Ralf Gamillscheg

Occupation Number Space Partitioning by Basis Reordering and Multiscale Approaches for Strongly-Correlated Material Hamiltonians

DOCTORAL THESIS

For obtaining the academic degree of

Doktor der technischen Wissenschaften

Doctoral Programme of Technical Sciences
Technical Physics

Graz University of Technology

Supervisor:
Univ.-Prof. Wolfgang von der Linden
Institute of Theoretical and Computational Physics

Graz, September 2010

TU Graz
Graz University of Technology
Senat

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

Zusammenfassung

Die numerische Behandlung eines Hamilton-Operators stark korrelierter Vielteilchensysteme führt zu exponentiell großen Eigenwertproblemen als Funktion der Systemgröße. In dieser Dissertation werden einige neue Zugänge dazu dargelegt.

Die Two-Subsystem Ground State Approximation (TSGSA) wurde durch eine Methode der linearen Elastomechanik inspiriert. Diese beruht auf einer Partitionierung der Elementknoten einer Finiten-Elemente oder Finiten-Differenzen Diskretisierung in kleinere Untersysteme. Die Besetzungszahlbasis eines Vielteilchensystems kann auf eine ähnliche Weise zerlegt werden. Die tensorielle Struktur des Hamilton-Operators macht diese Systeme ideal für eine solche Behandlung.

Eine weiterer Ansatz ist die Numerical Projection Technique (NPT), bei der die Basis nach Maßgabe eines abstoßenden Potentials der Teilchen umgeordnet wird. Durch eine Transformation mit Hilfe des Schur-Komplements können verhältnismäßig kleine Matrizen verwendet werden, die die Eigenwertberechnung drastisch erleichtern.

Die sehr erfolgreiche Multigrid-Methode erlaubt es, die Auffindung von Lösungen linearer Gleichungssysteme durch Verwendung unterschiedlicher Diskretisierungen stark zu beschleunigen. Ähnliche Techniken können auf die Suche nach Eigenlösungen der Vielteilchensysteme angewendet werden.

Abstract

Solving effective Hamiltonians of strongly correlated materials leads to exponentially large eigenvalue calculations as a function of system size. Several new approaches to these problems are presented in this thesis.

The Two-Subsystem Ground State Approximation (TSGSA) was inspired by a method of elasto-mechanics, the Automated Multilevel Substructuring Method (AMLS). It relies on partitioning the node space of a finite element or finite difference discretization into smaller sub-domains. To apply a similar approach in the field of many-body Hamiltonians, the occupation number space is partitioned accordingly. The tensorial structure of the Hamiltonians make them even more suitable for this method.

Another approach is a Numerical Projection Technique (NPT) using a reordering scheme on the occupation number basis, which is based on the value of a repulsive potential. In this case a Schur congruence transformation can be used to lower the size of the matrices involved and make the eigenvalue calculation more feasible.

A very successful method of applied mathematics is the Multigrid approach. Here, finding the solution is accelerated by using different lattice discretizations. Similar techniques could be used to accelerate the search for eigensolutions of strongly correlated Hamiltonians.

Contents

Preface

This work was done within the *Doktoratskolleg (DK) Numerical Simulations in Technical Sciences* which is an interdisciplinary doctoral program in Graz funded by the FWF Austrian Science Fund. At the time of the start of this PhD project, there were 10 proposing professors and 15 PhD positions. One of the aims was to bring together students from science, mathematics and engineering for the benefit of collaboration in research and education.

For this thesis, the project title was *Sparse Eigenvalue/-Vector Solvers and Clustering*. Eigenvalue computations play an important role in physics, applied mathematics, civil engineering, and electrical engineering. The project had the objective to exchange ideas and approaches to eigenvalue computation between different research fields. Interdisciplinarity may result in profit for all involved parties.

The main topic of this thesis was to find new approaches to the many-body problem for strongly correlated materials in many-body physics, inspired by methods of applied mathematics and engineering. In this research field, large algebraic eigenvalue problems arise from the Schrödinger equation for systems of interacting particles inside a solid. The main challenge is the exponential increase of the dimensionality of the eigenvalue problems with respect to the system size. The details will be discussed in chapter 1.

Strongly correlated particles play an important role in many solid state bodies. A famous effect is the appearance of insulating phases in metals due to strong correlations (Mott insulators [1]). There are prominent examples of chemical compounds like the cuprates and the manganites, which were under heavy investigation in the past decades. The cuprates [2] were examined with the objective to find an explanation for the phenomenon of *high temperature superconductivity*. It is believed that strong interactions give rise to this important effect, which is used in many technical applications. Manganites [3] led to the exploration of phenomena like the *colossal magneto-resistance*, used in modern hard disks of almost every computer nowadays. More recently, optical lattices and light-matter interaction systems attained attention by the scientific community. These systems play an important role for e.g. the development of *quantum computers*. In addition to that there are many more interesting quantum effects due to the astonishing properties of strongly correlated systems.

In order to study the unique properties of these quantum systems one relies on the introduction of effective models which limit the amount of degrees of freedom, which depends exponentially on the system size. This is necessary to make the calculations feasible for modern computation techniques. At the same time the models are required to reflect the rich quantum effects which are observed in the experiment. In this way numerical simulations can be used to explore them and explain their origins.

An example for an effective model is the multi-banded *Hubbard model* (see sec. 1.6). For many calculations it is still too complicated, so further models with even more restrictions on the degrees of freedom have been introduced like the *Heisenberg model* or the *tJ-model*. They can be obtained from the Hubbard model by projection techniques (see app. I).

There exists a whole bunch of specialized numerical methods for the evaluation of these models. A few of them are listed in the next paragraphs.

Quantum Monte-Carlo (QMC): This powerful method relies on the evaluation of system observables by the means of stochastics [4]. A special decomposition of the quantum mechanical Hamiltonian introduced by Trotter [5] and Suzuki [6] allowed to treat quantum systems with Monte-Carlo [7, 8]. Further developments included the stochastic series expansion (SSE, [9]) and acceleration techniques like the loop algorithm [10]. Compared to other methods fairly large systems can be treated, and provided with large computational power the physical values can be determined without systematical error. The method has the major drawback of being limited to non-zero temperatures, and the analysis of very low temperatures becomes increasingly demanding. Furthermore, some quantum-mechanical models cannot be treated by QMC due to the *sign-problem*. It assigns a minus sign to certain configuration weights in some cases. In principle, this can be dealt with, but the computational effort becomes exponentially larger.

Density matrix renormalization group (DMRG): As a further development of quantum real-space renormalization groups [11] this method provides means to calculate ground state ($T = 0$) properties for large systems [12, 13, 14, 15]. It relies on consecutively adding sites to a given system and projecting the gained phase space on a small subspace in order to limit the computational effort in terms of memory and processing power. The major problem is addressing systems which extend in more than one dimensions. It is possible to treat quasi one-dimensional objects like ladders with this approach, but it fails for fully two-dimensional systems. Also, DMRG suffers from limitations concerning the evaluation of *dynamical properties* of quantum systems. They give insight into system excitations which are highly important to understand physical effects of materials. Only recently a mathematical foundation of the algorithm was developed by introducing matrix product states (MPS, [16, 17]). With this a connection between the performance of DMRG and the notion of *entanglement* between clusters was established.

Cluster perturbation theory (CPT) and variational cluster approach (VCA): Also a highly competitive method for the calculation of ground state and dynamical material properties [18, 19]. Here, small physical clusters of only a few particles are constructed which can be easily calculated exactly. These clusters are combined to build the Green's function (see sec. 1.8) of the whole system. In this approach, strong-coupling perturbation theory up to leading order is used to reproduce the previously omitted correlations between the clusters. An extension using variational parameters to improve the obtained Green's functions is the variational cluster approach (VCA, [20]).

Exact diagonalization (ED): For some cases, the only viable approach is to fall back on a purely algebraic eigenvalue solver. In this case only fairly small systems can be evaluated, although the matrix representations of the eigenproblems have the nice property of being *sparse*. For that particular case, many sophisticated iterative eigenvalue solvers exist [21], which can be applied here. The very powerful yet simple and effective *Lanczos* method [22, 23] (see app. C) is often used, as well as the *Davidson* method [24] (see app. F).

During the course of this dissertation multiple new approaches to the eigenvalue problem of strongly correlated systems were developed. One of them is described in chapter 2. The two-subsystem ground state approximation was motivated by a method used in linear elasto-mechanics for the solution of vibrational modes of e.g. membranes (see app. B). It relies on sub-dividing a physical space into smaller regions where the eigenvalue problems are solved individually. Later, the solutions of these sub-domains are used to form a smaller matrix representation of the full problem. This approach is particularly suitable for these many-body systems as the individual sub-systems can be combined by a tensor product.

In chapter 3 another approach is explained. It includes a basis reordering scheme to define partitions which are more or less relevant for the construction of the system ground state. The value of a repulsive potential between particles is taken as defining quantity for the partitions. This method can be seen as extension to existing *projection* techniques used to generate effective Hamiltonians.

In recent years the Multigrid approach to solve large linear systems of equations has been used very successfully. Chapters 4 and 5 illustrate the possibility of adapting the Multigrid idea for strongly correlated system Hamiltonians.

To be able to examine, verify and benchmark the different approaches listed above many different numerical algorithms, implementation techniques and computational frameworks had to be used. Some of them are explained in the appendices.

1 Introduction to Many-Body Physics

1.1 The many-body problem

Quantum mechanics in the context of solid-state physics leads to highly dimensional eigenvalue problems. The full wave function inside a bulk material is a function of the coordinates of all particles inside the domain.

$$\Psi = \Psi(\vec{r}_1, \vec{r}_2, \vec{r}_3, ..., \vec{r}_n; \vec{R}_1, \vec{R}_2, ..., \vec{R}_N) \ . \tag{1.1}$$

Here, the capital R denote the positions of the nuclei, and the lower case letters denote the electron locations. The number of electrons is n, the number of nuclei is N. The wave function is generally a complex function and observes special symmetries as will be described later. Note that it is not directly accessible in an experiment. Only the measurement of the charge density distribution is possible:

$$\rho = e \left|\Psi\right|^2 \ ,$$

with e the elementary charge constant.

The wave function must fulfill the stationary Schrödinger equation

$$\hat{H} \left|\Psi\right\rangle = E \left|\Psi\right\rangle \ . \tag{1.2}$$

It forms an eigenproblem with eigenvalues E which corresponds to the physical quantity of energy. $\hat{H}$ is a Hermitian differential operator, which reads in the most general form:

$$\begin{aligned}\hat{H} = \quad & -\frac{1}{2m}\sum_{i}^{n}\nabla_i^2 - \frac{1}{2M}\sum_{I}^{N}\nabla_I^2 \\ & +\frac{1}{2}\sum_{i}^{n}\sum_{j}^{n}\frac{1}{|\vec{r}_i-\vec{r}_j|} + \frac{1}{2}\sum_{I}^{N}\sum_{J}^{N}\frac{Z_I Z_J}{|\vec{R}_I-\vec{R}_J|} \\ & -\sum_{i}^{n}\sum_{I}^{N}\frac{Z_I}{|\vec{r}_i-\vec{R}_I|}\ .\end{aligned}$$

Here, M denotes the mass of the ions, m the mass of the electrons, and Z is the atomic number. One has to keep in mind, that the Nabla-operators ∇_i act on the coordinates of just one of the particles which is denoted by the index i. And again, capital indices denote sums over the nuclei while lower case indices correspond to the electrons. The first two terms correspond to the kinetic energies of the electrons and the nuclei, respectively. The further expressions include the electron-electron interaction, the nucleus-nucleus interaction and the nucleus-electron interaction.

1.2 The Born-Oppenheimer approximation

In order to reduce the complexity of the problem one fixes the positions of the nuclei [25]. This can be justified by considering that the masses of the nuclei are larger by more than three orders of magnitude.
The problem becomes:

$$\hat{H}_{\text{el.}}\left|\Phi\right\rangle = E_{\text{el.}}\left|\Phi\right\rangle\ , \tag{1.3}$$

with the electronic wave function $|\Phi\rangle$ and the Hamiltonian

$$\hat{H}_{\text{el.}} = -\frac{1}{2m}\sum_{i}^{n}\nabla_i^2 + \frac{1}{2}\sum_{i\neq j}^{n}\frac{1}{|\vec{r}_i-\vec{r}_j|} - \sum_{i}^{n}\sum_{I}^{N}\frac{Z_I}{|\vec{r}_i-\vec{R}_I|}\ . \tag{1.4}$$

1.3 Approximations to the correlations between particles

Well-known methods used in quantum chemistry approximate correlations between particles. As an example, the local density approximation (LDA) states that the wave function

itself is only a function of the local electronic density. This approximation reduces the amount of numerical effort significantly.
The Hartree-Fock method iteratively corrects the electronic distribution by the calculation of the interaction and exchange potentials based on the LDA or comparable approaches, recursively.
In the case of this work no correlations between particles are omitted. Certain quantum effects are believed to have their very origins in these correlations. To study these effects model Hamiltonians are introduced, which can be defined in the framework of the second quantization (see sec. 1.5).

1.4 Fermions and bosons

An important role in the field of solid state physics is played by particle statistics. A crucial property of elementary particles is the indistinguishability, which does not allow an individual particle to be followed by observing a trajectory. Only the particle distribution density can be tracked in an experiment.
Consider a wave function of two particles:

$$\Psi(\vec{r}_1, \vec{r}_2) \ ,$$

and define a permutation operator, which swaps these two particles

$$\mathcal{P}\Psi(\vec{r}_1, \vec{r}_2) = \Psi(\vec{r}_2, \vec{r}_1) \ .$$

As stated, due to the indistinguishability the physical observable quantity, the particle density $\rho = |\Psi|^2$ must not change:

$$\begin{aligned} |\mathcal{P}\Psi(\vec{r}_1, \vec{r}_2)|^2 &= |\Psi(\vec{r}_1, \vec{r}_2)|^2 \\ \mathcal{P}\Psi(\vec{r}_1, \vec{r}_2) &= \Psi(\vec{r}_1, \vec{r}_2)e^{i\phi} \ . \end{aligned}$$

So the permutation operator does nothing but apply a phase to the wave function. If the two particles are exchanged twice, the particles are in the initial state again, not only from the physics point of view, but also mathematically:

$$\begin{aligned} \mathcal{P}^2 &= 1 \\ \Psi(\vec{r}_1, \vec{r}_2) &= \pm\Psi(\vec{r}_2, \vec{r}_1) \ . \end{aligned}$$

The phase can either be 1 or -1. This leads to the following definition: particles which have a symmetric wave function (+ in the above equation) are called *bosons* and particles described by a anti-symmetric wave function $(-)$ are *fermions*.

Note that elementary particles which compose the matter in the universe like electrons and quarks are all fermions. Although the notion of changing sign of the overall wave function by swapping two particles seems to be rather odd it leads in the end to the stability of matter.

1.5 2$^{\text{nd}}$ quantization

The second quantization [26] replaces the description of a solid body system in continuous space with a description by discrete quantum operators. To represent the wave function a set of basis functions is introduced which is based on atomic orbitals. This can be motivated by the notion of localized particles, which are described by a localized wave function at a specific site. The dynamics of the particles is simulated by hopping processes from site to site.

The following considerations are restricted to fermions as they are the main subject of interest in this work.

1.5.1 Fermionic field operators

Define operators which create or annihilate a fermionic field at position $\vec{r}$ [25]:

$$\hat{f}^\dagger(\vec{r}), \quad \hat{f}(\vec{r}) \ .$$

In order to fulfill the symmetry requirements on the wave function (anti-) commutation rules have to be introduced:

$$\begin{aligned}\left\{\hat{f}(\vec{r}), \hat{f}^\dagger(\vec{r}\,')\right\} := \hat{f}(\vec{r})\hat{f}^\dagger(\vec{r}\,') + \hat{f}^\dagger(\vec{r}\,')\hat{f}(\vec{r}) &= \delta(\vec{r} - \vec{r}\,') \\ \left\{\hat{f}^\dagger(\vec{r}), \hat{f}^\dagger(\vec{r}\,')\right\} &= 0 \\ \left\{\hat{f}(\vec{r}), \hat{f}(\vec{r}\,')\right\} &= 0 \ .\end{aligned}$$

1.5.2 Discrete annihilation and creation operators

A specific basis function set $\xi_i(\vec{r})$ can be introduced to transform to discrete annihilation and creation operators $\hat{a}_i$ and $\hat{a}_i^\dagger$:

$$\begin{aligned}\hat{f}(\vec{r}) &= \sum_i \hat{a}_i \xi_i(\vec{r}) \\ \hat{f}^\dagger(\vec{r}) &= \sum_i \hat{a}_i^\dagger \xi_i^*(\vec{r}) \ .\end{aligned}$$

The discrete creation operator $\hat{a}_i$ can be imagined as creating a particle with wave function $\xi_i(\vec{r})$. For strongly correlated materials it describes a localized wave function. This picture has to be used primarily for e.g. f orbitals of transition metals, which hold locally bound electrons.

The particle creation and annihilation operators $\hat{a}_i$, $\hat{a}_i^\dagger$ inherit the commutation rules from the field operators:

$$\begin{aligned}\{a_i^\dagger, a_j^\dagger\} &= 0 \\ \{a_i, a_j\} &= 0 \\ \{a_i, a_j^\dagger\} &= \delta_{i,j} \ .\end{aligned}$$

1.5.3 The occupation number basis

The concept of indistinguishability leads to the possibility of defining the state of a quantum particle system merely by the number of particles at each site in space. These sites may correspond to a physical crystal lattice.

So a quantum basis state of the system looks like

$$|\phi\rangle = |n_1, n_2, n_3, ...n_L\rangle \ .$$

Here, n_i denotes the number of particles at site i and L is the total number of sites.

In the case of fermions the anti-symmetric property of the wave function forbids the location of more than one particle per site. So only two values for a specific site are valid: $n_i = 0$ or $n_i = 1$.

In general, annihilation and creation operators modify the number of electrons at each site.

$$\begin{aligned}\hat{a}^\dagger |n\rangle &= \sqrt{n+1}\, |n+1\rangle \\ \hat{a}\, |n\rangle &= \sqrt{n}\, |n-1\rangle \ .\end{aligned}$$

The creation operator $\hat{a}^\dagger$ switches from a state with n particles to a state with $n+1$ particles and therefore creates a particle, while the annihilation operator destroys a particle.

This allows to represent any state of the quantum system in terms of the operators:

$$|\phi\rangle = (\hat{a}_1^\dagger)^{n_1}(\hat{a}_2^\dagger)^{n_2}...(\hat{a}_L^\dagger)^{n_L}\, |0\rangle \ ,$$

where $|0\rangle$ is the so-called *vacuum state* without any particle.

Using these operators another operator can be introduced which counts the number of particles at a site.

$$\hat{n} = \hat{a}^\dagger \hat{a}$$

$$\hat{n}\, |n\rangle = \hat{a}^\dagger \hat{a}\, |n\rangle = \hat{a}^\dagger \sqrt{n}\, |n-1\rangle = \sqrt{n}\sqrt{n}\, |n\rangle = n\, |n\rangle \ .$$

One can easily see that each occupation number basis state is an eigenstate of the particle number operator with eigenvalue n.

1.5.4 Transformation of operators

To transform an operator specified within the framework of first quantization to second quantization one has to distinguish depending on how many particles it is acting on [27].

1.5.4.1 Transformation of single particle operators

An single particle operator $\hat{O}$ can be transformed using:

$$\sum_{ij} \langle i|\, \hat{O}\, |j\rangle\, \hat{a}_i^\dagger \hat{a}_j \ ,$$

where $\{|i\rangle\}$ are the single particle basis states and $\hat{O}$ is the operator in first quantization representation.

1.5.4.2 Transformation of two particle operators

An operator acting on two particles (e.g. an interaction potential) is transformed like:

$$\sum_{ijmn} \langle i,m| \hat{O}_2 |j,n\rangle \, \hat{a}_i^\dagger \hat{a}_m^\dagger \hat{a}_n \hat{a}_j \ ,$$

where $|i,m\rangle$ expresses, that the first particle is in single particle state $|i\rangle$ and the second in state $|m\rangle$.

1.5.5 General Hamiltonian in 2nd quantization

In the most general case, a many-body Hamiltonian consisting of single and two particle operators has the form

$$H = \sum_{ij} t_{ij} \hat{a}_i^\dagger \hat{a}_j + \sum_{qrst} V_{qrst} \hat{a}_q^\dagger \hat{a}_r^\dagger \hat{a}_t \hat{a}_s \ . \tag{1.5}$$

Here, the first term represents the kinetic part of the Hamiltonian. In the discrete case this is reflected by particles hopping from site to site. t_{ij} is the hopping-integral which is calculated by $t_{ij} = \langle i| \hat{H}_{\text{kin.}} |j\rangle$. The potential term is quantified by the manifold $V_{qrst} = \langle q,r| \hat{H}_{\text{pot.}} |s,t\rangle$ which holds information over the interaction between the particles. Usually, the dimensionality of this quantity is drastically reduced by the introduction of model Hamiltonians.

1.6 The Hubbard model

The operators introduced in the previous section can be used to define a basic model [28, 29] of a solid body, which though looking very simple describes a vast number of quantum effects. At the same time numerical treatment is very difficult.
The Hubbard model additionally introduces a second spin species. The two species are denoted *spin-up* (↑) and *spin-down* (↓).

1.6.1 The extended one-band Hubbard model

Starting from the generic Hamiltonian (eq. 1.5) one can derive an effective model by introducing some simplifications. A one-banded model restricts the number of orbitals per site to one. It is assumed that the orbitals are translational invariant and the hopping integral t_{ij} becomes a single number t for nearest-neighbor hoppings. Hoppings to sites further away are neglected.

The interaction is restricted to an *on-site* Coulomb interaction described by the parameter U and an interaction of nearest-neighbors described by V.

$$H = -t \sum_{\langle ij \rangle \sigma} \hat{a}^{\dagger}_{i\sigma} \hat{a}_{j\sigma} + V \sum_{\langle ij \rangle \sigma\sigma'} \hat{n}_{i\sigma} \hat{n}_{j\sigma'} + U \sum_{i} \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \ .$$

The first (*kinetic*) term describes the hopping of fermions from one site to a neighboring one. The second (*potential*) term gives an additional cost of energy, if two electrons occupy neighboring sites. The third (*potential*) term punishes two particles with different spins occupying the same site. The sum over indices in triangular brackets denotes summation over all neighboring site pairs, σ is the spin degree of freedom ($\uparrow$,$\downarrow$).

The short-ranged potentials can be physically justified by considering *screening*. The repulsion does not exhibit power law like behavior as usual Coulomb interactions, but due to the presence of other charges in the crystal the potential is screened and suppressed exponentially with distance.

1.6.2 The one-band Hubbard model

The interactions are restricted to on-site interactions:

$$H = -t \sum_{<i,j>\sigma} \hat{a}^{\dagger}_{i\sigma} \hat{a}_{j\sigma} + U \sum_{i} \hat{n}_{i\uparrow} \hat{n}_{i\downarrow} \ . \tag{1.6}$$

1.6.3 Particle filling

The Hubbard Hamiltonian conserves the number of particles per spin species in a system. This means that the Hamiltonian of a grand-canonical ensemble breaks down into sectors of different particle numbers, which can be calculated independently.

In physics, the most important case is that of half-filling. In this case a Hubbard-type system of L sites is occupied by $N_{\uparrow} = L/2$ spin-up and $N_{\downarrow} = L/2$ spin-down particles. At half-filling the Hubbard model exhibits particle-hole symmetry.

Generally, the size of the occupation number basis is

$$\binom{L}{N} \ ,$$

which is the binomial coefficient and corresponds the number of possibilities of distributing N indistinguishable particles on L sites.
For a system with two spin species the size becomes

$$\binom{L}{N_\uparrow} \cdot \binom{L}{N_\downarrow} .$$

Note that at half-filling these factors grow exponentially with the system size L.

1.7 Spin-less fermions with nearest neighbor interaction

A system of particles of only one spin-species is given by

$$H = -t \sum_{<i,j>} \hat{a}_i^\dagger \hat{a}_j + \frac{V}{2} \sum_{<i,j>} \hat{n}_i \hat{n}_j . \tag{1.7}$$

The Hamiltonian introduces additionally to the aforementioned hopping term a potential, which applies for system states where two or more particles are located next to each other. In the case $V > 0$ this corresponds to a repulsion between the particles.
In the case of half-filling a large repulsion term leads to a ground state where crystal sites are alternatingly occupied and empty.

1.7.1 Mapping to a spin system

For a one-dimensional system the Hamiltonian looks like

$$H = -t \sum_i (\hat{a}_i^\dagger \hat{a}_{i+1} + \hat{a}_{i+1}^\dagger \hat{a}_i) + V \sum_i \hat{n}_i \hat{n}_{i+1} .$$

It can be mapped to a pure spin system by

$$\begin{aligned} a_i^\dagger &\longrightarrow S_i^+ (-1)^{\tilde{n}_i} \\ a_i &\longrightarrow (-1)^{\tilde{n}_i} S_i^- \\ a_i^\dagger a_i &\longrightarrow S^z + \frac{1}{2} . \end{aligned}$$

This is called the Jordan-Wigner transformation [30]. The sign is necessary to account for the anti-commutation rules of the fermionic operators. $\tilde{n}_i$ counts the number of particles to the left side of site i.

It leads to a XXZ-Hamiltonian of a spin-$\frac{1}{2}$ system:

$$H_{XXZ} = \sum_i \left(J_{xy}(S^x_{i+1}S^x_i + S^y_{i+1}S^y_i) + J_z S^z_{i+1}S^z_i \right) \ ,$$

with

$$\begin{aligned} t &\hat{=} \frac{J_{xy}}{2} \\ V &\hat{=} J_z \ . \end{aligned}$$

1.7.2 Phase diagram

The phase diagram of such spin systems is well known and it exhibits multiple phase transitions: [30, p. 161]

$$\frac{J_z}{J_{xy}} \begin{cases} < -1 & \text{Ising Ferromagnet} \\ = -1 & \text{Heisenberg Ferromagnet} \\ > -1, < 1 & \text{XY} \\ = 1 & \text{Heisenberg Anti-Ferromagnet} \\ > 1 & \text{Ising Anti-Ferromagnet} \end{cases}$$

Thus, a system of spin-less particles with no (small) nearest neighbor interaction corresponds to a spin chain in the XY-phase. We can further deduct, that the system undergoes a phase transition at

$$\frac{V}{t} = 2\frac{J_z}{J_{xy}} = 2$$

from a free-fermion state to a density-wave driven state, where the sites are mostly alternatingly occupied and unoccupied. This corresponds to an anti-ferromagnet in the spin picture and it holds for systems at half-filling and a chemical potential $\mu = 0$. In the spin picture a magnetic field h corresponds to μ in the particle picture.

1.8 The Green's function

The Green's function plays an important role for analysis of dynamical processes of many-body systems. Generally, the one-particle retarded Green's function is defined as [31]

$$G^{\text{ret.}}_{i,j}(t,t') = -i\Theta(t-t')\left\langle \left[\hat{a}_i(t), \hat{a}^\dagger_j(t')\right]_\epsilon \right\rangle \ .$$

In the case of fermions $[.,.]_\epsilon$ denotes the anti-commutator, the angle brackets $\langle . \rangle$ denote the average evaluated in the ground state: $\langle . \rangle \equiv \langle \Psi_0 | \, . \, | \Psi_0 \rangle$.

The Green's function is closely related to the spectral function $S_{i,j}(E)$ which gives immediate insight into the possible excitations of the system. The relation of $S_{i,j}(E)$ and the fourrier-transformed Green's function reads

$$G^{\text{ret.}}_{i,j}(E) = \int_{-\infty}^{\infty} d\omega \frac{S_{i,j}(\omega)}{E-\omega+i0^+} \ .$$

The transformed Green's function in energy-space is given by

$$G^{\text{ret.}}_{i,j}(E) = \langle \Psi_0 | \, \hat{a}_i \frac{1}{E-(\hat{H}-E_0)+i0^+} \hat{a}^\dagger_j \, | \Psi_0 \rangle + \langle \Psi_0 | \, \hat{a}^\dagger_j \frac{1}{E+(\hat{H}-E_0)+i0^+} \hat{a}_i \, | \Psi_0 \rangle \ .$$

Here, the Fourrier integral has been calculated, the anti-commutator and the ground state $|\Psi_0\rangle$ have been written down explicitly.

2 Two-Subsystem Ground State Approximation

2.1 Introduction and motivation

The method presented in the following sections is inspired by the Automated Multilevel Sub-structuring Method [32, 33]. For details about this method, see appendix B. It is a successful method in the field of linear elastodynamics and it relies on partitioning the physical space into smaller pieces wherein the eigenproblem is solved separately as a starting point for the full system eigenproblem calculation. Here a similar technique on the occupation number state space of strongly-correlated systems is applied.

2.2 Partitioning

2.2.1 Occupation number basis

We restrict the discussion to an even amount of total particles N on an even amount of sites L. An extension to odd numbers is straight forward.

Consider as an example a restriction on the occupation number basis which only includes the system configurations listed below. The total physical system is divided into two equally sized subsystems A and B.

- $\frac{N}{2}$ particles on subsystem A, $\frac{N}{2}$ particles on subsystem B
- $\frac{N}{2} - 1$ particles on subsystem A, $\frac{N}{2} + 1$ particles on subsystem B
- $\frac{N}{2} + 1$ particles on subsystem A, $\frac{N}{2} - 1$ particles on subsystem B

See also fig. 2.1 for a graphical representation. This scheme can in principle be expanded to further numbers of particles *away* from equilibrium. So the number l as a measure

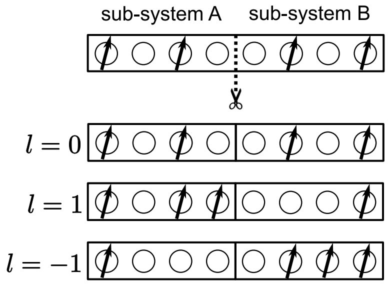

Figure 2.1: Representatives of basis states in different TSGSA partitions.

of the imbalance can be introduced. Then, the particles numbers in the two subsystems can be expressed as

$$N_A = \frac{N}{2} + l \qquad N_B = \frac{N}{2} - l$$

In principle this could be expanded to a maximum value $l^* = N/2$ with $l \in \{-l^*, ...l^*\}$. In this case all basis states would be included.

The particles can hop from one sub-system to the other and therefore all particle occupations from a totally empty to a totally filled sub-system are conceivable. However, the most probable is the even distribution with an equal amount of particles in both sub-systems ($N_A = N_B = N/2$).

2.2.2 Hubbard model

A possible partitioning for the Hubbard model is shown in tab. 2.1. Here, both spin species are partitioned separately.

The first partition consists again of two subsystems with equal particle numbers. The next class of partitions can be obtained by letting either a spin-up particle or a spin-down particle hop. In this case, there are four equivalent partitions of this type instead of two for the spin-less case.

A partition $(l_\uparrow, l_\downarrow)$ represents a sub-system configuration with

$$\begin{aligned} N_{A,\uparrow} &= \frac{N_\uparrow}{2} + l_\uparrow \qquad & N_{A,\downarrow} &= \frac{N_\downarrow}{2} + l_\downarrow \\ N_{B,\uparrow} &= \frac{N_\uparrow}{2} - l_\uparrow \qquad & N_{B,\downarrow} &= \frac{N_\downarrow}{2} - l_\downarrow \ . \end{aligned}$$

	subsystem 1		subsystem 2	
partition	#spin-up	#spin-down	#spin-up	#spin-down
0	$N_\uparrow/2$	$N_\downarrow/2$	$N_\uparrow/2$	$N_\downarrow/2$
1	$N_\uparrow/2-1$	$N_\downarrow/2$	$N_\uparrow/2+1$	$N_\downarrow/2$
2	$N_\uparrow/2+1$	$N_\downarrow/2$	$N_\uparrow/2-1$	$N_\downarrow/2$
3	$N_\uparrow/2$	$N_\downarrow/2-1$	$N_\uparrow/2$	$N_\downarrow/2+1$
4	$N_\uparrow/2$	$N_\downarrow/2+1$	$N_\uparrow/2$	$N_\downarrow/2-1$
		$\vdots$		

Table 2.1: A possible TSGSA partitioning of the Hubbard model.

2.2.3 Hamiltonian

In general a Hamiltonian of a systems consisting of two sub-system reads

$$\hat{H} = \hat{H}_A + \hat{H}_B + \hat{H}_{AB} \ .$$

Here, $\hat{H}_A$ and $\hat{H}_B$ include the interior hopping and potential terms of the sub-systems A and B, respectively. $\hat{H}_{AB}$ contains both the hopping over the sub-system boundary and possibly a nearest neighbor potential.

For spin-less fermions the parts of the Hamiltonian are:

$$\begin{aligned} \hat{H}_X &= -t \sum_{\langle i,j \rangle} \hat{a}^\dagger_{X,i} \hat{a}_{X,j} + V \sum_{\langle i,j \rangle} \hat{n}_{X,i} \hat{n}_{X,j} \ , \quad X \in \{A, B\} \\ \hat{H}_{AB} &= -t {\sum_{i,j}}' \left(\hat{a}^\dagger_{A,i} \hat{a}_{B,j} + \hat{a}^\dagger_{B,j} \hat{a}_{A,i} \right) + V {\sum_{i,j}}' \hat{n}_{A,i} \hat{n}_{B,j} \ , \end{aligned}$$

where $\sum'$ indicates that the indices i, j have to be chosen such that they describe opposite sites on the boundary.

For the Hubbard model the parts are:

$$\begin{aligned} \hat{H}_X &= -t \sum_{\langle i,j \rangle, \sigma} \hat{a}^\dagger_{X,i,\sigma} \hat{a}_{X,j,\sigma} + U \sum_{i} \hat{n}_{X,i\uparrow} \hat{n}_{X,i\downarrow} \ , \ X \in \{A, B\} \\ \hat{H}_{AB} &= -t {\sum_{i,j,\sigma}}' \left(\hat{a}^\dagger_{A,i,\sigma} \hat{a}_{B,j,\sigma} + \hat{a}^\dagger_{B,j,\sigma} \hat{a}_{A,i,\sigma} \right) \ . \end{aligned}$$

As the Hamiltonians conserve the total number of particles the calculations can be split into sectors of fixed number of particles. This means that the particle numbers in the sub-systems are subjects to the constraint

$$N_A + N_B = N \ ,$$

or in the case of two spin species

$$N_{A,\sigma} + N_{B,\sigma} = N_\sigma \ , \quad \sigma \in \{\uparrow, \downarrow\} \ .$$

2.2.4 Combination of sub-systems

Fig. 2.2 shows a diagrammatic view of the different sub-system configurations. Note that the elementary insertion or removal of a particle can only lead from one configuration to an adjacent one.

When combining two subsystem configurations to build a partition, the systems mirrored at the origin have to be used (see fig. 2.3). For an odd number of particles one has to combine the subsystems as pictured in fig. 2.4.

2.2.5 Hamilton matrix structure

Let's stick for the moment with a three-partitioned system ($l^* = 1$). The truncated Hamilton matrix of the system looks like:

$$H_t = \begin{pmatrix} H^{(0,0)} & H^{(0,0)}_{(-1,+1)} & H^{(0,0)}_{(+1,-1)} \\ \left(H^{(0,0)}_{(-1,+1)}\right)^\dagger & H^{(-1,+1)} & 0 \\ \left(H^{(0,0)}_{(+1,-1)}\right)^\dagger & 0 & H^{(+1,-1)} \end{pmatrix} , \tag{2.1}$$

where $H^{(0,0)}$, $H^{(-1,+1)}$ and $H^{(+1,-1)}$ are the *interior* Hamiltonians for the partitions listed above, respectively. The indices of $H^{(l,-l)}$ denote the particle number difference from $N/2$ in the subsystems A and B. In these parts, all hopping processes are included, which do not change the particle numbers in the two subsystems. By contrast, $H^{(0,0)}_{(-1,+1)}$ and $H^{(0,0)}_{(+1,-1)}$ contain the hoppings over the interface of the two subsystems A and B. Note that there is no possible hopping which connects the partitions $(-1, +1)$ and $(+1, -1)$ because two elementary processes would be needed.

Fig. 2.5 and fig. 2.6 show graphical representations of the Hamiltonians for multiple partitions with the ordering explained above.

