Thesis for the degree of Master of Science in Physics

HUBBARD MODELS

States and Transformations

Andreas Enqvist

Institute for Theoretical Physics
Chalmers University of Technology
and
Göteborg University

May 10, 2005

HUBBARD MODELS

States and Transformations

Andreas Enqvist
Institute for Theoretical Physics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden

Abstract

Condensed matter physics aims to describe macroscopic properties of materials by trying to model what the material looks like at atomic levels and electronic properties are often of interest. One common model for describing the electrons and their motion and interaction is the Hubbard model that describes both local atomic interaction and transport of the electrons. The model now have many forms and can be used for describing Mott-insulators, superconductivity and other fields.

One of the problems with calculations on these small scales are the very large number of particles we need to account for to have a sample of macroscopic size, the Mott insulator is such an example. To counter this problem many approximate methods have been created and we will investigate a particular such method.

We begin by looking at the spin 1/2 and spin 3/2 Hubbard models to verify previously published results. The calculations will then be applied to the 2-band Hubbard model. To be able to do the calculations when the model is half filled we will search for a transformation that takes us from the half filled case to vacuum, which bring the system from being a dense to a dilute Fermi gas (DFG). When we map the system to a dilute Fermi gas we make it possible to use standard methods to make realistic physical predictions for the system.

The Hubbard model is formulated as a second quantized Hamiltonian with quantum mechanical creation and destruction operators. To do the calculations, containing operators that obey fermion algebra, we will use the symbolic handling software Mathematica. This makes the complicated fermion algebra possible which is necessary to perform the mapping from the half filled case to vacuum. When we perform the mapping a new "Hamiltonian" is generated with different parameters. This results in a new effective Hamiltonian.

Acknowledgments

First I would like to thank my supervisor Stellan Östlund for taking me on and guiding me within the field. I also thank you for your help and patience even when the progress was delayed. My work could not have been done without the help of your deep knowledge and understanding.

Further I want to thank my former classmates and friends that has helped me during my education and at other times. Especially Christian, Jens and Jakob deserves mentioning for their nice company and discussions.

Finally I want to express my deep gratitude towards my other friends and my family for their support and love.

Contents

1

Introduction of the General Hubbard Model

The study of strongly correlated electrons emerged in the 1930's. Physicists noted that the conductivity data for some materials (transition metal oxides) showed them to be insulating. But the reigning theory at that time, the Bloch theory, suggested they would be metals. Counting the number of electrons in the outermost band gave a partially filled band so the material should be conducting according to Bloch theory.

As a solution to this it was proposed that Coulomb repulsion between band electrons might cause a barrier for tunneling between sites and Hubbard suggested his model as a simple paradigm for the physics. The Coulomb repulsion would then be responsible for electron localization and the failure of Bloch wave theory in this particular case. Therefore it became necessary to formulate theories for strongly correlated electrons.

Let us try and give a credible ground for the Hubbard model based on a gedankenexperiment [1]. Within condensed matter physics we learn that a materials' properties often have their explanation in the electronic structure of the material. Let's take a simple metal such as sodium (Na) which has one atom per unit cell. Using the tight binding model we can conclude that only the last (outermost) electron has a band of sizeable width. This is due to the fact that the deeper lying bands all are completely filled, while the 3s-band is half filled. Because of this we expect Na to be a metal.

The conclusion that Na is a metal can be drawn independently of what the lattice constant happens to be. Imagine now then that we increase the inter atomic spacing. We would predict that the sodium should still be a metal. When we increase the lattice constant the 3s-band becomes

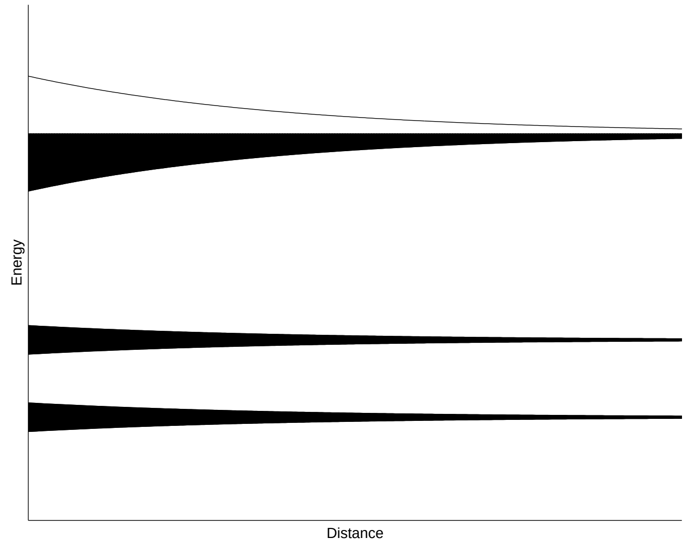


Figure 1.1: Here we see the bands as they depend on the distance away from the center of the atom.

narrower, but it remains half filled.

Thinking about this situation we realize that eventually this leads to an absurdity, since when the atomic spacing get large the system will behave as an array of individual sodium atoms. This means that the sample is then a non-conductor.

So why then is a widely spaced array of sodium atoms an insulator? For the material to be an electrical conductor the electrons need to move between sites.

$$\begin{array}{ccccccc} |\uparrow\rangle & + & |\downarrow\rangle & \longrightarrow & |0\rangle & + & |\uparrow\downarrow\rangle \\ \\ \mathrm{Na} & + & \mathrm{Na} & \longrightarrow & \mathrm{Na}^{+} & + & \mathrm{Na}^{-} \end{array}$$

The associated energies for this event are easily attained. In the first step the system has energy $\epsilon_{3s} + \epsilon_{3s}$ and in the second step we have the energy $0 + 2\epsilon_{3s} + U_{3s}$, where U_{3s} is the Coulomb energy coming from the Coulomb repulsion when the two electrons occupy the same site.

$$U_{3s} = \int d\vec{r_1} \int d\vec{r_2}\ |\phi_{3s}\left(\vec{r_1}\right)|^2 \frac{e^2}{|\vec{r_1} - \vec{r_2}|}\ |\phi_{3s}(\vec{r_2})|^2 \tag{1.1}$$

For the hopping to occur there needs to be an energy investment U_{3s}. This interactomic Coulomb energy, which will reappear frequently in the coming Hubbard models, is then for larger lattice constants stopping the metallic conduction. When this happens we say that the system is undergoing a Mott transition [2].

Having seen the utility of the Hubbard model to explain the plausible physics, we can now start using it for the same purpose it once was created for; to explain the failures of single electron models. The model owes its name to J. Hubbard who developed the model [3]. It was also independently investigated by M.C. Gutzwiller [4] and Kanamori [5] at the same time. The Hubbard model is a model in which band electrons interact via a local two-body interaction which mimics the short range behavior of the Coulomb repulsion. The model is a simplification of the many body interaction system and does not contain phonons. In general no attractive interactions are included.

This is in contrast to superconductivity which in most cases (eg BCS) is associated with the result of attractive interactions. Originally the Hubbard model was the simplest model for describing a Mott insulator. Though more recently superconductivity of high T_c materials have been thought to originate from purely repulsive interactions and the effect of doping an otherwise insulating state. Many attempts to use the model for high T_c have been tried.

The main simplification of the Hubbard model is that we imagine that we need only to deal with states in one band (or a few bands in multi-band Hubbard models). A Hamiltonian [6] describing this system is

$$\begin{aligned} H &= -\sum_{r_i,r_j,\sigma} \Big(c^\dagger_\sigma(r_i) t_{ij} c_\sigma(r_j) + c^\dagger_\sigma(r_j) t_{ij} c_\sigma(r_i) \Big) + \\ & \quad + \frac{1}{2} \sum_{i,j,i',j',\sigma} U_{ij,i'j'} \, c^\dagger_\sigma(r_i) \, c^\dagger_\sigma(r_j) \, c_\sigma(r_{j'}) \, c_\sigma(r_{i'}) \end{aligned} \tag{1.2}$$

with $c^\dagger_\sigma(r_i)$ $(c_\sigma(r_i))$ being the creation (annihilation) operators that creates (annihilates) an electron with spin σ at site r_i. Note that we here only deal with one band so no band index is needed on our operators. The operators obey Fermi anti-commutation rules as:

$$\begin{aligned} \{c_\sigma(r_i) \, , \, c^\dagger_{\sigma'}(r_j)\} &= \delta_{\sigma\sigma'} \, \delta_{r_i \, r_j} \\ \{c_\sigma(r_i) \, , \, c_{\sigma'}(r_j)\} &= 0. \end{aligned} \tag{1.3}$$

The Hubbard model is an approximation to Eq. (1.2) which restricts the hopping to nearest neighbours:

$$t_{ij} = \begin{cases} t & \text{if nearest neighbours} \\ 0 & \text{otherwise.} \end{cases} \tag{1.4}$$

Since the Coulomb interaction decays as the distance increases we only keep the on-site term ie $U_{ij,i'j'} = U\,\delta_{ij}\,\delta_{ij'}\,\delta_{ii'}$, giving us the one band Hubbard Hamiltonian

$$H = -t \sum_{\langle i,j \rangle,\sigma} \Big(c^\dagger_\sigma(r_i)\, c_\sigma(r_j) + h.c. \Big) + U \sum_i n_{\uparrow,i} n_{\downarrow,i} \tag{1.5}$$

where $\langle i, j \rangle$ indicates nearest neighbour sites and $n_{\sigma,i} = c^\dagger_\sigma(r_i) c_\sigma(r_j)$ is the number operator. The number of particles on a site depends on what type of particles they are. Electrons for example can be 0, 1 or 2 per site which fluctuates by the hopping term.

As can be seen the Hubbard Hamiltonian consists of two parts, a hopping part H_{hop} and an interaction part H_{int}. The model is itself very simple and the two components alone can be easily solved. The sum $H_{hop} + H_{int}$ however exhibits numerous non-trivial phenomena, such as Mott-transitions, magnetism and even superconductivity. That the simple model may give explanations to things as diverse as these might be realized if we look at what type of properties the solutions to the different components alone contain. The H_{hop} implies that the electrons behave as waves in a non-interacting Hubbard model, while the solutions to H_{int} implies that the electrons behave as particles [7].

The combined Hubbard Hamiltonian $H_{hop} + H_{int}$ is then something that is consistant with the particle-wave duality in quantum physics and therefore be a good candidate for describing the many fields in which it is currently deployed. We notice that the parts H_{hop} and H_{int} have very different properties, meaning effectively that although the parts may be simple it is indeed very difficult to find the properties of their sum.

Mott insulators may appear as an effect of the on-site potential term in the Hubbard model H_{int}. If the interaction is large compared to the hopping H_{hop}, this can result in a localization of the electrons when the model is half filled. If the system is half filled with one electron per site, a large interaction term will prevent any electron from moving to a different lattice site since that site will then be doubly occupied. This can result in a Mott insulator. There could in principle be metal-insulator transitions in our system if the number of particles is varied around the half filled case where the material displays insulating properties.

