



UPPSALA  
UNIVERSITET

# Computation of the Susceptibility for the Anderson Impurity Model

---

Jesper Dürebrandt

**Project in Computational Science: Report**

January 2014

PROJECT REPORT



## **Abstract**

This report describes the process of making a Fortran90 implementation that computes the susceptibility in the Anderson Impurity Model (AIM). The implementation makes use of existing routines in the RSpt (Relativistic Spin Polarized test) code and is contained in a Fortran90 module that extends the RSpt code. Future plans for the module include to add routines to solve the AIM using the so-called superperturbation theory, in which the susceptibility is one of the major building blocks. The resulting implementation calculates the susceptibility for multi-orbital problems but is only checked with analytical results for the single-orbital problem. It also contains the routines for the superperturbation, but they need further work to function properly. Remaining work is to finalize the superperturbation solver and to increase performance by making use of MPI parallelization and compressed sparse array format.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Theory</b>	<b>3</b>
2.1	Anderson Impurity Model . . . . .	3
2.2	Dynamical Mean-Field Theory . . . . .	3
2.3	Exact Diagonalization . . . . .	4
2.4	Superperturbation . . . . .	5
2.5	Relativistic Spin Polarized test . . . . .	6
2.6	The Susceptibility . . . . .	7
<b>3</b>	<b>Implementation</b>	<b>8</b>
3.1	Verifying the Results . . . . .	10
<b>4</b>	<b>Results</b>	<b>10</b>
4.1	The Fortran90 Module <code>green_sp_solver</code> . . . . .	10
4.2	The Susceptibility . . . . .	10
<b>5</b>	<b>Conclusion and Discussion</b>	<b>12</b>
5.1	Improvements and future work . . . . .	13
<b>6</b>	<b>Appendix</b>	<b>14</b>
6.1	Analytical Expression for the Susceptibility $\chi$ . . . . .	14

# 1 Introduction

A great challenge in theoretical physics is to calculate the properties of materials. These properties are defined by the movement and interaction between the electrons within the material. The Hamiltonian  $H$  is an operator corresponding to the total energy of a system and it is known for this type of problem. It describes the kinetic energy, the lattice potential and the Coulomb interaction between the electrons respectively in the following three terms

$$H = \sum_i \left[ -\frac{\hbar^2 \Delta_i}{2m_e} - \sum_l \frac{e^2 Z_l}{4\pi\epsilon_0 |r_i - R_l|} \right] + \frac{1}{2} \sum_{i \neq j} \frac{e^2}{4\pi\epsilon_0 |r_i - r_j|}, \quad (1)$$

where  $r_i$  and  $R_l$  denote the position of electron  $i$  with charge  $-e$  and ion  $l$  with charge  $Z_l e$ ,  $\Delta_i$  is the Laplace operator for the kinetic energy of electrons with mass  $m_e$ ,  $\epsilon_0$  is the vacuum dielectric and  $\hbar$  is the reduced Planck constant. The many-body Schrödinger equation,  $H\Psi = E\Psi$ , describes the energy  $E$  of all the constituent states  $\Psi$  of the system. Although the expression is known, it becomes unfeasible to solve it numerically for more than  $\mathcal{O}(10)$  particles. This is due to the Coulomb interaction, the correlated movement of electron  $i$  with every other electron  $j$ , which is a problem that grows exponentially in the number of electrons. Approximations of  $H$  are needed in order to solve it, even with today's supercomputers. The Local Density Approximation (LDA) is an approximation within the Density Functional Theory (DFT) where the electrons are assumed to move independently in an averaged local density of the other electrons. This works well for describing the electrons with weak correlation, but fails to describe those with stronger correlation [1].

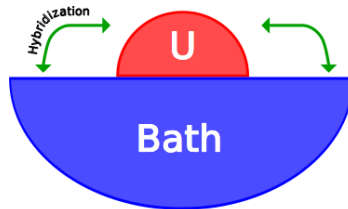
Relativistic Spin Polarized test (RSPT) is an Open Source project and a program for band structure calculations that is developed at the Division of Materials Theory. It is implemented in the programming language Fortran90. The user provides the program with an input file that describes the material to be treated and then the program computes the electronic structure by solving the many-body Schrödinger equation under a number of approximations. The interaction between itinerant electrons can be treated using DFT, meanwhile localized and strongly correlated electrons need a more rigorous treatment, and is solved explicitly using for example an Exact Diagonalization solver.

The goal of this project is to compute the susceptibility for the Anderson impurity model. The susceptibility is a measurable physical quantity that describes the response to a perturbation, from which a lot of information about a material can be derived. Unfortunately, the complexity of the susceptibility grows very fast for large problems and it is therefore computationally expensive. Using output from the Exact Diagonalization solver in the existing RSPT code an efficient implementation can be done. The project will be carried out in Fortran 90 and as an extension module to the RSPT code. Focus lies on completing a robust implementation for the single-orbital model, but in the future the result of this project is planned to be scaled up to the multi-orbital model and ultimately developed into a superperturbation solver for the RSPT code, able to solve the electronic Hamiltonian for any range of hybridization and electron-interaction strength at a reasonable computational cost.