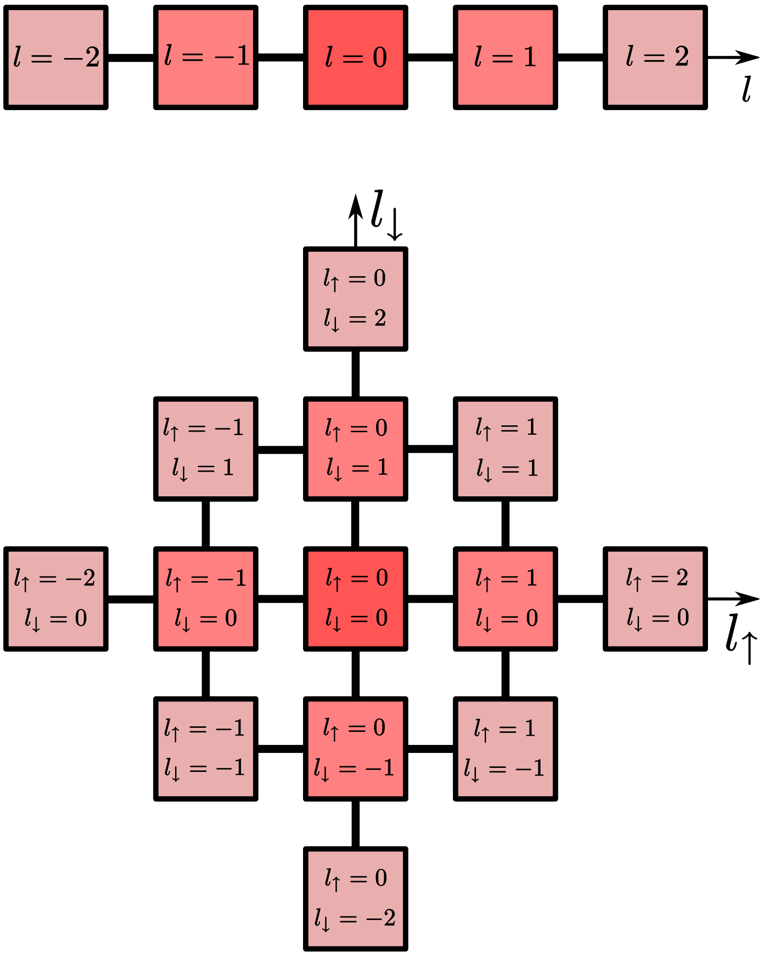
$l = -2$
$l = -1$
$l = 0$
$l = 1$
$l = 2$
l
$l_\downarrow$
$l_\uparrow = 0$ $l_\downarrow = 2$
$l_\uparrow = -1$ $l_\downarrow = 1$
$l_\uparrow = 0$ $l_\downarrow = 1$
$l_\uparrow = 1$ $l_\downarrow = 1$
$l_\uparrow = -2$ $l_\downarrow = 0$
$l_\uparrow = -1$ $l_\downarrow = 0$
$l_\uparrow = 0$ $l_\downarrow = 0$
$l_\uparrow = 1$ $l_\downarrow = 0$
$l_\uparrow = 2$ $l_\downarrow = 0$
$l_\uparrow$
$l_\uparrow = -1$ $l_\downarrow = -1$
$l_\uparrow = 0$ $l_\downarrow = -1$
$l_\uparrow = 1$ $l_\downarrow = -1$
$l_\uparrow = 0$ $l_\downarrow = -2$

Figure 2.2: Diagrammatic view of the different subsystems and their connections for the spin-less fermion system (above) and the Hubbard system (below).

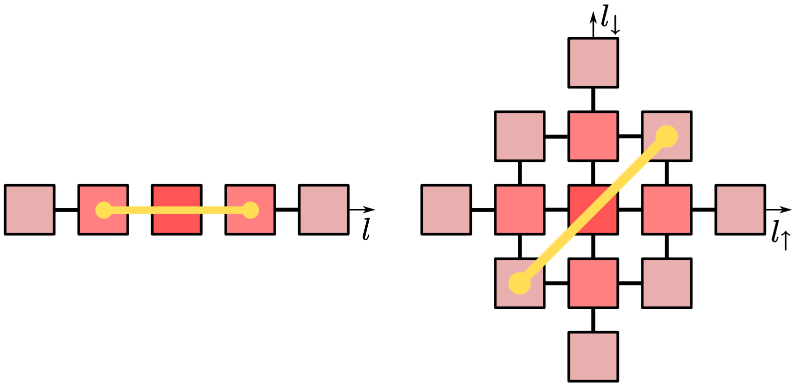

Figure 2.3: Combining subsystems to build partitions for the spin-less fermion system (left) and the Hubbard system (right).

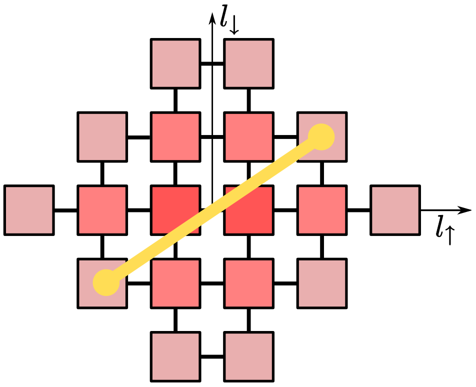

Figure 2.4: Combining subsystems to build partitions for the Hubbard system with an odd amount of spin-up particles.

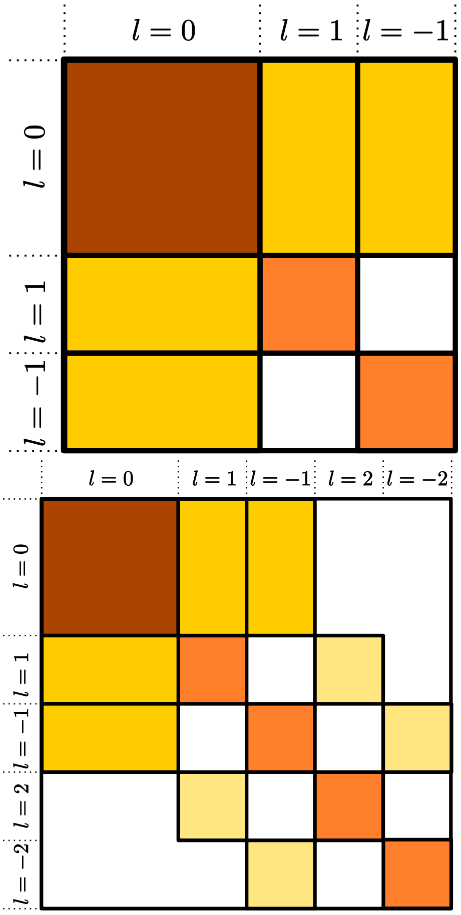

Figure 2.5: The TSGSA matrix for a system of spin-less fermions including partitions with one particle off balance (left) or two particles off balance (right). l denotes the number of particles away from equilibrium.

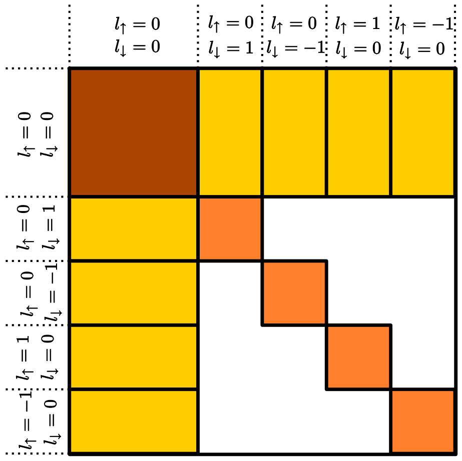
$l_\uparrow = 0$ $l_\downarrow = 0$
$l_\uparrow = 0$ $l_\downarrow = 1$
$l_\uparrow = 0$ $l_\downarrow = -1$
$l_\uparrow = 1$ $l_\downarrow = 0$
$l_\uparrow = -1$ $l_\downarrow = 0$
$l_\uparrow = 0$ $l_\downarrow = 0$
$l_\uparrow = 0$ $l_\downarrow = 1$
$l_\uparrow = 0$ $l_\downarrow = -1$
$l_\uparrow = 1$ $l_\downarrow = 0$
$l_\uparrow = -1$ $l_\downarrow = 0$

Figure 2.6: The TSGSA matrix for a Hubbard-type system including partitions with 1 particle off balance. $l_\uparrow$ and $l_\downarrow$ denote the number of spin-up and spin-down particles off-equilibrium, respectively.

In general the basis vectors in partition $(+l, -l)$ are tensor products of basis vectors of the individual sub-systems A and B:

$$|v_\nu^{(l,-l)}\rangle = |v_{\nu_A}^{(l)}\rangle_A \otimes |v_{\nu_B}^{(-l)}\rangle_B \ ,$$

where $\nu = (\nu_A, \nu_B)$ is the index pair for the basis vectors of the total system, while ν_X enumerates the basis vectors of sub-system X. The outer index indicates the sub-system the basis vectors belong to. In terms of quantum mechanical creation and annihilation operators, the basis states are constructed by:

$$|v_{\nu_X}^{(l)}\rangle_X = \prod_{i=1}^{L_X} \left(a_{X,i}^\dagger\right)^{n_\nu(X,i)} |0\rangle_X \ , \quad \sum_i n_\nu(X,i) = N_X \pm l \ , \quad n_\nu(X,i) \in \{0,1\} \ .$$

Here $a_{X,i}^\dagger$ creates a particle in sub-system $X \in \{A, B\}$ at site i, and $|0\rangle_X$ is the vacuum state of the corresponding sub-system.

2.3 Hamiltonian transformation

Consider the three-partition Hamiltonian of type 2.1

$$H = \begin{pmatrix} H_1 & H_{1,2} & H_{1,3} \\ H_{1,2}^\dagger & H_2 & 0 \\ H_{1,3}^\dagger & 0 & H_3 \end{pmatrix} ,$$

and rotate the full Hamilton matrix by the following orthogonal matrix

$$V = \begin{pmatrix} V_1 & & \\ & V_2 & \\ & & V_3 \end{pmatrix} ,$$

which contains the eigenvectors of each individual partition:

$$\begin{aligned} H_1 V_1 &= V_1 D_1 \\ H_2 V_2 &= V_2 D_2 \\ H_3 V_3 &= V_3 D_3 \, . \end{aligned}$$

Here, D_1, D_2 and D_3 are diagonal matrices with the partition eigenvalues as their entries. The transformed Hamiltonian reads

$$H_r = V^\dagger H V = \begin{pmatrix} D_1 & B_1 & B_2 \\ B_1^\dagger & D_2 & 0 \\ B_2^\dagger & 0 & D_3 \end{pmatrix} , \tag{2.2}$$

with $B_1 = V_1^\dagger H_{1,2} V_2$ and $B_2 = V_1^\dagger H_{1,3} V_3$.

By using only a small number of eigenstates of the individual partitions one arrives at a much smaller matrix H_r compared to the original Hamiltonian. In this case the V_i become rectangular. They are not unitary any more, but $V_i^\dagger V_i = I$ still holds.

In terms of the description used in quantum mechanics, we first solve the individual decoupled eigensystems for different partitions $|l| \le l^*$:

$$\hat{H}^{(l,l)} \left|\psi_\nu^{(l)}\right\rangle = E_\nu^{(l)} \left|\psi_\nu^{(l)}\right\rangle \, .$$

From the calculated eigenvectors only a small *cropping number* is kept. Subsequently the approximation to the total system ground-state is constructed using a linear combination of the kept eigenvectors of the selected partitions:

$$|\Psi_0^{(l^*)}\rangle = \sum_{l=-l^*}^{l^*} \sum_{\nu}^{\text{crop.}} C_{l,\nu}^{(l^*)} \, |\psi_\nu^{(l)}\rangle \ .$$

For the Hubbard system partitions up to $(l_\uparrow^*, l_\downarrow^*)$ are included using a Manhattan distance measure, e.g. $(l_\uparrow, l_\downarrow) \in \{(0,0),(1,0),(-1,0),(0,1),(0,-1)\}$ for $l_\uparrow^* = 1$, $l_\downarrow^* = 1$.
The coefficients $C_{l,\nu}^{(l^*)}$ are given by the eigensolution of the matrix

$$\tilde{H}_{\nu',\nu}^{(l',l)} = \langle\psi_{\nu'}^{(l')}|\hat{H}|\psi_\nu^{(l)}\rangle \ ,$$

using the partition eigensolutions $|\psi_\nu^{(l)}\rangle$. This corresponds to matrix 2.2.

2.4 Solutions for partitions

For solving the eigenvalue problem for each partition the hopping of the subsystem boundaries is turned off. So it suffices to evaluate the eigenstates of the smaller systems and build the eigenvectors and eigenvalues of the the larger system by a tensor product evaluation.

2.4.1 Simple tensorial system

Let H_A and H_B be the Hamiltonians of two subsystem A and B, respectively. All further indices indicating the partition have been dropped temporarily. A Hamiltonian describing the tensorial system reads:

$$H = H_A \otimes I_B + I_A \otimes H_B \ , \tag{2.3}$$

with I_X the identity matrix of corresponding size. The solution for the eigenvalue equation can be obtained by using the eigensolutions of the two subsystems:

$$\begin{aligned} HV &= (H_A \otimes I_B + I_A \otimes H_B)(V_A \otimes V_B) \\ &= (H_A V_A) \otimes V_B + V_A \otimes (H_B V_B) \\ &= (V_A D_A) \otimes V_B + V_A \otimes (V_B D_B) \\ &= (V_A \otimes V_B)(D_A \otimes I_B + I_A \otimes D_B) \\ &= VD \ . \end{aligned}$$

The eigenvalues can be calculated by summing up the eigenvalues of the subsystems in all possible combinations. The eigenvectors are given by the Kronecker-product of the subsystem eigenvectors.
The matrix elements of H have the form:

$$\langle v^{(l)}_{\nu'} |\, \hat{H} \,| v^{(l)}_{\nu} \rangle = \langle v^{(l)}_{\nu'_A} | \hat{H}_A | v^{(l)}_{\nu_A} \rangle_A \, \delta_{\nu'_B,\nu_B} + \delta_{\nu'_A,\nu_A} \, \langle v^{(-l)}_{\nu'_B} | \hat{H}_B | v^{(-l)}_{\nu_B} \rangle_B \ .$$

Let the eigenvalue problem of the two sub-systems be given by

$$\hat{H}_X \, | \Psi^{(l)}_{\nu_X} \rangle_X = E^{(l)}_{\nu_X} \, | \Psi^{(l)}_{\nu_X} \rangle_X \ .$$

Then, the eigensolutions of the combined system can be constructed using

$$\begin{aligned} |\psi^{(l)}_{\nu}\rangle &:= |\Psi^{(l)}_{\nu_A}\rangle_A \otimes |\Psi^{(-l)}_{\nu_B}\rangle_B \\ E^{(l)}_{\nu} &:= E^{(l)}_{\nu_A} + E^{(-l)}_{\nu_B} \ . \end{aligned}$$

2.4.2 Inter-partition hopping

For calculation of the off-diagonal elements of the matrix 2.2 one needs to determine the inter-partition overlaps.
One can write these overlaps as a sum over tensor products:

$$\mathcal{O} = \sum_{\alpha} o_{A,\alpha} \otimes o_{B,\alpha} \ .$$

α denotes different paths, where a particle can hop from one to another sub-system. The matrices $o_{X\alpha}$ correspond to the creation and annihilation operators of quantum mechanics restricted to sites at the sub-system boundary.
In terms of the operators, the matrix elements of the off-diagonal blocks with $l' = l + 1$, which describe a hopping from sub-system B to A have the form

$$\langle v^{(l')}_{\nu'} |\, (H_{AB})^{(l',l)} \,| v^{(l)}_{\nu} \rangle = -t {\sum_{i,j}}' \langle v^{(l')}_{\nu'_A} |\, \hat{a}^{\dagger}_{A,i} \,| v^{(l)}_{\nu_A} \rangle_A \, \langle v^{(-l')}_{\nu'_B} |\, \hat{a}_{B,j} \,| v^{(-l)}_{\nu_B} \rangle_B \ .$$

The sum $\sum'$ selects only indices i, j on neighboring sites across the sub-system boundary. Due to the reordering of the operators there may be an additional sign factor. The matrix

System	full system			border potentials turned off		
	E_0	E_1	E_2	E_0	E_1	E_2
$L = 4$, $N = 2$	-0.3852	-0.3852	10.0000	-2.0000	-0.3852	0.0000
$L = 6$, $N = 3$	-0.6976	-0.4844	8.0000	-2.3695	-1.1789	-1.1745
$L = 8$, $N = 4$	-0.7919	-0.7919	7.0979	-3.1678	-1.8088	-1.6899
$L = 10$, $N = 5$	-1.0015	-0.9784	6.2745	-3.0451	-2.6718	-2.2373
$L = 12$, $N = 6$	-1.1880	-1.1880	5.8613	-3.9238	-2.9039	-2.8513
$L = 14$, $N = 7$	-1.3874	-1.3846	5.418	-3.7119	-3.5851	-2.9954
$L = 16$, $N = 8$	-1.5840	-1.5840	5.1297	-4.5339	-3.7477	-3.7219
$L = 18$, $N = 9$	-1.7822	-1.7818	4.8137	-4.3372	-4.2916	-3.6677
$L = 20$, $N = 10$	-1.9800	-1.9800	4.5695	-5.0690	-4.4458	-4.4323

Table 2.2: Lowest 3 eigenvalues of a given system of spin-less fermions compared with the same system where the nearest neighbor potentials across the subsystem borders (splitting in two) have been removed (pbc., $V = 10$).

elements describing a hopping from sub-system B to A follow from the hermiticity of the total Hamiltonian.

In an 1D system with periodic boundary conditions there are two possible pairs of sites where a particle can hop from one to another subsystem. So the sum reduces to an addition of two terms. One of the summands may describe the process of a particle hopping from the second subsystem at site s_2 to the first subsystem at site s_1. In this case the matrix o_1 is the matrix representation of a creation operator at the given site s_1.

2.4.3 Nearest neighbor potential

A problem arises when looking at the nearest neighbor potential at the boundary. Simply neglecting this is not an option as can be seen in tab. 2.2. On first impression one might think, that the approximation becomes more accurate when going to higher partitions, because in 1D a only fixed number of neighbor pairs (2 with pbc.) is affected. But one should keep in mind, that at half filling, the fraction of states affected because of 2 particles sitting at the boundary is constant for arbitrary system sizes.

The potential on the subsystem boundaries can be expressed as follows:

$$U_b = V \sum_\alpha R_{A,\alpha} \otimes R_{B,\alpha} \, .$$

Here, $R_{X,\alpha}$ is a diagonal matrix with entries 1 at the indices, where the corresponding basis states are occupied with a particle at the boundary, X denotes the subsystem. The sum goes over all possible neighbor pairs across subsystem boundaries. The matrices correspond to the particle number operators of quantum mechanics. Using them the matrix elements on the diagonal can be expressed as

$$\langle v_\nu^{(l)} | \left(H_{AB}^{\mathrm{int}}\right)^{(l,l)} | v_\nu^{(l)} \rangle = V {\sum_{ij}}' \langle v_{\nu_A}^{(l)} | \hat{n}_{A,i} | v_{\nu_A}^{(l)} \rangle_A \, \langle v_{\nu_B}^{(-l)} | \hat{n}_{B,j} | v_{\nu_B}^{(-l)} \rangle_B \, .$$

Again, the sum $\sum'$ selects only indices i, j on neighboring sites across the sub-system boundary.

2.5 Basis transformation

As seen in the previous paragraphs, all involved blocks of the Hamilton matrix 2.2 have the same structure:

$$M^{(l',l)} = \sum_\alpha M_{A,\alpha}^{(l',l)} \otimes M_{B,\alpha}^{(-l',-l)} \, ,$$

with a very limited number of summands denoted by α.

The expression can be easily rotated to a different basis set given in the columns of a matrix $\tilde{V}^{(l)} = \tilde{V}_A^{(l)} \otimes \tilde{V}_B^{(-l)}$:

$$\begin{aligned} \left(\tilde{V}^{(l')}\right)^\dagger M^{(l',l)} \tilde{V}^{(l)} &= \sum_\alpha \left(\tilde{V}_A^{(l')} \otimes \tilde{V}_B^{(-l')}\right)^\dagger \left(M_{A,\alpha}^{(l',l)} \otimes M_{B,\alpha}^{(-l',-l)}\right) \left(\tilde{V}_A^{(l)} \otimes \tilde{V}_B^{(-l)}\right) \\ &= \sum_\alpha \left(\left(\tilde{V}_A^{(l')}\right)^\dagger M_{A,\alpha}^{(l',l)} \tilde{V}_A^{(l)}\right) \otimes \left(\left(\tilde{V}_B^{(-l')}\right)^\dagger M_{B,\alpha}^{(-l',-l)} \tilde{V}_B^{(-l)}\right) \end{aligned}$$

These calculations reduce the computational effort enormously, when considering a much smaller basis set introduced by $\tilde{V}^{(l)}$ which becomes highly rectangular in this case. All the calculations are done at the numerical complexity of the sub-systems. The transformed matrices $\left(\tilde{V}_X^{(l')}\right)^\dagger M_{X,\alpha}^{(l',l)} \tilde{V}_X^{(l)}$ which are comparably small are then combined to form the partition.

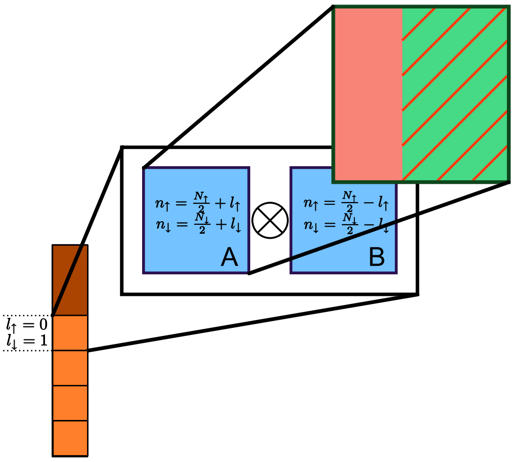
$n_\uparrow = \frac{N_\uparrow}{2} + l_\uparrow$
$n_\downarrow = \frac{N_\downarrow}{2} + l_\downarrow$
A
$\otimes$
$n_\uparrow = \frac{N_\uparrow}{2} - l_\uparrow$
$n_\downarrow = \frac{N_\downarrow}{2} - l_\downarrow$
B
$l_\uparrow = 0$
$l_\downarrow = 1$

Figure 2.7: Illustration of the construction of the TSGSA basis $\tilde{V}$. Each partition corresponds to a fixed number of spin-up and spin-down particles in both subsystems, which eigenvectors are combined tensorially (blue) to $\tilde{V}^{(l)}$. For each sub-system A and B the vectors are obtained by solving the corresponding eigenvalue problem. The eigenvectors are truncated e.g. by keeping only the vectors on the lower end of the spectrum (red).

2.5.1 Fermionic sign parity considerations

The fermionic commutation rules have to be obeyed when determining the off-diagonal Hamiltonian parts, which contain the hopping processes over the boundaries.
For each individual subsystem a parity of a particle creation/annihilation process can be determined using a simple site numbering of the subsystem. When combining this processes, a site numbering of the total system would have to be used.
In order to simplify the calculation one can think of a numbering scheme, where all sites of subsystem A are counted first, and then those of subsystem B. So, for the first subsystem the parity is the same as the subsystem parity. For processes in the second subsystem an additional sign-flip from the amount of particles in the first subsystem has to be considered.

2.6 Numerical results

2.6.1 Spin-less fermions

Tab. 2.3 shows that even ground states of large systems are well approximated by a very limited amount of partition eigenstates. Note that for this calculation each partition was calculated exactly and not using the tensor structure of the partitions. The practical application of this is limited because although the individual eigenproblems that have to be solved are smaller by a factor compared to the full problem, a comparably large number of eigenvectors has to be computed. If the tensor structure was used, the algorithm performance is increased exponentially. Nevertheless, the potential of this method can clearly be seen.
Tab. 2.4 shows a calculation taking advantage of the tensorial structure. Note that for the system with nearest neighbor interaction the tensorial combination does not give the exact partition eigensolutions. Therefore, the accuracy is worse than in the previous case.

2.6.2 Hubbard model

The Hubbard case has the nice property, that the tensorially combined eigenstates of the subsystems give the exact full system eigensolutions of the individual partitions. This provides a huge boost in terms of computational performance because the Hilbert space of the subsystems is exponentially smaller than that of the full system.

System	N_b	partitions size	$n_{\text{crop.}}$	E_{TSGSA}	E_{exact}
$L = 4$, $N = 2$	6	4/1/1	4	-0.385	-0.385
$L = 8$, $N = 4$	70	36/16/16	36	-0.792	-0.792
$L = 12$, $N = 6$	924	400/225/225	100	-1.187	-1.188
$L = 16$, $N = 8$	12870	4900/3136/3136	100	-1.582	-1.584
$L = 20$, $N = 10$	184756	63504/44100/44100	100	-1.978	-1.980

Table 2.3: Ground state energies for different systems estimated by TSGSA for spin-less fermions and a potential $V = 10$ compared with exact values. The individual partition eigensolutions were calculated exactly. N_b denotes the full basis size. For the first three partition classes, the individual size is given. $n_{\text{crop.}}$ indicates the number of eigenstates from the three partitions used for the approximation.

System	N_b	$N_b(p)$	$n_{\text{crop.}}(p)$	E_{TSGSA}	E_{exact}	$\lvert E_{\text{TSGSA}} - E_{\text{exact}} \rvert$
$L = 4$, $N = 2$	6	4/1	4/1	-0.385	-0.385	0.000
$L = 8$, $N = 4$	70	36/16	25/9	-0.755	-0.792	0.037
$L = 12$, $N = 6$	924	400/225	100/64	-1.161	-1.188	0.894
$L = 16$, $N = 8$	12870	4900/3136	100/64	-1.461	-1.584	0.123
$L = 20$, $N = 10$	184756	63504/44100	100/64	-1.817	-1.980	0.163
$L = 20$, $N = 10$	184756	63504/44100	400/225	-1.882	-1.980	0.098

Table 2.4: Ground state energies estimated by TSGSA for spin-less fermions and a potential $V = 10$ compared with exact values. $N_b(p)$ and $n_{\text{crop.}}(p)$ denote the full size and number taken for the approximation of each partition.

System	N_b	$N_b(p)$	$E_{\text{2-part.}}$	E_{exact}	$\lvert E_{\text{2-part.}} - E_{\text{exact}} \rvert$
$L = 4$, $N_\uparrow = N_\downarrow = 2$	36	16/4	-0.969	-1.100	0.131
$L = 8$, $N_\uparrow = N_\downarrow = 4$	4900	1296/576	-2.055	-2.177	0.122
$L = 12$, $N_\uparrow = N_\downarrow = 6$	853776	160000/90000	-3.116	-3.232	0.117

Table 2.5: Principle accuracy of the two-partition approximation (exact values) for the Hubbard model. N_b is the total basis size, $N_b(p)$ is the partition size, when including only the first two partition classes.

System	N_b	$N_b(p)$	$n_{\text{crop.}}$	E_{TSGSA}	E_{exact}
$L = 4$, $N_\uparrow = N_\downarrow = 2$	36	16/4	32	-0.969	-1.100
$L = 8$, $N_\uparrow = N_\downarrow = 4$	4900	1296/576	100	-2.053	-2.177
$L = 12$, $N_\uparrow = N_\downarrow = 6$	853776	160000/90000	100	-2.931	-3.232

Table 2.6: Ground state energies estimated by TSGSA for the Hubbard model and a potential $U = 10$ compared with exact values. For all partitions the same amount of lowest eigenstates $n_{\text{crop.}}$ was used.

Tab. 2.5 shows the accuracy of the approximation of a Hubbard system by only including the basis states with balanced particle numbers in both subsystems ($l_\uparrow = l_\downarrow = 0$) and additionally the partitions with one particle off-equilibrium ($|l_\uparrow| + |l_\downarrow| = 1$). Note that the accuracy increases with increasing system sizes. This can be understood by considering that the primary partition gains in weight for larger systems, as an imbalance in term of particle occupation becomes statistically less likely.

Tab. 2.6 shows results obtained by the method for a few small systems compared with exact diagonalization obtained by the standard ARPACK routine. Note that a very small fraction of up to 100 basis states was used.

Tab. 2.7 and 2.8 show the development of a two-partitioned approximation to a specific system when including successively more subsystem basis states. The fast convergence to the two partition solution can easily be seen. To improve the approximation of the exact solution, further partitions would have to be included.

Fig. 2.8 to 2.11 give a graphical representation of the accuracy dependent on the subsystem cropping numbers.

$n_{\text{crop.}}(p)$	E_{TSGSA}	$\lvert E_{\text{TSGSA}} - E_{\text{2-part.}}\rvert$	$\lvert E_{\text{TSGSA}} - E_{\text{exact}}\rvert$
2/2	-1.8308	0.2244	0.3459
10/10	-1.9339	0.1213	0.2428
100/100	-2.0530	0.0021	0.1236
500/200	-2.0550	0.0002	0.1217
1024/225	-2.0552	0.0000	0.1215
1296/576	-2.0552	0.0000	0.1215

Table 2.7: Ground state accuracy of a Hubbard system ($L = 8$, $N_\uparrow = 4$, $N_\downarrow = 4$, $U = 10$) for different croppings $n_{\text{crop.}}(p)$. The exact ground state energy of this system is $E_{\text{exact}} = -2.1767$, for the exact two-partition system: $E_{\text{2-part.}} = -2.0552$. The accuracy of estimation to the exact ground state are given as well as the accuracy of the full two-partition system.

$n_{\text{crop.}}(p)$	E_{TSGSA}	$\lvert E_{\text{TSGSA}} - E_{\text{2-part.}}\rvert$	$\lvert E_{\text{TSGSA}} - E_{\text{exact}}\rvert$
2/2	-2.8876	0.2278	0.3447
4/4	-2.8878	0.2277	0.3446
6/6	-2.9215	0.1940	0.3109
10/10	-2.9314	0.1841	0.3010
20/20	-2.9991	0.1164	0.2333
30/30	-3.0099	0.1056	0.2225
40/40	-3.1036	0.0119	0.1288
60/60	-3.1146	0.0009	0.1178
80/80	-3.1146	0.0008	0.1177
80/20	-2.9998	0.1157	0.2326
80/40	-3.1042	0.0113	0.1282
80/60	-3.1146	0.0008	0.1177

Table 2.8: Ground state accuracy of a Hubbard model ($L = 12$, $N_\uparrow = 6$, $N_\downarrow = 6$, $U = 10$) for different croppings $n_{\text{crop.}}(p)$. The exact ground state energy of this system is $E_{\text{exact}} = -3.2324$, for the exact two-partition system: $E_{2-part.} = -3.1155$. The accuracy of estimation to the exact ground state are given as well as the accuracy of the full two-partition system.

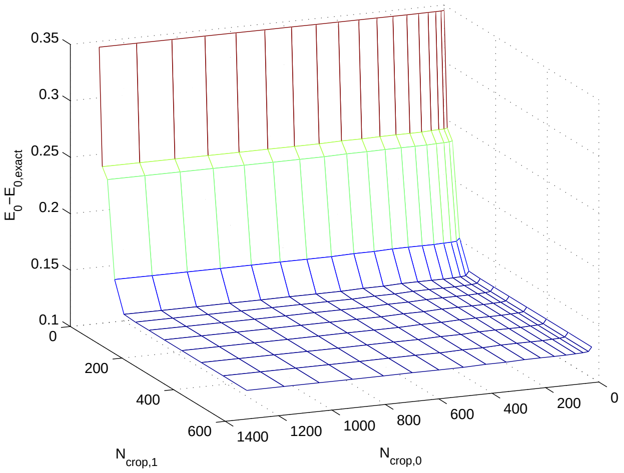

Figure 2.8: Accuracy of TSGSA ground state approximation vs. cropping. $N_{\mathrm{crop},0}$ refers to number of states included from partition with equal particles in both systems, $N_{\mathrm{crop},1}$ to partition with particle out of equilibrium. System parameters: $L = 8$, $N_\uparrow = 4$, $N_\downarrow = 4$, $U = 10$.

2.6.2.1 Higher partitions

Tab. 2.9 and 2.10 show the effect on the result of a specific Hubbard system when including higher partitions to the problem. It can be seen clearly, that introducing additional partitions increases the accuracy. Note that including further partitions increases the numerical effort only linearly up to the final eigenvalue calculation.

2.6.2.2 Extension to multiple dimensions

Extension to multiple dimensions is pretty straight forward, the system is simply split into two parts perpendicular to the first spatial axis. Tab. 2.11 shows the influence of the geometry of systems with equal numbers of sites on the approximation result. It can be seen that shapes with more possible particle exchange paths between the two subsystems give poorer results. This can be understood by considering, that in this case the equilibrium partition gets less weight, and the partitions off-equilibrium more, because a hopping over the boundary is statistically more probable.

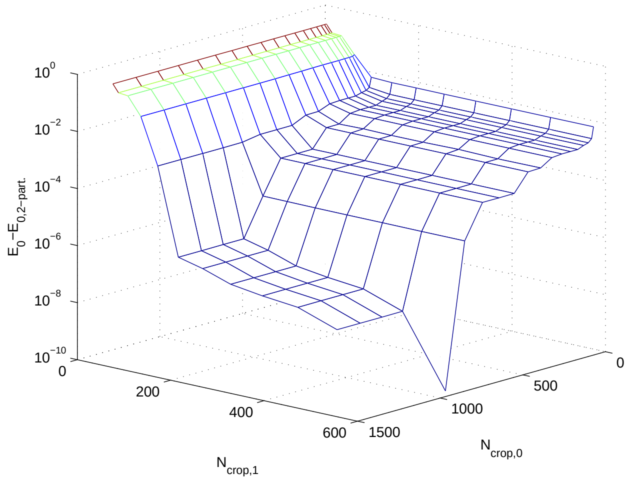

Figure 2.9: Distance of TSGSA ground state approximation from the full two-partition solution vs. cropping on logarithmic scale. $N_{\text{crop},0}$ refers to number of states included from partition with equal particles in both systems, $N_{\text{crop},1}$ to partition with one particle out of equilibrium. System parameters: $L = 8$, $N_\uparrow = 4$, $N_\downarrow = 4$, $U = 10$.

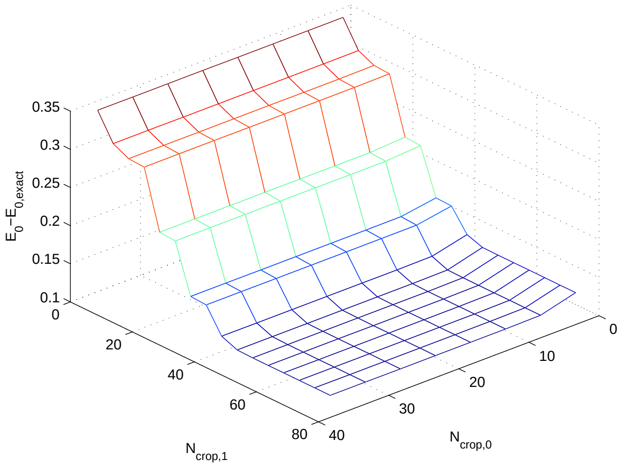

Figure 2.10: Accuracy of TSGSA ground state approximation vs. cropping. $N_{\text{crop},0}$ refers to number of states included from partition with equal particles in both systems, $N_{\text{crop},1}$ to partition with particle out of equilibrium. System parameters: $L = 12$, $N_\uparrow = 6$, $N_\downarrow = 6$, $U = 10$.

2.6.2.3 Comparison to DMRG

Tab. 2.12 shows comparisons with the DMRG method. The program implemented by Reinhard Noack was used. Note that the relative error is decreasing with system size for a constant number of eigenmodes taken into account, even if the system size is increased. This is due to the fact that with larger systems the primary partition becomes more important and hoppings over the subsystem boundary have less weight.

2.6.2.4 Same-time correlations

The same-time density-density correlations were evaluated from the ground state eigenvector and compared to the exact diagonalization results.

To calculate the values directly from the TSGSA basis, the following transformation can

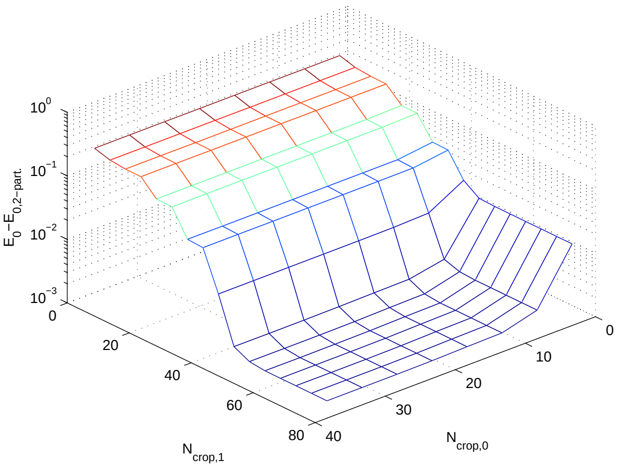

Figure 2.11: Distance of TSGSA ground state approximation from the full two-partition solution vs. cropping on logarithmic scale. $N_{\text{crop},0}$ refers to number of states included from partition with equal particles in both systems, $N_{\text{crop},1}$ to partition with one particle out of equilibrium. System parameters: $L = 12$, $N_\uparrow = 6$, $N_\downarrow = 6$, $U = 10$.

$n_{\text{crop.}}(p)$	E_{TSGSA}	$\lvert E_{\text{TSGSA}} - E_{\text{exact}}\rvert$
4/4	-1.831	0.346
16/16	-1.934	0.243
64/64	-2.027	0.150
100/100	-2.053	0.124
4/4/4	-1.831	0.346
16/16/16	-1.954	0.223
64/64/64	-2.108	0.069
100/100/100	-2.167	0.009

Table 2.9: TSGSA accuracy for a Hubbard system ($L = 8$, $N_\uparrow = N_\downarrow = 4$, $U = 10$, $N_b = 4900$) when including more than 2 partitions.