We want to study a few Hubbard models to see in what case we may perform a canonical transformation from the half filled ground state in that model to the vacuum. The reason for wanting to do this transformation is that in the half filled case we deal with a number of particles, usually electrons, that is comparable to the number of lattice points. This means that exact calculations for larger structures (crystals), to get macroscopic results, will be more or less impossible unless we use some simplified methods.

A well investigated area where many such methods exist is the area of Dilute Fermi Gas (DFG). The problem is that in the current state DFG theories can not be applied to our half filled Hubbard models since they are far from diluted electron gases. If we on the other hand map the ground state at half filling onto the vacuum and then consider the effects of slight doping, that doping now results in a diluted Fermi gas.

We will begin by looking at the spin 1/2 model. Since the later models will quickly become more complicated we start by considering the simplest case of spin 1/2. The calculations and ideas will then be easily introduced to be used later on in the more advanced models.

An important thing to remember is that we are searching for transformations that take us from half filling to the vacuum. In the spin 1/2 model this would mean going from a 1-particle state to a 0-particle state. It turns out that such a transition does not exist, because we cannot accomplish this locally without violating the Fermion algebra between different sites. For fermions we have the the anti-commutation rules (Eq. 1.3) and a transformation from half filling to the vacuum will transform the operators in the model as $c^{\dagger}(r_i) \rightarrow c^{\dagger}_{\sigma}(r_i)\, c_{\sigma}(r_i) + \ldots$

Operators of odd length can get terms of even length, but then the anti-commutation relations will turn into commutation relations.

$$\{c^{\dagger}_{\sigma}(r)\,,\, c_{\sigma'}(r')\} = 0 \quad \Longrightarrow \quad \left(c^{\dagger}_{\sigma}(r) \rightarrow c^{\dagger}_{\sigma}(r)\, c_{\sigma}(r)\right) \quad \Longrightarrow$$
$$\Longrightarrow \quad \{c^{\dagger}_{\sigma}(r)\, c_{\sigma}(r)\,,\, c_{\sigma'}(r')\} \neq 0.$$

This is due to the fact that odd and even length expressions of operators at different sites commute, so we consider more complicated models where a transformation from the half filled case to the vacuum is possible. If the states of the model at half filling have an even number of particles, then the fermion algebra is conserved under our transformation.

The simplest model that meets this criteria, that may also have a suitable ground state at half filling, is the spin 3/2 model. With four different creation operators the half filled states will be 2-particle states. Lastly we will also look at another model which have half filled states with two particles, namely the 2-band Hubbard model.

2

A spin-1/2 system

The main aspect of this paper is to look at a canonical transformation in more advanced Hubbard models. This will be done with help of computers and the use of the symbolic handling software Mathematica [8]. To get familiar with what needs to be done we shall in this chapter look at a simpler system in which the calculations could be done by hand.

The system we consider will be one compromised of only one point, which can be occupied by spin 1/2 particles. The Hilbert space for this one point consists of four states, corresponding to an empty state, a doubly occupied state and two singly occupied states with spin up or down. We can denote these states as follows $\Psi_1 = \left|0\right\rangle, \Psi_2 = \left|\uparrow\downarrow\right\rangle, \Psi_3 = \left|\uparrow\right\rangle, \Psi_4 = \left|\downarrow\right\rangle$.

We will now create a matrix $M_{i,j}$ which obeys $M_{i,j}\Psi_k = \Psi_i\delta_{j,k}$ [9]. To do this analytically we explore a few simplifying aspects of the matrix. Firstly the top row $M_{1,k}$ maps the state Ψ_k precisely to the vacuum. Therefore the element $M_{1,k}$ is a product of creation operators ($c_\sigma^\dagger$), that raises the state Ψ_k to the fully occupied state followed by the operator which lowers the state to the vacuum. With creation and destruction operators ($c_\sigma^\dagger, c_\sigma$) we can now easily write down the element $M_{1,1}$ following the formula above, $M_{1,1} = c_\uparrow c_\downarrow c_\uparrow^\dagger c_\downarrow^\dagger$.

As for the first column we can get that from the relation $M_{k,1} = (M_{1,k})^\dagger$. The rest of the matrix can be created from the relation $M_{i,j} = M_{i,1}M_{1,j}$. Finally we apply normal ordering to the elements of the matrix, such that we move destruction operators to the right and creation operators to the left using fermion anti-commutation rules. The matrix M can be seen in Table (2.1).

With the help of matrix M we can now find a way to express polynomials of Fermi operators as matrices and also convert transformation

$$\begin{pmatrix} 1-n_\uparrow-n_\downarrow+n_\uparrow n_\downarrow & c_\uparrow c_\downarrow & (1-n_\downarrow)\,c_\uparrow & (1-n_\uparrow)\,c_\downarrow \\ c_\uparrow^\dagger c_\downarrow^\dagger & n_\uparrow n_\downarrow & -c_\downarrow^\dagger n_\uparrow & c_\uparrow^\dagger n_\downarrow \\ c_\uparrow^\dagger(1-n_\downarrow) & -n_\uparrow c_\downarrow & n_\uparrow(1-n_\downarrow) & c_\uparrow^\dagger c_\downarrow \\ c_\downarrow^\dagger(1-n_\uparrow) & n_\downarrow c_\uparrow & c_\downarrow^\dagger c_\uparrow & n_\downarrow(1-n_\uparrow) \end{pmatrix}$$

Table 2.1: The M-matrix. $n_{\uparrow,\downarrow}$ is the number operator ($n_\uparrow = c_\uparrow^\dagger c_\uparrow$).

matrices (4×4) into corresponding Fermi polynomials.

The routine for turning a matrix into a Fermi polynomial will be referred to as M2F (Matrix to Fermion), likewise the opposite routine will be called F2M. To create these routines we need to create a vector (V) containing all possible fermionic terms. These will build our 16-dimensional space, built from the combination of the two 4-dimensional spaces of creation and destruction operators.

We now construct a 16×16 matrix U by using our M-matrix, which we flatten into a vector $M_{i,j} \to M'_k$, $k = 1, \ldots, 16$.

$$U_{i,j} = \frac{dM_j}{dV_i}. \tag{2.1}$$

With these vectors and matrices we can now build our routines as

$$M2F[m_{i,j}] = V \cdot U \cdot (m'_{\hat{ij}}), \tag{2.2}$$

where $m'_{\hat{ij}}$ indicates that we flatten the matrix $m_{i,j}$ to use one vector index $_{\hat{ij}}$. Similarly we have

$$F2M[p] = U^{-1} \cdot \frac{dp}{dV}, \tag{2.3}$$

where F2M[p] gives a vector of length 16. So to get the desired matrix we simply partition it by 4. With index notation our routines read

$$M2F[m'_j] = V_i U_{i,j} (m'_j) \tag{2.4}$$

$$F2M[p] = \left[U^{-1}\right]_{l,k} \frac{dp}{dV_k} \tag{2.5}$$

with summation over repeated indices being implied.

We know that since the two routines perform the opposite transformation, a double application should return the original expression ie $F2M[M2F[m]] = m$ as well as $M2F[F2M[p]] = p$. In fact we can use this property to convince ourselves that the routines are set up properly.

$$M2F[F2M[p]] = V_i\, U_{i,j}\, \left[U^{-1}\right]_{j,k} \frac{dp}{dV_k} = V \cdot \underbrace{U \cdot U^{-1}}_{\mathbf{1}} \cdot \frac{dp}{dV} = \underbrace{V \cdot \frac{dp}{dV}}_{\mathbf{1}} = p \tag{2.6}$$

The last equality depends on our fermionic system since that only have linear occurrence of any creation or destruction expression, since $c_\sigma \cdot c_\sigma = 0$, $c^\dagger_\sigma \cdot c^\dagger_\sigma = 0$ for any σ. Lastly we check the application of the routines in the opposite order

$$F2M\left[M2F[m]\right] = U^{-1} \cdot \frac{d}{dV}\left[V \cdot U \cdot m\right] = U^{-1} \cdot U \cdot m = m. \tag{2.7}$$

One of the main reasons for creating these routines is to be able to go from long and tedious fermionic algebra to matrix algebra which is computationally favourable. A way of illustrating this is to take the general anti-commutation relation $\{c^\dagger_\sigma\,,\, c_{\sigma'}\} = \delta_{\sigma,\sigma'}$. If we use our transformation routine to go from fermionic expressions to matrices then $c^\dagger_\sigma c_\sigma + c_\sigma c^\dagger_\sigma$ should return $\mathbf{1}$ in matrix form, which in this model means the 4×4 unity matrix.

2.1 Finding Commuting Operators

Finding all possible operators that commutes with a certain operator or expression is very straight forward in the matrix representation. Assume that we want to identify all expressions that commutes with the L_z-operator which in fermionic creation and destruction operators is expressed as

$$L_z = \frac{1}{2}\left(n_\uparrow - n_\downarrow\right) = \frac{1}{2}\left(c^\dagger_\uparrow c_\uparrow - c^\dagger_\downarrow c_\downarrow\right). \tag{2.8}$$

To find the commuting operators with respect to L_z we transform it to matrix form with $L_{z_{mat}} = F2M[L_z]$. Secondly we need a matrix B, which we will use to find the commutating operators. The matrix B is formed as follows

$$\begin{pmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{pmatrix}$$

Now we can solve the commutation equation $B \cdot L_{z_{mat}} - L_{z_{mat}} \cdot B = 0$ which now is a simple matrix equation easily solved by hand, or for a larger system by computer, with linear algebra. The solution will be a number of conditions on the different $B_{i,j}$'s. Substituting these conditions onto the original B shows us which operators that commute with L_z (the ones that are eliminated doesn't commute). In our case of L_z this results in a matrix B_{sol} as follows

$$\begin{pmatrix} B_{11} & B_{12} & 0 & 0 \\ B_{21} & B_{22} & 0 & 0 \\ 0 & 0 & B_{33} & 0 \\ 0 & 0 & 0 & B_{44} \end{pmatrix}$$

Remembering that our basis is $\{\Psi_1 = |0\rangle \,, \Psi_2 = |\uparrow\downarrow\rangle \,, \Psi_3 = |\uparrow\rangle \,, \Psi_4 = |\downarrow\rangle\}$, enables us to translate into the operators that we looked for

$$\begin{aligned} B_{12} &\Rightarrow c_\uparrow c_\downarrow \\ B_{21} &\Rightarrow c_\uparrow^\dagger c_\downarrow^\dagger \\ B_{22} &\Rightarrow c_\uparrow^\dagger c_\downarrow^\dagger c_\uparrow c_\downarrow \Rightarrow n_\uparrow n_\downarrow \\ B_{33} &\Rightarrow c_\uparrow^\dagger c_\uparrow \Rightarrow n_\uparrow \\ B_{44} &\Rightarrow c_\downarrow^\dagger c_\downarrow \Rightarrow n_\downarrow . \end{aligned}$$

The trivial solution is also present with B_{11}. But that is just the same as the relation $B_{11}L_z - L_z B_{11} = 0$ when B_{11} is the identity operator, which in this relation can be interpreted as L_z commutes with ordinary scalars.