## 2 Theory

### 2.1 Anderson Impurity Model

The Anderson impurity model (AIM) is a quantum many-body model that describes an impurity in a bath of free electrons. The model was first used in the context of materials science to investigate the behavior of impurities, such as nickel and iron, that are embedded in a metallic host [2]. The electron-electron interaction is assumed to only occur on the impurity and the impurity can hybridize with the bath, the conduction electrons of the host, by exchanging electrons. Moreover, the solution of this model is an important step in implementing so called Dynamical Mean-Field Theory (DMFT), which will be covered in more detail in the next section. Current solvers of the AIM provide controlled numerical results for spectral and thermodynamic properties, albeit being computationally expensive and non-scalable for multi-orbital models. Since many problems in materials science are described by multi-orbital models, it is important to find an efficient way of solving them. Superperturbation is a recently proposed method [3] of doing just that, which will be discussed below.

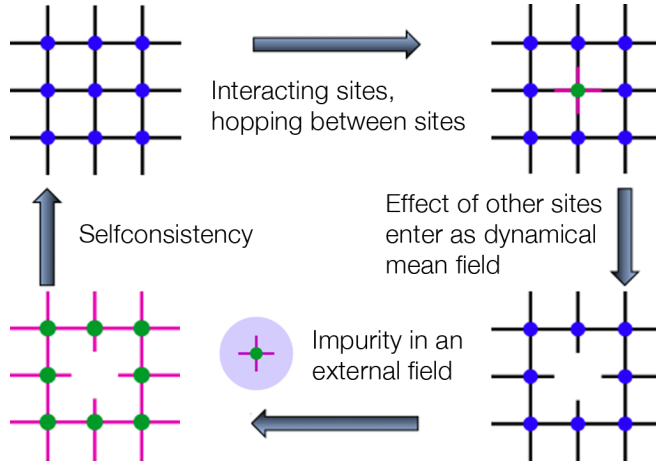


**Figure 1:** The Anderson Impurity model describes a magnetic impurity in a bath of free electrons.

### 2.2 Dynamical Mean-Field Theory

The main idea of DMFT is to represent the full lattice problem with a local impurity problem. Assuming that all the lattice sites are equivalent to each other and focusing on one site, it is clear that the rest of the lattice acts in the same manner upon every chosen site. Also, the average influence of the rest of the lattice on a local observable at a given site can be expressed by a time dependent effective field. Using this, the problem can be reduced to the solution of the AIM with a self-consistency

condition. A method to formulate this is cavity construction [2], illustrated in Figure 2. The first step is to choose a generic lattice site and then integrate out the external degrees of freedom of the rest of the lattice. In that way an effective local dynamics that includes the impact of the rest of the lattice can be defined for the chosen site. The problem is then solved self-consistently with the condition that the local impurity Green's function should coincide with the local lattice Green's function.



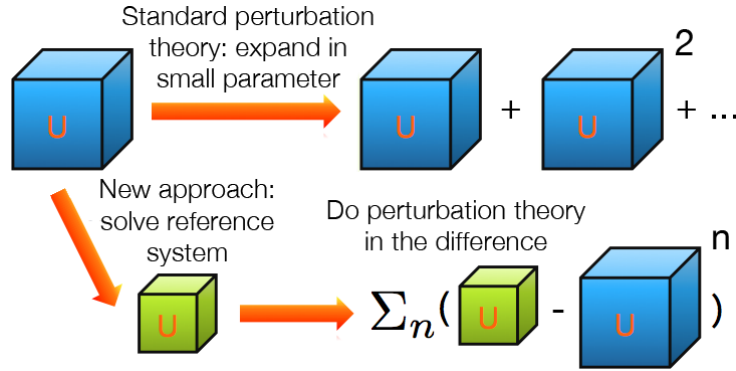
**Figure 2:** Illustration of the cavity construction in the DMFT loop.

### 2.3 Exact Diagonalization

Exact Diagonalization (ED) is a method of solving the Schrödinger equation of a quantum many body system numerically. When employing this method to the AIM, the bath is approximated by a finite number of bath sites. The Hamiltonian that describes the full system consisting of the impurity and the bath is then solved exactly and diagonalized. Since the dimension of the Hilbert space increases exponentially as  $2^{2(N_{\text{impurity}} + N_{\text{bath}})}$ , this method is limited by the available memory for a relatively small number of impurity and bath sites. For example, the Hamiltonian for a system with 1 impurity site and 6 bath sites would require 4 GB to store if it is represented with double precision complex numbers [2]. ED is a suitable technique for non-metallic materials, where the electrons are localized and their states can be well described by a discrete spectrum. Metals, on the other hand, have delocalized electrons that require a continuous spectrum in order to describe their states properly. One simply needs too many bath orbitals and ED becomes unfeasible for this type of materials.

## 2.4 Superperturbation

Conventional perturbation theory finds an approximate solution to a problem that cannot be solved exactly by perturbing the exact solution of a similar problem. This is done by expanding the system in a small parameter that specifies the difference from the exactly solvable problem. Superperturbation [3], on the other hand, starts from a reference problem that is constructed with as much of the essential physics as possible from the original problem. By transforming into a new set of fermionic variables, a perturbation theory can be formulated in the difference between both systems. This is illustrated in Figure 3. In this case, the reference problem is the AIM with a bath that is approximated with a small amount of bath orbitals. The reference problem is then solved exactly using ED. In fact, the use of ED is required here in order to obtain exact correlation functions, such as the susceptibility, that are needed for the superperturbation.



