

I. DETAILS OF NCA EVALUATION

The NCA diagrams correspond to the following analytic expression for the pseudoparticle self-energy

$$\begin{aligned} \Sigma_{mn}(i\omega) = & \sum_{\alpha\beta, m'n'} \left[F_{mm'}^{\beta}(F^{\alpha\dagger})_{n'n} \int d\xi f(\xi) A_{\alpha\beta}(\xi) G_{m'n'}(i\omega + \xi) \right. \\ & \left. + (F^{\beta\dagger})_{mm'} F_{n'n}^{\alpha} \int d\xi f(-\xi) A_{\beta\alpha}(\xi) G_{m'n'}(i\omega - \xi) \right]. \end{aligned} \quad (1)$$

where $F_{nm}^{\alpha} = \langle n|d_{\alpha}|m\rangle$ are matrix elements of destruction operator of a local electron, G_{nm} are pseudoparticle Green's functions and Σ_{mn} are corresponding self-energies, i.e.,

$$(G^{-1})_{mn} = \omega - E_{mn} - \lambda - \Sigma_{mn} \quad (2)$$

and E_{mn} is the energy of the local state in the atomic limit. The bath spectral functions $A_{\alpha\beta}$ is related to the hybridization function Δ by the following identity $A_{\alpha\beta} = -1/(2\pi i)[\Delta_{\alpha\beta}(\omega + i0^+) - \Delta_{\alpha\beta}(\omega - i0^+)]$.

To simplify the equations, we will write them in terms of spectral quantities instead of Green's functions. Let us define

$$\overline{G}_{mn}(\omega) = \frac{1}{2i}(G_{mn}(\omega + i0^+) - G_{mn}(\omega - i0^+)) \quad (3)$$

$$\overline{\Sigma}_{mn}(\omega) = \frac{1}{2i}(\Sigma_{mn}(\omega + i0^+) - \Sigma_{mn}(\omega - i0^+)). \quad (4)$$

The quantities with hat are obviously hermitian $\overline{\Sigma}^{\dagger} = \overline{\Sigma}$ and the spectral function of the Green's function is proportional to the spectral density ρ , i.e., $\rho = -\overline{G}/\pi$.

With the above definitions, the NCA equations simplify to

$$\overline{\Sigma}_{mn}(\omega) = \sum_{\alpha\beta, m'n'} \int d\xi f(\xi) \overline{G}_{m'n'}(\omega + \xi) \left[F_{mm'}^{\beta}(F^{\alpha\dagger})_{n'n} A_{\alpha\beta}(\xi) + (F^{\beta\dagger})_{mm'} F_{n'n}^{\alpha} A_{\beta\alpha}(-\xi) \right] \quad (5)$$

The local electron Green's function within NCA approximation is given by

$$G_{\alpha\beta}(i\omega) = - \sum_{mnm'n'} F_{n'n}^{\alpha}(F^{\beta\dagger})_{mm'} \frac{1}{Q} \int \frac{d\xi}{\pi} \exp(-\beta\xi) \left[\overline{G}_{m'n'}(\xi) G_{nm}(\xi + i\omega) - G_{m'n'}(\xi - i\omega) \overline{G}_{nm}(\xi) \right], \quad (6)$$

where the grand canonical expectation value of the charge Q (or equivalently the partition function of the impurity) is given by

$$Q = - \int \frac{d\xi}{\pi} \exp(-\beta\xi) \sum_m \overline{G}_{mm}(\xi). \quad (7)$$

We will again write the equation for the spectral function rather than the Green's function. The local electron spectral function ρ becomes

$$\rho_{\alpha\beta}(\omega) = \sum_{mnm'n'} F_{n'n}^{\alpha}(F^{\beta\dagger})_{mm'} \frac{1}{Q} \int \frac{d\xi}{\pi^2} \exp(-\beta\xi) [1 + e^{-\beta\omega}] \overline{G}_{m'n'}(\xi) \overline{G}_{nm}(\xi + \omega) \quad (8)$$

The system of equations Eq. (2), (5), (8) and (7) is thus closed.

