Non-crossing and one-crossing approximations for quantum impurity models

Lei Zhang

Department of Physics
University of Michigan
lzphy@umich.edu

April 27, 2022

Overview

- Setup: model & expansions
- Self-consistent resummations
- Measurement
- Programming

Quantum impurity model

- For a general quantum impurity model:

$$H_{\mathrm{QI}} = H_{\mathrm{loc}} + H_{\mathrm{bath}} + H_{\mathrm{hyb}}$$

  - $H_{\mathrm{loc}} = H^0_{\mathrm{loc}} + H^1_{\mathrm{loc}}$ for the impurity

$$H^0_{\mathrm{loc}} = \sum_{ij} t_{ij} d^\dagger_i d_j, \quad H^1_{\mathrm{loc}} = \sum_{ijkl} U_{ijkl} d^\dagger_i d^\dagger_j d_l d_k + \cdots$$

  - H_{bath} for the bath

$$H_{\mathrm{bath}} = \sum_p \varepsilon_p c^\dagger_p c_p$$

  - H_{hyb} for the coupling between impurity and bath

$$H_{\mathrm{hyb}} = \sum_{pj} (V^j_p c^\dagger_p d_j + V^{j*}_p d^\dagger_j c_p)$$

- Partition function:

$$Z = \mathrm{Tr}[e^{-\beta H_{\mathrm{QI}}}] = \mathrm{Tr}[e^{-\beta H_a} T_\tau \exp(-\int_0^\beta d\tau H_b(\tau))] \,,$$

where $H_{\mathrm{QI}} = H_a + H_b$ and $H_b(\tau) = e^{H_a \tau} H_b e^{-H_a \tau}$

Hybridization expansion

- $H_a = H_{\mathrm{loc}} + H_{\mathrm{bath}}$, $H_b = H_{\mathrm{hyb}}$
- A perturbative expansion regrading to H_b

$$Z = \sum_{k=0}^{+\infty} (-1)^k \frac{1}{k!} \int_0^\beta d\tau_1 \cdots \int_0^\beta d\tau_k \mathrm{Tr}[T_\tau e^{-\beta H_a} H_b(\tau_k) \cdots H_b(\tau_1)]$$

- Can be simplified according to the following observations:
  - $H_b = \sum_{pj} (V_p^j c_p^\dagger d_j + V_p^{j*} d_j^\dagger c_p) = \tilde{H}_{\mathrm{hyb}} + \tilde{H}_{\mathrm{hyb}}^\dagger$. Each term contains only one bath operator $c/c^\dagger$
  - Only even k with equal numbers of $\tilde{H}_{\mathrm{hyb}}$ and $\tilde{H}_{\mathrm{hyb}}^\dagger$ can make non-zero contribution
  - Impurity and bath in the trace are decoupled from each other
  - Wick's theorem is valid for bath and therefore can be utilized to simplify the trace

Hybridization expansion

- Definition of hybridization function

$$\Delta_{lm}(\tau_l' - \tau_m) = -\sum_p V_p^{j_l' *} V_p^{j_m} \langle T_\tau c_p(\tau_l') c_p^\dagger(\tau_m) \rangle$$

- Final form of the partition function:

$$Z = Z_{\text{bath}} \sum_{k=0}^{+\infty} \iiint_{\substack{\tau_1 < \cdots < \tau_k \\ \tau_1' < \cdots < \tau_k'}} d\tau_1 \cdots d\tau_k' \sum_{\substack{j_1, \cdots, j_k \\ j_1', \cdots, j_k'}} \text{Tr}_d [T_\tau e^{-\beta H_{\text{loc}}} \times d_{j_k}(\tau_k) d_{j_k'}^\dagger(\tau_k') \cdots d_{j_1}(\tau_1) d_{j_1'}^\dagger(\tau_1')] \det \Delta$$

- Configuration space is described by $\boldsymbol{x} = (k, \boldsymbol{\tau}, \boldsymbol{j})$, where $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_k, \tau_1', \cdots, \tau_k')$ and $\boldsymbol{j} = (j_1, \cdots, j_k, j_1', \cdots, j_k')$
- This is an exact result as long as $H_{\text{hyb}}(\tau)$ is not infinite large