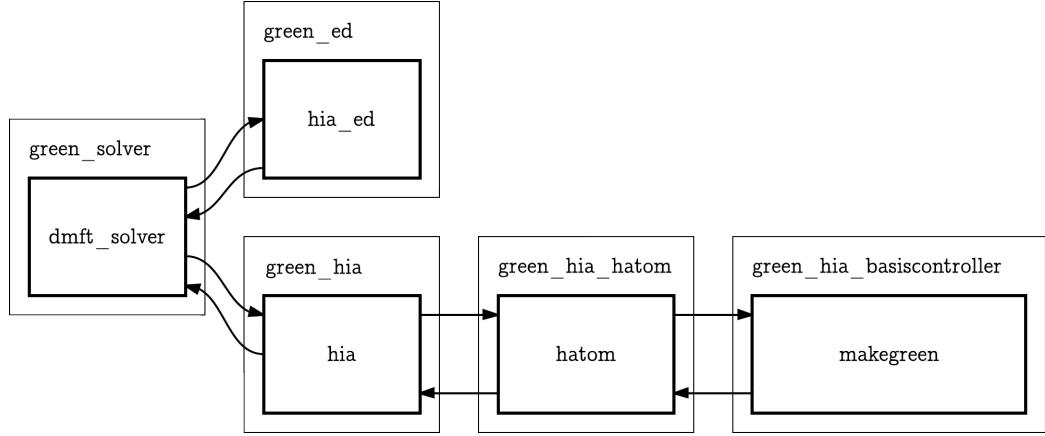
**Figure 3:** The general concept of superperturbation theory.

Applying superperturbation to the solution from ED enables us to properly describe the properties of insulators as well as metals, although the starting point is a relatively small reference system.



## 2.5 Relativistic Spin Polarized test

Relativistic Spin Polarized test (RSPt) is able to solve the many-body Schrödinger equation for almost any crystal material using DMFT with a range of solvers that can be selected by the user [4]. It is implemented in the programming language Fortran90, which is suitable for numerical computations. Most parts of the code are parallelized using MPI. The DMFT loop with the ED solver utilizes the modules and routines illustrated in Figure 4.



**Figure 4:** Illustration of the call tree in the RSPt code that comprises the DMFT loop. Modules are the outer boxes and routines are the inner boxes.

`green_solver` is the module that contains the routine `dmft_solver` that handles the DMFT loop. `dmft_solver` calls the routine `hia_ed` in the module `green_ed`, where the full Hamiltonian describing the impurity and the bath is solved exactly. When the solved Hamiltonian is returned from `green_ed`, the routine `hia` in the module `green_hia` administrates the computation of necessary functions for the DMFT loop using the solved Hamiltonian. Importantly, the routine `makegreen` in the module `green_hia_basiscontroller` computes the Green's function of the impurity, after being called through the routine `hatom` in the module `green_hia_hatom`. This is then iterated until the self-consistency condition is fulfilled, i.e. when the Green's function of the impurity is found and coincides with the local lattice Green's function.

The Green's function of the impurity is defined in the space that is spanned by the Matsubara frequencies as

$$\begin{aligned} g_{\alpha\beta}(i\omega) &= - \int_0^{\frac{1}{T}} e^{i\omega\tau} \langle c_\alpha(\tau) c_\beta^\dagger(0) \rangle \\ &= \sum_{mn} \frac{e^{-\frac{1}{T}E_m} + e^{-\frac{1}{T}E_n}}{E_m - E_n + i\omega} \langle m|c_\alpha|n \rangle \langle n|c_\beta^\dagger|m \rangle, \end{aligned}$$

where indices  $\alpha, \beta$  are orbitals,  $\tau_i$  are time variables,  $\omega_i$  are Matsubara frequencies given by  $\omega = \frac{\pi}{\beta}(2m+1)$ ,  $m = 0, 1, 2, \dots$ ,  $\frac{1}{T}$  is the inverse temperature and  $c_\alpha^\dagger$  ( $c_\alpha$ ) is an operator that creates (annihilates) an electron in orbital  $\alpha$ .  $E_i$  denotes the eigenvalue of the eigenstate  $|i\rangle$ . When the non-interacting Green's function of the impurity has been computed, all the non-zero elements on the form  $\langle i|c^\dagger|j \rangle$  and the eigenvalues of the states  $|i\rangle$  and  $|j\rangle$  are sorted and written to file. Since the Hermitian conjugate of  $\langle i|c^\dagger|j \rangle$  is equal to  $\langle j|c|i \rangle$ , it is redundant to also store the  $\langle i|c|j \rangle$  elements.

