

University of Augsburg
Faculty of Theoretical Physics III



Diagrammatic high-order perturbation theory for the Gutzwiller wave function

Master's thesis

by

Marc Alexander, B.Sc.

January 2016

Contents

1	Motivation	1
2	Analytic foundation	3
2.1	Hubbard model	3
2.2	Gutzwiller wave function (GWF)	5
2.3	Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$	6
2.4	Gebhard's diagrammatic expansion	22
3	Numerical investigation	27
3.1	Generating skeleton diagrams	27
3.2	Input data for the C programs	39
3.3	General procedure for numerical approach	41
3.4	Analytic solution for the one-dimensional chain	47
3.5	Program for the one-dimensional chain	49
3.6	Program for the square lattice	65
4	Summary	76

1 Motivation

In this master's thesis the Gutzwiller wave function (GWF) for a one- and two-dimensional lattice in equilibrium is investigated. The GWF is a variational wave function (VWF) for the Hubbard model, which is the most basic model for correlated electrons in a lattice. By studying the Hubbard model we aim to understand the effects of strongly correlated electrons in real materials. The strong correlations of the electrons result in extraordinary properties of some materials like high temperature superconductors or heavy fermions. The study of the Hubbard model is the foundation for a proper theoretical description of correlated materials.

The first option to investigate the Hubbard model is to do a perturbation theory in a small parameter, but in many cases there does not exist a small parameter. Another option is to use a VWF as an educated guess for the ground state. In this thesis we choose the second option, but we must be aware of the fact that the quality of the resulting expectation values strongly depend on the choice of the VWF. The VWF reduces the infinite-dimensional Fock space of all many-body wave functions to the variational parameter space of the VWF.

Evaluating the expectation values of a correlated VWF like the GWF is still a quantum many-body problem, which in general cannot be solved analytically. We aim to evaluate the expectation values numerically with sufficient accuracy by using a diagrammatic expansion in a small parameter. The advantage of a diagrammatic expansion for a VWF compared to diagrammatic expansion for Green functions is that the correlation function has no time dependence. Hence the problem is reduced by one dimension, the dimension of time, but the dependence on real space still remains.

We use two different diagrammatic expansions. The first one starts from the Fermi sea and was developed by Metzner and Vollhardt [1] [2] [3]. The second one is very similar, but it is an expansion around the exact solution of the GWF in infinite spatial dimensions and was developed by Gebhard [4]. First of all these two expansions will be applied to an one-dimensional chain of equal lattice sites with nearest neighbor hopping of infinite length. For this one-dimensional system an analytic solution for the GWF is available. By comparing the analytic result with the two different expansions the quality and the validity in parameter space of the two expansions can be examined.

The consecutive step is to apply the diagrammatic expansions on a two-dimensional square lattice of equal sites with just nearest neighbor hopping. For this lattice there does not exist an analytic solution and our purpose is to calculate numerically with

1 Motivation

high accuracy expectation values with respect to the GWF. The quantitative validity in parameter space we deduce from the validity in parameter space in one dimension and from an error estimation procedure.

In this thesis we study only the GWF, but all techniques and procedures can also be applied for a class of generalized GWFs, which use other uncorrelated wave functions than the Fermi sea.

2 Analytic foundation

2.1 Hubbard model

In the 1960s Gutzwiller proposed an Hamiltonian \hat{H} to describe the electrons in a solid including correlation between the electrons, which is now called the Hubbard Hamiltonian [5][6][7]. The Hamiltonian consists out of a single-particle operator, the kinetic energy of the electrons \hat{T} , and of a two-particle operator, the potential energy between two electrons \hat{V} ,

$$\hat{H} = \hat{T} + \hat{V} . \quad (2.1)$$

On each lattice site i of the solid an electron with spin σ is allowed to occupy the state $|\Phi_{i\sigma}\rangle$, which is localized at the site i . For a fixed particle number N the basis of the Hilbert space are the $N \times N$ Slater determinants of the states $|\Phi_{i\sigma}\rangle$. To avoid calculating with the explicit wave functions $|\Phi_{i\sigma}\rangle$ we expand the Hilbert space to the Fock space. This is the direct sum of Hilbert spaces with particle number N varying N from zero to infinity. Then in the Fock space we are allowed to use the creation and annihilation operators $\hat{c}_{i\sigma}^\dagger$ and $\hat{c}_{i\sigma}$ of second quantization and the kinetic energy operator can be written as:

$$\hat{T} = \sum_{i,j} \sum_{\sigma=1\downarrow} -t_{ij} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \quad \text{with} \quad t_{ij} = \int \Phi_{i\sigma}^*(\mathbf{r}) \left(\frac{\hbar^2}{2m} \Delta \right) \Phi_{j\sigma}(\mathbf{r}) \, d\mathbf{r} \quad (2.2)$$

In the formula above we use two assumptions, first that t_{ij} is spin-independent and second that the states $\langle \mathbf{r} | \Phi_{i\sigma} \rangle \equiv \Phi_{i\sigma}(\mathbf{r})$ are a product of a real-space function and a spin function and the spin functions of different spins are orthogonal to each other. The potential energy in second quantization is:

$$\hat{V} = \sum_{i,j,k,l} \sum_{\sigma,\sigma'} V_{ijkl} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma'}^\dagger \hat{c}_{k\sigma'} \hat{c}_{l\sigma} . \quad (2.3)$$

The symbol V_{ijkl} is an abbreviation for the integral

$$V_{ijkl} = \int \int \Phi_{i\sigma}^*(\mathbf{r}) \Phi_{j\sigma'}^*(\mathbf{r}') \left(\frac{e^2/4\pi\epsilon_0}{|\mathbf{r} - \mathbf{r}'|} \right) \Phi_{k\sigma'}(\mathbf{r}') \Phi_{l\sigma}(\mathbf{r}) \, d\mathbf{r} \, d\mathbf{r}' . \quad (2.4)$$

The idea of the Hubbard model is to neglect all the parameters V_{ijkl} except for the ones with the highest magnitude. If the wave functions $\Phi_{i\sigma}(\mathbf{r})$ are localized at their sites i , then naturally $V_{iiii} \equiv U_i$ has the highest magnitude of all V_{ijkl} . Localization implies that $|\Phi_{i\sigma}(\mathbf{r})|^2$ decreases fast with $|\mathbf{r} - \mathbf{r}_i|$ compared to the lattice constant. The potential energy \hat{V} simplifies to

$$\hat{V} = \sum_{i,j,k,l} \sum_{\sigma,\sigma'} U_i \delta_{i,l} \delta_{j,k} \delta_{i,j} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma'}^\dagger \hat{c}_{k\sigma'} \hat{c}_{l\sigma} = \sum_i U_i \hat{c}_{i\uparrow}^\dagger \hat{c}_{i\uparrow} \hat{c}_{i\downarrow}^\dagger \hat{c}_{i\downarrow} = \sum_i U_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} , \quad (2.5)$$

2 Analytic foundation

and the Hubbard Hamiltonian is

$$\hat{H} = \sum_{i,j} \sum_{\sigma=\uparrow\downarrow} -t_{ij} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} + \sum_i U_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} . \quad (2.6)$$

If the lattice is translational invariant, then U_i becomes independent of the lattice site ($U_i \equiv U$) and the kinetic energy can be diagonalized by Fourier transformation,

$$\hat{c}_{\mathbf{k}\sigma} = \frac{1}{\sqrt{L}} \sum_j e^{i\mathbf{k}\cdot\mathbf{r}_j} \hat{c}_{j\sigma} , \quad t_{ij} \longrightarrow t_{i-j} , \quad \epsilon_{\mathbf{k}} = - \sum_{\delta} t_{\delta} e^{-i\mathbf{k}\cdot\mathbf{r}_{\delta}} . \quad (2.7)$$

The Hubbard Hamiltonian for the translational invariant case is

$$\hat{H} = \sum_{\mathbf{k}} \sum_{\sigma=\uparrow\downarrow} \epsilon_{\mathbf{k}} \hat{c}_{\mathbf{k}\sigma}^\dagger \hat{c}_{\mathbf{k}\sigma} + U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} . \quad (2.8)$$

During the rest of this thesis we use the tight-binding Hamiltonian as the kinetic energy part of the translational invariant Hubbard Hamiltonian. In the tight-binding approximation electrons can only hop from one site to the nearest neighboring site, thus the hopping amplitudes t_{ij} for nearest neighbors are finite and the rest are zero. This is reasonable in the case of localized wave functions, because the nearest neighbor amplitudes t_{ij} have the highest magnitude of all t_{ij} . The on-site terms t_{ii} just act as a chemical potential and therefore can be neglected.

Further we restrict ourselves in this thesis to an infinite equally spaced one-dimensional chain of sites and to an infinite two-dimensional square lattice. In both cases the distance to each nearest neighbor is constant and so all t_{ij} for nearest neighbors are set to the same finite value t . For the tight-binding Hamiltonian the dispersion relation $\epsilon_{\mathbf{k}}$ is

$$\epsilon_{\mathbf{k}} = -2t \sum_{n=1}^d \cos(k_n) \quad (2.9)$$

and depends only on the hopping parameter t and the dimension d . The advantage of the tight-binding model with Hubbard interaction is that it contains only two parameters t and U . So in order to capture all physics of this model we need to vary only one parameter, the ratio of U to t .

2.2 Gutzwiller wave function (GWF)

Gutzwiller not only proposed the Hubbard Hamiltonian, he also investigated it. For zero Hubbard interaction U it describes non-interacting fermions. The exact solution to non-interacting fermions is the Fermi sea $|\Psi_0\rangle$ filled up to the Fermi energy ϵ_F ,

$$|\Psi_0\rangle = \prod_{\epsilon_{\mathbf{k}} < \epsilon_F} \hat{c}_{\mathbf{k}\uparrow}^\dagger \hat{c}_{\mathbf{k}\downarrow}^\dagger |0\rangle. \quad (2.10)$$

Increasing U from zero to a finite value the exact solution of the system evolves from the Fermi sea to a new state. Because U penalizes double occupations, the double occupation of the new state is reduced compared to the Fermi sea. Gutzwiller created a variational wave function, now called the Gutzwiller wave function, to approximate the exact ground state of the system. Gutzwiller applied an operator called the Gutzwiller correlator to the Fermi sea, that projects out the doubly occupied states in $|\Psi_0\rangle$ controlled by the Gutzwiller parameter g ,

$$|\Psi_g\rangle = g^{\sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow}} |\Psi_0\rangle \quad (2.11)$$

$$= \prod_i g^{\hat{n}_{i\uparrow} \hat{n}_{i\downarrow}} |\Psi_0\rangle \quad (2.12)$$

$$= \prod_i [1 + \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} (g - 1)] |\Psi_0\rangle. \quad (2.13)$$

For $g = 1$ the GWF $|\Psi_g\rangle$ becomes the Fermi sea and for $g = 0$ it has no double occupation for half or less filling. For this reason $|\Psi_{g=1}\rangle$ is the exact solution for $U = 0$ and $|\Psi_{g=0}\rangle$ should be a good approximation for $U \rightarrow \infty$. Therefore the physical region for the variational parameter g is between zero and one.

Hence the Fermi sea $|\Psi_0\rangle$ is normalized, the GWF $|\Psi_g\rangle$ is not normalized anymore and expectation values need a normalization factor. As for any VWF the energy expectation value with respect to the GWF $\langle \hat{H} \rangle_g$ is a strict upper bound for the ground-state energy E_0 .

$$E_0 = \langle \Omega | \hat{H} | \Omega \rangle \leq \min_{g \in [0,1]} \langle \hat{H} \rangle_g \quad \text{with} \quad \langle \hat{H} \rangle_g = \frac{\langle \Psi_g | \hat{H} | \Psi_g \rangle}{\langle \Psi_g | \Psi_g \rangle} \quad (2.14)$$

The GWF is a correlated many-body state and expectation values with respect to a correlated many-body state cannot be calculated analytically in general. Exact evaluations of the expectation values with respect to the GWF are only in one spatial and infinite spatial dimensions available. During the following sections will be a discussion of two diagrammatic expansions for the GWF in finite spatial dimensions. In the next section the diagrammatic expansion by Metzner and Vollhardt will be discussed and in the section after that Gebhard's diagrammatic expansion will be explained.

2.3 Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$

In this section the diagrammatic expansion by Metzner and Vollhardt [1][2][3] for the generalized GWFs is presented. The aim of this expansion is to calculate the expectation values of the single-particle density operator and the double occupation operator. The general steps in the procedure are first an expansion in the parameter $(g^2 - 1)$, second redefining the contractions to avoid restrictions in the sums, third applying Wick's theorem to write the terms as sums of diagrams, fourth using linked-cluster theorem to get rid of vacuum diagrams, fifth defining a self-energy and the sixth and seventh steps are using Dyson equation and dressed lines to get an expansion solely in skeleton diagrams for the self-energy.

The procedures in this section are not restricted to the translational invariant Hubbard model nor the GWF. Instead all calculations are also valid for the translational non-invariant Hubbard model and for generalized GWFs. In the generalized GWFs the Fermi sea of the GWF is replaced by an arbitrary product-state of single-particle wave functions. The Gutzwiller correlator is left unchanged.

The energy expectation value with respect to the GWF $\langle \hat{H} \rangle_g$ is an upper bound for the energy of the ground-state. $\langle \hat{H} \rangle_g$ splits into the expectation values of the single-particle density operator $\langle \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \rangle_g \equiv P_{\sigma ij}$ and double occupation operator $\langle \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \rangle_g \equiv \langle \hat{D}_i \rangle_g$ for the Hubbard model.

$$\langle \hat{H} \rangle_g = - \sum_{i,j} \sum_{\sigma=\uparrow\downarrow} t_{ij} \langle \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \rangle_g + \sum_i U_i \langle \hat{D}_i \rangle_g \quad (2.15)$$

Because the GWF is not normalized the expectation value for an arbitrary operator \hat{A} is

$$\langle \hat{A} \rangle_g = \frac{\langle \Psi_g | \hat{A} | \Psi_g \rangle}{\langle \Psi_g | \Psi_g \rangle} . \quad (2.16)$$

Expansion in the small parameter $g^2 - 1$

Here the first step is to bring the terms

$$\langle \Psi_g | \Psi_g \rangle , \quad \langle \Psi_g | \hat{D}_i | \Psi_g \rangle , \quad \langle \Psi_g | \hat{n}_{i\sigma} | \Psi_g \rangle \quad \text{and} \quad \langle \Psi_g | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_g \rangle \quad \text{for} \quad i \neq j \quad (2.17)$$

2.3 Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$

in the form of an series in a small parameter. We begin with

$$\langle \Psi_g | \Psi_g \rangle = \langle \Psi_0 | g^{\sum \hat{D}_i} g^{\sum \hat{D}_j} | \Psi_0 \rangle \quad (2.18)$$

$$= \langle \Psi_0 | (g^2)^{\sum \hat{D}_i} | \Psi_0 \rangle \quad (2.19)$$

$$= \langle \Psi_0 | \prod_i [1 + \hat{D}_i (g^2 - 1)] | \Psi_0 \rangle . \quad (2.20)$$

The product over the infinite lattice i is multiplied out and we obtain a series in $(g^2 - 1)$,

$$\langle \Psi_g | \Psi_g \rangle = \sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum'_{f_1 \dots f_m} \langle \Psi_0 | \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle . \quad (2.21)$$

The prime at the sum indicates that it is to be taken only over pairwise different f_n ($f_{n_1} \neq f_{n_2}$ for $n_1 \neq n_2$). For a finite lattice the identity

$$\prod_{i=1}^L [1 + \hat{D}_i (g^2 - 1)] = \sum_{m=0}^L \frac{(g^2 - 1)^m}{m!} \sum'_{f_1, \dots, f_m=1}^L \hat{D}_{f_1} \dots \hat{D}_{f_m} \quad (2.22)$$

can easily be shown by induction over the number of lattice sites L . By using the same steps and the projector feature $\hat{D}_i = \hat{D}_i^2$ the term with the double occupation can be expanded in a similar way,

$$\langle \Psi_g | \hat{D}_i | \Psi_g \rangle = \langle \Psi_0 | g^{\sum \hat{D}_j} \hat{D}_i g^{\sum \hat{D}_l} | \Psi_0 \rangle \quad (2.23)$$

$$= \langle \Psi_0 | (g^2)^{\hat{D}_i} \hat{D}_i (g^2)^{\sum_{j \neq i} \hat{D}_j} | \Psi_0 \rangle \quad (2.24)$$

$$= \langle \Psi_0 | [1 + \hat{D}_i (g^2 - 1)] \hat{D}_i \prod_{j \neq i} [1 + \hat{D}_j (g^2 - 1)] | \Psi_0 \rangle \quad (2.25)$$

$$= \langle \Psi_0 | g^2 \hat{D}_i \left[\sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum'_{f_1 \dots f_m \neq i} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right] | \Psi_0 \rangle . \quad (2.26)$$

Exactly the same procedure for the term with the density operator gives

$$\langle \Psi_g | \hat{n}_{i\sigma} | \Psi_g \rangle = \langle \Psi_0 | [1 + \hat{D}_i (g^2 - 1)] \hat{n}_{i\sigma} \prod_{j \neq i} [1 - \hat{D}_j (g^2 - 1)] | \Psi_0 \rangle . \quad (2.27)$$

Using the identity $\hat{n}_{i\sigma} \hat{n}_{i\bar{\sigma}} = \hat{D}_i \hat{n}_{i\sigma}$ and equation 2.22 we obtain

$$\langle \Psi_g | \hat{n}_{i\sigma} | \Psi_g \rangle = \langle \Psi_0 | [\hat{n}_{i\sigma} + \hat{D}_i (g^2 - 1)] \left[\sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum'_{f_1 \dots f_m \neq i} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right] | \Psi_0 \rangle . \quad (2.28)$$

2 Analytic foundation

In the non-local term

$$\langle \Psi_g | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_g \rangle = \langle \Psi_0 | \left(g^{\hat{D}_i} \hat{c}_{i\sigma}^\dagger g^{\hat{D}_i} \right) \left(g^{\hat{D}_j} \hat{c}_{j\sigma} g^{\hat{D}_j} \right) (g^2)^{\sum_{f \neq i, j} \hat{D}_f} | \Psi_0 \rangle \quad (2.29)$$

\hat{D}_i does not commute with $\hat{c}_{i\sigma}$, so we simplify

$$g^{\hat{D}_i} \hat{c}_{i\sigma} g^{\hat{D}_i} = [1 + \hat{D}_i(g-1)] \hat{c}_{i\sigma} [1 + \hat{D}_i(g-1)] \quad (2.30)$$

with $\hat{c}_{i\sigma} \hat{n}_{i\sigma} = \hat{c}_{i\sigma}$ and $\hat{n}_{i\sigma} \hat{c}_{i\sigma} = 0$.

$$[1 + \hat{D}_i(g-1)] \hat{c}_{i\sigma} [1 + \hat{D}_i(g-1)] = [1 + \hat{n}_{i\bar{\sigma}}(g-1)] \hat{c}_{i\sigma} \quad (2.31)$$

Applying the last relation and its adjunct relation, the term can also be expanded to

$$\begin{aligned} \langle \Psi_g | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_g \rangle &= \langle \Psi_0 | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} [1 + \hat{n}_{i\bar{\sigma}}(g-1)] [1 + \hat{n}_{j\bar{\sigma}}(g-1)] \\ &\quad \left[\sum_{m=0}^{\infty} \frac{(g^2-1)^m}{m!} \sum'_{f_1 \dots f_m \neq i, j} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right] | \Psi_0 \rangle. \end{aligned} \quad (2.32)$$

We thus finished the expansion of the four terms in the small parameter $(g^2 - 1)$.

Introducing a proper contraction

Next we replace the contraction $\langle \Psi_0 | \dots | \Psi_0 \rangle$ with a new contraction $\{ \dots \}_0$ in order to drop the restrictions in the sums for the terms

$$\sum'_{f_1 \dots f_m} \langle \Psi_0 | \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle, \quad (2.33)$$

$$\sum'_{f_1 \dots f_m \neq i} \langle \Psi_0 | \hat{D}_i \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle, \quad (2.34)$$

$$\sum'_{f_1 \dots f_m \neq i} \langle \Psi_0 | \hat{n}_{i\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle, \quad (2.35)$$

$$\sum'_{f_1 \dots f_m \neq i, j} \langle \Psi_0 | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle, \quad (2.36)$$

$$\sum'_{f_1 \dots f_m \neq i, j} \langle \Psi_0 | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{n}_{i\bar{\sigma}} \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle, \quad (2.37)$$

$$\sum'_{f_1 \dots f_m \neq i, j} \langle \Psi_0 | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{n}_{j\bar{\sigma}} \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle, \quad (2.38)$$

$$\sum'_{f_1 \dots f_m \neq i, j} \langle \Psi_0 | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{n}_{i\bar{\sigma}} \hat{n}_{j\bar{\sigma}} \hat{D}_{f_1} \dots \hat{D}_{f_m} | \Psi_0 \rangle. \quad (2.39)$$

2.3 Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$

In all terms we write \hat{D}_p operator as $\hat{n}_{p\sigma}\hat{n}_{p\bar{\sigma}}$ and then sort all operators with σ to the left and all with $\bar{\sigma}$ to the right. Because contractions with respect to $|\Psi_0\rangle$ of two \hat{c} operators with different spin are always zero the expectation value splits into a product of the expectation value of the σ -spin operators and the expectation of the $\bar{\sigma}$ -spin operators. Next we normal order the operators and then we are able to drop the restrictions in the sums, because for all restricted combinations of indices the terms vanishes due to two or more identical creation or annihilation operators. The result will be defined as the new contraction $\{\dots\}_0$ of the original operators in the old contraction $\langle\Psi_0|\dots|\Psi_0\rangle$. These applied steps for term 2.39 are:

$$\begin{aligned}
& \sum'_{f_1\dots f_m \neq i,j} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma}\hat{n}_{i\bar{\sigma}}\hat{n}_{j\bar{\sigma}}\hat{D}_{f_1}\dots\hat{D}_{f_m}|\Psi_0\rangle \\
&= \sum'_{f_1\dots f_m \neq i,j} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma}\hat{n}_{i\bar{\sigma}}\hat{n}_{j\bar{\sigma}}\hat{n}_{f_1\sigma}\hat{n}_{f_1\bar{\sigma}}\dots\hat{n}_{f_m\sigma}\hat{n}_{f_m\bar{\sigma}}|\Psi_0\rangle \\
&= \sum'_{f_1\dots f_m \neq i,j} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma}\hat{n}_{f_1\sigma}\dots\hat{n}_{f_m\sigma}\hat{n}_{i\bar{\sigma}}\hat{n}_{j\bar{\sigma}}\hat{n}_{f_1\bar{\sigma}}\dots\hat{n}_{f_m\bar{\sigma}}|\Psi_0\rangle \\
&= \sum'_{f_1\dots f_m \neq i,j} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma}\hat{n}_{f_1\sigma}\dots\hat{n}_{f_m\sigma}|\Psi_0\rangle\langle\Psi_0|\hat{n}_{i\bar{\sigma}}\hat{n}_{j\bar{\sigma}}\hat{n}_{f_1\bar{\sigma}}\dots\hat{n}_{f_m\bar{\sigma}}|\Psi_0\rangle \\
&= \sum'_{f_1\dots f_m \neq i,j} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma}\hat{c}_{f_1\sigma}^\dagger\hat{c}_{f_1\sigma}\dots\hat{c}_{f_m\sigma}^\dagger\hat{c}_{f_m\sigma}|\Psi_0\rangle \\
&\quad \langle\Psi_0|\hat{c}_{i\bar{\sigma}}^\dagger\hat{c}_{i\bar{\sigma}}\hat{c}_{j\bar{\sigma}}^\dagger\hat{c}_{j\bar{\sigma}}\hat{c}_{f_1\bar{\sigma}}^\dagger\hat{c}_{f_1\bar{\sigma}}\dots\hat{c}_{f_m\bar{\sigma}}^\dagger\hat{c}_{f_m\bar{\sigma}}|\Psi_0\rangle \\
&= \sum'_{f_1\dots f_m \neq i,j} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma}\hat{c}_{f_1\sigma}^\dagger\dots\hat{c}_{f_m\sigma}^\dagger\hat{c}_{f_m\sigma}\dots\hat{c}_{f_1\sigma}\hat{c}_{j\sigma}|\Psi_0\rangle \\
&\quad \langle\Psi_0|\hat{c}_{i\bar{\sigma}}^\dagger\hat{c}_{i\bar{\sigma}}\hat{c}_{j\bar{\sigma}}^\dagger\hat{c}_{j\bar{\sigma}}\hat{c}_{f_1\bar{\sigma}}^\dagger\dots\hat{c}_{f_m\bar{\sigma}}^\dagger\hat{c}_{f_m\bar{\sigma}}\dots\hat{c}_{f_1\bar{\sigma}}\hat{c}_{j\bar{\sigma}}|\Psi_0\rangle \\
&= \sum'_{f_1\dots f_m \neq i,j} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{f_1\sigma}^\dagger\dots\hat{c}_{f_m\sigma}^\dagger\hat{c}_{f_m\sigma}\dots\hat{c}_{f_1\sigma}\hat{c}_{j\sigma}|\Psi_0\rangle \\
&\quad \langle\Psi_0|\hat{c}_{i\bar{\sigma}}^\dagger\hat{c}_{j\bar{\sigma}}^\dagger\hat{c}_{f_1\bar{\sigma}}^\dagger\dots\hat{c}_{f_m\bar{\sigma}}^\dagger\hat{c}_{f_m\bar{\sigma}}\dots\hat{c}_{f_1\bar{\sigma}}\hat{c}_{j\bar{\sigma}}\hat{c}_{i\bar{\sigma}}|\Psi_0\rangle \\
&= \sum_{f_1\dots f_m} \langle\Psi_0|\hat{c}_{i\sigma}^\dagger\hat{c}_{f_1\sigma}^\dagger\dots\hat{c}_{f_m\sigma}^\dagger\hat{c}_{f_m\sigma}\dots\hat{c}_{f_1\sigma}\hat{c}_{j\sigma}|\Psi_0\rangle \\
&\quad \langle\Psi_0|\hat{c}_{i\bar{\sigma}}^\dagger\hat{c}_{j\bar{\sigma}}^\dagger\hat{c}_{f_1\bar{\sigma}}^\dagger\dots\hat{c}_{f_m\bar{\sigma}}^\dagger\hat{c}_{f_m\bar{\sigma}}\dots\hat{c}_{f_1\bar{\sigma}}\hat{c}_{j\bar{\sigma}}\hat{c}_{i\bar{\sigma}}|\Psi_0\rangle \\
&\equiv \sum_{f_1\dots f_m} \left\{ \hat{c}_{i\sigma}^\dagger\hat{c}_{j\sigma}\hat{n}_{i\bar{\sigma}}\hat{n}_{j\bar{\sigma}}\hat{D}_{f_1}\dots\hat{D}_{f_m} \right\}_0 \tag{2.40}
\end{aligned}$$

Both contractions $\langle\Psi_0|\dots|\Psi_0\rangle$ and $\{\dots\}_0$ have the same value at each of the non-restricted indices combination and at every restricted the new contraction $\{\dots\}_0$ is zero. Defining the new contraction $\{\dots\}_0$ was the second step in this section and we

2 Analytic foundation

can rewrite terms of equations 2.21, 2.26, 2.28 and 2.32 and drop the restrictions in the summations:

$$\hat{X} = \sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \hat{D}_{f_1} \dots \hat{D}_{f_m} , \quad (2.41)$$

$$\langle \Psi_g | \Psi_g \rangle = \left\{ \hat{X} \right\}_0 , \quad (2.42)$$

$$\langle \Psi_g | \hat{D}_i | \Psi_g \rangle = g^2 \left\{ \hat{D}_i \hat{X} \right\}_0 , \quad (2.43)$$

$$\langle \Psi_g | \hat{n}_{i\sigma} | \Psi_g \rangle = \left\{ \hat{n}_{i\sigma} [1 + \hat{n}_{i\bar{\sigma}}(g^2 - 1)] \hat{X} \right\}_0 , \quad (2.44)$$

$$\langle \Psi_g | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_g \rangle = \left\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} [1 + \hat{n}_{i\bar{\sigma}}(g - 1)] [1 + \hat{n}_{j\bar{\sigma}}(g - 1)] \hat{X} \right\}_0 . \quad (2.45)$$

Equation 2.45 is only valid for $i \neq j$ and equation 2.44 is the case $i = j$. These two equations can be combined to the formula

$$\begin{aligned} \langle \Psi_g | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_g \rangle = \sum_{f_1 \dots f_m} \left\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} [1 + (g - 1)(\hat{n}_{i\bar{\sigma}} + \hat{n}_{j\bar{\sigma}}) \right. \\ \left. + (g - 1)^2 (\hat{n}_{i\bar{\sigma}} \hat{n}_{j\bar{\sigma}} + \delta_{ij} \hat{n}_{i\bar{\sigma}})] \hat{X} \right\}_0 , \end{aligned} \quad (2.46)$$

which is valid for all i and j . Because $\{\dots \hat{n}_{i\bar{\sigma}} \hat{n}_{i\bar{\sigma}} \dots\}_0$ is zero due to the definition of $\{\dots\}_0$, the Kronecker delta appears.

