

ecalj method (detail for developments)

<https://github.com/tkotani/ecalj>

August 6, 2025

Abstract

xxxxxxxxxxx under construction xxxxxxxxxxxxxxxx Methodological Details on the ecalj package. Refs. [?] and [?].

Contents

•Reference

•Index of I/O files, shel scripts and executions.

1 MEMO1xxx

xxxxxxxxxxxxxxxxxxxxxxxxx MEMO xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Co on MgO slab:

ecalj/MATERIALS/ctrl.mgoco

-----from here -----

STRUC

ALAT=1.8897268777

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 -----

=== MAE by rotating crystal ===

--(we have a sample at

++(we have a sample at lm7K/TESTsmamples/MAEtest/, but only in GGA/LDA).

=== spin wave ===

J calculation.

xxxxxxxxxxxxxxxxxxxxx

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).

xxxxxxxxxxxxxxxxxxxxx

-- 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, QOPchoice(irrelevant but speed up).

--

--5. omg,dw (bins to accumulate imaginary part).

--

--6. `emax_sigm` (use 2Ry to 6Ry. And see stability. In principle, bigger is better.)
--
--7. Do GW with `XCFUN=1` or 103 (VWN or GGA)?
-- In principle, bigger `emax_sigm` reduce dependence on them.
-- But, not easy. We may take the difference as allowance of error.

2 Data structure and contents of files

2.1 Basic i/o files to start GW

We need basic data set for the system and eigenfunctions from which we start GW calculations. Note that **lmchk** show crystal structure informations.

1. Crystal structure. unit of lattice primitive cell reciprocal cell atomic position point group symmetry
core configuration.
 2. $q+G$ vectors.
 3. Eigenfunctions and eigenvalues. cutoffs of
- **LATTC** : Generated by **echo 0|lmfgw**. This contains the information of primitive translation vectors, **lmxa** and **konf**. A sample is,

```

10.26d0      ! alat      = lattice constant in a.u.
0d0 .5d0 .5d0 ! plat(1:3,1)= 1st primitive translation vector in alat.
.5d0 .0d0 .5d0 ! plat(1:3,2)= 2nd ...
.5d0 .5d0 .0d0 ! plat(1:3,3)= 3rd ...
-1d10 ! QpGcut_psi = maximum of |q+G| in a.u.
-----
2   4 ! nbas lmxax (max l for augmentation)
-----
-- ibas lmx konf(s) konf(p) konf(d)... ----
   1   4   3   3   3   4   5
   2   4   3   3   3   4   5

```

where negative **QpGcut_psi** in the 5th line means dummy. True **QpGcut_psi** is given in **GWIN0**. After "-- ibas lmx konf(s) konf(p) konf(d)... ----", you see numbers "3 3 3 4 5" for **ibas=1**. These mean *3s, 3p, 3d, 4f, 5g*, which specify the lowest principle quantum numbers of valence electrons. In other words, cores are *1s, 2s, 2p* for **ibas=1** in this case.

- **SYMOPS** : Generated by **echo 0|lmfgw**. The point group operations. It is written through

```

open(file='SYMOPS')
write(ifi,*) ngrp
do ig = 1, ngrp
  write(ifi,*) ig
  do i=1,3
    write(ifi,"(3d24.16)") symops(i,1:3,ig)
  enddo
enddo
close(ifi)

```

- **CLASS** : Generated by **echo 0|lmfgw**. An example is;

```

1  1
2  1
3  2
4  2

```

The first numbers of each line are atomic-site number in the primitive cell. (It should be the same as the line number). The second numbers of each line are atomic classes for each atomic-site.

- **NLAindx** : Generated by **echo 0|lmfgw**.

This file contains indexes (p_{valence}, l, a) for orbitals in the MT. (p_{valence} is radial function index, a is atomic site index). Eigenfunctions are expanded in this order. Usually we use **LMXA=4**, thus l takes from 0 through 4. An example is (two atoms in a cell);

```

NOTE: p=1,2,3 corresponds to phi,phidot,pz(local orbital)
p  l  a  ilma  label(unused)
----NLAindx start-----
105
1  0  1  0  4S_p
1  1  1  1  4p_p
1  2  1  4  4d_p
1  3  1  9  4f_p
1  4  1  16 5g_p
1  0  2  25 4S_p
1  1  2  26 4p_p
1  2  2  29 4d_p
1  3  2  34 4f_p
1  4  2  41 5g_p
2  0  1  50 4S_d
2  1  1  51 4p_d
2  2  1  54 4d_d
2  3  1  59 4f_d

```

2	4	1	66	5g_d
2	0	2	75	4s_d
2	1	2	76	4p_d
2	2	2	79	4d_d
2	3	2	84	4f_d
2	4	2	91	5g_d
3	2	1	100	3d_l

(last labels, 4s_p... are just for memo. p,d,l means radial function index 1,2,3 (first column n).)