## 2.6 The Susceptibility

The susceptibility is a measurable physical quantity that describes the response of a material to a perturbation, for example a magnetic field. It is defined in the Matsubara space as

$$\begin{aligned} \chi_{\alpha\beta\gamma\delta}(\omega_1, \omega_2, \omega_3, \omega_4) &= T^2 \int_0^{\frac{1}{T}} d\tau_1 \int_0^{\frac{1}{T}} d\tau_2 \int_0^{\frac{1}{T}} d\tau_3 e^{i(\omega_1\tau_1 + \omega_2\tau_2 + \omega_3\tau_3 + \omega_4\tau_4)} \\ &\quad \times \langle T_\tau c_\alpha(\tau_1) c_\beta^\dagger(\tau_2) c_\gamma(\tau_3) c_\delta^\dagger(\tau_4) \rangle, \end{aligned} \quad (2)$$

where the time translation invariance of the imaginary time two-particle Green's function has been used in order to eliminate the dependency on  $\tau_4$ . Introducing the sum over eigenstates, the susceptibility can be written as

$$\begin{aligned} \chi_{\alpha\beta\gamma\delta}(\omega_1, \omega_2, \omega_3, \omega_4) &= \frac{1}{Z} \sum_{ijkl} \sum_{\Pi} \phi(E_i, E_j, E_k, E_l, \omega_{\Pi_1}, \omega_{\Pi_2}, \omega_{\Pi_3}) \\ &\quad \times \text{sgn}(\Pi) \langle i|\mathcal{O}_{\Pi_1}|j \rangle \langle j|\mathcal{O}_{\Pi_2}|k \rangle \langle k|\mathcal{O}_{\Pi_3}|l \rangle \langle l|c_\delta^\dagger|i \rangle, \end{aligned} \quad (3)$$

where the first sum is over the eigenstates and the second sum is over all permutations  $\Pi$  of the indices 1,2,3. Furthermore,  $\mathcal{O}_1 = c_\alpha$ ,  $\mathcal{O}_2 = c_\beta^\dagger$ ,

$\mathcal{O}_3 = c_\gamma$  and  $\Pi_i$  denotes the permutation of the  $i^{\text{th}}$  index. The sign of the permutation,  $\text{sgn}(\Pi)$ , is given by  $(-1)^{\text{No. of permutations}}$ . The susceptibility is expected to decrease by  $\frac{1}{\omega}$ .

The function  $\phi$  is given by the integral

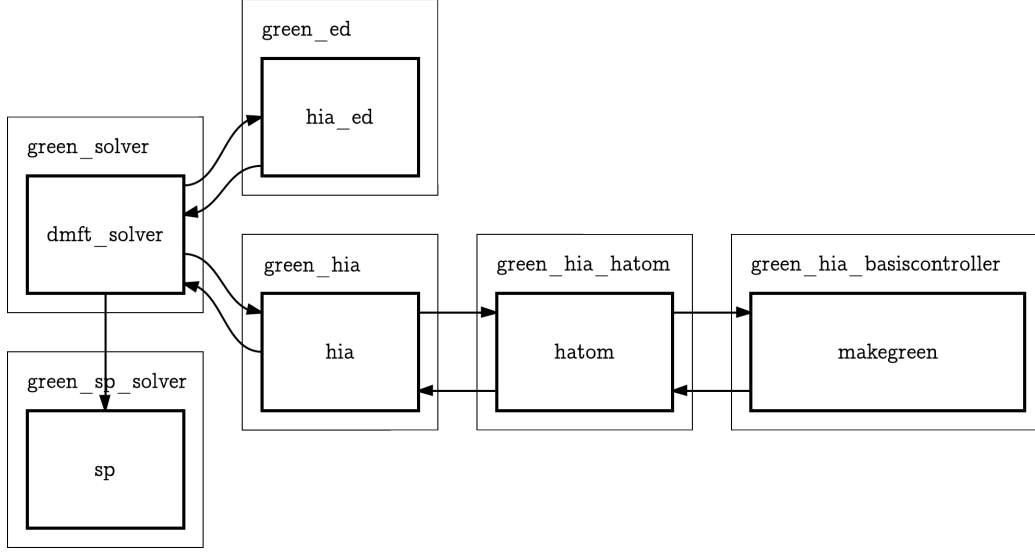
$$\phi(E_i, E_j, E_k, E_l, \omega_1, \omega_2, \omega_3) = \int_0^{\frac{1}{T}} d\tau_1 \int_0^{\frac{1}{T}} d\tau_2 \int_0^{\frac{1}{T}} d\tau_3 e^{-\frac{1}{T}E_i + (E_i - E_j)\tau_1} \times e^{(E_j - E_k)\tau_2 + (E_k - E_l)\tau_3 + i(\omega_1\tau_1 + \omega_2\tau_2 + \omega_3\tau_3)}, \quad (4)$$

which is evaluated as

$$\begin{aligned} \phi(E_i, E_j, E_k, E_l, \omega_1, \omega_2, \omega_3) = & \frac{1}{i\omega_3 + E_k - E_l} \left[ \frac{1 - \delta_{\omega_2, -\omega_3} \delta_{E_j, E_l}}{i(\omega_2 + \omega_3) + E_j - E_l} \right. \\ & \times \left( \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_j}}{i\omega_1 + E_i - E_j} - \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_l}}{i(\omega_1 + \omega_2 + \omega_3) + E_i - E_l} \right) \\ & - \frac{1}{i\omega_2 + E_j - E_k} \left( \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_j}}{i\omega_1 + E_i - E_j} + (1 - \delta_{\omega_1, -\omega_2} \delta_{E_i, E_k}) \right. \\ & \times \left. \frac{e^{-\frac{1}{T}E_k} - e^{-\frac{1}{T}E_i}}{i(\omega_1 + \omega_2) + E_i - E_k} + \frac{1}{T} \delta_{\omega_1, -\omega_2} \delta_{E_i, E_l} e^{-\frac{1}{T}E_i} \right) \\ & \left. + \delta_{\omega_2, -\omega_3} \delta_{E_j, E_l} \left( \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_j}}{(i\omega_1 + E_i - E_j)^2} - \frac{e^{-\frac{1}{T}E_j}}{i\omega_1 + E_i - E_j} \right) \right]. \quad (5) \end{aligned}$$