Having gone through the basic transformations and calculations in a simple model, we will now be in a good position to understand what comes next when we look at more advanced systems. The main reason for the following chapters are to find transformations for the models that maps especially the half filled case to a vacuum state. Since these transformations are unitary it is then easy to continue the calculations and analysis beyond this report by applying approximate mean field theory methods to the full Hilbert space to further analyze the model. In particular this method may be used to study the doped Mott insulator.

3

Spin 3/2 Hubbard Model

We continue by considering the spin 3/2 Hubbard model. The Hamiltonian looks as follows,

$$H = H_{hop} + H_{Hubb} + H_J. \tag{3.1}$$

Where H_{hop} is the hopping term:

$$H_{hop} = -t \sum_{<r,r'>,\sigma} (c^{\dagger}_{r,\sigma} c_{r',\sigma} + h.c.). \tag{3.2}$$

The Hubbard term H_{Hubb} for this model will have an arbitrary chemical potential absorbed in U and take on the form [10]:

$$H_{Hubb} = U \sum_{r} n_r \left(n_r - 2 \right) \tag{3.3}$$

where n_r is our local particle number at site r, $n_r = \sum_\sigma c^{\dagger}_{r,\sigma} c_{r,\sigma}$ The “2” is introduced since it preserves the particle hole symmetry at half filling.

The ordinary Hubbard model describing electrons may be extended with an extra term to incorporate the effect of spin-spin interactions. This is usually done by introducing the Heisenberg term H_{Heis} which is as follows,

$$H_{Heis} = J \sum_{\langle r,r' \rangle} S_r \cdot S_{r'} \tag{3.4}$$

where the symbol $\langle r, r' \rangle$ normally is taken as the nearest neighbour sites. One could also extend this to further interactions (2^{nd} nearest etc). When the Heisenberg term is included we usually talk about a $t - J$ model, or even a $t - J - U - \mu$-model if we also incorporate a chemical potential and a U-term.

In the spin 3/2 case we will take our last term in the Hamiltonian, H_J, a bit differently. Here the spin is given by $S_r = \sum_\sigma c^\dagger_{r,\sigma} S_{\sigma,\sigma'} c_{r',\sigma'}$ where $-\frac{3}{2} \leq \sigma \leq \frac{3}{2}$ and $S_{\sigma,\sigma'}$ are the generators of spin 3/2 rotations. Now we define the operator $P^\dagger_{2,m}(r)$ [11] which creates an $l = 2, l_z = m$ state with two spin 3/2 fermions.

$$P^\dagger_{2,m} = \sum_{\alpha,\beta} \left\langle \frac{3}{2}\frac{3}{2}\alpha\beta \right| \left. \frac{3}{2}\frac{3}{2}, 2, m \right\rangle c^\dagger_{r,\alpha} c^\dagger_{r,\beta} \tag{3.5}$$

We can now create an $SU(2)$ invariant operator as $P_r^2 \equiv \sum_m P^\dagger_{2,m}(r) P_{2,m}(r)$ and this is what we will use in our Hamiltonian so the last term has the form: $H_J = J \sum_r P_r^2$. In the spin 3/2 case we have a relationship as follows between P_r^2 and S_r^2: $P_r^2 = \frac{1}{3}S_r^2 - \frac{5}{2}n + \frac{5}{4}n^2$.

Lets look at the states and energies of our system.

# of particles	states
0	$\lvert\rangle$
1	$\lvert\Uparrow\rangle, \lvert\uparrow\rangle, \lvert\downarrow\rangle, \lvert\Downarrow\rangle$
$2_{quintet}$	$\lvert\Uparrow\uparrow\rangle, \lvert\Uparrow\downarrow\rangle, \lvert\Uparrow\Downarrow\rangle + \lvert\uparrow\downarrow\rangle, \lvert\uparrow\Downarrow\rangle, \lvert\downarrow\Downarrow\rangle$
$2_{singlet}$	$\lvert\Uparrow\Downarrow\rangle - \lvert\uparrow\downarrow\rangle$
3	$\lvert\Uparrow\uparrow\downarrow\rangle, \lvert\Uparrow\uparrow\Downarrow\rangle, \lvert\Uparrow\downarrow\Downarrow\rangle, \lvert\uparrow\downarrow\Downarrow\rangle$
4	$\lvert\Uparrow\uparrow\downarrow\Downarrow\rangle$

Table 3.1: Here ⇑/⇓ symbolizes spin $\pm^3\!/_2$, and ↑/↓ symbolizes spin $\pm^1\!/_2$.

What energy levels does the system have? Call the ground state for n particles $E_g(n)$, we then have $E_g(0) = 0, E_g(1) = -U, E_g(2_{sing}) = 0$. To find $E_g(3)$ we can use $P_r^2 = \frac{1}{3}S_r^2 - \frac{5}{2}n + \frac{5}{4}n^2$ which gives

$$E_g(3) = 3U + J\left(\frac{1}{3}\frac{3}{2}\frac{5}{2} - \frac{5}{2}\cdot 3 + \frac{5}{4}\cdot 9\right) = 3U + J\left(\frac{5}{4} - \frac{30}{4} + \frac{15}{4}\right) = 3U + 5J \tag{3.6}$$

In the same way we can get $E_g(4) = 8U + 10J$ and finally the quintet has ground state energy $E_g(2_{quin}) = 2J$. If we now realize these energies in a plot we have:

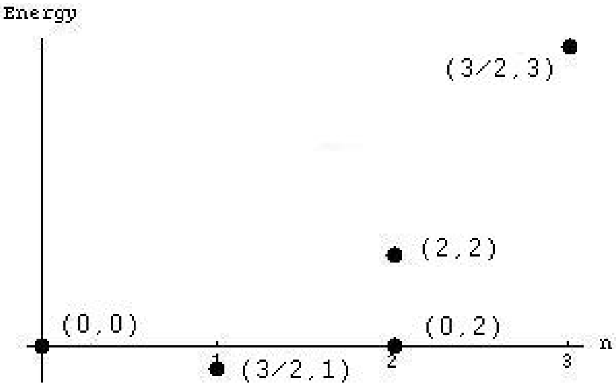


Figure 3.1: Energy of the different states indexed with (s,n), where s is spin an n is the number of particles.

3.1 Transformations

Lets focus on the case with low n and particle doping which is of the strength that a $-\mu N$-term in the Hamiltonian makes the energy levels of the (0,0) and (3/2,1) states have the same energy. In the notation (s,n) s is the spin and n is the number of particles. N is the total number of particles given by $N = \sum_{r,\sigma} n_{r,\sigma}$. In Fig. (3.2) this case would be represented by the dashed lines. and give us a energy hierarchy as follows

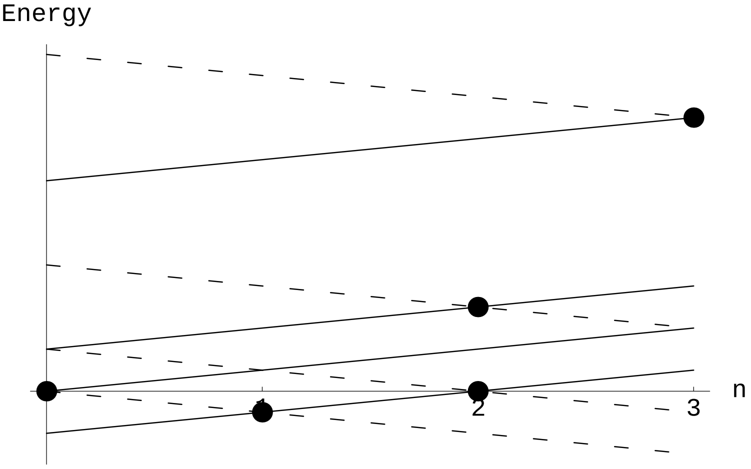


Figure 3.2: Here we have the energy levels plotted together with a schematic view of the energy levels if we apply a chemical potential such that the state (3/2,1) is degenerate with (0,0) (dashed lines). The case where (3/2,1) is degenerate with (0,2) can also be seen (solid lines).

Considering the half filled case with hole doping (solid lines) we get another energy spectrum, that now has the (0,2) and (3/2,1) states energetically degenerate (solid lines in Fig 3.2.)

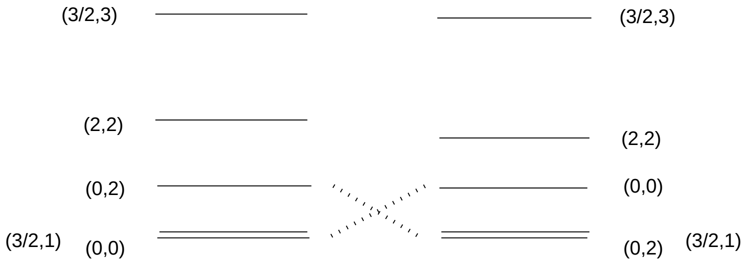


Figure 3.3: If we draw the energy hierarchy of the two cases mentioned in Fig. (3.2), we can identify what transformation that would take us between the two systems.

Mapping between these two cases can be done via the transformation matrix in Table (3.2).

$$\begin{pmatrix} 0 & 1 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & \\ \vdots & \vdots & & \ddots \end{pmatrix}$$

Table 3.2: Note that the basis in this matrix is the spin-number-states used before: (0,0), (0,2), (2,2), (3/2,1), (3/2,3), (0,4). The awkward positioning of (3/2,1) is used because that state doesn't get affected by the transformations used here, only the first three states will be of importance to us.

As can be predicted the condition $J > 2U \gg t$ gives us a configuration where the singlet 2-particle state is preferred around half filling. Let's now look at the case where we have a weak spin interaction meaning a small J. In the new energy plot (Fig. 3.4) we get the energy levels to come closer:

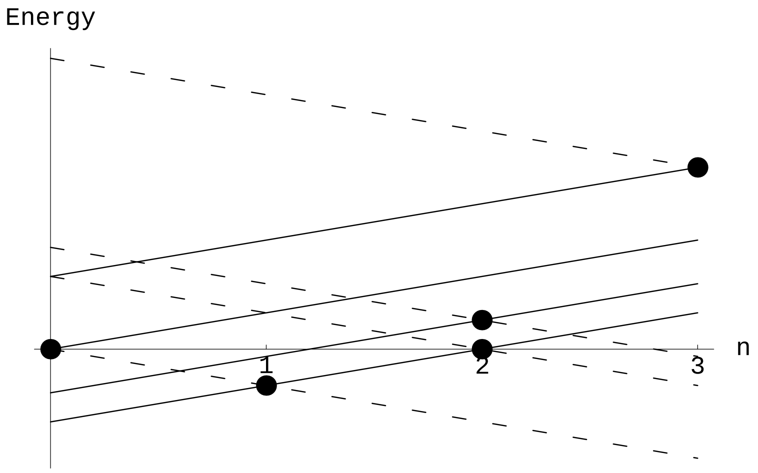


Figure 3.4: Considering the case of a weak spin interaction (small J), gives us this picture in the energies of different states.