Performing simulations

- Continuous-time quantum Monte Carlo methods (CT-QMC): sample each configuration $\boldsymbol{x}$ according to its weight $w_{\mathrm{loc}}(\boldsymbol{x})\det\Delta(\boldsymbol{x})$
  - Numerically exact
  - May suffer from severe sign problem (exponentially growing errors)

- Self-consistent resummations of diagrams
  - NCA and OCA are two frequently used approximations among various schemes
  - Not suffer from sign problem
  - The computational cost scales polynomially in the system size
  - Not numerically exact and not easy to estimate systemic errors

Expanded configuration space

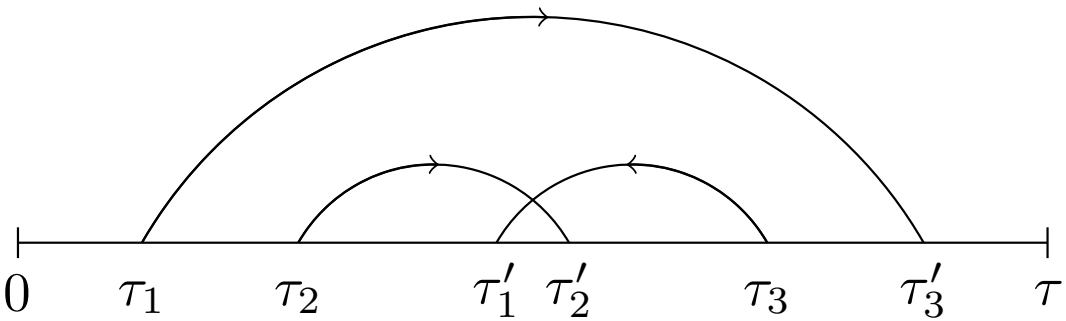
- $\det\Delta$ has $k!$ terms, which can be thought as all possible pairings between $(\tau_1, \cdots, \tau_k)$ and $(\tau_1', \cdots, \tau_k')$
- Different pairings can be represented by curves, called hybridization lines, connecting the corresponding pairs
- Expand the configuration space from $\boldsymbol{x}$ to $\boldsymbol{x}' = (\boldsymbol{x}, s)$, where s is a permutation of $(1, 2, \cdots, k)$
- The corresponding weight is $\mathrm{sgn}(s) w_{\mathrm{loc}}(\boldsymbol{x}) \Delta_{s(1)1} \Delta_{s(2)2} \cdots \Delta_{s(k)k}$

Pseudo-particle propagators

- Define the bare propagator $R_0(\tau) = e^{-H_{\rm loc}\tau}$ and the dressed propagator

$$R(\tau) = \sum_{\boldsymbol{x}'} \mathrm{sgn}(s)\Delta_{s(1)1}\Delta_{s(2)2}\cdots\Delta_{s(k)k} \times T_\tau e^{-\tau H_{\rm loc}} d_{j_k}(\tau_k) d^\dagger_{j'_k}(\tau'_k)\cdots d_{j_1}(\tau_1) d^\dagger_{j'_1}(\tau'_1)\ .$$

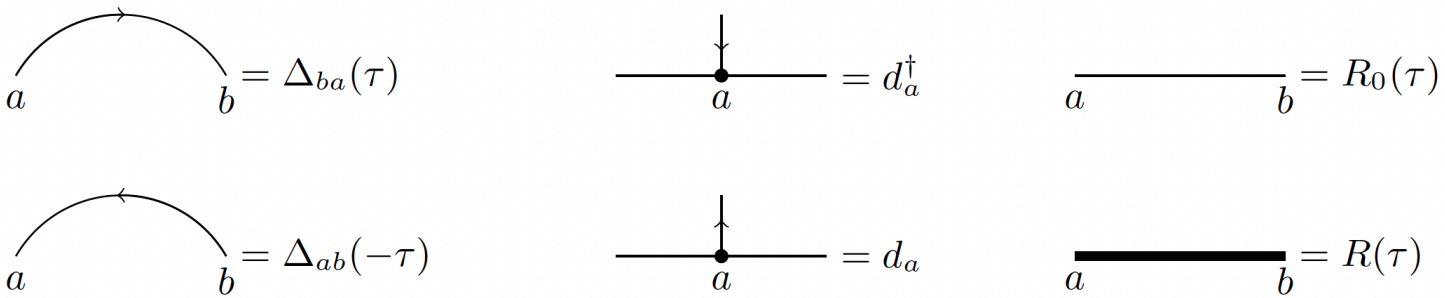
- It is easy to verify that $Z/Z_{\rm bath} = \mathrm{Tr}[R(\beta)]$.
- A typical configuration of $R(\tau)$:



- Any configuration of $R(\tau)$ can be constructed by following steps:

  1. Choose an order $k \in \mathbb{N}$;
  2. Specify values of $\boldsymbol{\tau}$ fulfilling the restrictions $0 < \tau_1 < \cdots < \tau_k < \tau$ and $0 < \tau'_1 < \cdots < \tau'_k < \tau$;
  3. Pair $(\tau_1, \cdots, \tau_k)$ and $(\tau'_1, \cdots, \tau'_k)$ using hybridization lines

- The contribution of each configuration can be written down by an overall sign "sgn" times the contribution calculated from

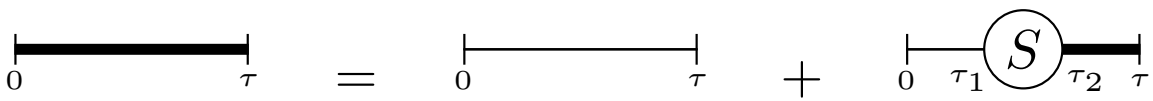


  - right direction: time increasing
  - time increment $\tau = \tau_b - \tau_a$
  - The overall sign comes from
    (i) time-ordering of d and $d^\dagger$ operators
    (ii) pairing between $(\tau_1, \cdots, \tau_k)$ and $(\tau_1', \cdots, \tau_k')$

Self-consistent equation 1

- Dyson equation:

$$R(\tau) = R_0(\tau) + \int_0^{\tau} d\tau_2 \int_0^{\tau_2} d\tau_1 R(\tau - \tau_2) S(\tau_2 - \tau_1) R_0(\tau_1) \ ,$$



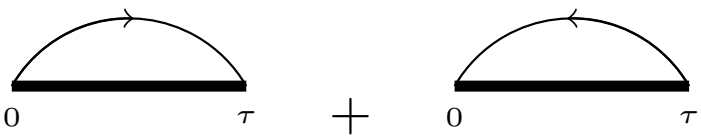
  - $S(\tau)$ ("self-energy") is the set of all one-particle irreducible diagrams

- Another equation to connect S and R?
  - Structure of $S(\tau)$ is very complicated
  - Need do some approximations to get a simple expression.

Self-consistent equation 2 - NCA

- For future discussions, hybridization line is required to be "regular" (cannot have more than one crossover point with another line)
- Only consider configurations without crossover points
- Self-similar structure of $S(\tau) = S_0(\tau)$ gives

$$S_0(\tau) = \quad + \quad$$



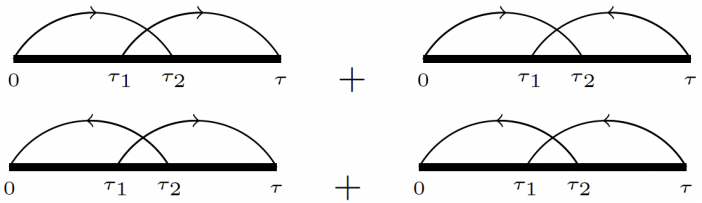
$$= \sum_{ab} \left[\mathrm{sgn}_1 d_b^\dagger R(\tau) d_a \Delta_{ba}(\tau) + \mathrm{sgn}_2 d_b R(\tau) d_a^\dagger \Delta_{ab}(-\tau) \right]$$

- sgn_1 and sgn_2 are signs of corresponding diagrams, which will be analyzed later.

