I. DETAILS OF FLEX EVALUATION

The FLEX approximation sums up three different infinite classes of diagrams, particle-particle ladder T_{pp} and two particle-hole channels which can be grouped together into particle-hole T-matrix T_{ph} in the following way

$$\begin{aligned}\hat{T}^{pp}(i\Omega) &= (1-\hat{U}\hat{\chi}^{pp}_{i\Omega})^{-1}\hat{U}\hat{\chi}^{pp}_{i\Omega}\hat{U}\hat{\chi}^{pp}_{i\Omega}\hat{U} &(1)\\ \hat{T}^{ph}(i\Omega) &= (1-(\hat{V}+\hat{W})\hat{\chi}^{ph}_{i\Omega})^{-1}(\hat{V}+\hat{W}) \\ &- (\hat{V}+\hat{W})\hat{\chi}^{ph}_{i\Omega}\hat{V}. &(2)\end{aligned}$$

Here $V_{1234}=U_{1324}$, $W_{1234}=-U_{1342}$ and polarization diagrams are

$$\chi^{pp}_{1234}(i\Omega) = -T\sum_{i\omega'} G_{23}(i\omega')G_{14}(i\Omega-i\omega'), \tag{3}$$

$$\chi^{ph}_{1234}(i\Omega) = -T\sum_{i\omega'} G_{23}(i\omega')G_{41}(i\Omega+i\omega'). \tag{4}$$

We assumed here a scalar product of the form $(\hat{A}\hat{B})_{1234}=\sum_{56}A_{1256}B_{5634}$. Finally the FLEX self-energy becomes

$$\Sigma^{(FLEX)}_{12}(i\omega) = T\sum_{i\Omega}\left(T^{pp}_{1432}(i\Omega)G_{34}(i\Omega-i\omega)+T^{ph}_{1432}(i\Omega)G_{43}(i\Omega+i\omega)\right). \tag{5}$$

The equations can be considerably simplified if the hybridization Δ is diagonal and the Coulomb repulsion is of the form $U_{\alpha\beta}n_\alpha n_\beta$. In this case, propagators in the above equations are also diagonal, i.e., $G_{12}=G_{11}\delta_{12}\equiv G_1$ and the two particle-hole channels do not mix anymore and can be separately summed up. The two ladders become a simple geometric series and can be explicitely inserted into self-energy expression

$$\Sigma^{pp}_{\alpha}(i\omega) = T\sum_{\beta}U_{\alpha\beta}\sum_{i\Omega}G_\beta(i\Omega-i\omega)\left[\frac{U_{\alpha\beta}\chi^{pp}_{\alpha\beta}(i\Omega)}{1-U_{\alpha\beta}\chi^{pp}_{\alpha\beta}(i\Omega)}-U_{\alpha\beta}\chi^{pp}_{\alpha\beta}(i\Omega)\right] \tag{6}$$

$$\Sigma^{ph}_{\alpha}(i\omega) = T\sum_{\beta}U_{\alpha\beta}\sum_{i\Omega}G_\beta(i\omega+i\Omega)\left[\frac{U_{\alpha\beta}\chi^{ph}_{\alpha\beta}(i\Omega)}{1-U_{\alpha\beta}\chi^{ph}_{\alpha\beta}(i\Omega)}-U_{\alpha\beta}\chi^{ph}_{\alpha\beta}(i\Omega)\right] \tag{7}$$

where the simplified polarisation diagrams are

$$\chi^{pp}_{\alpha\beta}(i\Omega) = -T\sum_{i\omega'}G_\alpha(i\omega')G_\beta(i\Omega-i\omega') \tag{8}$$

$$\chi^{ph}_{\alpha\beta}(i\Omega) = -T\sum_{i\omega'}G_\alpha(i\omega')G_\beta(i\Omega+i\omega'). \tag{9}$$

The third GW channel can also be analytically summed up for the SU(N) case, but if the crystal field splitting is assumed, one still needs to solve a matrix equation

$$T_{\alpha\beta}(i\Omega) = U_{\alpha\beta}-U_{\alpha\beta'}\chi^{ph}_{\beta'\beta'}(i\Omega)T_{\beta'\beta}(i\Omega) \tag{10}$$

to obtain the self-energy of this channel

$$\Sigma^{gw}_{\alpha}(i\omega) = -T\sum_{i\Omega}G_\alpha(i\omega+i\Omega)\left[T_{\alpha\alpha}(i\Omega)+\sum_\beta U^2_{\alpha\beta}\chi^{ph}_{\beta\beta}(i\Omega)\right]. \tag{11}$$

In the SU(N) case, the summation can be performed and the result is

$$\Sigma^{gw}_{\alpha}(i\omega) = -T\sum_{i\Omega}G(i\omega+i\Omega)\frac{N-1}{N}\left[\frac{U}{1+(N-1)U\chi^{ph}(i\Omega)}-\frac{U}{1-U\chi^{ph}(i\Omega)}+NU\chi^{ph}(i\Omega)\right]. \tag{12}$$