$n_{\text{crop.}}(p)$	$E_{\text{part.}}$	E_{TSGSA}	$\lvert E_{\text{TSGSA}} - E_{\text{part.}}\rvert$	$\lvert E_{\text{TSGSA}} - E_{\text{exact}}\rvert$	relative error
80/60	-3.1155	-3.1146	0.0008	0.1177	0.0361
80/60/30	-3.2302	-3.2216	0.0086	0.0108	0.0033
80/60/40		-3.2227	0.0075	0.0096	0.0030
80/60/50		-3.2249	0.0053	0.0075	0.0023
80/60/50/20	-3.2320	-3.2263	0.0057	0.0061	0.0019

Table 2.10: TSGSA accuracy for a Hubbard system ($L = 12, N_\uparrow = N_\downarrow = 6$) when including more than 2 partitions. The exact ground state energy is $E_{\text{exact}} = -3.232383$. $E_{\text{part.}}$ denotes the exact value for the corresponding partitions.

be used. The density operator reads:

$$\begin{aligned}
\langle \rho_i \rangle &= \langle \Psi_0 | \hat{n}_i | \Psi_0 \rangle \\
&= \sum_{j,k} \langle \Psi_0 | \phi_j \rangle \langle \phi_j | \hat{n}_i | \phi_k \rangle \langle \phi_k | \Psi_0 \rangle \\
&= \sum_{j} |f_j|^2 \langle \phi_j | \hat{n}_i | \phi_j \rangle \ ,
\end{aligned}$$

where Ψ_0 is the ground state, $|\phi_j\rangle$ is the TSGSA-basis and f_j are the coefficients of the ground state in the TSGSA-basis. Note that the last line follows from the diagonality of the occupation number operator $\hat{n}_i = \hat{a}_i^\dagger \hat{a}_i$.

Geometry	$N_\uparrow = N_\downarrow$	$n_{\text{crop.}}(p)$	E_{TSGSA}	$\lvert E_{\text{TSGSA}} - E_{\text{exact}}\rvert$	$\lvert E_{\text{TSGSA}} - E_{\text{2-part.}}\rvert$
$L_x = 2$, $L_y = 2$	2	50/25	-3.300	0.388	5E-14
$L_x = 4$, $L_y = 2$	4	50/25	-6.074	0.045	1E-12
$L_x = 12$, $L_y = 1$	6	50/25	-3.010	0.222	0.105
$L_x = 6$, $L_y = 2$	6	50/25	-8.964	0.200	0.161
$L_x = 4$, $L_y = 3$	6	50/25	-4.184	0.667	0.320
$L_x = 2$, $L_y = 6$	6	50/25	-4.647	4.517	1.550

Table 2.11: Hubbard-TSGSA for systems with more than one spatial dimension (pbc.).

System	$n_{\text{crop.}}(p)$	E_{exact}	E_{TSGSA}	E_{DMRG}	relative error
$L = 4$, $N_\uparrow = N_\downarrow = 2$	4/2	-0.911	-0.882	-0.911	0.032
$L = 8$, $N_\uparrow = N_\downarrow = 4$	36/24	-1.975	-1.937	-1.975	0.019
$L = 12$, $N_\uparrow = N_\downarrow = 6$	50/25	-3.041	-2.949	-3.041	0.030
$L = 16$, $N_\uparrow = N_\downarrow = 8$	50/25		-3.961	-4.109	0.036
$L = 20$, $N_\uparrow = N_\downarrow = 10$	50/25		-5.021	-5.178	0.030
$L = 24$, $N_\uparrow = N_\downarrow = 12$	50/25		-6.090	-6.245	0.025

Table 2.12: Hubbard-TSGSA compared with DMRG (obc.).

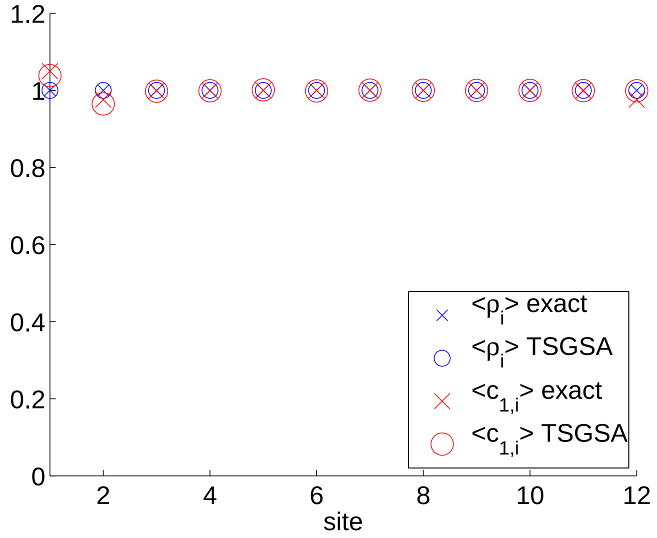

Figure 2.12: Density and same-time correlations for a Hubbard-type system ($L = 12$, $N_\uparrow = N_\downarrow = 6$). The correlations are measured from site 1. Compared are TSGSA and results obtained by exact diagonalization.

The same holds true for the same-time correlations:

$$\begin{aligned}\langle c_{ij} \rangle &= \langle \Psi_0 | \hat{n}_i \hat{n}_j | \Psi_0 \rangle \\ &= \sum_k |f_k|^2 \langle \phi_k | \hat{n}_i \hat{n}_j | \phi_k \rangle \ .\end{aligned}$$

The connected correlation function reads then:

$$\langle c'_{ij} \rangle = \langle c_{ij} \rangle - \langle \rho_i \rangle \langle \rho_j \rangle \ .$$

For numerical results see fig. 2.12. It shows that the correlations are slightly overestimated at small distances.

2.6.3 Boundary conditions for subsystem problems

This section introduces the idea to use periodic boundary-conditions for the subsystem eigenvalue problem. When combining the states to form a full problem subspace for the final computation this may give better results. Note that this contradicts the derivation

$n_{\text{crop.}}(p)$	N_{cores}	t [s]	E_{TSGSA}	rel. error
25/25/25	5	2223	-5.5101	0.066
50/50/50	10	5123	-5.5305	0.061
75/75/75	10	15916	-5.5322	0.061
100/100/100	20	23369	-5.8169	0.013

Table 2.13: Convergence of the method for a Hubbard-type system ($L = 22$, $N_\uparrow = N_\downarrow = 11$, pbc., $U = 10$), compared with DMRG (ground-state energy: -5.8907). N_{cores} indicates the number of computation cores used, t gives the total runtime in seconds.

of the problem, where the hoppings out of a certain subsystem is considered by the off-diagonal block in the ordered Hamiltonian.

Indeed, all calculations show, that using periodic boundary conditions give worse approximations. This is observed by the number of necessary eigensolutions from the subsystems to form equally accurate full system approximation. This number is generally larger in the case of periodic boundaries.

2.7 CPU time measurements

A world record for the diagonalization of the Hubbard model was achieved in 2008 by Yamada et al. at the Earth Simulator [34]. The largest system they calculated was of length 22 with 9 up and 8 down particles, which resulted in a matrix size of $159 \cdot 10^9$. Tab. 2.13 shows results for a system ($L = 22$, $N_\uparrow = N_\downarrow = 11$) at the for physics important case of half-filling. The basis size of this calculation was $5.0 \cdot 10^{11}$, which is three times larger than the record. The calculations were performed at a 8 QuadCore Opteron CPU cluster at Graz University of technology.

2.8 Vector selection

A crucial point for the algorithm is the amount of eigenvectors from the subsystems which are needed to build the TSGSA basis. Within this work only the lowest eigensolutions of the subsystems were considered. It may be a good idea to think of a more sophisticated method for selecting them. As can be seen in fig. 2.13 for a small system,

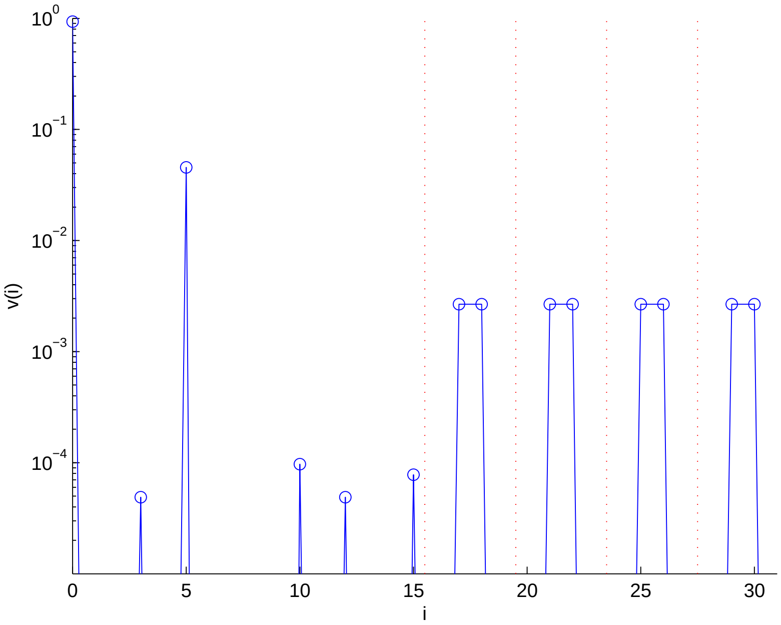

Figure 2.13: Ground state eigenvector coefficients of a system $L = 4$, $N_\uparrow = N_\downarrow = 2$ in the TSGSA basis. The modulus squared of each coefficient of the normed vector is shown. The partition boundaries are marked by vertical lines.

only a very limited amount of eigenvectors from the subsystems play an important role in contributing to the exact ground state. If they could be preselected, the computational effort could be decreased even more.

2.9 Implementation

2.9.1 Subsystems

The required subsystems are determined and arranged in a tree-shaped manner (see fig. 2.2). To every subsystem pointers to every *child*-subsystem are associated and the corresponding overlap matrices are calculated.

total cores	comp. cores	$t_{\text{subsystem}}$	t_{total}	speed-up
2	1	7358	7472	1.00
3	2	4236	4345	1.72
4	3	3134	3247	2.30
5	4	3319	3432	2.18
6	5	3048	3161	2.36

Table 2.14: MPI Speedup for parallel computation of subsystems ($L = 24$, $N_\uparrow = N_\downarrow = 12$, 5 subsystems). t_{total} and $t_{\text{subsystem}}$ denote the run times of the full calculation and the subsystem diagonalization only, respectively. One of the cores simply served as master to distribute the work.

2.9.2 Diagonalization of subsystems

For all the subsystems the specified amount of eigenvalues and eigenvectors is calculated.

2.9.3 Combination of subsystems

For the left part of the system all possible subsystems are assumed. The corresponding right subsystem is determined and the partition eigenvalues and eigenvectors are calculated. By recursion all child-subsystems are considered for each system and the corresponding overlaps are calculated.

2.9.4 Solving the final eigenproblem

Without calculating the matrix in the end a matrix product with a vector can be performed. This is used to calculate the lowest eigenvalues by a Lanczos-type algorithm.

2.9.5 Parallelization

A simple parallelization scheme can be devised by simply distributing the calculations of different subsystems to multiple computer cores. Tab. 2.14 shows results from a simple implementation using the OpenMPI framework performed on a computer cluster at Graz University of Technology (ghost.tugraz.at).

The limits of the speed-up can be explained by considering the system calculated. There are 5 partitions where one of them dominates because it is larger than the other four. So

total	comp.	50/25/25		60/40/25		80/40/25	
cores	cores	t	speed-up	t	speed-up	t	speed-up
2	1	15621	1.00	17700	1.00	20262	1.00
3	2	7050	2.22	10771	1.64	12092	1.66
4	3	5007	3.12	7425	2.38	10062	2.01
5	4	4748	3.29	7268	2.44	8008	2.53
6	5	3820	4.08	6774	2.61	8742	2.32
total	comp.	100/50/30		100/50/50/50			
cores	cores	t	speed-up	t	speed-up		
2	1	26981	1.00	54789	1.00		
3	2	17661	1.53	40074	1.37		
4	3	12599	2.14	21466	2.55		
5	4	12664	2.13	25458	2.15		
6	5	10901	2.46	20725	2.64		

Table 2.15: MPI speedup for parallel computation of subsystems ($L = 24$, $N_\uparrow = N_\downarrow = 12$, 3 or 4 subsystems). The numbers in the header line denote the subsystem cropping for each of the partition species.

more computational cores do not lower the necessary time. Furthermore, the total number of partitions is very small. More partitions may lead to a more efficient distribution of work.

Tab. 2.15 shows CPU run times for different croppings of the subsystem eigensolutions. Tab. 2.16 shows results for a specific system for different cropping numbers.

2.10 TSGSA and the Schur complement

Another possibility is using the Schur-complement to eliminate the higher partitions and get an effective Hamiltonian.

Consider the following block-structured Hamiltonian:

$$H = \begin{pmatrix} H_0 & H_{0,1} \\ H_{0,1}^\dagger & H_1 \end{pmatrix} ,$$

with H_0 the zeroth partition of equal amount of particles in both subsystems. H_1 is the

Levels	$n_{\text{crop.}}(p)$	E_{TSGSA}
2	50/25	-6.097
3	50/25/25	-6.097
3	60/40/25	-6.101
3	80/40/25	-6.101
3	100/50/30	-6.101
3	100/100/100	-6.103
4	100/50/50/50	-6.101

Table 2.16: Eigenvalue results obtained by TSGSA for a Hubbard model of specific system size ($L = 24$, $N_\uparrow = N_\downarrow = 12$, $U = 10$) for different cropping numbers. The solution obtained by DMRG is $E_0 = -6.245$. The calculations were performed on the computer cluster ghost.tugraz.at.

internal Hamiltonian of partition 1 and $H_{0,1}$ the part describing the hoppings over the boundary.

The problem can be transformed to a non-linear eigenvalue-problem of the size of H_0:

$$\left(H_0 - H_{0,1}(H_1 - \lambda)^{-1} H_{0,1}^\dagger\right) v = \lambda v \; . \tag{2.4}$$

H_1 consists of sub-partitions with no overlap:

$$H_1 = \begin{pmatrix} H_{1,1} & 0 \\ 0 & H_{1,2} \end{pmatrix} \; .$$

In the case of spin-less fermions there are two, in the case of the Hubbard model there are four sub-partitions. Such a matrix can easily be inverted by inversion of the individual matrices.

2.10.1 Numerical results

Tab. 2.17 shows, that indeed the eigenvalue problem can be solved by using the non-linear eigenproblem.

System	E_{exact}	$E_{\text{2-part.}}$	$E_{\text{1-part.}}$	$E_{\lambda=0}$	$n_{\text{iter.}}$	E_{TSGSA}
$L = 4$, $N = 2$	-0.385	-0.385	-0.385	-0.385	1	-0.385
$L = 8$, $N = 4$	-0.792	-0.792	-0.612	-0.812	3	-0.792
$L = 12$, $N = 6$	-1.188	-1.188	-0.992	-1.216	3	-1.188
$L = 16$, $N = 8$	-1.584	-1.584	-1.386	-1.623	3	-1.584

Table 2.17: Solving the non-linear eigenvalue problem obtained by Schur inversion iteratively. Spin-less fermions, pbc. V=10

2.11 TSGSA and the Green's function

An important quantity when dealing with strongly-correlated many-body systems is the dynamic Green's function (see sec. 1.8).
An approximation within the presented framework can be written in terms of matrix representations:

$$\begin{aligned} G_{i,j}(\omega) = \quad & \vec{g}^{\dagger}_{j,+1} V_{+1} \frac{1}{\omega - (V^{\dagger}_{+1} H_{+1} V_{+1} - E_0)} V^{\dagger}_{+1} \vec{g}_{i,+1} \\ + \quad & \vec{g}^{\dagger}_{i,-1} V_{-1} \frac{1}{\omega + (V^{\dagger}_{-1} H_{-1} V_{-1} - E_0)} V^{\dagger}_{-1} \vec{g}_{j,-1} \ . \end{aligned}$$

Here, $\vec{g}_{i,\pm1}$ denotes the excited vector obtained by adding/removing (±1) a particle at site i. $V_{\pm1}$ denotes the TSGSA basis for the system with one particle added/removed.
To determine the vectors $\vec{g}_{\pm1}$ a creation/annihilation operator in the TSGSA basis is needed:

$$\vec{g}_{\pm1} = A_{\pm1} \vec{g} \ .$$

This operator can again be written as block diagonal matrix where the blocks correspond to the individual partitions ($A_{\pm1} = \text{diag}(A_{1,\pm1}, A_{2,\pm1}, ...)$). Each partition operator can be written as a tensor product of the two individual subsystem operators:

$$A_{i,\pm1} = A_{i,\pm1,A} \otimes A_{i,\pm1,B} \ .$$

The single subsystem operator $A_{i,\pm1,X}$ (see also fig. 2.14)) can be obtained from the full matrix representation of the creation/annihilation operator in the Hubbard basis by rotation:

$$A_{i,\pm1,X} = V^{\dagger}_{(n\pm1)} a_{(n)\rightarrow(n\pm1)} V_{(n)} \ .$$

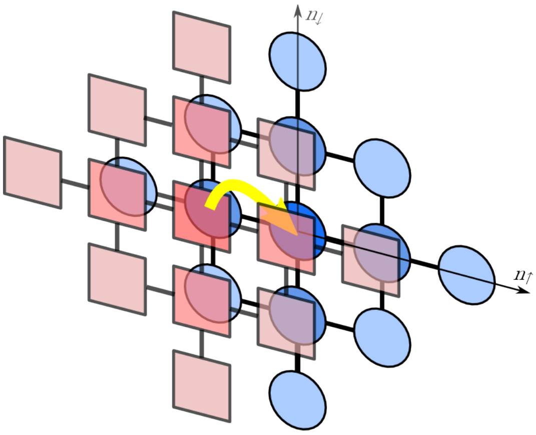

Figure 2.14: Illustration of the mapping from one subsystem to an adjacent one by the creation operator (here: $a^\dagger_{i,\uparrow}$). The squares denote the original TSGSA partitioning, the circles denote a partitioning with one more spin-up particle.

Here, $a_{(n)\to(n\pm1)}$ denotes the matrix representation of the creation/annihilation operator acting on the Hubbard basis of the subsystem with n particles and resulting in a vector in the basis with $n+1$ particles. $V_{(n)}$ and $V_{(n\pm1)}$ are the eigenvectors of the corresponding Hamiltonians.

In fact the operator $A_{\pm1}$ does not have to be calculated explicitly, but the application on the vector can be implemented. Once again it can be calculated with respect to the numerical complexity of the subsystem. For the Hubbard model the subsystem operators $a_{(n)\to(n\pm1)}$ can be easily constructed using individual operators for spin-up and spin-down, which further decreases the numerical effort.

Fig. 2.15 shows numerical results obtained by TSGSA. It can be seen that many features of the spectral function can be reproduced by the method.

2.12 Conclusions and Outlook

The approach presented in the previous sections gives a new perspective to calculate eigenvalue problems of strongly correlated systems. Furthermore, it has the advantage of producing an explicit matrix representation and an adjusted basis. The approach can easily be extended to multiple physical dimensions. In this case the partitions $l \neq 0$ may play a more important role due to an increased hopping probability between the

subsystems.
Potentially a large improvement to the algorithm can be achieved by a sophisticated selection process of the subsystem eigensolutions. As shown, only a comparably small number of them contribute to the full system ground state.

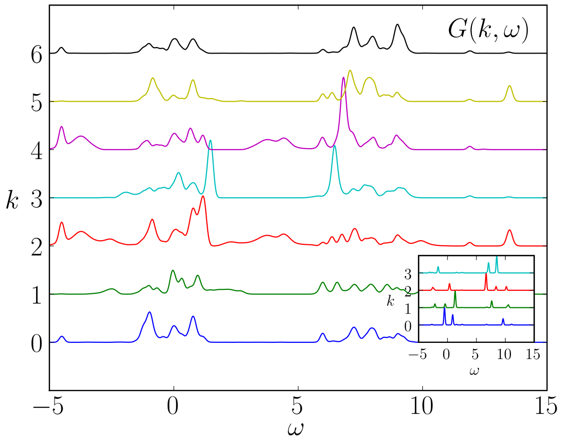

Figure 2.15: Plot of the spectral function obtained from the imaginary part of the Green's function calculated by TSGSA for a system of 12 sites at half-filling. The inset shows a direct computation of a system with size 6 for comparison.

3 A Numerical Projection Technique using Basis Reordering by Potential Terms

3.1 Motivation

$$H = -t \sum_{\langle ij \rangle} a_i a_j^\dagger + V \sum_{\langle ij \rangle} \hat{n}_i \hat{n}_j \ .$$

In a system of spin-less particles at half-filling and a repulsive nearest neighbor interaction particles tend to occupy sites alternatingly and leave unoccupied sites in between. Thus, it is reasonable to assume that the basis states contributing to the system ground state are mainly particle configurations with no or few particle pairs on adjacent sites.
The following considerations include a basis reordering by the number of nearest neighbor particles in such systems. This means, that the first partition of basis states corresponds to configurations with no pairs, then one, two, etc. number of pairs on adjacent sites. These partitions correspond to diagonal matrix entries in the Hamiltonian as multiples of V.

3.1.1 Example: 3 particles on 6 sites

A system of three spin-less particles on 6 sites has 20 occupation number basis states. They can be partitioned into 3 sets (assuming periodic boundary conditions of a 1D system):

- 2 states with no adjacent particles: $|\uparrow 0 \uparrow 0 \uparrow 0\rangle$ and $|0 \uparrow 0 \uparrow 0 \uparrow\rangle$
- 12 states with one pair of adjacent particles: $|\uparrow\uparrow 0 \uparrow 00\rangle$, $|\uparrow\uparrow 00 \uparrow 0\rangle$, ...
- 6 states with 2 pairs of adjacent particles: $|\uparrow\uparrow\uparrow 000\rangle$, $|0 \uparrow\uparrow\uparrow 00\rangle$, ...

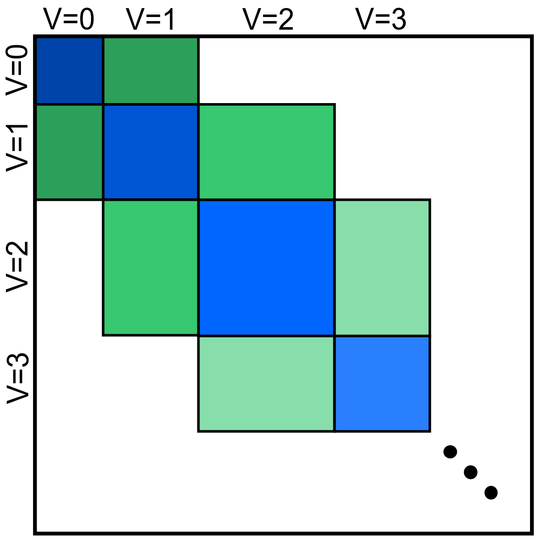

Figure 3.1: Illustration of the Hamilton matrix of a V-ordered occupation number basis. The blue parts contain the hopping terms inside a partition and the potential on the diagonal and the green parts contain the hopping terms between the partitions.

3.1.2 Hamilton matrix structure

Sorting the basis by the entries on the diagonal leads to a partitioning of the Hamilton matrix (see fig. 3.1) into different *sectors*. The diagonal entries have the form $N_\kappa V$, where N_κ is the number of particle pairs occupying nearest-neighbor sites. With strong coupling states with increasing N_κ have less weight in the low energy eigensolutions.

Note that the hopping of a particle can only change the number of particle pairs on neighboring sites N_κ by a maximum of one. Therefore, only off-diagonal blocks connecting neighboring partitions have entries.

A very similar partitioning can be devised for the Hubbard model. Here, the relevant quantity for the classification of the basis states is the number of doubly occupied sites. Again, only adjacent partitions are connected.

3.1.3 Projection formalism

The projection technique in the context of strongly correlated many body Hamiltonians (see app. I) allows to restrict the degrees of freedom and to construct an effective model

Hamiltonian. A well-known example would be the derivation of the tJ-model as a simplification of the Hubbard model. Here, the set of *dynamical variables*, corresponding to the occupation number basis states is reduced to the configurations with no interaction particle pairs. The dynamics of one interacting pair is projected onto that basis set using perturbation theory. Further states with more than one pair of particles are neglected. Looking back at fig. 3.1 this method projects the second sector onto the first.
The approach presented in the next sections can be viewed as an extension of the standard projection technique. The method is demonstrated using two strongly correlated material Hamiltonians, the spin-less fermion model with nearest neighbor repulsion and the Hubbard model with on-site repulsion.

3.2 Numerical projection technique

As a toy problem, consider a two-partitioned matrix with blocks A and B on the diagonal and a connecting off-diagonal block E. The corresponding eigenvalue problem reads

$$\begin{pmatrix} A & E \\ E^\dagger & B \end{pmatrix} \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} . \tag{3.1}$$

The blocks correspond to partitions of the original problem. A may be constructed from multiple sectors corresponding to individual N_κ. Let B consist of a single sector, which includes higher states than A in terms of energy contents. All further sectors are dismissed for the time being.
Transforming the Hamilton matrix by a Schur transformation results in an equivalent expression

$$\begin{pmatrix} S & 0 \\ E^\dagger & B \end{pmatrix} \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} I & -EB^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} ,$$

with $S = A - EB^{-1}E^\dagger$.
From the second line of the matrix equation follows

$$\vec{x}_2 = -(B - \lambda)^{-1} E^\dagger \vec{x}_1 . \tag{3.2}$$

Inserting into the first line and rearranging leads to

$$S_\lambda \vec{x}_1 := (A - E(B - \lambda)^{-1} E^\dagger) \vec{x}_1 = \lambda \vec{x}_1 . \tag{3.3}$$

Note that this equation is in principle exact, no approximations have been made so far. Note that solutions of this equation can be used to form solutions of the total eigenproblem. Solutions of the form $(\vec{0}, \vec{x}_2)$ cannot be calculated in this way, but their eigenvalues are of order $O(V)$ instead of $O(t)$ anyway.

In the end an equivalent problem to equation 3.1 of smaller size (just the size of B) is obtained but at the prize of a non-linear eigenvalue problem $(S_\lambda - \lambda)\vec{x}_1 = 0$.

3.2.1 Series expansion of $(B - \lambda)^{-1}$

The matrix B corresponds to a sector with a specific potential term $N_\kappa V$. For increasing partition number this entries become increasingly larger than the hopping term, so the following consideration can be justified. Split the matrix B in kinetic part of the Hamiltonian $\tilde{B}$, and in potential part $B_V = N_\kappa V I$, which is diagonal:

$$B = B_V + \tilde{B} \; .$$

Then expand the inverse in powers of $(B_v - \lambda)^{-1}$. Note that λ will be of order $O(t) \ll O(V)$, so $B_v - \lambda$ is of order $O(V)$. The expansion yields:

$$\begin{aligned}\frac{1}{B_V - \lambda + \tilde{B}} &= \frac{1}{B_V - \lambda} - \frac{1}{(B_V - \lambda)^2}\tilde{B} + \ldots \\ &= \sum_{\nu=0}^{\infty} \frac{1}{(B_V - \lambda)^{\nu+1}}(-\tilde{B})^\nu \; ,\end{aligned}$$

and the calculation of S_λ becomes

$$S_\lambda = A - \sum_{\nu=0}^{\infty} \frac{1}{(B_V - \lambda)^{\nu+1}} E(-\tilde{B})^\nu E^\dagger \; . \tag{3.4}$$

Note that if B corresponds to a single sector the expressions $1/(B_V - \lambda)^{\nu+1}$ are only numbers. For an evaluation of S_λ only these prefactors are modified, when the value λ is changed.

In terms of implementation the matrices $E(-\tilde{B})^\nu E^\dagger$ can be calculated once and reused for many different values λ. These matrices to be stored have the size of A and may be comparably small.

Using this expansion, also the expression needed in eq. 3.2 for an evaluation of the eigenvector can be calculated, too:

$$-(B - \lambda)^{-1} E^\dagger = -\sum_{\nu=0}^{\infty} \frac{1}{(B_V - \lambda)^{\nu+1}} (-\tilde{B})^\nu E^\dagger \; .$$

3.2.2 Solving the non-linear eigenproblem by successive evaluation

One possibility is to use a current approximation of λ to calculate S_λ, then solve the eigenvalue problem by suitable means and use the resulting eigenvalue as a new approximation for λ. This corresponds to

$$\lambda^{(\mathrm{new})} = \alpha\mathcal{E}(\lambda^{(\mathrm{old})}) + (1-\alpha)\lambda^{(\mathrm{old})} , \tag{3.5}$$

with $\alpha = 1$ and $\mathcal{E}(.)$ denoting the evaluation of the new lambda by solving the eigenvalue problem of S_λ. The fixed point of λ can be attained by recursion.

3.2.3 Solving the non-linear eigenproblem by a Newton-Raphson scheme

A more sophisticated possibility is to use the Newton-Raphson method to get a better new approximation for λ. Define $\Phi(\lambda) = \mathcal{E}(\lambda) - \lambda = 0$

$$\lambda^{(\mathrm{new})} = \lambda - \frac{\Phi(\lambda)}{\Phi'(\lambda)} = -\frac{1}{\mathcal{E}'(\lambda)-1}\mathcal{E}(\lambda) + \frac{\mathcal{E}'(\lambda)}{\mathcal{E}'(\lambda)-1}\lambda .$$

The last equality shows that this can be transformed to an equation of form 3.5 with

$$\alpha = -\frac{1}{\mathcal{E}'(\lambda)-1} .$$

3.2.3.1 Calculation of $\mathcal{E}'(\lambda)$

The derivative of S_λ (eq. 3.3) with respect to λ reads

$$S'_\lambda = -E(B-\lambda)^{-2}E^\dagger . \tag{3.6}$$

Using expansion 3.4 one can build the derivative:

$$S'_\lambda = -\sum_{\nu=0}^{\infty} \frac{(\nu+1)}{(B_V-\lambda)^{\nu+2}} E(-\tilde{B})^\nu E^\dagger . \tag{3.7}$$

Therefore, an approximation of the $\mathcal{E}'(\lambda)$ using the current approximation to the eigenvector $|x\rangle$ can be calculated using the Hellmann-Feynman theorem:

$$\begin{aligned} \mathcal{E}'(\lambda) &= \frac{\langle x| S'_\lambda |x\rangle}{\langle x|x\rangle} \\ &= -\sum_{\nu=0}^{\infty} \frac{(\nu+1)}{(B_V-\lambda)^{\nu+2}} \frac{\langle x| E(-\tilde{B})^\nu E^\dagger |x\rangle}{\langle x|x\rangle} . \end{aligned}$$

Again, the stored matrices as described in sec. 3.2.1 can be reused.

3.3 Iteration concept

The following algorithm defines an iterative scheme, which successively integrates further sectors into the calculation.

1. Solve the eigenvalue problem for an initial partition A_1 which may include several sectors $N_\kappa = 0, 1, 2..N'$ and get the corresponding unitary matrix V_1 containing the eigenvectors.

2. Expand the matrix to include the next sector (B_1):
$$A_2 = \begin{pmatrix} A_1 & E_1 \\ E_1^\dagger & B_1 \end{pmatrix} .$$
Transform it using the unitary matrix $\mathrm{diag}(V_1, I)$
$$\tilde{A}_2 := \begin{pmatrix} V_1^\dagger & \\ & I \end{pmatrix} \begin{pmatrix} A_1 & E_1 \\ E_1^\dagger & B_1 \end{pmatrix} \begin{pmatrix} V_1 & \\ & I \end{pmatrix} = \begin{pmatrix} V_1^\dagger A_1 V_1 & V_1^\dagger E_1 \\ E_1^\dagger V_1 & B_1 \end{pmatrix} , \tag{3.8}$$
with $D_1 := V_1^\dagger A_1 V_1$ diagonal.

3. Having solved the eigenvalue problem for $\tilde{A}_2$ by the expansion scheme and obtaining the eigenvectors $\tilde{V}_2$ from it one can create the new over-all eigenvector matrix:
$$V_2 = \begin{pmatrix} V_1 & \\ & I \end{pmatrix} \tilde{V}_2 .$$
Proof:
$$\begin{aligned} V_2^\dagger A_2 V_2 &= \tilde{V}_2^\dagger \begin{pmatrix} V_1^\dagger & \\ & I \end{pmatrix} \begin{pmatrix} A_1 & E_1 \\ E_1^\dagger & B_1 \end{pmatrix} \begin{pmatrix} V_1 & \\ & I \end{pmatrix} \tilde{V}_2 \\ &= \tilde{V}_2^\dagger \underbrace{\begin{pmatrix} D_1 & V_1^\dagger E_1 \\ E_1^\dagger V_1 & B_1 \end{pmatrix}}_{\tilde{A}_2} \tilde{V}_2 \\ &= D_2 . \end{aligned}$$
The second line follows from eq. 3.8 and the third by construction of $\tilde{V}_2$.

4. Continue the iteration by including the next sector B_2:
$$\tilde{A}_3 = \begin{pmatrix} D_2 & V_2^\dagger E_2 \\ E_2^\dagger V_2 & B_2 \end{pmatrix}$$

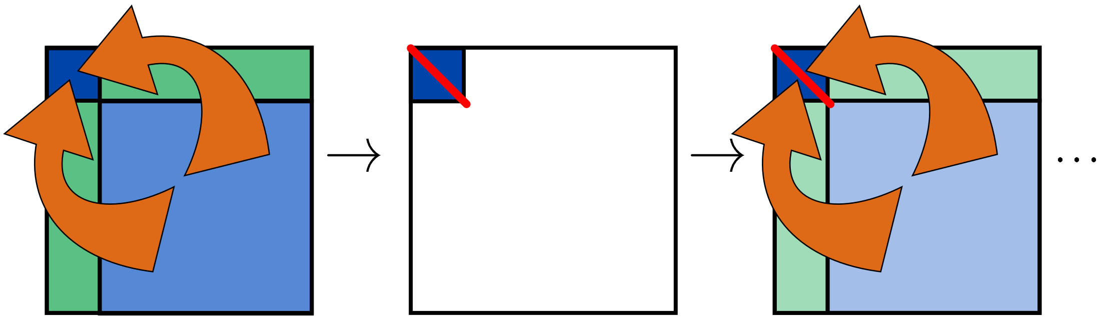

Figure 3.2: Truncation procedure for the scheme presented in the text. Each sector is consecutively projected onto the original effective matrix and diagonalized.

The iteration steps 3 and 4 can be continued up to a sector, which can be determined on-the-fly by attaining convergence. Then it can be assumed that further sectors do not contribute and the evaluation can be aborted.

3.3.1 Truncation

When consecutively including higher sectors, the reference part of the matrix in the upper left corner stays constant in size. One can imagine this as a projection of the higher sectors onto the same matrix, one after another.
Some systems may have large first sectors. The number of vectors retained in the first partition can be restricted based on a suitable criterion. The most simple would be to keep only the lowest eigenstates of the partition solution. More sophisticated criteria as in DMRG are also conceivable. The restriction can also take place at a later time in the iteration.
In this work, a fixed number of vectors $\tilde{p}_1$ is kept from the initial partition. This number also determines the size of the upper-left block during the recursion procedure.

3.4 Translational invariance

Due to the translational invariance of the systems with periodic boundary conditions, the problem can be separated in different regimes of fixed momentum k. The vectors necessary for the calculation of the overall ground state can be restricted to the vectors with the appropriate momentum. In the case of the system of fermions with nearest neighbor interaction we have a total momentum of $k = \pm\frac{\pi}{2}$. The selection by momentum leads to a significant reduction of the computational effort due to the smaller matrix sizes involved. This way, including vectors which do not contribute to the overall system ground state can be avoided.

3.4.1 Determination of the momentum

Momentum and translation operator are related by

$$\hat{T} = e^{i\hat{p}} \ .$$

They have the same set of eigenfunctions and the eigenvalues are related consequently by

$$\lambda_T = e^{ik} \ .$$

Since the momentum operator and the Hamiltonian commute, all eigenstates of the Hamiltonian are also eigenvalues of the momentum operator. So for non-degenerate states the momentum can easily be calculated by using the translation operator

$$k = -i \ln \langle \Psi | \hat{T} | \Psi \rangle \ .$$

For degenerate eigenstates the vectors have to be rotated in the subspace in order to get eigenstates of the translation operator. This can be achieved by solving the eigenvalue problem

$$(U^\dagger \hat{T} U) V = V \Lambda_T \ ,$$

where U is the matrix of degenerate eigenvectors in columns. The problem size is given by the multiplicity of the energy eigenvalue. The eigenvalues in the diagonal of Λ_T give the momenta of the subspace vectors. And the rotated eigenvectors can be determined by

$$U_{\hat{p}} = UV \ .$$

3.5 Numerical results for the fermion model with nearest neighbor repulsion

Tab. 3.1 shows the results for the presented algorithm for the case of spin-less fermions. In the case of periodic boundary conditions the first sector ($N_\kappa = 0$) consists of just 2 states with alternatingly occupied and unoccupied sites, i.e. $\tilde{p}_1 = 2$. Here, one can see that the algorithm is significantly faster than the highly optimized ARPACK routine (Arnoldi implementation) for increasing system size.

system	N_b	t_{NPT} [s]	E_{NPT}	t_{A} [s]	E_0	$\frac{E_{\text{NPT}}-E_0}{E_0}$
$L = 16$, $N = 8$	12870	0.01	-1.5724	0.21	-1.5840	0.007
$L = 18$, $N = 9$	48620	0.01	-1.7651	0.83	-1.7822	0.010
$L = 20$, $N = 10$	184756	0.10	-1.9565	4.05	-1.9800	0.012
$L = 22$, $N = 11$	705432	0.34	-2.1468	10.38	-2.1780	0.014
$L = 24$, $N = 12$	2704156	0.62	-2.3359	87.13	-2.3700	0.014

Table 3.1: Comparison between the numerical projection technique and a traditional Arnoldi eigensolver (ARPACK) for different systems at half-filling with $V = 10$ and periodic boundary conditions. N_b denotes the basis size. t_{NPT}, E_{NPT} and t_{A}, E_0 denote run time and ground state energy for NPT and ARPACK, respectively. The accuracy was set to $\epsilon_\lambda = 0.1$.

