

FlapwMBPT: Input Variables

Andrey L. Kutepov

June 19, 2021

Contents

1	Introduction	2
2	Mandatory input	3
2.1	iter_dft	3
2.2	iter_hf	3
2.3	iter_gw	3
2.4	iter_qp	3
2.5	iter_psi	3
2.6	iter_bsp	3
2.7	space_group	3
2.8	origin	3
2.9	symgen	4
2.10	bravais_lat	4
2.11	a_lat	4
2.12	b_lat	4
2.13	c_lat	4
2.14	alpha_lat	4
2.15	beta_lat	4
2.16	gamma_lat	4
2.17	Atomic_positions	4
3	General control	4
3.1	acc_level	4
3.2	restart_begin	5
3.3	restart_end	5
3.4	admix	5
3.5	adspin	5
3.6	adm_gw	5
3.7	acc_it_gw	5
3.8	iexch	5
3.9	irel	5
3.10	rel_interst	6
3.11	temperature	6
3.12	vv0	6
4	Group MPI	6
4.1	nproc_t	6
4.2	nproc_k	6
4.3	nproc_b	6
5	Group FILES	6
5.0.1	tmpfile	6
6	Group BASIS	6
6.1	rkmax	6
6.2	gmax_rho_fact	6
6.3	eps_pb	6
6.4	eps_pb_vrt	6
6.5	mode_pb	6
7	Group BND_PLOT	6
7.1	n_k_div	6
8	Group DOS	7
8.1	emindos	7
8.2	emaxdos	7
8.3	ndos	7
8.4	e_small	7
8.5	e_small_bos	7
8.6	pdos_lm	7
9	Group K_POINT	7
9.1	ndiv	7
9.2	k_line	7
9.3	ndiv_c	7
10	Group MAGNET	7
10.1	b_extval	7
10.2	iter_h_ext	7
10.3	b_ext	7

11 Group COULOMB	7	19 Group OMEGA MESH	13
11.1 eps_coul	7	19.1 n_omega_exa	13
11.2 vcoul_mi	8	19.2 n_omega_geom	13
11.3 vcoul_li	8	19.3 n_omega_asy	13
		19.4 omega_geom	13
12 Group HARTREE-FOCK	8	19.5 omega_max	13
12.1 alpha_hbr	8	19.6 omega_last_step	13
12.2 omega_hse	8	20 Group NU MESH	13
12.3 hybrid_type	8	20.1 n_nu_exa	13
13 Group W	8	20.2 n_nu_geom	13
13.1 w_sc_gw	8	20.3 n_nu_asy	13
13.2 w_sc_qp	8	20.4 nu_geom	14
13.3 w_sc_psi	8	20.5 nu_max	14
13.4 w_sc_bsp	9	20.6 nu_last_step	14
14 Group VERTEX_P	9	21 Group ATOMIC DATA	14
14.1 psi_p	9	21.1 magn_shift	14
14.2 bse_kernel_p	10	21.2 smt_wgt	14
14.3 iter_ladder_p	11	21.3 smt_scale	14
14.4 nu_w_stat	11	21.4 r1	14
15 Group VERTEX_S	11	21.5 nrad	14
15.1 psi_sig	11	21.6 lmb	14
15.2 psi2_sig	12	21.7 lmpb	14
15.3 iter_sigma_gwg	12	21.8 lmv	14
16 Group VERTEX_XI	12	21.9 low_augm	14
16.1 iter_ladder_xi	12	21.10high_augm	14
16.2 q_suscept	12	21.11lmax_ntle_add	15
16.3 chi_cmp	12	21.12ntle_add	15
16.4 vrt_x_appr	12	21.13augm	15
16.5 nrax_chi	12	21.14atocc	15
16.6 freq_chi	13	21.15ptnl	15
16.7 theta_bse	13	21.16corr	15
17 Group OPTICS	13	21.17idmd	15
17.1 opt_mode	13	21.18mode_pb	15
17.2 opt_loc fld	13	21.19lmpb_c	15
18 Group TAU MESH	13		
18.1 n_tau	13		
18.2 exp_tau_gw	13		

1 Introduction

The philosophy of the input file is to provide a general user with different options of setting up the calculation. For this purpose a default/automatic system of setting up has been developed. Only the very minimum of information is mandatory in the input file.

This mandatory information includes the specification of the approximation level (DFT, Hartree-Fock, QSGW, scGW, scGW+Vertex) and crystallographic information: space group number, lattice parameters and Wyckoff positions of the atoms. The rest of the input information is generated automatically. Of course an option of changing any automatically generated variable is provided as well.

