CTQMC user guide/design document

Patrick Sémon

February 4, 2019

1 Goal

The GW+DMFT module of COMSCOPE necessitates the solution of an impurity model. Continuous-time quantum Monte Carlo (CTQMC) is an algorithm which allows to obtain exact solutions (in a statistical sense) of impurity models, and for impurity models as encountered in real material simulations, hybridization-expansion CTQMC is best suited.

2 OVERVIEW

Since the last update half a year ago, the user guide/design document has been improved and contains more explanations, expecially when it comes to topics that are hard to find in the literature (or not available, to my knowledge, as the equation of motion for orbitally resolved susceptiblilties, see below). The whole code has been revised and modularity is improved (which is time consuming but worth it), and the following as been added to the solver:

- Bulla's trick for measuring the self-energy.
- Support for complex hybridisation functions (beta version).
- Orbitally resolved susceptibilities, calculated using both straightforward estimators and improved estimators (based on the equation of motion).
- Moments and tail of susceptibilities.
- Better estimator for the impurity reduced density matrix, from which all static observables are calculated.
- More options for predefined quantities which are common in real materials simulations.
- Legendre polynomials for measuring the Green function.
- Probabilities of the impurity states, both for occupation number states and eigenstates of the impurity.
- Endian safeness (for whatever kind of endian you can imagine).

An input file for the previous version is not fully compatible with the new version. Parameters that have been renamed are `green cutoff` → `green matsubara cutoff` and `shell` → `orbitals`. Also, `quantum numbers` is listed as field in the input file and not as part of `hloc` anymore, and quantum number susceptibilities are not measured by default. This list of changes is not exhaustive, please see Sec. 3.1 and 4.1. The output file of the post-processing has been renamed, `observables.json` → `params.obs.json`.

2.1 Impurity Model

An impurity model consists of a small interacting system, the impurity, immersed in a bath of non-interacting fermions. The action reads

$$S = -\sum_{ij} \iint_0^\beta c_i^\dagger(\tau)\mathcal{G}_{0,ij}^{-1}(\tau-\tau')c_i(\tau')d\tau d\tau' + \frac{1}{2}\sum_{ijkl} \iint_0^\beta c_i^\dagger(\tau^+)c_j^\dagger(\tau'^+)\mathcal{U}_{ijkl}(\tau-\tau')\mathcal{C}(\tau')c_l(\tau)d\tau d\tau', \quad (1)$$

where $c_i^\dagger$ creates a particle in the (generalized) orbital i, β is the inverse temperature, $\mathcal{G}_{0,ij}$ is the Weiss field and $\mathcal{U}_{ijkl}$ is the frequency dependent interaction on the impurity.
It is useful to distinguish between the instantaneous and the purely retarded parts of the action. To this end, the Weiss field and the frequency dependent interaction are split as

$$\mathcal{G}_{0,ij}^{-1}(i\omega_n) = (i\omega_n + \mu)\delta_{ij} - t_{ij} - \Delta_{ij}(i\omega_n) \qquad \text{and} \qquad \mathcal{U}_{ijkl}(i\nu_n) = V_{ijkl} + U_{ijkl}(i\nu_n), \tag{2}$$

respectively. Here μ is the chemical potential, t_{ij} are impurity orbital energies, Δ_{ij} is the hybridisation function and V_{ijkl} (U_{ijkl}) is the instantaneous (purely retarded) part of the interaction. The hybridisation function and the retarded interaction satisfy $\lim_{n\to\infty}\Delta_{ij}(i\omega_n) = \lim_{n\to\infty} U_{ijkl}(i\nu_n) = 0$. Collecting all instantaneous parts yields

$$S_{\text{loc}} = \int_0^\beta \sum_i c_i^\dagger(\tau)\partial_\tau c_i(\tau) + H_{\text{loc}}[c^\dagger(\tau), c(\tau)]d\tau \tag{3}$$

where

$$H_{\text{loc}} = \sum_{ij} c_i^\dagger(t_{ij} - \mu\delta_{ij})c_j + \frac{1}{2}\sum_{ijkl} c_i^\dagger c_j^\dagger V_{ijkl}\mathcal{C}c_l. \tag{4}$$

The purely retarded one body parts coming from Δ_{ij} yield

$$S_{\text{hyb}} = \sum_{ij}\iint c_i^\dagger(\tau)\Delta_{ij}(\tau-\tau')c_j(\tau')d\tau d\tau' \tag{5}$$

and the purely retarded two body parts coming from U_{ijkl} yield

$$S_{\text{dyn}} = \frac{1}{2}\sum_{IJ}\iint Q_I^\dagger(\tau)F_{IJ}(\tau-\tau')Q_J(\tau')d\tau d\tau' \tag{6}$$

when written in a basis of particle-hole bilinears

$$Q_I^\dagger = \sum_{il} c_i^\dagger c_l \langle i,l|I\rangle. \tag{7}$$

The retarded interaction F_{IJ} in this basis satisfies

$$U_{ijkl} = \sum_{IJ}\langle i,l|I\rangle F_{IJ}\langle J|k,j\rangle. \tag{8}$$

Since F_{IJ} has no instantaneous part, the equal-time ordering of the operators in Eq. 6 does not matter.

2.2 CTQMC

Solving the impurity problem with continuous time quantum Monte Carlo (CTQMC) starts by expanding the partition function of the impurity problem

$$Z = \int \mathcal{D}[c^\dagger, c]e^{-S_{\text{loc}}-S_{\text{hyb}}-S_{\text{dyn}}} = \sum_{km}\int \mathcal{D}[c^\dagger, c]\frac{1}{k!m!}(-S_{\text{hyb}})^k(-S_{\text{dyn}})^m e^{-S_{\text{loc}}} \tag{9}$$

in the hybridisation part S_{hyb} and the retarded part S_{dyn} of the impurity model action $S = S_{\text{loc}} + S_{\text{hyb}} + S_{\text{dyn}}$. If all particle-hole bilinears contributing to the retarded interaction are of the form

$$Q_I = \sum_i q_{iI}c_i^\dagger c_i \tag{10}$$

and commute with the local Hamiltonian, the retarded part can be traced out from the expansion, see Appendix B. This yields

$$Z = \sum_{k=0}^\infty \int_0^\beta d\tau_1 \cdots \int_0^{\tau_{k-1}} d\tau_k \int_0^\beta d\tau_1' \cdots \int_0^{\tau_{k-1}'} d\tau_k' \sum_{i_1\cdots i_k}\sum_{i_1'\cdots i_k'} w(i_1\tau_1 i_1'\tau_1' \cdots i_k\tau_k i_k'\tau_k') \tag{11}$$

with weights

$$\begin{aligned} w(i_1\tau_1 i_1'\tau_1' \cdots i_k\tau_k i_k'\tau_k') = \mathrm{Tr}[e^{-\beta\tilde{H}_{\mathrm{loc}}} \prod_{r=1}^{k} c_{i_r'}(\tau_r') c_{i_r}^\dagger(\tau_r)] \times \underset{1\le n,m\le k}{\mathrm{Det}} \Delta_{i_n i_m'}(\tau_n - \tau_m') \\ \times \exp\big[\frac{1}{2} \sum_{n,m=1}^{k} \sum_{IJ} q_{i_n I} q_{i_m J} K_{IJ}(\tau_n - \tau_m) - q_{i_n I} q_{i_m' J} K_{IJ}(\tau_n - \tau_m') \\ - q_{i_n' I} q_{i_m J} K_{IJ}(\tau_n' - \tau_m) + q_{i_n' I} q_{i_m' J} K_{IJ}(\tau_n' - \tau_m')\big], \end{aligned} \tag{12}$$

where

$$\tilde{H}_{\mathrm{loc}} := H_{\mathrm{loc}} + \frac{1}{2} \sum_{IJ} Q_I F_{IJ}(i\nu_n = 0) Q_J \tag{13}$$

and

$$K_{IJ}(\tau) := -\frac{1}{2\beta} F_{IJ}(i\nu_n = 0)|\tau|(\beta - |\tau|) + \sum_{n\neq 0} \frac{F_{IJ}(i\nu_n)}{\beta (i\nu_n)^2} e^{-i\nu_n \tau}. \tag{14}$$

Here and in the following, I assume that the q_{iI} in Eq. 10 are real.

Sampling the probability distribution[1] $p(\mathcal{C}) = w(\mathcal{C})/Z$ over the space of configurations $\mathcal{C} := i_1\tau_1 i_1'\tau_1' \cdots i_k\tau_k i_k'\tau_k'$ with a Metropolis-Hasting Markov-Chain algorithm yields estimates of observables. More precisely, the expectation value of an observable O is expanded in the hybridisation S_{hyb} and the retarded interaction S_{dyn},

$$\langle O \rangle = Z^{-1} \int \mathcal{D}[c^\dagger, c] O e^{-S} = Z^{-1} \sum_{km} \int \mathcal{D}[c^\dagger, c] \frac{1}{k!m!} O (-S_{\mathrm{hyb}})^k (-S_{\mathrm{dyn}})^m e^{-S_{\mathrm{loc}}} = Z^{-1} \sum_{\mathcal{C}} w_O(\mathcal{C}), \tag{15}$$

and the retarded contributions are integrated out,[2] in the same way as for the partition function. To obtain an estimate of this observable by sampling the partition function space, the sum is rewritten as

$$\langle O \rangle = Z^{-1} \sum_{\mathcal{C}} w_O(\mathcal{C}) = Z^{-1} \sum_{\mathcal{C}} \frac{w_O(\mathcal{C})}{w(\mathcal{C})} w(\mathcal{C}) = \sum_{\mathcal{C}} \frac{w_O(\mathcal{C})}{w(\mathcal{C})} p(\mathcal{C}), \tag{16}$$

which is possible *as long as*

$$\text{supp } w_O \subset \text{supp } w. \tag{17}$$

The quantity $w_O(\mathcal{C})/w(\mathcal{C})$ is called estimator of the observable. If Eq. 17 does not hold, sampling the partition function space is not enough, and the configuration space of both the observable (supp w_O) and the partition function (supp w) have to be sampled (the latter is necessary for normalisation, c.f. worm sampling). The problem is not a division by zero in Eq. 16 when sampling an observable for which Eq. 17 does not hold, since $w(\mathcal{C}) = 0$ means that $\mathcal{C}$ is never sampled. The problem is that not all the observable space is sampled.