3.5.1 Dependence on the starting partition

Fig. 3.3 shows what happens when starting at a specific initial partition size. That means, that initially the exact problem is solved for the system including all sectors up to a number N_κ^{max}, which is indicated by the color. Still, for all calculations the dimension of the upper-left part of the matrix was reduced to $\tilde{p}_1 = 2$ after the initial eigenvalue calculation. E.g. the turquoise upward triangles show the development, when using all sectors up to $N_\kappa = 3$ as initial partition. The relative accuracy increases by one order of magnitude as compared to $N_\kappa = 2$.

The figure shows, that for each starting partition convergence is reached for a different value. After about two steps of the iterative procedure no improvements to the eigenvalues are achieved. Tab. 3.2 shows this effect numerically.

The reason is that important vectors are thrown away during the recursion procedure which connect to the sectors with larger N_κ. Therefore introducing the new sectors does not contribute to the improvement of the low eigenstates.

For large initial problem sizes the accuracy is increased by about one to two orders of magnitude, but it levels off again to a certain value. This is obviously due to the small truncation size $\tilde{p}_1 = 2$.

3.5.2 Dependence on the vector truncation

This section shows the dependence on the truncation size $\tilde{p}_1$.

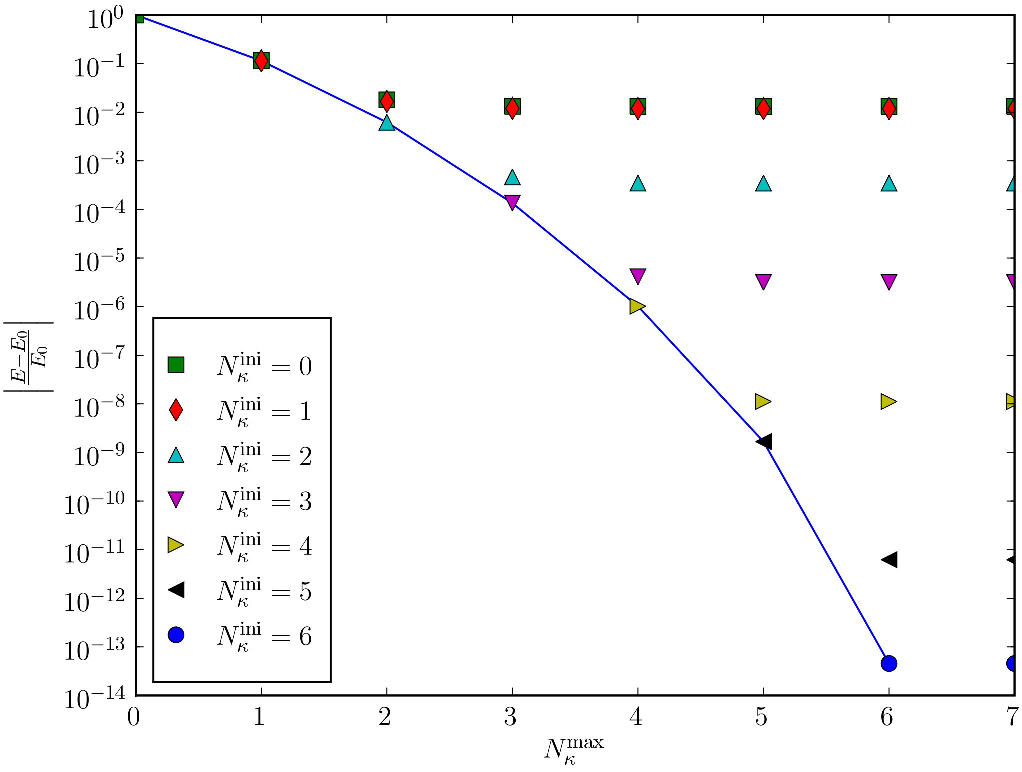

Figure 3.3: Convergence of the lowest eigenvalue when introducing successively higher sectors $N_\kappa^{\max}$, depending on the initial partition size N_κ^{ini}. In all cases, after the initial eigenvalue computation, only 2 vectors were kept, i.e. $\tilde{p}_1 = 2$. The blue solid line indicates the exact eigenvalues of each partition including sectors up to $N_\kappa^{\max}$. The results are presented on logarithmic scale as distance to the exact ground-state ($L = 20$, $N = 10$, $V = 10$).

system	N_κ^{ini}	N_b^{ini}	E_{NPT}	t_{NPT} [s]
$L = 12$, $N = 6$	0	2	-1.18388	0.00
	1	62	-1.18798	0.04
	2	362	-1.18800	0.94
	3	762	-1.18800	11.53
	exact	924	-1.18800	
$L = 14$, $N = 7$	0	2	-1.37977	0.02
	1	86	-1.38736	0.23
	2	716	-1.38743	38.08
	3	2116	-1.38743	556.04
	exact	3432	-1.38743	
$L = 20$, $N = 10$	0	2	-1.95649	0.59
	1	182	-1.95649	0.61
	2	3422	-1.97931	0.59
	3	23582	-1.97999	0.43
	4	76502	-1.98000	0.64
	exact	184756	-1.98000	

Table 3.2: Dependence of the algorithm result on different sizes of the initial partition consisting of N_κ^{ini} sectors and N_b^{ini} basis states ($V = 10$, $\epsilon_\lambda = 10^{-3}$, $\tilde{p}_1 = 2$).

$\tilde{p}_1$	E_{NPT}	$\frac{E_{\mathrm{NPT}}-E_0}{E_0}$	$t_{\mathrm{iters.}}$ [s]
2	-1.9565	0.012	0.66
4	-1.9573	0.011	1.34
6	-1.9594	0.010	2.25
10	-1.9629	0.008	3.76
14	-1.9713	0.004	6.79
16	-1.9749	0.003	9.21
3 sectors	-1.9793	0.001	
exact	-1.9800		

Table 3.3: Dependence of the lowest eigenvalue on the truncation size $\tilde{p}_1$ for a system with $L = 20$, $N = 10$, $V = 10$, $\epsilon_\lambda = 10^{-4}$. Here, the initial partition consists of the first two sectors ($N_\kappa^{\mathrm{ini}} = 1$) which includes 182 basis states. In addition the exact lowest eigenvalue is given for the sub-matrix including the first three sectors and the original matrix.

Tab. 3.3 shows the dependence of the final result on the dimension $\tilde{p}_1$ of the upper left part block of the iteration procedure. The improvement can be seen clearly, although the accuracy is limited by the fact, that only the first two sectors are used as the initial problem. The accuracy given by three initial sectors cannot be reached.

Fig. 3.4 shows the scalar product of all partition eigenvectors with the total system ground state. It can be seen that not only the lowest eigenstates of the smaller partition contribute to the exact solution, but also certain, although not many, higher states.

This shows that a fairly limited amount of partition eigenvectors are needed to represent a highly accurate approximation of the system ground state.

3.6 Analysis of the non-linear eigenvalue problem

This section deals with the analysis of a the blocked eigenvalue problem

$$F\vec{x} = \begin{pmatrix} A & E \\ E^T & B \end{pmatrix} \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} = \lambda \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} .$$

Note that when setting the eigenvector components of the first partition to zero ($\vec{x}_1 = 0$),

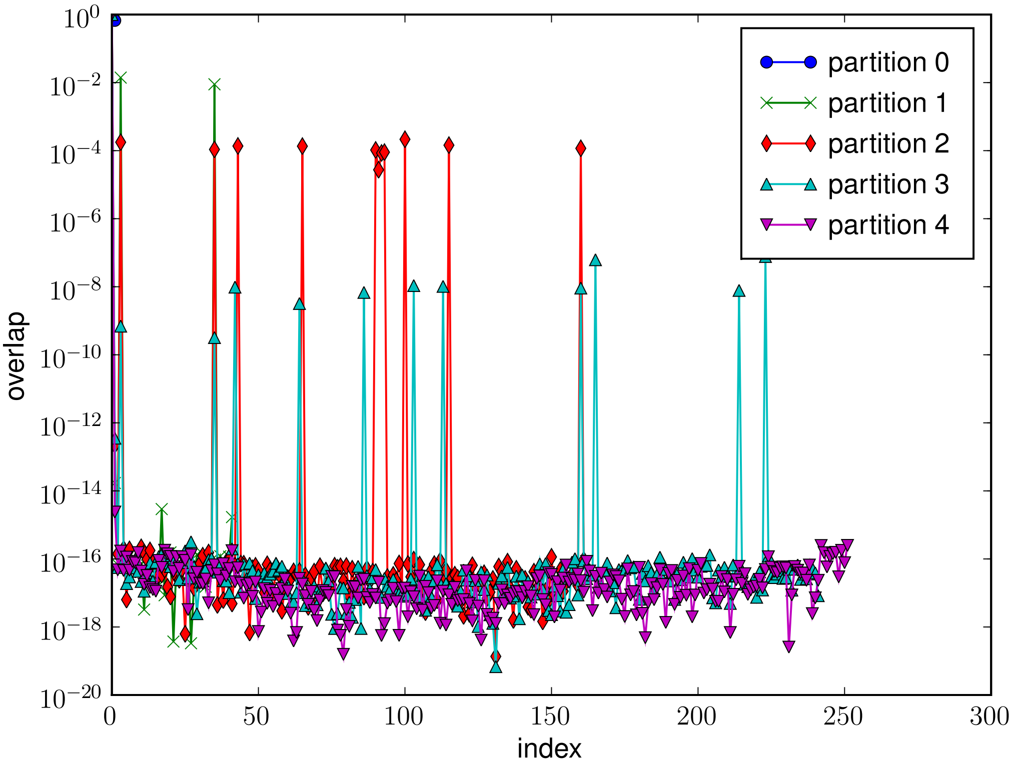

Figure 3.4: Scalar products of the exact eigenvectors (denoted by index) of different sectors with the exact ground state of the system. The largest contribution stems from the sector with zero double-occupancies (partition 0), which consists of only two states (blue).

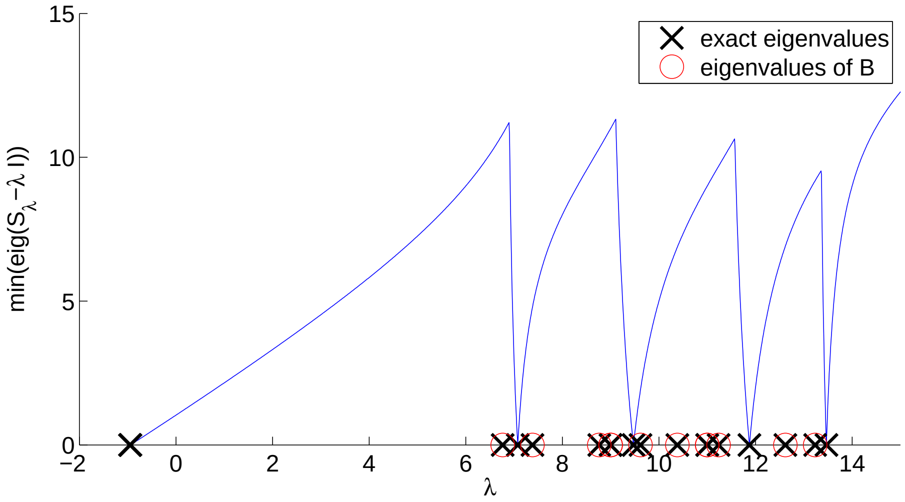

Figure 3.5: Minimal eigenvalue of $S_\lambda - \lambda I$ over λ for system $L = 10$, $N = 5$.

above equation results in

$$B\vec{x}_2 = \lambda\vec{x}_2 \ .$$

So the second part of the eigenvector is an eigenvector of B. This leads to a possible classification of the eigenvalues of F into eigenvalues of B and the rest.

Fig. 3.5 shows the exact solution of all eigenvalues and in addition it shows the minimum magnitude of the eigenvalues of

$$S_\lambda - \lambda I = (A - E(B-\lambda)^{-1}E^\dagger) - \lambda I \ .$$

One can see, that this value only vanishes at full system eigenvalues which are not an eigenvalue of B.

Due to multiplicity, the number of eigenvalues of the full system is generally not decomposed into the number of eigenvalues of B and the dimensionality of A. In case of fig. 3.5 there are 6 eigenvalues, which are not eigenvalues of B whereas $\dim(F) - \dim(B) = 2$

3.7 Alternatives for solving the non-linear eigenvalue problem

The slowest part of the algorithm is building the inverse of the matrix $(B - \lambda)$. This section deals with an alternative approach to solve the nonlinear eigenproblem (eq. 3.3)

$$S_\lambda \vec{x}_1 = (A - E(B - \lambda)^{-1} E^\dagger)\vec{x}_1 = \lambda \vec{x}_1 \ .$$

3.7.1 Matrix product with S_λ

A simple product of S_λ with a vector leads to

$$\begin{aligned} S_\lambda \vec{x} &= (A - E(B - \lambda)^{-1} E^\dagger)\vec{x} \\ &= Ax - E(B - \lambda)^{-1} \underbrace{E^\dagger \vec{x}}_{\vec{y}} \\ &= Ax - E(B - \lambda)^{-1} \vec{y} \ . \end{aligned}$$

Therefore one can solve the linear problem

$$(B - \lambda)\vec{z} = \vec{y} \ ,$$

for z and yields

$$S_\lambda \vec{x} = Ax - E\vec{z} \ .$$

Note that in the course of this algorithm the matrix A and therefore the vector $\vec{x}$ is usually very small compared to the large matrix B. So an inversion of B is much more complicated than solving the stated linear problem for a few vectors.
This also leads to the following possibility. Instead of evaluating S_λ by inversion of $B - \lambda$ one could apply the above procedure to the identity matrix. Since the dimension of this matrix is small compared to B the linear problem has to be solved only a comparable small number of times.

3.7.2 Non-linear Arnoldi method

A non-linear, sparse eigenvalue problem

$$T(\lambda)x = 0 \ , \tag{3.9}$$

can be solved using a non-linear Arnoldi method [35].
This is a preconditioned method with a suitable inverse approximation $M \approx T(\sigma)^{-1}$ with σ close to the wanted eigenvalue. It relies on solving a projected eigenproblem

$$V^T T(\lambda) V \vec{y} = 0 \ ,$$

where λ is a current approximation to the eigenvalue. After each diagonalization of the projected problem the search space V is extended by a new vector

$$v = MT(\mu)\vec{u} \ ,$$

where M is the preconditioner and $\vec{u}$ is a Ritz vector.
In this case it could be imagined to use

$$M = A - E^T (V_B - \lambda)^{-1} E \ .$$

as a preconditioner where V_B is the diagonal of B containing the potential only.

3.7.3 Method of successive linear problems

This method was proposed in [35] to solve the projected problem. Due to the characteristic of the present eigenvalue problem, which is actually not very large, but the evaluation of $T(\lambda)$ is rather complex, this algorithm could be useful for the full problem.
A first order approximation of eq. 3.9 yields

$$T(\lambda)x \approx (T(\mu) - \theta T'(\mu))\vec{x} = 0 \qquad \text{with} \quad \theta = (\mu - \lambda) \ ,$$

and leads to the following algorithm.
Solve successively the generalized eigenvalue problem

$$T(\mu)\vec{u} = \theta T'(\mu)\vec{u}$$

and update λ by

$$\lambda = \lambda_{old} - \theta$$

3.7.4 Numerical experiments

For a comparison of the mentioned methods, see tab. 3.4. The different methods are

- Expansion as explained in section 3.2.1.

	$L=16$, $N=8$		$L=12$, $N=6$	
method	time [s]	E_{NPT}	time [s]	E_{NPT}
Expansion	19.18	-1.58387670	0.78	-1.187984987
Expansion (Newton)	25.78	-1.58387665	0.99	-1.187984975
Arnoldi (GS)	34.93	-1.58387668	1.09	-1.187984975
Succ. Linear (Exp.)	26.52	-1.58387665	1.06	-1.187984975
Succ. Linear (Prod.)	113.00	-1.58387665	2.76	-1.187984975
Arnoldi (Exp.)	511.65	-1.58387668		

Table 3.4: Comparison of different algorithms for solving the Non-linear eigenproblem 3.9. See text for details of the methods.

- Expansion using Newton acceleration (sec. 3.2.3).
- Arnoldi with evaluating $S_\lambda \vec{v}$ by solving a linear equation (sec. 3.7.1).
- Method of successive linear products with expansion to calculate S_λ (sec. 3.7.3).
- Method of successive linear products evaluating $S_\lambda \vec{v}$ by solving a linear equation.
- Arnoldi with expansion for calculation of the product $S_\lambda \vec{v}$.

Tab. 3.4 shows that the sophisticated methods do not improve the computation time compared to the simple expansion scheme.

3.7.5 On the difference of the successive linear approach to standard Newton

Going back to sec. 3.2.3 the following method for the update of λ was introduced:

$$\lambda^{\text{new}} = \lambda - \frac{\Phi(\lambda)}{\Phi'(\lambda)} \quad \text{with} \quad \Phi(\lambda) = \vec{v}^\dagger S(\lambda)\vec{v} - \lambda \tag{3.10}$$

The approach of successive linear problems solves the generalized eigenvalue problem:

$$T(\lambda) - (\lambda^{\text{new}} - \lambda)T'(\lambda) = 0 \tag{3.11}$$

On clearly sees, that eq. 3.10 leads to a similar result, when multiplying with $\Phi'(\lambda)$

$$\Phi'(\lambda)(\lambda^{\text{new}} - \lambda) = \Phi(\lambda)$$

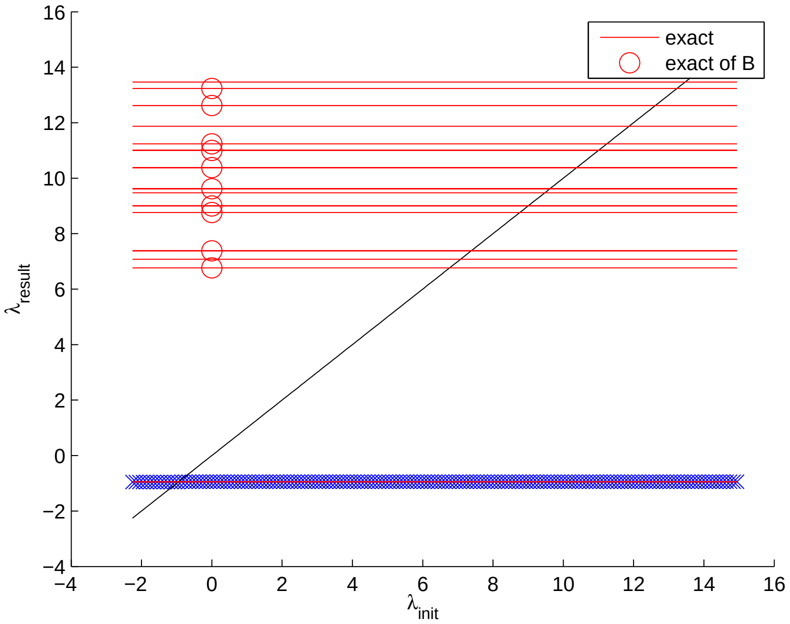

Figure 3.6: Convergence value of the direct method (blue crosses) for different initial values. Here, always the eigenvalue closest to the initial value was taken for the next iteration. The lowest level consists of two close energies. All initial values below and above this level converge eventually to the lower and higher state, respectively. The lines show the exact eigenvalues of the full problem, the eigenvalues of B are marked by a circle.

The difference is, that eq. 3.11 leaves the vectors open to calculation, while the other scheme keeps them fixed using the previous approximation.

3.7.6 Convergence properties

Differences in convergence behavior can be seen looking at fig. 3.6 to 3.8. The first picture shows, that taking the solution of the linear eigenproblem as the next approximation always leads to convergence to the lowest eigenvalues. Comparing figures 3.7 and 3.8 shows that the successive linear method is more reliable in converging to a eigenvalue close to the initial value. Additionally it is shown that the eigenvalues of B, which are also eigenvalues of the full problem, are never converged to.

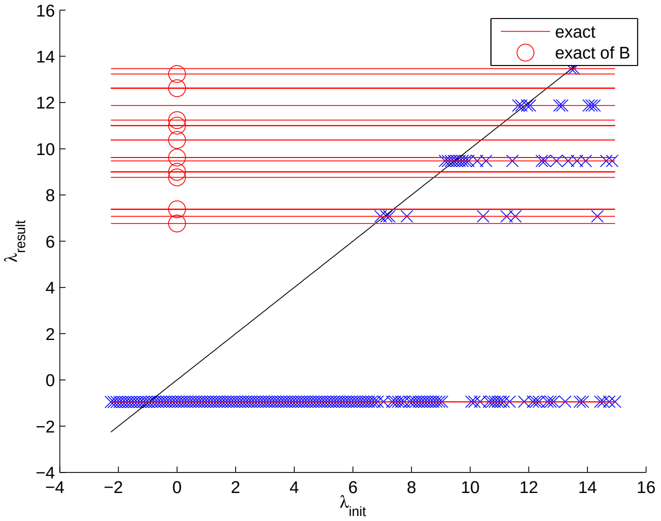

Figure 3.7: Convergence value of the Newton method (blue crosses) for different initial values. Here, always the eigenvalue closest to the initial value was taken for the next iteration. The lines show the exact eigenvalues of the full problem, the eigenvalues of B are marked by a circle.

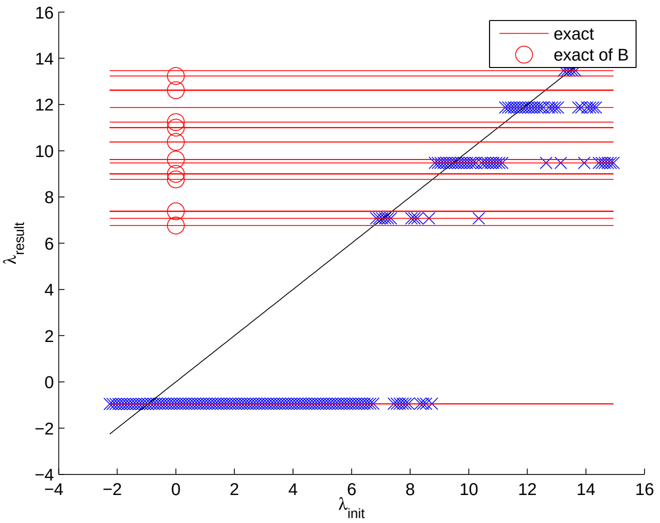

Figure 3.8: Convergence value of the method (blue crosses) of successive linear problems for different initial values. Here, always the eigenvalue closest to the initial value was taken for the next iteration. The lines show the exact eigenvalues of the full problem, the eigenvalues of B are marked by a circle.

system	time (full)	time (partial)
$L = 10,\ N_\uparrow = N_\downarrow = 5$	1.20	1.79
$L = 12,\ N_\uparrow = N_\downarrow = 6$	18.59	32.65
$L = 14,\ N_\uparrow = N_\downarrow = 7$	(mem)	371.00

Table 3.5: Algorithm running time using partial construction of a Hubbard Hamiltonian compared to full construction. (mem) indicates that the full Hamiltonian does not fit into memory of the used machine (2GB RAM).

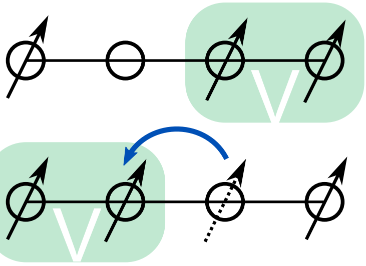

Figure 3.9: Conservation of number of nearest neighbor pairs in a hopping process of spin-less fermions.

3.8 Partial construction of the Hamiltonian

Since the Hamiltonian is not needed in full, one can construct only the parts of the Hamiltonian belonging to a single sector. This saves a large amount of system memory. Additionally it may save computational time, if not all sectors are necessary for convergence (see tab. 3.5).

3.8.1 Implementation

At a particular step the following matrices have to be known: the Hamiltonian inside a particular sector, and the overlap to the previous sector. This matrices can be calculated explicitly by considering elementary hopping processes. In the case of spin-less particles such a process conserves the number of nearest neighbor pairs only in the situation depicted in fig. 3.9. Considering only these cases the Hamiltonian can be computed. The same holds for the inter-sector parts, which change the number of pairs by one.

system	classic. projector	NPT	exact
$L = 14, N = 7, V = 10$	-1.4615	-1.3798	-1.3874
$L = 16, N = 8, V = 10$	-1.6697	-1.5724	-1.5840
$L = 18, N = 9, V = 10$	-1.8784	-1.7651	-1.7822
$L = 20, N = 10, V = 10$	-2.0871	-1.9565	-1.9800

Table 3.6: Comparison of results of the NPT algorithm and the traditional projector formalism.

3.9 Comparison to standard projection-formalism

Instead of successively solving eq. 3.3

$$(A - E(B - \lambda)^{-1}E^{\dagger})x_1 = \lambda x_1 \ ,$$

the projector formulation (see app. I) leads to the equation

$$(A - E(B - E_0)^{-1}E^{\dagger})x_1 = \lambda x_1 \ , .$$

The NPT procedure is not only self-consistently updating the value λ on the left-hand side, but it also includes all higher terms involving not only states with just one nearest neighbor pair.

Tab. 3.6 lists some values to compare both approaches. It shows that the NPT clearly exceeds the results of the classical projector scheme in terms of accuracy.

3.10 Numerical results for the Hubbard model

In contrast to the previous model the first sector of the Hubbard model is comparably large due to the spin degeneracy. The number of basis state in the first sector amounts to $\binom{L}{N_\uparrow}\binom{L-N_\uparrow}{N_\downarrow}$. At half-filling the first sector includes only states where every site is occupied by either a spin-up or a spin-down particle. The expression simplifies to $\binom{L}{L/2}$ basis states.

This means that at half-filling no hopping process without construction of a doubly occupied site is possible. Thus, the Hamiltonian of the first sector is exactly zero.

Tab. 3.7 shows the computation time of the NPT approach in comparison to the standard ARPACK routine.

		NPT			ARPACK		
	N_b	t_{tot} [s]	t_{its} [s]	E_{NPT}	t_{tot} [s]	t_{its} [s]	E_0
$L = 8$, $N_\uparrow = N_\downarrow = 4$	4900	0.10	0.06	-2.15497	0.07	0.04	-2.17668
$L = 10$, $N_\uparrow = N_\downarrow = 5$	63504	1.13	0.56	-2.65938	1.49	0.87	-2.70369
$L = 12$, $N_\uparrow = N_\downarrow = 6$	853776	17.74	8.37	-3.15762	31.76	22.03	-3.23238

Table 3.7: Comparison between NPT and a traditional Arnoldi eigensolver (ARPACK) for different systems with $U = 10$ and periodic boundary conditions. The accuracy was set to 0.01 (t_{tot}: total time, t_{its}: iteration time).

3.11 Comparison to reduced models

Standard projection techniques lead to the derivation of the tJ-model and the Heisenberg model from the more complicated Hubbard model. By using these models the Hilbert space of a system can be reduced drastically. The tJ-model describes the system by using only basis states without double occupancies. It reduces to the Heisenberg model at half-filling which is solely a spin-model. The effective spin-spin interaction is connected to the Hubbard-U by

$$J = \frac{4t^2}{U} \ .$$

In terms of the present framework, these models correspond to a projection of the second sector with $N_\kappa = 1$ to the primary sector with $N_\kappa = 0$. Furthermore, the non-linear eigenproblem is replaced by a linear problem with $\lambda = 0$.

Tab. 3.8 shows a comparison between the solution of the Hubbard model, the corresponding tJ/Heisenberg model, and the NPT approach for different model systems As can be seen the approximation of NPT surpasses the reduced models.

3.12 Three-partition recursion

Consider as a toy model the following 3-partitioned potential ordered matrix:

$$\begin{pmatrix} A & E & 0 \\ E^\dagger & B & F \\ 0 & F^\dagger & C \end{pmatrix} ,$$

system	model	E_{NPT}	$\frac{E-E_0}{E_0}$
$L = 8$, $N_\uparrow = N_\downarrow = 4$, $U = 10$	Hubbard	-2.1767	
	NPT ($\tilde{p}_1 = 20$)	-2.1526	0.0110
	Heisenberg	-2.2604	0.0385
$L = 10$, $N_\uparrow = N_\downarrow = 5$, $U = 10$	Hubbard	-2.7037	
	NPT ($\tilde{p}_1 = 26$)	-2.6572	0.0172
	Heisenberg	-2.8062	0.0379
$L = 10$, $N_\uparrow = N_\downarrow = 4$, $U = 10$	Hubbard	-5.6698	
	NPT ($\tilde{p}_1 = 80$)	-5.7303	0.0107
	tJ	-5.5282	0.0250
$L = 10$, $N_\uparrow = N_\downarrow = 5$, $U = 5$	Hubbard	-4.9334	
	NPT ($\tilde{p}_1 = 26$)	-4.6032	0.0669
	Heisenberg	-5.6123	0.1376

Table 3.8: Comparison of the lowest eigenvalue obtained by NPT with those of the Heisenberg or tJ-model and the exact result for the original Hubbard model (pbc., $N_\kappa^{\mathrm{ini}} = 1$).

where A, B, and C correspond to the inner Hamiltonians of the three partitions, respectively. E.g. B includes all states with one occupied nearest neighbor pair, C all with two. E and F are the inter-partition hoppings which form an additional particle pair to contribute to the potential.
The first approximation is to use only A:

$$(A - \lambda I)\vec{x} = 0 \; .$$

A second approximation is to use the Schur complement transformation as explained in the previous sections:

$$\left(A - \lambda I - E(B - \lambda I)^{-1} E^{\dagger}\right) \vec{x} = 0 \; .$$

A further extension is to include the third partition:

$$\left(A - \lambda I - E \left(B - \lambda I - F(C - \lambda I)^{-1} F^{\dagger}\right)^{-1} E^{\dagger}\right) \vec{x} = 0 \; . \tag{3.12}$$

Note that this is still exact, no approximation has been applied. The result can be obtained by applying a Schur complement transformation regarding B first and then on A.

3.12.1 Expansion of C -contribution

Consider the inner expression of eq. 3.12:

$$\left(B - \lambda I - F(C - \lambda I)^{-1} F^{\dagger}\right)^{-1} \; .$$

Here, the term $F(...)^{-1}F^{\dagger}$ is usually small compared to $B - \lambda I$ because it contains a larger partition with a larger potential term. Use this to form the expansion:

$$\begin{aligned}
&\left(B - \lambda I - F(C - \lambda I)^{-1} F^{\dagger}\right)^{-1} = \\
&\qquad (B - \lambda I)^{-1} \\
&\qquad + (B - \lambda I)^{-1} F (C - \lambda I)^{-1} F^{\dagger} (B - \lambda I)^{-1} \\
&\qquad + (B - \lambda I)^{-1} F (C - \lambda I)^{-1} F^{\dagger} (B - \lambda I)^{-1} F (C - \lambda I)^{-1} F^{\dagger} (B - \lambda I) \\
&\qquad + \mathcal{O}(C^{-3}) \; .
\end{aligned} \tag{3.13}$$

Up to the linear inverse part regarding C the full problem reads

$$\left(A - \lambda I - E\,(B-\lambda I)^{-1}\,E^\dagger - E(B-\lambda I)^{-1}F(C-\lambda I)^{-1}F^\dagger(B-\lambda I)^{-1}E^\dagger\right)\vec{x} = 0\,.$$

3.12.2 Expansion of the kinetic part

The kinetic part of B and C is considered small compared to the potential part.

$$E(B-\lambda I)^{-1}E^\dagger = \sum_{\nu=0}^{\infty}\frac{1}{(B_V-\lambda)^{\nu+1}}E(-\tilde{B})^\nu E^\dagger\,,$$

where B_V is the potential and $\tilde{B}$ is the kinetic part of B.
Using only the inverse linear correction by C the corresponding term reads

$$\begin{aligned} &E(B-\lambda I)^{-1}F(C-\lambda I)^{-1}F^\dagger(B-\lambda I)^{-1}E^\dagger = \\ &\quad\sum_{\nu\mu\rho}\frac{1}{(B_V-\lambda)^{\nu+1}}\frac{1}{(B_V-\lambda)^{\mu+1}}\frac{1}{(C_V-\lambda)^{\rho+1}}E(-\tilde{B})^\nu F(-\tilde{C})^\rho F^\dagger(-\tilde{B})^\mu E^\dagger\,. \end{aligned} \tag{3.14}$$

The first terms of this expansion are

$$\begin{aligned} &\frac{1}{(B_V-\lambda)^2}\frac{1}{(C_V-\lambda)}EFF^\dagger E^\dagger \\ &+\frac{1}{(B_V-\lambda)^3}\frac{1}{(C_V-\lambda)}\left(E(-\tilde{B})FF^\dagger E^\dagger + EFF^\dagger(-\tilde{B})E^\dagger\right) \\ &+\frac{1}{(B_V-\lambda)^2}\frac{1}{(C_V-\lambda)^2}EF(-\tilde{C})F^\dagger E^\dagger \\ &+\mathcal{O}((B_V-\lambda)^{-3}(C_V-\lambda)^{-2}) + \mathcal{O}((B_V-\lambda)^{-2}(C_V-\lambda)^{-3}) \end{aligned}$$

The matrices $E(...)E^\dagger$ appearing in this expansions are usually small and independent of λ. They could be cached for reusing.

3.12.3 Iteration concept

Again, a similar iteration concept as for the more simple case with involving just the next partition can be used. The only new aspect involves the construction of the overall unitary eigenvector approximation, which is used to transform the off-diagonal Hamiltonian blocks of the next partition.

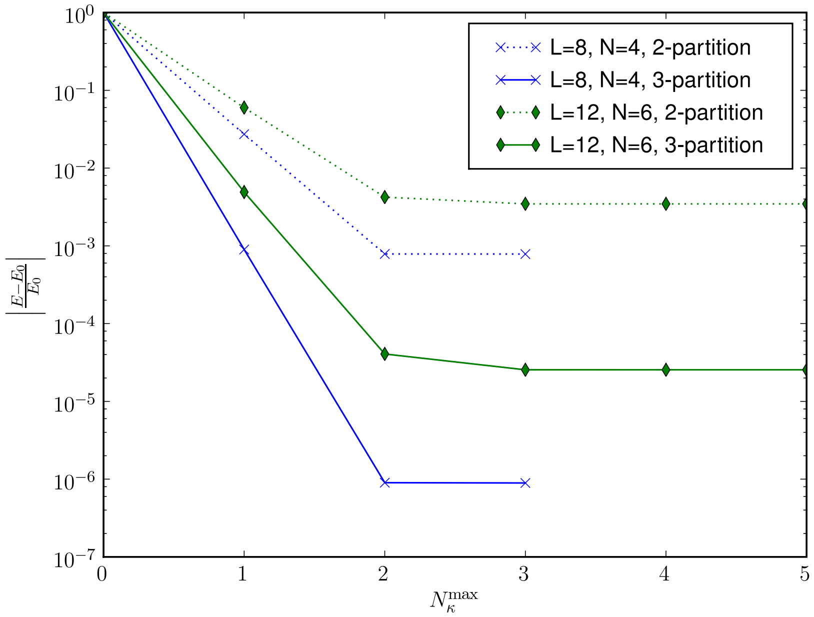

Figure 3.10: Comparison of the 2-partition approach to the 3-partition approach for two different system sizes. The development of the accuracy is given as a function of included sectors $N_\kappa^{\max}$. The values are given as distances to the exact ground state energy.

Note that in the case of involving additional sectors the new approximation is not any more a solution to the eigenproblem of the first two partitions. So the new matrix $\tilde{A}$ is not any more diagonal, and must be calculated additionally in every step.

3.12.4 Numerical experiments

Fig. 3.10 shows results from an actual implementation using the iterative scheme. As can be seen there is no qualitative improvement of including three partitions in every iteration step. What can be seen clearly is the quantitative improvement due to the more exact calculations of the first partitions. But still some saturation is reached after a small number of iterations.

3.13 Conclusions

The Numerical Projection Technique is a generalization of traditional projection schemes with respect to multiple aspects. No approximations are made to the original model and also no limitations are imposed on the number of included sectors. It combines the steps of projection and eigenvalue computation.

Another advantage is, that no effective Hamiltonian has to be formulated which may not be trivial for complex multi-banded systems, e.g. in the spin-orbital model for manganites [36]. Furthermore, also moderate coupling constants can be treated more effectively because higher sectors become more important in this case.

The approach is applicable to other algebraic eigenproblems of the same shape from other research fields. The diagonal entries should mostly be dominant compared to the off-diagonal elements, only a small fraction should have entries close to zero.