Now a little more about how automatically generated variables can be changed. They can be changed by placing the corresponding variables after the mandatory part of the input file.

Mandatory input is separated from the rest by the line '----- End of mandatory input -----'.

Within any of the above specified approximation level (i.e. DFT, HF, QSGW, scGW, or scGW+Vertex) the result depends on the convergence with respect to truncation of the expansions, like plane wave expansion in the interstitial region or spherical harmonic expansions inside the muffin tin spheres. The result also depends on the accuracy of the numerical methods involved, like numerical integration of radial equations or taking the integrals. Unfortunately, the accuracy is competitive with the computation time. So, one has to trade and select such set up which provides necessary convergence and at the same time is not too time consuming.

The simplest way to control automatically the accuracy level is to introduce variable 'acc_level' which sets up all the variables automatically according to the accuracy level (presently 0,1, or 2) and doesn't require any knowledge of the method. For knowledgeable users, and also when one needs to change only one (or a few) specific variable, the option of updating any specific entry is provided.

Below the variables are subdivided into subgroups. But this is only for convenience of reading. The order of variables (after the mandatory part) in the input file is irrelevant.

2 Mandatory input

2.1 iter_dft

Number of DFT iterations.

2.2 iter_hf

Number of Hartree-Fock/Hybrid iterations.

2.3 iter_gw

Number of GW iterations.

2.4 iter_qp

Number of QSGW iterations.

2.5 iter_psi

Number of iterations of Hedin's equations with those diagrams beyond GW approximation which can be constructed from a given Ψ -functional. Additional approximations apply (as compared to GW itself) - see below.

2.6 iter_bsp

Number of iterations of Hedin's equations with the diagrams for the polarizability following from the BSE (Bethe-Salpeter Equation) and with the diagrams for the self energy constructed from a given Ψ -functional. Additional approximations apply (as compared to GW itself) - see below.

2.7 space_group

Space group number. When space group is known, it can be used instead of explicit generators 'symgen' (see below).

2.8 origin

For some space groups there are ambiguities about choice of the origin. One has to make sure that atomic positions and the choice of the group generators use the same origin of the coordinate system. See file 'gr_gen_from_sg.F' for the correspondence of the 'origin' variable and the actual choice of the origin of the coordinate system. If 'origin' is not provided, it is assumed that it is equal to 1.

2.9 symgen

Provides the generators of the symmetry group. Gives the minimal set of operations which (after all mutual multiplications) produces full set of symmetry operations. In the string, each generator should end with the symbol `_`. The conventions are the following:

- E - identity operation;
- I - space inversion;
- RNA - N-fold rotation around axis A (A = X, Y, Z, D (diagonal of the cube), or S in case of screw axis);
- MV - reflection in the plane orthogonal to the vector V (V = X, Y, Z, or given explicitly by numbers like (0.5,0.5,0.5)). If V is given explicitly by 3 numbers then they should be expressed in terms of lattice translations vectors A, B, and C (see further in the text).
- S:TV - provides the fractional translation along vector V associated with an operation S which can be any of the above three (inversion, rotation, or reflection).
- For more complex generators one should use the symbol `*` with a meaning of multiplication. For example, a generator `R6Z*MZ` means that every rotation around Z-axis on $2\pi/6$ is accompanied with reflection in XY plane.
- Some operations leave the solid invariant only if associated with time reversal. This can be indicated by placing minus sign in front of the corresponding operator.

2.10 bravais_lat

Type of Bravais lattice. Can be one of the following: cP, tP, hP, oP, mP, cF, oF, cI, tI, oI, hR, oC, oA, mC.

This variable is needed only when `'symgen'` is used. When `'sym_group'` is provided, `'bravais_lat'` is assigned automatically.

2.11 a_lat

Only minimal information should (and must be only minimal!) be provided. This choice depends on the crystal type. For instance, in cubic system only one lattice parameter has to be provided.

Length of the lattice vector A. In Angstroms.

2.12 b_lat

Length of the lattice vector B. In Angstroms.

2.13 c_lat

Length of the lattice vector C. In Angstroms.

2.14 alpha_lat

Angle α between lattice vectors **B** and **C**. In degrees.

2.15 beta_lat

Angle β between lattice vectors **C** and **A**. In degrees.

2.16 gamma_lat

Angle γ between lattice vectors **A** and **B**. In degrees.

2.17 Atomic_positions

List of in-equivalent atoms (sorts) has to be provided. Each sort is characterized by a string. For instance, the string (Na '1/2,1/2,1/4') tells us that there is sodium atom at a position '1/2,1/2,1/4', where coordinates are relative to the conventional unit cell. The format of the numbers can be one of the following: 'a', 'a/b', 'sqrt(a)', or 'sqrt(a/b)' with a and b being integer or real.