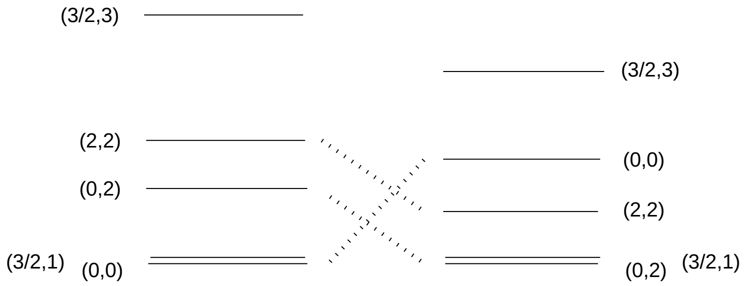


Figure 3.5: The energy hierarchy is now a bit more complicated which gives us a more complicated transformation between our two cases.

To find a transformation matrix as before we now must take the (2,2) state into consideration ie the quintet state. So therefore we do not have a single unique transformation and our rotational symmetry is broken.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & \\ \vdots & \vdots & \vdots & & \ddots \end{pmatrix}$$

Table 3.3: The transformation matrix corresponding to a weak spin interaction. As before the basis of this matrix is the spin-number-states: (0,0), (0,2), (2,2), (3/2,1), (3/2,3), (0,4).

It is true that the spin 3/2 model is the simplest model suitable for our desired mapping. But it is still not a very good model to investigate because of several reasons. The complexities introduced by the five-fold degeneracy of the preferred states, together with the fact that it is difficult to physically realize a spin 3/2 fermion system, cause us to address the conceptual issues in the 2-band Hubbard model.

4

2-Band Hubbard Model

The second model that we will examine closer is the 2-band Hubbard model, with two spin 1/2 fermions per band. For this model we have a different Hamiltonian which looks as follows

$$H = H_{hop} + H_{2-bHubb} \tag{4.1}$$

The hopping term H_{hop} will now account for lattice hopping as well as band hopping. The band hopping is an on-site term which looks as follows: $t_{\alpha\beta} \sum_{r,\sigma}(c^\dagger_{r,\sigma,\alpha} c_{r,\sigma,\beta} + h.c.)$ with α, β the two band indices. We also have an off-site term which will account for the hopping between lattice sites, without simultaneous band hopping.

$$H_{latt.hop} = \sum_{r,\sigma} t_\alpha c^\dagger_{r,\sigma,\alpha} c_{r,\sigma,\alpha} + t_\beta c^\dagger_{r,\sigma,\beta} c_{r,\sigma,\beta}. \tag{4.2}$$

When it comes to potential terms we now have a more complex situation with several different terms. The reason for this is that there is now a degenerate representation of spin among the states at half filling and symmetries does not constrain the model to as few terms as in the spin 3/2 case.

To find these terms we need to look at the different states at half filling. In the spin 3/2-model we had a quintet and a singlet state at half filling. In this case we will have

$$(\tfrac{1}{2}_\alpha \oplus \tfrac{1}{2}_\beta) \otimes (\tfrac{1}{2}_\alpha + \tfrac{1}{2}_\beta) = (1 \oplus 0)_{\alpha\alpha} + (1 \oplus 0)_{\alpha\beta} + (1 \oplus 0)_{\beta\beta}. \tag{4.3}$$

Since we are dealing with fermions the states $1_{\alpha\alpha}$ and $1_{\beta\beta}$ are not allowed. We then have 6 states.

state	degeneracy
$0_{\alpha\alpha}$	singlet
$1_{\alpha\beta}$	triplet
$0_{\alpha\beta}$	singlet
$0_{\beta\beta}$	singlet

These states need to be separated with potential terms in a complete Hamiltonian that describes the system accurately. Below one finds an example of such a local Hamiltonian.

$$H_{pot} = \sum_r V n_\alpha n_\beta + U_\alpha n_\alpha^2 + U_\beta n_\beta^2 + \Delta(n_\alpha - n_\beta) \tag{4.4}$$

Note that if we have a system with symmetries one might be able to simplify the terms or simply remove some of them without changing he physics as long as the absent term does not introduce any accidental symmetries. The Δ-term will be the cause of an asymmetry in occupation between the α-band and the β-band. There is also an on-site term in this model that depends on the spin, taken as $J \sum_r S_r^2$ where S_r is the total spin for the site r.

We want to represent the on-site problem with a matrix Hamiltonian. To do so efficiently we will first discuss a number of properties and simplifications of the model.

The first thing that we need is to enumerate the states in this model:

# of particles	states
0	$\lvert\rangle$
1	$\lvert\uparrow_\alpha\rangle, \lvert\downarrow_\alpha\rangle, \lvert\uparrow_\beta\rangle, \lvert\downarrow_\beta\rangle$
$2_{triplet}$	$\lvert\uparrow_\alpha\uparrow_\beta\rangle, \frac{1}{\sqrt{2}}\left(\lvert\uparrow_\alpha\downarrow_\beta\rangle + \lvert\downarrow_\alpha\uparrow_\beta\rangle\right), \lvert\downarrow_\alpha\downarrow_\beta\rangle$
$2_{singlets}$	$\lvert\uparrow_\alpha\downarrow_\alpha\rangle, \frac{1}{\sqrt{2}}\left(\lvert\uparrow_\alpha\downarrow_\beta\rangle - \lvert\downarrow_\alpha\uparrow_\beta\rangle\right), \lvert\uparrow_\beta\downarrow_\beta\rangle$
3	$\lvert\uparrow_\alpha\downarrow_\alpha\uparrow_\beta\rangle, \lvert\uparrow_\alpha\downarrow_\alpha\downarrow_\beta\rangle, \lvert\uparrow_\alpha\uparrow_\beta\downarrow_\beta\rangle, \lvert\uparrow_\alpha\uparrow_\beta\downarrow_\beta\rangle$
4	$\lvert\uparrow_\alpha\downarrow_\alpha\uparrow_\beta\downarrow_\beta\rangle$

Table 4.1: states in the 2-band Hubbard model.

Making assumptions that we anticipate do not change the physics, the Hamiltonian can be simplified with assumptions such as putting $U_\alpha = U_\beta$. We can now schematically draw the different energies in a energy diagram. With the $U_\alpha = U_\beta = U$ simplification we get the energies, $E_g(0) = 0$, $E_g(1) = \pm\Delta + \frac{3}{4}J + U$ for the particle in band α or β. For two particles we have $E_g(2_{same\,band}) = \pm 2\Delta + 4U$, or for particles in different bands $E_g(2_{sing}) = 2U + V$ and $E_g(2_{trip}) = E_g(2_{sing}) + 2J$. For three particles we have $E_g(3) = \pm\Delta + \frac{3}{4}J + 5U + 2V$, and finally $E_g(4) = 8U + 4V$. This can be displayed in the following energy plot.

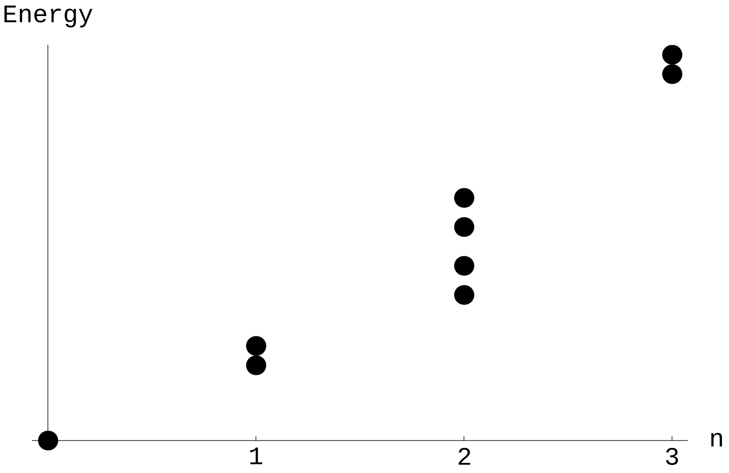


Figure 4.1: Schematic plot of the energies. For two particles the four different energies comes from the states: : $2_{sing}, 2_{\beta\beta}, 2_{\alpha\alpha}, 2_{trip}$. Note that the order of especially the two-particle states will vary depending on which parameters are used and their relative sizes.

A first glimpse at the energies tell us that this system might need a very complex Hamiltonian to be fully describing all the different states and degrees of freedom. The values of the multiple parameters can affect the model, so there are a few questions that need to be addressed. One is the relative sizes of the parameters as well as their signs, another is what typical magnitude the spin term has, for the spin 3/2-model we looked at both low and high J. That same comparison might need more consideration in this model due to the fact that there are now more energy levels in the system.

The Δ-term separates the two one-particle states that differ only by their band index - α or β. The sign and relative size of this term will in a very fundamental way decide the way the system behaves. So how should the parameters compare?

Some of the needed answers come from what type of system and energy hierarchy we are interested in. That might also lead us to symmetries and simplifications that can be implemented in the Hamiltonian. Lastly the Hamiltonian that we mentioned for the model, is it a complete description of the model or do we need even more parameters? Below we will try and look into some of these questions.

4.1 Singlet Ground State

The goal for the calculations that I will do on the model is to describe a mapping of the half filled singlet state (with one particle in each band) onto the ground state. For this to be possible and not overly complicated we want to remove the $|\alpha\alpha\rangle$ and $|\beta\beta\rangle$ two-particle states. For this to map the ground state at n=2 to the ground state at zero filling, we need the triplet state to have an energy that is higher than the one for the singlet. Having a system with all these properties would allow us to make a transformation as the ones seen in fig(3.2) and (3.4).

The first condition is satisfied by making the parameter U large; this effectively pushes up the two doubly occupied states. A note here that might need to be addressed is that choosing U large also affects the one-particle states (containing a term U) and the other two particle states ($2U$), although not as much as the doubly occupied states ($4U$). We could alter the U-term in a way to only affect the latter states by setting the terms as $Un_\alpha(n_\alpha - 1)$.

The second condition above is satisfied by having $J > 0$. The singlet is then lower in energy compare to the triplet. These conditions will lead us to a new energy diagram containing only the relevant states at low energies. Unfortunately the conditions mentioned above put restraints on the system that are not necessarily physically motivated but are useful to simplify the problem considerably.

We will use Mathematica for its symbolic handling properties to do the calculations involving the creation and destruction operators. So we must then assure ourselves that all of the terms in the Hamiltonian can be expressed in these operators.