### 3 Implementation

This section describes the details of the implementation. In order to make the most use of the existing routines in the RSPt code, the module that computes the susceptibility (`green_sp_solver`, where *sp* stands for superperturbation) is fitted into the code as described in Figure 5.



**Figure 5:** Illustration of how the `green_sp_solver` module is fitted into the existing RSPt code. Modules are the outer boxes and routines are the inner boxes.

In order to integrate the new module, minimal modifications of the existing code are necessary in this way. To execute the module, it is selected in the same manner as the other solvers in the input file of the RSPt code. The result is printed to the standard output of the program. Since the new code is put in the DMFT part and changes its data, the self-consistency iterations will make no sense for the time being. However, if the `green_sp_solver` module is written into a functioning superperturbation solver, it will return a result that satisfies the self-consistency condition and will exit the iterative loop.

The susceptibility depends on four orbital indices and three frequency indices. Therefore, its storage and computation requires a lot of memory and scale badly for larger problems. Due to the nature of the creation and annihilation operators the array is sparse. A lot of computational time can thus be saved by looping only over non-zero elements rather than all of the elements. Since the factor  $\langle i | \mathcal{O}_{\Pi_1} | j \rangle \langle j | \mathcal{O}_{\Pi_2} | k \rangle \langle k | \mathcal{O}_{\Pi_3} | l \rangle \langle l | c_{\delta}^{\dagger} | i \rangle$  in equation (3) is independent of the frequencies, one can first loop over the sorted array of nonzero elements  $\langle i | c^{\dagger} | j \rangle$  that is written to file by the existing ED routines. If the product of the four is non-zero, then the factor  $\phi$  is calculated.

### 3.1 Verifying the Results

The computed susceptibility may be verified by implementing an analytical expression for the susceptibility of a single-orbital model in MATLAB and then compare the results numerically. This especially confirms that the data written to file by the existing code is treated correctly and also that the permutations with their respective sign are correct.