3 General control

3.1 acc_level

Can be 0, 1, or 2. Sets up the accuracy level. Increase of this variable means making the result more accurate by changing (increasing) all the variables which potentially influence on the accuracy. It involves:

number of plane waves in the Interstitial, different Lmax inside the MT spheres, number of points in integrals, number of points in the Brillouin zone and so on. Level 0 is mostly for demonstrations. Level 1 is for most applications. Level 2 is for more accurate (but also more involved) calculations.

DEFAULT value is 1.

3.2 restart_begin

Works only with DFT calculations. DEFAULT value is 0, which means there is no restart. In order to use the feature, restart file has to be available. It is generated when the previous calculation was performed with 'restart_end' equal 1. When 'restart_begin' is equal 2, all information from the previous run is used. It allows smooth continuation. When 'restart_begin' is equal 1, information about the basis set from the previous run is ignored. It can be used when the basis set has changed. Some other variables (like Lmax for density, and Gmax for density) can be different. Ideally, in the future, restrictions on the differences should be reduced to a minimum.

3.3 restart_end

Works with DFT and/or with post DFT/HF calculations. DEFAULT value is 0. It should be equal 1 when the subsequent restart is anticipated. In this case the restart information is stored in files. Restart for DFT run is governed by restart_begin variable. Restart for post DFT/HF calculations is assumed when variable iter_dft is zero.

3.4 admix

Mixing in DFT iterations (charge).

3.5 adspin

Mixing in DFT iterations (spin).

3.6 adm_gw

Initial mixing for HF, QP, GW, or GW+ iterations. In all these cases self energy is mixed.

3.7 acc_it_gw

Final mixing for HF, QP, GW, or GW+ iterations. Mixing is changed linearly from iteration to iteration from its initial value to its final.

3.8 iexch

Specifies the LDA parametrization for the exchange-correlation energy. It is integer number: iexch=100*igga+ilda, where igga=0 (pure LDA), igga=1 for GGA91, and igga=2 for GGA96 (==PBE). For the LDA part (ilda):

- ILDA = 0 - no exchange-correlation
- ILDA = 1 - U.von.Barth and L.Hedin, J.Phys.C 5, 1629 (1972)
- ILDA = 2 - O.E.Gunnarsson and S.Lundqvist, Phys.Rev.B 13, 4274 (1976)
- ILDA = 3 - V.L.Moruzzi, J.F.Janak, and A.R.Williams, Calculated Electronic Properties of Metals (New York, Pergamon Press, 1978)
- ILDA = 4 - S.H.Vosko, L.Wilk, and M.Nusair, Can.J.Phys.58, 1200 (1980). Parametrization of the Ceperly-Alder form (D.M.Ceperly, B.J.Alder, Phys.Rev.Lett.45, 566 (1980))
- ILDA = 5 - J.Perdew, X.Wang, Phys.Rev. B, 45, 13244(1992)
- ILDA = 6 - Original Gaspar-Kohn-Sham ($\rho^{**1/3}$)

In case ILD0<0 only correlation is included.

3.9 irel

Takes care of the relativistic effects:

- = 0: nonrelativistic calculation;
- = 1: scalar-relativistic calculation;
- = 2: fully (Dirac) relativistic calculation.

3.10 rel_interst

If **rel_interst**=T and **irel**=2, the interstitial region is described fully relativistically (available only for DFT runs). In other situations spin-orbit effects are neglected in the interstitial region.

3.11 temperature

Temperature in K.

3.12 vv0

Defines the relative volume (V/V_0). V_0 is given by crystal structure and lattice parameter above.

4 Group MPI

4.1 nproc.t

Number of MPI processes associated with (τ, ω) -parallelization. Should be less or equal to the number $n_\tau/2 + 1$.

4.2 nproc.k

Number of MPI processes associated with parallelization over Brillouin zone k-points. Should be less or equal to the number of irreducible points in the Brillouin zone.

4.3 nproc.b

Number of MPI processes associated with parallelization over basis set.

The product **nproc.t*nproc.k*nproc.b** should be equal to the total number of MPI processes which are allocated for this job.

5 Group FILES

5.0.1 tmpfile

Defines where the files for restart go. It can be the same directory as the rest of the files. But if the local directory has small capacity, then it is of advantage

to use other (like SCRATCH) directory with larger capacity. Default value: `rstfile='./'`.

6 Group BASIS

6.1 rkmax

Defines the number of plane waves in the basis set by G_{max} from the relation: $S_{avg}G_{max} = rkmax$. S_{avg} is average radius of MT sphere. Default value depends on the accuracy level and the kind of atoms constituting the material.