The present CTQMC solver *supports* complex hybridisation functions. The only *restrictions* on the impurity model are that

- the tensors t_{ij} and U_{ijkl} are real,
- and that the frequency dependent interaction is of the form

  $$\mathcal{U}_{ijkl} = U_{ijkl} + \delta_{il}\delta_{jk} F(i\omega_n). \tag{18}$$

  That is, the purely retarded part of the interaction reads

  $$S_{\mathrm{dyn}} = \frac{1}{2} \iint N(\tau) F(\tau - \tau') N(\tau') d\tau d\tau' \tag{19}$$

  where $N = \sum_i c_i^\dagger c_i$. Notice that the retarded function $F(\tau)$ is real (hermicity) and even $F(\tau) = F(-\tau)$, which means that $F(i\nu_n)$ is real and falls of as $(i\nu_n)^{-2}$.

[1] Assume $w \geq 0$ for the sake of simplicity.

[2] Be carful if an observable does not commute with a quantum number operator, this leads to additional terms when integrating out retarded contributions.

2.3 Observables

The most important observable within DMFT is the self-energy

$$\Sigma_{ij}(i\omega_n) = \mathcal{G}^{-1}_{0,ij}(i\omega_n) - G^{-1}_{ij}(i\omega_n), \tag{20}$$

where

$$G_{ij}(\tau) = -\langle c_i(\tau) c^\dagger_j \rangle \tag{21}$$

is the Green function on the impurity. Here $\langle \circ \rangle = Z^{-1} \int \mathcal{D}[c^\dagger, c] e^{-S} \circ$ with Z the partition function of the impurity model.

Note: For the Green function, an expansion as in Eq. 15 is readily obtained as a functional derivative of the partition function,

$$G_{ij}(i\omega_n) = -\frac{\partial \ln Z}{\partial \Delta_{ji}(i\omega_n)} = Z^{-1} \sum_{\mathcal{C}} -\frac{\partial w(\mathcal{C})}{\partial \Delta_{ji}(i\omega_n)}. \tag{22}$$

For this expansion, the condition Eq. 17 implies that only entries G_{ij} for which Δ_{ji} is non zero can be obtained by sampling the partition function space, and the estimator is

$$G_{ij}(i\omega_n) = Z^{-1} \sum_{\mathcal{C}} w(\mathcal{C}) \sum_{r,s=1}^{|\mathcal{C}|} -\beta^{-1} e^{i\omega_n(\tau'_s - \tau_r)} \delta_{i,i'_s} \delta_{j,i_r} (M^{-1})_{sr}, \tag{23}$$

where $M_{rs} = \Delta_{i_r i'_s}(\tau_r - \tau_s)$.

Other observables that can be obtained with the CTQMC solver are

- susceptibilities of the form

  $$\chi_I(\tau) = \langle Q_I(\tau) Q_I \rangle - \langle Q_I \rangle \langle Q_I \rangle, \tag{24}$$

  where the particle-hole bilinears Q_I in Eq. 10 commute with H_{loc}, and orbital occupation resolved susceptibilities

  $$\chi_{ij}(\tau) = \langle n_i(\tau) n_j \rangle - \langle n_i \rangle \langle n_j \rangle, \tag{25}$$

  where $n_i = c^\dagger_i c_i$.

- the impurity reduced density matrix $\hat{\rho}$ which allows calculate the expectation value of any static observable $\hat{O}$ on the impurity as

  $$\langle O \rangle = \mathrm{Tr} \hat{O} \hat{\rho}, \tag{26}$$

  for example the total spin S^2. The diagonal part of $\hat{\rho}$ gives the probabilities of the impurity eigenstates. The impurity reduced density matrix is also useful for calculating moments of correlation functions.

2.4 Flowchart

I proceed in two steps when solving the impurity problem with CTQMC as described below. For a more detailed description, see Sec. 3 and 4.

2.4.1 Monte-Carlo Simulation

Input: Mandatory are the impurity model action Eq. 1, that is

- the inverse temperature β and chemical potential μ,
- the hybridisation function $\Delta_{ij}(i\omega_n)$ in Matsubara frequencies,
- the one body tensor t_{ij} and the two body tensor V_{ijkl} defining the local Hamiltonian,

and control parameters. Optionally

- the dynamic interaction $F(i\nu_n)$ in Matsubara frequencies

and more control parameters.

Output: "Raw" measurements contained in a single file. Besides that, Markov-Chain status files and a file containing information about the local Hamiltonian H_{loc} which was generated from t_{ij} and V_{ijkl}.

2.4.2 Post-Processing

Input: Mandatory are all input files of the Monte-Carlo simulation above, and the output files of the Monte-Carlo simulation containing the raw measurements and information about H_{loc}. Optionally

- the quantum numbers q_{iI} of particle-hole bilinears in Eq. 10, used for example for defining susceptibilities,
- static observables on the impurity, for example the total spin S^2,

and control parameters.

Output: "Processed" measurements, that is, expectation values of observables discussed in Sec. 2.3 and more.

3 Monte-Carlo Simulation

The input and output files have

- JSON (JavaScript Object Notation) format, see `https://www.json.org/` for a definition of this format. In short, this format defines 6 value types and two data structures. The data structures are objects, that is a collection of key and value pairs, and arrays of values. The 6 value types are the null type, the boolean, numbers, strings, as well as the data structures themselves. An array of the null type, the boolean true, three numbers and a string is

```
[ null, true, 42, 3.14, 1.2e3, "himmelhergottsakramentzifixhalleluja..." ]
```

and

```
{
    "candidates" : [1, 3],
    "passed" : false
}
```

is an object with two keys and values, an array and the boolean false. The arrays and objects can be nested at wish. The order of the key value pairs in an object does not matter. Notice however the missing semicolon after the last pair in an object. Numbers can be entered in any of the common formats such as `42`, `3.14` or `1.2e3`.

- all energies given in the same unit (which is arbitrary since the action is dimensionless, that is, the energy unit cancels in the definition of the action).
- orbitals enumerated as $i = 1, 2, \ldots, N_O$, where N_O is the number of orbitals.
- matrices written as arrays of arrays. The inner arrays are the rows of the matrix.

3.1 INPUT

The main input file for the CTQMC solver (called `params.json` in the following), consists of a JSON object. I start here with a minimalistic example, see below.

```
{
    "hloc" : {
        "one body" : [
            [0.02, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, -0.55, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, -0.6, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0.02, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, -0.55, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

```
            [0, 0, 0, 0, 0, 0, 0, 0, 0, -0.6]],

        "two body" : [ 0, 0, 0, 0, ... ]
    },

    "hybridisation" : {
        "matrix" : [
            [ "xy", "", "", "", "", "", "", "", "", ""],
            [ "", "yz", "", "", "", "", "", "", "", ""],
            [ "", "", "3z2r2", "", "", "", "", "", "", ""],
            [ "", "", "", "zx", "", "", "", "", "", ""],
            [ "", "", "", "", "x2y2", "", "", "", "", ""],
            [ "", "", "", "", "", "xy", "", "", "", ""],
            [ "", "", "", "", "", "", "yz", "", "", ""],
            [ "", "", "", "", "", "", "", "3z2r2", "", ""],
            [ "", "", "", "", "", "", "", "", "zx", ""],
            [ "", "", "", "", "", "", "", "", "", "x2y2"]],

        "functions" : "hyb.json"
    },

    "beta" : 50,
    "mu" : 27.15110125913311,

    "thermalisation time" : 15,
    "measurement time" : 60,

    "green matsubara cutoff" : 50
}
```

The first four keys define the impurity model.

- `beta` [*real*]: The inverse temperature.
- `mu` [*real*]: The chemical potential.
- `hloc`: The impurity Hamiltonian H_{loc}.
  - `one body`: The one particle tensor t_{ij} in matrix form.
  - `two body`: The two body tensor V_{ijkl}, flattened along $i \to j \to k \to l$,[3] in terms of an array.

  Important: Elements of the tensors which vanish by (abelian) symmetries should be entered as zero, and not just a small number. Otherwise the automatic partitioning of the impurity Hilbert space into invariant sub-spaces of H_{loc} is not optimal, and the simulation is slowed down. The partitioning can be checked by means of the number of sub-spaces and the maximal dimension, which are printed in the terminal at the beginning of the simulation.
- `hybridisation`: The hybridisation function Δ is defined in two steps.
  - `matrix`: First, the entries Δ_{ij} of the hybridisation function are labeled (empty quotes represent entries which are zero). Non-zero off-diagonal entries (not shown here) are supported, as long as the hybridisation function is block diagonal and all entries in a block are non-zero. It is also sufficient that the hybridisation function acquires a block diagonal form after some permutation σ of the orbitals, that is, $(\Delta_{\sigma(i)\sigma(j)})_{ij}$ is block diagonal.
  - `functions` [*string*]: Second, a file (here `hyb.json`) contains the function for each label.