Let me pause here for a moment and show that the equations are invariant with respect to the shift of frequency in the pseudo particle quantities due to local gauge symmetry, therefore λ that appears in the definition of the pseudo Green's functions can be an arbitrary number. In numerical evaluation, we use this to our advantage and choose zero frequency at the point where pseudo particle spectral functions diverge. First, we want to show that shift of frequency arguments of all pseudo-particle quantities by $\delta\lambda$ does not affect the equations provided that λ also changes to $\lambda \rightarrow \lambda + \delta\lambda$

$$[G^{-1}(\omega + \delta\lambda)]_{mn} = \omega - E_{mn} - \lambda - \Sigma_{mn}(\omega + \delta\lambda) \quad (9)$$

$$\overline{\Sigma}_{mn}(\omega + \delta\lambda) = \sum_{\alpha\beta, m'n'} \int d\xi f(\xi) \overline{G}_{m'n'}(\omega + \xi + \delta\lambda) \left[F_{mm'}^{\beta}(F^{\alpha\dagger})_{n'n} A_{\alpha\beta}(\xi) + (F^{\beta\dagger})_{mm'} F_{n'n}^{\alpha} A_{\beta\alpha}(-\xi) \right] \quad (10)$$

$$\rho_{\alpha\beta}(\omega) = \frac{\sum_{mnm'n'} F_{n'n}^{\alpha}(F^{\beta\dagger})_{mm'} \int \frac{d\xi}{\pi^2} \exp(-\beta\xi) [1 + e^{-\beta\omega}] \overline{G}_{m'n'}(\xi + \delta\lambda) \overline{G}_{nm}(\xi + \omega + \delta\lambda)}{\int (-\frac{d\xi}{\pi}) \exp(-\beta\xi) \sum_m \overline{G}_{mm}(\xi + \delta\lambda)}. \quad (11)$$

The above equations are obviously unchanged. Notice that the physical quantities like the local spectral function and the baths spectral function are not affected by the shift. The auxiliary charge Q , however, changes $Q \rightarrow Q \exp(-\beta\delta\lambda)$.

In numerical evaluation, we can either fix λ and calculate Q , or alternatively, fix Q and calculate λ at each iteration. The goal is to resolve the structure of the pseudo-particle functions and therefore it is best to choose the zero frequency at the point where the pseudo-quantities have a threshold (they diverge at zero temperature). In the program, user can fix Q to an arbitrary number and the corresponding λ , which is just the impurity free energy, is calculated. This guarantees that in zero temperature limit the threshold point is at zero frequency. For small number of local states, like for the one band model, the choice $Q = 1$ is best. When the number of bands grows, Q should grow as well. A good choice is $Q \approx N_{all}$ where N_{all} is the number of all local states.

The above NCA equations are still numerically intractable because of exponential factors appearing in Eq. (7) and (8). As we mentioned, the pseudo-quantities have a threshold, i.e., they exponentially vanish below a certain frequency (which is zero frequency if Q is a fixed constant). However, the exponential tails obviously enter the equation Eq. (7) and need to be calculated. The problem can be circumvented by defining new quantities that coincide with the original quantities above the threshold, but are nonzero also below the threshold and carry the information needed below the threshold. We will use the tilde sign for those new quantities

$$\tilde{G}(\omega) = \overline{G}(\omega)/f(-\omega) \quad (12)$$

$$\tilde{\Sigma}(\omega) = \overline{\Sigma}(\omega)/f(-\omega). \quad (13)$$

Substituting the tilde quantities in the Eq. (5), (7) and (8), we obtain

$$\tilde{\Sigma}_{mn}(\omega) = \sum_{\alpha\beta, m'n'} \int d\xi \frac{f(\xi)f(-\xi-\omega)}{f(-\omega)} \tilde{G}_{m'n'}(\omega+\xi) \left[F_{mm'}^{\beta} (F^{\alpha\dagger})_{n'n} A_{\alpha\beta}(\xi) + (F^{\beta\dagger})_{mm'} F_{n'n}^{\alpha} A_{\beta\alpha}(-\xi) \right] \quad (14)$$