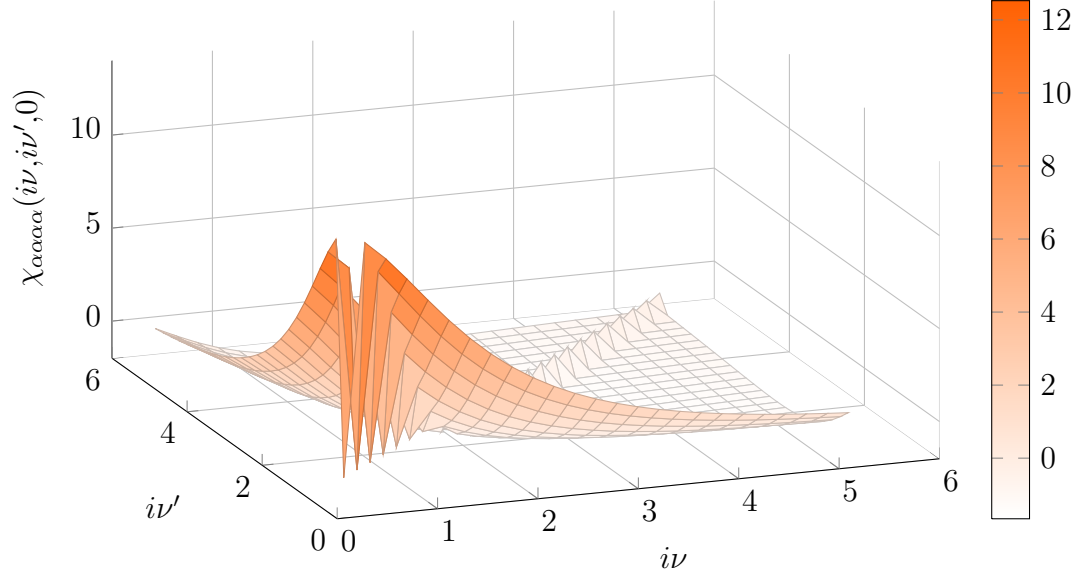
## 4 Results

### 4.1 The Fortran90 Module `green_sp_solver`

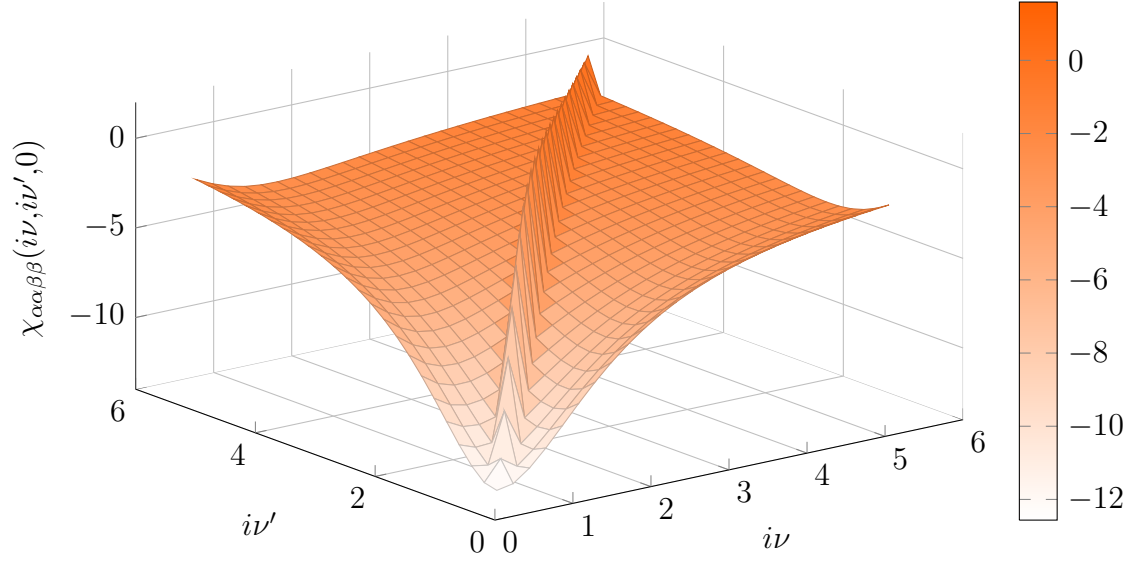
The result is a Fortran90 module that extends the existing RSPt code and computes the susceptibility for the single-orbital model. The module is executed by selecting the superperturbation solver in the input file of the RSPt code. In order to be integrated and run properly, it only requires small modifications of the related existing code. In addition to the routine that computes the susceptibility, the module contains routines that make out a good foundation for building it into a superperturbation solver.

### 4.2 The Susceptibility

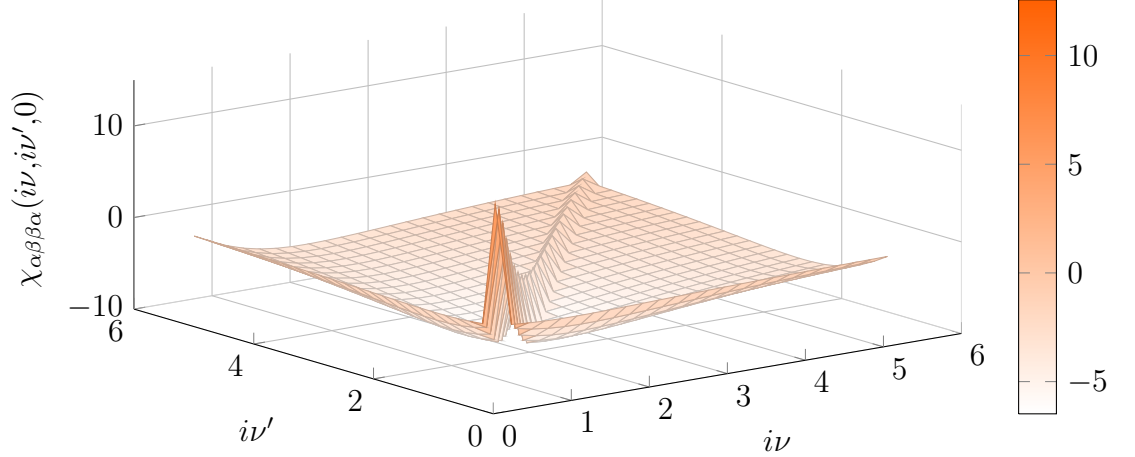
The susceptibility is verified by comparing the results with a MATLAB implementation of an analytical expression for the single-orbital model, where the permutations of the creation and annihilation operators with their respective sign are written explicitly. This confirms that the data from the existing code is treated correctly and that the structure of the computed susceptibility is correct. The analytical expression was derived from equation (3) and is found in the appendix. As shown in Figures 6-8, the computed susceptibility decreases by  $\frac{1}{\omega}$  for  $\nu \neq \nu'$  as expected. Since the computed susceptibility depends on four orbital and three frequency variables, it has been visualized in three dimensions by plotting the non-zero elements determined by the orbitals versus two frequencies and keeping the third frequency fixed at 0. Each orbital either contains an electron with spin up ( $\uparrow$ ) or spin down ( $\downarrow$ ).



**Figure 6:** The susceptibility  $\chi_{\uparrow\uparrow\uparrow\uparrow}(i\nu, i\nu', 0) = \chi_{\downarrow\downarrow\downarrow\downarrow}(i\nu, i\nu', 0)$ , keeping  $\omega$  fix.



**Figure 7:** The susceptibility  $\chi_{\uparrow\uparrow\downarrow\downarrow}(i\nu, i\nu', 0) = \chi_{\downarrow\downarrow\uparrow\uparrow}(i\nu, i\nu', 0)$ , keeping  $\omega$  fix.