Diagrammatic expansion

The third step is to write equations 2.42, 2.43 and 2.66 as sums of diagrams. Those three terms have the general form

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0 = \sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0 . \quad (2.47)$$

The operator $\hat{x}_{ij\sigma}$ contains the scalar factors and the \hat{c} operators with index i and j . When we apply Wick's theorem on $\{\dots\}_0$ it becomes a sum of products of $\langle \Psi_0 | \hat{c}_{p\sigma}^\dagger \hat{c}_{q\sigma} | \Psi_0 \rangle \equiv P_{\sigma qp}^0$.

$$\sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0 = \sum_{f_1 \dots f_m} \left[\sum_{\{p,q,r,s\}} (-1)^{f(\{p,q,r,s\})} \left(\prod_{\langle p,q \rangle} P_{\sigma qp}^0 \right) \left(\prod_{\langle r,s \rangle} P_{\sigma rs}^0 \right) \right] \quad (2.48)$$

The product of the $P_{\sigma qp}^0$ can be translated into a diagram. Every appearing index becomes a vertex and every $P_{\sigma qp}^0$ becomes a directed colored line from q to p . Black

2.3 Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$

lines indicate spin σ and orange spin $\bar{\sigma}$. One example of how to translate the product into a diagram is:

$$P_{\sigma i f_1}^0 P_{\sigma f_1 j}^0 P_{\bar{\sigma} i j}^0 P_{\bar{\sigma} j i}^0 P_{\bar{\sigma} f_1 f_1}^0 = \begin{array}{c} \text{orange line} \\ \downarrow \\ \text{triangle with } f_1 \text{ at top, } i \text{ at bottom-left, } j \text{ at bottom-right} \\ \text{with orange arcs between } i \text{ and } j \end{array} \quad (2.49)$$

Some of the diagrams from Wick's theorem are topologically equivalent, in this examination two diagrams are topologically equivalent, if changing the indices f_i of one diagram in the right way, the reverse translation of both diagrams into a product of $P_{\sigma qp}^0$ are identical. These diagrams

$$\begin{array}{c} f_1 \text{ --- } f_2 \\ \text{orange arcs between } f_1 \text{ and } f_2 \\ \text{orange lines from } f_1 \text{ and } f_2 \text{ to } i \end{array} \quad \text{and} \quad \begin{array}{c} f_1 \text{ --- } f_2 \\ \text{orange arcs between } f_1 \text{ and } f_2 \\ \text{orange lines from } f_1 \text{ and } f_2 \text{ to } i \end{array} \quad (2.50)$$

for example are topologically equivalent. Wick's theorem creates every possible contraction combination ones, so there are always $m!$ topologically equivalent diagrams, $m!$ is the number of different possible permutations of f_1, \dots, f_m . After the summation over the lattice for all f_i all topologically equivalent diagrams have the same value and the same sign $+1$ or -1 . Hence the sum created by applying Wick's theorem can be rewritten as the sum over all topologically distinct diagrams G times $m!$,

$$\sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0 = m! \sum_G (-1)^{f(G)} v(G) . \quad (2.51)$$

Here $v(G)$ is the value of the diagram G meaning the sum of f_1, \dots, f_m over the product of $P_{\sigma qp}^0$ and $f(G)$ is the sign function of a diagram G . The general form (2.47) can be expressed as a sum of topologically distinct diagrams:

$$\sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0 = \sum_{m=0}^{\infty} (g^2 - 1)^m \sum_G (-1)^{f(G)} v(G) . \quad (2.52)$$

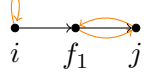
The diagrams G depend on the operator $\hat{x}_{ij\sigma}$ and the order m . They have the same topology as the diagrams in the so-called φ^4 statistical field theory. Detailed Feynman diagram rules are presented later in this section.

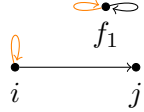
Linked-cluster theorem

Further comes the fourth step applying the linked-cluster theorem. This cancels the disconnected diagrams of the numerator of an expectation value with the vacuum diagrams of the denominator of the expectation value. We categorize the diagrams into two groups, connected and disconnected diagrams. In a connected diagram a

2 Analytic foundation

vertex has a continuous connection to every other vertex, in a disconnected at least between two vertices does not exist a continuous connection:

connected diagram:  (2.53)

disconnected diagram:  (2.54)

Disconnected diagrams can be split up into a product of a smaller connected diagram and vacuum diagrams. The connected diagram contains the vertices i and j and the vacuum diagrams have solely the vertices f_i . The term $\langle \Psi_g | \Psi_g \rangle$ is one plus all vacuum diagrams, because it does not have any operator with index i or j . An example of how to split a disconnected diagram is shown below:

$$\begin{aligned} \sum_{f_1} \left(\begin{array}{c} \text{self-loop at } f_1 \\ \downarrow \\ \begin{array}{ccc} \bullet & \xrightarrow{\quad} & \bullet \\ i & & j \end{array} \end{array} \right) &= \sum_{f_1} (P_{\sigma ij}^0 P_{\sigma f_1 f_1}^0 P_{\bar{\sigma} ii}^0 P_{\bar{\sigma} f_1 f_1}^0) \\ &= (P_{\sigma ij}^0 P_{\bar{\sigma} ii}^0) \left(\sum_{f_1} P_{\sigma f_1 f_1}^0 P_{\bar{\sigma} f_1 f_1}^0 \right) \\ &= \left(\begin{array}{ccc} \bullet & \xrightarrow{\quad} & \bullet \\ i & & j \end{array} \right) \left(\sum_{f_1} \begin{array}{c} \text{self-loop at } f_1 \end{array} \right) \end{aligned} \quad (2.55)$$

We take the general form 2.47 and split Wick's theorem into a term with solely connected diagrams and a second term with solely disconnected diagrams.

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0 = \left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0^c + \left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0^{\text{disc}} \quad (2.56)$$

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0^c = \sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0^c \quad (2.57)$$

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0^{\text{disc}} = \sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0^{\text{disc}} \quad (2.58)$$

The superscript c means that after applying Wick's theorem the connected diagrams are kept and the disconnected are discarded. The superscript 'disc' is the opposite and means to keep the disconnected and omit the connected diagrams. All disconnected

2.3 Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$

diagrams can be written as a product of a smaller connected diagram and a vacuum diagram.

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0^{\text{disc}} = \sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{n=1}^{m-1} \left(\frac{m!}{n!(m-n)!} \sum_{f_1 \dots f_n} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_n} \right\}_0^c \right) \left(\sum_{f_{n+1} \dots f_m} \left\{ \hat{D}_{f_{n+1}} \dots \hat{D}_{f_m} \right\}_0 \right) \quad (2.59)$$

The factor $\frac{m!}{n!(m-n)!}$ is necessary here, due to the fact that $\hat{x}_{ij\sigma}$ can be contracted with n different \hat{D}_{f_i} from $\hat{D}_{f_1}, \dots, \hat{D}_{f_m}$,

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0^{\text{disc}} = \sum_{m=0}^{\infty} \sum_{n=1}^{m-1} \left(\frac{(g^2 - 1)^n}{n!} \sum_{f_1 \dots f_n} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_n} \right\}_0^c \right) \left(\frac{(g^2 - 1)^{m-n}}{(m-n)!} \sum_{f_1 \dots f_{m-n}} \left\{ \hat{D}_{f_1} \dots \hat{D}_{f_{m-n}} \right\}_0 \right) \cdot \quad (2.60)$$

The double sum over m and n can be written as a product of sums of n and m with the upper limit ∞ .

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0^{\text{disc}} = \left(\sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0^c \right) \left(\sum_{n=1}^{\infty} \frac{(g^2 - 1)^n}{n!} \sum_{f_1 \dots f_n} \left\{ \hat{D}_{f_1} \dots \hat{D}_{f_n} \right\}_0 \right) \quad (2.61)$$

If we let the sum over n start with $n = 0$, additionally we get the term of the connected diagrams times 1. So finally the general form 2.47 is a product of the connected diagrams and the vacuum diagrams,

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0 = \left(\sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0^c \right) \left(\sum_{n=0}^{\infty} \frac{(g^2 - 1)^n}{n!} \sum_{f_1 \dots f_n} \left\{ \hat{D}_{f_1} \dots \hat{D}_{f_n} \right\}_0 \right) \cdot \quad (2.62)$$

The term of the vacuum diagrams is identical to $\langle \Psi_g | \Psi_g \rangle$ (2.42).

$$\left\{ \hat{x}_{ij\sigma} \hat{X} \right\}_0 = \left(\sum_{m=0}^{\infty} \frac{(g^2 - 1)^m}{m!} \sum_{f_1 \dots f_m} \left\{ \hat{x}_{ij\sigma} \hat{D}_{f_1} \dots \hat{D}_{f_m} \right\}_0^c \right) \langle \Psi_g | \Psi_g \rangle \quad (2.63)$$

2 Analytic foundation

Consequently in the expectation values the factor $\langle \Psi_g | \Psi_g \rangle$ from the disconnected diagrams cancels with $\langle \Psi_g | \Psi_g \rangle$ from the normalization. The expectation values of the double occupation

$$\langle \hat{D}_i \rangle = g^2 \left\{ \hat{D}_i \hat{X} \right\}_0^c \quad (2.64)$$

and single-particle density matrix

$$P_{\sigma ij} \equiv \frac{\langle \Psi_g | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_g \rangle}{\langle \Psi_g | \Psi_g \rangle} \quad (2.65)$$

$$= \left\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} [1 + (g-1)(\hat{n}_{i\bar{\sigma}} + \hat{n}_{j\bar{\sigma}}) + (g-1)^2(\hat{n}_{i\bar{\sigma}}\hat{n}_{j\bar{\sigma}} + \delta_{ij}\hat{n}_{i\bar{\sigma}})] \hat{X} \right\}_0^c. \quad (2.66)$$

Wick's theorem improved the expansion in two ways. First the vacuum and disconnected diagrams do not have to be evaluated. Second we gain an expansion in the small parameter $(g^2 - 1)$ for the expectation values, while before we had separate expansions for the nominator and the denominator of the expectation values.

Self-energy

In a fifth step we define a self-energy $S_{\sigma ij}$ from which we can calculate the expectation values $\langle \hat{D}_i \rangle_g$ and $P_{\sigma ij}$ in terms of a simpler diagrammatic expansion. The self-energy $S_{\sigma ij}$ is defined as

$$S_{\sigma ij} = \left\{ [(g^2 - 1)^2 c_{i\sigma}^\dagger c_{j\sigma} \hat{n}_{i\bar{\sigma}} \hat{n}_{j\bar{\sigma}} - (g^2 - 1) \delta_{ij} \hat{n}_{i\bar{\sigma}}] \hat{X} \right\}_0^c, \quad (2.67)$$

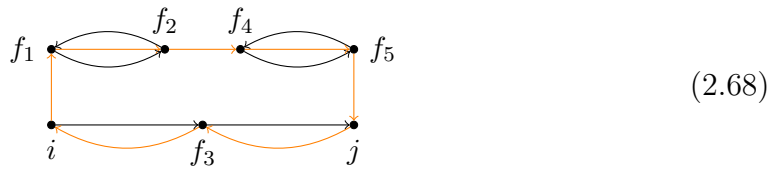
and it contains the same diagrams as the self-energy in the Green-function formalism. The same tools as in the Green-function formalism are available to construct the diagrams and to reduce the amount of diagrams, which we have to evaluate. One way to calculate $S_{\sigma ij}$ is to use Wick's theorem and to dismiss all disconnected diagrams. Wick's theorem gives also the sign of each diagram. Another way is to determine rules, that create all diagrams up to a order n in the factor $(g^2 - 1)$ with the correct sign and values. The rules to determine the order n in $(g^2 - 1)$ of $S_{\sigma ij}$ are shown in figure 1.

2.3 Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$

1. Draw all topologically distinct connected diagrams with n vertices. Between the vertices run $n - 1$ directed lines with spin σ and n directed lines with spin $\bar{\sigma}$. At each vertex each kind of spin has at most one incoming line and one outgoing line.
2. In each diagram the vertex without incoming σ line has index i and the vertex without outgoing σ line has index j . If i and j fall together on the same vertex, the vertex is index i and the diagram gets an factor δ_{ij} . The remaining vertices are indexed f_1 to f_m . m is equal to $n - 2$, if vertex i and j are distinct, else it is $n - 1$.
3. Each vertex gives a factor $(g^2 - 1)$ and every line gives a factor $P_{\sigma'qp}^0$, where σ' is the spin, q is the index of the outgoing vertex and p of the incoming vertex of the line.
4. To obtain the value of a diagram sum the indices f_1 to f_m over the product of all factors of the diagram.
5. The sign of the diagram is $(-1)^\ell$, where ℓ is the number of closed spin loops in the diagram.
6. Finally sum over all diagrams, each given by the product of its sign and value.

Figure 1: Rules to create the self-energy diagrams

The distinction between $i = j$ and $i \neq j$ in the first rule is due to the fact that δ_{ij} is zero for $i \neq j$ and $\{\dots \hat{n}_{i\bar{\sigma}} \hat{n}_{j\bar{\sigma}} \dots\}_0$ is zero for $i = j$. The rule about the sign is determined by the number of closed spin loops and needs a second thought. We start with an example of how to count closed spin loops:



The orange lines of spin $\bar{\sigma}$ build one closed loop from i to f_1 to f_2 to f_4 to f_5 to j to f_3 and back to i . Two closed loops are formed by the black lines of spin σ from f_1 to f_2 and back and the second is from f_4 to f_5 and back. The black line from i to f_3 and ending at j is not a closed loop, because it has a beginning and an ending. So the sign of this example diagram is $(-1)^3 = -1$. Deriving this rule is quite simple.

2 Analytic foundation

Whenever two spin loops get combined to a bigger spin loop in an otherwise identical diagram we gain a relative factor (-1) . This is valid for combining small loops

$$\begin{array}{c} \{c_{i\sigma}^\dagger c_{i\sigma}\}_0 \{c_{j\sigma}^\dagger c_{j\sigma}\}_0 \\ \begin{array}{cc} \bullet & \bullet \\ \leftarrow & \rightarrow \\ i & j \end{array} \end{array} \longrightarrow - \begin{array}{c} \{c_{j\sigma}^\dagger c_{i\sigma}\}_0 \{c_{i\sigma}^\dagger c_{j\sigma}\}_0 \\ \begin{array}{cc} \bullet & \bullet \\ \leftarrow & \rightarrow \\ i & j \end{array} \end{array} \quad (2.69)$$

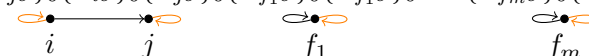
and combining arbitrary large loops

$$\{c_{i_1\sigma}^\dagger c_{i_2\sigma}\}_0 \cdots \{c_{i_{n-1}\sigma}^\dagger c_{i_n\sigma}\}_0 \{c_{i_n\sigma}^\dagger c_{i_1\sigma}\}_0 \{c_{j_1\sigma}^\dagger c_{j_2\sigma}\}_0 \cdots \{c_{j_{m-1}\sigma}^\dagger c_{j_m\sigma}\}_0 \{c_{j_m\sigma}^\dagger c_{j_1\sigma}\}_0 = \quad (2.70)$$

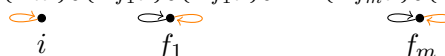
$$\{c_{i_1\sigma}^\dagger c_{i_2\sigma}\}_0 \cdots \{c_{i_{n-1}\sigma}^\dagger c_{i_n\sigma}\}_0 \{c_{i_n\sigma}^\dagger c_{j_1\sigma}\}_0 \{c_{j_m\sigma}^\dagger c_{j_1\sigma}\}_0 \{c_{j_1\sigma}^\dagger c_{j_2\sigma}\}_0 \cdots \{c_{j_{m-1}\sigma}^\dagger c_{j_m\sigma}\}_0 = \quad (2.71)$$

$$- \{c_{i_1\sigma}^\dagger c_{i_2\sigma}\}_0 \cdots \{c_{i_{n-1}\sigma}^\dagger c_{i_n\sigma}\}_0 \{c_{i_n\sigma}^\dagger c_{j_1\sigma}\}_0 \{c_{j_m\sigma}^\dagger c_{i_1\sigma}\}_0 \{c_{j_1\sigma}^\dagger c_{j_2\sigma}\}_0 \cdots \{c_{j_{m-1}\sigma}^\dagger c_{j_m\sigma}\}_0 . \quad (2.72)$$

If the number of spin loops changes by one, the sign also changes. Therefore counting the closed spin loops gives the correct relative sign. On the one hand the diagram

$$(g^2 - 1)^{2+m} \{c_{i\sigma}^\dagger c_{j\sigma}\}_0 \{\hat{n}_{i\bar{\sigma}}\}_0 \{\hat{n}_{j\bar{\sigma}}\}_0 \{\hat{n}_{f_1\sigma}\}_0 \{\hat{n}_{f_1\bar{\sigma}}\}_0 \cdots \{\hat{n}_{f_m\sigma}\}_0 \{\hat{n}_{f_m\bar{\sigma}}\}_0 \quad (2.73)$$


has an even number of spin loops and $+1$ as a sign, so all diagrams with i and j on different vertices have a plus sign if they have an even number of spin loops, else a minus sign. On the other hand the diagram

$$-(g^2 - 1)^{1+m} \delta_{ij} \{\hat{n}_{i\bar{\sigma}}\}_0 \{\hat{n}_{f_1\sigma}\}_0 \{\hat{n}_{f_1\bar{\sigma}}\}_0 \cdots \{\hat{n}_{f_m\sigma}\}_0 \{\hat{n}_{f_m\bar{\sigma}}\}_0 \quad (2.74)$$


sets the sign of all diagrams with $i = j$ and odd number of spin loops to minus one. We also need the connection of $S_{\sigma ij}$ to $\langle \hat{D}_i \rangle_g$ and $P_{\sigma ij}$. With the identities

$$\left\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{X} \right\}_0^c = P_{\sigma ij}^0 + \sum_{q,p} P_{\sigma iq}^0 S_{\sigma qp} P_{\sigma qj}^0 , \quad (2.75)$$

$$\left\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{n}_{i\bar{\sigma}} \hat{X} \right\}_0^c = \frac{-1}{g^2 - 1} \sum_q S_{\sigma iq} P_{\sigma qj}^0 , \quad (2.76)$$

$$\left\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{n}_{j\bar{\sigma}} \hat{X} \right\}_0^c = \frac{-1}{g^2 - 1} \sum_q P_{\sigma iq}^0 S_{\sigma qj} , \quad (2.77)$$

$$\left\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \hat{n}_{i\bar{\sigma}} \hat{n}_{j\bar{\sigma}} \hat{X} \right\}_0^c = \frac{1}{(g^2 - 1)^2} S_{\sigma ij} (1 - \delta_{ij}) , \quad (2.78)$$

$$\left\{ \hat{D}_i \hat{X} \right\}_0^c = \frac{-1}{g^2 - 1} \sum_q P_{\sigma iq}^0 S_{\sigma qi} , \quad (2.79)$$

$\langle \hat{D}_i \rangle_g$ and $P_{\sigma ij}$ can easily be expressed by $S_{\sigma ij}$

$$\langle \hat{D}_i \rangle_g = \frac{g^2}{g^2 - 1} \sum_q P_{\sigma iq}^0 S_{\sigma qi} , \quad (2.80)$$

$$P_{\sigma ij} = P_{\sigma ij}^0 - \delta_{ij} \left[\frac{S_{\sigma ij}}{(g+1)^2} + \frac{g-1}{g+1} \left(\sum_q P_{\sigma iq}^0 S_{\sigma qi} \right) \right] + \sum_{q,p} \left[\left(P_{\sigma iq}^0 - \frac{\delta_{iq}}{g+1} \right) S_{\sigma qp} \left(P_{\sigma qj}^0 - \frac{\delta_{pj}}{g+1} \right) \right] . \quad (2.81)$$

The validity of identity 2.78 is obvious from the definition of $S_{\sigma ij}$, the rest of the identities can be explained by a diagrammatic argumentation. The argumentation for the second identity 2.76 will be outlined in the following as an example for all identities.

The left hand side of equation 2.76 can be expressed in diagrams which can be constructed by rules which differ slightly from the rules for $S_{\sigma ij}$. Draw all topologically different connected diagrams with vertex i , vertex j and vertices f_1 to f_n . Vertex j has only one incoming σ line, the vertex i has one outgoing σ line and one outgoing and one incoming $\bar{\sigma}$, line and the other vertices have each one incoming and one outgoing σ and $\bar{\sigma}$ line. All vertices f_1 to f_n give an factor $g^2 - 1$ and the vertices i and j only a factor 1. The sign of a diagram is $-(-1)^\ell$, with ℓ the number of spin loops. The rest of the rules is identical to the rules for $S_{\sigma ij}$. The diagrams of $S_{\sigma iq}$ become a diagram of the left hand side of 2.76 if we connect the vertex q of $S_{\sigma iq}$ with a σ line to the new vertex j , sum q over the lattice and multiply with the factor $-(g^2 - 1)^{-1}$. The right hand side of 2.76 yields exactly the same, because it takes $S_{\sigma iq}$ connects it with the vertex j with a σ line $P_{\sigma qj}^0$, sums q over the lattice and multiplies everything with the factor $-(g^2 - 1)^{-1}$.

For the other identities the argumentation is similar, the diagrams of $S_{\sigma ij}$ need one or two additional lines, some vertices do not have the factor $g^2 - 1$, and there can be a global factor -1 . These diagrammatic techniques are well known from the Green-function formalism and translate directly to the GWF.

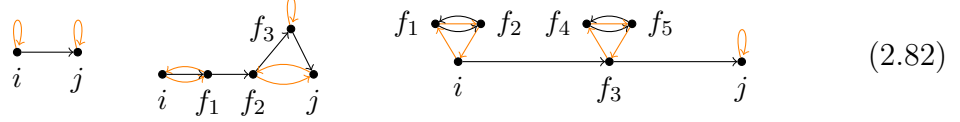
Dyson equation

As in the Green-function formalism the proper self-energy $S_{\sigma ij}^*$ defined without the one-particle-reducible diagrams by using the Dyson equation. The diagrammatic expansion for the proper self-energy has the advantage that we have to evaluate fewer diagrams, even if we have to solve the Dyson equation instead.

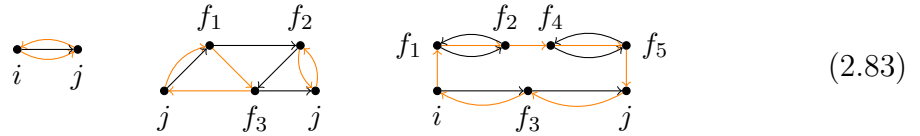
A one-particle-reducible diagram is a connected diagram of $S_{\sigma ij}$ which splits into a

2 Analytic foundation

disconnected diagram by removing one line. A one-particle-irreducible diagram is the opposite by removing any line the diagram is still connected afterwards. Examples for one-particle-reducible diagrams are:



In the left example by removing $P_{\sigma ij}^0$ the second order diagram separates into the product of two first-order diagrams. The example on the right can be split into a disconnected diagram by removing $P_{\sigma if_3}^0$ or $P_{\sigma f_3 j}^0$. Note that always a σ line has to be removed. Furthermore a one-particle-reducible diagram of order n can be constructed by connecting a lower order one-particle-irreducible diagram of order n_1 with a lower order connected diagram of order n_2 by a σ line. Of course $n = n_1 + n_2$ and the two vertices that got connected over the σ line each have to be summed over the lattice. Some examples for the one-particle-irreducible diagrams are:



The proper self-energy $S_{\sigma ij}^*$ is defined as only the one-particle-irreducible diagrams of $S_{\sigma ij}$,

$$S_{\sigma ij}^* = \left\{ [(g^2 - 1)^2 c_{i\sigma}^\dagger c_{j\sigma} \hat{n}_{i\bar{\sigma}} \hat{n}_{j\bar{\sigma}} - (g^2 - 1) \delta_{ij} \hat{n}_{i\bar{\sigma}} \hat{X}]_0^{\text{irr}} \right\}. \quad (2.84)$$

The superscript 'irr' indicates that disconnected and one-particle-reducible diagrams were removed. The self-energy $S_{\sigma ij}$ is the sum of $S_{\sigma ij}^*$ and all the one-particle-reducible diagrams. In order to create each one-particle-reducible diagram ones we need to connect the one-particle-irreducible diagrams to chains. All possible chains of two one-particle-irreducible diagrams are constructed by

$$\sum_{q,p} S_{\sigma iq}^* P_{\sigma qp}^0 S_{\sigma pj}^*. \quad (2.85)$$

$S_{\sigma ij}$ and $S_{\sigma ij}^*$ are entries of the $L \times L$ -Matrices S_σ and S_σ^* and L is the number of lattice sites. Then the Matrix S_σ is S_σ^* plus the sum of infinite sum of chains of S_σ^* -diagrams,

$$S_\sigma = S_\sigma^* + S_\sigma^* P_\sigma^0 S_\sigma^* + S_\sigma^* P_\sigma^0 S_\sigma^* P_\sigma^0 S_\sigma^* + \dots \quad (2.86)$$

$$= \left(\sum_{n=0}^{\infty} (S_\sigma^* P_\sigma^0)^n \right) S_\sigma^* \quad (2.87)$$

$$= (1 - S_\sigma^* P_\sigma^0)^{-1} S_\sigma^*. \quad (2.88)$$

This equation can be written without inversion as

$$S_\sigma = S_\sigma^* + S_\sigma^* P_\sigma^0 S_\sigma . \quad (2.89)$$

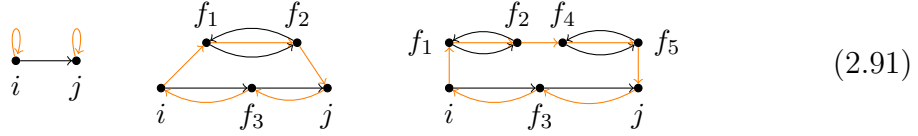
The matrix equation can be written explicitly as

$$S_{\sigma ij} = S_{\sigma ij}^* + \sum_{q,p} S_{\sigma iq}^* P_{\sigma qp}^0 S_{\sigma pj} , \quad (2.90)$$

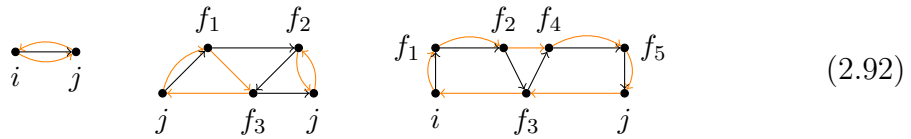
and this final equation is called the Dyson equation and connects the self-energy with the proper self-energy.

Dressed lines

By replacing the lines $P_{\sigma ij}^0$ with dressed lines the proper self-energy can be expanded solely in skeleton diagrams. A skeleton diagram is a one-particle-irreducible diagram, which cannot be split into a disconnected diagram by removing two lines. Examples for non-skeleton diagrams are:



The left diagram is not a skeleton diagram, because it is not even an one-particle-irreducible diagram. The diagram in the middle becomes disconnected if the lines from i to f_1 and f_2 to j are removed. For the right diagram exists three different choices of removing two lines to obtain a disconnected diagram. Examples for skeleton diagrams are:



We define dressed lines as

$$\bar{P}_{\sigma ij} \equiv \left\{ c_{i\sigma}^\dagger c_{j\sigma} \hat{X} \right\}_0^c = P_{\sigma ij}^0 + \sum_{q,p} P_{\sigma iq}^0 S_{\sigma qp} S_{\sigma qj}^0 . \quad (2.93)$$

The diagrams of the dressed line $\bar{P}_{\sigma ij}$ are the regular σ line from i to j plus the self-energy diagrams $S_{\sigma qp}$ connected to the vertices i and j . The vertices q and p are summed over the lattice. When replacing a line $P_{\sigma ij}^0$ in a skeleton diagram with a dressed line $\bar{P}_{\sigma ij}$, the skeleton diagram becomes an infinite sum of diagrams.