4 Multi-Dimensional Approach to the Fermionic Many-Body Problem

This chapter describes an alternative basis for the description of the many-body system. It relies on a description of the fermions on a lattice using their explicit coordinates, but at the same time restricts the coordinate space in order to obey the anti-symmetric property of the corresponding wave function and the Pauli's principle (see sec. 1.4).

4.1 Coordinate space restriction

Remember the anti-symmetric relation for fermions from section 1.4:

$$\Psi(x^{(1)}, x^{(2)}) = -\Psi(x^{(2)}, x^{(1)}) \ ,$$

where $x^{(1)}$ and $x^{(2)}$ represent the coordinates of particle 1 and 2, respectively.
In general, each particle has d coordinates, where d is the spatial dimension. The full coordinate space has $d{\cdot}N$ dimensions with particle number N. The number of possibilities for occupation configurations is

$$L^{d \cdot N} \ ,$$

where L is the number of lattice points in each dimension.
This number is drastically reduced by the property of indistinguishability and the symmetry of the wave function. In the full configuration space many entries of a proper wave-vector have the same entries on different lattice points up to a minus sign.
We can now apply a criterion for choosing points of the full coordinate space in order to obtain a smaller set of points. The value of the wave function for points outside of the small set can be extrapolated from inside.
The criterion for particles in 1D is a simple restriction on the coordinates of the particles which are only numbers in this case:

$$x^{(1)} < x^{(2)} < x^{(3)} < \dots$$
$$x^{(i)} < x^{(i+1)} \; .$$

It is a simple ordering of the individual coordinates. In addition, the Pauli principle implies a boundary condition onto the full coordinate space. At the *surface* where two particles have the same coordinates, the wave function must vanish.
In 2D, the condition reads:

$$x_1^{(i)} < x_1^{(i+1)} \vee \left(x_1^{(i)} = x_1^{(i+1)} \wedge x_2^{(i)} < x_2^{(i+1)} \right) \; .$$

The criterion applies a similar restriction to the first coordinate of each particle. But additionally, these numbers are allowed to be equal for both particles, when the condition is fulfilled for the second pair of coordinates. In general, the coordinates of the individual particles have to be ordered in a *lexicographical* way:

$$x_j^{(i)} < x_j^{(i+1)} \vee \left(x_j^{(i)} = x_j^{(i+1)} \wedge [\text{Criterion for } j \rightarrow (j+1)] \right) \; .$$

A state belonging to the restricted domain can be represented by

$$|\vec{r}_1\vec{r}_2 ... \vec{r}_N\rangle \; ,$$

the coordinates of the particles with $\vec{r}_1 \prec \vec{r}_2 \prec \vec{r}_3 \prec ...\vec{r}_{N-1} \prec \vec{r}_N$ according to a lexicographical ordering.

Example: $(N = 4, \; d = 2)$ Consider the state $|00021022\rangle$. It means that particle 1 has coordinates $(0,0)$, particle 2 has coordinates $(0,2)$, particle 3 has coordinates $(1,0)$, and particle 4 has coordinates $(2,2)$. The coordinates are correct with respect to the lexicographical ordering.
Examples of not valid coordinate sets: $|00121022\rangle$ and $|00020222\rangle$. The criterion is violated.

4.2 Geometric considerations

It is a rather difficult task to interpret the multi-dimensional space geometrically as the dimensionality of the needed space increases linearly with the number of particles. It is

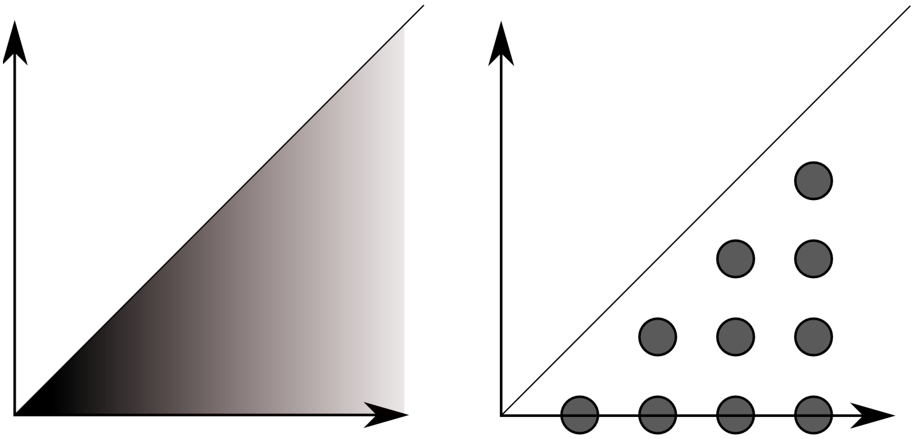

Figure 4.1: Graphical representation of the many-body wave function definition space for a two particle fermionic system in 1D. The axes reflect the positions of the two particles, respectively. **Left:** continuous, **right:** discrete.

only possible for very small particle numbers, e.g. for a system with 2 particles in 1D (see fig. 4.1).

Here, the lexicographical criterion restricts the full coordinate (two-dimensional) space to the space below the diagonal. The boundary condition states, that the wave function at this line must be zero. In the discrete case it can be seen that the points on the diagonal are also omitted, in order to obey the Pauli principle.

It is also possible within the limits of human's imaginativeness to consider three particles in 1D. The restricted space corresponds to a pyramid shaped space (see fig. 4.2).

Unfortunately, this simple cases do not express the complexity of fermionic systems in terms of the exchange interaction. Due to the antisymmetry of the wave function a sign change of the wave function occurs when two particles exchange their positions. This cannot happen in 1D due to model restrictions. So, the simplest model to reflect this phenomenon is a 2D system with 2 particles, which corresponds to a 4-dimensional wave function definition space.

This cannot be pictured easily, but this space can be projected to a three dimensional space using stereographic projection. Here, each direction in 4-dimensional space is reproduced as a point in 3D. Each canonical direction is represented by a corner point of a tetrahedron. The planes correspond to one of the 4-dimensional coordinates being zero. So the system can be drawn and interesting conclusions can be obtained (see fig.

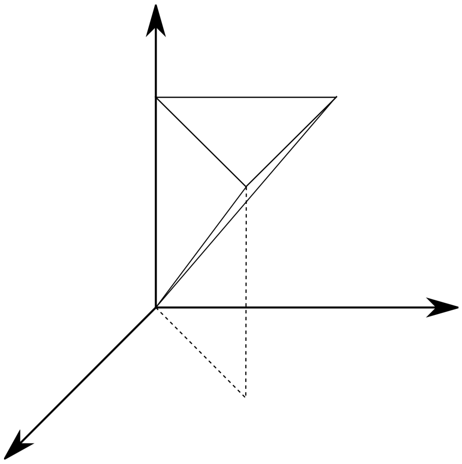

Figure 4.2: Graphical representation of the many-body wave function definition space for a three particle fermionic system in 1D. The axis reflect the positions of the three particles, respectively.

4.3).

Let (x_1, x_2) be the coordinates of the first particle, and (x_3, x_4) be the coordinates of the second particle. This space can be partitioned in two halves where either $x_1 > x_3$ or $x_1 < x_3$. So the lexicographical criterion can be applied and one half is dismissed, i.e. mapped to the other half-space. Furthermore, when both coordinates are equal, which is the case at the intersecting plane of the two halves (see fig. 4.3 right), the secondary criterion for the other two coordinates has to be applied. The plane is once more divided in two sub-sections with $x_2 > x_4$ and $x_2 < x_4$, respectively. One of them can be mapped to another.

4.3 Generation of valid states

In order to apply the restrictions by the lexicographical criterion to build a fermionic wave function an effective algorithm is needed to go through the space of admissible points.

With this method matrix expressions like the vector-matrix-product can be evaluated without the actual need to calculate the matrix itself.

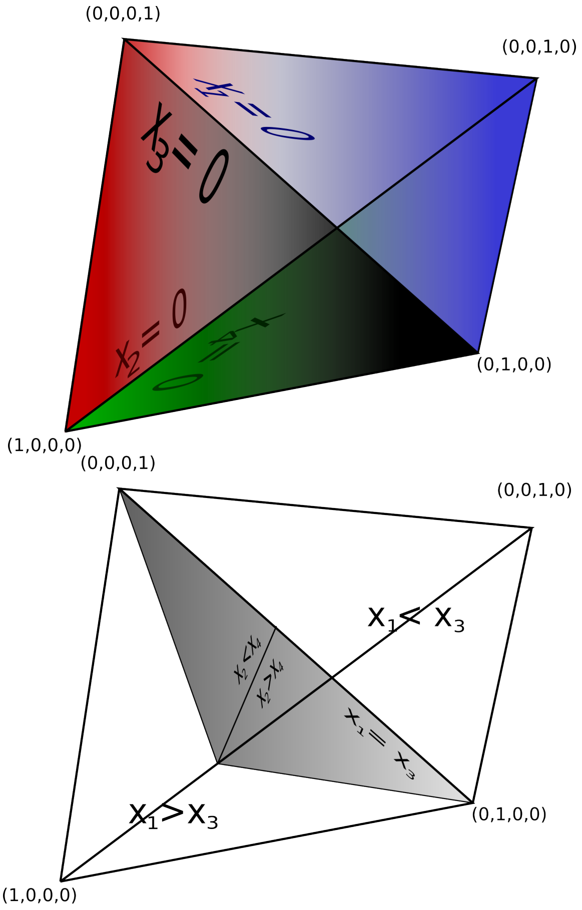

Figure 4.3: Graphical representation of the many-body wave function definition space for a two particle fermionic system in 2D using stereographic projection. **Left:** Each point in three dimensional space corresponds to a direction in 4-dimensional space. **Right:** Application of the fermionic lexicographical criterion to the projection (see text).

Initial state: Since the aim is to go through the state space in a lexicographical way, the *lowest* possible way of preparing the particle configuration is needed.
As an example for $N = 4$, $d = 2$, $L = 10$ the initial state would be $|00010203\rangle$, which means that the first fermion has coordinates $(0, 0)$, the second $(0, 1)$ and so on. So the electrons are prepared to occupy the lowest possible coordinates in an ascending way.

The next state: To go from one state to another is fairly simple. Just increase the most right coordinate of the most right particle: $|00010204\rangle ... |00010299\rangle$.
The next part is more tricky. We have to ensure the lexicographical ordering between particles. In order to do so, we increase the coordinate of the second rightest particle and prepare the coordinates of the rightest particle to be just one step ahead: $|00010304\rangle$. Then increase again the most right coordinate of the most right particle.
Further considerations include the following state: $|00019899\rangle$. Here, increasing the coordinates of the second particle from the right leaves no possibility for the most right to be in. So this state has to be dismissed and the next valid state is: $|00020304\rangle$.

4.3.1 Interpretation of a nearest neighbor interaction

The nearest neighbor interaction model (see sec. 1.7) includes a term punishing states with particles occupying neighboring sites.
In the case of the simple two particle in 1D model the potential is applied to states next to the main diagonal in the full coordinate space (see fig. 4.4).
For the three particles in 1D, the potential appears next to the planes of the pyramid shaped restricted domain (see fig. 4.5).
Fig. 4.6 shows a quantitative analysis of the three-fermion system.

4.4 Mapping between the occupation number basis and the $N \cdot d$-dimensional coordinate representation

Using the criterion defined in previous sections one arrives at basis states for particles which are very similar to the occupation number basis. In the occupation number basis a state is defined with the number of particles at each site. In the case of spin-less fermions this number is either 0 or 1. The occupation number states can be seen as representatives

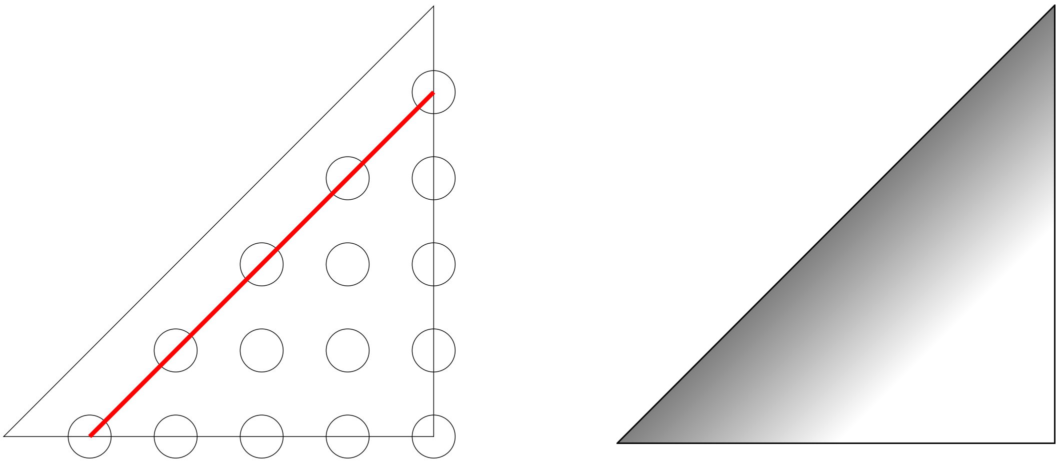

Figure 4.4: Nearest-neighbor potential in a system of 2 particles in 1D. **Left:** discrete, **right:** continuous.

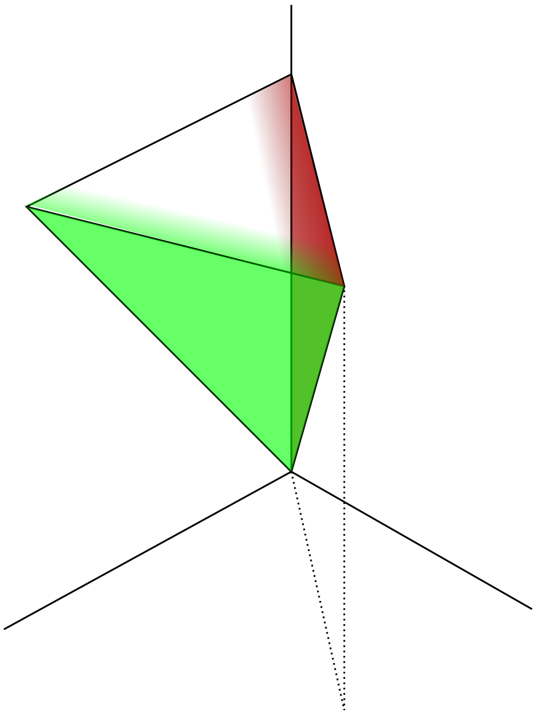

Figure 4.5: Nearest-neighbor potential in a system of 3 particles in 1D.

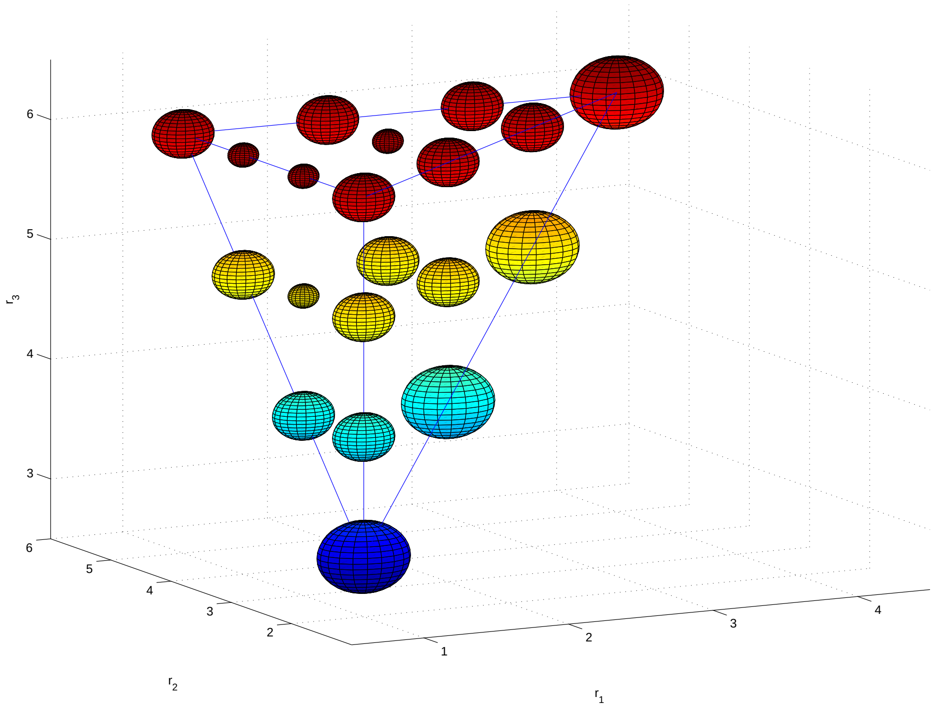

Figure 4.6: Next nearest neighbor potential in the 3-dimensional phase space for a system with 3 particles in 1D ($L = 6$). The radius of the spheres is proportional to the number of nearest neighbor pairs of each set of coordinates.

for many states of the full many-body-state by permuting the coordinates of the involved particles.

It turns out, that the much fewer states described by the lexicographical criterion have a one-to-one relation to the occupation number basis states. Using the basis states defined by the criterion allows to map the occupation basis states on the $N \cdot d$-dimensional many-body-space.

Furthermore we will see, that the matrix representation of the many-body Hamiltonian has a very similar structure compared to an FDM discretization. The symmetries of the wave function translate to a specially shaped computational domain and rather exotic boundary conditions.

4.4.1 Kinetic operator

The hopping term, which appears for example in the Hubbard model (see sec. 1.6) can be mapped exactly to the discretization of the Laplacian in the multi-dimensional picture.

In the case of a discretization of differential equations the Laplacian can be represented by a standard difference star in the discretized space. It is constructed of *legs* to all the canonical neighboring directions. These legs are exactly the same as the hopping paths of the particles in the occupation number basis.

Additional complexity comes into play when looking at systems with at least two fermions in at least two spatial dimensions. Here it may occur that a particle hops and the overall coordinate are not lexicographically ordered anymore. Then the state has to be ordered manually again, a procedure which can be seen as a back-projection into the calculation domain with states obeying the fermionic criterion.

Translated into the language of finite differences this corresponds to a finite-difference star where on some legs an additional minus signs appears. This is of course due to the antisymmetry of the fermionic wave function.

In the case of 2 particles in 2D (see fig. 4.3) this means that the points on one side of the interacting plane have neighbors on the other side. So it can be imagined that the to halves of the plane have to be folded and collapsed to reflect the fermionic boundary conditions.

4.4.2 Potential operator

In the occupation number basis a potential appears on the diagonal of the matrix-representation. This holds also true for the multi-dimensional approach. Here, an additional term on the diagonal can be added for all states.

4.5 Creation of the kinetic FDM-Matrix

In order to create the kinetic part of the FDM-Matrix in the $N\cdot d$-dimensional description, which corresponds to the hopping term in terms of the second quantization, the neighbor relations have to be established. Due to the antisymmetry of the fermionic wave function the geometry of such a system becomes complicated. This especially applies to problems in more than one spatial dimensions, where the particles can swap locations by multiple hopping processes.
The following section describes a way to find all neighboring relations.

4.5.1 Finding neighbors

To find a neighbor of a specific point $|X\rangle$ it does not suffice to change each of the coordinates by one position to get the points $|Y_i\rangle$. When having created such a neighbor, the criterion (sec. 4.1) has to be applied. The point can either be valid, so a correct neighbor relation is established, or it can violate the criterion. In the latter case further two possibilities may occur.
If the coordinates can be rearranged by swaps of the particle positions and ending up with a set of coordinates obeying the criterion also a neighbor relation to this ordered point can be created. In this case due to the antisymmetry an additional sign appears, which is determined by the number of necessary particle swappings. This process corresponds to a back-projection of a point outside the computational domain into the domain.
The other possibility is that two set of particle coordinates are equal which violates the Pauli's principle and the corresponding point has to be dismissed. Then this leg has no neighbor.

4.5.2 Example: creation of FDM-Matrix for 2 particles in 2D

Let $N = 2$,$d = 2$,$L = 2$ where N is the number of fermions, d is the number of spatial dimensions and L is the number of discretization points per dimension.

There are 6 points of the full coordinate space (16 points), which fulfill the criterion (sec. 4.1):

$$|\text{①}\rangle = |0001\rangle\,, |\text{②}\rangle = |0010\rangle\,, |\text{③}\rangle = |0011\rangle\,, |\text{④}\rangle = |0110\rangle\,, |\text{⑤}\rangle = |0111\rangle\,, |\text{⑥}\rangle = |1011\rangle$$

Let's calculate the neighbors of $|\text{②}\rangle = |0010\rangle$:

$$\begin{array}{ccl} |0011\rangle & \rightarrow & |\text{③}\rangle\,\text{OK} \\ |0000\rangle & \rightarrow & \text{INVALID (Pauli)} \\ |0110\rangle & \rightarrow & |\text{④}\rangle\,\text{OK} \\ |1010\rangle & \rightarrow & \text{INVALID (Pauli)} \end{array}$$

In principle 4 different neighbors are possible. These are created by adjusting one of the coordinates by 1. Two of the neighbors turn out to be valid ($|\text{③}\rangle$ and $|\text{④}\rangle$) while two other turn out to violate Pauli's principle.

Now let's calculate the neighbors of $|\text{④}\rangle = |0110\rangle$:

$$\begin{array}{ccl} |0111\rangle & \rightarrow & |\text{⑤}\rangle\,\text{OK} \\ |0100\rangle & \rightarrow & \text{SORT} \rightarrow |0001\rangle = |\text{①}\rangle\,,\ \text{PARITY:}-1 \\ |0010\rangle & \rightarrow & |\text{②}\rangle\,\text{OK} \\ |1110\rangle & \rightarrow & \text{SORT} \rightarrow |1011\rangle = |\text{⑥}\rangle\,,\ \text{PARITY:}-1 \end{array}$$

Again, two perfectly valid neighbors turn up, while the two others can be rearranged to fulfill the criterion. In this case the coordinates of the two particles have to be swapped, so an additional minus sign appears.

Putting all information together the following FDM-Matrix can be evaluated:

$$\begin{pmatrix} 8 & 0 & -1 & 1 & 0 & 0 \\ 0 & 8 & -1 & -1 & 0 & 0 \\ -1 & -1 & 8 & 0 & -1 & -1 \\ 1 & -1 & 0 & 8 & -1 & 1 \\ 0 & 0 & -1 & -1 & 8 & 0 \\ 0 & 0 & -1 & 1 & 0 & 8 \end{pmatrix},$$

The number 8 on the diagonal comes from the usual definition of the finite-difference star, which allows for an additional value 2 for each dimension.

4.5.3 Example: nearest neighbor potential

The nearest neighbor potential appears on the diagonal only. It essentially counts, how many pairs of particles are located next to each other.
Looking at the state $|0111\rangle$ which has coordinates $(0, 1)$ of the first particle and $(1, 1)$ of the second we can see that the two particles are located next to each other.
The nearest neighbor potential adds the following terms to the previous example:

$$\begin{pmatrix} 8+V & 0 & -1 & 1 & 0 & 0 \\ 0 & 8+V & -1 & -1 & 0 & 0 \\ -1 & -1 & 8 & 0 & -1 & -1 \\ 1 & -1 & 0 & 8 & -1 & 1 \\ 0 & 0 & -1 & -1 & 8+V & 0 \\ 0 & 0 & -1 & 1 & 0 & 8+V \end{pmatrix},$$

where V is the value of the potential.
Generally only multiples of V appear as additional terms on the diagonal, but note that some states have no neighboring particles and are not influenced by the potential.
Typical numerical values for V are in the order of $0 - 10$

5 Multigrid

5.1 Introduction

Multigrid [37, 38] was introduced as a technique to solve linear equations of a finite element discretization. The basic idea is to use different lattice spacings to accelerate the evaluation. In this way different regions of the frequency domain can be efficiently treated on a suitable lattice, respectively.

High frequencies of the solution are obtained on a fine grid, while the convergence of low-frequency components can be drastically accelerated by using a coarser discretization (see fig. 5.1).

Note that the coarse grid calculations are numerically much less demanding because of much smaller number of degrees of freedom.

5.1.1 The standard V-cycle

By introducing a V-shaped cycle it is assured that a symmetric operation takes place (see alg. 1). This so-called V-cycle consists of smoothing steps as well as restriction and

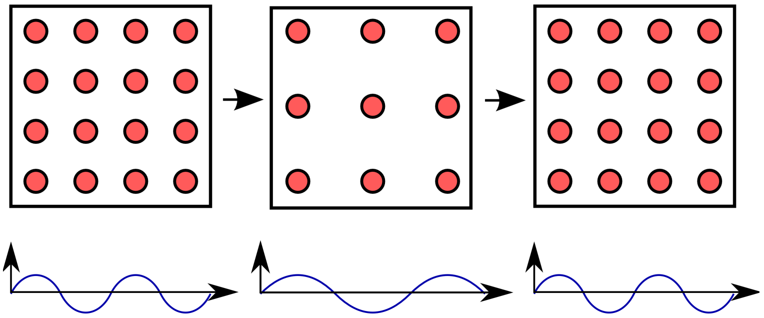

Figure 5.1: Basic idea of multigrid: Solving different frequency regimes of the solution on grids with different lattice spacings.

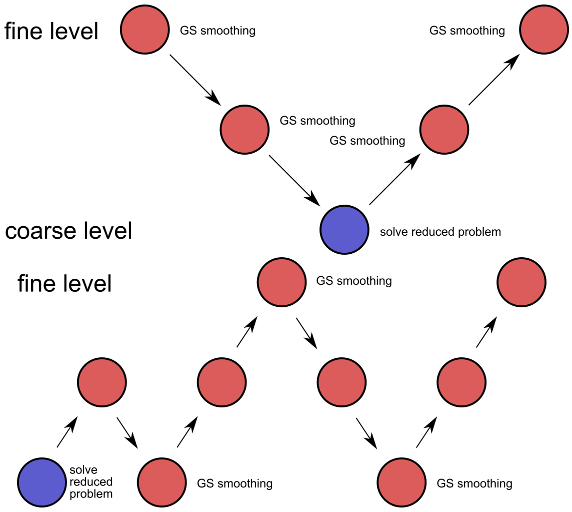

Figure 5.2: **Left:** Standard multigrid V-cycle, **right:** FAS (Full Approximation Scheme).

prolongation processes. Usually, smoothing consists of the application of a few steps of a standard iterative linear solver (e.g. Gauss-Seidel).

The transfer of solutions to coarser grids is called *restriction*, going to finer grids is called *prolongation*.

Algorithm 1 Standard multigrid V-cycle

```
Input: current vector $\vec{v}_l$, current level $l$
Relax $\vec{v}_l$ for $\nu_1$ times on level $l$: $\vec{v}_l \leftarrow \mathcal{R}_l\vec{v}_l$
if $l > 1$ then
    Go to level $l-1$: Restrict $\vec{v}_{l-1} \leftarrow I^l_{l-1}\vec{l}$
    Do standard V-cycle with $v_{l-1}$ on level $l-1$
    Go back to level $l$: Correct $\vec{v}_l$ with $\left(I^l_{l-1}\right)^T \vec{v}_{l-1}$
end if
Relax $\vec{v}_l$ for $\nu_2$ times on level $L$: $\vec{v}_l \leftarrow \mathcal{R}_i\vec{v}_l$
```

A more sophisticated method is the full approximation scheme (FAS, see [37]) which starts at a coarse level and uses consecutive V-cycles while at the same time reaches finer levels (see fig. 5.2 right).

Usually smoothing involves some sweeps of an iterative solver, e.g. a Gauss-Seidel step.

Often a very small number (e.g. 2) cycles is sufficient.
It has advantages to use the transposed restriction matrix for prolongation and vice-versa because it guarantees to have a symmetric operation.

5.2 Solving eigenvalue-problems with multigrid

In principle there are three methods of solving eigenvalue problems with multigrid [39]:

- Eigensolvers using multigrid as a linear solver
- Rayleigh quotient multigrid minimization
- Direct multigrid solvers

The first method relies on a standard recursive technique to update the eigenvalue, while multigrid is only used to smooth the eigenvector. For this purpose a multigrid cycle for the evaluation of linear equations is used.
The Rayleigh quotient method [40] uses multigrid to update the eigenvalue by assisting in searching a minimum value for the Rayleigh-Ritz quotient in a search space $\mathcal{S}$.

$$\lambda^{\text{new}} = \min_{\vec{v}\in\mathcal{S}} \frac{\vec{v}^\dagger A \vec{v}}{\vec{v}^\dagger \vec{v}} \ .$$

A more sophisticated approach is realized by the direct multigrid solvers which update the eigenvalue on all involved lattice levels.

5.3 FAS-based multigrid

A direct multigrid eigenvalue solver may be based on the FAS scheme. The following lines describe a possible realization of such a direct solver [41].
Let the fine grid equation be denoted by

$$A_f V_f - \Lambda_f V_f = \tau_f \ , \tag{5.1}$$

where A_f denotes a $n \times n$ matrix, V_f is an $n \times n_v$ matrix with the eigenvectors in its columns, and Λ is a $n_v \times n_v$ diagonal matrix with the eigenvalues on the diagonal. Without loss of generality it can be assumed that there is an additional right-hand side τ_f which may be set to zero for the finest level.

Let U_f be the current approximation of the exact equation and write down the residual equation:

$$A_f(V_f - U_f) = \Lambda V_f - A_f U_f + \tau_f \ .$$

Let $V_f - U_f =: R_f$ be the error and $\Lambda_f V_f - A_f U_f + \tau_f =: F_f$ be the right-hand side:

$$A_f R_f = F_f \ .$$

This equation is transformed to the coarse level using the restriction operator I_f^c:

$$\begin{array}{rcl} A_c R_c & = & F_c \\ \text{with} \quad R_c & = & I_c^f R_f = I_c^f V_f - I_c^f U_f = V_c - I_c^f U_f \\ F_c & = & I_c^f F_f = \Lambda V_c - I_c^f A_f U_f + I_c^f \tau_f \ . \end{array}$$

This equation can be rewritten in the following form:

$$\begin{array}{rcl} A_c V_c - \Lambda V_c & = & \tau_c \\ \tau_c & = & I_c^f \tau_f + A_c I_c^f U_f - I_c^f A_f U_f \ . \end{array}$$

The last line gives a transformation rule for the inhomogeneity τ. Note that although there may be no such term on a fine level it appears on coarser levels. We arrive at an equation which looks very similar to eq. 5.1, so the above scheme can be applied recursively and therefore be extended to multiple grids.

5.3.1 Updating the fine grid vector

After the cycle on the coarser grid has completed, a correction must by applied to the actual fine grid approximation:

$$\begin{array}{rcl} U_f^{(\text{new})} & = & U_f + R_f = U_f + I_f^c R_c = U_f + I_f^c (V_c - U_c) \\ & = & U_f + I_f^c U_c - I_f^c I_c^f U_f \ . \end{array}$$

For the last line the current approximation of the coarse grid vector U_c and the current fine grid approximation U_f is used.

5.3.2 Eigenvalue update

Note that the eigenvalues Λ are the same on all levels. It can be updated using the Rayleigh-Ritz quotient

$$\Lambda^{(\text{new})} = \frac{(AU - \tau, U)}{(U, U)} \ .$$

5.4 CT-multigrid

Costiner and Ta'asan [42] developed a multigrid scheme for evaluation of multiple eigenvectors and eigenvalues. This algorithm is based on FAS and features eigenvector separation on coarse levels (Generalized Rayleigh-Ritz, see sec. 5.4.1), backrotations to prevent eigenvector mixing, adaptive completion of eigenvalue clusters and selective treatment of individual clusters.

5.4.1 Generalized Rayleigh-Ritz projection

Compared with the coarse grid correction cycle defined in the previous section a Rayleigh-Ritz projection on the fine level proves to be quite a numerical effort. To avoid this the separation of the eigenvectors can be moved to coarser levels by using the Generalized Rayleigh-Ritz projection (GRR, [43]).
Let's assume that the correct eigenvectors can be calculated from the current approximations by multiplying with a square $n_v \times n_v$ matrix E, where n_v is the numbers of wanted eigenvalues.

$$V = U \cdot E \; ,$$

where V is the correct solution, U is the current approximation.
The task now is to find a pair (E, Λ) which solves

$$AUE = UE\Lambda + \tau E \; .$$

To do this, multiply from left by $U^\dagger$

$$U^\dagger(AU - \tau)E = (U^\dagger U)E\Lambda \; . \tag{5.2}$$

This equation turns out to be a small $n_v \times n_v$ generalized eigenvalue problem with the pencil $(U^\dagger(AU - \tau), (U^\dagger U))$ and eigenvalues Λ and eigenvectors E.
Solving this small eigenvalue problem can be used to update the current approximations and the inhomogeneity term:

$$\begin{aligned} U &\leftarrow UE \\ \tau &\leftarrow \tau E \; . \end{aligned}$$

Note that in general the matrix U does not consist of orthonormal eigenvectors on the coarse level. Furthermore the eigenproblem eq. 5.2 does not involve a symmetric matrix because of the inhomogeneity term.

5.4.1.1 Adaptivity

In some cases GRR works increasingly worse for larger numbers of meshes. This may be due to a bad representation of fine level vectors on coarse levels. Also some symmetries existing on the fine meshes may not be represented correctly on the coarse level.
Therefore the GRR has to be moved to finer levels. This can be done automatically by the algorithm when observing the convergence rates.

Algorithm 2 Update of GRR-BR level

```
Start on coarsest level
Track the development of the convergence rate by observing the ratio of residuals of the eigenvalue equation
if convergence fails then
    Lower the level of GRR-BR
end if
```

5.4.2 Backrotations

When performing a GRR projection prevention of permutations, rescalings, and sign-changing have to be taken care of. E.g. for degenerate eigenvalues the matrix E may mix the eigenvectors.

Algorithm 3 Backrotation

```
Sort by eigenvalues in $\Lambda$ and $E$
Determine clusters of eigenvalues
For each diagonal block in $E$ with non-degenerate eigenvalues: bring dominant values to diagonal
Let F be a matrix composed of diagonal blocks of $E$ which correspond to degenerate clusters
Set $E \leftarrow EF^{-1}$
Change signs of $E$ to get positive diagonal
Normalize columns of $E$
```

6 Multigrid – Numerical Experiments

The mapping introduced in section 4.4 provided the motivation to apply the multigrid method (ch. 5) to the problem of strongly correlated fermions. The viewpoint of the multi-dimensional problem which introduces a kinetic matrix identical to the discretization of the Laplace-operator in the $N \cdot d$-dimensional space gives the opportunity to use well-known multigrid methods for eigenvalue problems.

6.1 Restriction/prolongation

The special nature of multi-dimensionality of the many-body eigenvalue problem in solid-state physics makes it necessary to examine special ways of transferring the problem to coarser grids. According to ch. 4, the mapping to a multi-dimensional space results in as much dimensions as there are particles. On the other hand the number of spatial discretization points which corresponds to lattice sites is fairly limited. Having in mind that a reduction by a small number of points per dimension results in a large reduction of overall lattice points and because it is desirable to do a symmetric transformation for all directions it is reasonable to restrict the number of points by one for each dimension. A bisection of the number of points in each dimension, which is quite common for multigrid may lead to such a small coarse level problem which is not able to approximate the fine level solution adequately. In addition such a large reduction is not necessary to benefit from the advantages of multigrid.

6.1.1 Restriction by removing one discretization point per dimension

For every dimension of the multi-dimensional phase space the number of discrete points is reduced by one. Considering a coarse and a fine mesh with the following coordinate representations (see fig. 6.1):

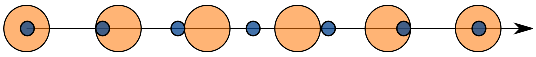

Figure 6.1: Restriction by removing one lattice point per dimension, blue: fine grid points, yellow: coarse grid points.

$$\begin{aligned} x_{i,F} &= \frac{i-1}{N-1} \qquad i = 1, 2, ..N \\ x_{i,C} &= \frac{i-1}{N-2} \qquad i = 1, 2, ..(N-1) \; . \end{aligned}$$

The distances of a fine grid point to the previous and next coarse grid point are

$$\begin{aligned} d_{i,F}^{i-1,C} &= \frac{N-i}{(N-1)(N-2)} \qquad i = 2, 3, ..(N-1) \\ d_{i,F}^{i,C} &= \frac{i}{(N-1)(N-2)} \qquad i = 2, 3, ..(N-1) \; . \end{aligned}$$

A prolongation matrix from N to $N-1$ discretization points reads:

$$I_{N,N-1}^{1d} = \begin{pmatrix} 1 & & & & & & \\ & \frac{N-3}{N-2} & \frac{1}{N-2} & & & & \\ & & \frac{N-4}{N-2} & \frac{2}{N-2} & & & \\ & & & \ddots & \ddots & & \\ & & & & \frac{1}{N-2} & \frac{N-3}{N-2} & \\ & & & & & & 1 \end{pmatrix} \; .$$

For multiple dimensions, these projection matrices have to be applied on each real-space coordinate:

$$I_{N^n,(N+1)^n}^{nD} = \bigotimes_{i=1}^{n} I_{N,N+1}^{1d} \; .$$

In terms of matrix elements the above equation becomes:

$$v_{i_1,i_2,\ldots i_n} = \sum_{j_1} \sum_{j_2} \cdots \sum_{j_n} \left(I^{1D}\right)_{i_1,j_1} \left(I^{1D}\right)_{i_2,j_2} \cdots \left(I^{1D}\right)_{i_n,j_n} u_{j_1,j_1,\ldots j_n} \; .$$

Here, the indices i and j denote the multi-dimensional coordinates of the points u and v, respectively.