Self-consistent equation 2 - OCA

- Neglects any configurations in which any one hybridization line has more than one crossover points in total with all other lines.
- Self-similar structure of $S(\tau) = S_0(\tau) + S_1(\tau)$ gives

$S_1(\tau)$ =

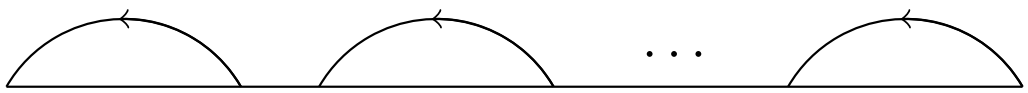


$$= \sum_{abcd} \int_0^{\tau} d\tau_2 \int_0^{\tau_2} d\tau_1 \Big[\mathrm{sgn}_1 d_d^\dagger R(\tau - \tau_2) d_c^\dagger R(\tau_2 - \tau_1) d_b R(\tau_1) d_a \Delta_{db}(\tau - \tau_1) \Delta_{ca}(\tau_2)$$
$$+ \mathrm{sgn}_2 d_d R(\tau - \tau_2) d_c^\dagger R(\tau_2 - \tau_1) d_b^\dagger R(\tau_1) d_a \Delta_{bd}(\tau_1 - \tau) \Delta_{ca}(\tau_2)$$
$$+ \mathrm{sgn}_3 d_d^\dagger R(\tau - \tau_2) d_c R(\tau_2 - \tau_1) d_b R(\tau_1) d_a^\dagger \Delta_{db}(\tau - \tau_1) \Delta_{ac}(-\tau_2)$$
$$+ \mathrm{sgn}_4 d_d R(\tau - \tau_2) d_c R(\tau_2 - \tau_1) d_b^\dagger R(\tau_1) d_a^\dagger \Delta_{bd}(\tau_1 - \tau) \Delta_{ac}(-\tau_2) \Big] .$$

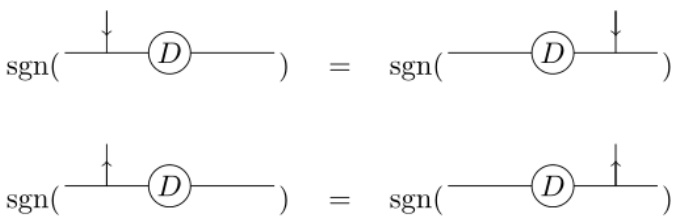
- sgn to be determined later.

Determining the overall sign

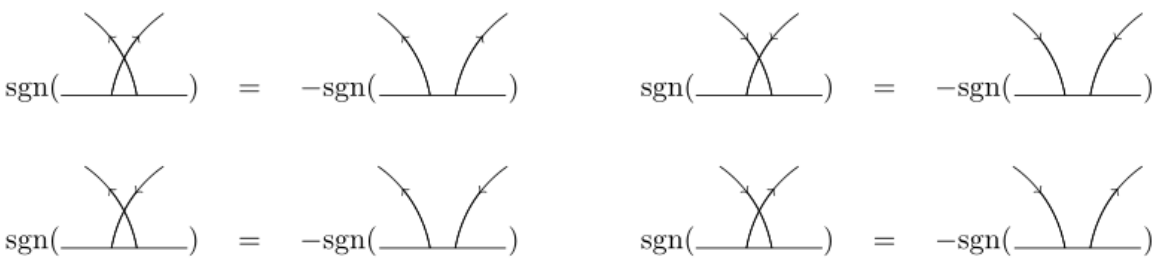
- Find connection to the following configuration which has $+1$ sign
  - $\tau_1' < \tau_1 < \cdots < \tau_k' < \tau_k$
  - pairing (1, 1), (2, 2) $\cdots (k,\ k)$



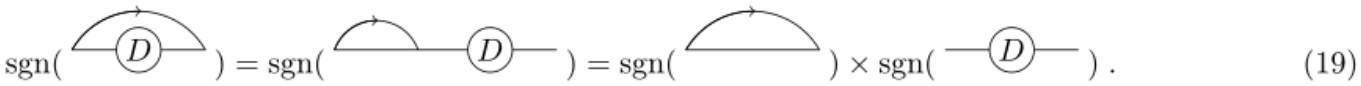
Theorem 1 *Moving a hybridization line across an arbitrary complete diagram D (any hybridization lines starting within D will also end within D) will not change its sign. It has the following graphical expressions:*