```
{
    "3z2r2" : {
        "imag" : [ -0.567907, -0.757559, -0.76344, -0.72206, -0.67151, ... ],
        "real" : [ -0.55361, -0.39431, -0.2635, -0.18363, -0.134052, ... ]
    },
    "x2y2" : {
```

[3] This is equivalent to $k \to l \to j \to i$ is since H_{loc} is assumed to be real

```
        "imag" : [ -2.336763, -2.150589, -1.81075, -1.56728, -1.391208, ... ]
        "real" : [ -0.80103, 0.025778, 0.23724, 0.282958, 0.28262, ... ]
    },
    .
    .
    .
}
```

The functions are written in Matsubara frequencies ω_n, with an array $[\mathrm{Re}\Delta(i\omega_1), \mathrm{Re}\Delta(i\omega_2), \dots]$ for the `real` part and an array $[\mathrm{Im}\Delta(i\omega_1), \mathrm{Im}\Delta(i\omega_2), \dots]$ for the `imag`-inary part. The positive Matsubara frequencies are sufficient to define the hybridisation function since $\Delta_{ij}(-i\omega_n) = \overline{\Delta_{ji}}(i\omega)$ by hermicity of the impurity model hamiltonian.

Note: The tail of the functions is added by the impurity solver, that is, the functions should be entered up to frequencies where the asymptotic behavior $\propto (i\omega_n)^{-1}$ is reached.

Here I have specified the json value associated with a key, except if it is a data structure. The JSON format does not distinguish between *integer* numbers $\mathbb{Z}$ and *real* numbers $\mathbb{R}$. I distinguish here between them, and a number in $\mathbb{R} \setminus \mathbb{Z}$ is not accepted if an *integer* number is expected. Apart from that, both *real* and *integers* number can be entered in any of the common formats, e.g., it is legal to write `1.2e2` if an *integer* is expected or `30` if a *real* number is expected). The remaining three keys are

- `thermalisation time` [*real*]: Thermalisation stage of the simulation in minutes. To specify the thermalisation stage in Metropolis-Hasting steps, replace this key by `thermalisation steps` [*integer*].
- `measurement time` [*real*]: Measurement stage of the simulation in minutes. To specify the measurement stage in Metropolis-Hasting steps, replace this key by `measurement steps` [*integer*].
- `green matsubara cutoff` [real]: The cutoff energy for measuring the Green function. For example, if the energy is 30, the Green function is measured for all frequencies with $0 < \omega_n < 30$.

These are the mandatory input parameters. The optional input parameters are:

- `dyn` [*string*]: This file (here `dyn.json`) contains F in Eq. 19 as an array

```
[-9, -8.997779, -8.991126, -8.980058, -8.964609, -8.944823, -8.920760, ... ]
```

in Matsubara frequencies, starting with $F(i\nu_n = 0)$. Not listing this parameter in the `params.json` input file means no retarded interaction.

Note: As for the hybridisation functions, the tail is added by the impurity solver, that is, the function $F(i\nu_n)$ should be entered up to frequencies where the asymptotic behavior $\propto (i\nu_n)^{-2}$ is reached.

- `green bulla` [*boolean*]: If `true`, Bulla's trick is used to calculate the self-energy. Rather than obtaining the self-energy from measuring the Green function and using Dyson's equation $\Sigma = \mathcal{G}_0^{-1} - G^{-1}$, the equation of motion of the Green function is used to write the self-energy as a product $\Sigma = LG^{-1}$, and the correlation function

$$L_{ij}(\tau) = -\langle [c_i, \hat{V}](\tau) c_j^\dagger \rangle - \int_0^\beta F(\tau - \tilde{\tau}) \langle \hat{N}(\tilde{\tau}) c_i(\tau) c_j^\dagger \rangle d\tilde{\tau} \tag{27}$$

is measured in addition to the Green function. This is Bulla's trick. Here $\hat{V}$ is given by the static interaction tensor V_{ijkl} on the impurity.

Figure 3.1 compares the self-energies (of a d-Shell impurity model) obtained with Bulla's trick and with Dyson's equation (which is used if the parameter is `false`). Bulla's trick reduces the noise at higher Matsubara frequencies. This is expected since the error when using Dyson's equation is $\delta\Sigma = G^{-2}\delta G$, while the error when using Bulla's trick is $\delta\Sigma = LG^{-2}\delta G + G^{-1}\delta L$. The terms $LG^{-2}\delta G$ and $G^{-1}\delta L$ are small compared to $G^{-2}\delta G$ at higher frequencies because L and G are small (supposing $\delta G \approx \delta L$). Since the two approaches seem equivalent in terms of noise at low frequencies, the default is using Bulla's trick. If one is not interested in high-quality self-energies, Bulla's Trick should not be used since it may slow down the simulation.

One possibility to obtain an estimator for the first term on the right hand side of Eq. 27 is to evaluate the commutator $[c, \hat{V}] = \sum_{ijkl} V_{ijkl}[c, c_i^\dagger c_j^\dagger c_k c_l]$, and to use worm sampling for the terms of the form

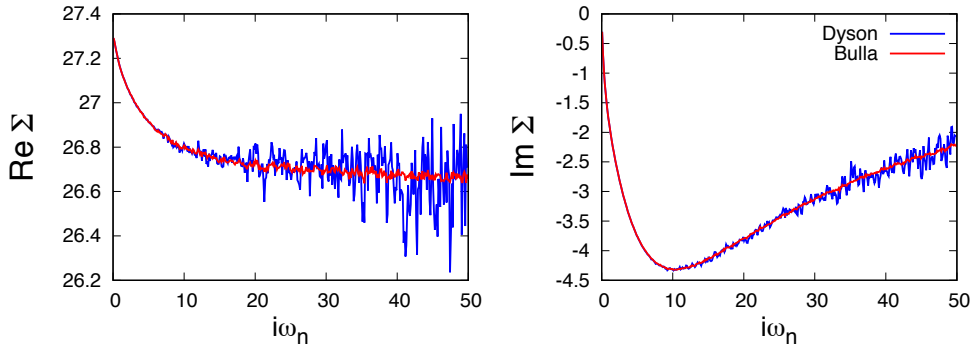

Figure 1: Real and imaginary part (left and right panel, respectively) of the xy self-energy of a d-shell impurity model with retarded interaction, obtained with Dyson's equation by measuring the green function (blue cruves) and with Bulla's trick by measuring the Green function and an additional correlation function (red curves). The same computational resources are used in both cases.

$-\langle (c_i^\dagger c_j c_k)(\tau) c_l^\dagger \rangle$, as presented in [2]. Another possibility is, loosely speaking, to use Eq. 23 and replace c with $[c, \hat{V}]$ to obtain the estimator

$$L_{ij}^{(1)}(i\omega_n) = Z^{-1} \sum_{\mathcal{C}} w(\mathcal{C}) \sum_{r,s=1}^{|\mathcal{C}|} -\beta^{-1} e^{i\omega_n(\tau_s' - \tau_r)} \delta_{i,i_s'} \delta_{j,i_r} (M^{-1})_{sr} \frac{w_V(\mathcal{C}, s)}{w(\mathcal{C})} \tag{28}$$

for the first term in the right hand side of Eq. 27. Here $w_V(\mathcal{C}, s)$ is obtained from the term $w(\mathcal{C})$ of the partition function expansion by replacing $c_{i_s'}$ with $[c_{i_s'}, \hat{V}]$ in the trace of the impurity operator product. It is reasonable to assume that $w_V(\mathcal{C}, s)$ is zero if $w(\mathcal{C})$ is zero since $\hat{V}$ transforms as the identity under the symmetry operations of the impurity, so that Eq. 17 is satisfied. I use the estimator Eq. 28 since worm sampling is expensive. An estimator for the second term on the right hand side of Eq. 27 is obtained in the same way (and the integral can be done analytically since the total charge N commutes with H_{loc}).

- `green basis` [*string*]: The Green function (and the correlation function L) can be measured in any basis. One option here is `matsubara`, where the Matsubara-Fourrier basis $e^{i\omega_n \tau}$ is used, and `green matsubara cutoff` determines the number of Matsubara frequencies which are measured, see above. The other option is `legendre` and uses a basis of legendre polynomials. In this case, one needs to specify `green legendre cutoff`, see below. The default is `matsubara`.

- `green legendre cutoff` [*integer*]: The number of legendre polynomials used for measuring the Green function (and the correlation function L). If this number is too small, *systematic errors are introduced*, and if this number is too big, the noise filtering property of the Legendre basis is diminished. This parameter has *no default value*.

- `complex hybridisation` [*boolean*]: If `true`, complex hybridisation functions $\overline{\Delta}_{ij}(\tau) \neq \Delta_{ij}(\tau)$ are allowed. Notice that since $\overline{\Delta}_{ij}(\tau) = \Delta_{ji}(\tau)$ by hermicity, a hybridisation function is complex if and only if $\Delta_{ij}(\tau) \neq \Delta_{ji}(\tau)$, which means in particular that complex hybridisation function are off-diagonal. If this parameter is `false`, the impurity solver checks if the hybridisation function is real, that is, if $\Delta_{ij}(i\omega_n) = \Delta_{ji}(i\omega_n)$, and stops execution otherwise. The default for this parameter is `false`.

  Note: This option is not yet sufficiently tested.

- `quantum numbers`: The quantum numbers q_{iI} in Eq. 10, given by an array $[q_{1I}, q_{2I}, \ldots, q_{N_O I}]$ for every I (here `N` and `Sz`) as

