ecalj — Usage (feb2013)
xxxxxx under construction xxxxxx

Takao Kotani

February 26, 2014

Contents

1 Introduction

The `ecalj` is a package for DFT/GW. Especially, we can perform the quasi-particle self-consistent GW (QSGW) calculation based on the PMT method (=Linearized APW+MTO method).

After you read EcaljGetstarted.pdf, read this document (not completed yet...) In this document, we will explain method, input files (ctrl and GWinput), output files, and how to read output. In addition, we explain how to calculate physical quantities based on the LDA/GGA or on the QSGW.

2 Method

The PMT method and the QSGW method are the basis of the `ecalj` package.

— I will describe minimum formulas here. Not yet.—

2.1 the PMT method

2.2 the PMT-QSGW method

3 ctrl file

A ctrl file is usually generated from a ctrls file by the `ctrlgenM1.py` (a crystal structure file is not "ctrl" but "ctrls".). It contains self explanation. Here we give complementary explanations to it. Let us Look into a ctrl file. This is a head part of `ctrl.cu` generated by ctrlgenM1.py:

```
### This is generated by ctrlgenM1.py from ctrls
### For tokens, See http://titus.phy.qub.ac.uk/packages/LMTO/tokens.html.
### Do lmf --input to see all effective category and token ###
### It will be not so difficult to edit ctrlge.py for your purpose ###
VERS    LM=7 FP=7        # version check. Fixed.
IO      SHOW=T VERBOS=35 TIM=2,2
             # SHOW=T shows readin data (and default setting at the begining of
console output)
             # It is useful to check ctrl is read in correctly or not
               (equivalent with --show option).
             # larger VERBOSE gives more detailed console output.
SYMGRP find  # 'find' evaluate space-group symmetry automatically.
             # Usually 'find is OK', but lmf may use lower symmetry
...
```

Note that `#` means comment lines. We can also use lines starting from `% const ...` to define variables and set constant.

We see "categories" such as `VERS`, `IO`, and so on. The beginning of categories are starting from the first column. Under categories, we have "tokens" such as `VERBOSE`. Thus we specify full name of token `VERBOSE` under category `IO` as `IO_VERBOSE`.

- `IO_TIM` is for debugging. It shows which subroutines are called and so on. Bigger number shows deeper subroutines.

- `SYMGRP` is a category without token under it; we set generators of space group (See explanation in previous paragraph). When we set `find`, it automatically calculate symmetry of crystal lattice. If we like to enforce symmetry, set some of generators which are shown by lmchk.

- We see `ctrls` is embedded in the `ctrl` by `ctrlgenM1.py`.

```
... (skip) ...
% const  da=0 alat=6.798
STRUC   ALAT={alat} DALAT={da}
        PLAT=  0.0 0.5 0.5  0.5 0.0 0.5   0.5 0.5 0.0
        NL=4 NBAS= 1  NSPEC=1
SITE    ATOM=Cu POS=0 0 0
... (skip) ...
```

  NL, NBAS(number of SITE) and NSPEC(number of SPEC) are automatically added by ctrlgenM1.py. It is possible to deform unit cell by adding optional tokens (it is possible to rotate PLAT for magnetic anisotropy calculation). See `http://titus.phy.qub.ac.uk/packages/LMTO/tokens.html#STRUCcat`. For new calculations, it is better to find some examples first.

- **SITE** category: As for MT sites, we have two categories. (1)SPEC(species) and (2)SITE(specify centers of atoms(species) in primitive cell). As for SPEC, we specify MTs(radius, Z, MTOs on it) appeared in the cell. These are defined subtokens under SPEC_ATOM=foobar (we have multiple SPEC_ATOM=foobar).

  Then we place these MTs at SITE sections in the cell. At SITE, we specify atomic sites (What SPEC_ATOM is placed to positions by POS) in a primitive cell. We set `POS=` by direct form (Cartesian) but with the unit of `ALAT+DLAT`. Total number of SITE (number of tokens SITE_ATOM) is the number of atoms in the primitive cell. Setting `POS=` under SITE_ATOM=foobar means that we place MT named as foobar defined in SPEC_ATOM=foobar. In addition, we can set SITE_ATOM_RELAX, if you like to find relaxed structure (we simultaneously set DYN category) in LDA. As for relaxation, see `LaGaO_relax/ctrl.lagao` example, and read `http://titus.phy.qub.ac.uk/packages/LMTO/tokens.html#DYNcat`.