- **ldima** : Generated by **echo 0|lmfgw**. Number of $LMXA^{**2}$ for atomic sites. (this is used only from **hqpe_sc**—QSGW mode; this information is also in other files such as **LATTC**; we currently only allow all $LMXA$ should be the same).

- **ves*** : Generated by **echo 0|lmfgw**. Electro static potential **lmf** or **echo 1|lmfgw**. (only for the bandoffset calculations.)

- **rhoMT*** : Radial electron density within MT **echo 1|lmfgw**.

(used only for spin susceptibility mode in **fpgw/main/hbasfp0.m.F**—**fpgw/gwsr/basnfp.F**).

- **QGpsi, QGcou** : generated by **echo 1|qg4gw**.

IPW for psi and for coulomb interaction. See the last part of **lqg4gw**. This is written by **fpgw/main/qg4gw.F**—**gwsr/mkqg.F**. Search file handles of them, **ifiqg** and **ifigqc** in **mkqg.F**

mkqg.F; Let us search **ifiqg**.

This routine generates all required q points and G for q by **QpGcut_psi, QpGcut_cou**.

```

-----
      write(ifiqg ) nqnum,ngpmx,QpGcut_psi,nqbz,nqi,imx,nqibz
      write(ifigqc) nqnum,ngcmx,QpGcut_cou,nqbz,nqi,imxc

      allocate( ngvecprev(-imx:imx,-imx:imx,-imx:imx) )      !mar2012takao
      allocate( ngveccrev(-imxc:imxc,-imxc:imxc,-imxc:imxc) ) !mar2012takao
      ...
      do iq = 1, nqnum
        q = qq(1:3,iq)
        ...
        write (ifiqg) q, ngp, irr(iq)
        write (ifiqg) ngvecp,ngvecprev !ngvecprev is added on mar2012takao
        write (ifigqc) q, ngc
        write (ifigqc) ngvecc,ngveccrev
      enddo
      write(6,*) ' --- Max number of G for psi =',ngpmx
      write(6,*) ' --- Max number of G for Cou =',ngcmx
-----

```

NOTE:

```

      ngvecp(1:3,1:ngp) : G vector for phi, integer qlat unit
      ngvecc(1:3,1:ngp) : G vector for cou, integer qlat unit
      ngvecprev(n1,n2,n3) : inversion table
      ngveccrev(n1,n2,n3) : inversion table
      nqnum:    total number of q points.
      nqbz:     number of q point in the BZ
      nqibz:    number of irreducible q points of regular mesh (read from QIBZ).
      nqi:      number of irreducible points including offset Gamma points.
                We calculate eigenfunction and Vxc only for these points.
      irr(iq):  irreducible point or not. For irr=1, calculate eigenfunctions.
      imx:      to allocate ngvecprev as follows.
-----
      alat :    unit of length (in a.u.).
      plat(3,3): primitive lattice vectors in the unit of alat
      qlat(3,3): invese lattice vector of plat. \delta_ij = sum(plat(:,i)*qlat(:,j))
-----
      q+g(1:3,iq,ig) = 2*pi/alat*( q(1:3,iq)+sum(qlat(1:3,:)*(ngvecp(:,ig))) )