If we look at the earlier mentioned Hamiltonian, then our on-site Hamiltonian was:

$$H_{on-site} = t_{\alpha\beta}\sum_{\sigma}(c^{\dagger}_{r,\sigma,\alpha}c_{r,\sigma,\beta}+h.c.)+V\,n_\alpha n_\beta+U\,(n_\alpha^2+n_\beta^2)+\Delta\,(n_\alpha-n_\beta)+J\,S_r^2 \tag{4.5}$$

Since the number operator is just $n_\alpha = n_{\alpha,\uparrow} + n_{\alpha,\downarrow} = \sum_\sigma c^{\dagger}_{\alpha,\sigma}\,c_{\alpha,\sigma}$. we have easy expressions for all terms except the J-term. S_r is the total spin for a site, ie for both levels added on a given site. We want the matrix representation for the spin term in an operator basis, consisting of creation and destruction operators. Using the relation that $S_r = S_\alpha + S_\beta$ means that the square is expressed as

$$S_r^2 = (S_\alpha + S_\beta)^2 = S_\alpha^2 + S_\beta^2 + 2\,S_\alpha \cdot S_\beta. \tag{4.6}$$

We can find the terms of eq. (4.6) if we for example break S into components according to

$$S_1 \cdot S_2 = S_{z_1}\,S_{z_2} + \frac{1}{2}\left(S_1^+ S_2^- + S_1^- S_2^+\right) \tag{4.7}$$

where $S_z = \frac{1}{2}(n_\uparrow - n_\downarrow)$, $S^+ = c_\uparrow^\dagger c_\downarrow$ and $S^- = c_\downarrow^\dagger c_\uparrow$. Inserting our terms leads to

$$\begin{aligned} S_\alpha^2 = S_\alpha \cdot S_\alpha = \left[\tfrac{1}{4}(n_\uparrow - n_\downarrow)(n_\uparrow - n_\downarrow) + \tfrac{1}{2}\left(c_\uparrow^\dagger c_\downarrow c_\downarrow^\dagger c_\uparrow \; + \; c_\downarrow^\dagger c_\uparrow c_\uparrow^\dagger c_\downarrow\right)\right]_\alpha = \\ = \left[\tfrac{1}{4}\left(n_\uparrow^2 - 2n_\uparrow n_\downarrow + n_\downarrow^2\right) + \tfrac{1}{2}\left(n_\uparrow - n_\uparrow n_\downarrow + n_\downarrow - n_\uparrow n_\downarrow\right)\right]_\alpha = \\ \tfrac{3}{4}\left(n_{\alpha,\uparrow} + n_{\alpha,\downarrow}\right) - \tfrac{3}{2} n_{\alpha,\uparrow} n_{\alpha,\downarrow} \end{aligned} \tag{4.8}$$

and similar for S_β^2, finally the last term in Eq. (4.6) becomes

$$\begin{aligned} 2\, S_\alpha \cdot S_\beta = \\ \tfrac{2}{4}(n_{\alpha,\uparrow} - n_{\alpha,\downarrow})\,(n_{\beta,\uparrow} - n_{\beta,\downarrow}) + \tfrac{2}{2}\left(c_{\alpha,\uparrow}^\dagger c_{\alpha,\downarrow} c_{\beta,\downarrow}^\dagger c_{\beta,\uparrow} \; + \; c_{\alpha,\downarrow}^\dagger c_{\alpha,\uparrow} c_{\beta,\uparrow}^\dagger c_{\beta,\downarrow}\right) = \\ = \tfrac{1}{2}\left(n_{\alpha,\uparrow} n_{\beta,\uparrow} \; + n_{\alpha,\uparrow} n_{\beta,\downarrow} + n_{\alpha,\downarrow} n_{\beta,\uparrow} + n_{\alpha,\downarrow} n_{\beta,\downarrow}\right) - \\ \left(c_{\alpha,\uparrow}^\dagger c_{\beta,\downarrow}^\dagger c_{\alpha,\downarrow} c_{\beta,\uparrow} \; + \; c_{\alpha,\downarrow}^\dagger c_{\beta,\uparrow}^\dagger c_{\alpha,\uparrow} c_{\beta,\downarrow}\right) \end{aligned} \tag{4.9}$$

Using these terms we can now construct the J-term in a matrix Hamiltonian directly.

What symmetries do we expect a Hamiltonian to have, and what operators should the Hamiltonian commute with? The first property of it should be that it should preserve particle number. This mean that the desired Hamiltonian H should commute with $\hat{N}$, where $\hat{N}$ here is the total number operator

$$\hat{N} = n_{\alpha,\uparrow} + n_{\alpha,\downarrow} + n_{\beta,\uparrow} + n_{\beta,\downarrow}. \tag{4.10}$$

The second property is that it should be invariant under spin rotations, ie H should commute with the spin operators: S_x, S_y, S_z. The different spin operators can be derived from the formula $S^a = c_\sigma^\dagger \tau_{\sigma,\sigma'}^a c_{\sigma'}$, where τ^a are the different spin-1/2 Pauli matrices. If we also use the fact that $S_a = S_{a,\alpha} + S_{a,\beta}$, this results in the following expressions

$$\begin{aligned} S_x &= \frac{1}{2}\left(c_{\alpha\uparrow}^\dagger c_{\alpha\downarrow} + c_{\beta\uparrow}^\dagger c_{\beta\downarrow} + c_{\alpha\downarrow}^\dagger c_{\alpha\uparrow} + c_{\beta\downarrow}^\dagger c_{\beta\uparrow}\right) \\ S_y &= \frac{i}{2}\left(c_{\alpha\uparrow}^\dagger c_{\alpha\downarrow} + c_{\beta\uparrow}^\dagger c_{\beta\downarrow} - c_{\alpha\downarrow}^\dagger c_{\alpha\uparrow} - c_{\beta\downarrow}^\dagger c_{\beta\uparrow}\right) \\ S_z &= \frac{1}{2}\left(n_{\alpha,\uparrow} - n_{\alpha,\downarrow} + n_{\beta,\uparrow} - n_{\beta,\downarrow}\right). \end{aligned} \tag{4.11}$$

By computing the commutator between a Hamiltonian containing all possible operators and the number- and spin-operators we will end up with a list of the total number of commuting operators that exist in this model. The last thing that we also need to accommodate for is that $H = H^\dagger$. Where the dagger denotes hermitian conjugate. As we saw in section 2.1 all of this can be easily done in the matrix representation of the Fermi operators.

To perform this analysis most efficiently we convert the number and spin operators to matrices and then calculate what conditions we get out from the commutation relations. Finally we demand $H = H^\dagger$. To implement this procedure numerically, we start with creating a matrix H that looks as follows

$$H = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,16} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,16} \\ \vdots & \vdots & \ddots & \\ h_{16,1} & h_{16,2} & & h_{16,16} \end{pmatrix} \tag{4.12}$$

The total equation system that needs to be solved is then

$$\begin{cases} [H,N] & = & H \cdot N - N \cdot H = & 0 \\ [H,J_x] & = & H \cdot J_x - J_x \cdot H = & 0 \\ [H,J_y] & = & H \cdot J_y - J_y \cdot H = & 0 \\ [H,J_z] & = & H \cdot J_z - J_z \cdot H = & 0 \\ H - H^\dagger & = & 0 & \end{cases} \tag{4.13}$$

Solving this equation system gives us the information that there are 15 independent operators that commute with H. This means that to fully describe the model we need 15 parameters in the Hamiltonian. More about this result and the parameters will be discussed in the subsequent sections.

5

Implementing the Calculation

In order to perform the complex algebraic manipulation necessary, we will be using symbolic handling software, in form of Mathematica, to perform our calculations. The main thing we gain from using Mathematica is that we can keep our parameters with undefined values and therefore look directly at what happens to our parameters and operators.

5.1 The Structure of Mathematica

Normally the structure in computational physics is that you transfer your problem to a purely numerical algorithm and then execute the program. This method is easily understood and used in Matlab for example. Using Mathematica one can incorporate both symbolic and numerical calculations. Mathematica also have a slightly different approach when dealing with computational problems, which is made possible by the fact that the algebra used in Mathematica can be modified by the user.

The perfect example of when one would like to modify the algebra used is in quantum mechanics. Normal multiplication of numbers and symbols is taken to be commutative so that $a \cdot b = b \cdot a$, this corresponds to $[a, b] = 0$, if formulated using Lie brackets. In quantum mechanics one often comes across situations where one uses operators which does not obey to a commutative algebra. The easiest example being fermionic or bosonic creation operators.

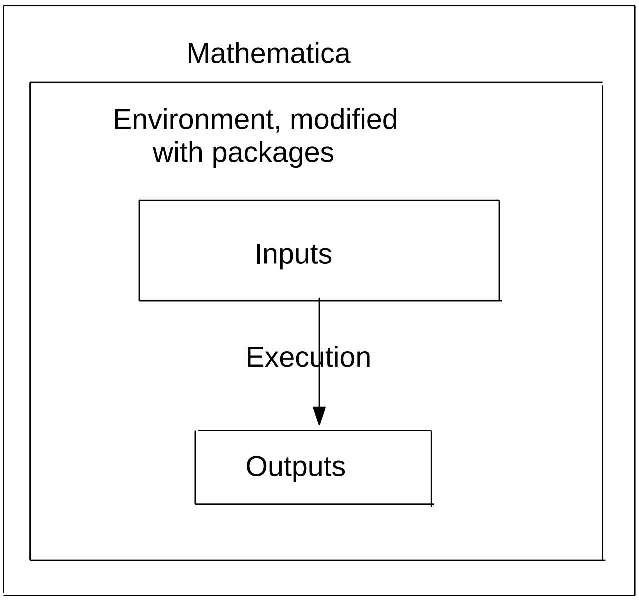
With this insight we realize that we need to make sure that Mathematica does not make the canonical sorting of variables to substitute ba with ab, unless it is allowed. The place for defining algebras as well as important operations is inside a package-file (.m). This package is then loaded into Mathematica to ensure that the right calculation will be performed. The package can therefore be seen as the environment in which we then write our code to perform the desired calculations.

Packages need only to be loaded once since Mathematica integrates the changes into its existing environment. This is very convenient since it means that we can still rely on all the functions and commands that Mathematica has predefined. This also means that the definitions and changes we introduce should be chosen in such a way that they don't conflict with the ones Mathematica use.

An illustration of this is if we need a multiplication that isn't commutative. It is then convenient to choose the symbol of that type of multiplication differently from the one Mathematica uses for normal multiplication e.g using ** for non-commuting multiplication to not interfere with *. Doing it this way then enables us to use both types of multiplication without worrying about Mathematica doing the wrong interpretation.

The same reasoning goes for operations or functions defined by the user. Say we want to define an integration-command ourselves. Instead of using Mathematica's name 'Integrate' for it we are usually better of calling it something else such as 'MyIntegrate' we are then free to use both versions and can use Mathematica's version to verify the answer we get from 'MyIntegrate'.

Now that we have a suitable environment for our calculations we write our code/commands directly in Mathematica in the form of notebooks (.nb). The answers that we receive will be in the form of outputs in our notebook. The output can be of many forms such as expressions, vectors and matrices and different types of graphs or other graphical outputs.