$$\rho_{\alpha\beta}(\omega) = \sum_{mm'n'} F_{n'n}^{\alpha} (F^{\beta\dagger})_{mm'} \frac{1}{Q} \int \frac{d\xi}{\pi^2} \frac{f(\xi)f(-\xi-\omega)}{f(-\omega)} \tilde{G}_{m'n'}(\xi) \tilde{G}_{nm}(\xi+\omega) \quad (15)$$

$$Q = - \int \frac{d\xi}{\pi} f(\xi) \sum_m \tilde{G}_{mm}(\xi). \quad (16)$$

The system of NCA equations Eq. (14), (15) and (16) is numerically more tractable and free of diverging exponential factors. Namely, the combination of Fermi functions $f(\xi)f(-\xi-\omega)/f(-\omega)$ is always less than unity and can be very accurately calculated by realizing that

$$\frac{f(\xi)f(-\xi-\omega)}{f(-\omega)} = \frac{f(-\xi)f(\xi+\omega)}{f(\omega)} = f(\xi)f(-\xi-\omega) + f(-\xi)f(\xi+\omega),$$

is just a window function.

A. A note on calculating convolutions

Most of the Green's functions and specially Fermi function at low temperature are very steep or peaked at certain points (usually around zero). The use of equidistant mesh is therefore impractical and time consuming, rather a non-equidistant mesh is desired. We usually use logarithmic mesh at low frequency and tanges mesh at high frequencies to resolve almost any function with less than 300 points. The prices one needs to pay using nonequidistant mesh is that a convolution like

$$C(z) = \int g(x)f(x-z)dx \quad (17)$$

is much harder to calculate since it is not resolved on the first mesh neither on the second. Suppose that $g(x)$ is defined on a certain mesh $\{x_i\}$ where it is resolved $g(x_i) \equiv g_i$ and function $f(y)$ is defined on another mesh $\{y_i\}$, i.e., $f(y_i) \equiv f_i$, the convolution can be safely calculated on the union of both meshes $\{x_i, y_j + z\}$. One of the meshes should be shifted for z , thus for each outside frequency a different union of the two meshes should be formed and only then the convolution can be safely evaluated. This is very time consuming and seldom done in practice.

We have two strategies to circumvent the problem. If one knows that both functions are peaked around certain point, lets suppose zero frequency, we can form a two-dimensional mesh (somewhere in the outside loop such that it is heavily reused) in which the first mesh $\{x_i\}$ is superposed with couple of points around zero frequency from mesh $\{y_j\}$.

However, when a certain function needs to be convolved with many other functions (typical case is NCA with many nonequivalent pseudo-particles), one can use a trick. With the help of two additional precalculated functions, the integral and first moment

$$F_1(\epsilon_i) = \int_{-\infty}^{\epsilon_i} f(u) du \quad (18)$$

$$F_2(\epsilon_i) = \int_{-\infty}^{\epsilon_i} u f(u) du \quad (19)$$

$$(20)$$

one can calculate the convolution without building a new inside mesh. Lets use the mesh $\{x_i\}$ which resolves function g . Then, in the spirit of trapezoid rule, we can linearly interpolate g between the points

$$C(z) = \sum_i \int_{x_i}^{x_{i+1}} \left[g_i + \frac{g_{i+1} - g_i}{x_{i+1} - x_i} (x - x_i) \right] f(x - z) dx. \quad (21)$$

This integral can be expressed by the above defined functions. To show that, let us rewrite the convolution and expressed it by the new function $\langle f \rangle_i$ which is defined on the same mesh as g and with which the convolution is a simple scalar product

$$C(z) = \sum_i g_i \left[\int_{x_i - z}^{x_{i+1} - z} \frac{x_{i+1} - z - u}{x_{i+1} - x_i} f(u) du + \int_{x_{i-1} - z}^{x_i - z} \frac{z + u - x_{i-1}}{x_i - x_{i-1}} f(u) du \right] \equiv \sum_i g_i \langle f \rangle_i dh_i. \quad (22)$$