Theorem 2 *Switching two adjacent crossing lines will contribute a minus sign. It has the following graphical expressions:*



As an exercise, sgn_1 in NCA can be obtained conveniently:

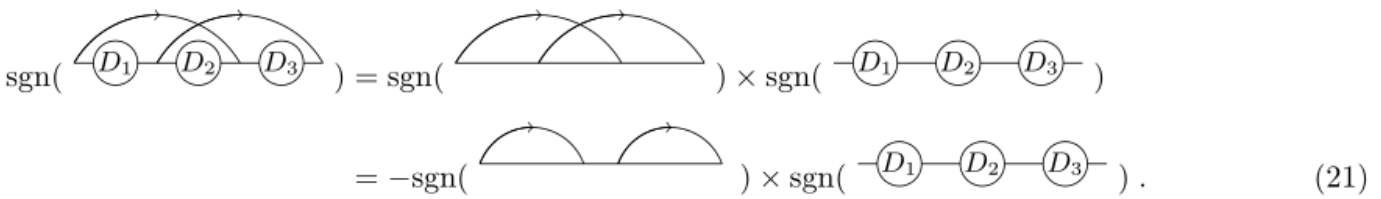


(19)

Because $\mathrm{sgn}($  $)$ has been incorporated into R, we have $\mathrm{sgn}_1 = \mathrm{sgn}($  $) = -1$. Using similar procedure, we get $\mathrm{sgn}_2 = +1$. So the final from of NCA self-energy is

$$S_0(\tau) = -\sum_{ab}\left[d_b^\dagger R(\tau) d_a \Delta_{ba}(\tau) + d_b R(\tau) d_a^\dagger \Delta_{ab}(\beta-\tau)\right] . \tag{20}$$

As another exercise, we calculate sgn_1 in OCA.



(21)

So $\mathrm{sgn}_1 = -\mathrm{sgn}($  $) = -1$. Similarly, we get $\mathrm{sgn}_2 = \mathrm{sgn}_3 = -\mathrm{sgn}_4 = +1$. So the final from of $S_1(\tau)$ in OCA is

$$\begin{aligned} S_1(\tau) = -\sum_{abcd}\int_0^\tau d\tau_2 \int_0^{\tau_2} d\tau_1 \Big[& d_d^\dagger R(\tau-\tau_2) d_c^\dagger R(\tau_2-\tau_1) d_b R(\tau_1) d_a \Delta_{db}(\tau-\tau_1)\Delta_{ca}(\tau_2) \\ & + d_d R(\tau-\tau_2) d_c^\dagger R(\tau_2-\tau_1) d_b^\dagger R(\tau_1) d_a \Delta_{bd}(\beta-\tau+\tau_1)\Delta_{ca}(\tau_2) \\ & + d_d^\dagger R(\tau-\tau_2) d_c R(\tau_2-\tau_1) d_b R(\tau_1) d_a^\dagger \Delta_{db}(\tau-\tau_1)\Delta_{ac}(\beta-\tau_2) \\ & + d_d R(\tau-\tau_2) d_c R(\tau_2-\tau_1) d_b^\dagger R(\tau_1) d_a^\dagger \Delta_{bd}(\beta-\tau+\tau_1)\Delta_{ac}(\beta-\tau_2)\Big] . \end{aligned} \tag{22}$$

Determining the overall sign

Corollary 1 *$S(\tau)$ can be obtained by: i) replace R by R_0 in the self-energy expression; ii) write down its contribution, including* sgn*; iii)replace R_0 by R.*

Corollary 2 *For any configuration, its sign equals $(-1)^n$, where n equals the total number of hybridization lines flowing rightward plus the total number of crossover points. In other words,*

n = # of () + # of () .