  The SITE_ATOM=foobar (with same foobar with different POS) are not necessarily equivalent with respect to the space group operation of a system. Thus `SITE_ATOM=foobar` are divided into "classes" which are connected by the operation. The lmf automatically judge "classes" (see also info by lmchk). Thus not need to specify it, but it may be better to check it. A sample is `lmchk lagao` at `~/ecalj/lm7K/TESTsamples/LaGaO_relax`

- **SPEC** category: In ctrls, we have not yet specified contents of SPEC; we have just given default symbols or only Z= when we use non-default names (shown by ctrlgenM1.py –showatomlist). The command `ctrlgenM1.py` adds default SPEC sections.

  We have some `SPEC_ATOM`, under which we give subtokens such as `SPEC_ATOM_R`(MT radius), `SPEC_ATOM_Z`(nucleus charge), cutoff parameters of angular momentum, and so on. These `SPEC_ATOM` is refereed to in SITE.

  An example of SPEC category is

```
SPEC
    ATOM=Fe Z=26 R=1.70
      KMXA={kmxa}  LMX=3 LMXA=4 NMCORE=1
      PZ=0,3.9,4.5
      EH=-1 -1 -1 -1  RSMH=0.85 0.85 0.85 0.85
      EH2=-2 -2 -2   RSMH2=0.85 0.85 0.85
      MMOM=0 0 2 0

    ATOM=... (then the similar block of ATOM= are repeated.)
       ...
```

  Under the token `ATOM=Fe`, we have subtokens `SPEC_ATOM_Z`,`SPEC_ATOM_R`, and so on.

  Subtokens Z= is the nucleus charge and R= MT radius. Note that Fe is just a name to distinguish MT sphere in the cell. If you set SPEC_ATOM_Z=27, it is recognized as Co (since Z=27). `LMX=3` is the maximum l of MTOs. Thus maximum l of MTO is l=3. The maximum of l to expand electron density and potential within MT is LMXA (in contrast to usual LAPW), we can use quite small LMXA such as LMXA=4. NMCORE=1 means we calculate core density without non magnetically-polarization. This can reduce computational confusion.

  PZ is to set local orbital (if not, no local orbitals). EH and RSMH are to specify first set of MTOs.(We can check how local orbitals are set by lmfa explained in the next section). EH2 and RSMH2 are to specify second set of MTOs.

  After PZ=, we have three numbers. These are numbers for s,p,d,f,g,... channels. Zero means not exist. You can use space or comma(,) as delimiter. Here not only the integer part of principle quantum number, but also the fractional part should be supplied (If PZ=0,3,4, it does not work.) Now PZ=3.9 for p and PZ=4.5 for d. This means we use local orbital for 3p, and local orbital for 4d (fractional parts (continuous principle quantum number) are large ~ 0.9 for core like orbital, and smaller for extended orbital ~ 0.3 or something. See Logarithmic Derivative Parameters

at `http://titus.phy.qub.ac.uk/packages/LMTO/lmto.html`). This is a little confusing, thus we will explain this in appendix. See Sec.**??**.

EH(damping factor), and RSMH (where the smooth Hankel function bent) determines MTOs (or its envelope function as a smooth Hankel function). We now set four numbers for them. Thus we set MTOs s,p,d,f with EH=-1 and RSMH=0.85. Our current test shows that RSMH is one half of R (that is, 0.85=1.70/2, but minimum RSMH is 0.5) and not need to be dependent on s,p,d,f. (If LMX=2, s,p,d are allowed and no f MTOs.) EH is -1; not need to change except test purpose. In a similar manner, EH2 and RSMH2 for second set of MTOs are given. Just three numbers means these for s,p,d.