2 Analytic foundation

The first element of the sum is the skeleton diagram itself and the rest are non-skeleton one-particle-irreducible diagrams. The thick line in the left hand side of 2.94 represents the dressed line and the thin the regular lines,

$$\begin{array}{c} \text{Dressed line} \end{array} = \begin{array}{c} \text{Skeleton} \end{array} + \begin{array}{c} \text{Self-energy} \end{array} + \begin{array}{c} \text{Self-energy} \end{array} + \begin{array}{c} \text{Self-energy} \end{array} + \begin{array}{c} \text{Self-energy} \end{array} + \dots \quad (2.94)$$

By replacing all regular lines with dressed lines in all skeleton diagrams all diagrams of $S_{\sigma ij}^*$ are created. The signs of these diagrams are also correct, because the number of spin loops of such a diagram is the sum of the spin loops of the skeleton diagram plus the spin loops of the included self-energy diagrams. Furthermore, no diagram is generated twice. In conclusion the proper self-energy can be calculated by evaluating just the skeleton diagrams and replacing all $P_{\sigma ij}^0$ with $\bar{P}_{\sigma ij}$.

In order to illustrate the advantage of the Dyson equation and line dressing. Three tables follow, which each contains the diagrams up to the third order for the three different kinds of diagrams. The diagrams for the self-energy $S_{\sigma ij}$ up to order three are shown in the table 2 and there are a total of 17.

$n = 1$	
$n = 2$	
$n = 3$	

Figure 2: Self-energy diagrams up to order 3

The number of diagrams for the proper self-energy $S_{\sigma ij}^*$ up to order three is only eleven and are displayed in the next table 3.

2.3 Metzner and Vollhardt's diagrammatic expansion in powers of $g^2 - 1$

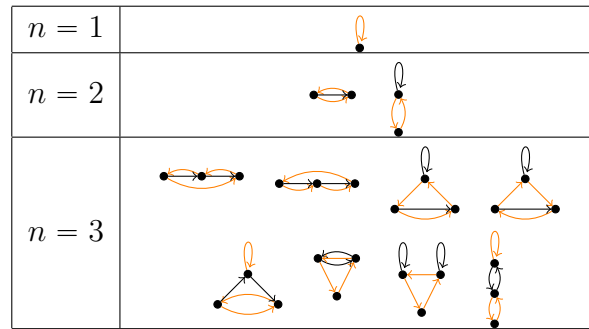


Figure 3: Proper self-energy diagrams up to order three

Finally, for the proper self-energy with dressed lines up to order three are only four. The lines are drawn thicker in table 4 to show that the lines are dressed lines.

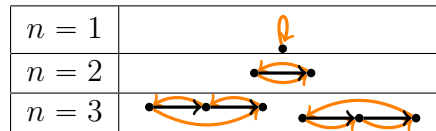


Figure 4: Proper self-energy diagrams with dressed lines up to order three

2.4 Gebhard's diagrammatic expansion

Gebhard reformulated the diagrammatic expansion in order to expand for high dimensions in $1/d$ [4]. He decomposed the double occupation operator \hat{D}_i into a single-particle Hartree-Fock term \hat{D}_i^{HF} and the remaining two-particle term $(\hat{D}_i - \hat{D}_i^{\text{HF}})$. The single-particle term is applied to the uncorrelated state $|\Psi_0\rangle$ and the diagrammatic expansion is in $(\hat{D}_i - \hat{D}_i^{\text{HF}})$ instead of \hat{D}_i . The resulting diagrammatic expansion has a topologically equivalent self-energy as in Metzner and Vollhardt's formalism, but every vertex i gives a factor x_i instead of $g^2 - 1$ and every line is $\tilde{P}_{\sigma ij}^0$ instead of $P_{\sigma ij}^0$. The general steps in this expansion are very similar, therefore this section is restricted to the basic idea of Gebhard's formalism, a short outline of the remaining steps and the results for the diagrammatic evaluation.

Decomposing the double occupation into its single-particle Hartree-Fock part \hat{D}_i^{HF} and the rest \tilde{D}_i is the main idea.

$$\hat{D}_i = \hat{D}_i^{\text{HF}} + \tilde{D}_i \quad (2.95)$$

$$\tilde{D}_i \equiv \hat{D}_i - \hat{D}_i^{\text{HF}} \quad (2.96)$$

The single-particle Hartree-Fock contribution is given by

$$\hat{D}_i^{\text{HF}} = \hat{n}_{i\sigma} \langle \hat{n}_{i\bar{\sigma}} \rangle_0 + \hat{n}_{i\bar{\sigma}} \langle \hat{n}_{i\sigma} \rangle_0 - \langle \hat{n}_{i\sigma} \rangle_0 \langle \hat{n}_{i\bar{\sigma}} \rangle_0 . \quad (2.97)$$

We will contract with Wick's theorem only the operators

$$\tilde{D}_i = \tilde{n}_{i\sigma} \tilde{n}_{i\bar{\sigma}} , \quad (2.98)$$

$$\tilde{n}_{i\sigma} \equiv \hat{n}_{i\sigma} - \langle \hat{n}_{i\sigma} \rangle_0 = \hat{c}_{i\sigma}^\dagger \hat{c}_{i\sigma} - \delta_{ii} \langle \hat{c}_{i\sigma}^\dagger \hat{c}_{i\sigma} \rangle_0 \quad (2.99)$$

in order to get only the propagators without on-site term

$$\tilde{P}_{\sigma ij}^0 \equiv \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} - \delta_{ij} \langle \hat{c}_{i\sigma}^\dagger \hat{c}_{i\sigma} \rangle_0 . \quad (2.100)$$

By doing this the lines in the diagrams are going to be $\tilde{P}_{\sigma ij}^0$ instead of $P_{\sigma ij}^0$. For getting there the expectation values of the GWF have to be rewritten in the operators \tilde{D}_i and $\tilde{n}_{i\sigma}$ instead of the natural operators \hat{D}_i and $\hat{n}_{i\sigma}$.

Previously we used the operator

$$g^{2\hat{D}} = \prod_i \left(1 + (g^2 - 1) \hat{D}_i \right) \quad \text{with} \quad \hat{D} = \sum_i \hat{D}_i , \quad (2.101)$$

which generated all the diagrams. Now we want to replace \hat{D}_i by \tilde{D}_i with the purpose to decrease the on-site contributions of the operator and use a new expansion variables x_i . Herefore we use the operator

$$g^{2\hat{K}} = \prod_i (1 + x_i \tilde{D}_i) . \quad (2.102)$$

The hermitian operator \hat{K} equals \hat{D} plus local single-particle operators with real variables, because these single-particle operators are easy to deal with. Explicitly,

$$\hat{K} = \sum_i \hat{K}_i , \quad (2.103)$$

$$\hat{K}_i = \hat{D}_i - \mu_{i\uparrow} \hat{n}_{i\uparrow} - \mu_{i\downarrow} \hat{n}_{i\downarrow} + \eta_i . \quad (2.104)$$

By equation 2.102 the real parameters x_i , $\mu_{i\uparrow}$, $\mu_{i\downarrow}$ and η_i are determined,

$$g^{2\hat{K}_i} = 1 + x_i \tilde{D}_i , \quad (2.105)$$

$$g^{2\eta_i} = 1 + x_i \langle \hat{n}_{i\uparrow} \rangle_0 \langle \hat{n}_{i\downarrow} \rangle_0 , \quad (2.106)$$

$$g^{2\eta_i - 2\mu_{i\uparrow}} = 1 + x_i \langle \hat{n}_{i\uparrow} \rangle_0 (\langle \hat{n}_{i\downarrow} \rangle_0 - 1) , \quad (2.107)$$

$$g^{2\eta_i - 2\mu_{i\downarrow}} = 1 + x_i \langle \hat{n}_{i\downarrow} \rangle_0 (\langle \hat{n}_{i\uparrow} \rangle_0 - 1) , \quad (2.108)$$

$$g^{2\eta_i - 2\mu_{i\uparrow} - 2\mu_{i\downarrow} - 2} = 1 + x_i (1 - \langle \hat{n}_{i\downarrow} \rangle_0 - \langle \hat{n}_{i\uparrow} \rangle_0 + \langle \hat{n}_{i\downarrow} \rangle_0 \langle \hat{n}_{i\uparrow} \rangle_0) . \quad (2.109)$$

Solving for the parameter x_i leads to

$$x_i = \frac{-1 + (1 - g^2)(n_{i0} - 2d_{i0}) + \sqrt{1 + (g^2 - 1)[n_{i0}(2 - n_{i0}) + g^2 m_{i0}]}}{2(1 - g^2)d_{i0}(1 - n_{i0} + d_{i0})} \quad (2.110)$$

with the use of the following abbreviations

$$n_{i0} = \langle \hat{n}_{i\uparrow} \rangle_0 + \langle \hat{n}_{i\downarrow} \rangle_0 , \quad (2.111)$$

$$d_{i0} = \langle \hat{n}_{i\uparrow} \rangle_0 \langle \hat{n}_{i\downarrow} \rangle_0 , \quad (2.112)$$

$$m_{i0} = \langle \hat{n}_{i\uparrow} \rangle_0 - \langle \hat{n}_{i\downarrow} \rangle_0 . \quad (2.113)$$

The single-particle part of \hat{K} is called \hat{S} ,

$$\hat{S} = \hat{K} - \hat{D} = \sum_i -\mu_{i\uparrow} \hat{n}_{i\uparrow} - \mu_{i\downarrow} \hat{n}_{i\downarrow} + \eta_i . \quad (2.114)$$

As before we want to calculate the term (for $i \neq j$)

$$\langle \Psi_g | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_g \rangle = \langle \Psi_0 | g^{\hat{D}} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} g^{\hat{D}} | \Psi_0 \rangle \quad (2.115)$$

$$= \langle \Psi_0 | g^{-\hat{S}} g^{\hat{K}_i + \hat{K}_j} \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} g^{\hat{K}_i + \hat{K}_j} \prod_{f \neq i, j} g^{2\hat{K}_f} g^{-\hat{S}} | \Psi_0 \rangle \quad (2.116)$$

$$= \langle \Phi_0 | g^{\hat{K}_i} \hat{c}_{i\sigma}^\dagger g^{\hat{K}_i} g^{\hat{K}_j} \hat{c}_{j\sigma} g^{\hat{K}_j} \prod_{f \neq i, j} (1 + x_f \tilde{D}_f) | \Phi_0 \rangle \quad (2.117)$$

But now we contract with the state

$$|\Phi_0\rangle = g^{-\hat{S}} |\Psi_0\rangle \quad (2.118)$$

2 Analytic foundation

and not with the state $|\Psi_0\rangle$. $|\Phi_0\rangle$ is still a product of single-particle wave functions, because $|\Psi_0\rangle$ is a Slater determinant and $g^{-\hat{S}}$ is just a single-particle operator. The expectation values with respect to the uncorrelated state are now with respect to $|\Phi_0\rangle$, including the ones in equation 2.97,

$$\langle \hat{A} \rangle_0 \equiv \frac{\langle \Phi_0 | \hat{A} | \Phi_0 \rangle}{\langle \Phi_0 | \Phi_0 \rangle}. \quad (2.119)$$

If $|\Psi_0\rangle$ is the Fermi sea, then $|\Phi_0\rangle$ is also the Fermi sea, because $g^{-\hat{S}}$ just gives a constant factor when applied to the Fermi sea, which cancels in the expectation values. However, in general the states $|\Phi_0\rangle$ and $|\Psi_0\rangle$ differ more than just by a factor. From now on we assume that $|\Phi_0\rangle$ is normalized. The operators

$$g^{\hat{K}_i} \hat{c}_{i\sigma}^\dagger g^{\hat{K}_i} = \hat{c}_{i\sigma}^\dagger \sqrt{q_{i\sigma}} [1 + x_i \alpha_{i\sigma} \tilde{n}_{i\bar{\sigma}}], \quad (2.120)$$

$$g^{\hat{K}_j} \hat{c}_{j\sigma} g^{\hat{K}_j} = \hat{c}_{j\sigma} \sqrt{q_{j\sigma}} [1 + x_j \alpha_{j\sigma} \tilde{n}_{j\bar{\sigma}}], \quad (2.121)$$

can be expressed with the operators $\tilde{n}_{i\bar{\sigma}}$ by using the abbreviations

$$\sqrt{q_{j\sigma}} = (1 + x_i (d_{i0} - \langle \hat{n}_{i\bar{\sigma}} \rangle_0)) [1 + \langle \hat{n}_{i\bar{\sigma}} \rangle_0 (g - 1 - g \frac{x_i \langle \hat{n}_{i\sigma} \rangle_0}{1 + x_i d_{i0}})], \quad (2.122)$$

$$\alpha_{i\sigma} = \frac{g - 1 - g \frac{x_i \langle \hat{n}_{i\sigma} \rangle_0}{1 + x_i d_{i0}}}{x_i [1 + \langle \hat{n}_{i\bar{\sigma}} \rangle_0 (g - 1 - g \frac{x_i \langle \hat{n}_{i\sigma} \rangle_0}{1 + x_i d_{i0}})]}. \quad (2.123)$$

Similarly the operators

$$g^{\hat{K}_i} \hat{n}_{i\sigma} g^{\hat{K}_i} = \langle \hat{n}_{i\sigma} \rangle_0 + \tilde{n}_{i\sigma} + x_i (1 - 2 \langle \hat{n}_{i\sigma} \rangle_0) \tilde{D}_i + x_i (1 - \langle \hat{n}_{i\sigma} \rangle_0) \langle \hat{n}_{i\sigma} \rangle_0 \tilde{n}_{i\bar{\sigma}}, \quad (2.124)$$

$$g^{\hat{K}_i} \hat{D}_i g^{\hat{K}_i} = [1 + x_i (1 - n_{i0} + d_{i0})] [d_{i0} + (1 - x_i d_{i0}) \tilde{D}_i + \sum_{\sigma} \langle \hat{n}_{i\sigma} \rangle_0 \tilde{n}_{i\sigma}], \quad (2.125)$$

can be expressed only with the operators $\tilde{n}_{i\sigma}$ and \tilde{D}_i . Now all remaining steps are similar as before. We have the general form

$$\langle \Phi_0 | \tilde{x}_{\sigma ij} \prod_{f \neq i, j} (1 + x_f \tilde{D}_f) | \Phi_0 \rangle = \sum_{m=0}^{\infty} \frac{1}{m!} \sum'_{f_1 \dots f_m \neq i, j} x_{f_1} \dots x_{f_m} \langle \Phi_0 | \tilde{x}_{\sigma ij} \tilde{D}_{f_1} \dots \tilde{D}_{f_m} | \Phi_0 \rangle. \quad (2.126)$$

The contraction $\langle \Phi_0 | \dots | \Phi_0 \rangle$ will be replaced by a new contraction $\{\dots\}_0$, which has the same value if any two operators $\tilde{n}_{i\sigma}$ are not identical, else the new contraction is zero,

$$\langle \Phi_0 | \tilde{x}_{\sigma ij} \prod_{f \neq i, j} (1 + x_f \tilde{D}_f) | \Phi_0 \rangle = \sum_{m=0}^{\infty} \frac{1}{m!} \sum_{f_1 \dots f_m} x_{f_1} \dots x_{f_m} \{\tilde{x}_{\sigma ij} \hat{D}_{f_1} \dots \hat{D}_{f_m}\}_0. \quad (2.127)$$

The disconnected diagrams from the numerators of the expectation values cancel with the denominator,

$$\frac{\langle \Phi_0 | \tilde{x}_{\sigma ij} \prod_{f \neq i,j} (1 + x_f \tilde{D}_f) | \Phi_0 \rangle}{\langle \Phi_0 | \prod_f (1 + x_f \tilde{D}_f) | \Phi_0 \rangle} = \sum_{m=0}^{\infty} \frac{1}{m!} \sum_{f_1 \dots f_m} x_{f_1} \dots x_{f_m} \{ \tilde{x}_{\sigma ij} \hat{D}_{f_1} \dots \hat{D}_{f_m} \}_0^c . \quad (2.128)$$

We define a self-energy

$$\tilde{S}_{\sigma ij} = -x_i \delta_{ij} \{ \tilde{n}_{i\bar{\sigma}} \tilde{X} \}_0^c + x_i x_j \{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \tilde{n}_{i\bar{\sigma}} \tilde{n}_{j\bar{\sigma}} \tilde{X} \}_0^c , \quad (2.129)$$

$$\tilde{X} = \sum_{m=0}^{\infty} \frac{1}{m!} \sum_{f_1 \dots f_m} x_{f_1} \dots x_{f_m} \hat{D}_{f_1} \dots \hat{D}_{f_m} . \quad (2.130)$$

The self-energy $\tilde{S}_{\sigma ij}$ has the same rules as the previous self-energy $S_{\sigma ij}$, except that every line stands for

$$\tilde{P}_{\sigma ij}^0 \equiv \{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \}_0 - \delta_{ij} \{ \hat{c}_{i\sigma}^\dagger \hat{c}_{i\sigma} \}_0 = \langle \Phi_0 | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Phi_0 \rangle - \delta_{ij} \langle \hat{n}_{i\sigma} \rangle_0 \quad (2.131)$$

and not for $P_{\sigma ij}^0 = \langle \Psi_0 | \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} | \Psi_0 \rangle$, and every vertex i gives a factor x_i instead of $g^2 - 1$. With diagrammatic arguments the terms

$$\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \tilde{X} \}_0^c = \tilde{P}_{\sigma ij}^0 + \sum_{q,p} \tilde{P}_{\sigma iq}^0 \tilde{S}_{\sigma qp} \tilde{P}_{\sigma pj}^0 \quad (i \neq j) , \quad (2.132)$$

$$\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \tilde{n}_{i\bar{\sigma}} \tilde{X} \}_0^c = \frac{-1}{x_i} \sum_p \tilde{S}_{\sigma ip} \tilde{P}_{\sigma pj}^0 \quad (i \neq j) , \quad (2.133)$$

$$\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \tilde{n}_{j\bar{\sigma}} \tilde{X} \}_0^c = \frac{-1}{x_j} \sum_q \tilde{P}_{\sigma iq}^0 \tilde{S}_{\sigma qj} \quad (i \neq j) , \quad (2.134)$$

$$\{ \tilde{D}_i \tilde{X} \}_0^c = \frac{-1}{x_i} \sum_q \tilde{P}_{\sigma iq}^0 \tilde{S}_{\sigma qi} , \quad (2.135)$$

can again be expressed by the self-energy $\tilde{S}_{\sigma ij}$. Directly from the definition of the self-energy we deduce

$$\{ \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} \tilde{n}_{i\bar{\sigma}} \tilde{n}_{j\bar{\sigma}} \tilde{X} \}_0^c = \frac{1}{x_i x_j} \tilde{S}_{\sigma ij} \quad (i \neq j) , \quad (2.136)$$

$$\{ \tilde{n}_{i\sigma} \tilde{X} \}_0^c = \frac{-1}{x_i} \tilde{S}_{\sigma ii} . \quad (2.137)$$

2 Analytic foundation

The expectation values as functions of the self-energy $\tilde{S}_{\sigma ij}$ are

$$P_{\sigma ij} = \sqrt{q_{i\sigma}}\sqrt{q_{j\sigma}} \left(\tilde{P}_{\sigma ij}^0 + \sum_{q,p} (\tilde{P}_{\sigma iq}^0 - \delta_{iq}\alpha_{i\sigma}) \tilde{S}_{\sigma qp} (\tilde{P}_{\sigma pj}^0 - \delta_{pj}\alpha_{j\sigma}) \right) \quad (i \neq j) , \quad (2.138)$$

$$P_{\sigma ii} = \langle \hat{n}_{i\sigma} \rangle_0 - \frac{1}{x_i} \tilde{S}_{\sigma ii} - (1 - \langle \hat{n}_{i\sigma} \rangle_0) \langle \hat{n}_{i\sigma} \rangle_0 \tilde{S}_{\sigma ii} - (1 - 2\langle \hat{n}_{i\sigma} \rangle_0) \sum_q \tilde{P}_{\sigma iq}^0 \tilde{S}_{\sigma qi} , \quad (2.139)$$

$$\langle \hat{D}_i \rangle_g = [1 + x_i(1 - n_{i0} + d_{i0})] [d_{i0} - \frac{1}{x_i} \sum_{\sigma} \langle \hat{n}_{i\sigma} \rangle_0 \tilde{S}_{i\sigma} - \frac{(1 - x_i d_{i0})}{x_i} \sum_q \tilde{P}_{\sigma iq}^0 \tilde{S}_{\sigma qi}] . \quad (2.140)$$

Analogously to the previous section the evaluation of the skeleton diagrams is enough if we use the Dyson equation and dressed lines. By using the Dyson equation the self-energy $\tilde{S}_{\sigma ij}$ gets stripped of one-particle reducible diagrams,

$$\tilde{S}_{\sigma ij} = \tilde{S}_{\sigma ij}^* + \sum_{q,p} \tilde{S}_{\sigma iq}^* \tilde{P}_{\sigma qp}^0 \tilde{S}_{\sigma pj} . \quad (2.141)$$

The proper self-energy $\tilde{S}_{\sigma ij}^*$ can again be calculated by evaluating only the skeleton self-energy diagrams and replacing the normal lines $\tilde{P}_{\sigma ij}^0$ with the dressed lines,

$$\tilde{P}_{\sigma ij} = \tilde{P}_{\sigma ij}^0 + \sum_{q,p} \tilde{P}_{\sigma iq}^0 \tilde{S}_{\sigma qp} \tilde{P}_{\sigma pj}^0 . \quad (2.142)$$

3 Numerical investigation

3.1 Generating skeleton diagrams

We want to calculate the proper self-energy in Metzner's and Gebhard's formalism up to high orders for the GWF, for that we need the skeleton diagrams up to these orders. These diagrams are generated by a Mathematica program, which was developed and tested by us during a "Projektarbeit" [8]. The procedure and the results of this "Projektarbeit" will be discussed in this section 3.1.

The initial uncorrelated states $|\Psi_0\rangle$ and $|\Phi_0\rangle$ are the paramagnetic Fermi sea of the tight binding Hamiltonian. Thus we have a translational invariant system, because the Gutzwiller correlator acts uniformly. Consequently the self-energies and their diagrams solely depend on distance between vertex i and j and not on the actual positions in the lattice. Hence the vertex i can be set to zero for the following calculations. Furthermore our system is invariant under inversion of the direction and has spin degeneracy. For that reasons the value of a diagram only depends on which vertices are connected by a line and not by the direction or the spin of the line. Hence the diagrams with a topologically equal connections between their vertices have the same value. Instead of evaluating each of these diagrams separately, we evaluate the value of just one diagram and multiply it with a weight. The weight is the sum of the signs of the topologically distinct connected diagrams.

$$\begin{array}{c} \text{Diagram 1} \end{array} + \begin{array}{c} \text{Diagram 2} \end{array} + \begin{array}{c} \text{Diagram 3} \end{array} = -3 \begin{array}{c} \text{Diagram 4} \end{array} \quad (3.1)$$

Above is the example of a fourth-order diagram. There are three topologically different diagrams, but with the same vertices connected in each diagram. They differ in the kind of their lines or in the direction of their lines. -3 is the weight of the diagram without directions and different spins, because the first diagram on the left hand side has three spin loops and the other two have one spin loop.

One possibility to compute all skeleton diagrams is to use the wick theorem and create all possible diagrams. Then sort all unwanted diagrams like disconnected and non-skeleton diagrams out and finally sort all diagrams together with the same values and sum its weight together. This method is relatively simple to implement, but produces lots of unwanted disconnected diagrams and that slows this method in high orders.

The way we computed the diagrams is by using a program written in Mathematica by Kastening et al. [9]. Their program generates the connected diagrams by construction from the lower order diagrams. This procedure creates only a few disconnected diagrams and therefore is faster for higher orders. However, their program generates

3 Numerical investigation

the diagrams for a φ^4 -field theory and not the diagrams for our problem. The φ^4 -field theory diagrams have only non-directed lines and one type of line. Thus the diagrams of Kastening et al. program look like the diagram on the right hand side of the equation 3.1.

The φ^4 -field theory diagrams correspond to the one-particle irreducible diagrams of the self-energy. Thus our task is now to write a program, that reads their output, then sorts out the non-skeleton diagrams, calculates the weights and finally returns them in an optimized form to evaluate them numerically.

The functionality of our program will be discussed in the following. We start with the question of how to read their output. The Kastening program writes its n -th order diagrams in lists of length $n + 1$, in each entry i of this list is a list of i integers. The integer on position j in the i -th list tells us how many lines run from vertex i to vertex j . The length of the list is increasing with their position, because we now deal with non-directed graphs. It is sufficient to connect a vertex just with the vertices with lower numbers. The first vertex of each Kastening diagram is an external vertex and have to be removed to obtain the correct self-energy diagram. The following example will make the notation clear.

$$\begin{array}{l}
 \{\{0\}, \\
 \{0, 0\}, \\
 \{0, 3, 0\}, \\
 \{0, 1, 1, 0\}, \\
 \{2, 0, 0, 2, 0\}\}
 \end{array}
 \longrightarrow
 \begin{array}{c}
 \begin{array}{ccccc}
 & 3 & & 4 & \\
 & \bullet & & \bullet & \\
 & \parallel & & \parallel & \\
 & \bullet & & \bullet & \\
 2 & & & & 5 \\
 & & & & \bullet \\
 & & & & 1
 \end{array}
 \longrightarrow
 \begin{array}{ccc}
 & f_2 & & f_3 & \\
 & \bullet & & \bullet & \\
 & \parallel & & \parallel & \\
 & \bullet & & \bullet & \\
 f_1 & & & & i
 \end{array}
 \end{array}
 \quad (3.2)$$

The third entry $\{0, 3, 0\}$ in this example 3.2 implies that three lines are running from vertex 3 to vertex 2, because it is the third entry. Then the fourth entry $\{0, 1, 1, 0\}$ implies one line from 4 to 2 and another line from 4 to 3. Finally the last entry $\{2, 0, 0, 2, 0\}$ stands for two lines from 5 to 1 and two lines from 5 to 4. The second arrow in 3.2 removes the external vertex and labels the vertices in the familiar way. In the code we remove the external vertex at the end, because then we can easily identify the vertices i and j over which we do not sum.