Thus $\langle f \rangle_i$ is

$$\langle f \rangle_i = \frac{2}{x_{i+1} - x_{i-1}} \left[\int_{x_i - z}^{x_{i+1} - z} \frac{x_{i+1} - z - u}{x_{i+1} - x_i} f(u) du + \int_{x_{i-1} - z}^{x_i - z} \frac{z + u - x_{i-1}}{x_i - x_{i-1}} f(u) du \right] \quad (23)$$

$$= \frac{2}{x_{i+1} - x_{i-1}} \left[\frac{(x_{i+1} - z)[F_1(x_{i+1} - z) - F_1(x_i - z)] - F_2(x_{i+1} - z) + F_2(x_i - z)}{x_{i+1} - x_i} - \right. \quad (24)$$

$$\left. - \frac{(x_{i-1} - z)[F_1(x_i - z) - F_1(x_{i-1} - z)] - F_2(x_i - z) + F_2(x_{i-1} - z)}{x_i - x_{i-1}} \right] \quad (25)$$

This method is specially suitable if function f needs to be convolved with many different functions that live on the same mesh $\{x_i\}$ because the function $\langle f \rangle_i$ can be calculated ones and reused many times. This is the case of NCA with many nonequivalent local states. The method is also crucial for implementing higher-order vertex correction to NCA. The implementation can be found in the header file *average.h*.

II. HOW TO GENERATE THE NCA INPUT FILE

In general case, the matrix $F_{nm}^\alpha = \langle n | d_\alpha | m \rangle$ is a full matrix between the Hilbert subspaces of n and $n + 1$ electrons. The best choice of base for solving NCA equations is the local eigenbase, where the impurity energies $E_{mn} = \delta_{mn} E_m$ are diagonal. In this case, one needs to write the F matrix in local eigenbase as well. First, one can write F in direct base, where it is very simple and mostly zero. However, one needs to pay some attention to the fermionic sign that appears due to the fermionic nature of the electron operator d_α . Then, one can transform F to the local eigenbase.

The matrix F is greatly simplified when the hybridization $\Delta_{\alpha\beta}$ is diagonal and Coulomb repulsion is of the form $U_{\alpha\beta} n_\alpha n_\beta$ since in this case, the direct base is also the local eigenbase. For this case, the program called **generate** generates the necessary base and calculates the F matrix, which is necessary to run the NCA code.

Let me give a simple example of what the program **generate** does. In the two band case, with the crystal field

splitting, the local states are

| # | State | n | Deg | Energy |
|---|-------|-----|-----|--------------------|
| 0 | 00 | 0 | 1 | 0 |
| 1 | 01 | 1 | 2 | E_2 |
| 2 | 10 | 1 | 2 | E_1 |
| 3 | 02 | 2 | 1 | $2E_2 + U$ |
| 4 | 11 | 2 | 4 | $E_1 + E_2 + U$ |
| 5 | 20 | 2 | 1 | $2E_1 + U$ |
| 6 | 12 | 3 | 2 | $E_1 + 2E_2 + 3U$ |
| 7 | 21 | 3 | 2 | $2E_1 + E_2 + 3U$ |
| 8 | 22 | 4 | 1 | $2E_1 + 2E_2 + 6U$ |

(26)

where n is the total number of electrons on the impurity, Deg is degeneracy of the state and a state 21 represents an atomic state with 2 electrons in the first band and 1 electron in the second. To generate this output, one needs to give the following data to the program

- number of baths : 2
- degeneracy of the first bath : 2
- degeneracy of the second bath: 2

In the program **generate**, user can then take only some part of the atomic states (to speed up the calculation) or all of the states. For example, if one is interested in the $\langle n \rangle \sim 1$ and U is large, there is no point of keeping the state with 4 electrons on the local level since the probability for the state number 8 is very small. The program **generate** then calculates the F matrix and writes it to a file which needs to be given to the NCA program. If one is interested in the SU(N) problem, it is best to choose

- number of baths : 1
- degeneracy of the bath : N .

III. HOW TO RUN THE NCA PROGRAM

The program can be compiled on linux PC by typing `"make PLATFORM=PC RELEASE=1"`. The second option produces the optimized code.