```

```

,where iq=1,nqnum and ig=1,ngp(iq).
-----
Inverse mapping is
ngvecp(1:3,ig)= ngvecprev(ngvecp(1,ig),ngvecp(2,ig),ngvecp(3,ig))

• DATA4GW_V2 : Crystal structures and so.
Generatage by lmf2gw, and read by rdata4gw_v2.
-----
& nsp,      ! =1 or 2, corresponding to para or ferro.
& nbas,     ! Number of atom in the primitive cell
& nclass,   ! Dummy. =natom=nbas in the current code.

& nrmx,     ! = maxval(nr(1:nclass)) Maximum number of nr
& ncoremx,  ! = maxval(ncore(1:nclass))
c
& lmxamx,   ! = maxval(lmxa(1:nclass))
& ngpmx,    ! Maximum number of G vector.
& nband,    ! Number of bands given by GWINO
& ldim,     ! = sum ( (lmxa(1:nbas)+1)**2 )
& ldim2,    ! = total number of augmentation functions nRlm (total of NLAindx).
& nphi,     ! = total number of augmentation functions nRl
& nqbze     ! = nqbz*(1+nq0i). Number of q points given by qg4gw

nindx(ldim2),lindx(ldim2),ibasindx(ldim2): nla index shown in NLAindx

-----
Cell DATA:
alat: unit
plat: primitive vector
qbze(3,nqbze): q vector.
efermi: fermi energy

-----
ATOMIC DATA:
iclass(ibas)=ibas: dummy
aa(nclass),bb(nclass): specify radial mesh. r(i)=b(exp((i-1)*a)-1)
bas(3,nbas): atomic position
lmxa_d(nclass): LMXA (in the current version. all LMXA should be the same.
nr(nclass): radial mesh
konf(0:lmxa): principle quantum number of valence electron.
ncore_d(nclass): number of cores.

CORE:
ec_d(ncoremx, nclass, nsp), : core eigenvalue
gx_d (nrmx, 0:lmxamx, nphimx, nclass,nsp): valence radial function
gcore_d(nrmx, ncoremx, nclass,nsp) core radial funciton.

VAL:
evl_d (nbandmx, nqbze, nsp) : valence eigenvalue
vxclda(nbandmx, nqbze, nsp): LDA Vxc

-----
Coefficients:
cphi(ldim2,nbandmx,nsp,nqbze): Coefficient of eigenfunciton within MTs. See NLAindx.
geig(1:ngp,1:nbandmx,nsp,nqbze): Coefficient of eigenfunciton for IPW exp(i q+G r).

```

```

This is a head part of lmf2gw
Cr--- From DATA4GW_V2 -----
Cr  nbas          : the number atom in the primitive cell
c
Cr  alat          : lattice constant in a.u.
Cr  plat(1:3,1)   : 1st primitive translation vector in the unit of alat
Cr  plat(1:3,2)   : 2nd primitive translation vector
Cr  plat(1:3,3)   : 3rd primitive translation vector
c
c-- eigenfunctions for all q points.
Cr  evl (1:nband)      : eigenvalue
Cr  cphi(1:ldim2,1:nband) : the coefficients of eigenfunction for phi(augmentation wave)
Cr  geig(1:ngp, 1:nband) : the coefficients of eigenfunction for IPW
Cr
Cr  nindx  (1:ldim2) : n index (phi=1 phidot=2 localorbital=3)
Cr  lindx  (1:ldim2) : l index
Cr  ibasindx(1:ldim2) : ibas index . These are used to re-ordering cphi.
Cr                      : mindx(1:ldim2) is generated under the assumption that
Cr                      m=-1:1 is successively ordered.
Cr
Cr  vxclda (1:nband) : lda XC potential <psi|Vxc(n)|psi> for each eigenfunctions
c
c-- Atomic data for all the atom in the cell, ibas=1,nbas
Cr  Z:
Cr  nr a,b; mesh is r(i)=b*(exp(a(i-1))-1)
Cr  ncore: number of core.
Cr  ec(1:ncore)
Cr  gx : radial wave function phi.

lmf2gw: CphiGeig write(ifcg) cphi(1:ldim2,1:nbandmx)
write(ifcg) ngp write(ifcg) geig(1:ngp,1:nbandmx)
VXCFP.chk Eigenvalue and Vxc check. only for check

```

3 spin susceptibility (old memo)

3.1 transverse spin susceptibility χ^{+-} calculation

[I changed sign of $\chi^!$ (July2007); This definition may be different from other text book. Be careful] The non-interacting transverse spin susceptibility $\chi^{0+-}(\mathbf{r}, \mathbf{r}', t - t')$ is given as

$$\chi^{0+-}(\mathbf{r}, \mathbf{r}', t - t') = -i\langle T(S_+(\mathbf{r}, t)S_-(\mathbf{r}', t')) \rangle = -iG_\uparrow(\mathbf{r}, \mathbf{r}', t - t')G_\downarrow(\mathbf{r}', \mathbf{r}, t' - t) \quad (1)$$

for non-interacting system (Lindhard-like polarization function). In ω space, this reduced to

$$\begin{aligned} \chi^{0+-}(\mathbf{r}, \mathbf{r}', \omega) &= \sum_{n\downarrow}^{\text{occ}} \sum_{n'\uparrow}^{\text{unocc}} \frac{\Psi_{n\downarrow}^*(\mathbf{r})\Psi_{n'\uparrow}(\mathbf{r})\Psi_{n'\uparrow}^*(\mathbf{r}')\Psi_{n\downarrow}(\mathbf{r}')}{\omega - (\epsilon_{n'\uparrow} - \epsilon_{n\downarrow}) + i\delta} + \sum_{n\downarrow}^{\text{unocc}} \sum_{n'\uparrow}^{\text{occ}} \frac{\Psi_{n\downarrow}^*(\mathbf{r})\Psi_{n'\uparrow}(\mathbf{r})\Psi_{n'\uparrow}^*(\mathbf{r}')\Psi_{n\downarrow}(\mathbf{r}')}{-\omega - (\epsilon_{n\downarrow} - \epsilon_{n'\uparrow}) + i\delta}, \quad (2) \end{aligned}$$