```
"quantum numbers" : {
    "N" : [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ],
    "Sz" : [-0.5, -0.5, -0.5, -0.5, -0.5, 0.5, 0.5, 0.5, 0.5, 0.5 ]
}
```

  The total charge `N` is defined by default and given here merely by way of an example. Any user defined `N` will be overwritten.

  Note: If the quantum numbers are not compatible with H_{loc}, an error message is displayed. However, at the moment, this error message is displayed in the post-processing only.

- `quantum number susceptibility` [*boolean*]: If true, quantum number susceptibilities Eq. 24 are measured. Similarly to the Green function in Sec. 2.3, an estimator for quantum number susceptibilities is readily obtained with the help of functional derivatives

$$\chi_I(i\nu_n) = -2\frac{\partial \ln Z}{\partial F_{II}(i\nu_n)} = Z^{-1}\sum_{\mathcal{C}} -2\frac{\partial w(\mathcal{C})}{\partial F_{II}(i\nu_n)}, \tag{29}$$

where $i\nu_n \neq 0$. In contrast to the Green function, the condition Eq. 17 is satisfied even if the source field (here F_{II}) is zero, and quantum number susceptibilities can always be obtained by sampling the partition function space. This is because for a term $w(\mathcal{C})$ in the partition function expansion, taking the functional derivative is the same as multiplying by a number, and we have

$$\chi_I(i\nu_n) = -\sum_{\mathcal{C}} \frac{w(\mathcal{C})}{\beta(i\nu_n)^2} \sum_{r,s=1}^{|\mathcal{C}|} (q_{i_r I}e^{-i\nu_n\tau_r} - q_{i'_r I}e^{-i\nu_n\tau'_r})(q_{i_s I}e^{i\nu_n\tau_s} - q_{i'_s I}e^{i\nu_n\tau'_s}) \tag{30}$$

for $i\nu_n \neq 0$, and

$$\begin{aligned}\chi_I(i\nu_n) =& \beta(\langle Q_I^2\rangle - \langle Q_I\rangle^2) + \sum_{\mathcal{C}} \frac{w(\mathcal{C})}{2\beta} \sum_{r,s=1}^{|\mathcal{C}|} q_{i_r I}q_{i_s I}g(\tau_r - \tau_s) - q_{i_r I}q_{i'_s I}g(\tau_r - \tau'_s) \\ & - q_{i'_r I}q_{i_s I}g(\tau'_r - \tau_s) + q_{i'_r I}q_{i'_s I}g(\tau'_r - \tau'_s)\end{aligned} \tag{31}$$

for $i\nu_n = 0$, where $g(\tau) = |\tau|(\beta - |\tau|)$.

The default is `false`.

- `occupation susceptibility direct` [*boolean*]: If `true`, orbital occupation susceptibilities Eq. 25 are measured. Working with functional derivatives is not useful here because there is no field in the action coupling n_i and n_j. It is more straightforward to proceed just as outlined in Eq. 15 and write

$$\chi_{ij}(i\nu_n) = \beta^{-1}\sum_{\mathcal{C}} \iint_0^\beta e^{i\nu_n(\tau_1-\tau_2)} w_\chi(\mathcal{C}, i\tau_1, j\tau_2) d\tau_1 d\tau_2. \tag{32}$$

Here $w_\chi(\mathcal{C}, i\tau_1, j\tau_2)$ is obtained by taking the term $w(\mathcal{C})$ of the partition function expansion and inserting n_i and n_j at imaginary time τ_1 and τ_2 in the trace of the impurity operator product. It is reasonable to assume that $w_\chi(\mathcal{C}, i\tau_1, j\tau_2)$ is zero if $w(\mathcal{C})$ is zero since n_i and n_j are occupation number operators, or in other words, that partition function space sampling is enough and

$$\chi_{ij}(i\nu_n) = \beta^{-1}\sum_{\mathcal{C}} w(\mathcal{C}) \iint_0^\beta e^{i\nu_n(\tau_1-\tau_2)} \frac{w_\chi(\mathcal{C}, i\tau_1, j\tau_2)}{w(\mathcal{C})} d\tau_1 d\tau_2. \tag{33}$$

Since the integrals from 0 to β are expensive, they are included in the sampling, that is, τ_1 and τ_2 is chosen randomly for each sample.

It is instructive to compare the estimator for orbital occupation susceptibilities with the estimator for quantum number susceptibilities. Although less convenient, an estimator for quantum number susceptibilities can be derived as outlined for the orbital occupation susceptibilities in Eqs. 32 and 33. The difference is that the integrals from 0 to β in Eq. 33 can be done *analytically* since Q_I commutes with H_{loc}. For $i\nu_n \neq 0$, the so obtained estimator is the same as in Eq. 30.

It is thus expected that the orbital occupation susceptibilities have much more noise than the quantum number susceptibilities, since the integrals from 0 to β are sampled rather than calculated analytically. For a one to one comparison, calculate the quantum number susceptibilities from the orbital occupation susceptibilities as

$$\chi_I(i\nu_n) = \sum_{ij} q_{iI}q_{jI}\chi_{ij}(i\nu_n). \tag{34}$$

The result is shown in Fig. 3.1 right panel. The default for this parameter is `false`.

Note: For $i\nu_n = 0$, the estimator obtained by proceeding for the quantum number susceptibilities as in Eqs. 32 and 33 looks slightly different from the one in 31, and for some systems, seems to give better results. At the moment I am using the estimator in Eq. 31 for quantum number susceptibilities. I need to do a more careful analysis to decide which one is better.

- `occupation susceptibility bulla` [*boolean*]: If `true`, orbital occupation susceptibilities are measured, but proceeding in a way similar to Bulla's trick. First, integration by parts and the equation of motion are used to write

$$\chi_{ij}(i\nu_n) = \frac{1}{(i\nu_n)^2}\Big(\langle[[H,n_i],n_j]\rangle - \beta^{-1}\iint_0^\beta e^{i\nu_n(\tau_1-\tau_2)}\langle[H,n_i](\tau_1)[H,n_j](\tau_2)\rangle d\tau_1 d\tau_2\Big) \tag{35}$$

for $i\nu_n \neq 0$, and

$$\chi_{ij}(i\nu_n) = \beta(\langle n_i n_j\rangle - \langle n_i\rangle\langle n_j\rangle) + (2\beta)^{-1}\iint_0^\beta g(\tau_1-\tau_2)\langle[H,n_i](\tau_1)[H,n_j](\tau_2)\rangle d\tau \tag{36}$$

for $i\nu_n = 0$, where $g(\tau) = |\tau|(\beta - |\tau|)$ and H is the Hamiltonian of the impurity problem.[4] Evaluating the commutators yields

$$\begin{aligned}\langle[H,n_i](\tau_1)[H,n_j](\tau_2)\rangle = \langle[H_{\rm loc},n_i](\tau_1)[H_{\rm loc},n_j](\tau_2)\rangle + \langle[H_{\rm loc},n_i](\tau_1)[H_{\rm hyb},n_j](\tau_2)\rangle \\ + \langle[H_{\rm hyb},n_i](\tau_1)[H_{\rm loc},n_j](\tau_2)\rangle + \langle[H_{\rm hyb},n_i](\tau_1)[H_{\rm hyb},n_j](\tau_2)\rangle,\end{aligned} \tag{37}$$

where $H_{\rm hyb}$ is the hybridisation Hamiltonian between impurity and bath (which is, together with the Hamiltonian of the electronic bath, encapsulated in $S_{\rm hyb}$). Next, an estimator for

$$\iint_0^\beta e^{i\nu_n(\tau_1-\tau_2)}\langle[H,n_i](\tau_1)[H,n_j](\tau_2)\rangle d\tau_1 d\tau_2 \quad \text{and} \quad \iint_0^\beta g(\tau_1-\tau_2)\langle[H,n_i](\tau_1)[H,n_j](\tau_2)\rangle d\tau_1 d\tau_2 \tag{38}$$

is obtained, proceeding as outlined in Eqs. 32 and 33. In this estimator, all the integrals involving an imaginary time associated with $H_{\rm hyb}$ can be done *analytically*. The integration over times associated with $H_{\rm loc}$ has to be included in the Monte-Carlo sampling, as in the case without equation of motion.

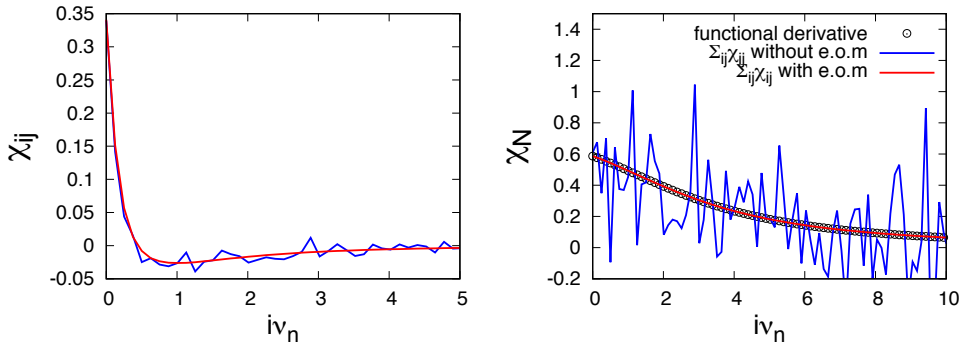

Figure 2: Left panel: orbital occupation susceptibility obtained with equation of motion (red curve, see parameter `occupation susceptibility bulla`) and without (blue curve, see parameter `occupation susceptibility direct`). Right panel: The black circles show the total charge susceptibility obtained from functional derivatives Eq. 29. Also shown is the total charge susceptibility calculated by summing over the orbital occupation susceptibilities, obtained with and without equation of motion (red and blue curve, respectively).

An obvious advantage of using the equation of motion for higher frequencies is that corrections to the asymptotic behavior are measured, and that the noise in these corrections gets suppressed by $(i\nu_n)^{-2}$, see Fig. 3.1 left panel.

Analytically integrating at least some terms in the estimator for Eq. 38 reduces noise, and has an interesting consequence when calculating quantum number susceptibilities from orbital occupation susceptibilities. Using in Eq. 34 the $\chi_{ij}(i\nu_n)$ obtained from the equation of motion in Eqs. 35 and 36, all terms coming from Eq. 37 with an $H_{\rm loc}$ vanish since $[H_{\rm loc}, Q_I] = 0$, and we have

$$\chi_I(i\nu_n) = \sum_{ij}\frac{q_{iI}q_{jI}}{(i\nu_n)^2}\Big(\langle[[H_{\rm hyb},n_i],n_j]\rangle - \beta^{-1}\iint_0^\beta e^{i\nu_n(\tau_1-\tau_2)}\langle[H_{\rm hyb},n_i](\tau_1)[H_{\rm hyb},n_j](\tau_2)\rangle d\tau_1 d\tau_2\Big) \tag{39}$$

for $i\nu_n \neq 0$, and

$$\chi_I(i\nu_n) = \sum_{ij} q_{iI}q_{jI}\Big(\beta(\langle n_i n_j\rangle - \langle n_i\rangle\langle n_j\rangle) + (2\beta)^{-1}\iint_0^\beta g(\tau_1-\tau_2)\langle[H_{\rm hyb},n_i](\tau_1)[H_{\rm hyb},n_j](\tau_2)\rangle d\tau_1 d\tau_2\Big) \tag{40}$$

[4] One needs to play with integration constants in order to obtain Eq. 36.

for $i\nu_n = 0$. In the estimator for Eqs. 39 and 40, all the integrals from 0 to β are done analytically, and the so obtained quantum number susceptibilities are as good as the quantum number susceptibilities obtained from Eq. 29, see Fig. 3.1 right panel. In fact, the estimator for Eqs. 39 and 40 is given by Eqs. 30 and 31.

The default for this parameter is `false`.

Note: It can be challenging to obtain orbital occupation susceptibilities (and even quantum number susceptibilities) of reasonable quality, for example when studying the Kondo effect.

- `susceptibility cutoff` [*real*]: The cutoff energy for measuring susceptibilities, defined in the same way as for the Green function. This parameter must be provided when measuring susceptibilities. There is no default value.

- `density matrix precise` [*boolean*]: The impurity reduced density matrix $\hat{\rho}$ is defined by $\langle\hat{O}\rangle = \mathrm{Tr}\hat{O}\hat{\rho}$ for any static observable $\hat{O}$ acting on the impurity. Its elements $\langle n|\hat{\rho}|m\rangle$, where $|n\rangle$ is a basis of the impurity, can be measured by inserting $|m\rangle\langle n|$ in the trace of the impurity operator product. The straightforward and cheap way is to always insert $|m\rangle\langle n|$ at imaginary time $\tau = 0$, and this is done when the parameter is `false`. Statistics can be improved by using time translational invariance and integrating $|m\rangle\langle n|$ from 0 to β. This is done when the parameter is `true` and is more expensive. However, a preciser density matrix does not only improve static observables, but also moments of correlation functions and orbital occupation resolved susceptibilities. The default is `false`.

- `clean` [*integer*]: Number of Monte-Carlo steps between recalculating the determinant of the hybridization functions from scratch, see [1] and [3] for details on how the determinants are handled in CTQMC. The default is 50000.

- `seed` [*integer*]: Pseudo random number generator (Mersenne Twister 19937) seed. The default is 41085.

- `seed inc` [*integer*]: MPI rank increment of the seed, that is, `seed` + $r\times$`seed inc` is the seed for rank r. The default is 857.

- `expansion histogram` [*boolean*]: The expansion order histrogram is measured if this parameter is `true`. The default is `true`.

- `sweepA` [*integer*]: Number of Markov-Chain steps between sampling observables which are either better sampled often since the estimators are not efficient, for example orbitally resolved susceptibilities, or which are quite cheap, such as the Green function. The default is 50.

- `sweepB` [*integer*]: Number of Markov-Chain steps between sampling observables which are quite expensive (but have good estimators, such as the correlation function L in Bulla's trick or the precise density matrix). The default is 250.

- `trunc dim` [*integer*]: Cuts the rows i and columns i of the operator matrices if i and $j >$ `trunc dim`, respectively. The default is no truncation.

  Note: Bulla's Trick and Dyson's equation for calculating the self-energy may yield different results when truncating the size of the operator matrices. The same holds for the orbital occupation susceptibilities obtained with and without equation of motion.

3.2 Real Materials

Entering the interaction tensor can be tedious. To make the users life easier, some parametrisations of the interaction, which are common in real materials simulations, are implemented. Since the interaction tensor (generally) depends on the one particle basis and the user still has to provide the orbital energies t_{ij} as well as the hybridisation Δ_{ij}, we first define the one particle basis.

3.2.1 Basis

There are two options for the basis, the spherical harmonics and a generic basis.

1) For the spherical harmonics, add the key