A short help can always be obtained by executing the program and giving no arguments. Thus typing `nca` gives the following output

```

Sig = sig.inp      # The name of the input Self-energies
Ac = Ac.inp       # The name of the input bath function
cix = cix.dat      # The name of the input file containing information about bands and their degeneracy
out = .           # The name of the output directory
gloc = gloc.out    # The name of the output Green's function
sig = sig.out      # The name of the output Self-energy
Ed = -2           # Energy level
U = 4             # Coulomb repulsion
T = 0.2           # Temperature
Q = 1             # Default average Q in grand-canonical ansamble
alpha = 0.5       # The fraction of the new self energy to be used in the next iteration
max_diff = 1e-06  # Criterium to finish the procedure
max_steps = 300   # Criterium to finish the procedure
StartLambda = -1 # Where to start searching for the lambda0
EndLambda = 1     # Where to stop searching for the lambda0
dLambda = 0.1    # Step in searching for the lambda
followPeak = -3   # Wheather to determin zero frequency from the diverging pseudo-particle
                  (-2: lamdba0==StartLambda, -1: Q==Q0, 0: follow b, 1: follow f, 2: foolow a)
prt = 1           # Weather to print intermediate results
MissDopSt = -100 # Missing doping due to projection and finite mesh starting at MissDopSt

```

All control parameters are printed with there default values. After the symbol # a short description of each parameter is also given. Below we will give some more description to each parameter but let us run an example program first.

The necessary files can be found in the subdirectory **PC/work** and **PC/work/start**. The example is for the f-electrons with around 6 electrons on the impurity. The crystal field splits the 7 band problem into 3+3+1 band problem (To generate the input file one thus needs to use 3 bands and degeneracy 6,6,2). All the impurity states with 4,5,6 and 7 electrons on the impurity are keep which results in a 64 nonequivalent pseudoparticle (atomic) states. The necessary coefficients F 's to run the program are already prepared in the file **PC/work/start/cix4567.cix**.

To run the program, one also needs a bath spectral function (it is in **PC/work/start/Ac.12**) and starting guess for the pseudo self-energies (can be found in **PC/work/start/Sigma.000**). The input bath spectral function contain four columns, frequency and three nonequivalent bath spectral functions. The pseudo self-energy files contains 65 columns, in addition to frequency, a guess self-energy for all 64 pseudo particles.

An example of how to run the program can be found in the file **history.nca** in a subdirectory **PC/work** and reads `../nca Sig=start/Sigma.000 Ac=start/Ac.12 cix=start/cix4567.cix U=4 T=0.1 "Ed={-20.67,-20.58,-20.64}"`

Each time the program is executed, the command line is saved (appended) to the file **history.nca** for easier restart of a job at a later time.

In the example, the self-consistency is reached after about 30 iterations (the difference between the steps below 10^{-6}) and the following results are ready to be plotted

- Impurity Green's function is in the file gloc.out (the filename can be changed in the command line)
- The physical self-energy is in the file sig.out (the filename can also be chnaged if desired)
- the latest pseudo self-energies are in the file Sigma.000
- the latest pseudo spectral functions are in the file Spec.000
- pseudo self energies for each iteration are saved in a file Sigma.xxx
- pseudo spectral functions for each iteration are saved in a file Spec.xxx

The spectral function obtained in this example is plotted in figure 1. At each iteration, the following information is written in the standard output

- The value of λ discussed in the chapter I. ($\lambda=-64.8101064793649$)
- The value of Q that was used ($Q=9438$ which is total number of states here)
- The occupancy of each atomic state (the sum of all numbers is unity)
- Total impurity occupancy (nd: 5.73398078352997)

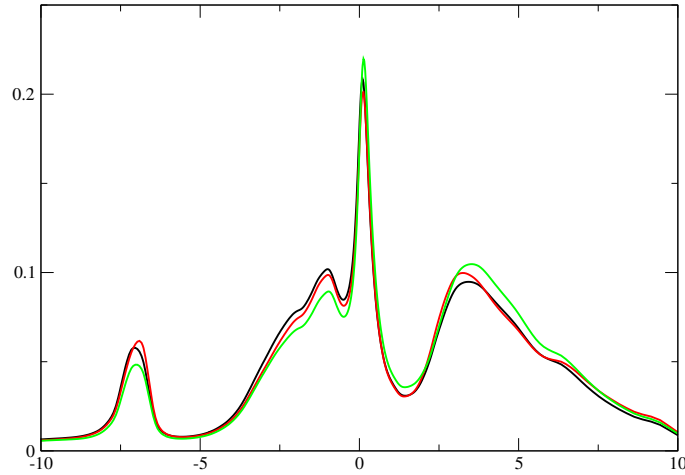


FIG. 1: The spectral function in the example.