5.1.1 2-Band Hubbard Model

So what is needed to perform the calculations needed in the 2-band Hubbard Model? First we need some extra algebra. We will be working with fermionic creation and destruction operators. Thus we need the multiplication of two operators to be anti-commuting. We also need to define a few new operations among them Hermitian Conjucation and Normal Ordering. Conjugation is already defined by Mathematica, but that operation doesn't work if we try it on creation or destruction operators. Therefore we need some extended hermitian conjugation.

The notebook we create need to include routines for going from fermionic representation to matrix representation as mentioned in the section 2.1. Also needed is an analysis of the number of parameters we have in our model. Using that knowledge we can create a Hamiltonian containing operators that account for all of the parameters the system can have.

Lastly we need to perform the transformation that maps the two-particle singlet onto the empty (vacuum) state, and observe what happens to the parameters of the model.

5.2 Package

The package we used for the fermionic calculations contain several things to make the calculations possible. Among those are hermitian conjucate and normal ordering for operator expressions.

To make sure Mathematica knows when we want to use fermionic algebra, we designate the letters c, d respectively for creation and destruction operators. We also need arguments for the operators, in the 2-band Hubbard model we need one argument for the spin of the particle and one argument for the band. Thus our operators will be of the form $\{c_{i,\sigma}\,,\, d_{i,\sigma} \quad : \quad i = \alpha, \beta \;\; ; \;\; \sigma = \uparrow, \downarrow\}$ where α, β denotes the band or level. Mathematica will see the operator as `c[i,σ],d[i,σ]`.

5.2.1 Hermitian Conjugation

The difficulties of hermitian conjugation comes whenever we have an expression containing `c,d`. We will name the function for hermitian conjugation as `HC`. We still want `HC` to work as normal conjugation whenever it encounters an expression that doesn't contain `c` or `d`.

To make sure that it does so, we need to look into how Mathematica treats numbers and symbols. The software has a few different ways to store numbers depending on if they are integers, rationals, complex or real numbers. When Mathematica encounters a number it will store it with the appropriate head to classify the entity. This also makes it easy for us to setup HC. We know for example that integers are not affected when subject to (hermitian) conjugation.

Other known aspects of HC is that if applied twice we end up with our original expression. HC is also distributive that is:

```
HC[e (a+b)]= HC[e] HC[a+b]=HC[e] (HC[a]+HC[b])
```

There are a few other possible heads that we need to get HC working with. The most notable of those is how Mathematica deals with negative numbers. when Mathematica sees something like $-a$ it stores it as $(-1) \cdot a$ or in Mathematica's notation $-a$=Times[-1,a]. Another head to consider is powers. Mathematica uses that on expressions such as $e^x, \sqrt{2} = 2^{1/2}$.

Below is how these thing were accomplished in our package:

```
SetAttributes[HC, Listable];
HC[x_Real] := x;
HC[I] := -I;
HC[x_Integer] := x;
HC[x_Complex] := Conjugate[x];
HC[x_Rational] := x /. Complex[z_, y_] :> Complex[z, -y];
HC[Power[E_, x_]] := Power[E, x] /. Complex[z_, y_] :>
          :> Complex[z, -y];
HC[HC[x_]] := x;
HC[-x_] := -HC[x];  (* takes care of head - Times[-1, ..] *)
HC[x_ + y_] := HC[x] + HC[y]; (* Listable *)
HC[x_ y_] := HC[x] HC[y];
```

The last thing we need to do for HC is to make it work properly for creation and destruction operators. It is easy to define what happens to a single operator with:

```
HC[c[s__]] := d[s];
HC[d[s__]] := c[s];
```

But the operators may also come as factors in longer expressions. When performed the hermitian conjugation substitutes the c's and d's. But it also reverses the order of the operators when they come as factors in an expression that is composed with non commuting multiplication. Mathematica's normal multiplication has the head Times[...]. To differentiate the Non Commuting Multiplication we define our own multiplication-head as NCM[...]. This means that whenever HC sees an expression containing NCM, it knows that it also needs to reverse the order of the factors. In our package this is accomplished by

```
HC[NCM[x___]] := Module[{z}, z = Reverse[{x}]; z =
    z //. {c[s__] -> dD[s], d[s__] -> cC[s]}; z = z //.
                {cC[s__] -> c[s], dD[s__] -> d[s]};
    z = NCM[z]; z=z //. NCM[{y__}] -> NCM[y]; Return[z]];
```

5.2.2 Normal Ordering

The routine of normal ordering has two tasks in the calculation we perform. The first one is to make sure that operator-multiplication has the head NCM, and that only relevant things are left inside NCM. This means that scalars should not be inside the expression since they obey under normal multiplication. It is then favourable to exclude them from NCM and let Mathematica deal with that type of multiplications, leaving only c's and d's within the head NCM. The second and main task for the routine is to perform quantum mechanical normal ordering. The normal ordering moves all destruction operators to the left and all creation operators to the right using the Fermi anti-commutation rules, Eq. (1.3).

The relative order of the creation (destruction) operators is just canonical with respect to what arguments they contain. In the case of multiple argument like level and spin, one need to chose what argument takes precedence. In my package the level (first argument) is the main thing the operators order themselves after and secondly they use the spin for ordering.

All of this is accomplished with the following definition of normal-ordering

```
NormalOrder[xx_] := Module[{ss},
    ss = xx //. {a__ ** b__ -> NCM[a,b]};
    ss = ss //. NCMArrange;
    ss = ss /. NCM[x__] :> Normalord[NCM[x]];
    ss = ss //. NCMArrange;
    Return[ss]];
```

NormalOrder consists of several parts. First we take the raw expression and adjust it with NCMArrange. We then end up with expressions that only contain the proper operators inside NCM. This procedure is simple but has a few different cases that need to be accounted for

```
NCMArrange=Dispatch[{
    NCM[a___, -b_, e___] -> -NCM[a, b, e],
    NCM[a___, b_.1NCM[x__], e___] -> b NCM[a, x, e],
    NCM[a___, b__ ** e__, f___] -> NCM[a, b, e, f],
    NCM[a___, b_?ScalarQ, e___] -> b NCM[a,e],
    NCM[a___, b_ c[x__], e___] -> b NCM[a, c[x], e],
    NCM[a___, b_ d[x__], e___] -> b NCM[a, d[x], e],
    a_ NCM[e___] + b_ NCM[e___] -> (a + b) NCM[e],
    NCM[] NCM[x___] -> NCM[x],
    NCM[]^x_ -> NCM[],
    NCM[a___, g_.1(b__ + e__), f___] :>
         :> Distribute[g NCM[a, b + e, f]] /; FreeQ[g, NCM]}];
```

Now that we have simplified the NCM-expressions, the normal ordering can be performed easily without dealing with scalars and normal multiplication. The normal ordering part is

```
Normalord[xxx_] := Module[{zzz}, zzz = xxx //. Moving;
     Return[zzz]]
Normalord[NCM[]] := 1;
```

where the routine calls upon the predefined moving rules that makes sure the operators order themselves in manner that ensures that the expression is normal ordered. The moving rules is just a simple list of case for case movements.

```
Moving = Dispatch[{NCM[a___, c[k__], b___, c[k__], e___] :>
               :> 0 /; FreeQ[{b}, d[k]],
    NCM[a___, d[k__], b___, d[k__], e___] :>
               :> 0 /; FreeQ[{b}, c[k]],
    NCM[a___, d[i__], c[i__], b___] ->
               ->NCM[a, b] - NCM[a, c[i], d[i], b],
    NCM[a___, d[k_,i__], c[l_,j__], b___] :>
               :> -NCM[a, c[l,j], d[k,i], b] /;
               /; !MatchQ[i,j]|| ! MatchQ[l, k],
    NCM[a___, c[k_,i__], c[l_,j__], b___] :>
               :> -NCM[a, c[l,j], c[k,i], b] /; k > l,
    NCM[a___, c[k_,i__], c[k_,j__], b___] :>
               :> -NCM[a, c[k,j], c[k,i], b] /; i > j,
    NCM[a___, d[k_,i__], d[l_,j__], b___] :>
               :> -NCM[a, d[l,j], d[k,i], b] /; k > l,
    NCM[a___, d[k_,i__], d[k_,j__], b___] :>
               :> -NCM[a, d[k,j], d[k,i], b] /; i > j}];
```

Note that the efficiency can be improved by a routine that identifies expressions where it doesn't exist creation and destruction operators of the same kind (same level and spin). The normal ordering is then just canonical ordering, with a negative sign if the total number of translocations is odd. With the right environment setup we can now look at how we realize the calculations needed in Mathematica.

5.3 Execution

Although it is in principle possible to use the fermion algebra symbolically, complex expressions become very cumbersome so it is useful to resort to matrix operations. Once that is done we can calculate the number of parameters for the system and also perform the desired transformation and see what that means when we go back to operator formalism.

5.3.1 Fermi Polynomials and Matrix Formulations

As before, we to need build a matrix $M_{i,j}$ that obeys the relation $M_{i,j}\Psi_k = \Psi_i\delta_{j,k}$. The main difference compared to earlier is that the matrix will now be larger, with very long polynomials of Fermi operators. There are now four creation operators and thus 16 different states that the system can have. One empty state, four one-particle states, six two-particle states, four three-particle states and one completely filled four-particle state.

The two-particle states can be chosen in different ways depending on what basis we want. We will chose a spin state basis as opposed to an operator basis which is just the simplest forms of the two-particle states. The reason for this is that operators like the spin operator then have simple matrix representations in our basis. The difference between the base-sets can be seen below.

Operator basis	Spin basis	S_z	S^2
$\lvert\uparrow_\alpha\downarrow_\alpha\rangle$	$\lvert\uparrow_\alpha\downarrow_\alpha\rangle$	0	0
$\lvert\uparrow_\alpha\uparrow_\beta\rangle$	$\frac{1}{\sqrt{2}}\left(\lvert\uparrow_\alpha\downarrow_\beta\rangle - \lvert\downarrow_\alpha\uparrow_\beta\rangle\right)$	0	0
$\lvert\uparrow_\alpha\downarrow_\beta\rangle$	$\lvert\uparrow_\beta\downarrow_\beta\rangle$	0	0
$\lvert\downarrow_\alpha\uparrow_\beta\rangle$	$\lvert\uparrow_\alpha\uparrow_\beta\rangle$	1	1
$\lvert\downarrow_\alpha\downarrow_\beta\rangle$	$\frac{1}{\sqrt{2}}\left(\lvert\uparrow_\alpha\downarrow_\beta\rangle + \lvert\downarrow_\alpha\uparrow_\beta\rangle\right)$	0	1
$\lvert\uparrow_\beta\downarrow_\beta\rangle$	$\lvert\downarrow_\alpha\downarrow_\beta\rangle$	-1	1

Table 5.1: Spin basis versus the operator basis for the two-particle states.