```
"basis" : {
    "orbitals" : "d",
    "type" : "product",
    "transformation" : [
        [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
        [ 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 ],
        [ 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 ],
        [ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 ],
        [ 0, 0, 0, 0, 0, 0, 1, 0, 0, 0 ],
        [ 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 ] ]
}
```

to the `params.json` input file . Here

- `orbitals` [*string*]: The angular quantum number ℓ. Possible values are `s` ($\ell = 0$), `p` ($\ell = 1$), `d` ($\ell = 2$) and `f` ($\ell = 3$).
- `type` [*string*]: The `product` basis $|\ell, m, \sigma\rangle$ or the `coupled` basis $|j, m_j\rangle$.
  - The product basis is defined as
    $$|\ell, m, \sigma\rangle := Y_{\ell,m} \otimes \sigma, \tag{41}$$
    where σ is the spin and $Y_{\ell,m}$ are the real spherical harmonics, which are defined as
    $$Y_{\ell,m} := \begin{cases} \frac{i}{\sqrt{2}} \left(Y_\ell^m - (-1)^m Y_\ell^{-m}\right) & \text{if } m < 0 \\ Y_\ell^0 & \text{if } m = 0 \\ \frac{1}{\sqrt{2}} \left(Y_\ell^{-m} + (-1)^m Y_\ell^m\right) & \text{if } m > 0 \end{cases} \tag{42}$$
    in terms of the complex spherical harmonics Y_ℓ^m. The product basis will be enumerated as
    $$|\ell, -\ell, \downarrow\rangle,\ |\ell, -\ell+1, \downarrow\rangle, \ldots,\ |\ell, \ell, \downarrow\rangle,\ |\ell, -\ell, \uparrow\rangle,\ |\ell, -\ell+1, \uparrow\rangle, \ldots,\ |\ell, \ell, \uparrow\rangle \tag{43}$$
    in the following.
  - The coupled basis is defined as
    $$|j, m_j\rangle := \sum_{m\sigma} Y_\ell^m \otimes \sigma \langle \ell, m; \tfrac{1}{2}, \sigma | j, m_j \rangle, \tag{44}$$
    where $\langle \ell, m; \frac{1}{2}, \sigma | j, m_j \rangle$ are the Clebsch-Gordan coefficients, and will be enumerated as
    $$|\ell - \tfrac{1}{2}, -\ell + \tfrac{1}{2}\rangle, \ldots,\ |\ell - \tfrac{1}{2}, \ell - \tfrac{1}{2}\rangle,\ |\ell + \tfrac{1}{2}, -\ell - \tfrac{1}{2}\rangle, \ldots,\ |\ell + \tfrac{1}{2}, \ell + \tfrac{1}{2}\rangle \tag{45}$$
    in the following.
- `transformation`: A transformation matrix U which finally defines the basis as
  $$c_i = \sum_j U_{ij} \tilde{c}_j, \tag{46}$$
  where $\tilde{c}_j$ is the basis defined by `orbitals` and `type` above. The transformation matrix does not need to be square i.e. can be a projection. The transformation defaults to the identity of not provided.

  The example `basis` above thus yields
  $$|\ell, m, \sigma\rangle = |2, -2, \downarrow\rangle,\ |2, -1, \downarrow\rangle,\ |2, 1, \downarrow\rangle,\ |2, -2, \uparrow\rangle,\ |2, -1, \uparrow\rangle,\ |2, 1, \uparrow\rangle \tag{47}$$
  with $|\ell, m, \sigma\rangle$ given by Eq. 41, that is, a basis of the t_{2g} orbitals $\{xy, yz, zx\}$.

Important: The orbital energies and the hybridisation function have to be entered in the transformed basis defined by Eq. 46.

2) For the generic basis, add the key

```
"basis" : {
    "orbitals" : 3
}
```

to the `params.json` input file. Here

- `orbitals` [*integer*]: Defines a basis of D orbitals with spin as

$$|1,\downarrow\rangle, \ldots, \ |D,\downarrow\rangle, \ |1,\uparrow>, \ldots, \ |D,\uparrow\rangle. \tag{48}$$

The parameters `type` and `transformation` are not defined for this basis.

3.2.2 Interaction Tensor

Having defined a basis, the interaction tensor can now conveniently be defined by a few parameters.

1) For the basis of spherical harmonics, replace the