In our code we use two different representations of the diagrams, because we need to handle directed diagrams. The first one is the adjacency matrix, which is very similar to Kastening's representation. For a n -th order diagram the adjacency matrix is a $(n + 1) \times (n + 1)$ matrix and the matrix element in row i and column j is an

3.1 Generating skeleton diagrams

integer, that tells us how many lines run from vertex i to vertex j . The example 3.1 in an adjacency matrix form is

$$\begin{aligned} & \left\{ \begin{aligned} & \{0\}, \\ & \{0, 0\}, \\ & \{0, 3, 0\}, \\ & \{0, 1, 1, 0\}, \\ & \{2, 0, 0, 2, 0\} \end{aligned} \right\} \longrightarrow \begin{array}{c} \begin{array}{c} 3 \quad 4 \\ \bullet \quad \bullet \\ \left| \quad \left| \right. \\ \bullet \quad \bullet \\ 2 \quad 5 \\ \left| \quad \left| \right. \\ \bullet \quad \bullet \\ 1 \end{array} \end{array} \longrightarrow \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 \end{pmatrix}. \end{aligned} \quad (3.3)$$

The second representation is the edge list form. This is a list and each entry stands for one connection of the diagram. Each connection is represented by a list of two entries, the first entry is the outgoing vertex of the directed line and the second is the incoming vertex.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 \end{pmatrix} \longrightarrow \begin{array}{c} \begin{array}{c} 3 \quad 4 \\ \bullet \quad \bullet \\ \left| \quad \left| \right. \\ \bullet \quad \bullet \\ 2 \quad 5 \\ \left| \quad \left| \right. \\ \bullet \quad \bullet \\ 1 \end{array} \end{array} \longrightarrow \left\{ \begin{aligned} & \{3, 2\}, \quad \{3, 2\}, \quad \{3, 2\}, \\ & \{4, 2\}, \quad \{4, 3\}, \quad \{5, 1\}, \\ & \{5, 1\}, \quad \{5, 4\}, \quad \{5, 4\} \end{aligned} \right\} \quad (3.4)$$

Excluding the non-skeleton diagrams

The first stage of our program is to sort out the non-skeleton diagrams. In order to do that each diagram gets checked, whether it is a skeleton diagram or not. For that we represent the diagram in the edge list form and remove two entries from the edge list and then check whether the diagram is still connected or not. Removing two entries from the edge list is equivalent to cutting these two lines in the diagram. If we remove the last two entries in the edge list of example 3.4, then the diagram gets disconnected, so the diagram from example 3.4 is not a skeleton diagram.

$$\left\{ \begin{aligned} & \{3, 2\}, \quad \{3, 2\}, \quad \{3, 2\}, \\ & \{4, 2\}, \quad \{4, 3\}, \quad \{5, 1\}, \\ & \{5, 1\} \end{aligned} \right\} \longrightarrow \begin{array}{c} \begin{array}{c} 3 \quad 4 \\ \bullet \quad \bullet \\ \left| \quad \left| \right. \\ \bullet \quad \bullet \\ 2 \quad 5 \\ \left| \quad \left| \right. \\ \bullet \quad \bullet \\ 1 \end{array} \end{array} \quad (3.5)$$

For an arbitrary diagram with n connections we have to remove all possible pairs of connections and check, if they are still connected or not. Only if in all $n(n-1)$ cases the diagram is connected after removing each pair, then the diagram is truly a skeleton diagram. Whether a diagram is connected or not can easily be checked with the Mathematica package Combinatorica, because it contains instructions, which act on diagrams, and one of these build-in functions checks if a diagram is connected or

3 Numerical investigation

not. For example the surviving of the second-order skeleton diagram of this check is shown below:

$$\begin{aligned}
 & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{1, 2\}, \{2, 3\}, \{3, 1\}, \{2, 3\}, \{3, 2\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{3, 1\}, \{2, 3\}, \{3, 2\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{2, 3\}, \{2, 3\}, \{3, 2\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{2, 3\}, \{3, 1\}, \{3, 2\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{2, 3\}, \{3, 1\}, \{2, 3\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{1, 2\}, \{2, 3\}, \{3, 2\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{1, 2\}, \{3, 1\}, \{3, 2\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{1, 2\}, \{3, 1\}, \{2, 3\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{1, 2\}, \{2, 3\}, \{3, 2\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{1, 2\}, \{2, 3\}, \{2, 3\}\} \\
 \longrightarrow & \begin{array}{c} 2 \quad 3 \\ \curvearrowright \quad \curvearrowleft \\ \bullet \\ 1 \end{array} = \{\{1, 2\}, \{2, 3\}, \{3, 1\}\}
 \end{aligned} \tag{3.6}$$

Computing the diagram weights

Now we have the skeleton diagrams and the next step is to calculate the weight of each skeleton diagram. For that we generate all directed diagrams, sort out the not allowed and topologically equivalent. Next from each directed diagram we generate all diagrams with spin σ and spin $\bar{\sigma}$ lines and sort out again the not allowed and topologically equivalent diagrams.

In order to generate all directed diagrams, we take each undirected diagram and represent it in the edge list form. This edge list we interpret as the first directed diagram, the other directed diagrams we obtain by doing every possible combination

3.1 Generating skeleton diagrams

of line reversings. So if a diagram has n connections, we generate 2^n directed diagrams. Due to the rules of how to construct the self-energy (figure 1) each vertex in each directed diagram is only permitted to have two ingoing and two outgoing lines except for the vertex 1, which has only one ingoing and one outgoing line. So most of the 2^n direct diagrams are going to be dismissed. To check the amount of outgoing and incoming lines in a vertex of a diagram can easily be done in the adjacency matrix representation of the diagram. The sum of the elements in the i -th row is the number of outgoing lines of the vertex i and the sum of the elements in the j -th column is the number of incoming lines of vertex j . So forth a directed diagram is allowed if the sum of the first row is one, the sum of the first column is one and the sum of each other row or column is two. An example for a not allowed diagram is

$$\begin{array}{c}
 \begin{array}{ccccc}
 & 3 & & 4 & \\
 & \bullet & & \bullet & \\
 & \curvearrowright & & \curvearrowleft & \\
 & \bullet & & \bullet & \\
 & \curvearrowleft & & \curvearrowright & \\
 & 2 & & 5 & \\
 & \bullet & & \bullet & \\
 & \curvearrowright & & \curvearrowleft & \\
 & \bullet & & \bullet & \\
 & 1 & & &
 \end{array}
 & = &
 \begin{pmatrix}
 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 1 \\
 0 & 0 & 1 & 0 & 0 \\
 1 & 1 & 0 & 1 & 0
 \end{pmatrix},
 \end{array} \tag{3.7}$$

because the sum of the last row is three and not two. The next example is an allowed directed diagram

$$\begin{array}{c}
 \begin{array}{ccccc}
 & 3 & & 4 & \\
 & \bullet & & \bullet & \\
 & \curvearrowright & & \curvearrowleft & \\
 & \bullet & & \bullet & \\
 & \curvearrowleft & & \curvearrowright & \\
 & 2 & & 5 & \\
 & \bullet & & \bullet & \\
 & \curvearrowright & & \curvearrowleft & \\
 & \bullet & & \bullet & \\
 & 1 & & &
 \end{array}
 & = &
 \begin{pmatrix}
 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 1 \\
 0 & 0 & 1 & 0 & 1 \\
 1 & 1 & 0 & 0 & 0
 \end{pmatrix},
 \end{array} \tag{3.8}$$

in which the sum rule for the rows and columns is fulfilled. For the undirected diagram

$$\begin{array}{c}
 \begin{array}{ccc}
 & 3 & \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & \curvearrowleft & \\
 2 & & 4 \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & 1 &
 \end{array}
 & \tag{3.9}
 \end{array}$$

we find four allowed directed diagrams

$$\begin{array}{cccc}
 \begin{array}{ccc}
 & 3 & \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & \curvearrowleft & \\
 2 & & 4 \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & 1 &
 \end{array}
 & , &
 \begin{array}{ccc}
 & 3 & \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & \curvearrowleft & \\
 2 & & 4 \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & 1 &
 \end{array}
 & , &
 \begin{array}{ccc}
 & 3 & \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & \curvearrowleft & \\
 2 & & 4 \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & 1 &
 \end{array}
 & , &
 \begin{array}{ccc}
 & 3 & \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & \curvearrowleft & \\
 2 & & 4 \\
 & \bullet & \\
 & \curvearrowright & \\
 & \bullet & \\
 & 1 &
 \end{array}
 & .
 \end{array} \tag{3.10}$$

Of these four directed diagrams the first two diagrams are topologically equivalent and the last two are also topologically equivalent, because they become identical, when exchanging vertex 2 with vertex 4. Of these four diagrams we keep just two of them, one from the first two and one from the last two, because we only want the topologically distinct diagrams in the end.

3 Numerical investigation

Next we need to dismiss all the topologically equivalent diagrams except for one, so that we have exactly one diagram of each topological kind. For that we have to check whether a directed is topologically equivalent to one of the previous directed diagrams. If it is, we dismiss it, and else we keep it. If we already have m topologically distinct directed diagrams and we generate a new allowed diagram, we have to test this new diagram for topological equivalence with all previous m diagrams.

Two diagrams are topologically equivalent if it is possible to make the two diagrams identical by renaming the vertices. One way is to rename the vertices 1 to $(n + 1)$ of a n -th order diagram in all $(n + 1)!$ possible permutations and check for each permutation if the diagrams become identical. But it is more efficient to exploit the properties of our diagrams. The vertex 1 has only one incoming and one outgoing line and due to this it is distinguishable from the other vertices. So vertex 1 can be excluded from the permutation and the number of permutations decreases from $(n + 1)!$ to $n!$. From vertex 1 leaves one outgoing line, it ends in another vertex. By this, this vertex becomes also distinguishable from the other vertices. This vertex can be named 2. The same holds true for the other vertex, which is connected to 1, and can in general be named 3. The case when the two lines from vertex 1 connect with the same vertex is not important for us, because for the skeleton diagrams this happens only in the first order diagram and not in higher order diagrams.

Whether two diagrams are identical or not can easily be done with their adjacency matrices, because the adjacency matrix of a diagram is unique. The process of testing, if two diagrams are topologically equivalent is illustrated by an example. We have one fourth-order diagram and rename the vertices, which are connected to 1 with 2 and 3.

$$\begin{array}{c}
 \begin{array}{ccc}
 3 & & 4 \\
 \curvearrowright & & \curvearrowleft \\
 \downarrow & & \downarrow \\
 2 & & 5 \\
 \curvearrowleft & & \curvearrowright \\
 1 & &
 \end{array}
 \longrightarrow
 \begin{array}{ccc}
 5 & & 4 \\
 \curvearrowright & & \curvearrowleft \\
 \downarrow & & \downarrow \\
 2 & & 3 \\
 \curvearrowleft & & \curvearrowright \\
 1 & &
 \end{array}
 =
 \begin{pmatrix}
 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 \\
 0 & 0 & 1 & 1 & 0
 \end{pmatrix}
 \end{array}
 \quad (3.11)$$

Then we have a second diagram, which gets checked for topological equivalence with the first diagram.

$$\begin{array}{ccc}
 3 & & 4 \\
 \curvearrowright & & \curvearrowleft \\
 \downarrow & & \downarrow \\
 2 & & 5 \\
 \curvearrowleft & & \curvearrowright \\
 1 & &
 \end{array}
 \quad (3.12)$$

3.1 Generating skeleton diagrams

The vertices 1, 2 and 3 are fixed, but for the vertices 4 and 5 we need all two possible permutations.

$$\begin{array}{c} \begin{array}{ccccc} & 5 & & 4 & \\ & \bullet & & \bullet & \\ & \nearrow & & \nwarrow & \\ 2 & \bullet & & \bullet & 3 \\ & \nwarrow & & \nearrow & \\ & 1 & & & \end{array} \end{array} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}, \quad \begin{array}{c} \begin{array}{ccccc} & 4 & & 5 & \\ & \bullet & & \bullet & \\ & \nearrow & & \nwarrow & \\ 2 & \bullet & & \bullet & 3 \\ & \nwarrow & & \nearrow & \\ & 1 & & & \end{array} \end{array} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \quad (3.13)$$

The first permutation, in which 5 is the upper left vertex and 4 is the upper right vertex, is not identical to the diagram in 3.11, but the second permutation, in which 4 is top left and 5 top right, is identical to the given diagram in 3.11, because their adjacency matrices are identical.

Now we allocated all the allowed and topologically distinct directed diagrams, the next step is to generate from these diagrams the diagrams with spin σ and spin $\bar{\sigma}$ lines. Because it is easier to handle diagrams with one kind of line in the computer, we take one directed diagram and split it into two diagrams, one diagram with solely spin σ lines and the other with solely spin $\bar{\sigma}$ lines. The first diagram represents the σ lines of the diagram with two types of lines and the second diagram the $\bar{\sigma}$ lines. A diagram of order n has $(2n + 1)$ lines and $(n + 1)$ lines are σ lines and the remaining n lines are $\bar{\sigma}$ lines. Compared to the self-energy rules (figure 1) there are two additional σ lines in each diagram, which connect with the external vertex 1.

From the $(2n - 1)$ lines the two lines attached to vertex 1 and additional $(n - 1)$ lines go to the first diagram and the remaining n lines go to the second diagram. This means that from one directed diagram we create n out of $(2n - 1)$, i.e.,

$$\binom{2n - 1}{n} = \frac{(2n - 1)!}{(n - 1)!n!} \quad (3.14)$$

diagram pairs. The disallowed diagram pairs are excluded by checking their adjacency matrices. In each vertex of each line type only one line converges and one line emerges, so the sum over a row or column in one of the adjacency matrices is always one. If the sum over one row or one column is not one, the diagram pair is not allowed. An example of an allowed fourth order diagram pair is

$$\begin{array}{c} \begin{array}{ccccc} & 4 & & 5 & \\ & \bullet & & \bullet & \\ & \nearrow & & \nwarrow & \\ 2 & \bullet & & \bullet & 3 \\ & \nwarrow & & \nearrow & \\ & 1 & & & \end{array} \end{array} \longrightarrow \left\{ \begin{array}{c} \begin{array}{ccccc} & 4 & & 5 & \\ & \bullet & & \bullet & \\ & \nearrow & & \nwarrow & \\ 2 & \bullet & & \bullet & 3 \\ & \nwarrow & & \nearrow & \\ & 1 & & & \end{array} \\ \begin{array}{ccccc} & 4 & & 5 & \\ & \bullet & & \bullet & \\ & \nearrow & & \nwarrow & \\ 2 & \bullet & & \bullet & 3 \\ & \nwarrow & & \nearrow & \\ & 1 & & & \end{array} \end{array} \right\}. \quad (3.15)$$

3 Numerical investigation

Expressing the diagram pair in adjacency matrices:

$$\begin{array}{c}
 \begin{array}{ccc}
 4 & \leftarrow & 5 \\
 & \diagdown & / \\
 2 & \leftarrow & 3 \\
 & \diagup & \diagdown \\
 & 1 &
 \end{array}
 \end{array}
 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad
 \begin{array}{ccc}
 4 & \xrightarrow{\text{orange}} & 5 \\
 | & & | \\
 2 & \xrightarrow{\text{orange}} & 3
 \end{array}
 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (3.16)$$

The adjacency matrix of the $\bar{\sigma}$ diagram is smaller by one row and column than the adjacency matrix of the σ diagram, because the $\bar{\sigma}$ diagram does not contain the external vertex 1. These two adjacency matrixes are allowed, hence the sum over each row or column of each matrix is one.

We could proceed similarly for the directed diagrams and could sort out the topologically equivalent diagram pairs, but there exists a computationally cheaper way, because checking all diagram pairs for topological equivalence is computationally expensive. We calculate for each directed diagram how many topologically equivalent pairs appear and divide the contribution of this directed diagram to the weight of the undirected diagram by this number.

If we have two lines emerging from the same vertex and converging in the same vertex again, these lines are called parallel. For each parallel line pair in a directed diagram twice as many allowed diagram pairs are generated. If we have a parallel line in a directed diagram, one of these lines has to be a σ line and the other has to be a $\bar{\sigma}$ line. The computer generates both possibilities namely that the first line is a σ line and the second is the $\bar{\sigma}$ line and the reverse. Hence we need to count the number of parallel lines in every directed diagram and divide by two to the power number of parallel lines.

Counting the parallel lines is enough to count the number of identical diagrams, but we also need the number of topologically equivalent diagrams. Therefore we count how many permutations of a directed diagram are identical to each other. The procedure is analogous to checking two directed diagrams for topological equivalence, but now the diagram is going to be checked with itself. The symmetry factor is the number of times a permutation of the diagram itself is identical to itself. The vertices 1, 2 and 3 are determined as before and are excluded from the permutation procedure. In the following example we see the two possible permutations of a directed diagram and they are identical

$$\begin{array}{c}
 \begin{array}{ccc}
 4 & \xrightarrow{\text{orange}} & 5 \\
 | & & | \\
 2 & \xrightarrow{\text{orange}} & 3 \\
 & \diagup & \diagdown \\
 & 1 &
 \end{array}
 \end{array}
 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad
 \begin{array}{ccc}
 5 & \xrightarrow{\text{orange}} & 4 \\
 | & & | \\
 2 & \xrightarrow{\text{orange}} & 3 \\
 & \diagup & \diagdown \\
 & 1 &
 \end{array}
 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \quad (3.17)$$

Thus the symmetry factor for this directed diagram is two. Now we gathered all tools to calculate the weight of the undirected diagrams. The undirected diagrams and their weights up to order four are shown in table below (figure 5).







1	-1					
2	-1					
3	-2					
4	-3		-1		-1	

Figure 5: Undirected diagrams and their weights up to order four

Generating the arguments of the diagram values

After computing the weights of the undirected diagrams, we now proceed to calculating their values. In this section we do not perform the summation, rather we only write the argument of the summation in a proper way to use it later in a C program. In this C program we do the summation numerically up to a cutoff r_{\max} . Due to the rules of the self-energy $S_{\sigma ij}$ (figure 1) every line gives a factor $P_{\sigma'qp}^0$. We use the skeleton diagrams for the proper self-energy $S_{\sigma ij}^*$, every line corresponds to $\bar{P}_{\sigma'qp}$. For symmetry reasons we can drop the spin dependence of P^0 and \bar{P} . Additionally the system is translationally invariant, so the dependence of i and j reduces to the dependence on $(\mathbf{R}_i - \mathbf{R}_j)$. Here i is the lattice site index and \mathbf{R}_i is the lattice vector.

$$P_{\sigma'qp}^0 \longrightarrow P^0(\mathbf{R}_q - \mathbf{R}_p), \quad \bar{P}_{\sigma'qp} \longrightarrow \bar{P}(\mathbf{R}_q - \mathbf{R}_p) \quad (3.18)$$

Additionally our system is invariant under inversion of space, so

$$P^0(\mathbf{R}_q - \mathbf{R}_p) = P^0(\mathbf{R}_p - \mathbf{R}_q), \quad \bar{P}(\mathbf{R}_q - \mathbf{R}_p) = \bar{P}(\mathbf{R}_p - \mathbf{R}_q). \quad (3.19)$$

Hence a line running from vertex q to vertex p gives the same factor as a line running in the reverse way and it suffices to calculate the value from the undirected diagrams. We represent an undirected diagram in its edge list form and obtain from it the product of \bar{P} , like in the following example:

$$\begin{array}{c} 4 \\ 2 \\ 1 \end{array} \begin{array}{c} \circ \\ \circ \\ \circ \end{array} \begin{array}{c} 5 \\ 3 \end{array} \longrightarrow \{ \{1, 2\}, \{2, 4\}, \{2, 4\}, \{4, 5\}, \{4, 5\}, \{5, 3\}, \{5, 3\}, \{3, 1\}, \{3, 2\} \} \\ \longrightarrow (\bar{P}(\mathbf{R}_2 - \mathbf{R}_4))^2 (\bar{P}(\mathbf{R}_4 - \mathbf{R}_5))^2 (\bar{P}(\mathbf{R}_5 - \mathbf{R}_3))^2 \bar{P}(\mathbf{R}_3 - \mathbf{R}_2) \quad (3.20)$$

3 Numerical investigation

Beware that the lines attached to the external vertex 1 do not contribute factors \bar{P} , because the external vertex was only added to identify the vertices i and j more easily. Due to translational invariance \mathbf{R}_2 can be set to zero and then \mathbf{R}_3 is set to \mathbf{R} , where as \mathbf{R} is the lattice vector on which the proper self-energy $S^*(\mathbf{R})$ depends.

$$\begin{array}{c} 4 \quad 5 \\ \circ \quad \circ \\ \backslash \quad / \\ \circ \quad \circ \\ / \quad \backslash \\ \circ \quad \circ \\ R \end{array} \longrightarrow (\bar{P}(\mathbf{R}_4))^2 (\bar{P}(\mathbf{R}_4 - \mathbf{R}_5))^2 (\bar{P}(\mathbf{R}_5 - \mathbf{R}))^2 \bar{P}(\mathbf{R}) \quad (3.21)$$

The remaining $(n - 1)$ vertices of a n -th order diagram should be renamed. The first vertices is named \mathbf{R}_0 and not \mathbf{R}_1 , because then the notation is more similar to the notation used in the C program.

$$\begin{array}{c} 1 \\ \circ \\ \backslash \quad / \\ \circ \quad \circ \\ / \quad \backslash \\ \circ \quad \circ \\ R \end{array} \longrightarrow (\bar{P}(\mathbf{R}_0))^2 (\bar{P}(\mathbf{R}_0 - \mathbf{R}_1))^2 (\bar{P}(\mathbf{R}_1 - \mathbf{R}))^2 \bar{P}(\mathbf{R}) \quad (3.22)$$

Optimizing the arguments of the diagram values

Now we could start with the summation over $\mathbf{R}_0, \dots, \mathbf{R}_{n-1}$, but the numerical summation up to a cutoff r_{\max} is the computationally most expensive part of the program. So the argument of the summation has to be optimized, that the contribution outside of the cutoff r_{\max} is minimal. Our optimization tool is to choose a new basis of lattice vectors $\mathbf{r}_0, \dots, \mathbf{r}_{n-1}$ instead of the old basis $\mathbf{R}_0, \dots, \mathbf{R}_{n-1}$. In this new basis the argument of the summation should decay as fast as possible for any $|\mathbf{r}_m|$ greater then r_{\max} .

For the Fermi sea the functions $P^0(\mathbf{R})$ decays algebraically in $|\mathbf{R}|$. If we have a factor $P^0(\mathbf{r}_m)$ in the argument of the summation for any \mathbf{r}_m , then we have ensured that the argument of the summation decays algebraically for any $|\mathbf{r}_m|$. The advantage of the basis transformation from $\{\mathbf{R}_m\}$ to $\{\mathbf{r}_m\}$ becomes more clear by considering the sixth-order diagram

$$\begin{array}{c} 1 \quad 2 \\ \circ \quad \circ \\ \backslash \quad / \\ \circ \quad \circ \\ / \quad \backslash \\ \circ \quad \circ \\ 0 \quad 3 \\ \circ \quad \circ \\ R \end{array} \longrightarrow \frac{(\bar{P}(\mathbf{R}_0))^2 (\bar{P}(\mathbf{R}_0 - \mathbf{R}_1))^2 (\bar{P}(\mathbf{R}_1 - \mathbf{R}_2))^2}{(\bar{P}(\mathbf{R}_2 - \mathbf{R}_3))^2 (\bar{P}(\mathbf{R}_3 - \mathbf{R}))^2} \bar{P}(\mathbf{R}), \quad (3.23)$$

where we sum $\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2$ and \mathbf{R}_3 over the infinite lattice. On the one hand the summation for \mathbf{R}_0 and \mathbf{R}_3 can be cut off for $|\mathbf{R}_0| > r_{\max}$ and $|\mathbf{R}_3| > r_{\max} + |\mathbf{R}|$, hence the terms $(\bar{P}(\mathbf{R}_0))^2$ and $(\bar{P}(\mathbf{R}_3 - \mathbf{R}))^2$ get small for large \mathbf{R}_0 and \mathbf{R}_3 . On the other hand the summation over \mathbf{R}_1 and \mathbf{R}_2 can only be terminated if the factors $(\bar{P}(\mathbf{R}_1 - \mathbf{R}_2))^2$ and $(\bar{P}(\mathbf{R}_2 - \mathbf{R}_3))^2$ are small enough, which is the case for $|\mathbf{R}_1| > 2r_{\max} \geq r_{\max} + |\mathbf{R}_0|$ and $|\mathbf{R}_2| > 2r_{\max} + |\mathbf{R}| \geq r_{\max} + |\mathbf{R}_3| + |\mathbf{R}|$.

The way we determine the optimized basis $\{\mathbf{r}_m\}$ is to set the argument of one

\bar{P} equal to one of the \mathbf{r}_m . For the sixth-order diagram from before 3.23 the new basis is

$$\mathbf{r}_0 = \mathbf{R}_0 , \quad (3.24)$$

$$\mathbf{r}_1 = \mathbf{R}_0 - \mathbf{R}_1 , \quad (3.25)$$

$$\mathbf{r}_2 = \mathbf{R}_1 - \mathbf{R}_2 , \quad (3.26)$$

$$\mathbf{r}_3 = \mathbf{R}_2 - \mathbf{R}_3 \quad (3.27)$$

and the argument of the diagram becomes

$$\begin{array}{c} \text{1} \\ \circ \quad \circ \\ \text{0} \quad \text{2} \\ \circ \quad \circ \\ \text{3} \\ \circ \quad \circ \\ \text{R} \end{array} \longrightarrow \frac{(\bar{P}(\mathbf{r}_0))^2 (\bar{P}(\mathbf{r}_1))^2 (\bar{P}(\mathbf{r}_2))^2 (\bar{P}(\mathbf{r}_3))^2}{(\bar{P}(-\mathbf{r}_3 - \mathbf{r}_2 - \mathbf{r}_1 + \mathbf{r}_0 - \mathbf{R}))^2 \bar{P}(\mathbf{R})} . \quad (3.28)$$

Two additional aspects have to be considered, when we define a new \mathbf{r}_m . First it must be linearly independent of all the other $\mathbf{r}_{n \neq m}$ and second we get even faster convergence if the factor $\bar{P}(\mathbf{r}_m)$ occurs multiple times instead of once. Therefore the exact steps to determine the optimized basis $\{\mathbf{r}_m\}$ for a n -th order diagram are in the table of figure 6.

1. Write the undirected diagram in its edge list form and that way that each line goes from the vertex with lower number to the vertex with a higher number.
2. Sort the connections accordingly to how often they appear in the edge list. Connections that appear most often in the edge list are sorted to the beginning of the edge list and connections that appear once are sorted to the end of the edge list.
3. The difference of the vertices in the first connection becomes \mathbf{r}_m with $m = 0$.
4. The difference of the vertices in the next connection is set to \mathbf{r}_{m+1} .
5. If the vectors $\mathbf{r}_0 \dots \mathbf{r}_{m+1}$ are linearly independent, then raise m by one and else keep m constant.
6. Repeat steps four and five until m is equal to $n - 3$.

Figure 6: Steps to determine the new basis $\{\mathbf{r}_m\}$

Step two ensures that the potency of $\bar{P}(\mathbf{r}_m)$ are as high as possible and step five ensures that the basis $\{\mathbf{r}_m\}$ is linearly independent. Finally step six aborts the algorithm, when we have the $n - 2$ linear independent \mathbf{r}_m for the n -th order diagram. The idea of choosing a new basis in this way is influenced by the work of Gulacsi and Gulacsi [10].

3 Numerical investigation

Here is one final remark to the optimized basis $\{\mathbf{r}_m\}$. The best way to choose $\{\mathbf{r}_m\}$ depends on how the numerical summation is done. We choose to approximate the exact summation by summing each \mathbf{r}_m over $|\mathbf{r}_m| \leq r_{\max}$ and for this approximation our choice of $\{\mathbf{r}_m\}$ appears to be convenient.

3.2 Input data for the C programs

Next the actual output of the Mathematica program from the previous section 3.1 will be discussed for the one dimensional case up to order four. Translating the results from the "Projektarbeit" [8] into C code was part of this thesis. The algorithm produces a header file with the following content.

```

static inline double pow2(double x){return x*x;}

static inline double pow3(double x){return x*x*x;}

double order1(int R, int r[], double G[], int GMax)
{
    int R0 = (GMax-1)/2;
    return -G[R0];
}

double order2(int R, int r[], double G[], int GMax)
{
    int R0 = (GMax-1)/2;
    return -pow3(G[R+R0]);
}

double order3(int R, int r[], double G[], int GMax)
{
    int R0 = (GMax-1)/2;
    return -2*G[R+R0]*pow2(G[R+R0-r[0]])*pow2(G[R0+r[0]]);
}

double order4(int R, int r[], double G[], int GMax)
{
    int R0 = (GMax-1)/2;
    return -(G[R+R0]*G[R+R0-r[1]]*G[R+R0-r[0]-r[1]]*G[R0+r[1]]
            *G[R0+r[0]+r[1]]*pow2(G[R0+r[0]]))
            -G[R+R0+r[0]]*G[R+R0-r[1]]*G[-R+R0-r[0]+r[1]]
            *pow2(G[R0+r[0]])*pow2(G[R0+r[1]])
            -3*G[R+R0]*pow2(G[R0+r[0]])*pow2(G[R0+r[1]])
            *pow2(G[-R+R0-r[0]+r[1]]);
}

```

For order one to three this defines functions, that return the value of the diagram for a given lattice vector 'R' and a list of lattice vectors 'r'. In case of order four the function returns the sum of the three diagrams times their weights. The functions of higher orders, which are not shown here, also sum up the diagrams times their weights, this has the advantage of performing the lattice summation for all diagrams of one order at the same time instead of doing the summation for each diagram separately. The list 'G' in one dimension contains the values of $\bar{P}(R)$, in which

$$G[R + R0] = \bar{P}(R) . \quad (3.29)$$

3 Numerical investigation

The integer 'GMax' is the length of the list 'G' in one dimension. For two dimensions the list 'G' is a one dimensional object, but it has to incorporate the values for a two dimensional lattice. For the two dimensional lattice 'GMax' is the number of allowed x -values. The length of 'G' for two dimension is 'GMax²', because there are 'GMax' allowed x -values and as many y -values. For example the second-order function for two dimensions is

```
double order2(int R[2], int r[][2], double G[], int GMax)
{
    int R0 = (GMax*GMax-1)/2;
    return -pow3(G[R[0]*GMax+R[1]+R0]);
}
```

and the lattice integer 'R' and 'r[m]' from one dimension become integer lists of length two. The first entry is the x -value and the second is the y -value. The different \bar{P} are in the following way

$$G[x * GMax + y + R0] = \bar{P}(x, y) \quad (3.30)$$

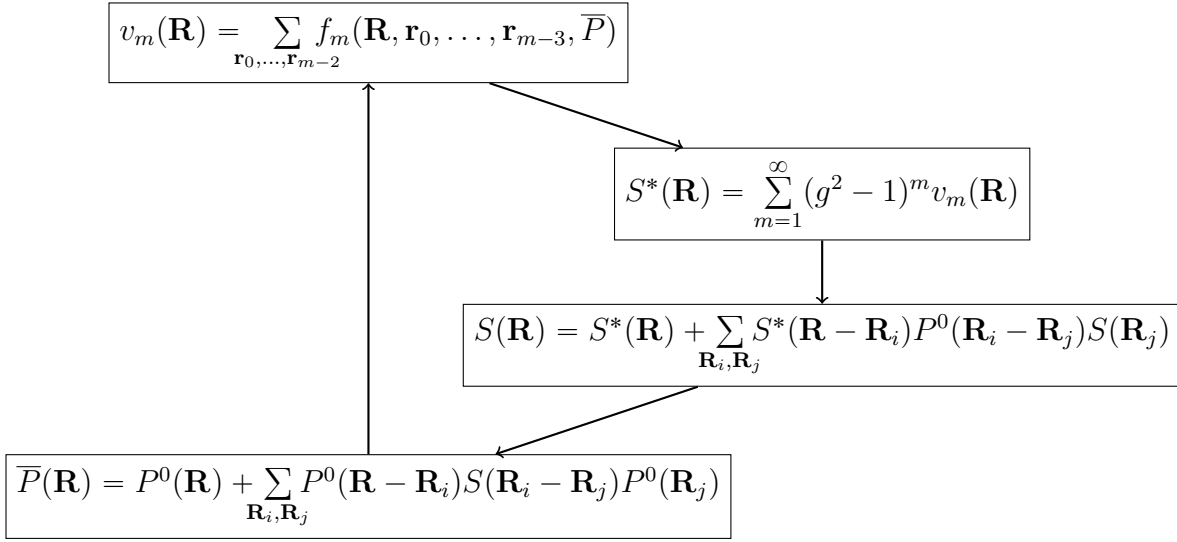
inserted in the list 'G'. For this thesis the functions up to order seven for one and two dimensions were generated. To generate the diagrams of higher orders our code described in previous section needs additional optimization.