6.2 gmax_rho_fact

Scales G_{max} for the LAPW basis to the G_{max} for the density/potential. Default value is 2.

6.3 eps_pb

Optimization parameter in constructing the product basis inside of MT spheres. Usual choice is $10^{-3} \div 10^{-4}$.

6.4 eps_pb_vrt

Optimization parameter in constructing of the reduced product basis for vertex corrected calculations. Usual choice is $10^{-2} \div 10^{-3}$.

6.5 mode_pb

If it is equal to 1 then product basis inside MT explicitly includes normalized r^L functions. DEFAULT value is 0.

7 Group BND_PLOT

7.1 n_k_div

Number of divisions along every line, which connects two k-points of high symmetry. For bands plotting only. The option relevant only for the DFT presently.

8 Group DOS

8.1 emindos

The lowest energy for DOS plotting (relative to chemical potential, in Ry).

8.2 emaxdos

The highest energy for DOS plotting (relative to chemical potential, in Ry).

8.3 ndos

Number of points for DOS plotting (equidistant mesh).

8.4 e_small

A small positive shift when calculating retarded fermionic functions on real axis.(Ry) Default value is 0.0005 Ry.

8.5 e_small_bos

A small positive shift when calculating retarded bosonic functions on real axis.(Ry) Default value is 0.001 Ry.

8.6 pdos_lm

Can be used to evaluate also LM-resolved PDOS. It should be provided as 'pdos_lm lm isort'.

9 Group K_POINT

9.1 ndiv

Three numbers (divisions) which define the k-mesh in Brillouin zone.

9.2 k_line

Symbol A, B, or C, defying the direction in the reciprocal space for certain output purposes.

9.3 ndiv_c

Defines (similar to the variable **ndiv**) the auxiliary mesh in the Brillouin zone. Each of these three numbers should divide exactly the corresponding number of the variable **ndiv_b**. This mesh is used to evaluate the high order diagrams for the polarizability and self energy (beyond GG and GW correspondingly).

10 Group MAGNET

10.1 b_extval

Absolute value of external magnetic field (Ry/ μ_B).

10.2 iter_h_ext

Seven digits placed together. They are used for initial spin polarization together with variable **magn_shift** (see below in the atom's description). First digit means how many DFT iterations are to be performed with applied field as specified by **magn_shift** for every atom in the unit cell. Correspondingly the rest of the digits mean the same for HF, GW, QP, or PSI runs. One can start with non-spin-polarized DFT calculation but then switch on the polarization at, for instance, GW level (**iter_h_ext**=0020000 if we want to start with two GW iteration with applied field).

10.3 b_ext

Three numbers, defying the direction of constant external magnetic field, associated with variable **b_extval**.

11 Group COULOMB

11.1 eps_coul

The variable is used to optimize the product basis by keeping only those eigen vectors of the matrix of Coulomb interaction which correspond to the eigen values (normalized to the largest eigen value) larger than **eps_coul**.

11.2 vcoul_mi

Governs the representation of Coulomb interaction $V(r,r')$ when arguments r belongs to one of the MT spheres whereas argument r' belongs to the interstitial region. Can be 'KSP' or 'EWD'. In the first case, it is evaluated as the matrix elements of the plane wave expansion of the Coulomb interaction with respect to the Product Basis set. In the second case, the Ewald method is used to evaluate $V^k(tL, r')$ and after that FFT is used to find the coefficients in the plane wave expansion. By default it is 'KSP'.

11.3 vcoul_ii

Governs the representation of Coulomb interaction $V(r,r')$ when both arguments (r and r') belong to the interstitial region. Can be 'KSP' or 'EWD'. In the first case, the coefficients in the normalized plane wave expansion are $\frac{8\pi}{|k+G|^2}$. In the second case, the Ewald method is used to evaluate $V^k(r, r')$ and after that FFT is used to find the coefficients in the plane wave expansion. By default it is 'KSP'.

12 Group HARTREE-FOCK

12.1 alpha_hbr

Effective only during Hartree-Fock iterations. Shows the share of the non-local exchange in hybrid functional. Default value is 0.25.

12.2 omega_hse

Yukawa screening parameter for YS-PBE0 functional. Default is 0.165.