Here \downarrow is for majority (**isp**=1) and \uparrow is for minority (**isp**=2), $\begin{pmatrix} \text{isp}=1 \\ \text{isp}=2 \end{pmatrix} = \begin{pmatrix} \downarrow \\ \uparrow \end{pmatrix}$.

$$\begin{aligned} S^+(\mathbf{r}) &= \Psi_{\text{isp}=1}^*(\mathbf{r})\Psi_{\text{isp}=2}(\mathbf{r}) = \Psi_\downarrow^*(\mathbf{r})\Psi_\uparrow(\mathbf{r}), \\ S^-(\mathbf{r}') &= \Psi_{\text{isp}=2}^*(\mathbf{r}')\Psi_{\text{isp}=1}(\mathbf{r}') = \Psi_\uparrow^*(\mathbf{r}')\Psi_\downarrow(\mathbf{r}'), \end{aligned}$$

Notes:

- This definition of χ^{0+-} results in $\chi^{0+-} \rightarrow \frac{m}{\omega - \Delta_{\text{ex}}}$ at $\mathbf{q} = 0$ in the case of shifted-band model as $\epsilon_{\mathbf{k}}^\uparrow = \epsilon_{\mathbf{k}}^\downarrow + \Delta_{\text{ex}}$. Here Δ_{ex} means exchange splitting, and $m = N_\downarrow - N_\uparrow$. For paramagnetic case, this definition gives $\chi_0 = 2\chi^{0+-}$, where χ_0 is usual Lindhard polarization function for density response.
- Recall $e^{-i(\epsilon - \epsilon')t}\theta(t) = e^{-i\epsilon t}\theta(t) \times e^{+i\epsilon' t}\theta(t)$ or equivalently $\frac{1}{\omega - (\epsilon - \epsilon') + i\delta} = \int d\omega' \frac{1}{\omega - \omega' - \epsilon + i\delta} \frac{1}{-\omega' - \epsilon' - i\delta}$.
- If no occupation for minority channel, only the 1st term in Eq.(??) remains. In contract to charge density, χ^{0+-} is not symmetric for $\omega \leftrightarrow -\omega$.

By the way, the physically meaningful quantity is the retarded version of χ^{0+-} , named as χ_{Ret}^{0+-} , which is given by changing the sign of $+i\delta$ in Eq.(??) so as to make it propotional to $\theta(t)$. Let us assume colinear case (z-axis), and consider adding external transversal magnetic fields $B_x(\mathbf{r}, t)$ and $B_y(\mathbf{r}, t)$ for our system. For non-interacting system, this χ_{Ret}^{0+-} specify the linear repsonse to such B_x and B_y . Instead of them, it is convenient to use the complex field, $b^- = b_x - ib_y$, (I introduce \mathbf{b} in unit of $g\mu_B/2$, so that $(g\mu_B/2)B_x = b_x$: electron's spin magnetic momeents is $-2g\mu_B\mathbf{s}$.) Then the additional Hamiltonian due to this magnetic field is written as

$$\begin{aligned} H_{\text{ext_mag}} &= \int d^3r B(\mathbf{r}) \cdot g\mu_B\mathbf{s}(\mathbf{r}) = 2 \int d^3r \mathbf{b}(\mathbf{r}) \cdot \mathbf{s}(\mathbf{r}) \\ &= \int d^3r [b^+(\mathbf{r})s^-(\mathbf{r}) + b^-(\mathbf{r})s^+(\mathbf{r}) + 2b_z(\mathbf{r})s_z(\mathbf{r})], \quad (3) \end{aligned}$$

where $s^-(\mathbf{r}) = s_x(\mathbf{r}) - is_y(\mathbf{r})$ and so on. $s_x(\mathbf{r}) = \sum_{\alpha, \beta} \langle \hat{\psi}_\alpha^\dagger(\mathbf{r}) \frac{1}{2} \sigma_{\alpha\beta}^x \hat{\psi}_\beta(\mathbf{r}) \rangle$ and so on, where $\sigma_{\alpha\beta}^x$ is the Pauli matrix. The induced spin moment Δs^- for non-interacting system is given as

$$\Delta s^-(\mathbf{r}, t) = \int dt' d^3r' \chi_{\text{Ret}}^{0+-}(\mathbf{r}, \mathbf{r}', t - t') b^-(\mathbf{r}', t'). \quad (4)$$

[because of $\theta(t)$ in χ_{Ret}^{0+-} , $t - t' > 0$.]