The static inline definitions of 'pow2(x)' and 'pow3(x)' increase the performance of the program compared to the regular exponential function from the math package 'pow(x,y)', because the 'pow(x,y)' works for arbitrary doubles 'x' and 'y'.

3.3 General procedure for numerical approach

Our aim is to evaluate the GWF in high order diagrammatic expansions numerically in one and two dimensions. We have two different diagrammatic expansions the first by Metzner and Vollhardt [1][2][3] and the second by Gebhard [4]. These two diagrammatic expansions are similar and have the same structure. In this section the general procedure of evaluating these two expansions numerically will be discussed. Further we insert the Fermi sea as the uncorrelated state in both formalisms and get the formula, which get actually inserted in the algorithms.

The general idea is that we have a self-consistency loop for $\bar{P}(\mathbf{R})$, $S(\mathbf{R})$ and $S^*(\mathbf{R})$.



The step in the self-consistency loop displayed at the top is to sum the diagrams of order m times their weights over the lattice. The function f_m is the sum of the skeleton diagrams of order m times their weights. So f_m corresponds exactly to the function generated in the last chapter 3.2. The lines in the diagrams stand for a \bar{P} . The values $v_m(\mathbf{R})$ have to be evaluated for every natural number m and for every lattice vector \mathbf{R} . To evaluate the exact $v_m(\mathbf{R})$ we need $\bar{P}(\mathbf{R})$ for all lattice vectors. The next step in the self-consistency loop is to sum all the values $v_m(\mathbf{R})$ times $(g^2 - 1)^m$ up to gain the proper self-energy $S^*(\mathbf{R})$. The third step is to solve the Dyson equation. The Dyson equation is a linear system of equations for $S(\mathbf{R})$ and the dimension of this linear system of equation is the number of lattice vectors. The final step is to calculate the dressed line $\bar{P}(\mathbf{R})$ from the self-energy $S(\mathbf{R})$ by summing twice over the lattice. The $\bar{P}(\mathbf{R})$ again is inserted in the first step the diagram evaluation.

3 Numerical investigation

This set of equations is exact, but we need $\bar{P}(\mathbf{R})$, $S(\mathbf{R})$ or $S^*(\mathbf{R})$ to calculate the other quantities. Our approach is that we initially set

$$\bar{P}^{(0)}(\mathbf{R}) = P^0(\mathbf{R}) \quad (3.31)$$

and then go through the self-consistency loop n times until $\bar{P}(\mathbf{R})$, $S(\mathbf{R})$ and $S^*(\mathbf{R})$ converge, i.e.,

$$\bar{P}^{(n-1)}(\mathbf{R}) \approx \bar{P}^{(n)}(\mathbf{R}) , \quad (3.32)$$

$$S^{(n-1)}(\mathbf{R}) \approx S^{(n)}(\mathbf{R}) , \quad (3.33)$$

$$S^{*(n-1)}(\mathbf{R}) \approx S^{*(n)}(\mathbf{R}) . \quad (3.34)$$

The self-consistency loop we cannot iterate exactly because we have an infinite number of lattice sites and the computer cannot do infinite sums or solve a linear system of infinite equations numerically. So we introduce three different cutoff parameters m_{\max} , r_{\max} and R_{\max} . The first cutoff parameter m_{\max} determines the maximal order of the skeleton diagrams, so the sum for the proper self-energy over m from one to infinity will be shortened to one to m_{\max} . The program from the previous chapter 3.2 has generated f_m up to order seven, so the highest order for m_{\max} in this thesis is seven.

$$S^*(\mathbf{R}) \approx \sum_{m=1}^{m_{\max}} (g^2 - 1)^m v_m(\mathbf{R}) \quad (3.35)$$

The parameter r_{\max} restricts the summation of the diagram evaluation. For the one-dimensional lattice the summation parameters $\mathbf{r}_0, \dots, \mathbf{r}_{m-2}$ run each from $-r_{\max}$ to r_{\max} . For the two dimensional lattice the x -value and the y -value for each $\mathbf{r}_0, \dots, \mathbf{r}_{m-2}$ run from $-r_{\max}$ to r_{\max} . Why we choose this summation over a quadratic plane, and not over for example an area of a circle, will be discussed in the chapter about the program for the two-dimensional lattice.

$$v_m(\mathbf{R}) \approx \sum_{\mathbf{r}_0, \dots, \mathbf{r}_{m-2} = -r_{\max}}^{r_{\max}} f_m(\mathbf{R}, \mathbf{r}_0, \dots, \mathbf{r}_{m-2}, \bar{P}) \quad (3.36)$$

The last parameter R_{\max} restricts for which lattice vectors \mathbf{R} the diagrams are evaluated. For one dimension $v_m(R)$ will be calculated for $R = 0$ to $R = R_{\max}$, the values for $R = -R_{\max}$ to $R = -1$ do not need to be evaluated, due to the symmetry of the system.

$$v_m(R) = v_m(-R) \quad (3.37)$$

In two dimensions the x -value of \mathbf{R} runs from zero to R_{\max} for the evaluation of $v_m(\mathbf{R})$ and due to the symmetry of the two dimensional system it is enough to run the y -value of \mathbf{R} from zero to the x -value for each x -value.

$$v_m(x, y) = v_m(\max(|x|, |y|), \min(|x|, |y|)) \quad (3.38)$$

3.3 General procedure for numerical approach

The functions for the uncorrelated state $P^0(x, y)$ are also only calculated for $|x| \leq R_{\max}$ and $|y| \leq R_{\max}$.

The scheme presented above is formulated for Metzner and Vollhardt's formalism, but can be applied exactly identical to Gebhard's formalism, when we replace

$$P^0(\mathbf{R}) \longrightarrow \tilde{P}^0(\mathbf{R}) \quad (3.39)$$

$$S^*(\mathbf{R}) \longrightarrow \tilde{S}^*(\mathbf{R}) \quad (3.40)$$

$$S(\mathbf{R}) \longrightarrow \tilde{S}(\mathbf{R}) \quad (3.41)$$

$$\bar{P}(\mathbf{R}) \longrightarrow \tilde{P}(\mathbf{R}) \quad (3.42)$$

$$(g^2 - 1) \longrightarrow x_i \equiv x . \quad (3.43)$$

The function $P^0(\mathbf{R})$ is the Fourier transform of the occupation number of states in k -space for the uncorrelated state

$$\langle \hat{n}_{\mathbf{k}\sigma} \rangle_0 \equiv \langle \Psi_0 | \hat{c}_{\mathbf{k}\sigma}^\dagger \hat{c}_{\mathbf{k}\sigma} | \Psi_0 \rangle = \frac{1}{L} \sum_{\mathbf{R}} e^{-i\mathbf{k}\cdot\mathbf{R}} P^0(\mathbf{R}) = \begin{cases} 1 & \text{for } \epsilon_{\mathbf{k}} \leq \epsilon_F \\ 0 & \text{else} \end{cases} . \quad (3.44)$$

The discrete convolution of $P^0(\mathbf{R})$ with itself simplifies for the Fermi sea.

$$\sum_{\mathbf{R}_j} P^0(\mathbf{R} - \mathbf{R}_j) P^0(\mathbf{R}_j) = \frac{1}{L^2} \sum_{\mathbf{R}_j} \sum_{k_1, k_2} e^{i\mathbf{k}\cdot(\mathbf{R}-\mathbf{R}_j)} e^{i\mathbf{k}\cdot\mathbf{R}_j} \langle \hat{n}_{k_1\sigma} \rangle_0 \langle \hat{n}_{k_2\sigma} \rangle_0 \quad (3.45)$$

$$= \frac{1}{L} \sum_{k_1} e^{i\mathbf{k}\cdot\mathbf{R}} \langle \hat{n}_{k_1\sigma} \rangle_0 \langle \hat{n}_{k_1\sigma} \rangle_0 \quad (3.46)$$

$$= \frac{1}{L} \sum_{k_1} e^{i\mathbf{k}\cdot\mathbf{R}} \langle \hat{n}_{k_1\sigma} \rangle_0 = P^0(\mathbf{R}) \quad (3.47)$$

Using this analytic result one of the two summations in \bar{P} in Metzner's formalism can be evaluated analytically if we replace the summation of \mathbf{R}_j with the summation of $\mathbf{R}_l = \mathbf{R}_i - \mathbf{R}_j$.

$$\sum_{\mathbf{R}_i, \mathbf{R}_j} P^0(\mathbf{R} - \mathbf{R}_i) S(\mathbf{R}_i - \mathbf{R}_j) P^0(\mathbf{R}_j) = \sum_{\mathbf{R}_i} S(\mathbf{R}_l) \sum_{\mathbf{R}_i} P^0(\mathbf{R} - \mathbf{R}_i) P^0(\mathbf{R}_i - \mathbf{R}_l) \quad (3.48)$$

$$= \sum_{\mathbf{R}_i} S(\mathbf{R}_l) P^0(\mathbf{R} - \mathbf{R}_l) \quad (3.49)$$

3 Numerical investigation

\bar{P} and P (equation 2.81) in Metzner and Vollhardt's formalism simplify for the Fermi sea to:

$$\bar{P}(\mathbf{R}) = P^0(\mathbf{R}) + \sum_{\mathbf{R}_l} S(\mathbf{R}_l) P^0(\mathbf{R} - \mathbf{R}_l) \quad (3.50)$$

$$P(\mathbf{R}) = P^0(\mathbf{R}) - \frac{S(\mathbf{R})}{(g+1)^2} (1 - \delta_{\mathbf{R},0}) + \sum_{\mathbf{R}_l} S(\mathbf{R}_l) P^0(\mathbf{R} - \mathbf{R}_l) \left(1 - \frac{2}{g+1} - \frac{(g-1)\delta_{\mathbf{R},0}}{g+1}\right) \quad (3.51)$$

Also in Gebhard's formalism we can analytically do the same summations, because for the Fermi sea

$$\tilde{P}^0(\mathbf{R}) = P^0(\mathbf{R})(1 - \delta_{\mathbf{R},0}) = P^0(\mathbf{R}) - \frac{n}{2}\delta_{\mathbf{R},0} . \quad (3.52)$$

The parameter n is the filling of the Fermi sea and it varies from zero to two. Zero means that there are no electrons in the system and if n is equal to two, then every site is occupied twice with one up and one down electron. The dressed lines $\tilde{P}(\mathbf{R})$ in Gebhard's formalism become

$$\tilde{P}(\mathbf{R}) = \tilde{P}^0(\mathbf{R}) + \delta_{\mathbf{R},0} \frac{n}{2} \left(1 - \frac{n}{2}\right) \tilde{S}(\mathbf{R}) + (1-n) \sum_{\mathbf{R}_l} \tilde{S}(\mathbf{R}_l) \tilde{P}^0(\mathbf{R} - \mathbf{R}_l) . \quad (3.53)$$

Due to the translational invariance of the system and the symmetry of the spin the values

$$x_i \equiv x , \quad \sqrt{q_{i\sigma}} \equiv \sqrt{q} , \quad \alpha_i \equiv \alpha \quad (3.54)$$

become independent of the lattice site and their spin. These three values solely depend on the filling n and the Gutzwiller parameter g .

$$G = \sqrt{1 + (1-g^2)(n-2)n} \quad (3.55)$$

$$x = \frac{4(1-G)}{(n-2)n(1+G)} \quad (3.56)$$

$$\alpha = \frac{(1-n)(G+1)}{2(g+G)} \quad (3.57)$$

$$q = 2 \frac{1 + (1+g)n(n-2) - G}{(1+g)^2 n(n-2)} \quad (3.58)$$

3.3 General procedure for numerical approach

The expectation values (equation 2.138, 2.139, 2.140) in Gebhard's formalism then become

$$P(\mathbf{R} \neq 0) = q \left(\tilde{P}^0(\mathbf{R}) + (\alpha^2 + \frac{n}{2}(1 - \frac{n}{2}))\tilde{S}(\mathbf{R}) + (1 - n - 2\alpha) \sum_{\mathbf{R}_l} \tilde{S}(\mathbf{R}_l) \tilde{P}^0(\mathbf{R} - \mathbf{R}_l) \right), \quad (3.59)$$

$$P(\mathbf{R} = 0) = \frac{n}{2} - \tilde{S}(0) \left(\frac{1}{x} + \frac{n}{2}(1 - \frac{n}{2}) \right) - (1 - n) \sum_{\mathbf{R}_l} \tilde{S}(\mathbf{R}_l) \tilde{P}^0(\mathbf{R}_l), \quad (3.60)$$

$$\langle \hat{D} \rangle_g = \left(1 + x(1 - \frac{n}{2})^2 \right) \left(\frac{n^2}{4} - \frac{n}{x} \tilde{S}(0) - \frac{4 - xn^2}{4x} \sum_{\mathbf{R}_l} \tilde{S}(\mathbf{R}_l) \tilde{P}^0(\mathbf{R} - \mathbf{R}_l) \right). \quad (3.61)$$

For the GWF exists an analytical relation, which we later use to check the quality of the numerical results. In Metzner's paper [3] is the formula

$$n_\sigma^< = n_\sigma - \frac{1-g}{1+g} (n_\sigma n_{\bar{\sigma}} - \langle \hat{D} \rangle_g). \quad (3.62)$$

The value $n_\sigma^<$ is defined as

$$n_\sigma^< \equiv \frac{1}{L} \sum_k \langle \hat{n}_\sigma \rangle_g \langle \hat{n}_\sigma \rangle_0 = \sum_{\mathbf{R}} P(\mathbf{R}) P^0(\mathbf{R}). \quad (3.63)$$

Because our system is symmetric in the spin,

$$n_\sigma = \frac{n}{2}. \quad (3.64)$$

The relation therefore is

$$\sum_{\mathbf{R}} P(\mathbf{R}) P^0(\mathbf{R}) = \frac{n}{2} - \frac{1-g}{1+g} \left(\frac{n^2}{4} - \langle \hat{D} \rangle_g \right). \quad (3.65)$$

Using the identity

$$\frac{n}{2} = P^0(0) = \sum_{\mathbf{R}} P^0(\mathbf{R}) P^0(\mathbf{R}) \quad (3.66)$$

the relation can be rewritten as

$$0 = \left(\sum_{\mathbf{R}} P^0(\mathbf{R}) (P(\mathbf{R}) - P^0(\mathbf{R})) \right) + \frac{1-g}{1+g} \left(\frac{n^2}{4} - \langle \hat{D} \rangle_g \right). \quad (3.67)$$

As a measurement of the combined quality of $\langle \hat{D} \rangle_g$ and $P(\mathbf{R})$ we do the summation numerically up to the cutoff R_{\max} and normalize the result,

$$\text{err}_{\text{Check}} = \frac{\left| \left(\sum_{|\mathbf{R}| \leq R_{\max}} P^0(\mathbf{R}) (P(\mathbf{R}) - P^0(\mathbf{R})) \right) + \frac{1-g}{1+g} \left(\frac{n^2}{4} - \langle \hat{D} \rangle_g \right) \right|}{\frac{n}{2} + \frac{1-g}{1+g} \frac{n^2}{4}}. \quad (3.68)$$

3 Numerical investigation

If $\text{err}_{\text{Check}}$ of our simulation is of the magnitude of 1 or greater, then the results from this simulation are not trustworthy. If $\text{err}_{\text{Check}}$ is small and close to zero, then this is an indication that our results are quantitatively good.

3.4 Analytic solution for the one-dimensional chain

We need the analytic solution in one dimension to compare it to our numerical investigation in one dimension. The analytic solution was published first in [3], but the results presented here are from paper [11].

The analytic double occupation is

$$\langle \hat{D} \rangle_g = \frac{g^2}{2(1-g^2)^2} \left(-\ln(1 - (1-g^2)n) - (1-g^2)n \right) . \quad (3.69)$$

The density matrix $P_{\sigma ij} = P(R_i - R_j)$ is not directly accessible, but we calculate it by numerical Fourier transformation from $\langle \hat{n}_{k\sigma} \rangle_g$.

$$P(R) = \frac{1}{L} \sum_k \langle \hat{n}_{k\sigma} \rangle_g e^{-ikR} = \int_{-\frac{1}{2}}^{\frac{1}{2}} dk \langle \hat{n}_{k\sigma} \rangle_g e^{-i2\pi kR} \quad (3.70)$$

$$= 2 \int_0^{\frac{1}{2}} dk \langle \hat{n}_{k\sigma} \rangle_g \cos(2\pi kR) \quad (3.71)$$

In the paper [11] the one-dimensional Brillouin zone is normalized to have the length one. In the rest of this thesis except for this section the Brillouin zone has the volume $(2\pi)^d$ ($d = \text{dimension}$).

The occupation number $\langle \hat{n}_{k\sigma} \rangle_g$ has a discontinuity at the Fermi surface of the uncorrelated state.

$$\langle \hat{n}_{k\sigma} \rangle_g = \Theta(k_F - k) n_{k\sigma}^< + \Theta(k - k_F) n_{k\sigma}^> \quad \text{with } k_F = \frac{n}{4} \quad (3.72)$$

The function inside the Fermi surface is

$$n_{k\sigma}^< = 1 - \frac{1-g}{1+g} \frac{n}{2} + \frac{g^2}{(1+g)^2} \left(\mathcal{R}_0\left(\frac{4k}{n}, (1-g^2)n\right) - 1 \right) \quad (3.73)$$

and outside the Fermi surface the function is

$$n_{k\sigma}^> = -\frac{1-g}{1+g} \frac{n}{2} + \frac{1}{(1+g)^2} \left(-\frac{1}{2} \ln(1 - (1-g^2)n) + \mathcal{Q}_0\left(\frac{4k-2n}{n}, (1-g^2)n\right) \mathcal{Q}_0\left(\frac{4k-2n}{n}, (1-g^2)n\right) \right) . \quad (3.74)$$

The auxiliary function

$$\mathcal{R}_0 = \frac{4}{\pi \sqrt{(2-z)^2 - x^2 z^2}} K\left(\frac{z\sqrt{1-x^2}}{\sqrt{(2-z)^2 - x^2 z^2}}\right) \Theta(1-|x|) \quad (3.75)$$

3 Numerical investigation

contains the complete elliptic integral of the first kind

$$K(k) = \int_0^{\frac{\pi}{2}} d\phi \frac{1}{\sqrt{1 - k^2 \sin^2(\phi)}}. \quad (3.76)$$

The second auxiliary function $\mathcal{Q}_0(x, z)$ can be expressed in terms of $\mathcal{R}_0(x, z)$ and its integral.

$$\mathcal{Q}_0(x, z) = \left(\mathcal{W}_0(x, z) + \frac{z}{2} [(1-x)\mathcal{R}_0(x, z) + \mathcal{R}_1(x, z)] \right) \Theta(1 - |x|) \quad (3.77)$$

$$\mathcal{W}_0(x, z) = \frac{x-1}{2} \mathcal{R}_0(x, z) + \frac{z-2}{4} \mathcal{R}_1(x, z) + \frac{z(z-1)}{2} \frac{\partial \mathcal{R}_1(x, z)}{\partial z} \quad (3.78)$$

$$\mathcal{R}_1(x, z) = \int_1^x dy \mathcal{R}_0(y, z) \quad (3.79)$$

The numerical evaluation of these formulas we do with Mathematica, the complete elliptic integral of the first kind is available there as a function, which can be calculated with arbitrary numerical accuracy. The function $\mathcal{R}_1(x, z)$ and its derivative with respect to z we evaluate numerically as a function of x by doing the integration numerically for fixed $z = (1 - g^2)n$ and variable x ($-1 < x < 1$). Because we only integrate a well behaved one dimensional function, this numerical integration can be done by standard techniques with very high accuracy. The integration of the Fourier transformation we also do numerically, because the function \mathcal{Q}_0 we only access by a numerical approximation. This one-dimensional integral will also be calculated by standard techniques with high accuracy as long R is not too great. For large R ($R \gg 1$) the cos-function is highly oscillating and consequently the numerical integration becomes less accurate. The results from this evaluation can be considered as numerically exact.

3.5 Program for the one-dimensional chain

In this section the implementation for the one-dimensional lattice is presented as a test of our numerical approach. We can compare our results with the analytical solution and determine for given filling n and Gutzwiller parameter g , which cutoff parameters m_{\max} , r_{\max} and R_{\max} are necessary to produce proper results.

The function $P^0(\mathbf{R})$ in one dimension is extracted easily by Fourier transformation of $\langle \hat{n}_{k\sigma} \rangle_0$.

$$P^0(R) = \int_{-\pi}^{\pi} \frac{dk}{2\pi} \langle \hat{n}_{k\sigma} \rangle_0 e^{-ikR} = \int_{-k_F}^{k_F} \frac{dk}{2\pi} e^{-ikR} = \int_0^{k_F} \frac{dk}{\pi} \cos(kR) = \frac{\sin(Rk_F)}{\pi R} \quad (3.80)$$

The Fermi wave vector k_F is coupled to the filling n .

$$\frac{n}{2} = \int_{-\pi}^{\pi} \frac{dk}{2\pi} \langle \hat{n}_{k\sigma} \rangle_0 = P^0(R=0) = \frac{k_F}{\pi} \quad \Rightarrow \quad k_F = \frac{\pi n}{2} \quad (3.81)$$

In two dimensions $P^0(\mathbf{R})$ is a more complicated integral than in one dimension and cannot be computed easily for $|\mathbf{R}| \gg 1$. For that reason in two dimensions we use a cutoff for $P^0(\mathbf{R})$ and set

$$P^0(x, y) = 0 \quad \text{for} \quad \max(|x|, |y|) > R_{\max} . \quad (3.82)$$

In one dimension we also do this cutoff for $P^0(R)$ in order to transfer the estimated errors to two dimensions. Now we have all prerequisites to discuss the program in one dimension. Parts of the programs will be discussed in the following. We have two separate programs, one that calculates in Metzner and Vollhardt's formalism and the other in Gebhard's formalism. The input parameters of both programs are

$$g, n, R_{\max} \equiv \text{RMax}, r_{\max} \equiv \text{rMax}, m_{\max} \equiv \text{mMax} \text{ and } n_{\max} \equiv \text{nMax} . \quad (3.83)$$

The first five parameters are familiar from the previous chapter, the last parameter n_{\max} is the number of iterations. Thus the self-consistency cycle will be performed n_{\max} times. The two programs differ only in the definition of the intern functions, but the main function is identical, so it is convenient to discuss both together. The programs in two dimensions work the same way except some parts are bit more complicated due to the higher dimension.

The functions $S^*(R)$, $S(R)$ and $\bar{P}(R)$, which get calculated during the self-consistency loop, get each an array of length $2R_{\max} + 1$ to store their values from $-R_{\max}$ to R_{\max} .

$$S(R) \longrightarrow \text{array_S}[R + R0] , \quad (3.84)$$

$$S^*(R) \longrightarrow \text{array_S_star}[R + R0] , \quad (3.85)$$

$$\bar{P}(R) \longrightarrow \text{array_overline_P_new}[R + R0] \quad (3.86)$$

3 Numerical investigation

The values of function $S(R)$ are stored at position $R + R0$ of its array and the integer $R0 = R_{\max}$ points to the middle of the array. The array of $\overline{P}(R)$, which gets inserted in the beginning of the self-consistency loop, has to be longer than the other arrays. For example in the diagram of equation 3.28, we have the term

$$\overline{P}(-\mathbf{r}_3 - \mathbf{r}_2 - \mathbf{r}_1 + \mathbf{r}_0 - \mathbf{R}) . \quad (3.87)$$

The array

$$\overline{P}(R) \longrightarrow \text{array_overline_P_old}[R + R0\text{long}] \quad (3.88)$$

has to be long enough to contain the values, for all combinations of R in $-R_{\max}, \dots, R_{\max}$ and r_0, r_1, r_2 and r_3 in $-r_{\max}, \dots, r_{\max}$. So the array length have to be greater then $2(R_{\max} + 4r_{\max}) + 1$. We have chosen the length $2(R_{\max} + m_{\max}r_{\max}) + 1$ and $R0\text{long} = R_{\max} + m_{\max}r_{\max}$, which works fine for $m_{\max} = 7$ and several R_{\max} and r_{\max} .

The complete main program is printed in the appendix (page 79). It begins by including all packages, reading the input parameters and allocating space for the arrays. Especially we make an array called 'order'. This array contains the functions from section 3.2. The self-consistency loop is the heart of the program and is printed in the following:

```

int R;
for (R=-RMax; R<=RMax; R++){
    Pquer [R+R0long] = P0(R,n);
};
double X = x(g,n);

int iter;
for (iter=1; iter<=nMax; iter++){
    //setting arrays back to zero
    for (R=-RMax; R<=RMax; R++){
        array_S_star [R+R0] = 0.;
        array_S [R+R0] = 0.;
        array_overline_P_new [R+R0] = 0.;
    }

    //order mMax to 3
    int m;
    for (m=mMax; m>=3; m--){
        double *sumHelp = calloc (RMax+1, sizeof(double));
        double multHelp = pow(X, ((double) m) - 1.);
        {
            #pragma omp parallel for
            for (R=0; R<=RMax; R++){
                int r [m-2];
                recursive(&sumHelp [R], R, m-3, r, rMax,
                    array_overline_P_old, 2*R0long+1, order [m]);
            }
        }
    }
}

```

3.5 Program for the one-dimensional chain

```

    }
    for (R=0; R<=RMax; R++){
        array_S_star [R+R0] +=multHelp*sumHelp [R];
    }
}

//second order
int r [1];
if (mMax>=2){
    for (R=0; R<=RMax; R++){
        array_S_star [R+R0] += X*order [2](R, r, array_overline_P_old ,
            2*R0long+1);
    }
}

//first order
if (mMax>=1){
    array_S_star [0+R0] += order [1](0, r, array_overline_P_old ,
        2*R0long+1);
}

//make array_S_star symmetric
for (R=1; R<=RMax; R++){
    array_S_star [-R+R0] = array_S_star [R+R0];
};

//calc array_S from array_S_star by Dyson equation
calcSfromSstar (array_S_star, RMax, g, n, array\_S);

//calc array_overline_P_new from array_S
calcOverlinePFromS (array_S, RMax, g, n, array_overline_P_new);

//for the next iteration set array_overline_P_old
//equal to array_overline_P_new
for (R=-RMax; R<=RMax; R++){
    array_overline_P_old [R+R0long] = array_overline_P_new [R+R0];
}
}

```

The first step is to initialize 'array_overline_P_old' with 'P0(R,n)'. 'P0(R,n)' is $P^0(R)$ in Metzner and Vollhardt's formalism and $\tilde{P}^0(R)$ in Gebhard's formalism. The function 'x(g,n)' is $g^2 - 1$ in the first formalism and $x(g, n)$ from equation 3.56 in the second formalism.

