Hubbard I approximation

To be specific, we concentrate on the Anderson impurity Hamiltonian

$$H = \varepsilon_f \sum_{\alpha=1}^{N} f_\alpha^+ f_\alpha + \frac{1}{2} U \sum_{\alpha \neq \beta}^{N} n_\alpha^f n_\beta^f + \sum_{\mathbf{k}\alpha} E_{\mathbf{k}\alpha} c_{\mathbf{k}\alpha}^+ c_{\mathbf{k}\alpha} + \sum_{\mathbf{k}\alpha} [V_\alpha^*(\mathbf{k}) f_\alpha^+ c_{\mathbf{k}\alpha} + V_\alpha(\mathbf{k}) c_{\mathbf{k}\alpha}^+ f_\alpha], \tag{1}$$

describing the interaction of the impurity level ε_f with bands of conduction electrons $E_{\mathbf{k}\alpha}$ via hybridization $V_\alpha(\mathbf{k})$. U is the Coulomb repulsion between different orbitals in the f –band.

Now we turn to the Hubbard I approximation [1] which is closely related to the moments expansion method [2]. Consider many–body atomic states $\left|\Phi_\kappa^{(n)}\right\rangle$ which in $SU(N)$ are all degenerate with index κ numerating these states for a given number of electrons n. The impurity Green function is defined as the average

$$G_f(\tau) = -\left\langle T_\tau f_\alpha(\tau) f_\beta^+(0) \right\rangle. \tag{2}$$

and becomes diagonal with all equal elements in $SU(N)$. It is convenient to introduce the Hubbard operators

$$\hat{X}_{\kappa\kappa'}^{nn'} = \left|\Phi_\kappa^{(n)}\right\rangle \left\langle \Phi_{\kappa'}^{(n')}\right| \tag{3}$$

and represent the one–electron creation and destruction operators as follows

$$f_\alpha = \sum_n \sum_{\kappa\kappa'} \left\langle \Phi_\kappa^{(n)} \mid f_\alpha \mid \Phi_{\kappa'}^{(n+1)} \right\rangle \hat{X}_{\kappa\kappa'}^{nn+1}, \tag{4}$$

$$f_\alpha^+ = \sum_n \sum_{\kappa\kappa'} \left\langle \Phi_\kappa^{(n+1)} \mid f_\alpha^+ \mid \Phi_{\kappa'}^{(n)} \right\rangle \hat{X}_{\kappa\kappa'}^{n+1n}. \tag{5}$$

The impurity Green function (2) is given by

$$G_f(\tau) = \sum_{nm} G_{nm}(\tau), \tag{6}$$

where the matrix $G_{nm}(\tau)$ is defined as

$$G_{nm}(\tau) = -\sum_{\kappa_1\kappa_2\kappa_3\kappa_4} \left\langle \Phi_{\kappa_1}^{(n)} \mid f_\alpha \mid \Phi_{\kappa_2}^{(n+1)} \right\rangle \left\langle T_t \hat{X}_{\kappa_1\kappa_2}^{nn+1}(\tau) \hat{X}_{\kappa_3\kappa_4}^{m+1m}(0) \right\rangle \left\langle \Phi_{\kappa_3}^{(m+1)} \mid f_\alpha^+ \mid \Phi_{\kappa_4}^{(m)} \right\rangle. \tag{7}$$

Establishing the equations for $G_{nm}(\tau)$ can be performed using the method of equations of motion for the X operators. Performing their decoupling due to Hubbard [1,3], carrying out the Fourier transformation and analytical continuation to the real frequency axis, and summing over n and m after (6) we arrive to the main result

$$G_f^{-1}(\omega) = G_{at}^{-1}(\omega) - \Delta(\omega), \tag{8}$$

where hybridization $\Delta(\omega) = \sum_{\mathbf{k}} V(\mathbf{k})/(\omega - E_{\mathbf{k}})$ satisfies the DMFT self–consistency condition of the Hubbard model on a Bethe lattice

$$\Delta(\omega) = \left(\frac{W}{4}\right)^2 G_f(\omega), \tag{9}$$

with W being the bandwidth.

The $G_{at}(\omega)$ can be viewed in the matrix form (6) with the following definition of a diagonal atomic Green function

$$G_{nm}^{at}(\omega) = \delta_{nm} \frac{C_n^{N-1}(X_n + X_{n+1})}{\omega + \mu - E_{n+1} + E_n}. \tag{10}$$

with $E_n = \varepsilon_f n + Un(n-1)/2$ being the total energies of the atom with n electrons in $SU(N)$. The coefficients X_n are the probabilities to find atom with n electrons while combinatorial factor $C_n^{N-1} = \frac{(N-1)!}{n!(N-n-1)!}$ arrives due to equivalence of all states with n electrons in $SU(N)$.