In the case of interacting system, we need to construct χ^{+-} . We define $U(\mathbf{r}, \mathbf{r}', \omega)$ as

$$(\chi^{+-})^{-1} = (\chi^{0+-})^{-1} - U. \quad (5)$$

This is taken as the definition of $U(\mathbf{r}, \mathbf{r}', \omega)$. How to define U is the problem — See my **spin wave paper** in Arxiv. We utilize one degree of freedom per atom (so χ^{+-} is the matrix whose dimension is the number of magnetic atoms), and sum rule. (Some one may call this approximation as “rigid moment approximation”. But it can be misleagind. Be careful about what I did.

—MEMO—

eps_lmfh_chipm: Our fpgw code can calculate χ^{0+-} in this form of expansion

$$\chi^{0+-}(\mathbf{r}, \mathbf{r}', \omega) = \frac{1}{N} \sum_{\mathbf{q}} \chi_{\mathbf{q}}^{+-}(\mathbf{r}, \mathbf{r}', \omega) = \frac{1}{N} \sum_{\mathbf{q}} \sum_I \sum_J M_I^{\mathbf{q}}(\mathbf{r}) \chi_{\mathbf{q}IJ}^{0+-}(\omega) (M_J^{\mathbf{q}}(\mathbf{r}'))^*. \quad (6)$$

Here $\{M_I^{\mathbf{q}}(\mathbf{r})\}$ is the complete set with the periodicity specified by \mathbf{q} (Mixed basis). In other words, $\{M_I^{\mathbf{q}}(\mathbf{r})/e^{i\mathbf{q}\mathbf{r}}\}$ is the complete set to expand periodic function. However, I have not

used this now...;problem is determination of U . How to do it? (sum rule is not enough. we need static response?).

3.1.1 Sum rule(moment)

The equation of motion of spin is written as

$$i\dot{\hat{\mathbf{S}}} = [\hat{\mathbf{S}}, \hat{H}] \quad (7)$$

$$\begin{aligned} \chi^{+-}(\mathbf{r}, \mathbf{r}', t - t') &= -i\langle T(S_+(\mathbf{r}, t)S_-(\mathbf{r}', t')) \rangle \\ &= -i\langle S_+(\mathbf{r}, t)S_-(\mathbf{r}', t') \rangle \theta(t - t') + i\langle S_-(\mathbf{r}', t')S_+(\mathbf{r}, t) \rangle \theta(t' - t) \end{aligned} \quad (8)$$

Thus

$$\begin{aligned} \frac{\partial}{\partial t} \chi^{+-}(\mathbf{r}, \mathbf{r}', t - t') &= -i[S_+(\mathbf{r}, t), S_-(\mathbf{r}', t)]\delta(t - t') \\ &\quad - \langle [S_+(\mathbf{r}, t), H]S_-(\mathbf{r}', t') \rangle \theta(t - t') + \langle S_-(\mathbf{r}', t')[S_+(\mathbf{r}, t), H] \rangle \theta(t' - t), \end{aligned} \quad (9)$$

where $[S_+(\mathbf{r}, t), S_-(\mathbf{r}', t)] = 2S_z(\mathbf{r}, t)\delta(\mathbf{r} - \mathbf{r}')$. As $\int d^3r [S_+(\mathbf{r}, t), H] = 0$, we have

$$\int d^3r \frac{\partial}{\partial t} \chi^{+-}(\mathbf{r}, \mathbf{r}', t - t') = -i\langle [S_+(\mathbf{r}, t), S_-(\mathbf{r}', t)] \rangle \delta(t - t') = -2i\langle S_z(\mathbf{r}, t) \rangle \delta(t - t') \quad (10)$$

This reads

$$\int d^3r \omega \chi^{+-}(\mathbf{r}, \mathbf{r}', \omega) = 2\langle S_z(\mathbf{r}', t) \rangle = M_z(\mathbf{r}') \quad (11)$$

• At $\omega \rightarrow \infty$, this condition get stronger as

$$\chi^{+-}(\mathbf{r}', \mathbf{r}, \omega) \rightarrow \frac{M(\mathbf{r})}{\omega} \delta(\mathbf{r} - \mathbf{r}') + O(1/\omega^2). \quad (12)$$

See **spin wave paper**.