MMOM=s,p,d,f... gives initial condition of magnetic moment in μ_B (number of up-down electron).

In cases such as As, the local orbital given by default ctrl is responsible of rather deep core, and it is not need to be treated as valence electrons. In such a case, we don't need local orbital.

In the case of AntiFerro-II NiO, it contains two NiO in a primitive cell. Thus it is reasonable to have two SPEC_ATOM as Ni1 and Ni2, although subtokens under ATOM=Ni1 and ATOM=Ni2 (e.g. `SPEC_ATOM_EH` for them) are the same except initial condition of magnetic moment of MMOM=s,p,d,f... See example of NiO.

The minimum help of call Category_token_subtoken are listed with minimum explanation with

```
$ lmf --input
```

It gives a long output. But many of them are experimental and not need to manage them. A part of it is

```
Token            Input   cast  (size,min)
-----------------------------------------
   ... ...

    STRUC_ALAT        reqd   r8       1,  1
      Scaling of lattice vectors, in a.u.
   ... ..
```

This is an minimum explanation of it. "reqd" means "required" (no default). r8 means it read with real number, 1,1 means that ALAT=xxx should contain one number minimum (max is also one) (See also STRUC_PLAT, and so on).

There are kinds of examples in ecalj packages. Please look into their ctrls.* and ctrl.* These are in lm7K/TESTsample/* and ecalj/CMDsampls. In addition, ecalj/MATERIALS contain many samples (need a command); see a later subsection.

As for what is shown in `$ lmf --input`, most of important tokens are already described in the ctrl file generated by ctrlgenM1.py. So, we don't need to care many options shown by it.

But we have not yet explained some useful feaures; STRUC category to deform crystal; DYN category for dynamics; LDA+U treatment; Adding background charge; Core-Hole treatments. We will prepare examples for it if requested.

`http://titus.phy.qub.ac.uk/packages/LMTO/tokens.html#STRUCcat`
`http://titus.phy.qub.ac.uk/packages/LMTO/tokens.html#DYNcat`

4 GWinput

(QPNT.chk contains irreducible k point for given n1 n2 n3; KPTin1BZ.gwinit.chk contain all k points in Brillowin Zone). Generally speaking, you don't need to repeat mkGWIN_lmf2 as long as you don't change MTO sections in ctrl file (number of EH,EH2,PZ).

Look into `GWinput`. Because of historical reason, input system is different from ctrl. Each line is independent; "keyword" followed by some of numbers (or on or off for logical switch of keyword). Except such lines, there are tag sandwiched sections such as `<PRODUCT\_BASIS> ... </PRODUCT_BASIS>`, no comment lines in the middle allowed. They are read by read(*,*) (thus spaces cause no problem).

In the QSGW calculation, we have to set some cutoff parameters for self-energy calculations.

- `emax_sigm` is the maximum limit of the self-energy (measured from the Fermi energy). I think 2.5 $\sim$ 5Ry is reasonable choice. But in cases, small `emax_sigm` can give poor dispersion curve (slightly unnatural behavior) because of sudden cutoff by emax_sigm. However, we like to use smaller value to reduce computational time. That is, larger is better, but expensive. Generally speaking, accuracy less than 0.1eV (for bandgap) is allowance of current method. Probably, it may be not impossible to have better accuracy, but it may ask us to repeat many calculations with changing conditions to confirm stability.

- `0.100000D-02 ! =tolopt` controls a number of Product basis to expand the Coulomb interaction. (The product basis is to expand the Coulomb interaction is different from the basis to expand eigenfunction.) In our experience, `0.100000D-02` (=0.001) is not so bad. If you like to reduce computational time use 0.01 or so, but a little dangerous in cases. With 0.0001, we can check stability on it.

- `QGcut_psi` is a little (usually 0.5 or so) larger than `QpGcut_cou`. It becomes accurate if we use large `QpGcut_cou`. But it enlarge size of IPW(interstitial plane wave) part of Mixed product basis. For test, try 2.7, 3.2, 3.7 for `QGcut_cou` (and add 0.5 or 1 for `QGcut_psi`). Larger one is expensive.