- Norm of every pseudo spectral function (should be close to one if mesh resolves the structure and λ was chosen correctly).
- Impurity occupancy per each band (0: 2.5582 1: 2.4337 2: 0.7420)
- The difference between last two steps (Difference between steps: 6.768e-07).

Finally, let me give some more details of every parameter of the program

- Sig: The filename of the input guess for the pseudo self-energies
- Ac: The filename of the input bath spectral function
- cix: The input file that was generated by the program **generate** and discussed in chapter II. Note that the number of columns in the Sig file should match the number of pseudoparticle states from cix-file and the number of columns in Ac file should match the number of baths in the cix-file.
- out: The name of the output directory. Default is the current directory.
- gloc : Filename of the output local Green's function
- sig : Filename of the output local self-energy
- Ed : Impurity energies. The number of impurity energies should match the number of baths (one impurity energy for each nonequivalent bath). If not, the program assumes that the rest of the energies are equal to the last specified energy. They have to be given in curly brackets and in the shell, additional quotation marks should be used, for example " $Ed=\{-3,-3,-3\}$ ".
- U : The Coulomb repulsion
- T : Temperature
- Q : The desired charge Q explained in chapter I. This is used only when **followPeak** is set to -1.
- alpha : The mixing parameter for the pseudo self-energy
- max_diff : The difference between successive self-energies has to be smaller than this number to stop iterating.
- max_steps : The number of iterations is always less than this number (regardless of max_diff) to avoid infinite loops.
- StartLambda : If Q is fixed number, one needs to find the corresponding λ as explained in chapter I. For that, one needs to solve a nonlinear equation and the solution is always looked for in the interval StartLambda and EndLambda. StartLambda should be enough negative to catch the solution. If the solution is not found, tray to decrease this number (more negative) and if the solution is found, keep the new value. However, very small StartLambda can result in worse performance.

- EndLambda: Explained above
- dLambda : If one looks for the λ allover the interval [StartLambda,EndLambda], the solution is very unlikely to be cached. The reason is that when a guess λ is larger than the λ that corresponds to the solution, the spectral functions are more diverging and therefore the spectral functions are much worse resolved. The strategy is to start looking for solution at most negative lambda (StartLambda) and slowly move up until the solution is bracket and then use the Newton procedure. The small step when bracketing the solution is dLambda. Can be much decreased if the solution is hard to catch (very close to the Mott transition).
- followPeak: There are four strategies implemented to fix zero frequency of the pseudoparticle quantities. When followPeak is -3, the Q is fixed at the total number of all local states and λ is then looked for. If the followPeak is -2, λ is chosen to be equal to *StartLambda* and Q is calculated from the equation (16). If followPeak is -1, Q is fixed to the input Q (specified in the command line, the parameter Q above) and the corresponding λ is looked for. If follow peak is a non-negative number a , the zero frequency (and λ) is chosen such that the maximum of the pseudoparticle spectral function with index a is at zero frequency.
- prt : If prt is 1, pseudo self-energy and spectral functions are printed at each iteration. It can also be set to 0.
- MissDopSt: Sometimes the mesh on which the electron spectral function is calculated, does not catch all the spectral weight at negative frequencies, or alternatively, there is some weight of spectral function lost due to projection (if we take occupancies $n-1, n, n+1, \dots$ and $n-2$ was still having some nonzero probability). In this case, we want to calculate how much weight is missing when integrating spectral function on the output mesh. This is very important in the real material calculation, because the missing weight should be added to the total number of electrons below the Fermi level, to get the right occupancy and right chemical potential. The parameter should be set to the point below which the weight is considered to be lost.