- Improvement over NCA/OCA: $S(\tau) = \sum_{i=0}^{n} S_i(\tau)$
- $S_i(\tau)$ is obtained as
  - Generate all order-$(i+1)$ diagrams of $R(\tau)$
  - For each diagram D:
    - If D is not irreducible, move on to the next diagram
    - Write down its weight according to Fig. 2 and Corollary 2
    - Replace R_0 by R
  - Summation over the weights of all irreducible diagrams gives $S_i(\tau)$

Measurement

- Static quantity: $\langle O \rangle = \mathrm{Tr}[R(\beta)O]/\mathrm{Tr}[R(\beta)]$.
- Dynamical quantity, e.g. $G_{ab}(\tau_2 - \tau_1) = -\langle \mathcal{T}_\tau d_a(\tau_2) d_b^\dagger(\tau_1) \rangle$:

$$G_{ab}(\tau) = \begin{cases} -\mathrm{Tr}\left[R(\beta-\tau)d_a R(\tau) d_b^\dagger\right]/Z, & 0 < \tau < \beta; \\ \mathrm{Tr}\left[R(\beta+\tau)d_b^\dagger R(-\tau) d_a\right]/Z, & -\beta < \tau < 0. \end{cases}$$

$$\begin{aligned} G_{ab}(\tau) \;=\; & -\mathrm{Tr}[R(\beta-\tau)d_a R(\tau) d_b^\dagger]/Z \\ & -\sum_{cd}\int_0^\tau d\tau_1 \int_\tau^\beta d\tau_2 \mathrm{Tr}\left[R(\beta-\tau_2)d_c R(\tau_2-\tau) d_a R(\tau-\tau_1) d_d^\dagger R(\tau_1) d_b^\dagger \Delta_{dc}(\beta-\tau_2+\tau_1)\right]/Z \\ & -\sum_{cd}\int_0^\tau d\tau_1 \int_\tau^\beta d\tau_2 \mathrm{Tr}\left[R(\beta-\tau_2)d_c^\dagger R(\tau_2-\tau) d_a R(\tau-\tau_1) d_d R(\tau_1) d_b^\dagger \Delta_{cd}(\tau_2-\tau_1)\right]/Z, \end{aligned}$$

  - A rough calculation. Some non-/one- crossing terms are neglected
  - Solution: consider vertex correction

$$K(\tau_1,\tau_2,\tau_3) = \iint d\tau' d\tau'' R(\tau_3 - \tau'') K(\tau', \tau_2, \tau'') R(\tau' - \tau_1) \Delta(\tau'' - \tau')$$

(time-consuming and hard to implement in multi-orbital cases)

Codes

- XCA_solver: https://github.com/lzphy/XCA_solver
- Dependence:
  - ALPSCore (https://github.com/ALPSCore/ALPSCore)
  - Eigen (https://eigen.tuxfamily.org/)
- Run:
  - Prepare Input file
  - Change "xca.param" accordingly
  - ./Run.sh

Input & output file

$$H_{\mathrm{loc}} = \sum_{ij,\sigma} t_{ij,\sigma} d^{\dagger}_{i\sigma} d_{j\sigma} + \sum_{\substack{ijkl \\ \sigma\sigma'}} U_{ijkl,\sigma\sigma'} d^{\dagger}_{i\sigma} d^{\dagger}_{j\sigma'} d_{l\sigma'} d_{k\sigma}$$

- Currently support two kinds of input:
  - ed.input.h5
    - $t_{ij,\sigma} =$ _H0$[i, j, \sigma]$
    - $U_{ijkl,\sigma\sigma'} =$ _U$[\sigma, \sigma', i, j, k, l]$
    - _Epsk_i(nbath∗nno, ns); _Vk(nno,nbath∗nno,ns)
  - NCA_input_ntauxxx.h5
    - $t_{ij,\sigma} = \mathrm{H0}[\sigma, i, j]$
    - $U_{ijkl,\sigma\sigma'} = \mathrm{interactions}[i, j, k, l]$, regardless of σ, σ'
    - $\Delta_{i\sigma_i, j\sigma_j}(\tau) = \mathrm{Delta\_t}[\tau, \sigma_i, i, j]$ if $\sigma_i = \sigma_j$; 0 otherwise
- Plan: write a python script to get a uniform input format
- Output: $G_{i\sigma_i, j\sigma_j}(\tau) = \mathrm{G\_tau}[\tau, \sigma_i, \sigma_j, i, j]$