- `dw` and `omg_c` specify real space bins which we accumulate imaginary part weight of polarization functions. dw is bin width (in Ry) at omega=0, then bin width is twiced at `omg_c`. The bin width is quadratically larger (become rough). If bins are too wide, dielectric function can be less accurate, but results are not necessarily so much affected. For metal, our code can capture Drude weight numerically. We do not need to be so sensitive to the choice of them usually.

- n1n2n3: BZ division for k point integration. We usually take '4 4 4' to '8 8 8' for GaAs. For metal such as Fe, '10 10 10' or more is better.

- lcutmx(atom) is the l cutoff of product basis for atoms in the primitive cell (do lmchk for atom id). In the case of Oxgen, we can usually use lcutmx=2 (need check by the diffence when you use lcutmx=2 or lcutmx=4). Then the computational time is reduced well.

- Other part of product basis section in `<PRODUCT\_BASIS>... </PRODUCT\_BASIS>` is usually not need to be touched. But if you like to calculate big systems with smaller CPU time, or do very accurate calculations, we may need to touch it. It is described elsewhere.

- `<QPNT>` tag is to specify one-shot GW. At which point, do we calculate QPE, not for QSGW. If you set k point in it not on regular mesh point, you have to set 'Any Q on'; but it is expensive. Since QSGW have ability to plot energy band within full BZ, it should be better to do it.

- `<QforEPS>` and `<QforEPSL>` are to specify at which k point do we calculate susceptibility. It is for epsPP_lmfh, eps_lmfh (dielectric functions) and epsPP_chipm (spin susceptibility).

  We need a setting in ctrl file to read sigm file (HAM_SIGP). It is simplified now, and not need to care it so much. As we set RDSIG=12 in defaults, lmf read sigm file and add it to one-body potential as long as sigm.* exist.

  NOTE for old users: We now set `SIGP[MODE=3 EMAX=9999.]` in ctrl file to read self-energy in lmf (or lmf-MPIK). This is because we use very localized MTOs (similar with the Maxloc Wannier). Our test shows reasonable results and this simplify algorithms. In my previous version, we asked you to use `SIGP[MODE=3 EMAX=2.0]` where EMAX is a little (0.5Ry) less than `emax_sigm`. If something strange occurs, try this setting).

In principle, QSGW result should not depended on the choice of XCFUN. However, it can affect slightly. In our tests, it seems slightly better to use vwn (XCFUN=1) for QSGW calculations. (BUT need to check more...)

4.1 How to set local orbitals

```
  As we stated, do "lmfa |grep conf" to check used MTO basis.

  We have to set SPEC_ATOM_PZ=?,?,?
  (they ordered as PZ=s,p,d,f,... ) to set local orbitals.

  lmv7 (originally due to ASA in Stuttgart) uses a special terminology
  "continious principle quantum number for each l", which is just
  relatated to the logalismic derivative of radial funcitons at MT
  boundary. It is defined as
   P= principleQuantumNumber + 0.5-1/pi*atan(r* 1/phi dphi/dr),
  where phi is the radial function for each l.   For example,
   P= n.5 for l=0 of free electron (flat potential) because phi=r^0,
   P= n.25 for l=1 because phi=r^1;
   P= n.147584 for l=2 because phi=r^2; P=, n.102416 n.077979 for l=3,4.
  (Integer part can be changed). See Logarithmic Derivative Parameters in
  http://titus.phy.qub.ac.uk/packages/LMTO/lmto.html#section2

  Its fractioanl part 0.5-atan(1/phi dphi/dr) is closer to unity for
  core like orbital, but closer to zero for extended orbitals.

  Examples of choice:
  Ga p: in this case, choice 0 or choice 2 is recommended.
      We usually use lo for semi-core, or virtually unoccupied level.

     (0)no lo (4p as valence is default treatment without lo.)
        3p core, 4p valence, no lo: default.
        Then we have choice that lo is set to be for 3p,4p,5p.
     (1)3p lo ---> 4p val (when 3p is treated as valence)
        3d semi core, 4d valence
        Set PZ=0,3.9
        (P is not requied to set. *.9 for core like state. It is just an initial condition.)
     (2)5p lo ---> 4p val (PZ>P)
        Set PZ=0,5.5
        5.5 is just simply given by a guess (no method have yet
implemented for
        If 5.2 or something, it may fails
        because of poorness in linear-dependency. We may need to observe
        results should not change so much on the value of PZ.
```

```
     (3xxx)4p lo ---> 5p val (we don't use this usually. this is for test purpose)
        4p lo, 5p valence
        Set PZ=0,4.5 P=0,5.5 (In this case, set P= simultaneously).
        (NOTE: zero for s channel is to use defalut numbers for s)

  Ga d: (in this case, choice 0 or choice 1 is recommended).
     (0)no lo (3d core, 4d valecne, no lo: default.)
          Then we have choice that lo is set to be for 3d,4d,5d.
     (1) 3d lo ---> 4d val  (when 3d is treated as valence)
         Set PZ=0,0,3.9  (P is not required to set)
     (2) 5d lo ---> 4d val  (PZ>P)
         Set PZ=0,0,5.5
     (3xxx) 4d lo ---> 5d val  (this is for test purpose)
         Set PZ=0,0,5.5 P=0,0,4.5
         (NOTE: zero  for s,p are to use defalut numbers )

   If you like to read from atm.ga file instead of rst file(if exist).
   You have to do lmf --rs=1,1,0,0,1, for example. See lmf --help
   Becase rst file keeps the setting of MTO, thus change in ctrl is not
   reflected without the above option to lmf.
============================================================
```