**Figure 8:** The susceptibility  $\chi_{\uparrow\downarrow\uparrow}(i\nu, i\nu', 0) = \chi_{\downarrow\uparrow\downarrow}(i\nu, i\nu', 0)$ , keeping  $\omega$  fix.

## 5 Conclusion and Discussion

By employing suitable approximations of the physics by different methods, and combinations thereof, one can drastically reduce the complexity of the problem and still produce a result that correctly portrays the nature of the original problem. The combination of LDA, ED and superperturbation is a good example of that. LDA captures the essential physics of the electrons with a low level of interaction. ED contributes with the essential physics of the interacting electrons by including a finite amount of bath orbitals. The solution from ED adequately describes the properties of non-metallic materials, which have more localized electrons that are prone to stay put rather than jumping to another atom. The density of states of such materials is characterized by having distinctive peaks, meaning that the states that the electrons can occupy does not overlap and can be approximated by a discrete set of states. Metals, however, have a density of states with less distinctive peaks in general and cannot be approximated by a discrete set of states. This is because the states that the electrons can occupy are overlapping. By applying superperturbation to the solution of ED, one perturbs the solution so that it properly describes the sought system. This means that it opens way for studying materials of all kinds, with weakly or strongly correlated electrons. But this does not come for free; superperturbation is computationally expensive and needs to be implemented efficiently. The susceptibility is one of the parts that requires the most time and memory.

## 5.1 Improvements and future work

The current implementation that computes the susceptibility is serial, meaning that it would only utilize one processor when executed on a parallel machine with multiple cores. It would be highly beneficial to parallelize the code, since many parts of the code can be run independently and also that most of the existing RSpt code is parallelized using MPI. The susceptibility is also stored as a dense array, even though it is sparse. Other sparse arrays in the RSpt code are represented in compressed sparse format, meaning that only the non-zero elements are stored and all the zeros are discarded. There are efficient routines that perform operations on arrays in compressed sparse format, meaning that the computational time of elements that are zero is saved. By storing the susceptibility in the same manner, one would save a lot of memory and computational time.

Future work is to verify the susceptibility for multi-orbital problems and to finalize the superperturbation solver. Finalizing the superperturbation solver is probably the most convenient way to verify that the susceptibility is correct, since the output of the solver are entities that have a lower dimensionality than the susceptibility itself and are easier to relate to other results.

## References

- [1] K. Held (2007) *Electronic structure calculations using dynamical mean field theory*, Advances in Physics, 56:6, 829-926
- [2] H. Hafermann (2010), *Numerical Approaches to Spatial Correlations in Strongly Interacting Fermion Systems*, Institute for Theoretical Physics, University of Hamburg.
- [3] H. Hafermann, C. Jung, S. Brener and A. I. Lichtenstein (2008), *Superperturbation solver for quantum impurity models*, Institute for Theoretical Physics, University of Hamburg.
- [4] J. Wills, P. Andersson, T. Björkman, I. Di Marco, O. Grånäs and P. Thunström (2008), *RSpt Manual v1.0*, Division of Materials Theory, Uppsala University.



## 6 Appendix

### 6.1 Analytical Expression for the Susceptibility $\chi$

Analytical expression for the susceptibility  $\chi(\alpha, \beta, \gamma, \delta, \omega_1, \omega_2, \omega_3)$  for 1 impurity site and 0 bath sites, where  $\alpha, \beta, \gamma, \delta$  are orbitals and  $\omega_1, \omega_2, \omega_3$  are Matsubara frequencies.

$$\begin{aligned}\chi(\uparrow, \uparrow, \uparrow, \uparrow, \omega_1, \omega_2, \omega_3) &= \chi(\downarrow, \downarrow, \downarrow, \downarrow, \omega_1, \omega_2, \omega_3) = \\ &+ \phi(E_0, E_\uparrow, E_0, E_\uparrow, \omega_1, \omega_2, \omega_3) \times \langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle\langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle \\ &+ \phi(E_\downarrow, E_{\uparrow\downarrow}, E_\downarrow, E_{\uparrow\downarrow}, \omega_1, \omega_2, \omega_3) \times \langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle\langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle \\ &- \phi(E_0, E_\uparrow, E_0, E_\uparrow, \omega_3, \omega_2, \omega_1) \times \langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle\langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle \\ &- \phi(E_\downarrow, E_{\uparrow\downarrow}, E_\downarrow, E_{\uparrow\downarrow}, \omega_3, \omega_2, \omega_1) \times \langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle\langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle\end{aligned}$$

