$$\hat{H} = \sum_{i,j} h_{ij}\hat{E}_{IJ} + \hat{U} + \sum_{i} v_i^{'}\hat{n}_i = \sum_{i,j} h_{ij}\hat{E}_{IJ} + \hat{U} - \sum_{i} \mu_i\hat{n}_i = \sum_{i\neq j} h_{ij}\hat{E}_{IJ} + \hat{U} + \sum_{i} v_i^{ext}\hat{n}_i \quad (1)$$

Note that we split one-electron operator on 2 contributions: where $i \neq j$ and where $i = j$. This is also used in python code (`Molecule.t` and `Molecule.v_ext` respectively)

x	**Description**
`Molecule.Ne`	Number of electrons in the lattice
`Molecule.Ns`	Number of atoms in the lattice
`Molecule.u`	1D array that represents on site repulsion: $x_i = U_{i,i,i,i} = \langle ii \vert ii \rangle$
`Molecule.t`	2D array that represents one-electron Hamiltonian where $i! = j$: $x_{ij} = \begin{cases} 0 & \text{i and j are not connected} \\ t_{ij} & \text{otherwise} \end{cases}$ it is analog to the $\langle \chi_i \vert \hat{h} \vert \chi_j \rangle$
`Molecule.v_ext`	1D array; sum of $h_{ii} - \mu_i = h_{ii} + v_i^{'} = v_i^{ext}$. It is predefined by our system and it doesn't change during our calculations
`Molecule.equiv_atom_groups`	dictionary of different sites that are equivalent (and because of that recalculation would be just repetitive. It is a dict with keys $[0, N_{sites}]$ and for each entry value is tuple of all sites (according to Hamiltonian matrix) e.g., `{0: (0, 3), 1: (1, 2, 4, 5)}`
`Molecule.embedded_mol`	It is a representation of a cluster in `class_Quant_NBody.QuantNBody` class which is class created by me on top of Saad's library `Quant_NBody`

1 Kohn-Sham

x	**Description**
`Molecule.mu_s`	1D vector that ensures that KS system will have correct density with only 1-electron interactions + additional potential called Hartree-Exchange correlation potential: $x_i = \mu^s = \mu^{ext} - v^{Hxc}$
`Molecule.h_ks`	2D vector. It combines kinetic energy part of one electron Hamiltonian ($i \neq j$) and Kohn-Sham potential: $x_{ij} : h^{KS}_{ij} = h_{i \neq j} - \mu^s_i \delta_{ij}$
`Molecule.wf_ks`	2D array that represents all wave functions of non-interacting system ordered by energy. Coefficients for the j-th wave function on the site i on the lattice is represented by x_{ij}. This means that wave function is presented by vector$\vec{v}_k = x_{ik}$
`Molecule.epsilon_s`	1D vector of Kohn-Sham energies
`Molecule.y_a`	2D array of 1RDM. y due to similarity to γ and a to remind that this is only density per spin (α). It is generated form KS equation.
`Molecule.n_ks`	1D vector that is made from diagonal elements of γ_α. It represents occupation numbers for each site that are important for CASCI part of algorithm.
`Molecule.mu_hxc`	1D vector that is the connection between KS and CASCI. In KS part it is represent Hxc chemical potential ($x_i = \mu^{Hxc}_i = -v^{Hxc}_i$) and in CASCI it represents impurity potential: $x_i = \tilde{\mu}^{imp}_i$

2 CASCI

Here are variables that are nested into CASCI method

x	**Description**
`site_group`	Key of a dictionary that goes over all equivalent atom groups.
`site_id`	index of 1 (first) site in the group where we actually calculate CASCI
`y_a_correct_imp`	2D matrix that is a copy of γ_α with permutated indices in such way that our our impurity gets index 0 (this is important since Householder doesn't transform site with index 0). This is done by simply switching columns (`site_id`, 0) and then also same rows.
`v`	1D Householder vector that creates bath and environment orbitals. $x_i = v_i = \begin{cases} 0 & i = 0 \\ \frac{\gamma_{10}-\xi}{\sqrt{2\xi(\xi-\gamma_{10})}} & i = 1 \\ \frac{\gamma_{i0}}{\sqrt{2\xi(\xi-\gamma_{10})}} & i \geq 2 \end{cases}$ where $\xi = -\mathrm{sign}(\gamma_{10})\sqrt{\sum_{j>0}\gamma_{j0}^2}$
`P`	2D Transformation matrix that transforms 1RDM: $x_{ij} = P_{ij} = \delta_{ij} - 2v_i v_j$
`y_a_tilde`	This is also 1RDM but in Householder transformed space: $X = P \cdot \gamma_\alpha \cdot P$. Note that this is not Molecule class object because now when we have embedding different impurities will have different Householder transformations.
`h_tilde`	In save way we transformed 1RDM we also transform one-electron Hamiltonian. This is then $P \cdot (t + v^{ext}) \cdot P$
`h_tilde_dimer`	just cropped transformed one-electron Hamiltonian that will be input for CASCI in `Quant_NBody`
`u_0_dimer`	4D tensor that represents $\langle ij\|kl\rangle$. In our case this is for dimer with all values 0 ("Non-interacting bath") but impurity ($x_{0000} = U$)

`mu_imp`

This is number that corresponds to chemical potential of the impurity. At each cycle it is read from Hxc potential at site that corresponds to `site_id` and in the end written there.

`optimized_v_imp_obj`

object that is returned by `scipy.optimize.minimize` optimizer. It has information about how close we were able to come to KS result and optimized impurity chemical potential.

`cost_function_CASCI`

function that calculates square difference of occupation in KS system and CASCI at given impurity potential.