5 MEMO random

These are memo randoms. I have to explain them.
xxx under construction xxxxxxxxxxxxxxx

```
== not meaningful total energy ===
 Total energy shown in QSGW mode in current version is not meaningul.
 (just treat as an indicator to convergence).

== Do we use vwn or gga for QSGW? ===
  In prinicple, QSGW results should not depends on vwn or gga
  (XCFUN=1 or 103 in ctrl). But there is minor dependence, because
   1. frozen core density.
   2. core eigenfunction.
   3. radial basis funcitons
   4. Slight numerical reason
      (This is probably because Sigma-interpolation procedure
       But not exactly figured out yet
       -->affect about 0.02eV as for band gap for GaAs. ).
  In anyway, use vwn (HAM_XCFUN=1) as standard.
  And such technical things affects, 0.05 eV level of error for band gap.

== one show QSGW ==
```

```
  one-shot QSGW may be useful in cases.
  As it contains off-diagonal part, we can resolve band tanglement
  problem in Ge (no band gap).


== Restart calculation in lda ==
 lmf(lmf-MPIK) read rst.* in defaluts.
 rst contains electron density.
 If rst is already converged, it stops after two iteration.
 rst contains atomic positions.
 So, in order to read atomic positions change in ctrl,
 Use options shown in lmf --help.

== Restart calculation in qsgw ==
  To remove mixsigm* (mixing for sigm), maybe required.

== iteration check ===
  First, watch console output of gwsc (do redirect to output file)
  Need to check OK! signs arrayed on 1st columns.

  gwsc iteration is time cosuming,
  So we need to check calculations are normally going on or not.

  Memory inefficiency.
  Set 'KeepEigen off' an 'KeepPPOVL' off.
  In fact, out code is still inefficient for memory usage.

  grep gap llmf ---> minimum gap at mesh point.
  see save.* ,or grep '[xc] ' save.*
  the end of iteration of lmf is shown as x or c.
  (if failed, QPU file.
  dqpu QPU.4run QPU.3run
  As for usual semi-conductor, accuracy abou t0.1 eV is limit of current implementation.
  Set vwn (xcfun=1) looks better (stable) for GW.

  $grep rms lqpe*
  shows
           ...  rmsdel=2.44D-04
           ...  rmsdel=4.91D-03
           ...   rmsdel=2.44D-04
           ...   rmsdel=3.37D-04
  If rsmdel is getteing to be smaller, it is on convergence path.
  (but in magnetic cases, it may give be too good even not yet going to
be converged..., beccause magnetic energy is so small)
```

```
  grep diffe llmf  ---> difference of energies of each iteration.

  ehf (harris energy)
  ehk (Hohenberg kohn energy)

== emax cutoff for APWs. ==
  We can not use so many APWs in current version,
  because of overcompleteness (this is because null vector within MTs),
  In anyway, use pwemax=3 as standard (test it with 4 or 5).
  To avoid failure of calculation, we may use smaller MT radius for
  alkali, and alkali-earth elements.
  In feature, I think we can introduce pseudopotentials for these atoms only.

== Check Used MTO
 Near beginig of console output, what MTO you use is shown as: (GaAs case).
 sugcut:  make orbital-dependent reciprocal vector cutoffs for tol= 1.00E-06
 spec      l    rsm    eh     gmax    last term    cutoff
  Ga       0*   1.13  -1.00   6.579    1.19E-06    1459
  Ga       1*   1.13  -1.00   7.028    1.26E-06    1807