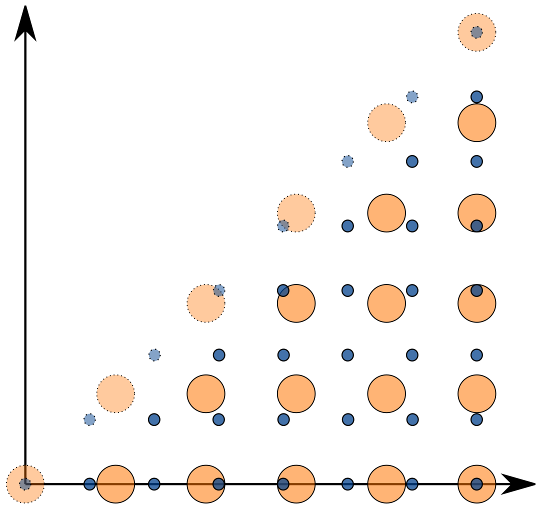

Figure 6.2: Restriction of a two particle system observing the wave function symmetry.

Fig. 6.2 shows a system of 2 particles and a possible coarse grid discretization. Here, the fermionic property reduces the definition space to the triangular region.

6.1.2 Restriction/prolongation by Galerkin ansatz

Let the generalized eigenvalue problem on a fine level be given by

$$A^f x^f = \lambda S^f x^f \ . \tag{6.1}$$

Generally, to retain symmetry, when going to coarser grids the prolongation matrix $I_f^c = (I_c^f)^T$ is used. The eigenvalue equation is transformed to the coarser level:

$$(I_f^c)^T H_f I_f^c x_c = \lambda (I_f^c)^T I_f^c x_c \ .$$

Using the definitions

$$\begin{aligned} H_c &= (I_f^c)^T H_f I_f^c \\ S_c &= (I_f^c)^T S_f I_f^c \ , \end{aligned}$$

one arrives at a coarse grid eigenvalue formulation

$$H_c x_c = \lambda S_c x_c \ ,$$

which looks exactly like eq. 6.1. So the transformation can be applied recursively to coarser grids. Note that if the fine grid equation is a standard eigenvalue problem, the coarse grid problems will result in a generalized problem using this approach.

6.1.3 Restriction using a singular value decomposition

Using a singular value decomposition decomposition a standard eigenvalue problem on fine grids can be transformed to a standard eigenvalue problem on coarser grids. Starting from the fine grid eigenvalue equation ($H_f x_f = \lambda x_f$) and transferring it again to the coarse grid the following expression with an operator I to be determined should be minimized:

$$\frac{x_c^T I^T H_f I x_c}{x_c^T I^T I x_c} \; .$$

It corresponds to the generalized eigenvalue problem

$$H_c x_c = \lambda S_c x_c \qquad \text{with} \quad H_c = I^T H_f I, \; S_c = I^T I \; .$$

Now, reformulate the coarse grid equation to a standard eigenvalue problem by multiplying with $S_c^{-1/2}$ from the left:

$$S_c^{-1/2} H_c S_c^{-1/2} S_c^{1/2} x_c = \lambda S_c^{1/2} x_c \; ,$$

which corresponds to a standard eigenvalue problem with $\tilde{x}_c = S_c^{1/2} x_c$ and $\tilde{H}_c = S_c^{-1/2} H_c S_c^{-1/2}$:

$$\tilde{H}_c \tilde{x}_c = \lambda \tilde{x}_c \; .$$

Finally we can create a projection operator $\tilde{I}$ which directly translates to the coarse standard eigenvalue problem

$$x_f = I x_c = I S_c^{-1/2} \tilde{x}_c = \tilde{I} \tilde{x}_c \qquad \text{with} \quad \tilde{I} = I S_c^{-1/2} \; .$$

$\tilde{I}$ can be calculated by using an SVD-decomposition of I:

$$\begin{aligned}
I &= U D V^T \\
S &= I^T I = V D U^T U D V^T = V D^2 V^T \\
S^{-1/2} &= V D^{-1} V^T \\
I S^{-1/2} &= U D V^T V D^{-1} V^T = U V^T \; .
\end{aligned}$$

Here, the indices c,f have been dropped temporarily. It follows

$$\tilde{I} = UV^T ,$$

which can be used for prolongation and restriction.

6.2 Finding a coarse grid formulation of the eigenvalue problem

The special needs of the fermionic problem, especially the multi-dimensionality and the symmetry considerations make it necessary to look for an adjusted transfer of the fine grid problem to coarser levels.
There exist a number of possibilities for restriction schemes as listed in the following paragraphs.

Simple coarse grid discretization (SIMPLE) This approach allows to build the coarse grid matrix with the same procedure as the fine grid. So no extra computation has to be performed and the matrices can be evaluated independently. Furthermore the sparsity of all level matrices is guaranteed.

Galerkin-ansatz This strictly algebraic ansatz calculates coarser grids from the fine grid matrix using the restriction/prolongation operators (see sec. 6.1.2).

$$A_c = I^T A_f I .$$

General eigenvalue problem on coarse level (GENEIG) When using the Galerkin ansatz for eigenvalue-problems the problem becomes a generalized eigenvalue problem on the coarse level:

$$A_c \vec{x}_c = \lambda S x_c \qquad \text{with} \quad S = I^T I .$$

A possibility is to solve this eigenvalue problem on coarser grids directly.

Using SVD to get a standard eigenvalue problem on coarse levels (SVD) In order to obtain a standard eigenvalue problem on coarser levels SVD (Singular Value Decomposition) can be used (see sec. 6.1.3). This leads to modified restriction/prolongation operators.

$$A_c\vec{x}_c = \lambda x_c \qquad \text{using} \quad \tilde{I} = UV^T$$

Use Lumping to simplify the generalized eigenvalue problem (LUMPING) The matrix on the right side may be simplified using *Lumping*. Here, the matrix is replaced by its diagonal matrix.

$$S_L = \text{diag}(S)$$

Use optimized prolongation/restriction matrices (LOADED1D) This procedure was applied for comparison only. Optimized prolongation matrices were used, which minimize the Rayleigh-Ritz quotient for the exact ground state. Due to the vast numerical effort to find these optimized projection matrices, this method is not practicable.

6.2.1 Comparison

To evaluate the quality of a coarse grid formulation the following testing procedure was performed:

- Calculate an eigenvector of the coarse grid using a Davidson-Lui algorithm up to machine precision.
- Transfer the solution to the fine grid using the prolongation operator.
- Compare number of necessary fine grid iterations to reach a level of convergence as a measure of accuracy of the coarse grid approximation.

For a description of the Davidson algorithm see app. F.
As can be seen in fig. 6.3 and 6.4 all methods provide a better performance than the direct calculation on the fine grid (DIRECT), starting with an arbitrary vector. Furthermore, as expected, the optimized projection operators have the lowest number of fine grid iterations. The interesting point is, that using the coarse grid discretization (SIMPLE) leads to quite good performance. This is also interesting when going to the CT-type eigensolvers (sec. 5.4). The error bars have been devised by the statistics of multiple runs.

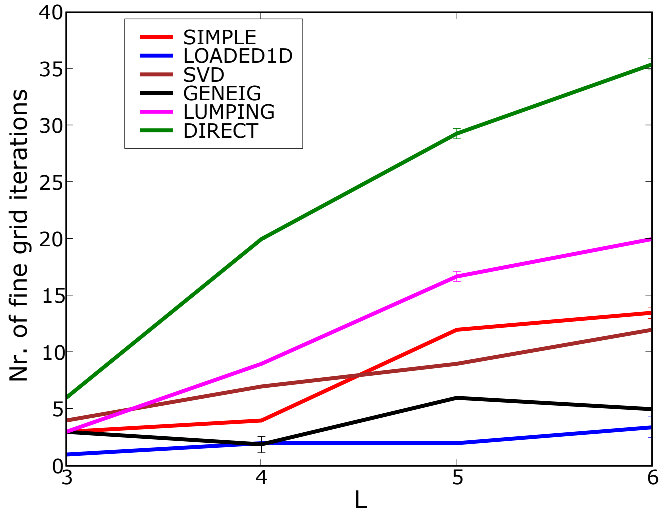

Figure 6.3: Comparison of different coarse grid formulation procedures ($d = 2$, $N = 2$, $V = 0$). Pictured is the number of necessary fine grid iterations to reach convergence over the system size per dimension L.

6.3 CT-multigrid: benchmarks

To compare the CT-multigrid-Ansatz with a traditional Davidson algorithm multiple test series were performed. Fig. 6.5 shows the iterations of a CT-type multigrid calculation for a specific system of fermions. The convergence towards the true value can clearly be seen.

Note that the tests presented in this section are at *half-filling* which is a critical physical configuration.

Further note that all CPU times were measured during the actual iterations without preparation time. Also the numerical experiments including a nearest neighbor potential were done using optimal coarse potential parameters found empirically.

The almost constant number of iterations with system time for $V = 0$ (see fig. 6.6) is clearly visible. Fig. 6.7 shows that the computational complexity between the two algorithms differs only by a constant factor (as expected, see appendix M). The CPU time shows a very limited improvement to Davidson, this also holds true for $V \neq 0$ (fig. 6.8).

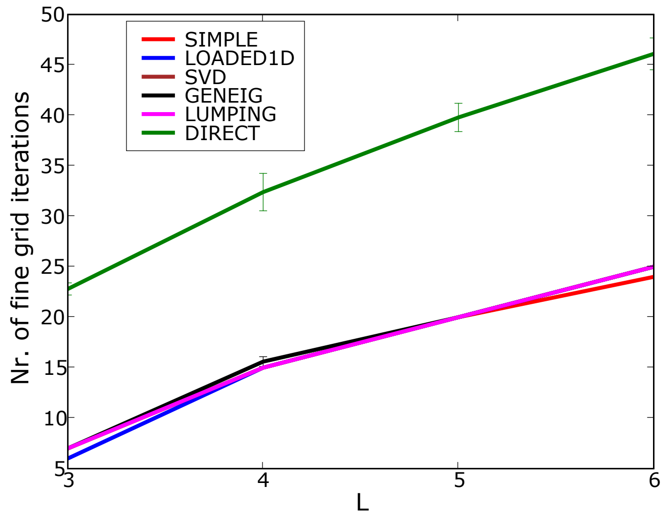

Figure 6.4: Comparison of different coarse grid formulation procedures with included potential ($d = 2$, $N = 2$, $V = -3$).Pictured is the number of necessary fine grid iterations to reach convergence over the system size per dimension L.

6.4 Difficulties

Applying a repulsive neighbor potential the ground state of the system becomes a state with alternating occupied and unoccupied sites. The problem is now to represent this state on coarser levels with a smaller amount of sites. From the perspective of the physicist this becomes then a conversion between systems of different particle density because the number of particles stays the same.

In terms of the multi-dimensional representation (chapter 4) such a ground state could be described with a peaked wave function in terms of the basis function. Such a state is of course difficult to represent on coarse grids, especially if the peaked region is not represented by points on the coarse grid.

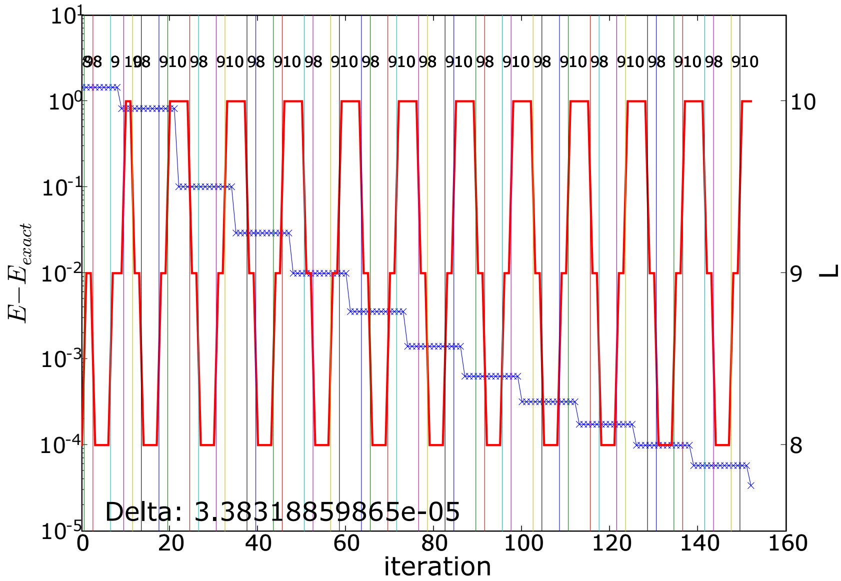

Figure 6.5: 3-level CT-multigrid, development of the lowest eigenvalue over the iterations ($d = 1$, $N = 3$, $L = 10$, $V = 0$). The red line indicates the current lattice size, and the blue marks indicate the current eigenvalue approximation measured from the exact ground state. Note the logarithmic scale.

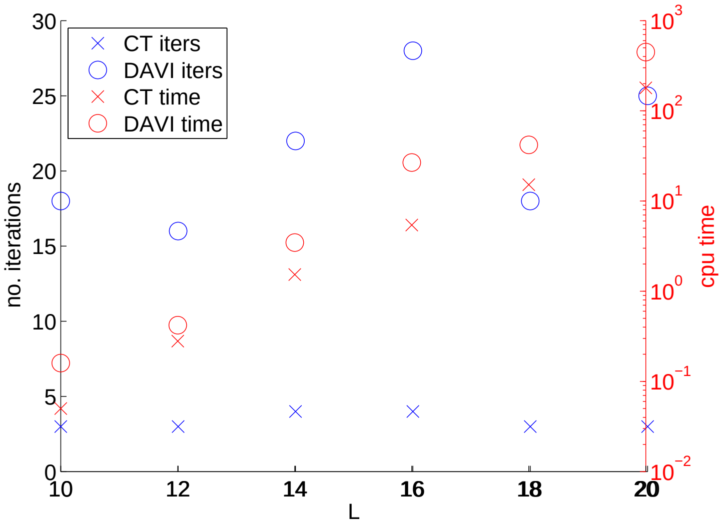

Figure 6.6: Comparison between CT-type multigrid and Davidson algorithm for $V = 0$ at half-filling in 1D.

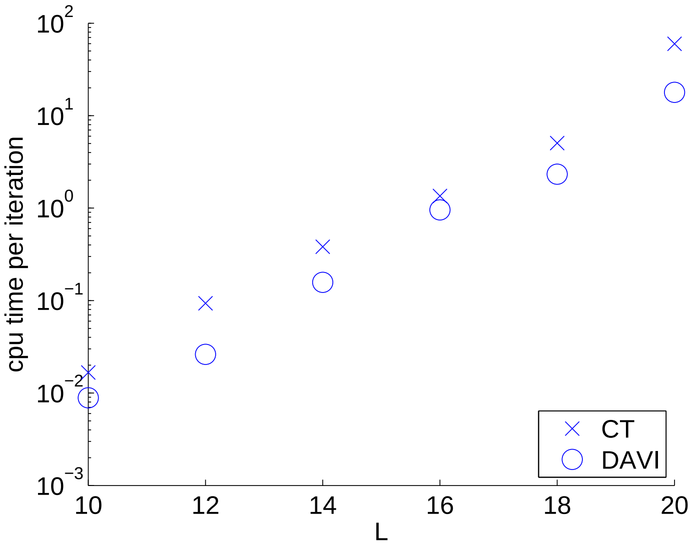

Figure 6.7: Comparison between CT-type multigrid and Davidson algorithm for $V = 0$ at half-filling in 1D. CPU-time per iteration.

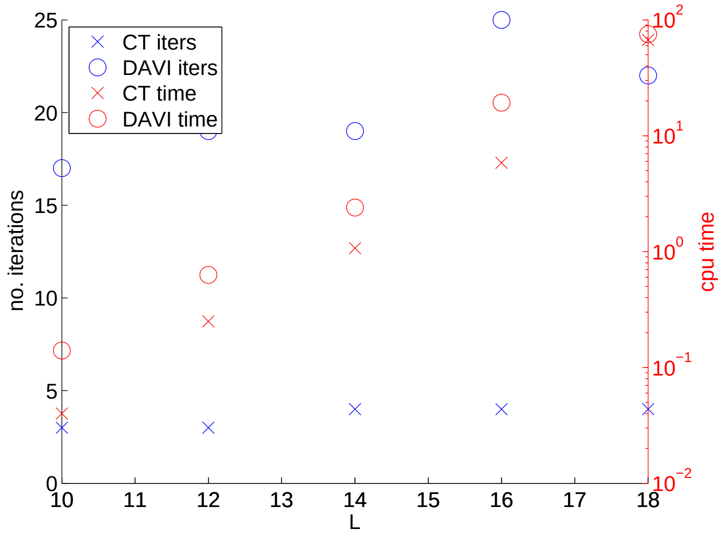

Figure 6.8: Comparison between CT-type multigrid and Davidson algorithm for $V = 3$ at half-filling in 1D.

7 Conclusions

Bridging the communities of computational mathematics and physics can be difficult at times. First of all, one has to find a common language between scientists from different research field. Doing that it becomes clear that in many cases similar problems appear and a benefit from the collaboration seems obvious.

But there is one crucial point which relativizes the gain from the interdisciplinary work in this case. It is the question of the goals of the work. Physicists as explorers for new effects want to understand principle tendencies of measurement curves resulting from experiments. It is in many cases more important to get an impression of the solution rather than high-level accuracy up to the tenth decimal place. The scientist knows that there are many other effects when doing a quantitative analysis of an experiment, which contribute to the outcome. This is not only due to the limitations of the experiments but also to the limitation of the used numerical model in the computations. Therefore, the conclusion is that an algebraic exact calculation does not make sense in many cases. Furthermore, numerical computation in terms of many-body physics leaded to many very specialized algorithms which deliver impressive results. An algebraic formulation of the problem, and then using standard mathematical algorithms is hugely out of range of these methods in terms of performance.

In the course of this thesis, a new specialized algorithm for solving eigenproblems of strongly correlated systems emerged from the adaption of AMLS (Automated Multilevel Substructuring, see app. B) to the problem of many strongly correlated particles. The major break-through is certainly the decomposition of the individual partitions in a tensorial matrix structure which makes sure that the problem can be solved at the numerical complexity of the subsystems. This of course is in opposition to the original idea of an algebraic partitioning of the algorithm. Nevertheless, it can be seen that the algorithm TSGSA works and delivers an impressive numerical efficiency.

It is still to be determined, how TSGSA relates to other, similar methods like DMRG or VCA, which are highly successful methods for strongly correlated systems. In any way

the least it does is throwing new light on established algorithms.
The Numerical Projection Technique (NPT) can not profit from an exponential speed-up as above, but it may be interpreted as an extension to traditional projection techniques in form of an algebraic solver. It was shown that the resulting accuracy exceeds the simple projections by orders of magnitudes in some cases.
The interpretation of the many-body wave function as a similar object than the discrete solution of an FD/FE algebraic problem brings many difficulties with it. Of course, there is the obvious problem of the fermionic sign which does certainly not appear in e.g. continuous material problems. Secondly, the resulting wave vector is not a smooth function. In the extreme case of large repulsion between particles and periodic boundary conditions at half-filling there are only two entries in the ground state vector. This solution cannot be approximated with coarse grid methods to a satisfying extend.

A Derivation of the Hubbard model from a continuous description

A.1 Continuous formulation

Consider the many-body Hamiltonian (see ch. 1):

$$\begin{aligned} H &= H_{\text{kin.}} + H_{\text{pot.}} \\ &= -\sum_i \frac{\nabla_i^2}{2m} + U' \sum_{ij} \delta(\vec{r}_i - \vec{r}_j) \ . \end{aligned}$$

Let the solution of $H_{\text{kin.}}$ in real space be $\phi_n(r)$:

$$H_{\text{kin.}} \phi_n(r) = \epsilon_n \phi_n(r) \ .$$

The ansatz function for the full system with interaction is a Wannier basis set:

$$\psi_{nk}(r) = \frac{1}{\sqrt{N_i}} \sum_{j=1}^{N_i} e^{ikR_j} \phi_n(r - R_j) \ .$$

A.2 Transformation of the operators

The transformation of a single-particle operator to a second quantization formulation (sec. 1.5.4) is given by

$$A^{(1)}_{2Q.} = \sum_{i\sigma, j\sigma'} \langle i\sigma | \, A^{(1)} \, | j\sigma' \rangle \, a^\dagger_{i\sigma} a_{j\sigma'} \ .$$

The transformation of a two-particle operator reads

$$A^{(2)}_{2Q.} = \sum_{i\sigma, j\sigma', k\sigma'', l\sigma'''} \langle i\sigma, j\sigma' | \, A^{(2)} \, | k\sigma'', l\sigma''' \rangle \, a^\dagger_{i\sigma} a^\dagger_{j\sigma'} a_{k\sigma''} a_{l\sigma'''} \ .$$

To transfer the kinetic term to the formulation of second quantization the following steps have to be performed:

$$\begin{aligned}
H_{kin.,2Q.} &= \sum_{i\sigma,j\sigma'} \langle i\sigma| H_{kin.} |j\sigma'\rangle\, a_{i\sigma} a_{j\sigma'} \\
\langle i\sigma| H_{kin.} |j\sigma'\rangle &= \sum_{k\sigma_1} \langle i\sigma| H_{kin.} |k\sigma_1\rangle \langle k\sigma_1| |j\sigma'\rangle \\
&= \sum_{k,\sigma_1} \epsilon_k \langle i\sigma|k\sigma_1\rangle \langle k\sigma_1|j\sigma'\rangle \\
\langle i\sigma|k\sigma_1\rangle &= \int d^3r\, \langle i\sigma|r\rangle \langle r|k\sigma_1\rangle \\
&= \int d^3r \phi^*(r-R_i)\psi_k(r)\delta_{\sigma,\sigma_1} \\
&= \frac{1}{\sqrt{N_i}}\delta_{\sigma,\sigma_1} \sum_m e^{ikR_m} \int d^3r \phi^*(r-R_i)\phi(r-R_m) \\
&= \frac{1}{\sqrt{N_i}}\delta_{\sigma,\sigma_1} \sum_m e^{ikR_m} \delta_{i,m} \\
&= \frac{1}{\sqrt{N_i}}\delta_{\sigma,\sigma_1} e^{ikR_i} \\
\langle i\sigma| H_{kin.} |j\sigma'\rangle &= \frac{1}{N_i} \sum_{k,\sigma_1} \epsilon_k \delta_{\sigma,\sigma_1} e^{ikR_i} \delta_{\sigma',\sigma_1} e^{-ikR_j} \\
&= \frac{1}{N_i}\delta_{\sigma,\sigma'} \sum_k \epsilon_k e^{ik(R_i-R_j)} \ .
\end{aligned}$$

The next step is to transfer the potential term to the framework of second quantization:

$$\begin{aligned}
H_{pot,2Q.} &= \sum_{i\sigma,j\sigma',k\sigma'',l\sigma'''} \langle i\sigma, j\sigma'| H_{pot} |k\sigma'', l\sigma'''\rangle\, a^\dagger_{i\sigma} a^\dagger_{j\sigma'} a_{k\sigma''} a_{l\sigma'''} \\
\langle i\sigma, j\sigma'| H_{pot} |k\sigma'', l\sigma'''\rangle &= \int dr_1 \int dr_2 \langle i\sigma, j\sigma'| H_{pot} |r_1 r_2\rangle \langle r_1 r_2|k\sigma'', l\sigma'''\rangle \\
&= \int dr_1 \int dr_2 v(r_1,r_2) \langle i\sigma, j\sigma'|r_1 r_2\rangle \langle r_1 r_2|k\sigma'', l\sigma'''\rangle \\
&= \int dr_1 \int dr_2 v(r_1,r_2) \phi^*_\sigma(r_1-R_i)\phi^*_{\sigma'}(r_2-R_j) \\
&\qquad \cdot\phi_{\sigma''}(r_1-R_l)\phi_{\sigma'''}(r_2-R_m) \ .
\end{aligned}$$

Set $v(r_1, r_2) = U'\delta(r_1 - r_2)$:

$$\begin{aligned}\langle i\sigma, j\sigma'|\, H_{pot}\, |k\sigma'', l\sigma'''\rangle &= U' \int dr_1 \int dr_2 \delta(r_1 - r_2) \phi^*_\sigma(r_1 - R_i) \phi^*_{\sigma'}(r_2 - R_j) \\ &\quad \cdot \phi_{\sigma''}(r_1 - R_l) \phi_{\sigma'''}(r_2 - R_m) \\ &= U' \int dr_1 \phi^*_\sigma(r_1 - R_i) \phi^*_{\sigma'}(r_1 - R_j) \phi_{\sigma''}(r_1 - R_l) \phi_{\sigma'''}(r_1 - R_m)\ .\end{aligned}$$

The simplest approximation is to neglect all terms other than the on-site repulsion:

$$\langle i\sigma, j\sigma'|\, H_{pot}\, |k\sigma'', l\sigma'''\rangle = U' \int dr \phi^*_\sigma(r) \phi^*_{\sigma'}(r) \phi_{\sigma''}(r) \phi_{\sigma'''}(r) \delta_{i,j} \delta_{j,l} \delta_{l,m}\ .$$

Without loss of generality and considering the Pauli-principle one can derive:

$$\begin{aligned}\langle i\sigma, j\sigma'|\, H_{pot}\, |k\sigma'', l\sigma'''\rangle &= 4U' \delta_{i,j} \delta_{j,l} \delta_{l,m} \int dr \phi^*_\uparrow(r) \phi^*_\downarrow(r) \phi_\uparrow(r) \phi_\downarrow(r) \delta_{\sigma,\uparrow} \delta_{\sigma',\downarrow} \delta_{\sigma'',\uparrow} \delta_{\sigma'',\downarrow} \\ &= 4U' \delta_{i,j} \delta_{j,l} \delta_{l,m} \delta_{\sigma,\uparrow} \delta_{\sigma',\downarrow} \delta_{\sigma'',\uparrow} \delta_{\sigma'',\downarrow} \int dr \phi^*_\uparrow(r) \phi^*_\downarrow(r) \phi_\uparrow(r) \phi_\downarrow(r) \\ &= U \delta_{i,j} \delta_{j,l} \delta_{l,m} \delta_{\sigma,\uparrow} \delta_{\sigma',\downarrow} \delta_{\sigma'',\uparrow} \delta_{\sigma'',\downarrow}\ .\end{aligned}$$

In the end, the potential term becomes:

$$\begin{aligned}H_{pot,2Q.} &= U \sum_{i\sigma, j\sigma', k\sigma'', l\sigma'''} \delta_{\sigma,\uparrow} \delta_{\sigma',\downarrow} \delta_{\sigma'',\uparrow} \delta_{\sigma'',\downarrow} \delta_{i,j} \delta_{j,l} \delta_{l,m} a^\dagger_{i\sigma} a^\dagger_{j\sigma'} a_{k\sigma''} a_{l\sigma'''} \\ &= U \sum_i a^\dagger_{i\uparrow} a^\dagger_{i\downarrow} a_{i\uparrow} a_{i\downarrow} \\ &= U \sum_i n_{i\uparrow} n_{i\downarrow}\ .\end{aligned}$$

The result is the usual form of the single-band Hubbard model

$$H = -t \sum_{<i,j>} a^\dagger_i a_j + U \sum_i n_{i\uparrow} n_{i\downarrow}\ .$$

B The Automated Multi-Level Substructuring method

The Automated Multi-Level Substructuring method (AMLS) was introduced by Bennighof in the field of linear elastodynamics [33, 32, 44]. It was initially applied to frequency response problems as an extension to traditionally Lanczos-based eigenmode calculations.
The method is applied to finite element discretizations of physical objects and relies on partitioning the total domain. It recursively generates subspaces, in which the eigenproblem can be solved more easily.

B.1 Outline

Following Bekas [32], consider an elasto-mechanic, algebraic eigenproblem stemming from a FEM discretization

$$Ax = \lambda x \; .$$

Without loss of generality, divide the domain in two sub-domains and an interface (see fig. B.1). Reorder the unknowns by their domain affiliation to get the following matrix

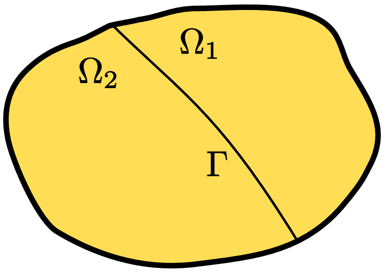

Figure B.1: Domain partitioning of AMLS into two sub-domains Ω_1 and Ω_2, and an interface Γ.

structure:

$$A = \begin{pmatrix} A_1 & 0 & C_{1,I} \\ 0 & A_2 & C_{2,I} \\ C_{1,I}^\dagger & C_{2,I}^\dagger & A_I \end{pmatrix} ,$$

where A_1 and A_2 are the two matrices describing the interior degrees of freedom of the two domains Ω_1 and Ω_2, respectively. A_I includes the degrees of freedom of the interface, $C_{1,I}$ and $C_{2,I}$ describe the couplings between the domains and the interface.
For the further description abbreviate:

$$A_\Omega = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}, \quad C = \begin{pmatrix} C_{1,I} \\ C_{2,I} \end{pmatrix}, \quad A = \begin{pmatrix} A_\Omega & C \\ C^\dagger & A_I \end{pmatrix} .$$

Introduce the Schur complement

$$S = A_I - C^\dagger A_\Omega^{-1} C .$$

Note that calculating the inverse A_Ω^{-1} may not be too difficult, since it is block-diagonal because it consists of the decoupled domains.
Now, form the unitary matrix, which transforms A to block-diagonal form:

$$U = \begin{pmatrix} I & -A_\Omega^{-1}C \\ 0 & I \end{pmatrix} .$$

The transformed eigenvalue equation becomes:

$$U^\dagger A U u = \lambda U^\dagger U u \tag{B.1}$$

$$\begin{pmatrix} A_\Omega & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} u_\Omega \\ u_I \end{pmatrix} = \lambda \begin{pmatrix} I & -A_\Omega^{-1}C \\ -C^\dagger A_\Omega^{-1} & M_S \end{pmatrix} \begin{pmatrix} u_\Omega \\ u_I \end{pmatrix} , \tag{B.2}$$

with

$$M_S = M_I - C^\dagger A_\Omega^{-2} C$$

The coupling terms $-A_\Omega^{-1}C$ and $-C^\dagger A_\Omega^{-1}$ on the right-hand side are neglected and the lowest eigenvalues of the two decoupled problems are evaluated:

$$\begin{aligned} A_\Omega u_\Omega &= \lambda_\Omega u_\Omega \\ S u_I &= \lambda_I M_S u_I . \end{aligned}$$

Note that the eigenproblem of A_Ω can be further decoupled due to its block-diagonality which reduces the numerical effort to the complexity of the sub-domain eigenproblem computation.
Combine the gained eigenvectors (columns of U_Ω and U_I) in a large unitary matrix R.

$$R = \begin{pmatrix} U_\Omega & 0 \\ 0 & U_I \end{pmatrix} ,$$

and perform a Rayleigh-Ritz projection of the full eigenvalue problem B.1:

$$R^\dagger U^\dagger AURx = R^\dagger U^\dagger URx$$

The resulting eigenvalues are approximations to the real eigenvalues of A, the eigenvectors can be obtained as the Ritz vectors URx.
To simplify the eigenvalue computation, the matrix R can be constructed by a limited amount of eigenvectors from the sub-domains, e.g. the vectors corresponding to the lowest eigenvalues. This modal reduction results in a much smaller eigenvalue problem, which is computed much more easily.

C Lanczos

The Lanczos method [23, 28] is an algorithm which is especially suitable for the tridiagonalization of large sparse matrices. A major advantage is that it preserves the sparsity of the matrix during the evaluation as opposed to the dense eigenvalue solvers like QR or Jacobi. In principle, an explicit representation of the matrix is not required for the algorithm but only the evaluation of the matrix-vector product.

The properties of this method include fast convergence to eigenvalues at both ends of the spectrum. In the course of the algorithm a comparable small tridiagonal matrix is calculated which reflects the desired parts of the eigenspectrum very accurately. This small tridiagonal matrix is then easily diagonalized.

C.1 Krylov subspace

An important ingredient of Lanczos is the generation of a Krylov subspace which is defined as:

$$\mathcal{K}_n = \mathrm{span}\{\vec{x}_0, A\vec{x}_0, A^2\vec{x}_0, A^3\vec{x}_0, ...A^{n-1}\vec{x}_0\} \ , \tag{C.1}$$

which is the space spanned by an initial vector $\vec{x}_0$ and the vectors gained by successive application of the matrix A.

To get an orthogonal basis of the Krylov subspace it suffices to orthogonalize the actual computed new vector against just the previous two Krylov vectors. This holds for symmetric matrices A only.

Proof: Let $Q_n = (\vec{q}_0, ..\vec{q}_{n-1})$ be a basis of $\mathcal{K}_n$ obtained by the following scheme:

$$\begin{aligned}
\vec{q}_0 &= \frac{\vec{x}_0}{||\vec{x}_0||} \\
\vec{q}_1 &= \frac{A\vec{q}_0 - (A\vec{q}_0, \vec{q}_0)\vec{q}_0}{||A\vec{q}_0 - (A\vec{q}_0, \vec{q}_0)\vec{q}_0||} \\
\vec{q}_2 &= \frac{A\vec{q}_1 - (A\vec{q}_1, \vec{q}_1)\vec{q}_1 - (A\vec{q}_1, \vec{q}_0)\vec{q}_0}{||A\vec{q}_1 - (A\vec{q}_1, \vec{q}_1)\vec{q}_1 - (A\vec{q}_1, \vec{q}_0)\vec{q}_0||} \\
&\vdots \\
\vec{q}_i &= \frac{A\vec{q}_{i-1} - \sum_{j=0}^{i-1}(A\vec{q}_{i-1}, \vec{q}_j)\vec{q}_j}{\left|\left|A\vec{q}_{i-1} - \sum_{j=0}^{i-1}(A\vec{q}_{i-1}, \vec{q}_j)\vec{q}_j\right|\right|} \ .
\end{aligned}$$

This scheme creates an orthogonal basis for the Krylov subspace. So, all basis vectors up to index j are members of the according Krylov subspace:

$$\begin{aligned}
\forall i \in \{0, ...j\} : &\quad \vec{q}_i \in \mathcal{K}_j \\
\forall i \in \{0, ...j\} : &\quad A\vec{q}_i \in \mathcal{K}_{j+1} \ .
\end{aligned}$$

The second line follows from the construction of $\mathcal{K}_{j+1}$. Suppose that for the basis vector $\vec{q}_{j+2}$ holds $\vec{q}_{j+2} \perp \mathcal{K}_{j+1}$. Thus,

$$\begin{aligned}
\forall i \in \{0, ...j\} : &\quad (A\vec{q}_i, \vec{q}_{j+2}) = 0 \\
\forall i \in \{0, ...j\} : &\quad (\vec{q}_i, A\vec{q}_{j+2}) = 0 \ , \qquad \text{(C.2)}
\end{aligned}$$

which means that the vector $A\vec{q}_{j+2}$ is automatically perpendicular to all vectors with index up to j. The last line follows from the symmetry of A. By induction this holds for all sizes of the Krylov space. □

The so computed vectors give then a basis in which A becomes tridiagonal:

$$Q^\dagger A Q = T \qquad \text{(C.3)}$$

where T is tridiagonal symmetric. From eq. C.2 follows that T is upper Hessenberg, the symmetry is obvious because the transformation conserves the symmetry of A. The values on the secondary diagonal have the form $(A\vec{q}_{j-1}, \vec{q}_j)$ and are non-zero, if the Lanczos vectors $\vec{q}_0, \vec{q}_1, ...\vec{q}_n$ have full rank. The entries on the main diagonal look like $(\vec{q}_j, A\vec{q}_j)$.

To sum things up in a single expression, a new Lanczos vector is constructed by evaluating

$$\vec{q}_{i+1} = \frac{1}{N}(A\vec{q}_i - k_{i-1}\vec{q}_{i-1} - e_i\vec{q}_i) \qquad \text{with} \quad k_{i-1} = (A\vec{q}_i, \vec{q}_{i-1}), \quad e_i = (A\vec{q}_i, \vec{q}_i)$$

Rearranged, we have:

$$A\vec{q}_i = k_{i-1}\vec{q}_{i-1} + e_i\vec{q}_i + k_i\vec{q}_{i+1} \tag{C.4}$$

Here, the normalization constant N from the previous equation was replaced by $k_i = (A\vec{q}_{i+1}, \vec{q}_i)$, because by the symmetry of A this is the same as $(A\vec{q}_i, \vec{q}_{i+1})$. Note that from these considerations one gets an alternative expression for $k_i = ||A\vec{q}_i - k_{i-1}\vec{q}_{i-1} - e_i\vec{q}_i||$;
The equation can also be expressed in matrix form:

$$AQ_j = Q_jT_j + k_j\vec{q}_{j+1}\vec{e}_j^{\dagger}$$

Here, e_i and k_i turn out to be the values on the main and secondary diagonal of T, respectively. $\vec{e}_j$ is the j^{th} canonical basis vector.
The last term is the remainder which vanishes when reaching the full matrix size of A. It only appears in the last column of the matrix equation and it consists of the *next* Krylov vector multiplied by the *next* secondary diagonal entry.
Diagonalization of the tridiagonal matrix $U^{\dagger}TU = D$ gives then good approximations of the eigenvalues. Their accuracy is quickly increased with the dimension of T. The corresponding eigenvectors can be computed using the eigenvectors of the tridiagonal system: QU.