Creating the M-matrix in Mathematica is done in steps. First the first row is created. Note that we move our states around in order to get a simple matrix configuration. The first column is then just the first row hermitially conjugated. Lastly the interior of the matrix is calculated through $M_{i,j} = M_{i,1}M_{1,j}$.

```
<< Fermlevel.m
A = Table[0, {i, 16}, {j, 16}];
Dtotal = d[1, 1] ** d[1, 2] ** d[2, 1] ** d[2, 2];
Ctotal = C[4] ** C[3] ** C[2] ** C[1];
A[[1, 1]] = Dtotal ** Ctotal;
Table[A[[1, i + 1]] = Dtotal ** Ctotal /.
        /.  C[i] -> 1, {i, 1, 4}];
mov = A[[1, 3]]; A[[1, 3]] = A[[1, 4]]; A[[1, 4]] = mov;
Table[A[[1, 4 + j]] = Dtotal ** Ctotal /.
        /.  {C[1] -> 1, C[j] -> 1}, {j, 2, 4}];
Table[A[[1, 4 + i + j]] = Dtotal ** Ctotal /.
        /.  {C[i] -> 1, C[j] -> 1}, {i, 2, 3}, {j, i + 1, 4}];
A18 = A[[1, 8]] + A[[1, 9]]; A19 = A[[1, 8]] - A[[1, 9]];
A[[1, 8]] = A18/Sqrt[2]; A[[1, 9]] = A19/Sqrt[2];[1, 4];



(*Basis notes :  spin basis is used, to get
triplet/singlets grouped together use the moving below *)

temp = Table[A[[1, i]], {i, 6, 11}];
A[[1, 7]] = temp[[4]];
A[[1, 8]] = temp[[6]];
A[[1, 9]] = temp[[2]];
A[[1, 10]] = temp[[3]];
A[[1, 11]] = temp[[5]];
Table[A[[1, i + 11]] = Dtotal ** C[5 - i], {i, 4}];
mov = A[[1, 13]]; A[[1, 13]] = A[[1, 14]]; A[[1, 14]] = mov;
A[[1, 16]] = Dtotal;
A = A /.  {C[1] -> c[1, 1], C[2] -> c[1, 2],
        C[3] -> c[2, 1], C[4] -> c[2, 2]};
A = NormalOrder[A];
Table[A[[i, 1]] = HC[A[[1, i]]], {i, 2, 16}];
Table[A[[i, j]] = A[[i, 1]] ** A[[1, j]],
        {i, 2, 16}, {j, 2, 16}];
A = NormalOrder[A];
```

As can be seen in the beginning of the program we first load our previously constructed package that contains the fermion algebra. The symbol that we have defined for our non commuting multiplication is `**`. In the code one can also note that we move the states into an order that is suitable for the coming calculations. This means for example that the 2-particle states are ordered in the manner that they are shown in Table

(5.1).

The Hilbert space in this model is 256-dimensional and consists of the tensor product between the creation and destruction operators. To know what polynomials different matrices represents we need a vector containing all existing fermionic terms. The part below allows us to create that in a combinatoric manner.

```
Combine[N_, n_] := Module[{i, Elem, Cond, result},
    Elem = Table[i[j], {j, n}]; i[0] = 0; Cond = {};
    For[k = 1, k <= n, k++, Cond = Append[Cond,
        {i[k], i[k - 1] + 1, N}]];
    Tot[a_, {b__}] := Table[a, b]; result = Tot[Elem, Cond];
    result = Flatten[result, n - 1]; Return[result]];
Op[1] = c[1, 1]; Op[2] = c[1, 2]; Op[3] = c[2, 1];
Op[4] = c[2, 2]; Op[5] = d[1, 1]; Op[6] = d[1, 2];
Op[7] = d[2, 1]; Op[8] = d[2, 2];
OpConvert[x_] := Module[{y}, y = Outer[Op, x];
    y = Map[NCM, y]; y = y /.  NCM[{z__}] -> NCM[z];
    Return[y]];
V = {NCM[]};
For[i = 1, i <= 8, i++, V = Join[V,
    OpConvert[Combine[8, i]]]];
```

Now we can create the routines M2F and F2M which transforms a matrix to Fermi polynomial and vice versa respectively.

```
Aflat = Flatten[A];
U = Table[D[Aflat, V[[i]]], {i, 1, Length[V]}];
U = SparseArray[U]
Uinv = Inverse[U];
Uinv = SparseArray[Uinv];
M2F[m_] := Module[{vect, x}, vect = V.U;
    x = vect.Flatten[m]; x = NormalOrder[x]; Return[x]]
F2M[f_] := Module[{matrix, x},
    matrix = Table[D[f, V[[i]]], {i, 1, 256}];
    matrix = Uinv.matrix; x = Partition[matrix, 16];
    Return[x]]
```

If one wishes to test the the algorithms and code written so far that can be done by for example double application of the transformation routines such as `M2F[F2M[p]]==p`. Another test is transforming a creation and it's corresponding destruction operator (`c1,d1`) into matrices and the checking the anti-commutation relation that they should obey. Namely `c1.d1+d1.c1` which should equal the identity matrix.

5.3.2 Number of Parameters

The next thing that we want to find out is the number of parameters needed to describe the system. The way to do this was mentioned in section (4.1). Using commutation equations like $[H, J_x] = 0$ we get a system of equations in matrix form that gives us the number of parameters in the system. It also tells us what operators that needs to be included in the terms in a Hamiltonian. In Mathematica this was achieved in the following way

```
(* Commutation calculations *)
H = Array[h, 16, 16];
Num = NCM[c[1, 1], d[1, 1]] + NCM[c[1, 2], d[1, 2]] +
    + NCM[c[2, 1], d[2, 1]] + NCM[c[2, 2], d[2, 2]];
Nm = F2M[Num];
Jx = (NCM[c[1, 1], d[1, 2]] + NCM[c[2, 1], d[2, 2]]) +
    + (NCM[c[1, 2], d[1, 1]] + NCM[c[2, 2], d[2, 1]]);
Jy = I(NCM[c[1, 1], d[1, 2]] + NCM[c[2, 1], d[2, 2]]) -
    - I(NCM[c[1, 2], d[1, 1]] + NCM[c[2, 2], d[2, 1]]);
Jz = 1/2((NCM[c[1, 1], d[1, 1]] + NCM[c[2, 1], d[2, 1]]) -
    - (NCM[c[1, 2], d[1, 2]] + NCM[c[2, 2], d[2, 2]]));
Jxm = F2M[Jx];
Jym = F2M[Jy];
Jzm = F2M[Jz];
Nm // MatrixForm;
solv = Solve[{H.Nm - Nm.H == 0, H.Jxm - Jxm.H == 0,
    H.Jym - Jym.H == 0, H.Jzm - Jzm.H == 0,
    H == Transpose[H]}];
256 - Length[solv[[1]]];
H1 = H //.  solv[[1]];
```

The output can be seen in matrix form as a representation of which operators that need to be included in the Hamiltonian. The answer will be on the form shown in Table (5.2).

We find that 15 parameters are needed to fully describe the system.

$$\begin{pmatrix}
h_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & h_2 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & h_4 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & h_2 & h_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & h_4 & h_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & h_5 & h_8 & h_9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & h_8 & h_6 & h_{10} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & h_9 & h_{10} & h_7 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{11} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{11}
\end{pmatrix}$$

Table 5.2: this is the solution for 0,1 and 2 particles, the 3 and 4 particle bits mirror that of the 1 and 0 particle bit (first five columns/rows). Leading to a total of 15 parameters

5.3.3 Parameter Basis

There is a very large number of different ways that one can choose the 15 parameters and still cover the operators needed. So before we start listing suitable terms for the Hamiltonian it is a good idea to think about how we want the terms to look like and what they will be used for.

If we desire to later perform a mean field analysis of the parameters then it is favourable if the parameters only have operators that are homogeneous functions of the creation and annihilation operators. This is simply illustrated with our spin term that is $S^2 = S_\alpha^2 + S_\beta^2 + 2S_\alpha \cdot S_\beta$, meaning that the spin term will contain terms with both two and four creation and destruction operators. To fix this one can choose to take the spin term as only $S_\alpha \cdot S_\beta$.

Another case that illustrates the idea is a possible term as n_α^2 when normal ordering that expression it results terms with 2 operators as well as terms with 4 operators

$$n_\alpha^2 = (n_{\alpha,\uparrow} + n_{\alpha,\downarrow})^2 = 2\left(n_{\alpha,\uparrow}n_{\alpha,\downarrow} + n_{\alpha,\downarrow}n_{\alpha,\uparrow}\right) + n_{\alpha,\uparrow} + n_{\alpha,\downarrow}. \tag{5.1}$$

To eliminate the shorter terms one can choose to take the n_α^2-term as $n_\alpha(n_\alpha - 1)$ instead. Below is the terms used and coded in Mathematica.

```
(* Clean Basis (15) *)
One = NCM[];
Na = NCM[c[1, 1], d[1, 1]] + NCM[c[1, 2], d[1, 2]];
Nb = NCM[c[2, 1], d[2, 1]] + NCM[c[2, 2], d[2, 2]];
Mu = Na + Nb;
Delta = Na - Nb;
Num = (Na + Nb);
Na2right = Na ** (Na - 1);
Nb2right = Nb ** (Nb - 1);
NaNbright = NormalOrder[(Na + Nb) ** (Na + Nb - 1)];
Na2Nbright = Na ** Na ** Nb - Na ** Nb;
NaNb2right = Na ** Nb ** Nb - Na ** Nb;
Num4 = Num ** (Num - 1) ** (Num - 2) ** (Num - 3);



(* Spin *)
Hspin = 1/4(NCM[c[1, 1], d[1, 1]] ** NCM[c[2, 1], d[2, 1]] -
    - NCM[c[1, 1], d[1, 1]] ** NCM[c[2, 2], d[2, 2]] -
    - NCM[c[1, 2], d[1, 2]] ** NCM[c[2, 1], d[2, 1]] +
    + NCM[c[1, 2], d[1, 2]] ** NCM[c[2, 2], d[2, 2]]) -
    - 1/2(NCM[c[1, 1], c[2, 2], d[1, 2], d[2, 1]] +
    + NCM[c[1, 2], c[2, 1], d[1, 1], d[2, 2]]);



(* hopterm *)
Hop = NCM[c[1, 1], d[2, 1]] + NCM[c[2, 1], d[1, 1]] +
    + NCM[c[1, 2], d[2, 2]] + NCM[c[2, 2], d[1, 2]];
Hopa = NCM[c[1, 1], d[2, 1]] + NCM[c[1, 2], d[2, 2]];
Hopb = NCM[c[2, 1], d[1, 1]] + NCM[c[2, 2], d[1, 2]];
HopbNa = Na ** Hopb;
HopNa = NormalOrder[HopbNa];
HopNaright = HopNa + NormalOrder[HC[HopNa]];
HopaNb = Nb ** Hopa;
HopNb = NormalOrder[HopaNb];
HopNbright = HopNb + HC[HopNb];
Antihop = NCM[c[1, 1], c[1, 2], c[2, 1],
    d[1, 1], d[2, 1], d[2, 2]] + NCM[c[1, 1], c[1, 2],
    c[2, 2], d[1, 2], d[2, 1], d[2, 2]];
HopN2 = Antihop + HC[Antihop];
Hop2 = NCM[c[1, 1], c[1, 2], d[2, 1], d[2, 2]] +
    + NCM[c[2, 1], c[2, 2], d[1, 1], d[1, 2]];
```