Before calculating the arrays for S^* , S and \bar{P} we clear the arrays and then start the self-consistency loop with the evaluation of the diagrams. We first evaluate the highest-order diagrams and then work our way down to the first order, because the contribution of the diagrams decreases with increasing order, and one should

3 Numerical investigation

always start with the smallest terms and end with the bigger terms in a numerical summation to decrease rounding errors. The next optimization in calculating S^* is that we calculate $\frac{S^*}{X}$ instead of S^* . In the equations for $\langle \hat{D} \rangle_g$ and $P(R)$ appears the term $\frac{S}{X}$ and if we divide S by a small 'X', we get huge rounding errors. In order to calculate $\frac{S}{X}$ with the Dyson equation we initially need $\frac{S^*}{X}$.

Evaluating the diagrams

The first and second order diagrams get evaluated separately from the higher orders, because there no summation over the lattice is necessary. The first order diagram has a factor $\delta_{0,R}$ and thus this diagram contributes only to $S^*(0)$. More interesting is the evaluation of the diagrams of order three and higher. We have to evaluate the diagrams for R in $\{0, \dots, R_{\max}\}$. In order to speed up this code block we parallelize this for-loop with the expression "#pragma omp parallel for". So for different R the evaluation of the diagrams is distributed to different cores or CPUs of the system, if available. We choose to parallelize explicitly only this code block, because it is the computationally most expensive part and the parallelization on this level can be done very easily. Instead the parallelization could alternatively be done in the summation, which does the function "recursive(...)".

The function "recursive(...)" does the following summation

$$\text{sumHelp}[R] = \sum_{r[0]=-r_{\max}}^{r_{\max}} \dots \sum_{r[m-3]=-r_{\max}}^{r_{\max}} \text{order}[m](R, r[0], \dots, r[m-3]),$$

array_overline_P_old) (3.89)

as proposed in equation 3.36. Such nested summation for 'm-2' summation variables can be programmed by 'm-2' nested for-loops. The issue here is that the number 'm-3' is a variable and not fixed. So the program must be enabled to nest for-loops by itself and this is done here by the function "recursive(...)". The function "recursive(...)" makes a for-loop and in each iteration of the for-loop the function calls itself, but counts the integer 'n' down by one and writes in the array entry 'r[n]' a lattice position. The number 'n' must be in the first call 'm-3', then the function "recursive(...)" calls itself recursively until 'n' is equal to zero and finally there are 'm-2' nested for-loops and the summation is executed. Next is the exact code of "recursive(...)" displayed.

```
void recursive( double *returnValue ,
                int R,
                int n,
                int r [] ,
                int rMax,
                double array_overline_P [] ,
                int GMax,
```


3.5 Program for the one-dimensional chain

```

ORDER_FUNC orderfunc
)
//n = length of the array r[] = m-3
{
  if (n>0) {
    int i;
    for (i = -rMax; i <= rMax; i++){
      r[n] = i;
      recursive(returnValue, R, n - 1, r, rMax,
                array_overline_P, GMax, orderfunc);
    }
  }
  else{
    int i;
    for (i = -rMax; i <= rMax; i++){
      r[n] = i;
      *returnValue = *returnValue
                    + orderfunc(R, r, array_overline_P,
                                GMax);
    }
  }
}

```

Solving the Dyson equation

The consecutive step in the self-consistency loop is to calculate $S(R)$ from $S^*(R)$ by the Dyson equation. The Dyson equation reads

$$S(R) = S^*(R) + \sum_{R_i, R_j} S^*(R_i) P^0(R_i - R_j) S(R_j) . \quad (3.90)$$

We rewrite it as a system of linear equation for $\frac{S(R)}{X}$ and cut the summations for $|R_i|, |R_j| > R_{\max}$.

$$\sum_{R_j = -R_{\max}}^{R_{\max}} \left[\delta_{R, R_j} - X \left(\sum_{R_i = -R_{\max}}^{R_{\max}} \frac{S^*(R_i)}{X} P^0(R_i - R_j) \right) \right] \frac{S(R_j)}{X} = \frac{S^*(R)}{X} \quad (3.91)$$

$S(R)$ and $S^*(R)$ are invariant under inversion of space, so the linear system in the upper equation can be reduced from $2R_{\max} + 1$ for $R \in \{-R_{\max}, \dots, R_{\max}\}$ equations to $R_{\max} + 1$ equation for $R \in \{0, \dots, R_{\max}\}$.

$$\sum_{R_j=0}^{R_{\max}} M_{R, R_j} \frac{S(R_j)}{X} = \frac{S^*(R)}{X} \quad (3.92)$$

The matrix M with the entries M_{R, R_j} is

$$M = \sum_{R=0}^{R_{\max}} \mathbf{e}_R \mathbf{e}_R^T - \sum_{R' = -R_{\max}}^{R_{\max}} P^H(R - R') \mathbf{e}_R \mathbf{e}_{|R'|}^T . \quad (3.93)$$

3 Numerical investigation

by the use of the abbreviation

$$P^H(R) = X \sum_{R'=-R_{\max}}^{R_{\max}} \frac{S^*(R')}{X} P^0(R' - R) \quad (3.94)$$

and the factor $\mathbf{e}_R \mathbf{e}_{R'}^T$, which is the $(R_{\max} + 1) \times (R_{\max} + 1)$ matrix with every entry equal to zero except for the entry in row $R + 1$ and column $R' + 1$ and that is one. There exist several packages to solve linear equations, we use the LAPACK package for solving the linear equation.

After solving the Dyson equation we calculate $\bar{P}(R)$ by equation 3.51 in Metzner's formalism or by 3.53 in Gebhard's Formalism. Finally the new values for $\bar{P}(R)$ are copied from the new array to the old array and the self-consistency cycle is repeated n_{\max} times.

Results

In the previous part the functionality of the programs in one dimension was examined and now their results will be discussed. We have three different quantities to analyze, the double occupation $\langle \hat{D} \rangle_g$, the density matrix $P(R)$ and $\text{err}_{\text{Check}}$ from section 3.3. For the first two quantities we are especially interested in their deviation from the analytic solution to determine the quality of our simulation. For the double occupation the error is defined as

$$\text{err}_{\langle \hat{D} \rangle_g} = \frac{|\langle \hat{D} \rangle_g^{\text{analytic}} - \langle \hat{D} \rangle_g^{\text{numeric}}|}{|\langle \hat{D} \rangle_g^{\text{analytic}}|}. \quad (3.95)$$

Analogously the deviation for the density matrix is

$$\text{err}_P = \frac{\sum_{R=0}^{R_{\max}} |P(R)^{\text{analytic}} - P(R)^{\text{numeric}}|}{\sum_{R=0}^{R_{\max}} |P(R)^{\text{analytic}}|}. \quad (3.96)$$

In figure 7 are the results of Metzner and Vollhardt's formalism shown for variable number of iterations n_{\max} and the remaining parameters are fixed. First of all we are interested in the convergence speed of the self-consistency cycle. All graphs in figure 7 are converged after the fourth iteration. So choosing the parameter n_{\max} in the range of five to ten is sufficiently large to ensure convergence for these given parameters.

3.5 Program for the one-dimensional chain

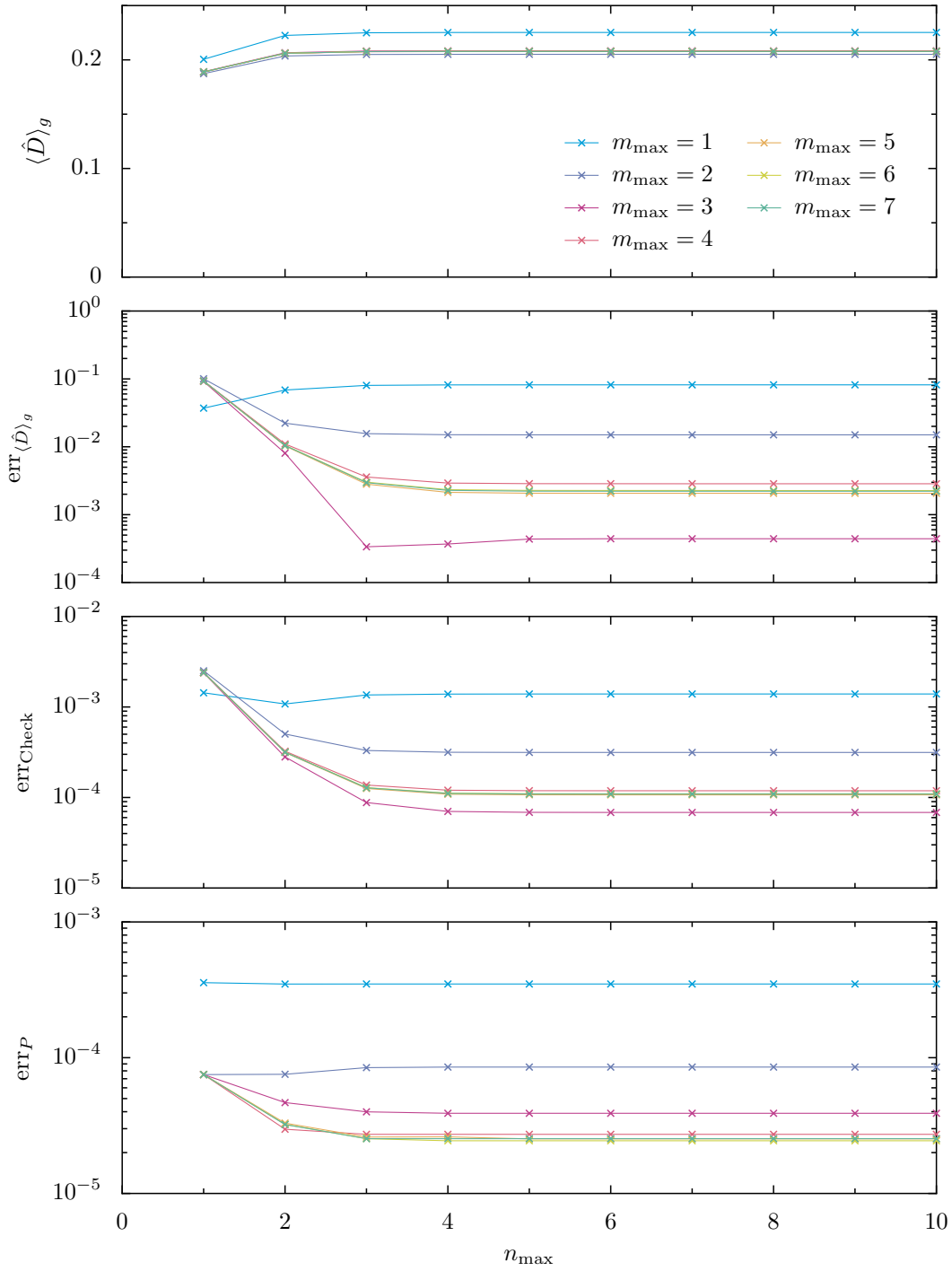


Figure 7: Metzner and Vollhardt's formalism for variable number of iterations n_{max} in one dimension at fixed $g = 0.9$, $n = 0.95$, $R_{\text{max}} = 10$ and $r_{\text{max}} = 10$

3 Numerical investigation

Next we are interested in the trends of the errors. As expected all error graphs decrease to a plateau. From order one to three the errors decrease and this is good, because we expected a lowering of the error by using diagrams up to higher orders. For order four and higher this is not the case, because the errors for order four to seven are nearly identical and for $\text{err}_{\langle \hat{D} \rangle_g}$ and $\text{err}_{\text{Check}}$ the error even increases compared to the third order. In conclusion it is not reasonable to use higher order diagrams than order three in this numerical approach for the formalism of Metzner and Vollhardt.

Next follows the discussion of our results with Gebhard's formalism. In figure 8 are the corresponding graphs for this formalism. Similarly the convergence of the self-consistency cycle is reached after the fifth iteration. We investigated the convergence rate for various parameters and especially g closer to zero and after five to six iterations the convergence was always reached. Figure 9 confirms this claim by presenting the convergence for g equal to 0.3. For the further numerical investigation we set the parameter n_{max} equal to ten to ensure convergence for each point in each following plot.

The errors for double occupation and density matrix in general decrease with the increase of the used order in figures 8 and 9. This is the behavior we expected and we can utilize this fact in the investigation of two dimensions by constructing an error estimator. But the error does not decrease evenly from order one to order seven, instead it decreases significantly from an odd order to the next higher even order and only slightly from an even order to the following odd order. This anomaly is observed in all plots for Gebhard's formalism of the one-dimensional lattice. In some plots the errors for the even order are even smaller than the error of the next higher odd order. Thus, it is advising to perform the calculations up to an even order to reach high accuracy.

3.5 Program for the one-dimensional chain

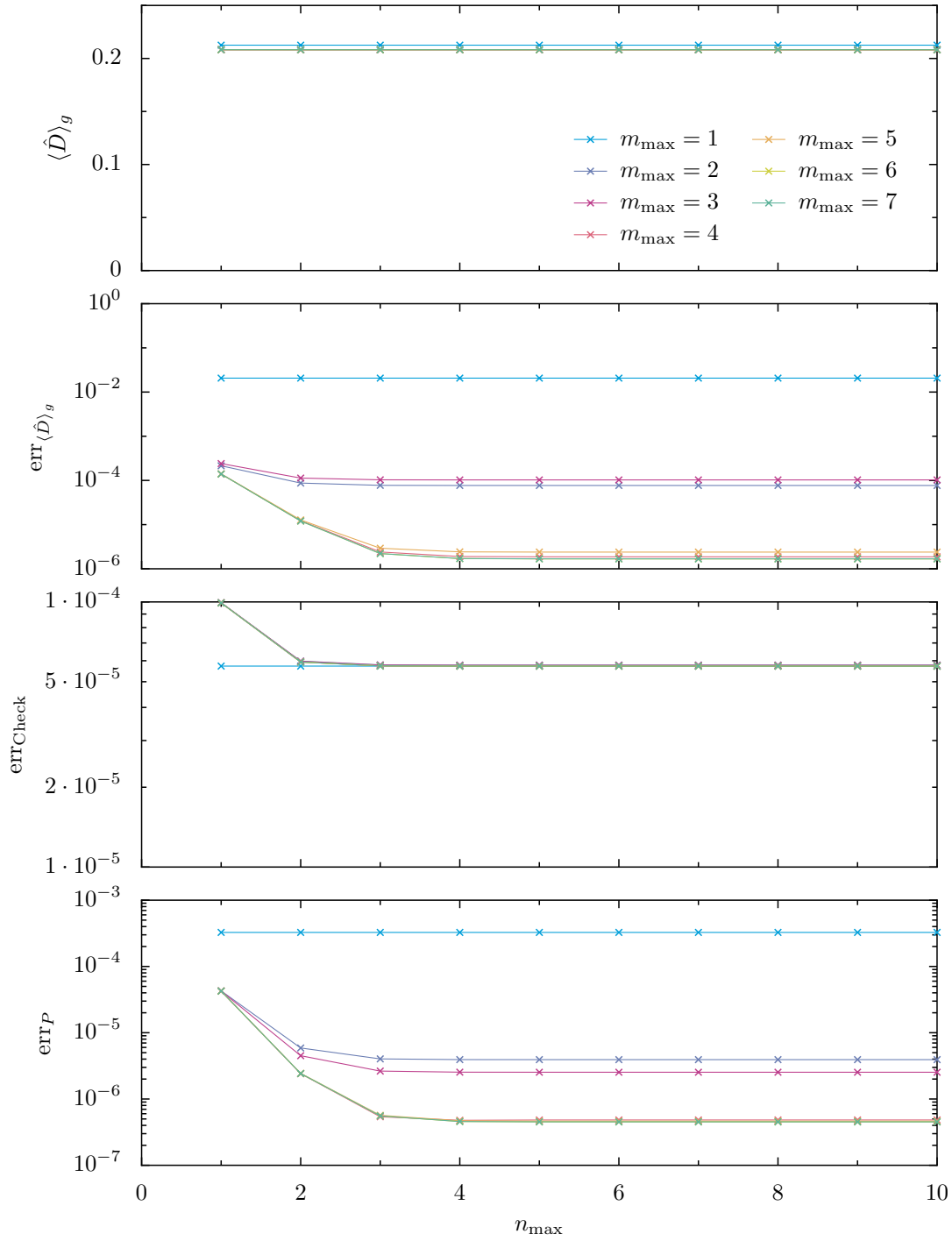


Figure 8: Gebhard's formalism for variable number of iterations n_{max} in one dimension at fixed $g = 0.9$, $n = 0.95$, $R_{\text{max}} = 10$ and $r_{\text{max}} = 10$

3 Numerical investigation

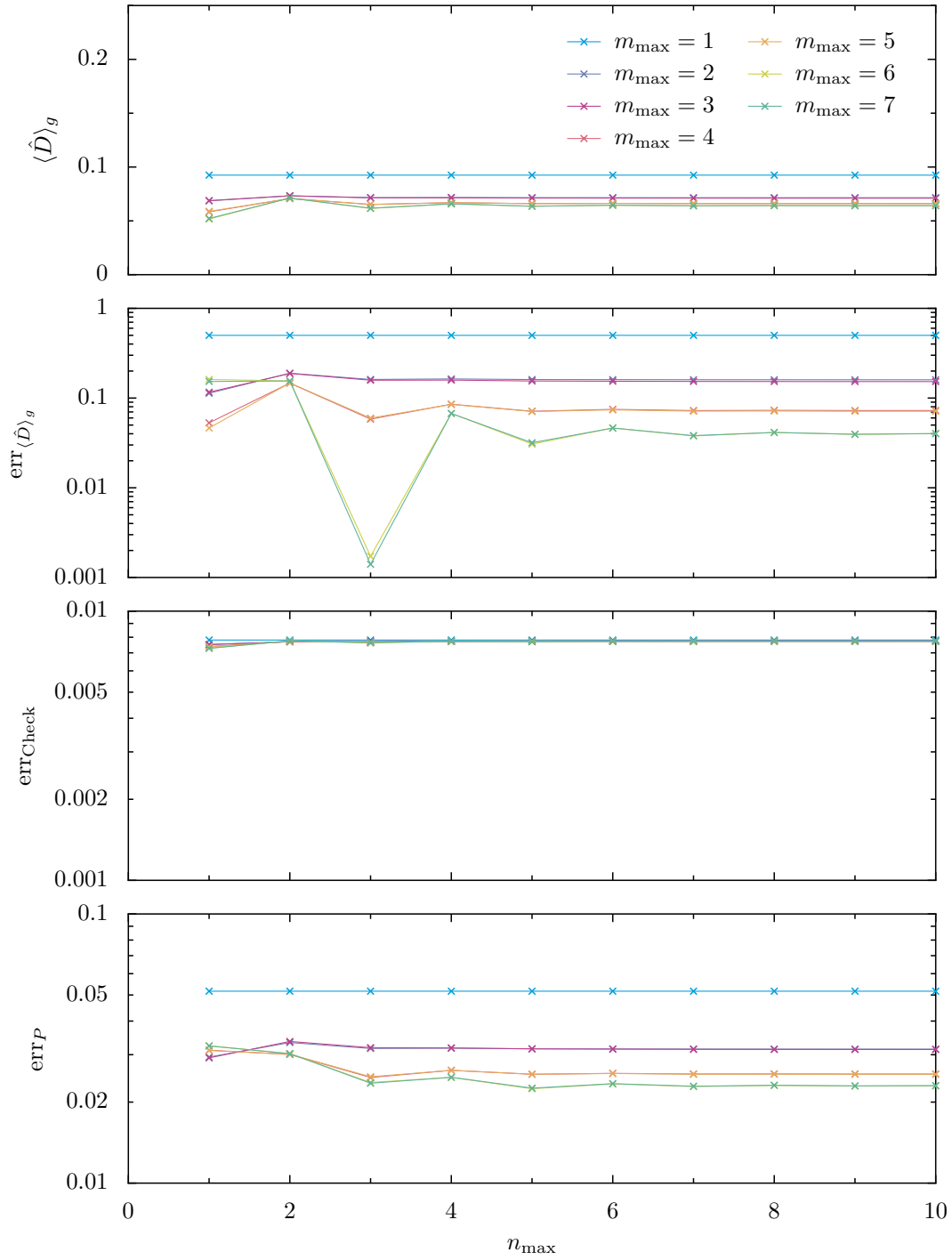


Figure 9: Gebhard's formalism for variable number of iterations n_{max} in one dimension at fixed $g = 0.3$, $n = 0.95$, $R_{\text{max}} = 10$ and $r_{\text{max}} = 10$

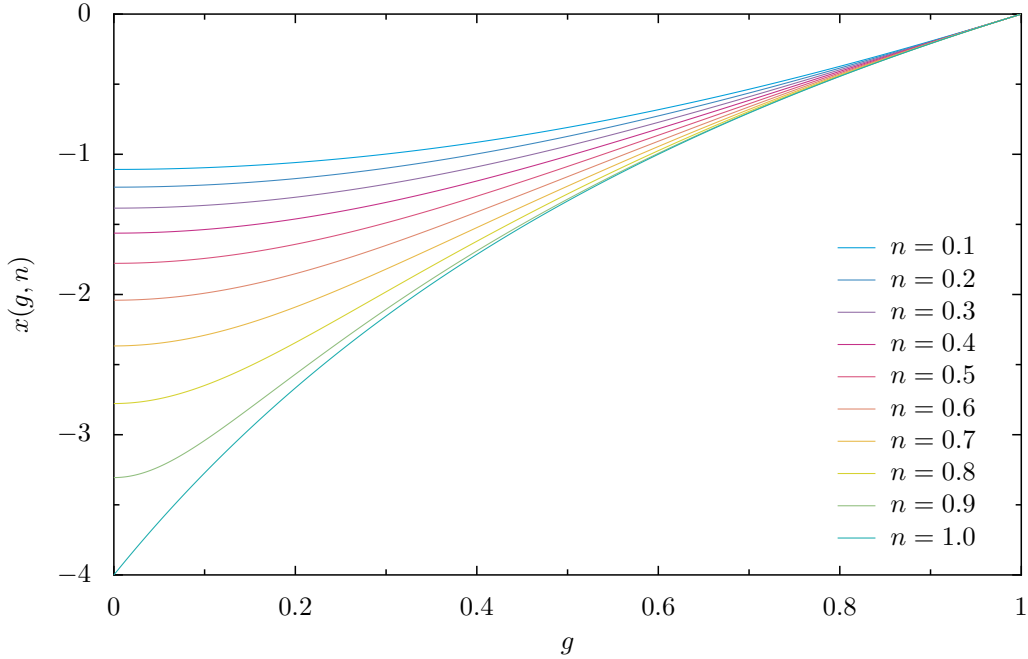


Figure 10: The expansion parameter $x(g, n)$ in Gebhard's formalism

In our discussion we focus on Gebhard's formalism, due to the fact that it achieves clearly higher accuracy than the other formalism. For example the errors in figure 8 are significantly smaller than those in figure 7 and this discrepancy increases further for smaller g . When we compare both expansion parameters $g^2 - 1$ and $x(g, n)$, this behavior is remarkable, because the Gutzwiller parameter g takes values between zero and one, thus $g^2 - 1$ takes values between minus one and zero. On the contrary $x(g, n)$ takes values between minus four and zero, the behavior of $x(g, n)$ is displayed in figure 10. An explanation for the advantage of Gebhard's formalism over the other one is that Gebhard's expansion starts from the Gutzwiller approximation, which is a much closer to the exact solution than the uncorrelated state.

In contrast to the errors of double occupation and density matrix the error $\text{err}_{\text{Check}}$ is nearly constant for all orders. The zeroth order in Gebhard's formalism is the Gutzwiller approximation and for the Gutzwiller approximation $\text{err}_{\text{Check}}$ is exactly zero in the limit of $R_{\text{max}} \rightarrow \infty$. The first order diagram in Gebhard's formalism has no contribution to the self-energy, when we use the non-dressed lines. Due to the fact that we start the self-consistency cycle with the non-dressed lines, the calculations with $m_{\text{max}} = 1$ are identical to the Gutzwiller approximation. The deviation from zero in the $\text{err}_{\text{Check}}$ plots for $m_{\text{max}} = 1$ are caused solely by the cutoff in the summation of equation 3.68. So even if the calculations with higher order diagrams are closer to the analytic solution, we cannot expect a smaller deviation

3 Numerical investigation

from zero in the $\text{err}_{\text{Check}}$ diagrams than as the first order. Thus it is a positive signal, that in our approach the lines for different orders lie on top of each other in the $\text{err}_{\text{Check}}$ plots.

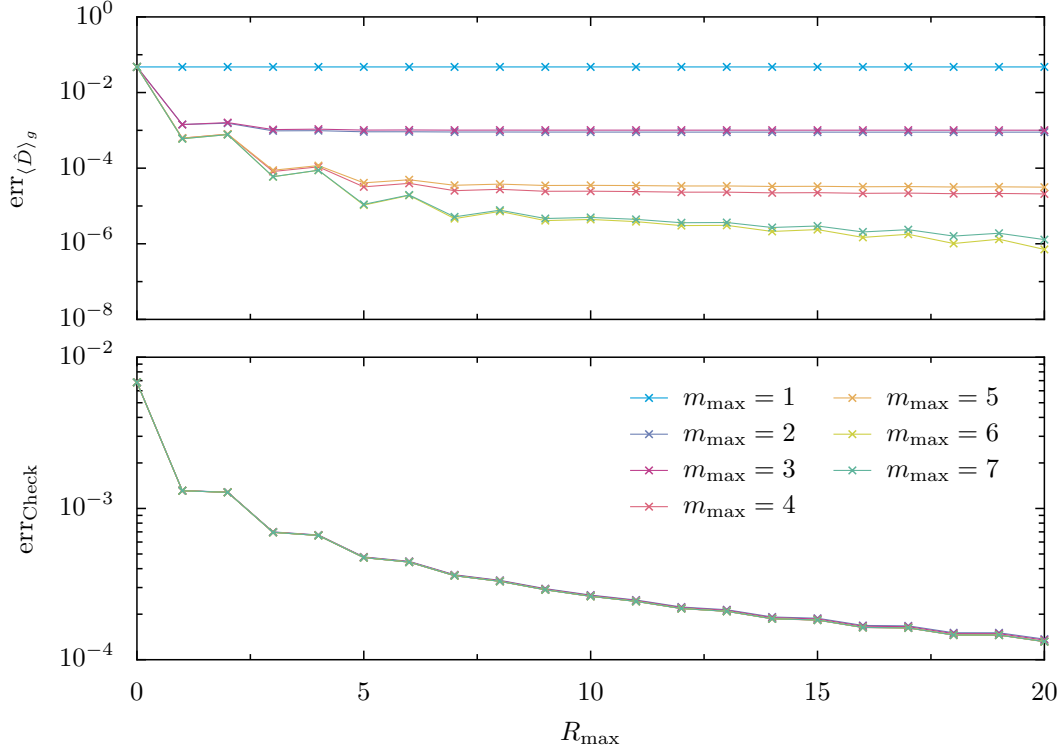


Figure 11: Gebhard's formalism for variable R_{max} at fixed $g = 0.8$, $n = 0.95$, $r_{\text{max}} = 10$ and $n_{\text{max}} = 10$

The dependence of $\text{err}_{\text{Check}}$ on the cutoff parameter R_{max} is explicitly shown in figure 11. There $\text{err}_{\text{Check}}$ decreases rapidly for small R_{max} and later on it still decreases steadily for increasing R_{max} . The error of the double occupation shows also a declining behavior for rising R_{max} . But the different orders decline each to a minimal error at a certain R_{max} . The minimal error decreases for increasing order, but also the R_{max} to reach each minimum increases with the order. We conclude that higher order diagrams unfold their full advantage, only if R_{max} is sufficiently large.

