmermann-Neufang, Ulmen 1991
- [15] H. Nishimori: Titpack Ver. 2, User's Manual, February 1991

- [16] Robert Sedgewick: *Algorithmen*, Addison-Wesley Publishing Company, 1988
- [17] H.Q.Lin: Exact diagonalization of quantum-spin models, PRB Vol.42, No.10, pp.6561-6567 (1990)
- [18] J.K.Cullum, R.A.Willoughby: *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Vol.1 Theory*, Birkhäuser 1985