Now we want to control that the 15 terms, that we have chosen to give us 15 parameters in the Hamiltonian, are indeed linearly independent so that they span the operators that we need. This can be achieved by constructing a matrix (256 × 15) that contains the matrices of the parameters as vectors in the rows. The rank of this created matrix should then be 15 for them to be linearly independent.

```
mat = Table[0, {i, 15}];
Ranker = {};
mat[[1]] = F2M[One];
mat[[2]] = F2M[NormalOrder[Mu]];
mat[[3]] = F2M[NormalOrder[Delta]];
mat[[4]] = F2M[NormalOrder[Na2right]];
mat[[5]] = F2M[NormalOrder[Nb2right]];
mat[[6]] = F2M[NormalOrder[NaNbright]];
mat[[7]] = F2M[NormalOrder[Na2Nbright]];
mat[[8]] = F2M[NormalOrder[NaNb2right]];
mat[[9]] = F2M[NormalOrder[Num4]];
mat[[10]] = F2M[NormalOrder[Hop]];
mat[[11]] = F2M[NormalOrder[HopNaright]];
mat[[12]] = F2M[NormalOrder[HopNbright]];
mat[[13]] = F2M[NormalOrder[HopN2]];
mat[[14]] = F2M[NormalOrder[Hop2]];
mat[[15]] = F2M[NormalOrder[Hspin]];
For[i = 0, i < 15, i++, Ranker = Append[Ranker,
    Flatten[mat[[i + 1]]]]]
MatrixRank[Ranker] == 15
```

Receiving an output for the last line that says `true` means that we have succeeded in finding a representation for the parameters.

5.3.4 Transformation

We are now ready to perform our desired transformation, which was to map the half filled singlet onto the vacuum. Writing the transformation using operators it will be on the form

$$U = \Delta_{2,0}\,(n-3)\,(n-4) \tag{5.2}$$

$$U^{\dagger} = \Delta^{\dagger}_{2,0}\,(n-1)\,(n-2) \tag{5.3}$$

where the operator $\Delta^{\dagger}_{2,0}$ creates the two-particle spin-0 singlet with one particle in each band. The ending parenthesis, in the equations above, are there to ensure that the operators only affect the states we are interested in exchanging.

The terms in the Hamiltonian are all preceded by parameters, whose name can be seen in the table below

oper.	in code	const.
1	One	E_0
$n_\alpha + n_\beta$	Mu	μ
$n_\alpha - n_\beta$	Delta	Δ
$n_\alpha(n_\alpha - 1)$	Na2right	N2a
$n_\beta(n_\beta - 1)$	Nb2right	N2b
$n(n-1)$	NaNbright	N2
$n_\alpha n_\beta(n_\alpha - 1)$	Na2Nbright	N3a
$n_\alpha n_\beta(n_\beta - 1)$	NaNb2right	N3b
$n(n-1)(n-2)(n-3)$	Num4	N4
$\sum_\sigma c^\dagger_{\alpha,\sigma} c_{\beta,\sigma} + h.c.$	Hop	t1
$n_\alpha \sum_\sigma c^\dagger_{\beta,\sigma} c_{\beta,\sigma}$	HopNaright	tNa
$n_\beta \sum_\sigma c^\dagger_{\alpha,\sigma} c_{\alpha,\sigma}$	HopNbright	tNb
$\sum_\sigma c^\dagger_{\alpha,\uparrow} c^\dagger_{\alpha,\downarrow} c^\dagger_{\beta,\sigma} c_{\alpha,\sigma} c_{\beta,\uparrow} c_{\beta,\downarrow} + h.c.$	HopN2	tN2
$c^\dagger_{\alpha,\uparrow} c^\dagger_{\alpha,\downarrow} c_{\beta,\uparrow} c_{\beta,\downarrow} + h.c.$	Hop2	t2
$S_\alpha \cdot S_\beta$	Hspin	J

Table 5.3: The operators listed here are also defined in the code p. 32-33.

In Mathematica we've created a routine called TroutineF that takes the fermionic expression of a parameter and then performs the mapping and calculates what parameters it gets transformed into.

```
Ordo = Ranker;
Ordoinv = PseudoInverse[Ordo];
TransU = NormalOrder[((NCM[c[1, 1], c[2, 2]] -
    - NCM[c[1, 2], c[2, 1]]) / Sqrt[2]) **
    ** (Num - 1) ** (Num - 2)/2]
TransUback = NormalOrder[((NCM[d[1, 1], d[2, 2]] -
    - NCM[d[1, 2], d[2, 1]])/Sqrt[2]) **
    ** (Num -3) ** (Num -4)/2
Symbvec = {E0, N1a, N1b, N2a, N2b, N2, N3a, N3b,
    , N4, t1, tNa, tNb, tN2, t2, J};
matsymb = {One, Na, Nb, Na2right, Nb2right, NaNbright,
    , Na2Nbright, NaNb2right, Num4, Hop, HopNaright,
    , HopNbright, HopN2, Hop2, Hspin};
place = Table[0, {i, 16}, {j, 16}];
place[[1, 1]] = 1; place[[7, 7]] = 1;
TU = F2M[TransU] + F2M[TransUback];

TroutineF2[p_] :=
    Module[{matrix, symb, ident, shift},
    matrix = F2M[NormalOrder[p]];
    ident = Transpose[place].matrix.place;
    shift = Transpose[TU].matrix.TU;
    matrix = matrix - ident + shift;
    symb = Flatten[matrix].Ordoinv;
    symb = Symbvec.symb; Return[symb]]
mapit2 = ;
For[i = 1, i < 16, i++, mapit2 = Append[mapit2,
    TroutineF2[matsymb[[i]]]]]
Transpose[{Symbvec, mapit2}] // MatrixForm]
```

6

Conclusions

We have looked at the states and energy structures of several Hubbard models. The spin 1/2 half and spin 3/2 models were looked at in order to independently verify previous results. We wanted to find a model where we could find a transformation from the half filled case to the vacuum, in order to map a dense Fermi gas to a dilute Fermi gas. By doing so we make it possible to use different theories and models from the DFG area.

The results for the spin 3/2 model and the 2-band model confirmed that in the 2-band model we could map a 2-particle singlet to the vacuum. While keeping the model as general as possible we found that a total of 15 parameters are needed to describe the system fully if we do not use any symmetries or simplifications for the model.

Choosing 15 operators for representation can be done in many ways, but some guidance to a good basis can come from considering that we may later on want to use them in a mean field analysis. When applying the mapping from a 2-particle singlet to vacuum, we have shown the parameters transform in a manner shown in Table (6.1).

With the transformation algorithm used in Mathematica it is possible to do the mapping on selected terms in a Hamiltonian for the model. This will result in a new effective Hamiltonian that may have new terms introduced by the mapping. We started with the on-site Hamiltonian (Eq. 4.5) and slightly adjusted the terms to be in accordance to the terms in the 15 parameter basis:

$$\begin{aligned} H_{on-site} = \quad & t_{\alpha\beta} \sum_\sigma \left(c^\dagger_{r,\sigma,\alpha} c_{r,\sigma,\beta} + h.c. \right) + V n \left(n - 1 \right) + U \left(n_\alpha^2 + n_\beta^2 \right) \\ & + \Delta \left(n_\alpha - n_\beta \right) + \mu (n_\alpha + n_\beta) + J \, S_\alpha \cdot S_\beta . \end{aligned} \tag{6.1}$$

Orig	After mapping
E_0	E_0
μ	$2E_0 + 2J - \mu + \frac{3N2}{4} + \frac{N2a}{4} + \frac{N2b}{4} - \frac{N3a}{2} + \frac{N3b}{2}$
Δ	Δ
$N2a$	$N2a$
$N2b$	$N2b$
$N2$	$2E_0 + 2J - 2\mu + \frac{7N2}{4} + \frac{N2a}{4} + \frac{N2b}{4} - \frac{N3a}{2} - \frac{N3b}{2}$
$N3a$	$N3a$
$N3b$	$N3b$
$N4$	$N4$
$t1$	$t1$
tNa	tNa
tNb	tNb
$tN2$	$tN2$
$t2$	$t2$
J	$-\frac{3E_0}{4} + \frac{J}{4} + \frac{3\mu}{4} - \frac{9N2}{32} - \frac{3N2a}{32} - \frac{3N2a}{32} + \frac{3N3a}{16} + \frac{3N3b}{16}$

Table 6.1: The result of mapping the general operator basis. The full operators can be found in Table (5.3).

Transforming the terms in this Hamiltonian mixes the parameters, for a few interesting parameters the result can be seen below:

Orig	New	operator
Δ	Δ	$(n_\alpha - n_\beta)$
$-\mu$	$3/4J + \mu - 2V$	$(n_\alpha + n_\beta)$
J	$1/4J - 2\mu + 2V$	$S_\alpha \cdot S_\beta$
V	$-9/32J - 3/4\mu + 7/4V$	$n\,(n-1)$

As can be seen many of the parameters gets changed, particularly the J-term change is interesting. When the hopping is small ($t_{\alpha\beta} \approx 0$), then the lowest single particle state and the singlet with one particle in each band has the same energy when

$$\mu \to \Delta - 3/4J + 2V.$$

This results in an effective spin parameter that is

$$J_{eff} \approx -2(\Delta + V) + 7/4J.$$

With a small J (or $J = 0$) this effectively means that the spin term has changed sign.

Bibliography

[1] P. Fazekas, Lecture Notes on Electron Correlation and Magnetism, World Scientific Publishing Co., Singapore, 1999.

[2] N.F. Mott, Metal-Insulator Transitions, Taylor & Francis Ltd, London, 1974.

[3] J. Hubbard, Proc. Roy. Soc. **A276** p238 (1963), London.

[4] M.C. Gutzwiller, Phys Rev Lett. **10** p159 (1963).

[5] Kanamori, Progr. Theor. Phys. **30** p275 (1963), Tokyo.

[6] E. Fradkin Field Theories of Condensed Matter Systems, Addison-Wesley, Redwood City, 1991.

[7] H. Tasaki, Progr. Theor. Phys. **99** (1998).

[8] Wolfram Research, Inc., Mathematica, Version 5.0, Champaign, IL (2003).

[9] S. Östlund, E. Mele, Phys. Rev. B **44** 12413 (1991).

[10] S. Östlund, T.H. Hanson, A. Karlhede, ArXiv Cond. matt. e-prints, cond-mat/0406717 (2004).

[11] C. Wu, J.P. Hu, S.C. Zhang, Phys. Rev. Lett. **91** p186402 (2003).