3.1.2 Rigid moment approximation

This means that “the magnetic moments are very rigid that they changes without changing its form”. In other words, $\Delta s^-(\mathbf{r})$ induced by any $b_-(\mathbf{r})$ are propotional to its original moment $s_z(\mathbf{r})$. Rigid rotation in spin space can be expressed by e.g., $U^x(\theta) = \exp\left(\frac{i\sigma^x\theta}{2}\right)$ in the case of x-axis rotation. Then you can easily verify $(U^x(\theta))^\dagger \sigma^z U^x(\theta) \propto \sigma^y$. This means $\Delta s^-(\mathbf{r}) \propto s_z(\mathbf{r})$. Note that we did rotation only in spin space. We neglect the mappling of \mathbf{r} when we rotate spin—this will cause little problem when the moment is rather spherical. Be careful; our approximation (one-degree of freedom per magnetic atom) may be a little different from the “rigid moment approximation”. I did not want to mix it up, thus I avoided this terminology in my **[spin wave paper]**.

3.1.3 static $J(q)$ calculation— Heisenberg Model

(See kotani’s SW paper). The total energy of our spin system is assumed to be

$$\mathcal{H} = - \sum_{Rn} \sum_{R'n'} J_{RnR'n'} \mathbf{S}_{Rn} \cdot \mathbf{S}_{R'n'} + g\mu_B \sum_{Rn} \mathbf{S}_{Rn} \cdot \mathbf{B}_{Rn} \quad (13)$$

. Here we take all site indexes Rn and $R'n'$ ($J_{RnRn} = 0$, $J_{RnR'n'} = J_{R'n'Rn}$. It we restrict sum as $Rn > R'n'$, factor 2 appears.). \mathbf{S}_{Rn} is the spin at Rn (R is for primitive cell, n specify site in a cell). The equation of motion $-i\hbar\dot{\mathbf{S}}_{Rn} = [\mathcal{H}, \mathbf{S}_{Rn}]$, is reduced to be

$$\hbar\dot{\mathbf{S}}_{Rn} = \mathbf{S}_{Rn} \times \left(2 \sum_{R'n'} J_{RnR'n'} \mathbf{S}_{R'n'} - g\mu_B \mathbf{B}_{Rn} \right) \quad (14)$$

We introduce $g\mu_B \mathbf{B} = 2\mathbf{b}$, and $\mathbf{S}_{Rn} = \mathbf{S}_{Rn}^0 + \Delta\mathbf{S}_{Rn}$. Then Eq.(??) reduce to

$$\begin{aligned} \hbar\dot{\Delta\mathbf{S}}_{Rn} &= \mathbf{S}_{Rn}^0 \times \left(2 \sum_{R'n'} J_{RnR'n'} \Delta\mathbf{S}_{R'n'} \right) + \Delta\mathbf{S}_{Rn} \times \left(2 \sum_{R'n'} J_{RnR'n'} \mathbf{S}_{R'n'} \right) - 2\mathbf{S}_{Rn}^0 \times \mathbf{b}_{Rn} \\ &= \sum_{R'n'} (2\mathbf{S}_{Rn}^0 J_{RnR'n'}) \times \Delta\mathbf{S}_{R'n'} - \left(2 \sum_{R'n'} J_{RnR'n'} \mathbf{S}_{R'n'}^0 \right) \times \Delta\mathbf{S}_{Rn} - 2\mathbf{S}_{Rn}^0 \times \mathbf{b}_{Rn} \end{aligned} \quad (15)$$

Introduce the fourier transformation as $\Delta \mathbf{S}_{Rn} = \frac{1}{N} \sum_{\mathbf{k}} \Delta \mathbf{S}_n(\mathbf{k}) e^{i\mathbf{k}\mathbf{R}}$. Then Eq.(??) reduce to

$$\hbar \Delta \mathbf{S}_n(\mathbf{k}) = \sum_{n'} \left(2\mathbf{S}_n^0 J_{nn'}(\mathbf{k}) - \left(2 \sum_{n''} J_{nn''}(0) \mathbf{S}_{n''}^0 \right) \delta_{nn'} \right) \times \Delta \mathbf{S}_{n'}(\mathbf{k}) - 2\mathbf{S}_n^0 \times \mathbf{b}_n(\mathbf{k}) \quad (6)$$