```
"two body" : [ 0, 0, 0, 0, ... ]
```

key in the `hloc` object by

```
"two body" : {
    "parametrisation" : "slater-condon",
    "F0": 15.0,
    "F2": 5.169230769230769,
    "F4": 3.230769230769231,
    "approximation": "none"
},
```

were

- `parametrisation`: The type of the parametrisation. At the moment, only the Slater-Condon parametrisation of the interaction is implemented. *With respect to the basis $Y_\ell^m \otimes \sigma$ of complex spherical harmonics*, the tensor is defined by

$$\tilde{V}_{m_1\sigma_1 m_2\sigma_2 m_3\sigma_3 m_4\sigma_4} = \delta_{\sigma_1\sigma_4}\delta_{\sigma_2\sigma_3} \sum_{k=0,2,\ldots,2\ell} \frac{4\pi}{2k+1} F^k \sum_{m=-k}^{k} \langle Y_2^{m_1} Y_k^m | Y_2^{m_4}\rangle \langle Y_2^{m_2} | Y_k^m Y_2^{m_3}\rangle. \tag{49}$$

- `F0`, `F2`, ... [*real*]: The Slater-Condon parameters. There are $\ell + 1$ parameters for an ℓ-shell.
- `approximation` [*string*]: Permits to truncate the interaction tensor written in the basis defined in Sec. 3.2.1. The options are `none`, i.e. no truncation, and `ising`, i.e. only the elements $V_{\alpha\beta\beta\alpha}$ and $V_{\alpha\beta\alpha\beta}$ of the interaction tensor are kept.

2) For the generic basis, replace the

```
"two body" : [ 0.1, 0, 0, 0, 0, 0, 0, 0, 0.2, 0, ... ],
```

key in the `hloc` object by

```
"two body" : {
    "parametrisation" : "kanamori",
    "U" : 4,
    "J" : 0.2,
    "Uprime" : 4.7,
    "approximation": "none"
},
```

were

- `parametrisation` [*string*]: The type of the parametrisation. At the moment, only the Kanamori parametrisation is implemented. In the $2 \times D$ dimensional generic basis, the tensor is defined by

$$V_{m_1\sigma_1 m_2\sigma_2 m_3\sigma_3 m_4\sigma_4} = \delta_{\sigma_1\sigma_4}\delta_{\sigma_2\sigma_3} v_{m_1 m_2 m_3 m_4} \tag{50}$$

  with $1 \leq m \leq D$, and the non-zero entries of $v_{m_1 m_2 m_3 m_4}$ are

$$v_{mmmm} = U, \quad v_{mnnm} = U' \quad \text{and} \quad v_{mmnn} = v_{mnmn} = J, \tag{51}$$

  where $n \neq m$.

- `U, J, Uprime` [*real*]: The Kanamori parameters defined above. The key `Uprime` is optional and defaults to $U - 2J$ if not provided.

- `approximation` [*string*]: Permits to truncate the interaction tensor. The options are `none`, that is no truncation, and `ising`, that is only the elements $V_{\alpha\beta\beta\alpha}$ and $V_{\alpha\beta\alpha\beta}$ of the interaction tensor are kept.

3.2.3 Quantum Numbers

For the product basis of spherical harmonics and for the generic basis, the quantum number S_z is defined and can be used as

```
"quantum numbers" : {
    "Sz" : {}
}
```

For the coupled basis of spherical harmonics, the quantum number J_z is defined and can be used as

```
"quantum numbers" : {
    "Jz" : {}
}
```

3.3 OUTPUT

The output files are

- a `params.meas.json` file, containing all the measurements collected during the Monte-Carlo simulation in raw format (interesting only for expert user), see Sec. 4 for post-processing of this file.

- a `params.info.json` file, containing information about the simulations, for example the total number of Monte-Carlo steps. This file is quite minimalistic at the moment.

- `config_{x}.json` files, containing the state of the Markov-Chain at the end of the simulation (one for each MPI rank x in parallel mode).

- a `hloc.json` file, containing information about the local hamiltonian H_{loc} (interesting only for expert user).

Note: The `hloc.json` and `params.meas.json` files contain binary data (endian safe encoded in Base64 format).

3.4 Execution

The CTQMC code is in the `ctqmc/host` folder. To run the simulation, type

```
./CTQMC params
```

in terminal. Do not append the `.json` extension of the parameter file!

4 Post-processing

The convention for the input and output files is the same as in Sec. 3.

4.1 INPUT

First, all input and output files of the Monte-Carlo simulation. Parameters for the post-processing are added (as JSON keys) to the `params.json` input file. It depends on the parameters chosen in Sec. 3 which parameters must be added. Two of the parameters are data structures

```
"observables" : {
    "S2" : {
        "one body": [[ ... ], ..., [ ... ]]
        "two body": [ ... ]
    }
},

"probabilities" : ["N", "energy", "S2", "Sz"]
```

where

- `observables`: Static observables on the impurity that can be evaluated with the impurity reduced density matrix, see Eq. 26. The one and two body tensors defining the observables, here only hinted at, are encoded as for the local Hamiltonian in Sec. 3.1, with the exception that the order of operators associated with the two body tensor is $c^\dagger c c^\dagger c$ instead of $c^\dagger c^\dagger c c$.

  Note: The observables are assumed to be invariant under the abelian symmetries of H_{loc}, otherwise an error message is displayed. This is without loss of generality as long as the impurity model has all the symmetries of H_{loc}, which is the case in normal phase simulations.

- `probabilities`: The probability of the impurity eigenstates (that is, the diagonal part of the impurity reduced density matrix) is printed and the eigenstates are lexicographically ordered according to the quantum numbers listed in the array. The quantum numbers here can either refer to the `quantum numbers` or the `observables` defined above, or the `energy` of the eigenstates. Of course, the observables (here `S2`) used to label the eigenstates have to commute with H_{loc}.

  Note: At the moment, the code proceeds in a very lazy way when it comes to use observables for labeling eigenstates. First, it is not checked if the observable commutes with H_{loc}. Second, it is assumed that the observable is diagonal in the eigen basis of H_{loc}. This does obviously not hold in general, but for the cases tested so far, it works fine.

The other parameters are

- `green matsubara cutoff` [*real*]: This parameter was used to define the cutoff for measuring the Green function (and the correlation function L) in the Matsubara basis. If the Legendre basis is used, the measured Green function is available for all Matsubara frequencies, and this parameter defines the frequency where the measurements are replaced by the analytical high frequency tail (determined by the moments). This is useful since the analytical high frequency tail has less fluctuations.

- `susceptibility tail` [*real*]: The parameter `susceptibility cutoff` was used to define the cutoff for measuring the susceptibilities. This parameter defines the cutoff for printing susceptibilities when the tail is added.

4.2 Real Materials

Again, entering the tensor of a static observable like S^2 is tedious, and we provide shortcuts for observables which are common in real materials simulations. This is very simple since we have already defined the one-particle basis in Sec. 3.2.1.

4.2.1 Observables

For the product basis of spherical harmonics and for the generic basis, the observable S^2 is defined and can be used as

```
"observables" : {
    "S2" : {}
}
```

For the coupled basis of spherical harmonics, the observable J^2 is defined and can be used as

```
"observables" : {
    "J2" : {}
}
```

4.3 OUTPUT

All observables are contained in the file `params.obs.json` and labeled by the keys of a JSON object. These are

- the sign of the Monte-Carlo simulation

```
"sign" : 0.876574
```

- the Green function G. The entries G_{ij} of the green function are labeled in the same way as the entries of the hybridisation function in Sec. 3.1, and for every label

```
"green": {
    "3z2r2": {
        "function": {
            "imag": [ -0.45112, -0.50022, -0.47180, -0.43415, -0.39886, -0.36833, ... ],
            "real": [ 0.60445, 0.34176, 0.231310, 0.176048, 0.144529, 0.124280, ... ]
        },
        "moments": [ 1, 4.276767837, 132.3784989 ]
    },
    "x2y2": {
        "function":
            "imag": [ -0.260174, -0.320135, -0.33454, -0.33037, -0.31923, -0.30564, ... ],
            "real": [ 0.18296, 0.114914, 0.095894, 0.088349, 0.0837670, 0.079702, ... ]
        },
        "moments": [ 1, 3.341894444, 129.9656024 ]
    },
    .
    .
    .
}
```

the `function` is printed and the first three `moments` of the Green function

$$G_{ij}(i\omega_n) = \sum_{k\geq 1} \frac{g_{ij}^{(k)}}{(i\omega_n)^k} \tag{52}$$

are given as an array $[g_{ij}^{(1)}, g_{ij}^{(2)}, g_{ij}^{(3)}]$. They are calculated as

$$\begin{aligned} g_{ij}^{(2)} &= \langle\{[H, c_i], c_j^\dagger\}\rangle \\ &= \langle\{[H_{\text{loc}}, c_i], c_j^\dagger\}\rangle - \delta_{ij} F(i\nu_n = 0)\langle N\rangle \end{aligned} \tag{53}$$

and

$$\begin{aligned} g_{ij}^{(3)} &= \langle\{[H, c_i], [H, c_j]^\dagger\}\rangle \\ &= \langle\{[H_{\text{loc}}, c_i], [H_{\text{loc}}, c_j]^\dagger\}\rangle - \int_0^\beta F(\tau)\langle N(\tau)(\{c_i, [H_{\text{loc}}, c_j]^\dagger\} + \{[H_{\text{loc}}, c_i], c_j^\dagger\})\rangle d\tau \\ &\quad + \delta_{ij}\left(-F(\tau = 0) + \iint_0^\beta F(\tau)F(\tau')\langle N(\tau)N(\tau')\rangle d\tau d\tau'\right) + h_{ij} \end{aligned} \tag{54}$$

where H is the impurity model Hamiltonian and h_{ij} is the first moment of the hybridisation function. Notice that the Green function satisfies $G_{ij}(-i\omega) = \overline{G_{ji}}(i\omega_n)$ by hermicity of the impurity model Hamiltonian, and that $G_{ij}(i\omega) = G_{ji}(i\omega_n)$ if the impurity model Hamiltonian is real (which is the case if the hybridisation function is real). For the moments this translates to $g_{ij} = \overline{g}_{ji}$ and $g_{ij} = g_{ji}$, respectively. That is, the moments are real if the hybridisation function is real. If the hybridisation function is complex, the real and imaginary parts are listed as