12.3 hybrid_type

- = 0 - Hartree-Fock; (Default)
- = 1 - PBE0;
- = 2 - YS-PBE0;

13 Group W

13.1 w_sc_gw

If **w_sc_gw**=scf then polarizability and W are renewed on every GW iteration ($P=GG$ with G being GW Green's function). If **w_sc_gw**=non then P and W are fixed during GW iterations. In this case W is defined by $P: W=V+VPW$. The polarizability P is defined based on the Green's function from an approach preceding the GW iterations. If **bse_kernel_p**='0000' then the Random Phase Approximation (RPA) is used to get P . If **bse_kernel_p**≠'0000' then the physical polarizability is evaluated (functional derivative of electronic density with respect to the total field).

13.2 w_sc_qp

If **w_sc_qp**=scf then polarizability and W are renewed on every QP iteration ($P=GG$ with G being QP Green's function). If **w_sc_qp**=non then P and W are fixed during QP iterations. In this case, in the evaluation of diagrams for P , the self-consistent Green function from LDA, HF, or GW calculations is used depending on what kind of iterations were performed immediately before QP iterations. W is defined by $P: W=V+VPW$. The polarizability P is defined based on the Green's function from an approach preceding the GW iterations. If **bse_kernel_p**='0000' then the Random Phase Approximation (RPA) is used to get P . If **bse_kernel_p**≠'0000' then the physical polarizability is evaluated (functional derivative of electronic density with respect to the total field).

13.3 w_sc_psi

If **w_sc_psi**=scf then polarizability and W are renewed on every PSI iteration ($P=G*VERTEX(G,W)*G$ with G, W being PSI Green's function and W). If **w_sc_psi**=non then P and W are fixed during PSI iterations. In this case, in the evaluation of diagrams for P , the self-consistent Green function from LDA, HF, GW, or QP calculations is used depending on what kind of iterations were performed immediately before

PSI iterations. W is defined by P : $W=V+VPW$. The polarizability P is defined based on the Green's function from an approach preceding the GW iterations. If `bse_kernel_p='0000'` then the Random Phase Approximation (RPA) is used to get P . If `bse_kernel_p≠'0000'` then the physical polarizability is evaluated (functional derivative of electronic density with respect to the total field).

13.4 w_sc_bsp

If `w_sc_bsp=scf` then polarizability and W are renewed on every BSP iteration ($P=G*\text{VERTEX}(G,W)*G$ with G , W being BSP Green's function and W). If `w_sc_bsp=non` then P and W are fixed during BSP iterations with P calculated according to the variable `bse_kernel_p` (see below). In the evaluation of diagrams for P the self-consistent Green function (and W - if appropriate) from LDA, HF, GW, QP, or PSI calculations is used depending on what kind of iterations were performed immediately before BSP iterations.

14 Group VERTEX_P

14.1 psi_p

It is a string of 5 symbols. It describes which diagrams (and in which approximation) to be used to approximate the irreducible polarizability during the iterations with vertex corrections based on Ψ -functional (conserving approximation). Each of the entries of the variable `psi_p` generates the diagrams for the irreducible polarizability from a specific diagram of the Ψ -functional.

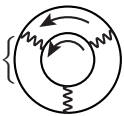
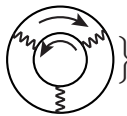
The conventions are the following (at the moment only first two entries and the first term of the third entry are implemented):

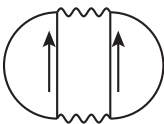
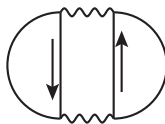
First entry:

$$\Psi_1 = \frac{1}{4} \left(\text{diagram} \right) \quad P_1 = \left(\text{diagram} \right)$$

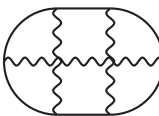


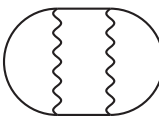
Second entry:



$$\Psi_2 = \frac{1}{6} \left\{ \text{diagram} + \text{diagram} \right\}$$



$$P_2 = \left(\text{diagram} \right) + \left(\text{diagram} \right)$$



Third entry:

$$\Psi_3 = \frac{1}{2} \left(\text{diagram} \right)$$


$$P_3 = \left(\text{diagram} \right)$$


$$+ \left(\text{diagram} \right) + \left(\text{diagram} \right)$$



Fourth entry:

$$\Psi_4 = \frac{1}{6} \left(\text{diagram} \right) \quad P_4 = \left(\text{diagram} \right)$$



Fifth entry:

$$\Psi_5 = \frac{1}{8} \left\{ \text{diagram} + \text{diagram} \right\} + \dots$$



$$P_5 = \text{[Diagram 1]} + \text{[Diagram 2]} + \dots$$

As for the approximation associated with every entry the conventions are the following:

- 0 - do not include
- 1 - both G and W are local, W is static (no dependence on frequency)
- 2 - both G and W are local, W is dynamic
- 3 - W is local and static; G is in band representation
- 4 - W is local and dynamic. G is in band representation
- 5 - both G and W are in full (bands + PB) basis, W is static
- 6 - both G and W are in full (bands + PB) basis, W is dynamic

This part of the code is under development. Not all above branches are available. Recently tested options for the variable **psi_p** are the following: '60000', '66000', '40000', '20000'. For the details about latest developments please contact the author (Andrey Kutepov, e-mail: andreykutepov65@gmail.com).