  Ga       2*   1.13  -1.00   7.475    1.09E-06    2109
  Ga       3    1.13  -1.00   7.920    1.06E-06    2637
  Ga       0*   1.13  -2.00   6.579    1.19E-06    1459
  Ga       1*   1.13  -2.00   7.028    1.26E-06    1807
  Ga       2    1.13  -2.00   7.475    1.09E-06    2109
  As       0*   1.18  -1.00   6.300    2.13E-06    1243
  As       1*   1.18  -1.00   6.720    1.26E-06    1471
  As       2*   1.18  -1.00   7.140    1.37E-06    1837
  As       3    1.18  -1.00   7.558    1.05E-06    2229
  As       0*   1.18  -2.00   6.300    2.13E-06    1243
  As       1*   1.18  -2.00   6.720    1.26E-06    1471
  As       2    1.18  -2.00   7.140    1.37E-06    1837

== gwsc cause error stop.
 Have you ever changed MTO setting? Consistent with GWinput?

== QSGW for Fe.
  It is better to use 3p as core. Furthermore, 3d+4d as valence is better.
  Thus we need to set PZ=0,3.9,4.5
  I also got aware that emax_sigm should be large enough (4$\sim$5 Ry)
  to have smooth band dispersion. n1n2n3 can be 10x10x10.

== RSRNGE: enlarge RSRNGE ===
  Use RSRNGE=10 or so (in cases, RARNGE=20 or more is required),
  for large number of k points. Try and enlarge it if it fails with a
  message "Exit -1 rdsigm: Bloch sum deviates more than allowed tolerance (tol=5e-6)".
```

```
  We will have to make it automatic in future.

== Q0P check
   In cases, it is better to use Q0Pchoice=2 instead of default Q0Pchoice=1.
   (For slabs, Q0Pchoice=2 may be better; need check more. In anyway,
    it is problematic to use unbalanced k points for anisotropic cell).
    See Copmuter Physics Comm. 176(2007)1-13).

=== When calculation in LDA level fails ===
when calculation fails in LDA level.
  (1) smaller MT
  (2) fewer PW. smaller pwemax.
  (3) core as semicore.


=== LDA+U ===
not yet written...

=== MAE by rotating crystal ===
(we have a sample at lm7K/TESTsmaples/MAEtest/, but only in GGA/LDA).

=== spin wave ===
J calculation.


====
If not stable convergence in gwsc, try to set
mixbeta 0.5
(and/or mixpriorit 3 or something)
at the begining of sigma.


=======
cleargw (directory):
This command clean up up intermediate files under (directory).
This recursively into deeper level. Be careful, or edit it.
I use it as '>cleargw .'



-----------------------------------
Magnetic moment within MTs are shown as
-----------------------------------
 charges:       old           new
 smooth      17.240314     17.240740   ...
```

```
 mmom         0.000024     -0.000010
 site    1    6.207135      6.206590
 mmom         1.062276      1.062991  <--- here
 site    2    6.207115      6.206834
 mmom        -1.062323     -1.062958  <--- here
 site    3    1.172718      1.172918
 mmom         0.000011     -0.000011
 site    4    1.172718      1.172918
 mmom         0.000011     -0.000011
In this case, MTsite1 has 1.062991 and MTsite2 has -1.062958.
>grep 'lin mix' -A30 llmf
can take out this message (if console output is in llmf).


-----------------------------
ORBITAL MOMENT in pertubation:
-----------------------------
Try
>lmf nio --rs=1,0 -vso=1 --quit=band >llmf
After converged, try
>grep IORBTM -A20 llmf
Then llmf shows shows orbital moment in first order perturbation.
(Here --rs=1,0 read rst.* file but not change it. See >lmf --help.
--quit=band means quit just after band calculation.)
```