```
"moments": {
    "imag": [ 0, 0.3322323, 12.235435 ],
    "real": [ 1, 3.532434, 112.345234 ]
}
```

- the self-energy Σ, printed in the same format as the Green function

```
"self-energy": {
    "3z2r2": {
        "function": {
            "imag": [ -0.161396, -0.416685, -0.632350, -0.81619, -0.97799, -1.12554, ... ],
            "real": [ 27.1923, 27.1658, 27.127, 27.0821, 27.0332, 26.9784, 26.92767, ... ]
        },
        "moments": [ 23.42464529, 112.8155748 ]
    },
    "x2y2": {
        "function": {
            "imag": [ -0.170162, -0.42747, -0.63979, -0.81713, -0.97271, -1.11371, ... ],
            "real": [ 26.744, 26.7320, 26.727, 26.7125, 26.7007, 26.6821, 26.661,  ... ]
        },
        "moments": [ 24.40935271, 116.271884 ]
    },
    .
    .
    .
}
```

The first two `moments` of the self-energy

$$\Sigma_{ij}(i\omega) = s_{ij}^{(0)} + \frac{s_{ij}^{(1)}}{i\omega_n} + O(\omega_n^{-2}) \tag{55}$$

are given as an array $[s_{ij}^{(0)}, s_{ij}^{(1)}]$. They are (easily) calculated from the Green function moments above with Dyson's equation, and are complex if the hybridisation function is complex.

A high frequency tail of the form

$$\Sigma_{ij}^{\infty}(i\omega_n) = s_{ij}^{(0)} + \frac{s_{ij}^{(1)}}{i\omega_n + a_{ij} + \frac{b_{ij}}{i\omega_n}} \tag{56}$$

is added up to *the cutoff of the hybridisation function*. The parameters a_{ij} and b_{ij} are chosen as to best match the last measured frequencies of the self-energy.

If Bulla's trick is used, the self-energy from the Dyson equation is calculated in addition and listed as `self-energy-dyson` (but without high frequency tail). If Bulla's trick is not used, the self-energy from the Dyson equation is listed as `self-energy` (and has the high frequency tail).

- if the parameter `quantum number susceptibility` is `true`, the susceptibility for every quantum number defined in Sec. 3.1.

```
"susceptibility": {
    "N": {
        "function": [ 0.156115, 0.15509, 0.15378, 0.15230, 0.15072, 0.14886, ... ],
        "moment": [ -5.783031934 ]
    },
    "Sz": {
        "function": [ 16.10882, 6.32185, 3.5623, 2.350757, 1.67991, 1.26989, ... ],
        "moment": [ -1.445757984 ]
    }
}
```

The function is given in Matsubara frequencies as an array $[\chi(i\nu_0), \chi(i\nu_1), \dots]$. Also listed is the first moment

$$m = \sum_{ij} q_i q_j \langle [[H_{\text{hyb}}, n_i], n_j] \rangle, \tag{57}$$

which can be read off Eq. 39. To measure this quantitiy, the identity

$$\langle [[H_{\text{hyb}}, n_i], n_j] \rangle = -\beta^{-1} \delta_{ij} \langle k_i \rangle, \tag{58}$$

where $\langle k_i \rangle$ is the average number of c_i and $c_i^\dagger$ in the partition function expansion, proves useful. A tail is added up to the frequency given by the parameter `susceptibility tail`.

Note: Quantum number susceptibilities are always real, both in imaginary time and in Matsubara frequencies.

- if the parameter `occupation susceptibilities bulla` is `true`, the orbital occupation susceptibilities obtained with the equation of motion.

```
"occupation-susceptibility-bulla": {
    "0_0": {
        "function": [ 1.854944, 0.818625, 0.504099, 0.36852, 0.293480, 0.243108, ... ],
        "moment": [ -0.6202525352 ]
    },
    "0_1": {
        "function": [ 1.06792, 0.327230, 0.143901, 0.0740146, 0.038030, 0.0191289, ... ],
        "moment": [ 0.08284811469 ]
    },
    .
    .
    .
}
```

A key `i_j` refers to the susceptibility χ_{ij}, e.g., the key `0_0` refers to χ_{00}. As above, the function is given in Matsubara frequencies as an array $[\chi(i\nu_0), \chi(i\nu_1), \dots]$ and the moment

$$m_{ij} = \langle [[H_{\text{loc}}, n_i], n_j] \rangle - \beta^{-1} \delta_{ij} \langle k_i \rangle, \tag{59}$$

given by Eqs. 35 and 58, is listed well. A tail is added up to the frequency given by the parameter `susceptibility tail`.

Note: Orbital occupation susceptibilities are always real in Matsubara frequencies.

- if the parameter `occupation susceptibility direct` is `true`, the orbital occupation susceptibilities obtained without the equation of motion.

  They are printed under the key `occupation-susceptibility-direct` in the same way as the ones obtained with equation of motion. The only difference is that no high-frequency tail is added.

- if the parameter `expansion histogram` is `true`, the expansion order histogram

```
"expansion histogram": [ 0, 0, 1.16527e-05, 0, 1.16527e-05, .... ]
```

- the orbital occupations $\langle c_i^\dagger c_j \rangle$, labeled as the entries Δ_{ij} of the hybridisation function.

```
"occupation": {
    "3z2r2": 0.7516048419,
    "x2y2": 0.6750796683,
    "xy": 0.5182722851,
    "yz": 0.5270957541,
    "zx": 0.5270735586
}
```

The off-diagonal orbital occupations are generally complex if the hybridisation is complex. The code makes no distinction between diagonal and off-diagonal components and prints all of them in complex format if the hybridisation is complex.

Note: The occupation given for a label is *per* orbital. That is, to obtain for example the total charge N, one would need to sum up the five occupation numbers and multiply by two (since both spin up and down have the same label).

- the expectation values of the quantum numbers (and the squares thereof), the observables,

```
"scalar": {
    "N": 5.998203761,
    "NN": 36.37620099,
    "S2": 1.985983937,
    "Sz": 0.002529882093,
    "SzSz": 1.32515506,
    "energy": -106.3285554,
    "k": 144.5958299
}
```

as well as the energy on the impurity and the average expansion order. The energy here is defined by $\beta^{-1}\langle S_{\mathrm{loc}} + S_{\mathrm{dyn}}\rangle$. The energy including the hybridisation can be obtained from $\langle S_{\mathrm{hyb}}\rangle = -\langle k\rangle$.

Note: If the parameter `density matrix precise` is `false`, the total charge `N` listed here is more precise than the total charge obtained by summing up the orbital occupations. Otherwise they are the same.

- Probability of the impurity states, that is, the diagonal part of the impurity reduced density matrix.

```
"probabilities": {
    "occupation numbers": [
        [ 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0.03211850711 ],
        [ 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0.03181846887 ],
        [ 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0.02073559621 ],
        [ 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0.0206083657 ],
        [ 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0.01185833975 ],
        [ 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0.01178427485 ],
        .
        .
        .
    },
    "quantum numbers": [
        [ 10, -60.771013, 0, 0, 5.081942411e-08 ],
        [ 9, -80.119911, 0.375, 0.5, 2.045662584e-06 ],
        [ 9, -80.119911, 0.375, -0.5, 2.050142719e-06 ],
        [ 9, -80.169911, 0.375, 0.5, 1.768948312e-06 ],
        [ 9, -80.169911, 0.375, -0.5, 1.764185693e-06 ],
        [ 9, -80.719911, 0.375, 0.5, 4.297462465e-06 ],
        .
        .
        .
    }
}
```

The results are printed in the basis of occupation number states $|n_1, n_2, \dots\rangle = (c_1^\dagger)^{n_1}(c_2^\dagger)^{n_2}\cdots|0\rangle$ and in the basis of eigenstates of H_{loc}.

For the occupation number states, an array first contains the occupations n_i of the state followed by the probability p of the state as $[n_1, n_2, \cdots, n_{N_O}, p]$. The states are in descending order, according to the probability.

For the eigenstates, an array first contains the quantum numbers (as specified by the parameter `probabilities` in Sec. 4.1) of the state, followed by the probability p of the state as $[N, \text{energy}, S^2, S_z, p]$. The states are in descending lexicographical order, according the quantum numbers (the first entry having the highest priority).

4.4 Execution

The post-processing code is in the `evalsim` folder. To run the post-processing, type