Parameters

- xca.param:

```
NORBITALS=2          %Number of orbitals; default: 1
NSPINS=2             %Number of spins; default: 2
BETA=700.0           %System temperature; default: 10.0
MAXITERATIONS=200    %Maximum iterations; default: 100
PRECISION=1.e-8      %Precision of XCA solver; default: 1.e-8
NSLICES=999          %Number of time slices;  default: 100
CUTOFF=true          %Whether do cutoffs or not; default: true
CUTOFF_VALUE=1.e-10 %A is considered as 0 if abs(A)<CUTOFFS; default: 1.e-8
SZSYMMETRY=true      %Whether perform Sz symmetry; default: true
NCA=true             %Perform NCA; default: true
OCA=false            %Perform OCA; default: false
DISCRETIZE=false     %Whether discretize hybridization; default: false
INPUT_FILE=EDbath/eg/NCA_inputs/NCA_input_ntau1000.h5 %Input file
OUTPUT_FILE=Output/eg/xca.sim.dis.eg1000.h5           %Output file
```

Fermionic matrix

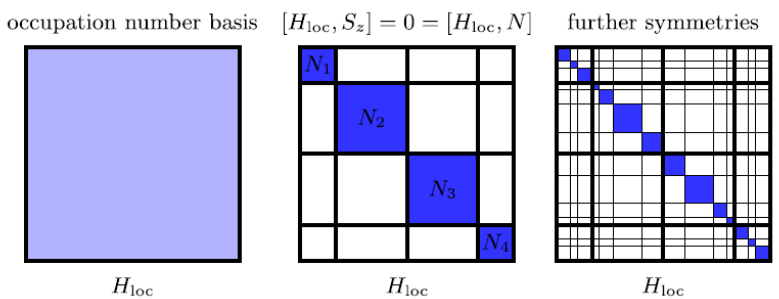
- Convention: $|\uparrow_0\downarrow_0\uparrow_1\downarrow_1\cdots\rangle = d^\dagger_{\uparrow_0} d^\dagger_{\downarrow_0} d^\dagger_{\uparrow_1} d^\dagger_{\downarrow_1} \cdots |0000\cdots\rangle$
- Labels: $|i_2\rangle$ is labeled as i_{10}
  - e.g., 0: $|0000\rangle$, 1: $|1000\rangle$, 2: $|0100\rangle$, 3: $|1100\rangle, \cdots$
- $\tilde{d}^{(\dagger)} = T' d^{(\dagger)} T$, where T is the transformation matrix
  - $\mathrm{O}(n^3)$ if calculating directly
  - $T_{ij} = \delta_{s(i),j} = \delta_{i,s^{-1}(j)}$ where $s(i)$ is the new location of i-th basis
  - $\tilde{d}^{(\dagger)}_{s(i)s(j)} = d^{(\dagger)}_{ij}$, $\mathrm{O}(n^2)$ costs, already fast enough

Block-diagonalizing

- Symmetries are dependent on the exact form of H_{loc}
- Usually the total particle number and the total spin z-component are conserved: $[H_{\text{loc}}, N_{\text{tot}}] = 0 = [H_{\text{loc}}, S^z_{\text{tot}}]$
- Switch the order of basis so that states with same $(N_{\text{tot}}, S^z_{\text{tot}})$ are grouped together
- the computation of matrix products scales as $O(n^3_{\text{max block}})$ instead of $O(n^3_{\text{loc Ham}})$
- Data structure:

```
typedef Eigen::Matrix<double, Eigen::Dynamic, Eigen::Dynamic>  mat;
typedef std::vector< mat > mat_vec;
```

- Operator overloading: $A + B$, $A * B$, $\lambda * A$, $A + \lambda$, ...



Rev. Mod. Phys 83, 349 (2011)

References

- E. Gull, A. J. Millis, A. I. Lichtenstein, A. N. Rubtsov, M. Troyer, and P. Werner, Rev. Mod. Phys 83, 349 (2011).
- A. Rüegg, E. Gull, G. A. Fiete, and A. J. Millis, Phys. Rev. B 87, 075124 (2013).

Thanks!