$$\begin{aligned}\chi(\uparrow, \uparrow, \downarrow, \downarrow, \omega_1, \omega_2, \omega_3) &= \chi(\downarrow, \downarrow, \uparrow, \uparrow, \omega_1, \omega_2, \omega_3) = \\ &+ \phi(E_0, E_\uparrow, E_0, E_\downarrow, \omega_1, \omega_2, \omega_3) \times \langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle\langle 0|c_\downarrow|\downarrow\rangle\langle\downarrow|c_\downarrow^\dagger|0\rangle \\ &- \phi(E_0, E_\uparrow, E_{\uparrow\downarrow}, E_\downarrow, \omega_1, \omega_3, \omega_2) \times \langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\downarrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle\langle\downarrow|c_\downarrow^\dagger|0\rangle \\ &+ \phi(E_\uparrow, E_0, E_\downarrow, E_{\uparrow\downarrow}, \omega_2, \omega_3, \omega_1) \times \langle\uparrow|c_\uparrow^\dagger|0\rangle\langle 0|c_\downarrow|\downarrow\rangle\langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\downarrow^\dagger|\uparrow\rangle \\ &- \phi(E_\uparrow, E_0, E_\uparrow, E_{\uparrow\downarrow}, \omega_2, \omega_1, \omega_3) \times \langle\uparrow|c_\uparrow^\dagger|0\rangle\langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\downarrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\downarrow^\dagger|\uparrow\rangle \\ &+ \phi(E_0, E_\downarrow, E_{\uparrow\downarrow}, E_\downarrow, \omega_3, \omega_1, \omega_2) \times \langle 0|c_\downarrow|\downarrow\rangle\langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle\langle\downarrow|c_\downarrow^\dagger|0\rangle \\ &- \phi(E_\uparrow, E_{\uparrow\downarrow}, E_\downarrow, E_{\uparrow\downarrow}, \omega_3, \omega_2, \omega_1) \times \langle\uparrow|c_\downarrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle\langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\downarrow^\dagger|\uparrow\rangle\end{aligned}$$

$$\begin{aligned}\chi(\uparrow, \downarrow, \downarrow, \uparrow, \omega_1, \omega_2, \omega_3) &= \chi(\downarrow, \uparrow, \uparrow, \downarrow, \omega_1, \omega_2, \omega_3) = \\ &+ \phi(E_\downarrow, E_{\uparrow\downarrow}, E_\uparrow, E_{\uparrow\downarrow}, \omega_1, \omega_2, \omega_3) \times \langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\downarrow^\dagger|\uparrow\rangle\langle\uparrow|c_\downarrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle \\ &- \phi(E_0, E_\uparrow, E_{\uparrow\downarrow}, E_\uparrow, \omega_1, \omega_3, \omega_2) \times \langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\downarrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\downarrow^\dagger|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle \\ &+ \phi(E_\downarrow, E_0, E_\downarrow, E_{\uparrow\downarrow}, \omega_2, \omega_3, \omega_1) \times \langle\downarrow|c_\downarrow^\dagger|0\rangle\langle 0|c_\downarrow|\downarrow\rangle\langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle \\ &- \phi(E_\downarrow, E_0, E_\uparrow, E_{\uparrow\downarrow}, \omega_2, \omega_1, \omega_3) \times \langle\downarrow|c_\downarrow^\dagger|0\rangle\langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\downarrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\uparrow^\dagger|\downarrow\rangle \\ &+ \phi(E_0, E_\downarrow, E_{\uparrow\downarrow}, E_\uparrow, \omega_3, \omega_1, \omega_2) \times \langle 0|c_\downarrow|\downarrow\rangle\langle\downarrow|c_\uparrow|\uparrow\downarrow\rangle\langle\uparrow\downarrow|c_\downarrow^\dagger|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle \\ &- \phi(E_0, E_\downarrow, E_0, E_\uparrow, \omega_3, \omega_2, \omega_1) \times \langle 0|c_\downarrow|\downarrow\rangle\langle\downarrow|c_\downarrow^\dagger|0\rangle\langle 0|c_\uparrow|\uparrow\rangle\langle\uparrow|c_\uparrow^\dagger|0\rangle\end{aligned}$$

where  $\phi$  is given by

$$\begin{aligned}
\phi(E_i, E_j, E_k, E_l, \omega_1, \omega_2, \omega_3) = & \frac{1}{i\omega_3 + E_k - E_l} \left[ \frac{1 - \delta_{\omega_2, -\omega_3} \delta_{E_j, E_l}}{i(\omega_2 + \omega_3) + E_j - E_l} \right. \\
& \times \left( \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_j}}{i\omega_1 + E_i - E_j} - \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_l}}{i(\omega_1 + \omega_2 + \omega_3) + E_i - E_l} \right) \\
& - \frac{1}{i\omega_2 + E_j - E_k} \left( \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_j}}{i\omega_1 + E_i - E_j} + (1 - \delta_{\omega_1, -\omega_2} \delta_{E_i, E_k}) \right. \\
& \times \frac{e^{-\frac{1}{T}E_k} - e^{-\frac{1}{T}E_i}}{i(\omega_1 + \omega_2) + E_i - E_k} + \frac{1}{T} \delta_{\omega_1, -\omega_2} \delta_{E_i, E_l} e^{-\frac{1}{T}E_i} \Bigg) \\
& \left. + \delta_{\omega_2, -\omega_3} \delta_{E_j, E_l} \left( \frac{e^{-\frac{1}{T}E_i} + e^{-\frac{1}{T}E_j}}{(i\omega_1 + E_i - E_j)^2} - \frac{e^{-\frac{1}{T}E_j}}{i\omega_1 + E_i - E_j} \right) \right].
\end{aligned}$$

The remaining elements are equal to zero due to the nature of the creation and annihilation operators.