14.2 bse_kernel_p

It is a string of 4 symbols. It describes which diagrams (and in which approximation) are to be used to approximate the irreducible polarizability during the iterations with vertex corrections based on the Bethe-Salpeter equation for the polarizability. Each of the entries of the variable **bse_kernel_p** generates the diagrams for the irreducible polarizability based on a specific kernel of the Bethe-Salpeter equation. Full kernel of the BSE is represented as a sum of the entries. The conventions for the corresponding entries (Θ_i) are the following:

First entry:

$$\Theta_1 = \text{[Diagram 1]}$$

Second entry:

$$\Theta_2 = \text{[Diagram 2]} + \text{[Diagram 3]}$$

Third entry (not yet implemented):

$$\Theta_3 = \text{[Diagram 4]} + \text{[Diagram 5]} + \text{[Diagram 6]}$$

Fourth entry (not yet implemented): Reserved for higher order diagrams for the kernel.

As for the approximation associated with every entry the conventions are the following (similar to the variable **psi_p**):

- 0 - do not include
- 1 - both G and W are local, W is static (no dependence on frequency)
- 2 - both G and W are local, W is dynamic
- 3 - W is local and static; G is in band representation
- 4 - W is local and dynamic. G is in band representation
- 5 - both G and W are in full (bands + PB) basis, W is static
- 6 - both G and W are in full (bands + PB) basis, W is dynamic

This part of the code is under development. Not all above branches are available. Recently tested

options for the variable **bse_kernel_p** are the following: '60000', '66000', '40000', '20000'. For the details about latest developments please contact the author (Andrey Kutepov, e-mail: andreykutepov65@gmail.com).

14.3 iter_ladder_p

Defines the number of iterations in solving the Bethe-Salpeter equation for vertex which is used in the polarizability.

14.4 nu_w_stat

Relevant only if static (frequency independent) W is used for the evaluation of high order diagrams. In this case, static W is defined as the value of dynamic W taken at specific frequency, namely at the frequency corresponding to the variable **nu_w_stat**.

15 Group VERTEX_S

15.1 psi_sig

It has the same meaning for the self energy as the variable **psi_p** has for the polarizability. It is a string of 5 symbols. It describes which diagrams (and in which approximation) are to be used to approximate the self energy based on the iterations with vertex corrections on Ψ -functional (conserving approximation). Each of the entries of the variable **psi_sig** generates the diagrams for the self energy from a specific diagram of the Ψ -functional.

The conventions are the following (at the moment only first two entries and the last two terms of the third entry are implemented):

First entry:

$$\Psi_1 = \frac{1}{4} \left(\text{circle with two vertical wavy lines} \right) \quad \Sigma_1 = \left(\text{horizontal line with two wavy lines above and below} \right)$$

Second entry:

$$\Psi_2 = \frac{1}{6} \left\{ \text{circle with two horizontal wavy lines and two arrows} \right\} + \left\{ \text{circle with two horizontal wavy lines and two arrows, rotated} \right\}$$

$$\Sigma_2 = \left(\text{horizontal line with two wavy lines and two arrows} \right) + \left(\text{horizontal line with two wavy lines and two arrows, rotated} \right)$$

Third entry:

$$\Psi_3 = \frac{1}{2} \left(\text{circle with four wavy lines} \right)$$

$$\Sigma_3 = \left(\text{horizontal line with four wavy lines} \right)$$

$$+ \left(\text{circle with four wavy lines, rotated} \right) + \left(\text{circle with four wavy lines, rotated} \right)$$

Fourth entry:

$$\Psi_4 = \frac{1}{6} \left(\text{circle with four wavy lines and two arrows} \right) \quad \Sigma_4 = \left(\text{horizontal line with four wavy lines and two arrows} \right)$$

Fifth entry:

$$\Psi_5 = \frac{1}{8} \left\{ \text{circle with four wavy lines and two arrows} \right\} + \left\{ \text{circle with four wavy lines and two arrows, rotated} \right\} + \dots$$

$$\Sigma_5 = \left(\text{horizontal line with four wavy lines and two arrows} \right) + \left(\text{horizontal line with four wavy lines and two arrows, rotated} \right) + \dots$$