6 lmf –help

lmf –help show option of –rs=(five numbers); this let lmf know how to read atm.* file which is the initial atom file by lmfa.

7 others

xxxxxxxxxxxxxxxxxxxxxxxxx MEMO xxxxxxxxxxxxxxxxxxxxxxxxx

Co on MgO slab:

```
ecalj/MATERIALS/ctrl.mgoco
------from here ------------------
STRUC
     ALAT=1.88972687777
     PLAT=       3.00591       0.00000       0.00000000000
                 0.00000       3.00591       0.00000000000
                 0.00000       0.00000      16.00000000000
```

```
SITE  ATOM=Mg       POS=   0.0000000   0.0000000   0.0000000  RELAX= 0 0 0
      ATOM=Mg       POS=   1.5029550   1.5029550   2.1723171  RELAX= 0 0 0
      ATOM=O        POS=   1.5029550   1.5029550   0.0000000  RELAX= 0 0 0
      ATOM=O        POS=   0.0000000   0.0000000   2.1032371  RELAX= 0 0 0
      ATOM=Co       POS=   0.0000000   0.0000000   4.1898024  RELAX= 0 0 1
      ATOM=Co       POS=   1.5029550   1.5029550   5.2139861  RELAX= 0 0 1
------to here ------------------


== EPS mode,
  Check Im part of chi0 is smoothly damping at high energy (typically
  1Ry or larger enengy range). If there is some large Im part remains,
  something strange (usually due to orthogonality problem of
  eigenfunctions when you set low q).

   Related source codes are in ecalj/lm7K/ .
   A command ecalj/lm7K/ctrlgenM1.py can generate 'standard input file (ctrl file)'
   just from a given crystal structure file called as ctrls file.
   Binaries are lmf and lmf-MPIK (MPI k-parelell verion).

=== MAE by rotating crystal ===
--(we have a sample at
++(we have a sample at lm7K/TESTsmaples/MAEtest/, but only in GGA/LDA).


=== spin wave ===
  J calculation.


xxxxxxxxxxxxxxxx
   Recently, I renewed some part of algolism of GW/QSGW calculations
   (some ideas are taken from from PRB.81,125102(2010)
    and Copmuter Physics Comm. 176(2007)1-13).
   ---> this is better than old versions; speed, memory (file size),
   and accuracy for anisortopic systems.
   For comparison, you can use old version in .git (gitk --all and check it out).
   See Copmuter Physics Comm. 176(2007)1-13).

xxxxxxxxxxxxxxxxx
-----------------------------------------------------------
-- QSGW: convergence check sheet.
-----------------------------------------------------------
--1. Basis to expand eigenfuncitons.
--   As for APW, try pwemax=3, 4, 5.
--   In principle, bigger is better.
```

```
--   Local orbitals for semicore were requied (for Fe, and so on).
--2. Re-expand eigenfuncions (QpGcut_psi for IPW part)
--   In principle, bigger is better.
--3. Mixed product basis.  QpGcut_cou for IPW part.
--   <ProductBasis> section for PB part.
--4. number of k points, Q0Pchoice(irrelevant but speed up).
--
--5. omg,dw  (bins to accumulate imaginary part).
--
--6. emax_sigm (use 2Ry to 6Ry. And see stability. In priniciple, bigger is better.)
--
--7. Do GW with XCFUN=1 or 103 (VWN or GGA)?
--   In priniple, bigger emax_sigm reduce dependence on them.
--   But, not easy. We may take the difference as allowance of error.
```