The coefficients X_n are normalized to unity, $\sum_{n=0}^{N} C_n^N X_n = \sum_{n=0}^{N-1} C_n^{N-1}(X_n + X_{n+1}) = 1,$ and are expressed via diagonal elements of $G_{nm}(i\omega)$ as follows:

$$X_n = -T\sum_{i\omega} G_{nn}(i\omega)e^{-i\omega 0^+}/C_n^{N-1}. \tag{11}$$

Their determination in principle assumes solving a non–linear set of equations while determining $G_f(\omega)$. The mean number of electrons can be measured as follows: $\bar{n} = \sum_{n=0}^{N} nC_n^N X_n$ or as follows $\bar{n} = TN\sum_{i\omega} G_f(i\omega)e^{i\omega 0^+}$. The numbers X_n can be also used to find the averages $\langle nn \rangle$ $\langle nnn \rangle$: density–density correlation function $\langle nn \rangle$ for local states with n electrons is proportional to the number of pairs formed by n particles C_2^n/C_2^N. Since the probability for n electrons to be occupied is given by: $P_n = X_n C_n^N$, the physical density–density correlator can be deduced from: $\langle nn \rangle = \sum_n C_2^n/C_2^N P_n$. Similarly, the triple occupancy can be calculated from $\langle nnn \rangle = \sum_n C_3^n/C_3^N P_n$.

If we neglect by the hybridization $\Delta(\omega)$ in Eq. (8), the probabilities X_n become simply statistical weights:

$$X_n = \frac{e^{-(E_n - \mu n)/T}}{\sum_{m=0}^{N} C_m^N e^{-(E_m - \mu m)/T}}. \tag{12}$$

We thus see that in principle there are several different ways to determine the coefficients X_n, either via self–consistent determination (11), or using statistical formula (12). To determine the best procedure let us first consider limits of large and small U's. When $\Delta(\omega) \equiv 0$, $G_f(\omega)$ is reduced to $\sum_{nm} G_{nm}^{at}(\omega)$, i.e. the Hubbard I method reproduces the atomic limit. Setting $U \equiv 0$ gives $G_f(\omega) = [\omega + \mu - \varepsilon_f - \Delta(\omega)]^{-1}$, which is the correct band limit. Unfortunately, at half–filling this limit has a pathology connected to the instability towards Mott transition at any interaction strength U.

0.6cm

[1] J. Hubbard, Proc. Roy. Soc. (London) A**281**, 401 (1964).
[2] W. Nolting, and W. Borgiel, Phys. Rev. B **39**, 6962(1989).
[3] L. M. Roth, Phys. Rev. **184**, 451 (1969).

Running program

- compile program typing "make" at the source files location. Makefile was tested on Linux OS using PGI compiler. Please adjust it depending on operating system and compiler used.
- Edit “**input**" file which has the following structure:

```
1          IMOD
0.0        EF
2.0        U
0.016      TEMP
4          NDEG
500        NMSB
```

500	OMAX
500	NOMG
50	WEND
F	COMPUTE_REAL

where IMOD = 1 is the Hubbard Model, IMOD=3 the Anderson impurity model, EF is the impurity level, U is value of Coulomb repulsion, TEMP is temperature to be used used to create for Matsubara points grid, NDEG is the degeneracy, NMSB is the number of Matsubara point, OMAX is the imaginary frequency cutoff, NOMG is the number is real frequency points, WEND is the real frequency cutoff, COMPUTE_REAL is flag once is “True" tell program to produce the self-consistent solution (last iteration) on real axis.

1. Program’s input consists from one more file (provided the Anderson impurity model item is chosen in “**input**" file):

“**delta.dat**" containing the hybridization function.
The structure is the following:

Column #	Value
1	Frequency
2	Re Delta
3	Im Delta

1. Run the program executable (“main" is the default name).
2. Program’s output consists from the following files:

1) “**gfsig_iw.dat**" containing Green’s function (GF) and the self-energy (SE) on Matsubara axis
2) “**gfsig_re.dat**" containing Green’s function (GF) and the self-energy (SE) on real axis provided flag "COMPUTE_REAL" is “TRUE".
They have the same structure:

Column #	Value
1	Frequency
2	Re GF
3	Im GF
4	Re SE
5	Im SE

3) “**grid_re.dat**" containing real frequency grid for the hybridization function (delta).
4) “**grid_im.dat**" containing Matsubara frequency grid for the hybridization function (delta).
Both files have the same structure:

Column #	Value
1	Frequency

Program content

Makefile	main make file
dmf_cmpdiag.f	The solution of the generalized eigenvalue problem

imp_sunatm.f	Solves Anderson impurity model, returns GF and SE.
lib_broy6.f	Broyden mixing.
lib_cinv.f	Finds inverse of square matrix.
lib_csplines.f	Splines complex function from one to another grid.
lib_deriv1.f	Calculates radial derivative.
lib_morefun.f	Calculations of a few auxiliary functions
lib_pade.f	Realization of Páde procedure.
lib_splin3.f	Computed a natural spline approximation of third order.
mod_common.f	File containing common modules used across the program.
mod_dimart.f	File containing common modules used across the program.
mod_init.f	File containing common modules used across the program.
sunhub1.f	Main program.