```
./EVALSIM params
```

in terminal.

5 Dependencies

The codes are written in C++11 and the required libraries are

- BLAS/LAPACK
- MPI (optional)
- CUDA (optional)

References

[1] Emanuel Gull, Andrew J. Millis, Alexander I. Lichtenstein, Alexey N. Rubtsov, Matthias Troyer, and Philipp Werner. Continuous-time monte carlo methods for quantum impurity models. *Rev. Mod. Phys.*, 83:349–404, May 2011.

[2] P. Gunacker, M. Wallerberger, T. Ribic, A. Hausoel, G. Sangiovanni, and K. Held. Worm-improved estimators in continuous-time quantum monte carlo. *Phys. Rev. B*, 94:125153, Sep 2016.

[3] Kristjan Haule. Quantum Monte Carlo impurity solver for cluster dynamical mean-field theory and electronic structure calculations with adjustable cluster base. *Phys. Rev. B*, 75(15):155113, 2007.

A Examples

This appendix contains some example input files.

A.1 d-Shell

Below is a complete input file for a d-shell impurity with retarded interactions, containing the parameters for both the Monte-Carlo simulation Sec. 3.1 and the post-processing Sec. 4.1:

```
{
    "basis" : {
        "orbitals" : "d",
        "type" : "product real"
    },

    "hloc" : {
        "one body" : [
            [0.02, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, -0.55, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, -0.6, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0.02, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, -0.55, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0, -0.6]],

        "two body" : {
            "parametrisation" : "slater-condon",
            "F0": 15.0,
            "F2": 5.169230769230769,
            "F4": 3.230769230769231,
            "approximation": "none"
        }
    },

    "hybridisation" : {
        "matrix" : [
            [ "xy", "", "", "", "", "", "", "", "", ""],
```

```
            [ "", "yz", "", "", "", "", "", "", "", ""],
            [ "", "", "3z2r2", "", "", "", "", "", "", ""],
            [ "", "", "", "zx", "", "", "", "", "", ""],
            [ "", "", "", "", "x2y2", "", "", "", "", ""],
            [ "", "", "", "", "", "xy", "", "", "", ""],
            [ "", "", "", "", "", "", "yz", "", "", ""],
            [ "", "", "", "", "", "", "", "3z2r2", "", ""],
            [ "", "", "", "", "", "", "", "", "zx", ""],
            [ "", "", "", "", "", "", "", "", "", "x2y2"]],

        "functions" : "hyb.json"
    },

    "dyn" : "dyn.json",

    "beta" : 50,
    "mu" : 27.15110125913311,

    "thermalisation time" : 15,
    "measurement time" : 60,

    "green matsubara cutoff" : 50,

    "quantum number susceptibility" : true,
    "occupation susceptibility direct" : true,
    "occupation susceptibility bulla" : true,
    "susceptibility cutoff" : 50,

    "density matrix precise" : true,

    "quantum numbers" : {
        "Sz" : {}
    },

    "observables" : {
        "S2" : {}
    },

    "susceptibility tail" : 200,
    "probabilities" : ["N", "energy", "S2", "Sz"]
}
```

Since the `transformation` of the `basis` is not specified, it is the identity, so the basis is given by Eq. 43 with $\ell = 2$. Bulla's trick is used and the Green function as well as the correlation function L are measured in the Matsubara basis, since this is the default.

A.2 f-Shell

A complete input file for an f-shell impurity with spin-orbit coupling but without retarded interactions is

```
{
    "basis": {
        "orbitals": "f",
        "type": "coupled"
    },

    "hloc": {
        "one body": [
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
```

```
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 1.08551, 0, 0, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 1.08551, 0, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 1.08551, 0, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.08551, 0, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.08551, 0, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.08551, 0, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.08551, 0 ],
            [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.08551 ] ],

        "two body": {
            "parametrisation": "slater-condon",
            "F0": 4.5,
            "F2": 6.10405,
            "F4": 4.0775,
            "F6": 3.0154,
            "approximation": "none"
        }
    },

    "hybridisation": {
        "functions": "hyb.json",

        "matrix": [
            [ "5/2", "", "", "", "", "", "", "", "", "", "", "", "", "" ],
            [ "", "5/2", "", "", "", "", "", "", "", "", "", "", "", "" ],
            [ "", "", "5/2", "", "", "", "", "", "", "", "", "", "", "" ],
            [ "", "", "", "5/2", "", "", "", "", "", "", "", "", "", "" ],
            [ "", "", "", "", "5/2", "", "", "", "", "", "", "", "", "" ],
            [ "", "", "", "", "", "5/2", "", "", "", "", "", "", "", "" ],
            [ "", "", "", "", "", "", "7/2", "", "", "", "", "", "", "" ],
            [ "", "", "", "", "", "", "", "7/2", "", "", "", "", "", "" ],
            [ "", "", "", "", "", "", "", "", "7/2", "", "", "", "", "" ],
            [ "", "", "", "", "", "", "", "", "", "7/2", "", "", "", "" ],
            [ "", "", "", "", "", "", "", "", "", "", "7/2", "", "", "" ],
            [ "", "", "", "", "", "", "", "", "", "", "", "7/2", "", "" ],
            [ "", "", "", "", "", "", "", "", "", "", "", "", "7/2", "" ],
            [ "", "", "", "", "", "", "", "", "", "", "", "", "", "7/2" ] ]
    },

    "beta": 50.0,
    "mu": 19.4567,

    "green matsubara cutoff": 15,

    "thermalisation time": 0,
    "measurement time": 1
}
```

Since the `transformation` of the `basis` is not specified, it is the identity, so the one-particle basis is given by Eq. 45 with $\ell = 3$. Bulla's trick is used and the Green function as well as the correlation function L are measured in the Matsubara basis, since this is the default. No susceptibilities are sampled.

B Expansion

Expanding the partition function in the retarded parts of the action yields

$$
\begin{aligned}
Z &= \int \mathcal{D}[c^\dagger, c] e^{-S_{\rm loc} - S_{\rm hyb} - S_{\rm dyn}} = \\
&= \sum_{mn} \frac{(-1)^m}{m!} \sum_{\substack{i_1 \cdots i_n \\ j_1 \cdots j_n}} \int_0^\beta d\tau_1 \cdots \int_0^{\tau_{n-1}} d\tau_n \int_0^\beta d\tau'_1 \cdots \int_0^{\tau'_{n-1}} d\tau'_n \sum_{\substack{I_1 \cdots I_m \\ J_1 \cdots J_m}} \int_0^\beta d\tilde{\tau}_1 \cdots d\tilde{\tau}_m \int_0^\beta d\tilde{\tau}'_1 \cdots d\tilde{\tau}'_m \\
&\quad \times \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r) \prod_{s=1}^m Q^\dagger_{I_s}(\tilde{\tau}_s) Q_{J_s}(\tilde{\tau}'_s)] \times \\
&\quad \times \mathrm{Det}[(\Delta_{i_l j_k}(\tau_l - \tau'_k))_{1 \le l,k \le n}] \times \prod_{s=1}^m F_{I_s J_s}(\tilde{\tau}_s - \tilde{\tau}'_s).
\end{aligned}
\tag{60}
$$

We now show that if the particle-hole bilinears commute with the local Hamiltonian and satisfy Eq. 10, the retarded part can be traced out from the expansion. First notice that in this case

$$
\frac{d}{d\tilde{\tau}} \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r) Q^\dagger_I(\tilde{\tau}) Q_J(\tilde{\tau}')] = \tag{61}
$$

$$
\left(\sum_{r=1}^n q_{j_r I} \delta(\tau'_r - \tilde{\tau}) - \overline{q}_{i_r I} \delta(\tau_r - \tilde{\tau}) \right) \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r) Q_J(\tilde{\tau}')] \tag{62}
$$

since $[c_j, Q^\dagger_I] = \overline{q}_{jI} c_j$, $[c^\dagger_i, Q^\dagger_I] = -\overline{q}_{iI} c^\dagger_i$ and the particle-hole bilinears commute. Integration by parts yields

$$
\begin{aligned}
&\int_0^\beta d\tilde{\tau} F_{IJ}(\tilde{\tau} - \tilde{\tau}') \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r) Q^\dagger_I(\tilde{\tau}) Q_J(\tilde{\tau}')] = \\
&\qquad K'_{IJ}(\tilde{\tau} - \tilde{\tau}') \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r) Q^\dagger_I(\tilde{\tau}) Q_J(\tilde{\tau}')] \Big|_{\tilde{\tau}=0}^{\tilde{\tau}=\beta} \\
&\qquad - \left(\sum_{r=1}^n \overline{q}_{j_r I} K'_{IJ}(\tau'_r - \tilde{\tau}') - \overline{q}_{i_r I} K'_{IJ}(\tau_r - \tilde{\tau}') \right) \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r) Q_J(\tilde{\tau}')],
\end{aligned}
\tag{63}
$$

where K_{IJ} is a second primitive of F_{IJ}. *Assuming* that K_{IJ} is β-periodic, the boundary term in Eq. 63 vanishes. Using this result and proceeding in the same way

$$
\begin{aligned}
&\iint_0^\beta d\tilde{\tau} d\tilde{\tau}' F_{IJ}(\tilde{\tau} - \tilde{\tau}') \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r) Q^\dagger_I(\tilde{\tau}) Q_J(\tilde{\tau}')] = \\
&\qquad \left(\sum_{r=1}^n \overline{q}_{j_r I} K'_{IJ}(\tau'_r - \tilde{\tau}') - \overline{q}_{i_r I} K'_{IJ}(\tau_r - \tilde{\tau}') \right) \mathrm{Tr}[e^{-\beta H_{\rm loc}} \mathrm{T}_\tau \prod_{r=1}^n c_{j_r}(\tau'_r) c^\dagger_{i_r}(\tau_r)],
\end{aligned}
\tag{64}
$$

[INCOMPLETE]