After the variation of R_{max} we examine the impact of the parameter r_{max} on our results. The error of P is affected very little by $r_{\text{max}} \geq 2$, but the error of $\langle \hat{D} \rangle_g$ is more influenced by it as seen in figure 12. For the diagrams of the first two orders happens no summation, so these two orders cannot be affected by the variation of r_{max} . But also for the third order r_{max} seems to have nearly no impact. For the four higher orders are these two errors generally sinking for rising r_{max} , but for r_{max}

3.5 Program for the one-dimensional chain

equal to three or five the error of $\langle \hat{D} \rangle_g$ gets especially small for the sixth and seventh order. For $r_{\max} \geq 6$ the error in $\langle \hat{D} \rangle_g$ also reaches a fix point. In conclusion it is not necessary to pick r_{\max} greater than six for this set of parameters, because the error will only decrease by an insignificant amount.

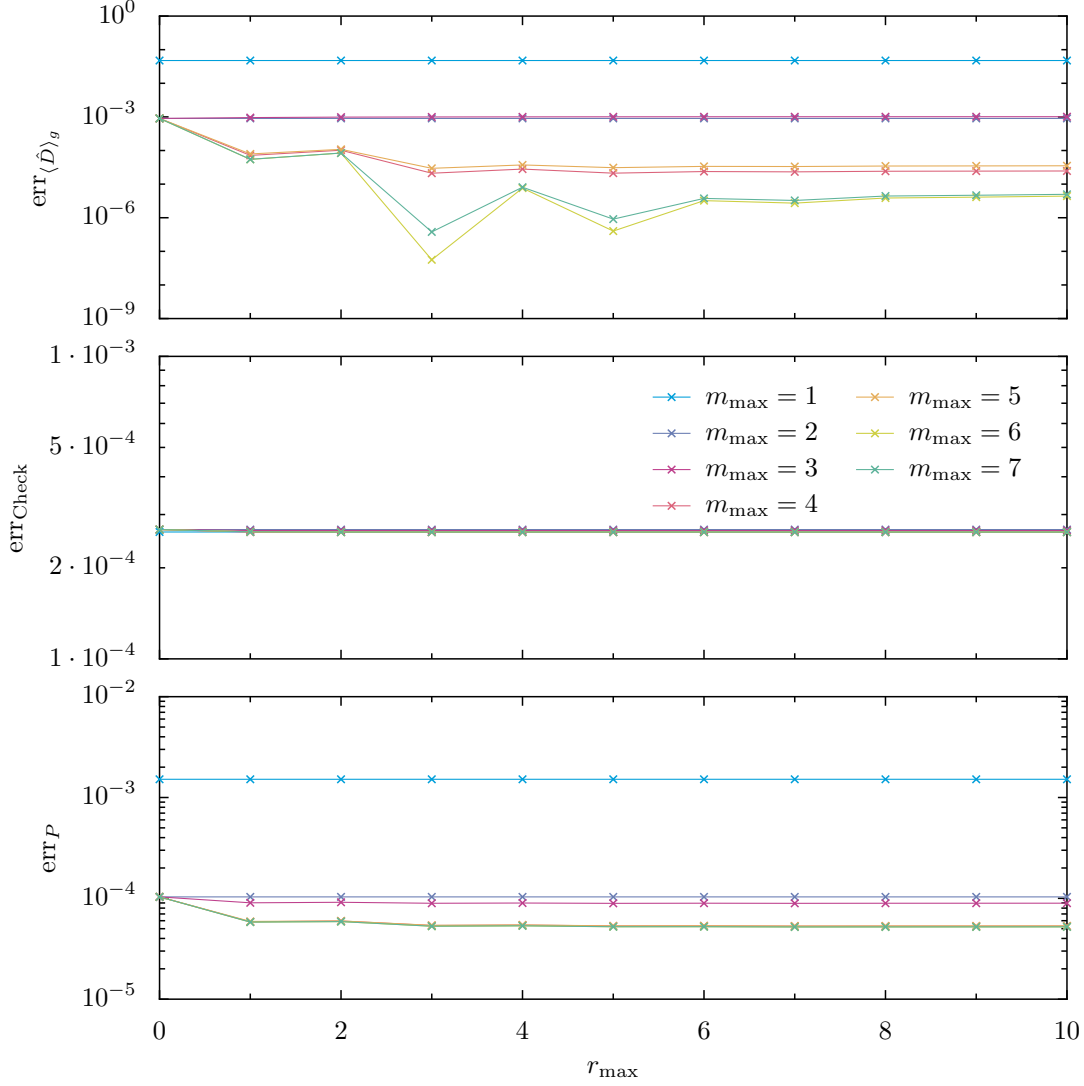


Figure 12: Gebhard's formalism for variable r_{\max} for $g = 0.8$, $n = 0.95$, $R_{\max} = 10$ and $n_{\max} = 10$

We are most interested in the quality of our approach for variable g and n near half filling. The double occupation for variable g for several fillings n is plotted in figure 13. There is a significant distance between the points of $m_{\max} = 1$ and the analytic solution except for g very close to one. The values for $m_{\max} = 2$ and higher orders are quite close to the analytic curve for n not too close to one or g greater then one

3 Numerical investigation

half. Only in the region of n close to one ($n = 0.99$ or $n = 0.95$) and g smaller than one half exists a significant discrepancy between analytic solution and our numerical results. For most values of g increasing the used order decreases the error significantly.

In figure 14 the errors for $n = 0.95$ and variable g are shown in detail. All errors are falling through several powers of ten, when g goes from zero to one. For double occupation and density matrix the errors decrease from order one to seven and only in the region of g smaller than 0.2 occurs some crossing of the lines. For $\text{err}_{\text{Check}}$ all lines lie again on top of each other, the only exception is again the region of g smaller than 0.2. There the lines of the graph diverge slightly from each other. In conclusion our approach works fine with Gebhard's formalism as long g is not too close to zero ($g < 0.2$) and additionally the filling is close to one. If the filling is lower than 0.8 the code produces proper results for all g .

For calculations at exactly half filling we advise to do this limit analytically at the mathematical level, because most equations simplify in this limit. Additionally there is an analytic solution available for $P^0(x, y)$ at half filling. Including this analytic solution speeds up the code and will increase the accuracy.

3.5 Program for the one-dimensional chain

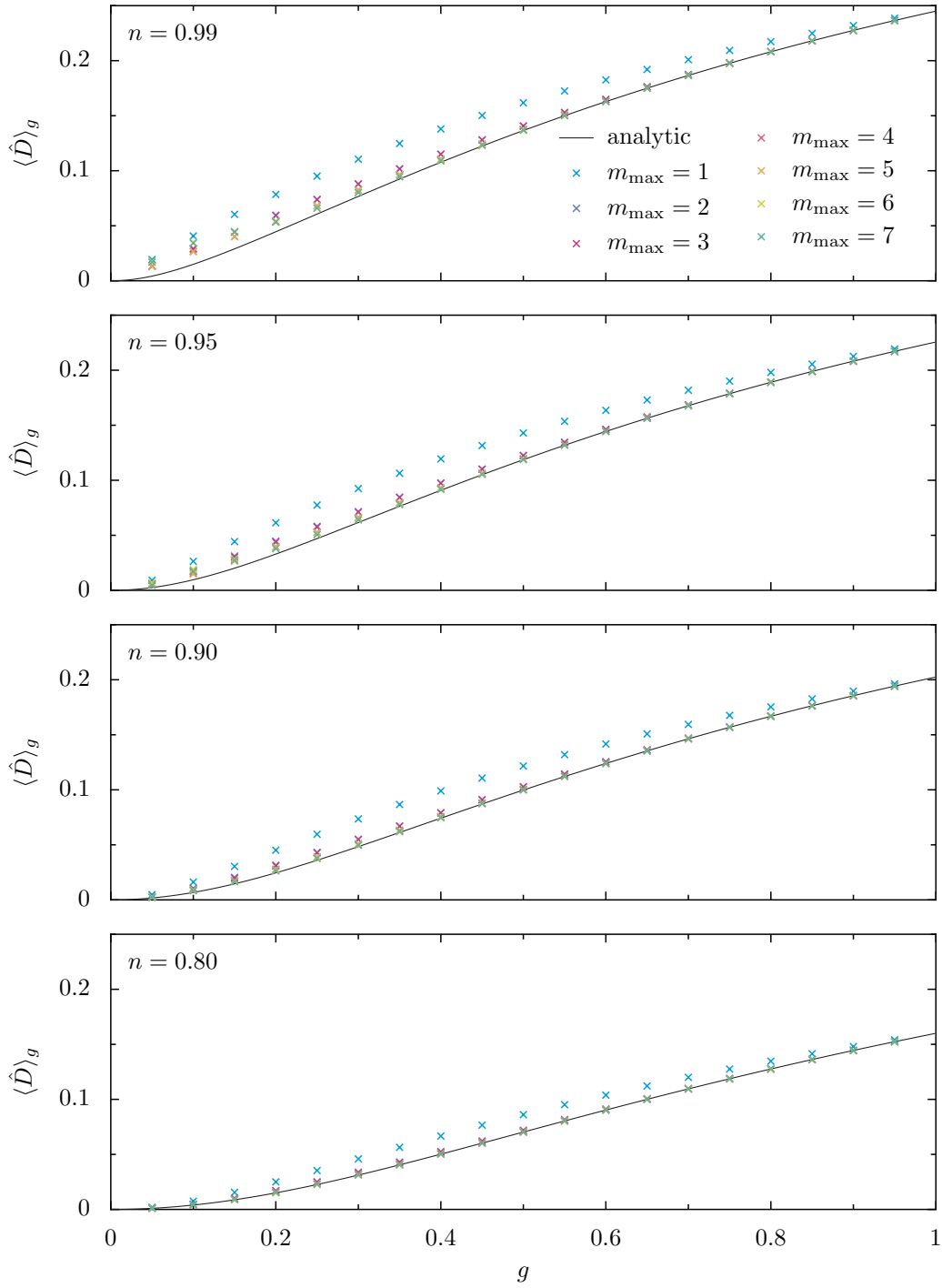


Figure 13: Gebhard's formalism for variable g for several n at fixed $R_{\max} = 15$, $r_{\max} = 5$ and $n_{\max} = 10$

3 Numerical investigation

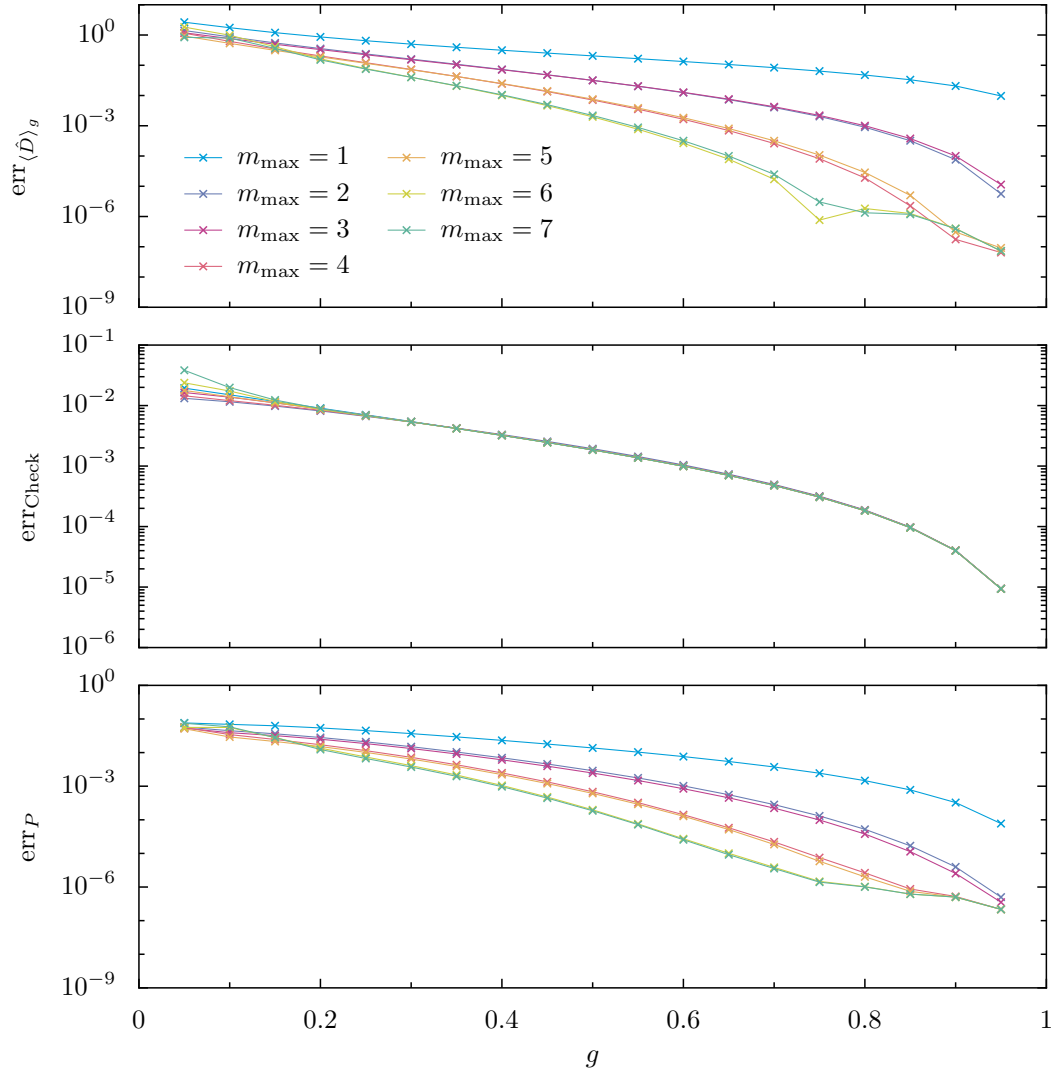


Figure 14: Gebhard's formalism for variable g at fixed $n = 0.95$, $R_{\max} = 15$, $r_{\max} = 5$ and $n_{\max} = 10$

3.6 Program for the square lattice

In one and two dimensions the numerical implementation is very similar and because of that we will focus on the differences. First the calculation of $P^0(x, y)$ is more complicated than the calculation of $P^0(R)$ and second we have more identical terms due to the symmetry of the system. This symmetry can be highly exploited while solving the Dyson equation.

The two general changes from one to two dimensions are that the for-loops over lattice site indices are replaced by two nested for-loops. These two nested for-loops iterate over the two dimensional square lattice. The second change is the storing of the arrays for S , S^* and \bar{P} . In one dimension these arrays are one-dimensional and so the storing of the values is done in the natural way as described in the previous section.

$$S(R) = \text{array_S}[R + R0] \quad (3.97)$$

The function $S(R)$ is restricted to the values $-R_{\max} \leq R \leq R_{\max}$. So the "array_S" has the length $2R_{\max} + 1$ and $R0 = R_{\max}$. Now we store the values of a two dimensional function in an one-dimensional array. We restrict $S(x, y)$ to $-R_{\max} \leq x, y \leq R_{\max}$ and so the "array_S" has the length $(2R_{\max} + 1)^2$.

$$S(x, y) = \text{array_S}[x(2R_{\max} + 1) + y + R0] \quad \text{with} \quad R0 = \frac{1}{2}((2R_{\max} + 1)^2 - 1) \quad (3.98)$$

In the end of section 3.2 the storing of the two dimensional arrays is also mentioned.

Calculating $P^0(x, y)$

The function $P^0(x, y)$ is

$$P^0(x, y) = \int_{-\pi}^{\pi} \frac{dk_x}{2\pi} \int_{-\pi}^{\pi} \frac{dk_y}{2\pi} \langle \hat{n}_{\mathbf{k}\sigma} \rangle_0 \cos(k_x x + k_y y) . \quad (3.99)$$

All states with an energy lower than the Fermi energy ϵ_F are occupied

$$\langle \hat{n}_{\mathbf{k}\sigma} \rangle_0 = \begin{cases} 1 & \text{for } -2(\cos(k_x) + \cos(k_y)) \leq \frac{\epsilon_F}{t} \\ 0 & \text{else} \end{cases} \quad (3.100)$$

The Fermi energy ϵ_F depends in our system only on the filling n and scales with the hopping parameter t .

$$\frac{n}{2} = P^0(0, 0) = \int_{-\pi}^{\pi} \frac{dk_x}{2\pi} \int_{-\pi}^{\pi} \frac{dk_y}{2\pi} \langle \hat{n}_{\mathbf{k}\sigma} \rangle_0 \quad (3.101)$$

3 Numerical investigation

For half filling ($n = 1$) these equations can be solved analytically. The Fermi energy ϵ_F is zero.

$$P^0(x, y, n = 1) = \frac{-\cos(\pi x) + \cos(\pi y)}{\pi^2(x^2 - y^2)} \quad (3.102)$$

The values for $x = y$ or $x = y = 0$ are also calculated correctly by the formula 3.102, if you take the correct limit.

$$P^0(0, 0, n = 1) = \frac{1}{2}, \quad P^0(x, x, n = 1) = \frac{\sin(\pi x)}{2\pi x} \quad (3.103)$$

Below half filling we need to evaluate the integral in 3.99 numerically. In order to use the standard techniques of one-dimensional numerical integration, which provide us with very high accuracy, we transform the two-dimensional integral into an one-dimensional one. $\langle \hat{n}_{\mathbf{k}\sigma} \rangle_0$ drops out in the integral, if the integration boundaries are chosen correctly.

$$P^0(x, y) = \int_{-k_F(0)}^{k_F(0)} \frac{dk_x}{2\pi} \int_{-k_F(k_x)}^{k_F(k_x)} \frac{dk_y}{2\pi} \cos(k_x x + k_y y) \quad (3.104)$$

The Fermi surface $k_F(k)$ is

$$k_F(k) = \arccos\left(-\frac{\epsilon_F}{2t} - \cos(k)\right). \quad (3.105)$$

The integration over k_y for ($y \neq 0$) is executed analytically.

$$P^0(x, y) = \int_{-k_F(0)}^{k_F(0)} \frac{dk_x}{4\pi^2 y} (\sin(k_x x + k_F(k_x) y) - \sin(k_x x - k_F(k_x) y)) \quad (3.106)$$

The argument of the integration is symmetric in k_x and so it is sufficient to do the integration from zero to $k_F(0)$ and multiply by two.

$$P^0(x, y) = \int_0^{k_F(0)} \frac{dk_x}{2\pi^2 y} (\sin(k_x x + k_F(k_x) y) - \sin(k_x x - k_F(k_x) y)) \quad (3.107)$$

The according numerical procedure is to choose a value between minus two and two for the parameter $\frac{\epsilon_F}{t}$ and then to calculate the one dimensional integrals 3.107 for ($y \neq 0$) and the integral

$$P^0(0, 0) = \frac{n}{2} = \int_0^{k_F(0)} \frac{dk_x}{\pi^2} k_F(k_x) \quad (3.108)$$

for ($y = x = 0$) and to determine the filling n numerically. Because $P^0(x, y)$ is invariant under changing the sign of x or y or interchanging x and y , it is sufficient to evaluate $P^0(x, y)$ solely for positive x and y and $x \leq y$.

$$P^0(x, y) = P^0(-x, y) = P^0(x, -y) = P^0(y, x) \quad (3.109)$$

To calculate $P^0(x, y)$ for $-R_{\max} \leq x, y \leq R_{\max}$ it is sufficient to do $\frac{1}{2}(R_{\max}+3)R_{\max}+1$ numerical integrations instead of $(2R_{\max}+1)^2$. We do the numerical evaluation with Mathematica and export the results to a header file for the C program.

Evaluating the diagrams

The self-energy $S^*(x, y)$ has the same symmetries as $P^0(x, y)$, thus it is adequate to evaluate the diagrams only for positive x and y and $x \leq y$. The procedure to evaluate the diagrams is identical to one dimension and on the program level we replace the for-loops in the "recursive" function by two nested for-loops. The for-loop corresponds to the sum over one lattice vector r_i . In one dimension we sum r_i from $-r_{\max}$ to r_{\max} and in two dimensions we now sum its x - and y -component each from $-r_{\max}$ to r_{\max} . This equals a summation over a quadratic plane in space.

Next comes a short discussion why we sum over a quadratic area for the r_i and not over the area of a circle. A circle would enclose all points of the lattice up to its radius. Due to this fact the area of a circle would be the right choice, if $\bar{P}(x, y)$ declines equally in all directions. We are investigating the two dimensional system primarily for n close to half filling. The dressed lines $\bar{P}(x, y)$ originate from the non-dressed lines $P^0(x, y)$ and we expect the $\bar{P}(x, y)$ to have a qualitatively similar behavior to the $P^0(x, y)$. For half filling the absolute value of $P^0(x, x)$ decreases proportional to $\frac{1}{x}$ (eq. 3.103) for the diagonal values and proportional to $\frac{1}{x^2}$ on the x - or y -axis (eq. 3.102). Consequently, our choice of the summation area has to favor the diagonal elements and this is achieved by the square area.

Dyson Equation

$S(x, y)$ and $S^*(x, y)$ has the same invariants as $P^0(x, y)$. They do not change, if we interchange x and y or change the sign of one or two of them. We utilize this fact when we solve the Dyson equation. Instead of a system of $(2R_{\max}+1)^2$ linear equations, we solve a system with only $\frac{1}{2}(R_{\max}+3)R_{\max}+1$ linear equations. We write the Dyson equation in the form

$$\sum_{R_j=0}^{\frac{1}{2}(R_{\max}+3)R_{\max}} \text{matrix_M}[R][R_j] \text{vector_S}[R_j] = \text{vector_S_star}[R] , \quad (3.110)$$

3 Numerical investigation

in which the parameter R runs from zero to $\frac{1}{2}(R_{\max} + 3)R_{\max}$. The vector arrays "vector_S" and "vector_S_star" have the length $\frac{1}{2}(R_{\max} + 3)R_{\max} + 1$ and contain only the independent values of $\frac{S(x,y)}{X}$ and $\frac{S^*(x,y)}{X}$. We stored the values in the following way.

$$\text{vector_S}[] = \left[\frac{S(0,0)}{X}, \frac{S(0,1)}{X}, \frac{S(1,1)}{X}, \frac{S(0,2)}{X}, \dots, \frac{S(R_{\max}, R_{\max})}{X} \right] \quad (3.111)$$

Again we calculate with $\frac{S(x,y)}{X}$ instead of $S(x,y)$ to eliminate rounding errors, if we divide by a small value X . In order to get an arbitrary value $\frac{S(x,y)}{X}$ from the vector array "vector_S"

$$\frac{S(x,y)}{X} = \text{vector_S}[f(\min(|x|, |y|), \max(|x|, |y|))] , \quad (3.112)$$

we need to read the entry specified by the function $f(a, b)$.

$$f(a, b) = \left(\sum_{x=1}^a 1 \right) + \left(\sum_{y=1}^b \sum_{x=0}^y 1 \right) = a + \frac{b}{2}(b + 1) \quad (3.113)$$

The final ingredient for the Dyson Equation is to create "matrix_M". First we set it equal to the identity matrix, with ones on the diagonal entries and zeros on every other entry. Then the contributions from S and P^0 are added in the following code:

```

int L0H = 4*RMax+1;
int R0H = L0H*L0H-1;
for (Ry=0; Ry<=RMax; Ry++){
  for (Rx=0; Rx<=Ry; Rx++){
    for (R2x=-RMax; R2x<=RMax; R2x++){
      for (R2y=-RMax; R2y<=RMax; R2y++){
        matrix_M [ f (Rx, Ry) ] [ f (R2x, R2y) ] -=
          vector_P_H [ (Rx-R2x)*L0H+(Ry-R2y)+R0H ] ;
      }
    }
  }
}

```

The array "vector_P_H[]" contains the sum of S with P^0 .

```

int L0 = 2*RMax+1;
int R0 = L0*L0-1;
for (Rx=-RMax; Rx<=RMax; Rx++){
  for (Ry=-RMax; Ry<=RMax; Ry++){
    vector_P_H [Rx*L0H+Ry+R0H] = 0.;
    for (R2x=-RMax; R2x<=RMax; R2x++){
      for (R2y=-RMax; R2y<=RMax; R2y++){
        vector_P_H [Rx*L0H+Ry+R0H] +=
          X*vector_S_star [R2x*L0+R2y+R0] * P0 (Rx-R2x, Ry-R2y) ;
      }
    }
  }
}

```



```

    }
  }
}

```

The array "vector_P_H[]" is longer by $2R_{\max}$ than the array "vector_S_star" in order to avoid memory errors during the initialization of matrix_M. All entries of array "vector_P_H[]", which are not filled in this code block, are set to zero. The multiplication by 'X' is necessary, because "vector_S_star" contains only $S^*(x, y)$ divided by 'X'. The linear system of equations 3.110 is again solved in our code with the LAPACK package.

Results

Finally we shall discuss the numerical results of the two dimensional case in Gebhard's formalism. The discussion for the program in one dimension showed that Gebhard's formalism is better suited for our numerical approach. In two dimensions it should be even better, because Gebhard's diagrammatic expansion starts from the exact solution in infinite dimensions. The basic idea of any variational wave function is to find an upper limit for the ground state energy and for our two dimensional Hubbard model the energy expectation value is

$$\langle \hat{H} \rangle_g = U \langle \hat{D} \rangle_g - 4tP(1, 0) . \quad (3.114)$$

Due to the fact that only $\langle \hat{D} \rangle_g$ and $P(1, 0)$ contribute to the energy expectation value, we are most interested in these two values. These two expectation values are plotted as functions of g in figure 15. In both diagrams of figure 15 all lines have qualitatively the same behavior. The results from order two and higher are on top of each other and the values from the first order are slightly above of those. When we compare the discrepancy between first order and the others, it is smaller than in one dimension (figure 13). Because the first order calculations is the exact solution in infinite dimensions, it converges to the exact solution by increasing the number of spatial dimensions.

Next we examine the differences between our results for different orders and define an error estimator to rate our approach. From one dimension we learned that our approach increases in accuracy with increasing order. Thus we take the normed difference between one order and the next higher one as an estimation for the deviation to the exact solution. For the double occupation the error estimator for order i is defined as

$$\text{errEst}_{\langle \hat{D} \rangle_g} = \frac{\left| \left(\langle \hat{D} \rangle_g^{\text{numeric}} \right)_{m_{\max}=i+1} - \left(\langle \hat{D} \rangle_g^{\text{numeric}} \right)_{m_{\max}=i} \right|}{\left| \left(\langle \hat{D} \rangle_g^{\text{numeric}} \right)_{m_{\max}=i} \right|} \quad (3.115)$$

3 Numerical investigation

and for $P(1,0)$ we analogously define an error estimator

$$\text{errEst}_{P(1,0)} = \frac{|(P(1,0)^{\text{numeric}})_{m_{\max}=i+1} - (P(1,0)^{\text{numeric}})_{m_{\max}=i}|}{|(P(1,0)^{\text{numeric}})_{m_{\max}=i}|}. \quad (3.116)$$

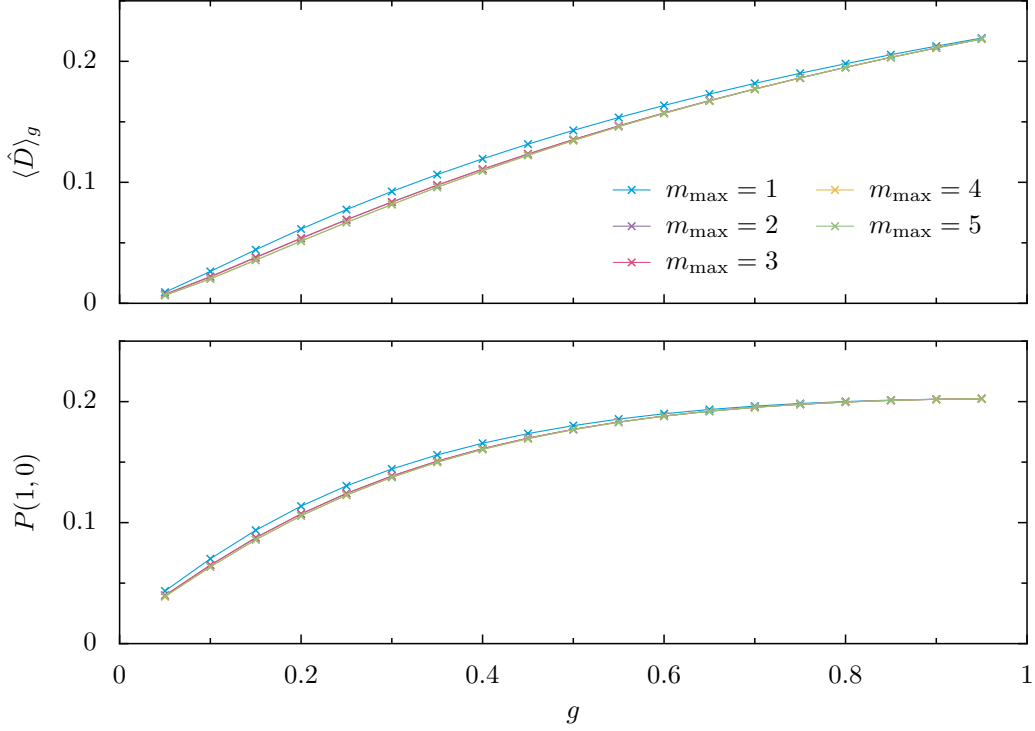


Figure 15: Gebhard's formalism for variable g at fixed $n = 0.95$, $R_{\max} = 15$, $r_{\max} = 5$ and $n_{\max} = 10$

These two predictions for the error and $\text{err}_{\text{Check}}$ are displayed in figure 16. As in one dimension the values of $\text{err}_{\text{Check}}$ are equally good for all different orders, also the magnitude of the values is very similar as in one dimension. For g close to zero in one dimension we observed a slight divergence of the lines (figure 14), which is not the case for the square lattice. This fact could be an indication that our approach works in two dimensions even better than in one for g close to zero.