C.2 The Lanczos algorithm

The algorithm does nothing else than constructing the Lanczos vectors $\vec{q}_i$ consecutively and orthogonalizing them against the previous two vectors (see alg. 4).
At any Lanczos iteration the tridiagonal matrix T can be diagonalized to get approximations of the eigenvalues. The computation of the eigenvalues is comparably fast compared to the Lanczos steps, because the matrix T does not become very large.
A possible stopping criterion is to monitor the development of the eigenvalue approximations. If they converge to a fixed value it can be assumed that an eigenvalue has been found.

Algorithm 4 Basic Lanczos algorithm

```
while not converged do
    $\vec{q}_{j+1} \leftarrow A\vec{q}_j$
    $e_j \leftarrow (\vec{q}_{j+1}, \vec{q}_j)$
    $\vec{q}_{j+1} \leftarrow \vec{q}_{j+1} - e_j\vec{q}_j$
    $\vec{q}_{j+1} \leftarrow \vec{q}_{j+1} - k_{j-1}\vec{q}_{j-1}$
    $k_j \leftarrow ||\vec{q}_{j+1}||$
    $\vec{q}_{j+1} \leftarrow \vec{q}_{j+1}/k_j$
end while
```

Also, the entries of the secondary diagonal k_j have to be monitored. If k_j becomes small, an invariant subspace was found, and the current approximations of the lowest eigenvalues already correspond to the lowest exact eigenvalue in this subspace.

It is possible to dismiss all previous Lanczos vectors except the last two to reduce memory consumption of the program. Then the eigenvectors have to be computed in a second run where the Lanczos vectors are created again and summed up accordingly.

A major problem of Lanczos is the loss of orthogonality due to numerical round-off errors. More on this can be found in app. D.

C.3 Properties of the Lanczos algorithm

- When the Krylov-space expands to fill an invariant subspace of the matrix A, the next eigenvector vanishes and the algorithm aborts.
- Due to the problem of losing orthogonality the correct multiplicity of degenerate eigenvalues may not be reflected in the results of Lanczos. Furthermore, although the number of identical eigenvalues is over-estimated the corresponding subspace of eigenvectors may not be exhausted fully.
- With modifications to the algorithm it can be used to calculate finite-temperature properties of materials in solid-state physics [45].

C.4 Calculate dynamic properties with Lanczos

For calculation of the Green's function in solid state physics expressions like

$$\langle\Psi|\, A^\dagger \frac{1}{\omega - H} A \,|\Psi\rangle$$

have to be computed (see sec. 1.8).
Introducing the vector $\vec{x}_0 \hat{=} A\,|\Psi\rangle$ one can use a Lanczos run with initial vector $\vec{x}_0$ using matrix H to approximate above quantity.
The shift is easily introduced:

$$Q^\dagger(\omega - H)Q = \omega - Q^\dagger H Q = \omega - T$$

The inverse of the tridiagonal matrix can serve as an approximation to a Lanczos run using the inverse of H.

$$Q^\dagger(\omega - H)^{-1}Q = (\omega - T)^{-1}$$

Since the latter expression is a matrix in the Lanczos basis only the first element of the first row is required, because $\vec{x}_0$ has been used as initial vector. The value corresponds to the expression $\vec{x}_0^\dagger(\omega - H)^{-1}\vec{x}_0$.

D Lanczos orthogonalization

Although orthogonalization against the last two vectors suffices in perfect arithmetic this may not be true in the limited arithmetic of numerics. Here, quick loss of convergence of the Lanczos vectors appears. For a theory behind that phenomenon see [23]. There are multiple techniques to prevent the loss of orthogonality. They are described below.

D.1 Full orthogonalization

Every vector is orthogonalized against all previous Lanczos vectors. This is a simple but at the same time a very time consuming procedure. Furthermore it has the disadvantage that previous calculated Lanczos vectors have to be kept in memory.

D.2 Selective orthogonalization (SO)

This more complicated approach limits the amount of orthogonalization procedures necessary by monitoring the angle of the actual Krylov vector with Ritz vectors at the actual step. This can be done without actually computing the Ritz vectors, although they have to be computed in order to orthogonalize against them in the case of necessity. This is only done if a certain threshold in the angle is exceeded.

D.2.1 Paige's theorem

This theorem [23] is required to make predictions about the loss of orthogonality.
Consider the Lanczos factorization

$$AV_j + V_jT_j = \vec{r}_j\vec{e}_j^\dagger + F_j \ ,$$

where $\vec{r}_j$ is the residual vector. Furthermore, consider the orthogonality condition

$$I - V_j^\dagger V_j = C_j^\dagger + \Delta_j + C_j \ .$$

F_j, C_j and Δ_j account for round-off effects. Δ_j is diagonal and C_j is strictly upper triangular. Consider the factorization of the triangular matrix:

$$T_j = U_j D_j U_j^\dagger \ ,$$

where D_j contains the eigenvalues on the diagonal. A few further conditions are also needed. Then the theorem states that the Ritz vectors $\vec{y}_i = V_j \vec{u}_i$ satisfy

$$\vec{y}_i^\dagger \vec{v}_{j+1} = \frac{G_{ii}^{(j)}}{\beta_{ji}} \ , \tag{D.1}$$

where $\beta_{ji} = \beta_j u_{ij}$ and $G^{(j)} = V_j^\dagger (K_j + N_j) V_j$ with

$$\begin{aligned} K_j &= F_j^\dagger V_j - V_j^\dagger F_j \\ N_j &= \Delta_j T_j - T_j \Delta_j \ . \end{aligned}$$

Furthermore the theorem states that

$$(\lambda_i - \lambda_k)\vec{y}_i^\dagger \vec{y}_k = G_{ii}^{(j)} \frac{u_{jk}}{u_{ji}} - G_{kk}^{(j)} \frac{u_{ji}}{u_{jk}} - (G_{ik}^{(j)} - G_{ki}^{(j)}) \ .$$

For the proof see Parlett [23, p. 296].

D.2.2 Losing orthogonality

Equation D.1 shows that if β_{ji} becomes small orthogonality is increasingly lost. So especially orthogonality is lost for good approximated eigenvalues (because $||A\vec{y} - \vec{y}\lambda|| = ||(AV - VT)\vec{s}|| = ||(\beta_j \vec{v}_{j+1} \vec{e}_j^\dagger)\vec{s}|| = \beta_j s_{ji} = \beta_{ji}$).

D.2.3 Selection

Monitor β_{ji} to determine, whether an orthogonalization is necessary. This can be done without much numerical effort.
This includes building a set of indices:

$$\mathcal{L}(j) = \left\{ i \,\middle|\, |\cos \angle(\vec{y}_i^{(j)}, \vec{r}_j)| \geq \kappa / \sqrt{j} \right\} \ .$$

A good choice is $\kappa = \sqrt{\epsilon}$ (see [23]). Together with $|G_{ii}^{(j)}| < \epsilon ||A||$ and ϵ the unit round-off this results in:

$$\mathcal{L}(j) = \left\{ i \,|\beta_{ij} < \sqrt{\epsilon} ||T_j|| \right\} \ .$$

Furthermore it is not important to orthogonalize against all vectors determined by that criterion. This is only necessary for Ritz vectors, which are currently becoming more and more approximate to a true eigenvector. Vectors which are already approximating eigenvectors to machine precision have also low β_{ij}, but they need not be considered in the orthogonalization process. Therefore it is proposed to take an interval $j\epsilon||A|| < \beta_{ij} < \sqrt{\epsilon}||T_j||$ as a criterion for necessary reorthogonalizations.

D.2.4 Derivation of the β_{ji} criterion

From Paige's theorem D.1:

$$\cos\angle = \vec{y}_i^\dagger \vec{v}_{j+1} = \frac{G_{ii}^{(j)}}{\beta_{ji}} \; . \tag{D.2}$$

Here we assume that $||\vec{y}_i|| = 1$. Furthermore we assume for the error due to round-off errors:

$$|G_{ii}^{(j)}| < \epsilon||A|| \; ,$$

and the constraint on the angle for selecting indices

$$|\cos\angle| > \frac{\kappa}{\sqrt{j}} \; , \tag{D.3}$$

because that means for all indices:

$$||V_j^\dagger \vec{v}_{j+1}|| = ||Y_j^\dagger \vec{v}_{j+1}|| < \kappa \; .$$

Combining eq. D.2 and D.3:

$$\begin{aligned} \frac{\kappa}{\sqrt{j}} < \frac{G_{ii}^{(j)}}{\beta_{ji}} &< \frac{\epsilon}{\beta_{ji}}||A|| \\ \beta_{ji} &< \frac{\epsilon}{\kappa}\sqrt{j}||A|| \; , \end{aligned}$$

and setting $\kappa = \sqrt{\epsilon}$ and $\sqrt{j}||A|| \approx ||T||$ gives

$$\beta_{ji} < \sqrt{\epsilon}||T|| \; .$$

step j	Local O.	SO	
	$\|\|V_j^\dagger V_j\|\|$	$\|\|V_j^\dagger V_j\|\|$	$\mathcal{L}(j)$
$\vdots$			
21	$1.30 \cdot 10^{-14}$	$1.30 \cdot 10^{-14}$	$\{\}$
$\vdots$			
26	$6.24 \cdot 10^{-11}$	$6.24 \cdot 10^{-11}$	$\{26\}$
27	$5.22 \cdot 10^{-10}$	$6.25 \cdot 10^{-11}$	$\{27\}$
28	$1.95 \cdot 10^{-9}$	$6.40 \cdot 10^{-11}$	$\{28\}$
29	$5.89 \cdot 10^{-9}$	$7.18 \cdot 10^{-11}$	$\{0, 28, 29\}$
$\vdots$			
33	$8.91 \cdot 10^{-5}$	$8.36 \cdot 10^{-10}$	$\{0-15, 29-33\}$

Table D.1: Local orthogonalization vs. selective orthogonalization : Development of orthogonality number $||V_j^\dagger V_j||$.

D.2.5 Example

The example depicted in tab. D.1 used following parameters: Hubbard model $L = 6$, $N_\uparrow = 4$, $N_\downarrow = 1$, $U = 10$, $\vec{v}_{\text{init.}} = (1, 1, ...)^\dagger$, 20 desired eigenvalues, accuracy 10^{-10}, $\kappa = 10^{-7}$. The SO algorithm starts to take effect at iteration $j = 26$ when it orthogonalizes against Ritz vector no. 26. Subsequently, it continues to orthogonalize against Ritz vectors from both ends of the spectrum. While the orthogonality becomes worse for the algorithm with only local orthogonalization, the orthogonality is kept by SO.

step j	$\|Y_j^\dagger V_{j+1}\|$	β_{ij}	$\mathcal{L}(j)$
21	3E-14,4E-15,...4E-14,2E-12	3E-3,6E-2,...8E-3,1E-4	$\{\}$
26	6E-13,5E-14,...3E-12,4E-10	2E-4,3E-2,...1E-4,8E-7	$\{26\}$
29	8E-11,6E-12,...1E-10,5E-16	1E-6,2E-4,...3E-6,1E-8	$\{0, 28, 29\}$
33	5E-16,1E-15,...4E-16,2E-15	3E-13,5E-11,...2E-10,3E-13	$\{0-15, 29-33\}$

Table D.2: Selective orthogonalization. The second column depicts the scalar products of the new Krylov vector with the 2 Ritz vectors from both ends of the spectrum. The third column depicts the corresponding values of β_{ij}.

Tab. D.2 shows the values of the cheaply produced β_{ij} in connection to the overlaps of the newest Krylov vector with the Ritz vectors for the same system. One sees, that the overlap with the last Ritz vector becomes larger, where the corresponding β_{ij} becomes small. So, it helps to determine the necessary orthogonalizations without actually computing the Ritz vectors. Furthermore, the table shows, that after a Ritz vector converged ($j = 33$) the overlap is small, but the β_{ij} becomes very small, too. In this case an orthogonalization is not necessary.

D.3 Post-orthogonalization

Only recently another approach was proposed [46]. Good results were achieved by performing a Lanczos run without orthogonalization and taking into account that the resulting Lanczos basis as well as the eigenvectors do not form an orthogonal basis.
After the run an analysis of the achieved eigenvectors is performed. The vectors are orthogonalized and spurious eigenmodes are removed. With the gained subspace basis they perform a Rayleigh-Ritz procedure on the original matrix.
Although the number of gained eigenvectors is probably reduced, in the end good approximations to the lowest eigenvectors are calculated.
The numerical effort of this method is quite low because it is only applied after the iterations have finished and its complexity is only proportional to the number of wanted eigenvectors and not to the size of the Lanczos basis.

D.4 Periodic reorthogonalization

This method [21] keeps track of mutual orthogonalization of the Lanczos basis using the iterative relation:

$$\begin{aligned} \tilde{\omega}_{j+1,i} &= k_i\omega_{j,i+1} + (e_i - e_j)\omega_{j,i} + k_{k-1}\omega_{j,k-1} - k_{j-1}\omega_{j-1,i} \\ \omega_{j+1,i} &= (\tilde{\omega}_{j+1,i} + \text{sign}(\tilde{\omega}_{j+1,i})2\epsilon_r||A||)/k_j \ , \end{aligned} \tag{D.4}$$

where $\omega_{j,i} = (\vec{q}_j^\dagger, \vec{q}_i)$ are the overlaps between the basis vectors, and ϵ_r is the rounding error. The relation can easily be derived from eq. C.4.
All $\omega_{i,i}$ are set to 1, all off-diagonal $\omega_{i-1,i}$ are set to the rounding error level ϵ_r. When expanding the Lanczos basis, the values for $\omega_{j+1,i}, i = \{1, ..j-1\}$ are calculated using relation D.4, $\omega_{j+1,j} = \epsilon_r$, $\omega_{j+1,j+1} = 1$.

When one element $\omega_{j+1,i}$ exceeds the threshold level, which is usually set to $\sqrt{\epsilon_r}$ the new vector is reorthogonalized to the Lanczos basis vector $\vec{q}_i$ and $\omega_{j+1,i}$ is reset to the rounding error level.

Another possibility is to reorthogonalize against the whole basis, if one of the $\omega_{j+1,k}$ exceeds the barrier. In this case all $\omega_{j+1,i}, i = \{1, ..j - 1\}$ are reset to ϵ_r.

D.5 Numerical results

Tab. D.3 shows a test series of different reorthogonalization approaches for a system with different numbers of wanted eigenvalues. It can be seen that without orthogonalization (first and second column) the orthogonality is lost for larger amounts of wanted eigenvalues. Full reorthogonalization is costly (third column) but the orthogonality is maintained. Selective orthogonalization is very demanding because of the necessary calculation of the eigenvectors within the primary iterations. Also, for the case $n = 20$ orthogonality is not achieved. Post orthogonalization is rather cheap, although in the end some eigenvalues and eigenvectors have to be omitted, and their number is not known before. The advantage is that it is only applied after performing the iterations. Periodic orthogonalization seems to work well in reasonable times.

		w/o orth.	w/o orth., keep	full orth.	SO (keep)	post orth. (n.keep)	periodic orth (keep)
n=2	t_{iters}	0.43	0.44	0.38	0.47	0.49	0.63
	t_{vectors}	0.4	0.0	7 0.06	0.05	0.44	0.07
	t_{Lanczos}	0.83	0.52	0.44	0.52	0.95	0.7
	iters.	90	96	84	84	102	101
	resid.	2E-4	2E-5	3E-4	1E-3	5E-5	5E-5
	ortho.		2E-11	9E-13	1E-12		9E-13
n=10	t_{iters}	3.12	3.1	13.73	28.52	2.55	4.11
	t_{vectors}	7.85	6.51	6.01	5.95	7.2	6.07
	t_{Lanczos}	10.96	9.62	19.74	34.4	7 10.18	10.18
	iters.	276	277	255	249	253	260
	resid.	9E-10	5E-10	7E-14	5E-14	5E-14	6E-14
	ortho.		2.8	2.3E-15	1.7E-6		7.5E-12
	remarks	mult. wrong	mult. wrong			1 dismissed	
n=20	t_{iters}	6.65	6.98	32.34	187.76	7.12	13.01
	t_{vectors}	18.24	17.74	18.91	19.28	19.31	19.69
	t_{Lanczos}	25.29	24.72	51.25	207.04	28.22	32.7
	iters.	356	364	384	405	363	392
	resid.	6E-14	8E-14	7E-14	7E-14	8E-14	8E-14
	ortho.		3.88	3E-15	2.8		1.2E-11
	remarks	mult. wrong	mult. wrong		mult. wrong	4 dismissed	

Table D.3: Overview of different orthogonalization methods for different number of wanted eigenvalues (Hubbard-model, $L = 10$, $N_\uparrow = N_\downarrow = 5$, $U = 10$, $\epsilon = 1E - 5$, diagonalization interval 1, with computation of eigenvectors.)

E Implicitly Restarted Lanczos

The Lanczos method converges quickly to the eigenvalues at the end of the spectrum of a symmetric matrix A but there is in principle no restriction on the size of the necessary basis. Increasingly large numbers of Lanczos vectors require increasing effort to keep them orthogonal.
Implicitly restarted Lanczos (IRL, [47, 21]) restricts the number of Lanczos vectors, say to twice the number of wanted eigenvalues. When this size is reached a transformation on the basis is done and it is truncated to the number of wanted eigenvalues. Afterwards starting from that rotated basis the Krylov iteration is resumed. In this fashion the Krylov basis is iteratively improved to reflect the wanted part of the eigenvalue spectrum.

E.1 Transformation

Let n be the number of wanted eigenvalues and p the number of additional Krylov dimension. The Lanczos factorization after $n+p$ steps looks like

$$AV_{n+p} = V_{n+p}T_{n+p} + \vec{f}_{n+p}\vec{e}_{n+p}^{\,T} \ ,$$

where $\vec{f}_{n+p}$ is a vector in the direction of the next Lanczos vector.
Introducing a shift μ results in

$$(A-\mu I)V_{n+p} = V_{n+p}(T_{n+p}-\mu I) + \vec{f}_{n+p}\vec{e}_{n+p}^{\,T} \ .$$

Using a QR-factorization to produce $T-\mu I = QR$ and multiplying from the right with Q leads to

$$\begin{aligned}(A-\mu I)V_{n+p}Q &= V_{n+p}QRQ + \vec{f}_{n+p}\vec{e}_{n+p}^{\,T}Q \\ AV_{n+p}Q &= V_{n+p}Q(RQ+\mu I) + \vec{f}_{n+p}\vec{e}_{n+p}^{\,T}Q \ .\end{aligned}$$

Using the definitions $\tilde{V}_{n+p} = V_{n+p}Q$ and $\tilde{T}_{n+p} = RQ + \mu I = Q^\dagger T_{n+p}Q$ which leaves $\tilde{T}_{n+p}$ also tridiagonal one obtains

$$A\tilde{V}_{n+p} = \tilde{V}_{n+p}\tilde{T}_{n+p} + \vec{f}_{n+p}\vec{e}^{\,T}_{n+p}Q \ .$$

Note that $\vec{f}_{n+p}\vec{e}^{\,T}_{n+p}Q$ is a matrix with zero entries up to the last two columns. When applying p shifts one ends up with:

$$A\tilde{V}_{n+p} = \tilde{V}_{n+p}\hat{T}_{n+p} + \vec{f}_{n+p}\vec{e}^{\,T}_{n+p}\hat{Q} \ , \tag{E.1}$$

with $\hat{Q} = Q_1...Q_p$ and $\hat{T}_{n+p} = \hat{Q}^\dagger T_{n+p}Q$. Here, $\vec{f}_{n+p}\vec{e}^{\,T}_{n+p}\hat{Q}$ has zero entries up to the n^{th} column. So equating just the first n columns of equation E.1 gives a proper new Lanczos factorization:

$$A\tilde{V}_n = \tilde{V}_n\hat{T}_n + \vec{f}_n\vec{e}^{\,T}_n \ .$$

It can be used to restart a Lanczos run and produce again p new vectors.

E.2 Selection of shifts

As values for the shift μ the eigenvalues of T_{n+p} which are approximations inside the unwanted part of the spectrum can be used. There also exist some more sophisticated choices for shifts [23].

E.2.1 Calculation of orthogonal transformation

To transform the tridiagonal matrix and the Krylov basis the orthogonal matrix Q has to be calculated. In principle a QR factorization is required:

$$T - \sigma_1 I = Q_1 R_1 \ .$$

The fact that T as well as the transformed matrix $\hat{T} = Q^\dagger T Q$ is tridiagonal can be exploited by using a *bulge-chasing* algorithm.
Initially, a plane rotation q_1 is devised which removes the first off-diagonal entry of the matrix $T - \sigma I$. When applying this rotation to T by calculating $q_1^\dagger T q_1$ a bulge emerges. This means an entry in the second sub-diagonal becomes non-zero. This is then repaired by applying another plane rotation q_2 which moves the bulge to the second position of the second sub-diagonal. Multiple applications result in moving the bulge towards the

lower right end of the second sub-diagonal where it eventually vanishes. The matrix $Q = \prod_i q_i$ is then the desired orthogonal transformation and it can be applied to the Lanczos basis.

Note that Q is upper-Hessenberg after the first shift. But using multiple shifts one additional sub-diagonal becomes populated per shift value. This fact can be used to lower the computational effort when transforming the basis.

E.3 Properties of implicitly restarted Lanczos

- Due to the limited size of the Krylov space, the orthogonality problem is reduced drastically.
- If orthogonalization is necessary the numerical effort is smaller than in traditional Lanczos because of smaller amount of vectors.
- The basis can be kept in storage.
- The eigenvector computation is much simpler.
- Largely reduced and bound memory consumption.

E.4 Numerical results

Tab. E.1 shows the dependence of the iteration time on the number of additional Krylov vectors. Note that in the case of small numbers of additional vectors (e.g. 10 in tab. E.1) it is possible that a part of the eigenvalues converge to the other end of the spectrum. This slows down the process drastically. Usually, one takes the number of desired eigenvalues for the dimension of the additional space. This seems like a good compromise.

Tab. E.2 shows that implicitly restarted Lanczos gives results with orthogonal Lanczos basis while at the same time provides a performance boost when compared to Lanczos with full orthogonalization.

Tab. E.3 gives an impression of the implementation at hand compared to a standard ARPACK routine which is based on a similar algorithm.

Add. Krylov dim	iterations	time [s]
5	302	51.4
10	73	20.2
20	34	16.7
30	22	16.3
40	19	18.4

Table E.1: Dependence of IR-Lanczos iterations on additional Krylov space dimension. Hubbard system ($L = 20$, $N_\uparrow = N_\downarrow = 5$, $U = 10$, $n_{ev} = 30$, $\epsilon_d = 10^{-5}$).

		Lanczos	Lanczos w. full reortho.	IR Lanczos
n=2	$t_{\text{iters.}}$	0.35	1.24	1.33
	t_{vectors}	0.04	0.04	0.00
	t_{Lanczos}	0.39	1.29	1.33
	iters.	69	69	17
	probe	0.01	0.01	3E-6
	ortho.		6E-16	1E-15
	remarks			
n=20	$t_{\text{iters.}}$	7.2	31.38	10.83
	t_{vectors}	17.54	18.09	0.98
	t_{Lanczos}	24.74	49.47	11.81
	iters.	357	384	33
	probe	7E-14	7E-14	1E-14
	ortho.	3.9	3E-15	8E-15
	remarks	orth. lost		

Table E.2: Implicitly restarted Lanczos vs. traditional Lanczos ($L = 10$, $N_\uparrow = N_\downarrow = 5$, $\epsilon = 10^{-5}$).

		IR Lanczos	ARPACK
$L = 10,\ N_\uparrow = N_\downarrow = 5$	$t_{\text{iters.}}$	1.33	
n=2	t_{vectors}	0.00	
	t_{Lanczos}	1.33	2.57
	iters.	17	
	probe	3E-6	1E-9
	ortho.	1E-15	
$L = 10,\ N_\uparrow = N_\downarrow = 5$	$t_{\text{iters.}}$	4.65	
n=10	t_{vectors}	0.26	
	t_{Lanczos}	4.91	7.83
	iters.	38	
	probe	1E-14	1E-14
	ortho.	4E-15	
$L = 10,\ N_\uparrow = N_\downarrow = 5$	$t_{\text{iters.}}$	7.42	
n=20	t_{vectors}	1.02	
	t_{Lanczos}	8.44	24.34
	iters.	23	
	probe	2E-14	1E-14
	ortho.	7E-15	
$L = 12,\ N_\uparrow = N_\downarrow = 6$	$t_{\text{iters.}}$	27.67	
n=2	t_{vectors}	0.06	
	t_{Lanczos}	27.75	83.06
	iters.	14	
	probe	8E-4	7E-9
	ortho.	3E-15	
$L = 12,\ N_\uparrow = N_\downarrow = 6$	$t_{\text{iters.}}$	120.7	
n=10	t_{vectors}	4.74	
	t_{Lanczos}	125.46	142.07
	iters.	62	
	probe	2E-10	1E-14
	ortho.	4E-9	

Table E.3: Implicitly restarted Lanczos vs. standard ARPACK routine for multiple Hubbard systems ($U = 10$, $\epsilon = 10^{-5}$).

E.5 Deflation

This section describes an orthogonal transformation of the Lanczos factorization with the goal to split off certain eigenvalues by setting certain off-diagonals to zero [21].

E.5.1 Transformation of the tridiagonal matrix

Without loss of generality it is explained how to set the first off-diagonal element of T (tridiagonal symmetric) to zero. For that it is necessary to produce an orthogonal matrix Q with $Q\vec{e}_1 = \vec{y}$ where y is an eigenvector of T with norm $||\vec{y}|| = 1$.
Q can be expressed as $R + \vec{y}\vec{e}_1^\dagger$ with R upper triangular and $R\vec{e}_1 = 0$, $R^\dagger\vec{y} = 0$.
Proof: $Q\vec{e}_1 = R\vec{e}_1 + \vec{y}\vec{e}_1^\dagger\vec{e}_1 = \vec{y}$ and $Q^\dagger\vec{y} = R^\dagger\vec{y} + \vec{e}_1\vec{y}^\dagger\vec{y} = \vec{e}_1$ because $\vec{y}^\dagger\vec{y} = \vec{e}_1^\dagger\vec{e}_1 = 1$.
Another form of Q is $Q = L + \vec{y}\vec{g}^\dagger$ with L lower triangular and $L\vec{e}_1 = 0$, $L^\dagger\vec{y} = \vec{e}_1 - \vec{g}$ and $\vec{g}^\dagger = \vec{e}_1^\dagger + \frac{1}{y_1}\vec{e}_1^\dagger R$ with $\vec{y}^\dagger = (y_1^*, y_2^*, ..)$.
Proof: $Q\vec{e}_1 = L\vec{e}_1 + \vec{y}\vec{g}^\dagger\vec{e}_1 = \vec{y}\vec{e}_1^\dagger\vec{e}_1 + \vec{y}\frac{1}{y_1}\vec{e}_1^\dagger R\vec{e}_1 = \vec{y}$ and $Q^\dagger\vec{y} = L^\dagger\vec{y} + \vec{g}\vec{y}^\dagger\vec{y} = \vec{e}_1$.
Now transform the tridiagonal matrix with Q:

$$\begin{aligned}
Q^\dagger T Q &= (L + \vec{y}\vec{g}^\dagger)^\dagger T R + Q^\dagger T \vec{y}\vec{e}_1^\dagger \\
&= L^\dagger T R + \vec{g}\vec{y}^\dagger T R + Q^\dagger \vec{y}\lambda\vec{e}_1^\dagger \quad \text{with} \quad T\vec{y} = \vec{y}\lambda \\
&= L^\dagger T R + \underbrace{\vec{g}\lambda\vec{y}^\dagger R}_{\vec{y}^\dagger R = 0} + \lambda\vec{e}_1\vec{e}_1^\dagger \\
&= L^\dagger T R + \lambda\vec{e}_1\vec{e}_1^\dagger \,.
\end{aligned}$$

$L^\dagger T R$ is upper-Hessenberg because of the triangular matrices L and R. Furthermore it is symmetric which can be seen by looking at the left hand side, i.e. $L^\dagger T R$ is tridiagonal. The first column and the first row of $L^\dagger T R$ are both zero because $R\vec{e}_1 = 0$ and $L\vec{e}_1 = 0$. So the final matrix has the form

$$\begin{pmatrix} \lambda & 0 \\ 0 & \hat{T} \end{pmatrix} .$$

E.5.2 Algorithm for creating Q

Alg. 5 constructs a matrix Q with $\vec{y}$ in its first column and an upper-triangular part R with orthogonal and normalized columns.

Algorithm 5 Construction of orthogonal deflating matrix

```
Input: $\vec{y}$ with $||\vec{y}|| = 1$ and length $j$
$\sigma_1 = y_1^2$, $\tau_1 = \sqrt{\sigma_1}$
for $i = 2, ..j$ do
    $\sigma_i = \sigma_{i-1} + y_i^2$, $\tau_i = \sqrt{\sigma_i}$
    if $\tau_{i-1} \neq 0$ then
        $Q_{1:(i-1),i} = -y_{1:(i-1)} \frac{y_i}{\tau_i \tau_{i-1}}$
        $Q_{i,i} = \frac{\tau_{i-1}}{\tau_i}$
    else
        $Q_{i-1,i} = 1$
    end if
end for
```

E.5.3 Transformation of the Lanczos factorization

The Lanczos factorization at step j looks like

$$AV = VT + \vec{r}\vec{e}_j^\dagger \ .$$

An orthogonal transformation with Q results in:

$$\begin{aligned} A(VQ) &= (VQ)Q^\dagger TQ + \vec{r}\vec{e}_j^\dagger Q \\ AV_1 &= V_1 T_1 + \vec{r}\vec{e}_j^\dagger Q \ . \end{aligned}$$

The expression $\vec{e}_j^\dagger Q$ is the last row of Q which looks like $(y_j, 0, ..\tau_{j-1})$.
This means, that when dismissing the first column of V_1 we have again a valid Lanczos factorization:

$$AV_2 = V_2 T_2 + \vec{r}_2 \vec{e}_{j-1}^\dagger \ , \tag{E.2}$$

with $(\vec{v}_1, V_2) = V_1$, $\vec{r}_2 = \vec{r}\tau_{j-1}$ and

$$Q^\dagger TQ = T_1 = \begin{pmatrix} \lambda_1 & 0 \\ 0 & T_2 \end{pmatrix} \ .$$

For the vector in the first column of V_1 holds:

$$A\vec{v}_1 = \vec{v}_1 \lambda_1 + \vec{r} y_j \qquad \text{with} \quad \vec{r}^\dagger \vec{v}_1 = 0 \ .$$

The last component of the eigenvector of T can be used as a criterion when to purge a vector. A possibility is to lock if

$$|y_j| < \epsilon_D ||T|| \ ,$$

where $\epsilon_M < \epsilon_D < 1$.

F Davidson-Liu

Another representative of an iterative eigenvalue solver is the Davidson-Liu method [24, 28, 48]. The difference to Lanczos is the more sophisticated selection of the next vector added to the subspace. This leads to a slightly more computationally demanding iteration step, but convergence can be assumed more quickly [49].

Algorithm 6 Davidson-Liu algorithm

```
Input: matrix $H$, initial subspace guess $V = (\vec{v})$
for $i = 1...n_{\max}$ do
    Calculate Rayleigh-Ritz quotient $\lambda = \frac{\vec{v}^\dagger H \vec{v}}{\vec{v}^\dagger \vec{v}}$.
    Calculate residual $\vec{r} = (H - \lambda I)\vec{v}$
    Calculate approximate correction $\vec{e} = -(D - \lambda I)^{-1}\vec{r}$ with $D = \mathrm{diag}(H)$
    Add $\vec{e}$ to subspace: $V \leftarrow (V, \vec{e})$, and orthonormalize $V$.
    Solve projected eigenproblem: $V^\dagger H V \vec{x} = \mu \vec{x}$
    Calculate new approximate ground state: $\vec{v} \leftarrow V\vec{x}$
end for
```

Alg. 6 shows a simple version of Davidson-Liu for a ground state calculation. In a given subspace, an approximation to the lowest eigenstate is calculated, then the corresponding residual is evaluated. Now, the error vector $\vec{e}$ could in principle be calculated by

$$(H - \lambda I)\vec{e} = -\vec{r}$$

Note that the correct value λ is not known. Furthermore, solving the linear system can be a large effort. Instead, $\vec{e}$ is calculated approximately using only the diagonal values of H and the Rayleigh-Ritz value. To rectify this shortcoming, the vector is not used to calculate directly a new vector $\vec{v}$, but it is added to the search space V.

F.1 Criticism of the Davidson-Liu algorithm

Consider using the matrix itself instead of the diagonal of the matrix , then the correction would give you exactly the same vector as you started from. So actually the motivation for using the diagonal is not to have an approximation of the matrix, but it can be seen as a preconditioner for the residual.

There exists a more complex extension of the Davidson method which combines the iterative scheme of the Davidson method and the Jacobi diagonalization method [50]. The algorithm is called Jacobi-Davidson [51] and includes an additional orthogonalization step to search for a solution in the subspace orthogonal to the previously found vectors.

G Lanczos parallelization using MPI

G.1 Hubbard Hamiltonian multiplication

To get an efficient method for a parallel computation of the eigenvalues of a Hubbard system a specialized scheme can be used [34].

The Hamiltonian of a Hubbard system can be written down as

$$H = H_\uparrow \otimes I + I \otimes H_\downarrow + D_P \ ,$$

where $H_\uparrow$ and $H_\downarrow$ are the kinetic parts of the two spin species, respectively. D_P is a diagonal matrix containing the on-site potential.

The multiplication with a vector can be performed by representing the vector as a matrix and using simple matrix multiplication with the individual kinetic matrices. The matrix multiplication is shown in alg. 7.

Algorithm 7 Matrix multiplication of a tensorial system

```
  Input: $H_\uparrow$,$H_\downarrow$, $P$, $V$
  $R \leftarrow P \bullet V$
  $R \leftarrow R + H_\uparrow V$
  $R \leftarrow R + V H_\downarrow^T$
  Output: $R$
```

P is the on-site potential and V is the vector to be multiplied. Both are given in matrix form of dimension according to the Hamiltonians. $\bullet$ refers to element-by-element multiplication.

G.1.1 Parallelization

This scheme is particularly suited for parallelization, because some evaluation parts and communication parts can be done at the same time.

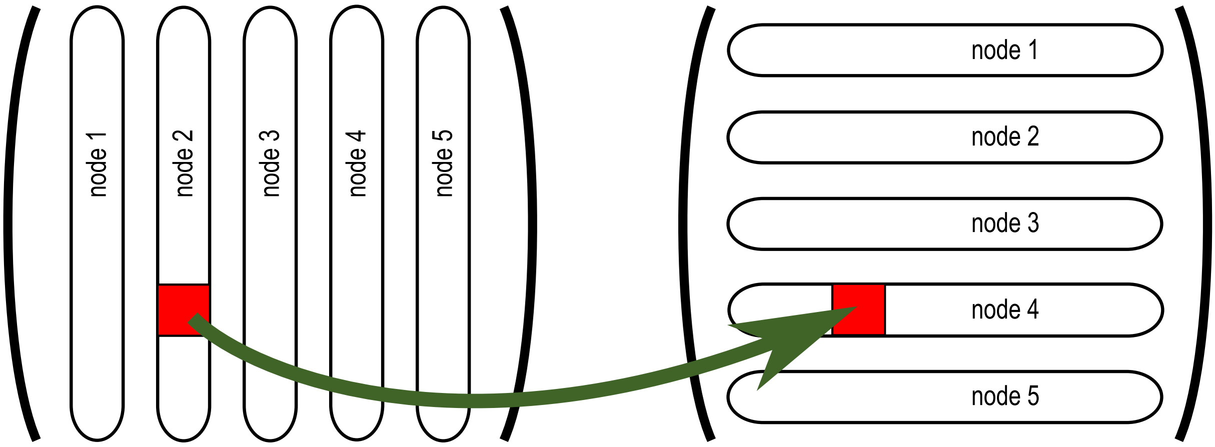

Figure G.1: Parallel data structure for Hubbard system calculations. The vectors are stored as matrices, each node holds a number of columns. When transposing, parts have to be transferred to other nodes, e. g. from node 2 to node 4.

Every node holds a number of columns of the current Lanczos vector in matrix form. Additionally each node holds the complete Hamiltonian.
Communication between nodes has to be done to transpose the columns to rows in order to perform the product with the Hamiltonian from the right. After that the result has to be retransformed to column vectors. See fig. G.1.

G.1.2 Storage format

To make the communication easy the values are stored block-wise on each node. See fig. G.2.

G.2 Numerical results

Tab. G.1 shows results from an implementation of an implicitly restarted Lanczos algorithm, parallelized with the MPI framework.
It can be seen that using the efficient communication scheme an impressive speed-up can be achieved.