Assume $\Delta \mathbf{S}_n(\mathbf{k}) \propto e^{-i\frac{\omega t}{\hbar}}$, we have

$$\sum_{n'} \left(\frac{i\omega \delta_{nn'}}{2} + \mathbf{S}_n^0 J_{nn'}(\mathbf{k}) - \left(\sum_{n''} J_{nn''}(0) \mathbf{S}_{n''}^0 \right) \delta_{nn'} \right) \times \Delta \mathbf{S}_{n'}(\mathbf{k}) = \mathbf{S}_n^0 \times \mathbf{b}_n(\mathbf{k}). \quad (17)$$

Let us consider colinear ground state, then $\mathbf{S}_n^0 = S_n \mathbf{e}_z$ (S_n is the size of spin with sign). You have

$$\sum_{n'} \left(\frac{i\omega \delta_{nn'}}{2S_n} \right) \Delta \mathbf{S}_{n'}(\mathbf{k}) + \sum_{n'} \left(J_{nn'}(\mathbf{k}) - \left(\sum_{n''} \frac{1}{S_n} J_{nn''}(0) S_{n''} \right) \delta_{nn'} \right) \mathbf{e}_z \times \Delta \mathbf{S}_{n'}(\mathbf{k}) = \mathbf{e}_z \times \mathbf{b}_n(\mathbf{k}). \quad (18)$$

As $\mathbf{S} = S^+ \frac{\mathbf{e}_x - i\mathbf{e}_y}{2} + S^- \frac{\mathbf{e}_x + i\mathbf{e}_y}{2} + S^z \mathbf{e}_z$, and $\mathbf{e}_z \times (\mathbf{e}_x \pm i\mathbf{e}_y) = \mp i(\mathbf{e}_x \pm i\mathbf{e}_y)$ we have,

$$\sum_{n'} \left(\frac{\omega \delta_{nn'}}{2S_n} - \bar{J}_{nn'}(\mathbf{k}) \right) S_{n'}^+(\mathbf{k}) = b_n^+(\mathbf{k}). \quad (19)$$

$$\sum_{n'} \left(\frac{\omega \delta_{nn'}}{2S_n} + \bar{J}_{nn'}(\mathbf{k}) \right) S_{n'}^-(\mathbf{k}) = b_n^-(\mathbf{k}), \quad (20)$$

where

$$\bar{J}_{nn'}(\mathbf{k}) = J_{nn'}(\mathbf{k}) - \left(\sum_{n''} \frac{1}{S_n} J_{nn''}(0) S_{n''} \right) \delta_{nn'} \quad (21)$$

This $\bar{J}_{nn'}(\mathbf{k})$ and also S_n are stored in **Jmat** file (or **JMAT** line when you run a script `ecal/util/calj_summary_mat` which calls `(calj_nlfc_mat)`). Only the differenc between $\bar{J}_{nn'}(\mathbf{k})$ and $J_{nn'}(\mathbf{k})$ are diagonal parts. These are determined so that $\int d^3k J_{nn}(\mathbf{k}) = 0$.

JJMAT contains another definition of J , which is to reproduce SW spectrum (but it does not work well in cases because the SW peaks are not well identified at high \mathbf{k} .)

See my spin wave paper.

3.1.4 $J(q)$ and T_c

— this section is my memo. Not need to read here —

(This section is not consistent with previous page. Only a case, with an atom in the cell). The total energy of our spin system is assumed to be

$$E_{\text{spin}} = - \sum_i \sum_j J_{ij} \mathbf{e}_i \cdot \mathbf{e}_j \quad (22)$$

. Here we take all site indexes i and j ($J_{ii} = 0$. $J_{ij} = J_{ji}$. It we restrict sum as $i > j$, factor 2 appears.). \mathbf{e}_i is the unit vector to specify the spin direction.

If we identify E_{spin} as the Heisenberg hamiltonian with a fixed spin moment $m = N^\downarrow - N^\uparrow$, the spin wave dispersion is given as

$$\omega_{\mathbf{q}} = \frac{4}{m} [J(0) - J(\mathbf{q})]. \quad (23)$$

This $\omega_{\mathbf{q}}$ is different from the true pole of $\langle m | (\chi^{+-}(\mathbf{q}, \omega))^{-1} | m \rangle$ except $\mathbf{q} \rightarrow 0$.

The critical temperature T_c is given as $T_c = \frac{2}{3} J(\mathbf{Q}) Q_{\text{factor}}$. This Q_{factor} can be $(S+1)/S$, but it seems to be taken as unity usually... $J(Q=0)$ is for ferromagnetic case.