Somethimes it is more convenient to work on real axis and thus avoid the problem of analytic continuation. The price one pays is that the functions on real axis have more structure to resolve and the equidistant mesh is in many

cases too expensive. Therefore, it is desired to work with nonequidistant meshes on real axes with more points around zero frequency where Fermi function drops quickly and most of sensitive structure is located. The FLEX equations can be analytically continued to real axes by the standard contour integration and the results is

$$\mathrm{Im}\{\chi^{ph}_{\alpha\beta}(\Omega)\} = \pi \int d\xi [f(\xi-\Omega) - f(\xi)] A_\alpha(\xi) A_\beta(\xi-\Omega) \tag{13}$$

$$\mathrm{Im}\{\chi^{pp}_{\alpha\beta}(\Omega)\} = \pi \int d\xi [f(\xi-\Omega) - f(\xi)] A_\alpha(\xi) A_\beta(\Omega-\xi) \tag{14}$$

$$T_{\alpha\beta}(\Omega) = U_{\alpha\beta} - U_{\alpha\beta'}\chi^{ph}_{\beta'\beta'}(\Omega) T_{\beta'\beta}(\Omega) \tag{15}$$

$$\mathrm{Im}\Sigma^{ph}_\alpha(\omega) = -\sum_\beta U_{\alpha\beta} \int d\xi [f(\xi-\omega) + n(\xi)] A_\beta(\omega-\xi) \mathrm{Im}\Big\{ \frac{U_{\alpha\beta}\chi^{ph}_{\alpha\beta}(\xi)}{1 - U_{\alpha\beta}\chi^{ph}_{\alpha\beta}(\xi)} - U_{\alpha\beta}\chi^{ph}_{\alpha\beta}(\xi) \Big\} \tag{16}$$

$$\mathrm{Im}\Sigma^{pp}_\alpha(\omega) = -\sum_\beta U_{\alpha\beta} \int d\xi [f(\xi-\omega) + n(\xi)] A_\beta(\xi-\omega) \mathrm{Im}\Big\{ \frac{U_{\alpha\beta}\chi^{pp}_{\alpha\beta}(\xi)}{1 + U_{\alpha\beta}\chi^{pp}_{\alpha\beta}(\xi)} - U_{\alpha\beta}\chi^{pp}_{\alpha\beta}(\xi) \Big\} \tag{17}$$

$$\mathrm{Im}\Sigma^{gw}_\alpha(\omega) = \int d\xi [f(\xi-\omega) + n(\xi)] A_\alpha(\omega-\xi) \mathrm{Im}\Big\{ T_{\alpha\alpha}(\xi) + \sum_\beta U^2_{\alpha\beta}\chi^{ph}_\beta(\xi) \Big\} \tag{18}$$

In the SU(N) case, the last contribution can again be simplified to

$$\mathrm{Im}\Sigma^{gw}_\alpha(\omega) = -\int d\xi [f(\xi-\omega) + n(\xi)] A_\alpha(\omega-\xi) U \frac{N-1}{N} \mathrm{Im}\Big\{ \frac{U\chi}{1-U\chi} + \frac{(N-1)U\chi}{1+(N-1)U\chi} - NU\chi \Big\} \tag{19}$$

II. RUNNING FLEX PROGRAM

There are two versions of the program awailable to download

- SU(N) program is written for the Hubbard model only, but include DMFT self-consistent loop
- the second version supports crystal field splitting and solves only the impurity problem. The DMFT loop should be used outside the program (using for example **ksum** program)

The compilation is straighforward on PC linux computers. User should type *make PLATFORM=PC RELEASE=1* for optimized compilation. The executable with name **flx** is created. To run the example program, one needs some files located in subdirectory **work/start**. To get short help on various options available, one should execute **flx** with no arguments.

A. SUN FLEX

The following information is printed by executing the SU(N) version of the program
[...|] flex_input_file [options]
Options:

file	Filename of the trial spectral function
-T	Temperature (0.001)
-U	Coulomb repulsion (1)
-mu	Chemical potential (0.5)
-alpha	Mixing parameter (0.2)
-alpha_mu	Mixing parameter for mu (1)
-nd	Desired doping (1)
-ph_reduce	Weather to screen PH diagrams with PP effective interaction (0)
-Na	Number of additional points around fermi function (50)
-Ns	Twice the number of bands (2)
-si	Data will be read from standard input
-max_diff	Criterium tu stop self-consistency (0.0001)
-max_steps	Criterium tu stop self-consistency (300)
-ph	Wheather to include PH-channel (1)
-pp	Wheather to include PP-channel (1)
-gw	Wheather to include GW-channel (1)
-g0	Use G or G0 (g0=1 - use G0, g0=0 - use G) (0)
-hs	Weather to fix hartree term to be proportional to the input nd (0)
-do	Debug output (prints spec. fun. every iteration) (0)