As for the approximation associated with every entry the conventions are the same as they are for **psi_p**:

- 0 - do not include
- 1 - both G and W are local, W is static (no dependence on frequency)
- 2 - both G and W are local, W is dynamic
- 3 - W is local and static; G is in band representation
- 4 - W is local and dynamic. G is in band representation
- 5 - both G and W are in full (bands + PB) basis, W is static
- 6 - both G and W are in full (bands + PB) basis, W is dynamic

This part of the code is under development. Not all above branches are available. Recently tested options for the variable **psi_p** are the following: '60000', '66000', '40000', '20000'. For the details about latest developments please contact the author (Andrey Kutepov, e-mail: andreykutepov65@gmail.com).

15.2 psi2_sig

Similar to the variable **psi_sig**, but is relevant during BSP iterations.

15.3 iter_sigma_gwg

Controls the evaluation of the correction to the self energy during **iter_bsp** iterations. If **iter_sigma_gwg** ≤ 1 then the variable **psi2_sig** has similar meaning as the variable **psi_sig**. However, if **iter_sigma_gwg** > 1 then the second order diagram in the first entry is replaced with an infinite sets of diagrams:

$$\Delta\Sigma = \text{diagram}_1 + \text{diagram}_2 + \text{diagram}_3 + \dots$$

In this case **iter_sigma_gwg** sets up the number of iterations in infinite summations for the self energy. The rest of entries are the same as for **psi_sig**.

16 Group VERTEX_XI

16.1 iter_ladder_xi

Defines the number of iterations in solving the Bethe-Salpeter equation for vertex to be used in charge and/or spin susceptibilities. Susceptibilities can be evaluated after finishing every sc approach. Opposite to the case of polarizability, where we can enforce specific set of diagrams, the set of diagrams used in calculating the susceptibilities is fixed and it is defined by the sc approximation after which they are being evaluated.

16.2 q_suscept

In case if the susceptibility has to be calculated only for one q-point, this variable indicates which q-point should be used for that.

16.3 chi_cmp

Defines the components of susceptibility (in vector notations) to be evaluated. Three digits correspond to the following components: "00" (charge); XY; ZZ. 0 means no evaluation, 1 means to be evaluated.

16.4 vrt_x_appr

Three digits define (1 or 2==yes;0==no) if we calculate correspondingly the uniform, static, or full dynamic susceptibilities. 1 means for one q-point, 2 means along the path in q-space (not all branches are implemented yet).

16.5 nrax_chi

Number of points in real frequency mesh in output files for susceptibilities. The mesh is quadratic with upper limit as defined below.

16.6 freq_chi

Upper limit (R_y) for real frequency mesh in output files for susceptibilities.

16.7 theta_bse

Under development.

17 Group OPTICS

17.1 opt_mode

Under development.

17.2 opt_loc fld

Logical variable. Defines whether the local field effects are included in the optics or not.

18 Group TAU MESH

18.1 n_tau

Number of points on TAU-mesh (inhomogeneous).

18.2 exp_tau_gw

Exponent α , which shows how much denser the τ -mesh is when it is close to 0 or β . Following the formula $\tau = x^\alpha$ with x being homogeneous mesh.

19 Group OMEGA MESH

19.1 n_omega_exa

Number of fermionic frequencies on the dense part of the mesh (all Matsubara's frequencies are taken into account on this part of the mesh).

19.2 n_omega_geom

Number of frequency points on the part of the mesh where the intervals between points are increased according to the geometric progression. See the variable

omega_geom below which defines the upper limit of this part of the mesh.

19.3 n_omega_asy

Number of frequency points on the first asymptotic part of the ω -mesh. See the variable **omega_max** below which defines the upper limit of this part of the mesh.

19.4 omega_geom

Defines the upper limit of the geometric progression part of the ω -mesh, (R_y).

19.5 omega_max

Defines the upper limit of the first asymptotic part of the ω -mesh, (R_y).

19.6 omega_last_step

20 Group NU MESH

20.1 n_nu_exa

Number of fermionic frequencies on the dense part of the mesh (all Matsubara's frequencies are taken into account on this part of the mesh).

20.2 n_nu_geom

Number of frequency points on the part of the mesh where the intervals between points are increased according to the geometric progression. See the variable **nu_geom** below which defines the upper limit of this part of the mesh.

20.3 n_nu_asy

Number of frequency points on the first asymptotic part of the ν -mesh. See the variable **nu_max** below which defines the upper limit of this part of the mesh.

20.4 nu_geom

Defines the upper limit of the geometric progression part of the ν -mesh, (Ry).