We calculated values up to the fifth order, so we have the predicted errors up to order four. Both $\text{errEst}_{\langle \hat{D} \rangle_g}$ and $\text{errEst}_{P(1,0)}$ have the same trend for g as the deviation from the analytic solution in one dimension (figure 14). They decrease for several magnitudes of ten, when we alter g from zero to one. The estimated errors for an even order are significantly smaller than the predicted errors for an odd order. In one dimension we observed the phenomenon that the results from

3.6 Program for the square lattice

an even order and the results from the next higher odd order are nearly identical (figure 14). If this phenomenon also exists in two dimensions, this would explain, why the error estimator produce so different results for even and odd orders. When we compare the shape and the values of the estimated errors and the exact errors in one dimension, than we find that curves of odd order fit very well to the exact error curves of one dimension. We conclude from this observation, that on the one hand our error estimator for odd orders predicts the error very well. On the other hand the predictions from the error estimator of even orders are not suitable.

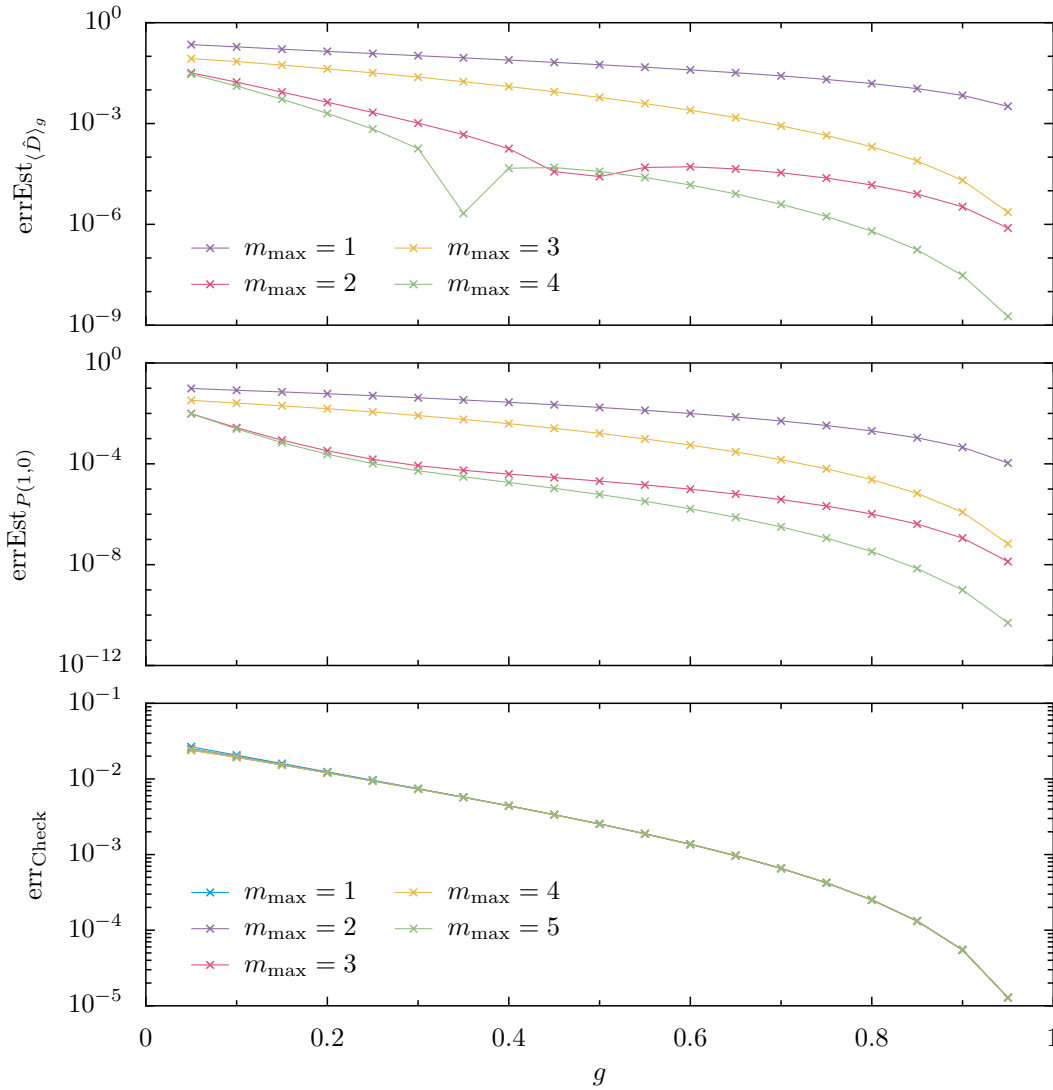


Figure 16: Gebhard's formalism for variable g at fixed $n = 0.95$, $R_{\max} = 15$, $r_{\max} = 5$ and $n_{\max} = 10$

Limit summation

In two dimensions the evaluation of the diagrams are much more computationally time consuming than in one dimension. In order to get contributions from sixth and seventh order diagrams, we have to restrict the summation for these diagrams even more. For each order the summation limitation $r_{\max}(m)$ is chosen small enough that the number of summands does not exceed a number N_{\max} .

$$(2r_{\max}(m) + 1)^{2(m-2)} \leq N_{\max} \tag{3.117}$$

The summation limit $r_{\max}(m)$ for the order m is the minimum of the global summation limit r_{\max} and the inversion of inequality 3.117.

$$r_{\max}(m) = \min \left(r_{\max}, \left\lfloor \frac{1}{2} \left(N_{\max}^{\frac{1}{2(m-2)}} - 1 \right) \right\rfloor \right) \tag{3.118}$$

The symbols $\lfloor \dots \rfloor$ indicate rounding down to the next integer. In the example of figure 17 we have the following values for $N_{\max} = 100000$

$$\begin{aligned} r_{\max}(3) &= 10, \\ r_{\max}(4) &= 8, \\ r_{\max}(5) &= 2, \\ r_{\max}(6) &= 1, \\ r_{\max}(7) &= 1. \end{aligned} \tag{3.119}$$

The values of $\text{err}_{\text{Check}}$ has the anticipated form in figure 17. Also the shape and the magnitude of the graphs up to order four of the estimated errors is very similar to the corresponding plots with only the simple r_{\max} (see figure 16). For the fifth order the estimated error behaves in the expected way, because its line runs parallel to that of order one and three. Also its magnitude is around the right values. The sixth order also continues the trend of order two and four. As we stated before, the estimated errors for odd orders reproduce the exact errors of one dimension in an adequate way, even the new order five.

On the one hand this order dependence of $r_{\max}(m)$ allows us to evaluate higher orders and gives proper results. But on the other hand the estimated errors become artificially smaller, because for each higher order the summation cutoff $r_{\max}(m)$ becomes even smaller. From one dimension we learned that higher orders profit more from increased r_{\max} , therefore it is counterproductive to restrict r_{\max} for higher orders. Thus higher orders cannot contribute as much as they should, which in return reduces the estimated errors artificially.

Despite this contradiction, we believe that the higher order diagrams contribute still in a sufficient manner. Due to the fact that the self-consistency cycle is iterated

several times, the higher order diagrams do not contribute only in their summation, but also in the lower order diagrams in the form of the lines $\bar{P}(x, y)$. The lines $\bar{P}(x, y)$ contain all diagrams from the previous iterations. Even if these results appear to be proper, we still have to be careful with those of the error estimators.

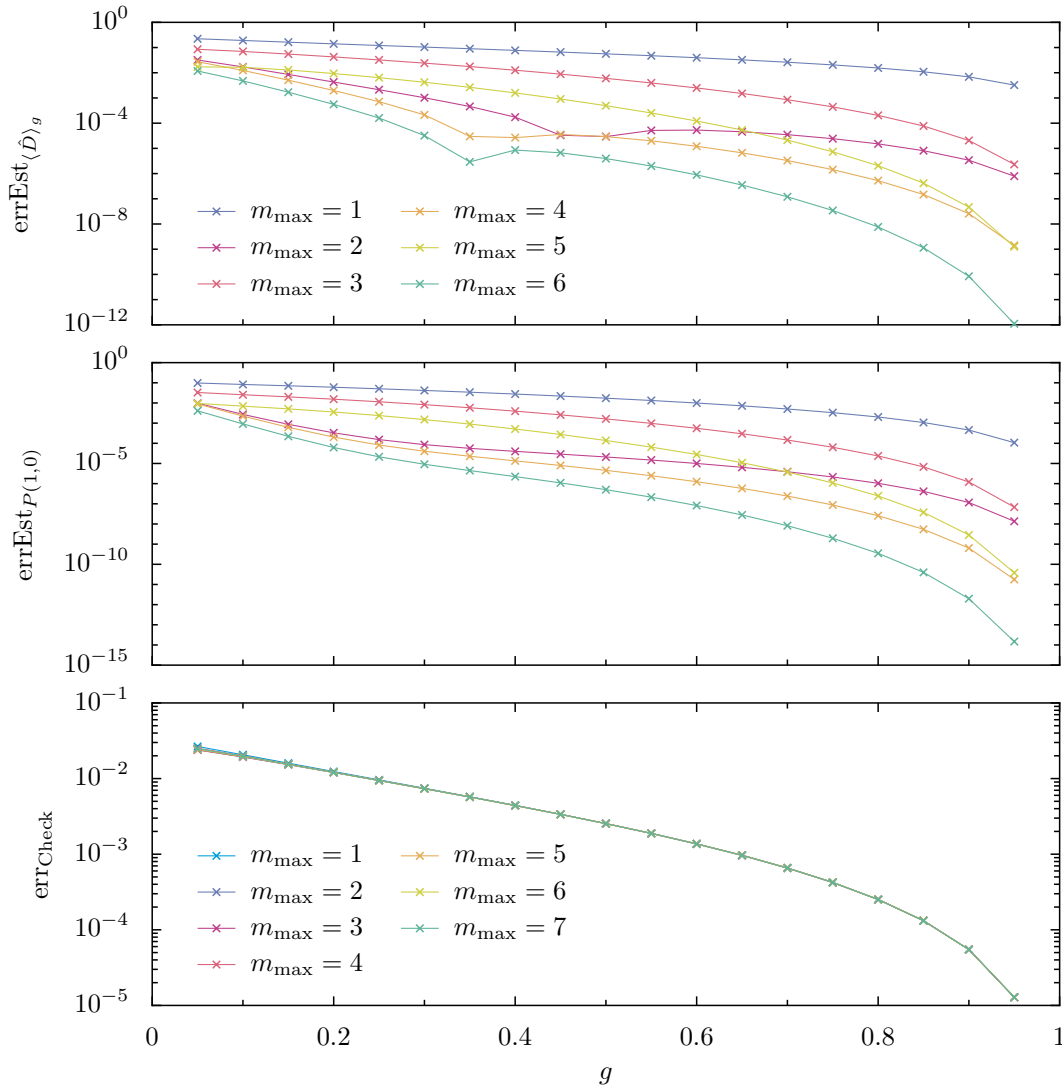


Figure 17: Gebhard's formalism for variable g at fixed $n = 0.95$, $n_{\max} = 10$, $R_{\max} = 15$, $r_{\max} = 10$ and $N_{\max} = 100000$

Stochastic summation

We showed that choosing r_{\max} depended of the different orders is a good and simple way to enable contributions of greater r_{\max} and higher order diagrams. But it still

3 Numerical investigation

remains the fact from one dimension, that the higher order diagrams contribute even more for sufficiently large r_{\max} . We attempt to solve this shortcoming by doing a stochastic summation for large r_{\max} in high orders.

When we evaluate the diagrams, we now start in each order with a summation up to cutoff $r_{\max}(m)$. Exactly the way we described in the previous part. If this $r_{\max}(m)$ is smaller than the global r_{\max} , we do the summation for the remaining region stochastically. In the stochastic summation we have to consider that the contribution of the diagrams are dropping for large lattice vectors \mathbf{r}_i . We incorporate this fact by doing the stochastic summation first for a small shell. This shell includes for order m all sets of lattice vectors $\{\mathbf{r}_0, \dots, \mathbf{r}_{m-3}\}$ outside of the original summation area ($\mathbf{r}_i \leq r_{\max}(m)$) and inside a by Δ increased summation area ($\mathbf{r}_i \leq r_{\max}(m) + \Delta$). A lattice vector \mathbf{r}_i is smaller than the integer $r_{\max}(m)$, if the absolute values of all components of \mathbf{r}_i are smaller than $r_{\max}(m)$.

In this shell we evaluate the m -th order diagrams for N_{\max} random lattice vector sets and sum these values up. The final contribution of this shell to the self energy is the value of the sum times the number of lattice vector sets inside the shell divided by N_{\max} . Next we evaluate the next bigger shell with the same procedure, for this we only have to replace the lower cutoff $r_{\max}(m)$ with $r_{\max}(m) + \Delta$ and the upper cutoff $r_{\max}(m) + \Delta$ with $r_{\max}(m) + 2\Delta$. We repeat this iteration over shells until the upper cutoff is larger than the global cutoff r_{\max} .

The results of this stochastic summation for shells is shown in figure 18. The graphs in these diagrams are very similar to those of figure 16 and so we conclude that the larger r_{\max} does not give much higher accuracy.

The advantage of this stochastic summation over shells is that we can include additional error estimators in the numerical process. If we do the stochastic summation in a shell we can calculate the mean value of all N_{\max} contributions and the estimated error is the deviation of the single points from the mean value. If the deviation is too large, then we could increase the number N_{\max} and add more random points. Secondly the contribution of additional shells to the proper self-energy S^* can be estimated by the quotient of the last calculated shell contribution divided by the proper self-energy. The proper utilization of these two additional error estimators is still in progress. These two predictions for errors can be used to determine the quality of our results or to spare the evaluation of diagrams in regions, where they contribute very little.

3.6 Program for the square lattice

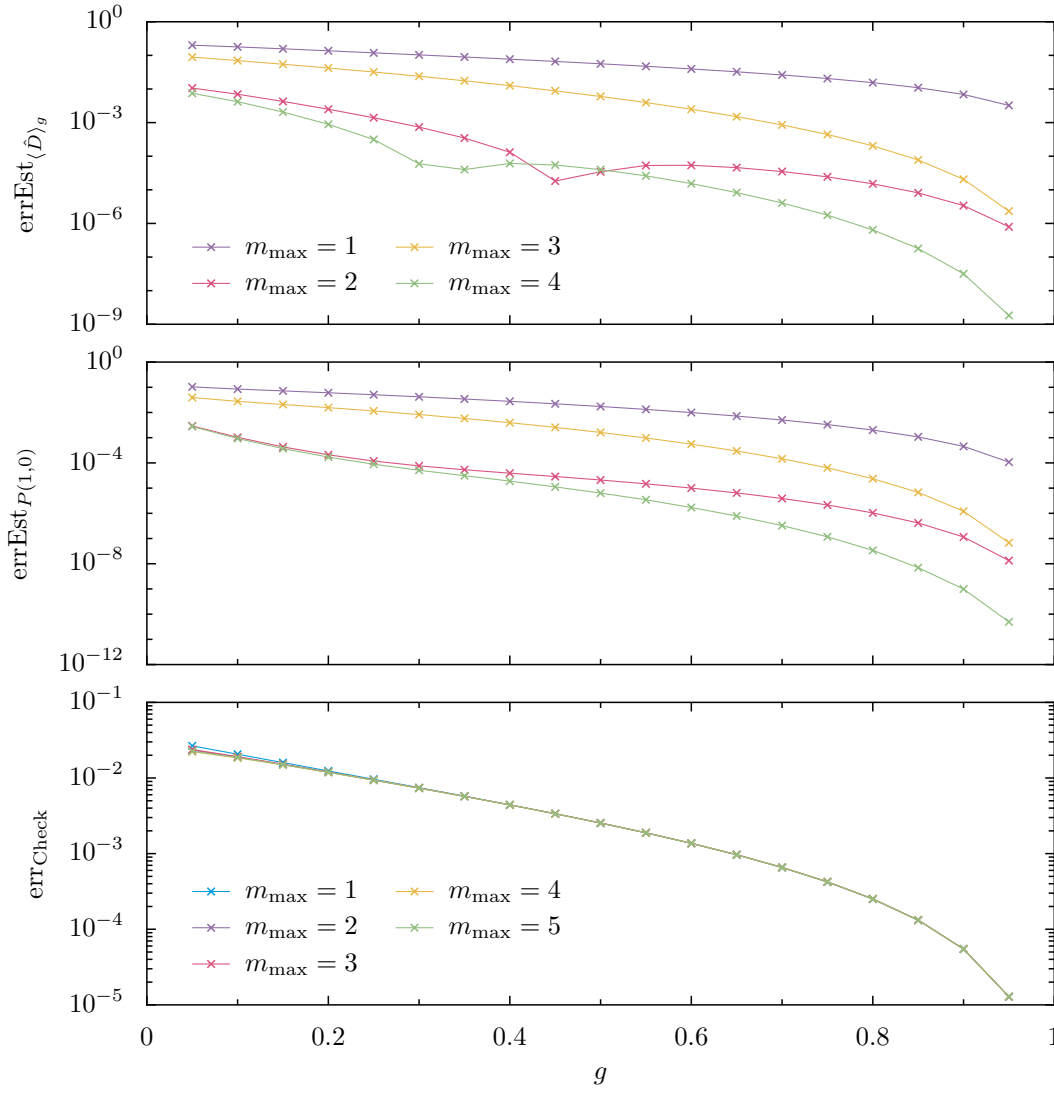


Figure 18: Gebhard's formalism for variable g at fixed $n = 0.95$, $n_{\max} = 10$, $R_{\max} = 15$, $r_{\max} = 10$, $N_{\max} = 10000$ and $\Delta = 1$

4 Summary

We started this thesis with the aim to evaluate numerically the Gutzwiller wave function (GWF) for the equally spaced square lattice in one and two dimensions in the Hubbard model. The method of our choice for calculating the expectation values of double occupation and density matrix are two diagrammatic expansions. Metzner and Vollhardt developed the first expansion and it is discussed in detail in section 2.3. The second expansion is an optimized form of the first expansion and was developed by Gebhard. Due to the similarity of the two expansions the discussion of the second one is much shorter in section 2.4.

Before evaluating the diagrammatic expansions we need all the skeleton diagrams up to our cutoff order. Therefore we used a Mathematica program that reads in the diagrams from a φ^4 -theory, because the diagrams of the φ^4 -theory are topologically equivalent to the diagrams we need. Then the program creates all skeleton diagrams for the numerical evaluation and export those in a suitable way. Our program exported the diagrams up to order seven for the numerical evaluation in one and two dimensions.

First, we applied the two formalisms in one dimension and compared the results with the available analytic solution. The main insight we gained was that Gebhard's formalism is suited much better for our numerical approach than the other formalism. On the one side Metzner and Vollhardt's formalism starts from the uncorrelated state and on the other side Gebhard's formalism is in zeroth order identical to the Gutzwiller approximation, which is the exact solution for infinite spatial dimensions and already a qualitatively description for one dimension. In the numerical investigation we therefore concentrate on Gebhard's formalism.

Next we investigated the dependency on the cutoff parameters in our numerical approach. Self-consistency is reached fast a few iterations for various parameters for the self-consistency cycle and we can stop to iterate further. The three major steps in the self-consistency cycle are evaluation of skeleton diagrams with dressed lines for the proper self-energy, calculating the self-energy by the Dyson equation and finally calculating new dressed lines for the next iteration. All equations are intended for real space and not momentum space and thus the diagrammatic evaluation happens also in real space.

We have two cutoff parameters in the diagrammatic evaluation, first a limit in the summation over lattice vectors and second a upper limit for the range of the proper self-energy. Choosing the summation limit greater than six affected the accuracy very little, but higher orders profit more from a larger summation limit

than lower orders. Increasing the range limitation improves accuracy, especially the error in a sum-rule, which is valid for the GWF in all dimensions, is strongly affected by this parameter. Also this parameter dependence rises with the maximally used order of diagrams. We conclude from that that the cutoff parameter in real space have to be enlarged, when we want to fully utilize higher order diagrams.

The physical parameters filling n and Gutzwiller parameter g strongly determine the quality of our results. For nearly half filled band and g close to zero, the relative accuracy of our approach decreases. For large g or less than half-filled band the accuracy is very high. An interesting phenomenon showed up: The accuracy of a result up to an even order is almost as good as the result of the next higher odd order. So the accuracy improvement from one order to the next is alternately small or large. In two dimensions this phenomenon was also observed. We defined an error estimator by the relative difference of one order with the result of the next higher order. The estimated error was in general greater for odd orders than for even orders. The estimated errors for the odd orders is very similar in size and qualitative behavior to the exact error in one dimension.

Due to the computationally more expensive summation over the lattice in two dimensions than in one dimension, we are not able to evaluate the diagrams up to order seven for a suitable set of cutoff parameters. Therefore we improved the summation first by choosing the summation cutoff dynamically for each order. This allowed us to utilize diagrams up to seventh order, but the summation cutoff for high orders is very small. This circumstance we improve by first determining the summation limit adaptively and replacing the exact summation by a stochastic summation. The proper implementation of the adaptive scheme and the stochastic summation is still in progress.

Appendix

Program in dim 1

In section 3.5 the self-consistency cycle of the program is discussed. In the following is the complete main file of the program printed including the self-consistency cycle:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <lapacke.h> //package to solve linear equation
#include <omp.h> //package to parallelize
#include "DiagramsForCinDim1Order7.h" //orderfunctions
#include "Diagrams_functions.h" //definitions of functions

typedef double (*ORDER_FUNC)(int R, int r [], double G [], int GMax);

int main(int argc, char *argv [])
{
    double g = atof(argv [1]);
    double n = atof(argv [2]);
    int RMax = atoi(argv [3]);
    int rMax = atoi(argv [4]);
    int mMax = atoi(argv [5]);
    int nMax = atoi(argv [6]);

    printf("g=%g\n", g);
    printf("n=%g\n", n);
    printf("RMax=%d\n", RMax);
    printf("rMax=%d\n", rMax);
    printf("mMax=%d\n", mMax);
    printf("nMax=%d\n", nMax);
    printf("\n");

    ORDER_FUNC *order = (ORDER_FUNC *)calloc((8), sizeof(ORDER_FUNC *));
    //order[0] is empty to keep it easier to read
    order [1] = order1;
    order [2] = order2;
    order [3] = order3;
    order [4] = order4;
    order [5] = order5;
    order [6] = order6;
    order [7] = order7;

    int R0long = RMax+mMax*rMax;
    int R0 =(Rlength-1)/2;
    double *array_overline_P_old = calloc (2*(RMax+mMax*rMax)+1,
                                             sizeof(double));
    double *arra_S_star = calloc (2*RMax+1, sizeof(double));
```

4 Summary

```
double *array_S = calloc(2*RMax+1, sizeof(double));

int R;
for (R=-RMax; R<=RMax; R++){
    array_overline_P_old [R+R0long] = P0(R,n);
};
double X = x(g,n);

int iter;
for (iter=1; iter<=nMax; iter++){
    //setting array_S_star back to zero
    for (R=-RMax; R<=RMax; R++){
        array_S_star [R+R0] = 0.;
        array_S [R+R0] = 0.;
        array_overline_P_new [R+R0] = 0.;
    }

    //order mMax to 3
    int m;
    for (m=mMax; m>=3; m--){
        double *sumHelp = calloc(RMax+1, sizeof(double));
        double multHelp = pow(X, ((double) m) - 1.);
        {
            #pragma omp parallel for
            for (R=0; R<=RMax; R++){
                int r [m-2];
                recursive(&sumHelp [R], R, m-3, r, rMax,
                    array_overline_P_old , 2*R0long+1, order [m]);
            }
        }
        for (R=0; R<=RMax; R++){
            array_S_star [R+R0] +=multHelp*sumHelp [R];
        }
    }

    //second order
    int r [1];
    if (mMax>=2){
        for (R=0; R<=RMax; R++){
            array_S_star [R+R0] += X*order [2](R, r, array_overline_P_old ,
                2*R0long+1);
        }
    }

    //first order
    if (mMax>=1){
        array_S_star [0+R0] += order [1](0, r, array_overline_P_old ,
            2*R0long+1);
    }
}
```

```

//make array_S_star symmetric
for(R=1; R<=RMax; R++){
    array_S_star[-R+R0] = array_S_star[R+R0];
};

//calc array_S from array_S_star by Dyson equation
calcSfromSstar(array_S_star, RMax, g, n, array\_S);

//calc array_overline_P_new from array_S
calcOverlinePFromS(array_S, RMax, g, n, array_overline_P_new);

//for the next iteration set array_overline_P_old
//equal to array_overline_P_new
for(R=-RMax; R<=RMax; R++){
    array_overline_P_old[R+R0long] = array_overline_P_new[R+R0];
}
}

//calculate double occupation
double D = calcDoubleOccupation(array_S, RMax, g, n);
printf("D=_%6.6g\n", D);
//calculate P
double *P = calloc(2*RMax+1, sizeof(double));
calcPfromS(array_S, RMax, g, n, P);
for(R=0; R<=RMax; R++){
    printf("P[%d]=_%6.6g\n", R, P[R+R0]);
}
double check = sumRuleCheck(g,n,RMax,P,D);
printf("Summation_Rule_Check=_%6.6g\n", check);

free(array_S_star);
free(array_S);
free(array_overline_P_old);
free(array_overline_P_new);
free(P);

return 0;
}

```


References

- [1] W. Metzner and D. Vollhardt, *Ground-state properties of correlated fermions: Exact analytic results for the Gutzwiller wave function*, Phys. Rev. Lett. **59**, 121 (1987).
- [2] W. Metzner and D. Vollhardt, *Analytic calculation of ground-state properties of correlated fermions with the Gutzwiller wave function*, Phys. Rev. B **37**, 7382 (1988).
- [3] W. Metzner, *Variational theory for correlated lattice fermions in high dimensions*, Zeitschrift für Physik B Condensed Matter **77**, 253 (1989).
- [4] F. Gebhard, *Gutzwiller correlated wave functions in finite dimensions d: A systematic expansion in $1/d$* , Phys. Rev. B **41**, 9452 (1990).
- [5] M. C. Gutzwiller, *Effect of Correlation on the Ferromagnetism of Transition Metals*, Phys. Rev. Lett. **10**, 159 (1963).
- [6] M. C. Gutzwiller, *Effect of Correlation on the Ferromagnetism of Transition Metals*, Phys. Rev. **134**, A923 (1964).
- [7] M. C. Gutzwiller, *Correlation of Electrons in a Narrow s Band*, Phys. Rev. **137**, A1726 (1965).
- [8] M. Alexander, *Computergenerierte Skelett-Diagramme für die Gutzwiller-Wellenfunktion*, University of Augsburg, 2015, Projektarbeit.
- [9] H. Kleinert, A. Pelster, B. Kastening, and M. Bachmann, *Recursive graphical construction of Feynman diagrams and their multiplicities in φ^4 and $\varphi^2 A$ theory*, Phys. Rev. E **62**, 1537 (2000).
- [10] Z. Gulácsi and M. Gulácsi, *Analytic description of the Hubbard model in D dimensions with the Gutzwiller wave function*, Phys. Rev. B **44**, 1475 (1991).
- [11] M. Kollar and D. Vollhardt, *Exact analytic results for the Gutzwiller wave function with finite magnetization*, Phys. Rev. B **65**, 155121 (2002).