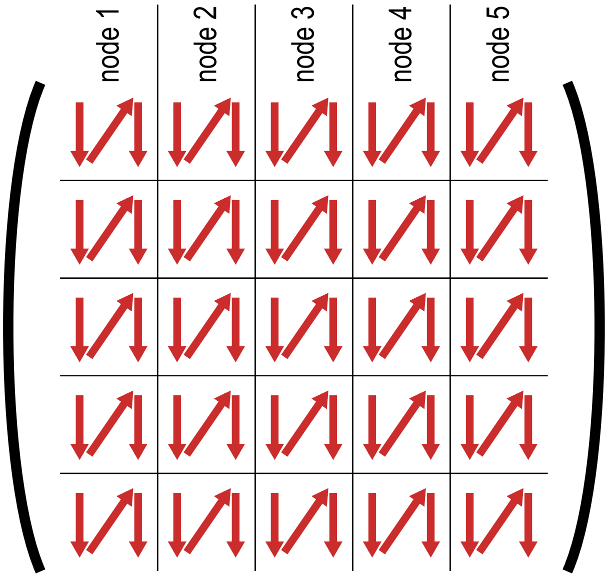

Figure G.2: Storage scheme of the Lanczos vectors. Each node holds one of the vertical structures. Inside, the values are stored block-wise again corresponding to the node structure to simplify transposing. Inside the blocks column-wise storage is used.

total cores	calc. cores	$L = 12$, $N_\uparrow = N_\downarrow = 6$			$L = 14$, $N_\uparrow = N_\downarrow = 7$		
		iters.	t_{total}	speed-up	iters.	t_{total}	speed-up
2	1	28	148.89	1.00	33	3790.83	1.00
3	2	15	41.28	1.93	36	1607.66	2.57
4	3	15	24.39	3.27	27	913.50	3.39
5	4	22	26.90	4.35	35	857.66	4.68
6	5	28	35.84	4.15	30	554.68	6.21
7	6	19	16.42	6.15	30	449.15	7.67

Table G.1: Parallel Hubbard Lanczos, implicitly restarted. The cluster ghost.tugraz.at at Graz University of Technology was used.

H Calculating dynamic properties with Lanczos

The dynamic Green's function reflects the dynamic properties of a many-body system. From it important quantities like the electronic excitation spectrum can be derived easily. This section introduces a method to calculate the Green's function by numerical means using a Lanczos factorization [28].
It is defined for fermions as follows:

$$\langle\langle \hat{O}_i, \hat{O}_j^\dagger \rangle\rangle = -i\Theta(t) \, \langle \{ \hat{O}_i(t), \hat{O}_j^\dagger \} \rangle \ ,$$

where the curly brackets denote the anti-commutator ($\{A, B\} = AB + BA$), and the operators $\hat{O}_i$ and $\hat{O}_i^\dagger$ could be e.g. the annihilation and creation operator of a particle at site i, respectively.
Introducing the Heisenberg time evolution operators $\hat{O}_i(t) = e^{i\hat{H}t}\hat{O}_i e^{-i\hat{H}t}$ and transforming the expression to Fourier space one arrives at:

$$\begin{aligned} \langle\langle \hat{O}_i, \hat{O}_j^\dagger \rangle\rangle_\omega &= -i\int_0^\infty dt e^{i\omega t} \, \langle e^{i\hat{H}t}\hat{O}_i e^{-i\hat{H}t}\hat{O}_j^\dagger + \hat{O}_j^\dagger e^{i\hat{H}t}\hat{O}_i e^{-i\hat{H}t} \rangle \\ &= -i\int_0^\infty dt e^{i\omega t} \, \langle \hat{O}_i e^{-i(\hat{H}-E_o)t}\hat{O}_j^\dagger \rangle + \langle \hat{O}_j^\dagger e^{i(\hat{H-E_0})t}\hat{O}_i \rangle \ . \end{aligned}$$

The last line was derived using: $e^{i\hat{H}t} |\phi_0\rangle = e^{iE_0t} |\phi_0\rangle$ if $H |\phi_0\rangle = E_0 |\phi_0\rangle$. Now integrate:

$$\langle\langle \hat{O}_i, \hat{O}_j^\dagger \rangle\rangle_\omega = \left\langle \hat{O}_i \frac{1}{\omega - (\hat{H} - E_0)} \hat{O}_j^\dagger \right\rangle + \left\langle \hat{O}_j^\dagger \frac{1}{\omega + (\hat{H} - E_0)} \hat{O}_i \right\rangle \ .$$

Introducing the excited, normalized vectors:

$$|\psi_i^+\rangle = \frac{1}{\left|\left| \hat{O}_i^\dagger |\phi_0\rangle \right|\right|} \hat{O}_i^\dagger |\phi_0\rangle \qquad |\psi_i^-\rangle = \frac{1}{\left|\left| \hat{O}_i |\phi_0\rangle \right|\right|} \hat{O}_i |\phi_0\rangle$$

leads to:

$$\begin{aligned}\langle\langle \hat{O}_i, \hat{O}_j^\dagger\rangle\rangle_\omega = & \left|\left|\hat{O}_i^\dagger \left|\phi_0\right\rangle\right|\right| \left|\left|\hat{O}_j^\dagger \left|\phi_0\right\rangle\right|\right| \left\langle \psi_i^+ \left| \frac{1}{\omega - (\hat{H} - E_0)} \right| \psi_j^+ \right\rangle \\ + & \left|\left|\hat{O}_i \left|\phi_0\right\rangle\right|\right| \left|\left|\hat{O}_j \left|\phi_0\right\rangle\right|\right| \left\langle \psi_j^- \left| \frac{1}{\omega + (\hat{H} - E_0)} \right| \psi_i^- \right\rangle .\end{aligned}$$

Now, one can use a Lanczos factorization of $(\hat{H} - E_0)$ with the tridiagonal matrix entries e_i and k_i to obtain the following expression:

$$Q^\dagger(\omega \pm (\hat{H} - E_0)Q = \begin{pmatrix} \omega \pm e_0 & k_0 & & & \\ k_0 & \omega \pm e_1 & k_1 & & \\ & k_1 & \omega \pm e_2 & k_2 & \\ & & k_2 & \omega \pm e_3 & \\ & & & & \ddots \end{pmatrix} .$$

To get an approximation of the quantity $\left\langle\psi\right| (\omega \pm (\hat{H} - E_0))^{-1} \left|\psi\right\rangle$ we simply need to invert the tridiagonal matrix above.

In the case of a symmetric expression, i.e. when both vectors are excitations at the same spatial site, this vector can be used as initial vector of the Krylov iteration. Then, only the upper-left matrix entry is the desired value. It can be calculated efficiently by using a recursive algorithm and determinants (see [28] for details).

For the asymmetric expressions 4 individual Lanczos runs with initial vectors in the form $\left|\Psi_i\right\rangle + \alpha \left|\Psi_j\right\rangle$ with $\alpha \in \{1, -1, i, -i\}$ have to be performed:

$$\begin{aligned}\left\langle \psi_i \left| \frac{1}{\omega - (\hat{H} - E_0)} \right| \psi_j \right\rangle = & \frac{1}{4} \left\langle \psi_i + \psi_j \left| \frac{1}{\omega - (\hat{H} - E_0)} \right| \psi_i + \psi_j \right\rangle \\ - & \frac{1}{4} \left\langle \psi_i - \psi_j \left| \frac{1}{\omega - (\hat{H} - E_0)} \right| \psi_i - \psi_j \right\rangle \\ - & \frac{i}{4} \left\langle \psi_i + i\psi_j \left| \frac{1}{\omega - (\hat{H} - E_0)} \right| \psi_i + i\psi_j \right\rangle \\ + & \frac{i}{4} \left\langle \psi_i - i\psi_j \left| \frac{1}{\omega - (\hat{H} - E_0)} \right| \psi_i - i\psi_j \right\rangle .\end{aligned}$$

I Projection-formalism

As real ab-initio Hamiltonians for strongly correlated many body systems are far beyond the capacities of nowadays computers it is necessary to reduce the degrees of freedoms. This leads to the construction of model Hamiltonians. An important example is the multi-band Hubbard model (see sec. 1.6 and app. A).

In many cases the Hubbard model has still to many degrees of freedom to be treated numerically. A further reduction to *effective* models like the tJ-model offers the possibility to perform numerical simulations and keep the key ingredients to study the quantum phenomena. Of course the results may not be quantitatively completely accurate but the qualitative physical properties can be examined.

The projection formalism [52, 53] allows to limit the degrees of freedom in an effective model of strongly correlated quantum systems. The dynamic variables which correspond to higher energy excitations are integrated out. They contribute by second order perturbation theory. In this way the configuration space of strongly-correlated systems is largely reduced which makes the calculations much less demanding.

This formalism relies on the division of the space of dynamic variables into two subsets:

$$\begin{aligned}\mathcal{P} &:= \{\left|\phi_1\right\rangle, \dots \left|\phi_{N_p}\right\rangle\} \\ \mathcal{Q} &:= \{\left|\phi_{N_p+1}\right\rangle, \dots \left|\phi_{N_p+N_q}\right\rangle\} \,,\end{aligned}$$

where $\left|\phi_i\right\rangle$ may be the basis function in occupation number space of size $N_p + N_q$. $\mathcal{Q}$ corresponds the basis states with large energy contributions, they will be projected by perturbation theory to the remaining states in $\mathcal{P}$.

Define the projector operators:

$$\begin{aligned}\hat{P} &:= \sum_{i=1}^{N_p} \left|\phi_i\right\rangle \left\langle\phi_i\right| \\ \hat{Q} &:= \sum_{i=N_p+1}^{N_p+N_q} \left|\phi_i\right\rangle \left\langle\phi_i\right| = \mathbb{1} - \hat{P} \,,\end{aligned}$$

which project a given state into the subspaces $\mathcal{P}$ and $\mathcal{Q}$, respectively.
For such a situation an effective Hamiltonian can be derived [53], which can be expressed in the basis set $\mathcal{P}$ alone:

$$\hat{H}_{\text{eff.}} = \hat{P}\hat{H}\hat{P} + \hat{P}\hat{H}\hat{Q}\frac{1}{\omega - \hat{Q}\hat{H}\hat{Q}}\hat{Q}\hat{H}\hat{P} \, .$$

For the occupation number system the term $\hat{P}\hat{H}\hat{P}$ corresponds to the internal hopping and potential in subspace $\mathcal{P}$. The second term can be interpreted as a virtual process, e.g. an initial state in $\mathcal{P}$ is converted by a hopping process into a state in $\mathcal{Q}$. A weight $\frac{1}{\omega - \hat{Q}\hat{H}\hat{Q}}$ is applied and then transformed back to $\mathcal{P}$.
This formalism can be used to derive the tJ-model as an effective approximation to the full 1-banded Hubbard model [28]. Here, the weight is assumed to be $1/U$ by reasoning due to magnitude considerations.

J Evaluation of the Green's Function in the NPT Approach

The Green's function (see sec. 1.8) can be calculated by

$$G_{i,j}(\omega) = \langle\Psi_0|\, a_j \frac{1}{\omega - H} a_i^\dagger \,|\Psi_0\rangle \ , \tag{J.1}$$

where $|\Psi_0\rangle$ is the ground state. $a_i^\dagger$ and a_j are creation and annihilation operators, respectively.

J.1 Solving the linear equation

Using a solver for linear systems the expression can be evaluated without matrix inversion [28]. At first, the vector $|e_i\rangle = a_i^\dagger\,|\Psi_0\rangle$ is calculated, then the linear system

$$(\omega - H)\,|x\rangle = |e_i\rangle$$

is solved for $|x\rangle$, and $\langle e_i|x\rangle$ is calculated.
Note that the coefficients of the excited vectors $|e_i\rangle$ have to be constructed in a different basis set, which has one more particle than the original set for $|\Psi_0\rangle$.

J.2 Solving using the partitioning

When going to subsequent higher partition one has an approximation to the ground state for partitions up to n^{th} order at a particular instance. When putting an additional particle into that ground state the maximum order needed to describe it in the excited basis is $n + 1$.

In principle one should consider the rest of the partitions when solving the linear system. In this region the vector is assumed to be zero

$$|e\rangle \equiv \begin{pmatrix} \vec{e}_1 \\ \vec{e}_2 \end{pmatrix} = \begin{pmatrix} \vec{e}_1 \\ \vec{0} \end{pmatrix} .$$

In the following equation, C represents all remaining partitions which have not been considered so far

$$\begin{pmatrix} A & E \\ E^\dagger & C \end{pmatrix} \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} = \begin{pmatrix} \vec{e}_1 \\ 0 \end{pmatrix} .$$

This leads to

$$(A - EC^{-1}E^\dagger)\vec{x}_1 = \vec{e}_1 .$$

Note that only the part $\vec{x}_1$ is needed for calculation of the scalar product:

$$\langle e_i|x\rangle \equiv \vec{e_1}^\dagger x_1 .$$

J.2.1 Approximation

As an approximation to that one could use

$$A\vec{x}_1 = \vec{e}_1 .R$$

This can be justified by using only the n^{th} partition approximation of the ground state. So larger partitions don't improve the result anymore.

J.3 Solving the linear equation

J.3.1 GMRES

For efficiently solving the linear equation the Generalized Minimal Residual Method (GMRES, [54]) can be used. This method creates a Krylov subspace and searches for the minimal residual inside the space spanned by the Krylov vectors.

Let the equation to solve be

$$A\vec{x} = \vec{b} , \tag{J.2}$$

where A is a large, sparse matrix.

Solving for an approximate solution inside the subspace spanned by the columns of matrix V_k means minimizing the functional

$$J(\vec{z}) = ||\vec{b} - A(\vec{x}_0 + \vec{z})|| \ ,$$

where x_0 is the initial approximation and z is the desired correction. The search is performed in the Krylov subspace, so $\vec{z} = V_k\vec{y}$. The functional can be transformed to:

$$\begin{aligned} J(\vec{z}) &= ||\vec{r}_0 - AV_k\vec{y}|| \\ &= ||\beta\vec{v}_1 - AV_k\vec{y}|| \ . \end{aligned}$$

In the last row the fact was used, that the residual of the initial vector can be expressed in terms of the first Krylov vector, multiplied by $\beta = ||\vec{r}_0||$. Furthermore, the Krylov space relation $AV_k = V_{k+1}\bar{H}_k$ holds. $\bar{H}_k$ is defined as

$$\bar{H}_k = \begin{pmatrix} & (H_k) & \\ 0\dots & 0 & H_{k+1}(k+1,k) \end{pmatrix} \ .$$

which is rectangular and Upper-Hessenberg. Transformation gives

$$\begin{aligned} J(\vec{y}) &= ||V_{k+1}(\beta\vec{e}_1 - \bar{H}_k\vec{y})|| \\ &= ||(\beta\vec{e}_1 - \bar{H}_k\vec{y})|| \ . \end{aligned}$$

A series of Givens rotations to transform $\bar{H}_k$ to a triangular form can be used to get

$$J(\vec{y}) = ||\beta Q\vec{e}_1 - \bar{R}\vec{y}|| \ ,$$

where Q is the accumulated orthogonal rotation matrix. The functional can now easily be minimized by solving the triangular linear system.

J.3.2 TMRES

A variant of GMRES is the Transformed Minimal Residual Method (TMRES, [55]). It is used to solve linear systems which are nearly singular. The Green's function is calculated for different ω, which may be close to an eigenvalue of the Hamiltonian and therefore constitute a nearly singular problem.

The difference to GMRES is, that the Krylov space is produced by a modified matrix:

$$M = S^{-1}T \ ,$$

obtained by decomposition of the original matrix $A = S - T$. For decomposing A an iterative scheme can be used:

$$S\vec{y}_{k+1} = T\vec{x}_k + \vec{b} \; ,$$

e.g. the Gauss-Seidel scheme. Important is, that the spectral radius of M is smaller than one.

J.4 Modification of GMRES for simultaneous evaluation of multiple ω

Instead of eq. J.2 one has to solve an equation involving a value ω, which is tuned over a certain range:

$$A\vec{x} = (\omega - H)\vec{x} = \vec{b} \; .$$

The presented scheme avoids doing GMRES for different values all over again. Introduce the functional to be minimized

$$J(\vec{z}) = ||b - (\omega - H)(\vec{x}_0 + \vec{z})|| \; ,$$

by varying z.

Note that when restarting the algorithm one needs to have different vectors $\vec{x}_0$ for different ω, so $\vec{x}_0 = \vec{x}_0(\omega)$. This is why $\vec{x}_0(\omega)$ cannot be used as starting vector for the Krylov space. The functional reads

$$J(\vec{z}) = ||\vec{\xi}_0(\omega) - (\omega - H)V_k\vec{y}|| \; ,$$

with $\vec{\xi}_0(\omega) := \vec{b} - (\omega - H)\vec{x}_0$, and $\vec{z} = V_k\vec{y}$ is an approximation in the Krylov-subspace $V_k := (\vec{v}_1, \vec{v}_2, ...\vec{v}_k)$.

Using the Krylov relation $HV_k = V_{k+1}\bar{H}_k$ and also $V_k = V_{k+1}\tilde{E}$ with $\tilde{E}$ being an identity matrix without last column, one arrives at

$$J(\vec{z}) = ||\vec{\xi}_0(\omega) - V_{k+1}(\omega\tilde{E} + \bar{H}_k)\vec{y}|| \; .$$

To eliminate V_{k+1} in the above equation one has to expand the vector $\vec{\xi}_0(\omega)$ in terms of the Krylov space:

$$\vec{\xi}_0(\omega) = V_{k+1}\vec{\chi}_0(\omega) \quad \text{with} \quad (\vec{\chi}_0(\omega))_i = \vec{\xi}_0^{\dagger}(\omega)\vec{v}_i \quad i = 1...(k+1) \; .$$

Because V_{k+1} is unitary, one can write

$$J(z) = ||\vec{\chi}_0(\omega) - (\omega\tilde{E} + \bar{H}_k)\vec{y}|| \ .$$

This is a rather small equation which can be evaluated for every ω needed.
The Krylov space is generated using H.

K An Alternative, Exact Approach to NPT

This approach uses the fact, that matrix elements connecting different V regimes only appear between adjacent domains (see fig. K.1). Consider dividing the matrix into two domains, the first one corresponds to $V = 0$ (A) and the the second is all the rest (B).

$$S_\lambda x_1 := (A - E(B-\lambda)^{-1}E^\dagger)x_1 = \lambda x_1 \; . \tag{K.1}$$

In order to apply eq. K.1 the inverse of the huge matrix B is needed. But due to the mentioned fact, that the domain is only connected to the domain with $V = 1$ only a small fraction of this inverse is needed.

K.1 Building the inverse

Consider the matrix:

$$\begin{pmatrix} A & E \\ E^\dagger & B \end{pmatrix} ,$$

and try to calculate the inverse of this matrix, but only the portion corresponding to A is needed.

This can be achieved using a Schur transformation:

$$\begin{pmatrix} A & E \\ E^\dagger & B \end{pmatrix} = \begin{pmatrix} I & EB^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} S_A & 0 \\ 0 & B \end{pmatrix} \begin{pmatrix} I & 0 \\ B^{-1}E^\dagger & I \end{pmatrix} \; .$$

Build the inverse:

$$\begin{aligned} \begin{pmatrix} A & E \\ E^\dagger & B \end{pmatrix}^{-1} &= \begin{pmatrix} I & 0 \\ -B^{-1}E^\dagger & I \end{pmatrix} \begin{pmatrix} S_A^{-1} & 0 \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} I & -EB^{-1} \\ 0 & I \end{pmatrix} \\ &= \begin{pmatrix} S_A^{-1} & -S_A^{-1}EB^{-1} \\ -B^{-1}E^\dagger S_A^{-1} & C^{-1} + B^{-1}E^\dagger S_A^{-1}EB^{-1} \end{pmatrix} \end{aligned}$$

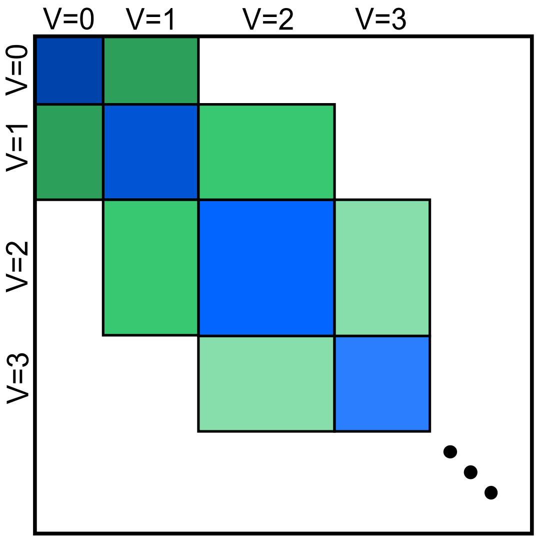

Figure K.1: Illustration of the Hamilton matrix of a V-ordered occupation number basis. The blue parts contain the interactions inside a partition and the potential on the diagonal and the green parts contain the interactions between the partitions.

We see that the needed portion of the inverse is given by the inverse of the Schur complement $S_A^{-1} = (A - BC^{-1}B^{\dagger})^{-1}$.

K.2 Iteration for the inverse

Note that the needed fraction of the large inverse of the domain $V = 1, ..V_{\text{max}}$ can be calculated iteratively. Starting from the largest V section and calculating the inverse of the Schur complement, one can go to successively smaller V.

L Singular Value Decomposition of a Kronecker product

Consider a singular value decomposition (SVD) of 2 matrices:

$$\begin{aligned} A &= U_A D_A V_A^T \\ B &= U_B D_B V_B^T \ . \end{aligned}$$

The question remains if

$$C = A \otimes B = (U_A \otimes U_B)(D_A \otimes D_B)(V_A \otimes V_B)^T = U_C D_C V_C^T$$

is a valid singular value decomposition.
To that end the prerequisites of an SVD have to be checked:

- Unitarity of $U_C = U_A \otimes U_B$:

$$\begin{aligned} U_C^T U_C &= (U_A \otimes U_B)^T (U_A \otimes U_B) \\ &= (U_A^T \otimes U_B^T)(U_A \otimes U_B) \\ \text{\tiny (mixed product property)} \quad &= (U_A^T U_A) \otimes (U_B^T U_B) \\ \text{\tiny (U_A, U_B unitary)} \quad &= \mathbb{1} \otimes \mathbb{1} \\ &= \mathbb{1} \ . \end{aligned}$$

- Diagonality of $D_C = D_A \otimes D_B$.
- Unitarity of $V_C = V_A \otimes V_B$ (proof see U_C).

M Multigrid - Complexity Considerations

M.1 Basic operations

To compare the complexity of different algorithms one can rely on evaluating the number of floating point operations (flops). Every basic operation like the addition or multiplication of 2 numbers is counted as 1 flop. Some examples for basic linear algebra operation are given in tab. M.1.

Operation	complexity (flops)
Matrix-Vector product (Matrix: $M \times N$, Vector: N)	$2MN - M$
Matrix-Vector product (Matrix: $N \times N$, Vector: N)	$2N^2 - N$
Matrix-Matrix product (Matrix: $M \times N$, Matrix: $N \times O$)	$2MNO - MO$
Matrix-Matrix product (Matrix: $N \times N$, Matrix: $N \times N$)	$2N^3 - N^2$
Vector norm (Vector: N)	$2N$
Vector normalization (Vector: N)	$3N$
Scalar product (Vectors: N)	$2N - 1$
Tensor product (Matrix: $M \times N$, Matrix: $O \times P$)	$MNOP$

Table M.1: Complexity of basic linear algebra operations (given in flops).

M.2 Gram-Schmidt orthonormalization

To orthonormalize q vectors of size N one has to do for each of them a normalization of complexity $3qN$.

For each but the first vector, the scalar product $(2N - 1)$, a projection (N), and the subtraction (N) has to be done. This gives a total of $4N - 1$ operations. The orthonormalization for $q - 1$ vectors costs $3(q - 1)N$ flops. The total number of operations for

Gram-Schmidt is given by

$$3qN + \sum_{k=1}^{q}(k-1)(4N-1) + 3(q-1)N = 2q^2N + 4qN - 3N - \frac{q^2}{2} + \frac{q}{2} \ .$$

M.3 Rayleigh-Ritz projection

Given: q vectors of size N.

- Orthonormalization: $2q^2N + 4qN - 3N - \frac{q^2}{2} + \frac{q}{2}$.
- Matrix product (U^TAU): $2N^2q + 2Nq^2 - Nq - q^2$.
- Diagonalization of subsystem: q^3.
- Calculation of Ritz vectors: $2Nq^2 - Nq$.

In total:

$$2N^2q + 6Nq^2 + 2Nq - 3N + q^3 - \frac{3}{2}q^2 + \frac{q}{2} \ .$$

M.4 Gauss-Seidel iteration

Given: matrix of size $N \times N$:
For each matrix row the following iterations have to be done:

- Multiplication: $N-1$.
- Addition: $N-1$.
- Subtraction from right side: 1.
- Division by diagonal element: 1.

Total operations per vector:

$$2N^2 \ .$$

Total operations for q vectors:

$$2N^2q \ .$$

M.5 Kaczmarz iteration

Given: matrix of size $N \times N$:
For each matrix row:

- Inner Product: $2N$.
- Norm squared: $2N$.
- Prefactor: 2.
- Correction evaluation: $2N$.

Total operations per vector:

$$6N^2 + 2N \ .$$

Total operations for q vectors:

$$6N^2 q + 2Nq \ .$$

M.6 Davidson iteration

Number of desired eigenvectors k, size of actual search space q, matrix size N:

- Rayleigh-Ritz projection: $2N^2q + 6Nq^2 + 2Nq - 3N + q^3 - \frac{3}{2}q^2 + \frac{q}{2}$.
- Calculate residuals: $2N^2k + Nk^2$.
- Calculate correction: $3Nk$.

Total operations per iteration:

$$2N^2q + 2N^2k + 6Nq^2 + Nk^2 + 2Nq + 3Nk - 3N + q^3 - \frac{3}{2}q^2 + \frac{q}{2} \ .$$

Leading two terms for n iterations:

$$2N^2qn + 2N^2q\frac{n(n+1)}{2} \ .$$

M.7 CT-Multigrid coarse cycle

Given: matrix size: N, number of vectors: q, number of primary and secondary iterations on each level: ν_1, ν_2, number of meshes: 2, coarse matrix size N_c.

- Primary GS iterations: $\nu_1 2N^2 q$.
- Transfer to coarse level: $2N_c N q - N_c q$.
- Calculation of coarse τ: $2N^2 q + 4N_c N q + 2N_c^2 q - Nq - N_c q$.
- Coarse GS iterations: $(\nu_1 + \nu_2)2N_c^2 q$.
- Transfer to fine level: $4N_c N q$.
- Secondary GS iterations: $\nu_2 2N^2 q$.

Total operations per cycle:

$$(\nu_1 + \nu_2)2N^2 q + 2N^2 q + 8N_c N q + (\nu_1 + \nu_2)2N_c^2 q + 2N_c^2 q - Nq - 2N_c q \; .$$

M.8 CT-Multigrid vs. Davidson

The leading term for the CT-Multigrid algorithm ($\nu_1 = \nu_2 = 2$) plus RR-projection on fine grid is $12N^2 q$, for Davidson: $2N^2 q + 2N^2 k$.
After n iterations the number of iterations are $12N^2 qn$ vs. $2N^2 qn + 2N^2 qn\frac{n(n+1)}{2}$.
After 19 iterations a CT-Multigrid algorithm has the same number of flops than a Davidson run with 19 iterations. Below that number, Davidson is less effort, above CT-multigrid because of linear dependence on iteration number.

Bibliography

[1] Masatoshi Imada, Atsushi Fujimori, and Yoshinori Tokura. Metal-insulator transitions. *Rev. Mod. Phys.*, 70(4):1039–1263, Oct 1998. xv

[2] Elbio Dagotto. Correlated electrons in high-temperature superconductors. *Rev. Mod. Phys.*, 66(3):763–840, Jul 1994. xv

[3] Elbio Dagotto, Takashi Hotta, and Adriana Moreo. Colossal magnetoresistant materials: the key role of phase separation. *Physics Reports*, 344(1-3):1 – 153, 2001. xv

[4] Kurt Binder. *Monte Carlo Simulation in Statistical Physics.* Springer Verlag, Berlin, Germany, 1979. xvi

[5] H. F. Trotter. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, 10(4):545–551, 1959. xvi

[6] Masuo Suzuki. Relationship between d-dimensional quantal spin systems and $(d+1)$-dimensional ising systems. *Progress of Theoretical Physics*, 56(5):1454–1469, 1976. xvi

[7] Wolfgang von der Linden. A quantum monte carlo approach to many-body physics. *Physics Reports*, 220(2-3):53 – 162, 1992. xvi

[8] John W Negele and Henri Orland. *Quantum Many-Particle Systems.* Addison-Wesley, Redwood City, USA, 1988. xvi

[9] A. W. Sandvik. A generalization of handscomb's quantum monte carlo scheme-application to the 1d hubbard model. *Journal of Physics A: Mathematical and General*, 25(13):3667, 1992. xvi

[10] H. G. Evertz. The loop algorithm. *Advances in Physics*, 52:1, 2003. xvi

[11] S. R. White and R. M. Noak. Real-space quantum renormalization-groups. *Physical Review Letters*, 68(24):3487–3490, 1992. xvi

[12] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov 1992. xvi

[13] Steven R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48(14):10345–10356, Oct 1993. xvi

[14] U. Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77(1):259–315, Apr 2005. xvi

[15] R. M. Noack and S. R. Manmana. Diagonalization- and numerical renormalization-group-based methods for interacting quantum systems. *Lectures on the Physics of Highly Correlated Electron Systems IX*, 789:93–163, 2005. 9th Training Course in the Physics of Correlated Electron Systems and High-Tc Superconductors, Salerno, Italy, 2004. xvi

[16] J. Ignacio Cirac and Frank Verstraete. Renormalization and tensor product states in spin chains and lattices. *Journal of Physics A – Mathematical and Theoretical*, 42(50):50404, 2009. xvii

[17] Peter Pippan. The density matrix renormalization group and matrix product states. Master thesis, Graz University of Technology, 2006. xvii

[18] D. Sénéchal, D. Perez, and M. Pioro-Ladrière. Spectral weight of the hubbard model through cluster perturbation theory. *Phys. Rev. Lett.*, 84(3):522–525, Jan 2000. xvii

[19] David Sénéchal, Danny Perez, and Dany Plouffe. Cluster perturbation theory for hubbard models. *Phys. Rev. B*, 66(7):075129, Aug 2002. xvii

[20] M. Potthoff. Self-energy-functional approach: Analytical results and the mott-hubbard transition. *The European Physical Journal B - Condensed Matter and Complex Systems*, 36(3):335–348, 2003. xvii

[21] Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst, editors. *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide.* Society for Industrial and Applied Mathematics, Philadelphia, USA, 2000. xvii, 125, 129, 134

[22] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, 1950. xvii

[23] Beresford N. Parlett. *The Symmetric Eigenvalue Problem.* SIAM, Philadelphia, USA, 1998. xvii, 115, 121, 122, 130

[24] Ernest R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics*, 17(1):87 – 94, 1975. xvii, 137

[25] Franz Schwabl. *Quantenmechanik für Fortgeschrittene.* Springer Verlag, Berlin, Germany, 1997. 2, 4

[26] Franz Schwabl. *Quantenmechanik.* Springer Verlag, Berlin, Germany, 1988. 4

[27] Hans-Gerd Evertz. Quanten und Felder. Lecture notes, 2004. 6

[28] Wolfgang Von der Linden. Numerical treatment of many-body problems. Lecture notes, 2001. 7, 115, 137, 143, 144, 146, 147

[29] Christian Scheiber. Variational cluster perturbation theory - applications to high temperature superconductors. Master thesis, Graz University of Technology, 2007. 7

[30] Thierry Giamarchi. *Quantum Physics in One Dimension.* Oxford University Press, Oxford, UK, 2004. 9, 10

[31] Wolfgang Nolting. *Grundkurs Theoretische Physik 7 - Viel-Teilchen-Theorie.* Springer Verlag, Berlin, Germany, 2002. 10

[32] Constantine Bekas and Yousef Saad. Computation of smallest eigenvalues using spectral schur complements. *SIAM J. Sci. Comput.*, 27(2):458–481, 2005. 13, 111

[33] Jeffrey K. Bennighof and R. B. Lehoucq. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM Journal on Scientific Computing*, 25(6):2084–2106, 2004. 13, 111

[34] Susumu Yamada, Toshiyuki Imamura, and Masahiko Machida. 16.14 TFLOPS eigenvalue solver on the Earth Simulator: Exact diagonalization for ultra largescale hamiltonian matrix. *High-Performance Computing*, 4759:402–413, 2008. 38, 139

[35] H. Voss. An arnoldi method for nonlinear eigenvalue problems. *BIT Numerical Mathematics*, 44(2):387–401, May 2004. 62

[36] Daghofer M., Oles A. M., Neuber D. R., and von der Linden W. Doping dependence of spin and orbital correlations in layered manganites. *Phys. Rev. B*, 73:104451, 2006. 74

[37] Gundolf Haase and Ulrich Langer. Multigrid - Methoden. Lecture notes, 2004. 87, 88

[38] William L. Briggs. A multigrid tutorial. Lecture slides, 2000. 87

[39] Klaus Neymeyr. Solving mesh eigenproblems with multigrid efficiency. Internet publication, 2003. 89

[40] U. Hetmaniuk. Rayleigh quotient minimization algorithm based on algebraic multigrid. *Numerical linear algebra with applications*, 14(7):563–580, 2007. 89

[41] A. Brandt, S. McCormick, and J. Ruge. Multigrid methods for differential eigenproblems. *SIAM Journal on Scientific and Statistical Computing*, 4(2):244–260, 1983. 89

[42] Sorin Costiner and Shlomo Ta'asan. Adaptive multigrid techniques for large-scale eigenvalue problems: Solutions of the Schrödinger problem in two and three dimensions. *Phys. Rev. E*, 51(4):3704–3717, Apr 1995. 91

[43] Sorin. Costiner and Shlomo. Ta'asan. *The algebraic multigrid projection for eigenvalue problems; backrotations and multigrid fixed points.* Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Langley, USA, 1994. 91

[44] Constantine Bekas and Yousef Saad. Amls, spectral schur complements and iterative computation of eigenvalues. Scientific Computing Seminar, 2005. 111

[45] J. Jaklič and P. Prelovšek. Lanczos method for the calculation of finite-temperature quantities in correlated systems. *Phys. Rev. B*, 49(7):5065–5068, Feb 1994. 118

[46] A. Honecker and S. Wessel. Magnetocaloric effect in quantum spin-s chains. *arXiv:0907.3736v1*, 2009. 125

[47] D. Calvetti, L. Reichel, and D. C. Sorensen. An implicitly restarted lanczos method for large symmetric eigenvalue problems. *ETNA*, 2:1–21, 1994. 129

[48] Andreas Stathopoulos and Charlotte F. Fischer. A Davidson program for finding a few selected extreme eigenpairs of a large, sparse, real, symmetric matrix. Internet publication, 1993. 137

[49] M. Crouzeix, B. Philippe, and M. Sadkane. The Davidson method. *SIAM J. Sci. Comput.*, 15(1):62–76, 1994. 137

[50] Gerard L. G. Sleijpen and Henk A. Van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 17(2):401–425, 1996. 138

[51] Gerard L. G. Sleijpen, Albert G. L. Booten, Diederik R. Fokkema, and Henk A. van der Vorst. Jacobi-davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT Numerical Mathematics*, 36(3):595–633, 1996. 138

[52] K. W. Becker and P. Fulde. Ground-state energy of strongly correlated electronic systems. *Zeitschrift für Physik B Condensed Matter*, 72(4):423 – 427, 1988. 145

[53] P. Fulde. *Electron Correlations in Molecules and Solids*. Springer series in solid-state sciences. Springer Verlag, Berlin, Germany, 1991. 145, 146

[54] Youcef Saad and Martin H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, July 1986. 148

[55] William W. Hager. Iterative methods for nearly singular linear systems. *SIAM Journal on Scientific Computing*, 22:747–766, 2000. 149

Acknowledgments

I wish to thank my supervisor Prof. Wolfgang von der Linden for his gentle guidance during the past three years. This work would not be possible without the freedom to explore new ideas he provided. On the other side the joint decisions for further investigations and to dismiss other paths lead to an efficient usage of the time at our disposal.

Nonetheless, I wish to thank Prof. Gundolf Haase from the University of Graz for the many meetings and his important contributions as my second supervisor. I am also thankful to Prof. Steinbach for the initiative to found the *Doktoratskolleg Numerical Simulations in Technical Sciences* which provided the necessary environment for this interdisciplinary work.

I want to thank Babette Hebenstreit for proof-reading this thesis.

I am grateful to the many members of the many-body physics group at the Institute of theoretical Physics at Graz University of Technology. Firstly, I'd like to mention my esteemed colleague and friend Peter Pippan who helped me out so many times I can't even think about. Further important members are Anna Fulterer, Michael Knap, and many more. There are also Andreas Hirczy, Ingrid Ziegler, and Winfried Kernbichler, who I thank for many entertaining conversations.

There were many more important people, which made this time of my life so enjoyable: Heinz-Georg Flesch, Armin Moser, Christian Scheiber as well as Anton Tamtögl, Andrej Golubkov, Marco Marchl, Thomas Traub, Lukas Wittwer, and all further members of the *Basisgruppe Physik*.