20.5 nu_max

Defines the upper limit of the first asymptotic part of the ν -mesh, (Ry).

20.6 nu_last_step

21 Group ATOMIC DATA

Now we input the information for every inequivalent atom in the unit cell.

21.1 magn_shift

External magnetic field (Ry/ μ_B) for initial spin splitting. Associated with the variable **iter_h_ext**. Manual update of this variable should be of the following format: 'magn_shift 2 0.5' where 2 stands for the sort of atom, and 0.5 for the shift. The string, similar to the one above, should be used for each sort of atom which is to be updated.

21.2 smt_wgt

Weight associated with each sort of atoms. Default algorithm finds the maximal MT radii compatible with the geometry. The weights specify the radii when there is a competition between the sorts. Default is given in the subroutine 'smt.default.F'. Manual update of this variable should be of the following format: 'smt_wgt 2 2.1' where 2 stands for the sort of atom, and 2.1 for its weight.

21.3 smt_scale

Allows additional flexibility in choosing the MT radii. The MT radii $S_0(t)$ obtained after automatic procedure (which uses 'smt_wgt') can be scaled according to $S(t)=\text{smt_scale}(t) * S_0(t)$, where 't' stands for the sort.

Default value is 1.

This variable also allows to enforce a specific radius by providing it with negative sign. Manual update of this variable should be of the following format: 'smt_scale 2 0.9' where 2 stands for the sort of atom, and 2.1 for the scale.

21.4 r1

First step on the radial mesh. Defines the **h**-parameter in the equation for the radial mesh inside MT sphere $r_i = S \frac{e^{hi}-1}{e^{hN}-1}$, $i = 0, N$, where S-MT radius, N- number of points (N=**nrad**, see next input variable). The **h**-parameter is defined from the condition **r1**= r_1 .

21.5 nrad

Number of points on radial mesh.

21.6 lmb

Maximal L for LAPW basis.

21.7 lmpb

Maximal L for product basis.

21.8 lmv

Maximal L for electronic density.

21.9 low_augm

Specifies the augmentation type for the orbitals with L lower or equal to the **lmax_nble_add**. Default is 'LAPW' for **acc_level** equal to 0, 'APW+lo' for **acc_level** equal to 1, and it is 'APW+lo+HDLO' otherwise.

21.10 high_augm

Specifies the augmentation type for the orbitals with L higher than **lmax_nble_add**. Default is 'APW' for **acc_level** equal to 0, and it is 'LAPW' otherwise.

21.11 `lmax_ntle_add`

Highest L where `ntle_add` is effective. Default value is equal to highest atomic L with nonzero occupations.

21.12 `ntle_add`

Number of additional HDELO orbitals for $L=0$, `lmax_ntle_add`.

Default value depends on the accuracy level: it is equal to `acc_level` if only DFT or HF calculation is performed; it is equal to $2*\text{acc_level}$ if other type of calculation is performed;

Now for every L and principal quantum number we input the parameters `augm`, `atocc`, `ptnl`, `corr`, `idmd`.

21.13 `augm`

Overrides any other definitions of the type of the functions in MT spheres. Should be given after 3 integers (E,L,isor) specifying the orbital. The value itself can be one of the following list:

- `APW+lo+HDLO`
- `LAPW+HDLO`
- `APW+lo`
- `HDELO`
- `LAPW`
- `HELO`
- `APW`

21.14 `atocc`

Initial occupancy of the orbital.

21.15 `ptnl`

Potential parameter. $P = N + 1/2 - \arctan(D)/\pi$, where N is the principal quantum number of the orbital, and D is the logarithmic derivative.

21.16 `corr`

Specifies whether the orbital is included in the reduced LAPW basis set (=L for both ϕ and $\dot{\phi}$; =A for only ϕ) or not included (=N).

21.17 `idmd`

Defines the way we solve the radial equations:

- `IDMD = 0` - for the given energy (center of the band);
- `IDMD = 1` - for the given potential parameter `ptnl` (see above);
- `IDMD = 2` - $\Phi(\text{Smt})=0$; $\Phi'(\text{Smt})\neq 0$ - bound sol-n, but not deriv;
- `IDMD = 3` - $\Phi(\text{Smt})=\Phi'(\text{Smt})=0$ - totally bound sol-n (for deep E);
- `IDMD = 4` - for the given energy (equal to the Chemical Potential).

21.18 `mode_pb`

If equal to 1, then r^l is used as one of the product basis functions for each l and atom. The rest of product basis functions are made orthogonal to this function.

21.19 `lmpb_c`

Maximal L for the vertex product basis. Default: -1 if calculation is without vertex corrections, and is equal to LMPB if vertex corrections are included.