Below, some more information is given for each parameter of the program. But first, let us run the example program to illustrate how it works. In the subdirectory **work** a file with name **history.flex** is located which saves all previous command lines for easier restart of the program in the future. The last command line is already written and user can copy the command line to the shell and execute
../flx start/Aflex_N_4_U_1_nd_0.8_T_0.001_pp_ph_gw -nd 2 -ph_reduce 1 -Ns 4 -do 1
After approximately 40 iterations the self-consistency is reached. Results are printed in the following files

- Aflex_... - The self-consistent local spectral function
- brisi.xxx - The spectral function and self-energy in each iteration

Information, printed to the standard output includes

- Difference = 9.86515e-05 is the difference between self-energies in the last two successive steps
- nd = 2; is the total impurity occupation
- mu = 1.50118; is the chemical potential
- alpha=0.2; is the mixing parameter used
- dmu=-1.99091e-07; if mu is changed in each iteration to get the desired doping (alpha_mu>0) this is the difference between last two chemical potentials

Finally, let me give some more information on the input parameters for the SUN flex program

- alpha: is the mixing parameter for the Green's function
- alpha_mu: is the mixing parameter for the chemical potential. If calculation is done at contant doping, chemical potential is adjusted in every step and alpha_mu is the corresponding mixing parameter. If it is set to zero, the calculation is done at fixed chemical potential.
- nd: desired total impurity occupation (relevant only if alpha_mu>0)
- ph_reduce: Particle hole channel gives unphysically large polarization χ for intermediate and large U. User can choose to reduce Coulomb repulsion U in particle-hole channels by screening it with the particle-particle graphs: $U_{eff} = U/(1 + U\chi^{pp}(0))$
- Na : The mesh used in calculating convolutions should be densed at two different localtion. Additional points are added to the neighborhood of the second peak or Fermi function. Then number of those additional points is Na.
- Ns : For SU(N) model it is just N.
- si : Instead of specifying input file, user can redirect certain output in the shell to the input of the program. In this case, -si switch should be used in the command line.

- max_diff: When the difference between two successive Green's functions is less than that number, the iteration is finished.
- max_steps: Iteration is never longer than max_steps to avoid infinite loops.
- ph: Every channel of the FLEX three channels can be switched on or off. If ph is set to 1(0), particle-hole ladder is included (excluded).
- pp: If pp is set to 1(0), particle-particle ladder is included (excluded).
- gw: If gw is set to 1(0), gw diagrams are included (excluded).
- g0: The diagrams can be evaluated with fully dressed propagators (electron Green's functions) or undressed propagators (only Hartree term self-consistently added). If g0 is set to 1, bare propagators are used, otherwise they are fully dressed.
- hs: If working at fix doping (alpha_mu>0) it is much easier to get converged results by fixing hartree term to the input n. The reason is that the results are very sensitive to the Hartree term and small change can result in numerical instability.
- do: If set to 1, there is much more output at each iteration

B. FLEX for Crystal field splitting

Executing **flx** in the directory **CSF** results in the following short help message

[...|] flex_input_file [options]
Options:

-T	Temperature (0.001)
-U	Coulomb repulsion (1)
-Ns	Degeneracy of bands ((2))
-alpha	Mixing parameter (0.2)
-ph_reduce	Weather to screen PH diagrams with PP effective interaction (0)
-Na	Number of additional points around fermi function (50)
-si	Data will be read from standard input
-ph	Wheather to include PH-channel (1)
-pp	Wheather to include PP-channel (1)
-gw	Wheather to include GW-channel (1)
-gi	The input is green's function rather than spectral functions (0)

All parameters here have the same meaning as above except the parameter *-gi* is added and parameter *Ns* is more complicated to set properly. The case that we are able to treat with this simple peace of code is the case of crystal field splitted levels that have arbitrary degeneracy but they are not coupled through the off-diagonal elements of hybridization Δ. A typical case is an f shell that can acquire 7 electrons and is usually splitted into 6+6+2 multiplets. The input for Ns should than be *-Ns "(6,6,2)"*. Note here that the quotation marks are required for every nonatomic expression for shell to interpret the input properly.

The example command line is as usual prepared in the **CFS/work/history.flex** and reads

../flx start/gloc.inp.210 -gi -U 4 -T 0.03 -Ns "(6,6,2)" -ph_reduce 1 -gi

In couple of second, the output is written in the file called **sig.out** that contains all nonequivalent self-energies (in this case three). In addition, there are also output files called **Sigma.xxx**. Each of them corresponds to one nonequivalent bath (Sigma.000 to the first bath, Sigma.001 to the second and so on) and contains the following information

- second column contains the real part of the input green's function
- third column contains the input spectral function
- fourth and fifth column is the total self-energy
- sixth column is the second order self-energy only
- seventh column is the particle hole ladder contribution only
- eighth column is the particle particle ladder contribution only
- ninth column is the gw contribution only