In order to calculate $J(0)$, integrate the left hand side of Eq.(??) in the BZ and use $\int \frac{\Omega d^3q}{2\pi} J(\mathbf{q}) = 0$. It gives

$$T_c = \frac{2}{3} J(0) = \frac{2}{3} \int \frac{\Omega d^3q}{2\pi} \langle m | (\chi^{+-}(\mathbf{q}))^{-1} | m \rangle = \frac{2}{3} \frac{m}{4} \int \frac{\Omega d^3q}{2\pi} \omega_{\mathbf{q}} \quad (24)$$

This equation contains two problems.

- (1) Mapping to a Heisenberg model.
This may cause a problem in the case of transition metals, and so. The spin waves have strong dumping. Further, we don't include the temperature-dependence of the model itself.
- (2) Mean field approximation to solve the Heisenberg model.
In other words, Q_{factor} should be a functional of $J(\mathbf{q})$ for all \mathbf{q} . We can divide the problem into classical part and quantum part. A controversial point is that the integral in Eq.(??) is rather dominated by the contribution around the BZ boundaries, though we can expect that T_c can be rather strongly controlled by low energy $\omega_{\mathbf{q}}$.

— (this is what Mark says)—

[MF Quantum T_c] — too high. $(S+1)/S * \text{MFC}$

[Full Quantum T_c] — Exact solution of the Heisenberg Model.

[MF Classical T_c]

[Full Classical T_c] (probably $\sim 80\%$ of MFC)

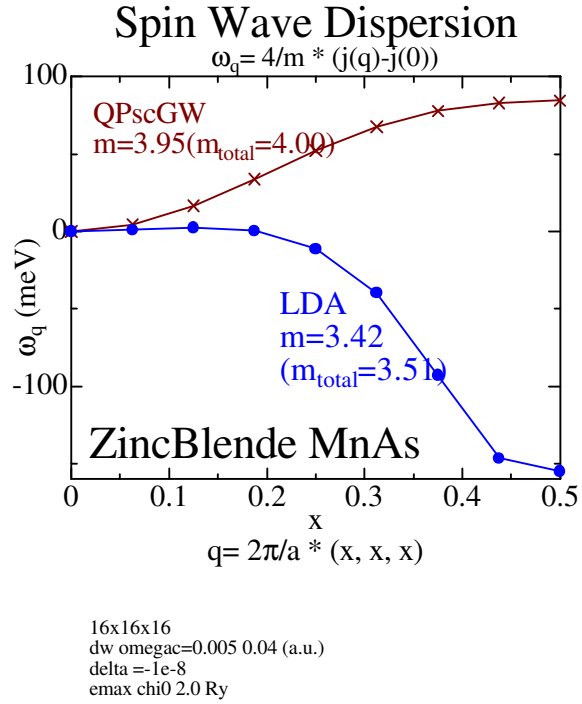


Figure 1: $\omega_{\mathbf{q}}$ for ZincBlend MnAs by Eq.???. LDA case vs. QSGW case. QSGW makes ZincBlend MnAs as half-metallic. m denotes on-site moment in μ_B . The mean-field T_c in QSGW is about 600 K. NOTE: not yet published. We have to re-examine this!!!

4 Memo(Method and Equations)

Hilbert transformation(called as Sergey mode)):

This is standard calculation now. Our standard version (a script `gw_lmfh`, `gwsc`) is with the Hilbert transformation (Kramers-Krönig relation). We first calculate only the imaginary part of Π Then we get full Π through the Hilbert transformation. In other words, we calculate the imaginary part of Π by replacement of

$\left(\frac{1}{\omega - \epsilon_{\mathbf{q}+\mathbf{k}_{n'}} + \epsilon_{\mathbf{k}_n} + i\delta} - \frac{1}{\omega + \epsilon_{\mathbf{q}+\mathbf{k}_{n'}} - \epsilon_{\mathbf{k}_n} - i\delta}\right)$ with $\delta(\omega - \epsilon_{\mathbf{q}+\mathbf{k}_{n'}} + \epsilon_{\mathbf{k}_n})$ in Eq.xxx. We accumulate weights for each bin. Then $\delta(\omega - \epsilon_{\mathbf{q}+\mathbf{k}_{n'}} + \epsilon_{\mathbf{k}_n})$ is replaced by the original one (each bin of weight is converted to corresponding real part).

Core orthogonalization problem:

At you see in EQS.(32) or (33). it should be $\langle \exp(-i\mathbf{q}\mathbf{r})|\Pi|\exp(i\mathbf{q}\mathbf{r})\rangle \rightarrow 0$ at $\mathbf{q} \rightarrow 0$ because $\langle \psi_{\mathbf{k}n}|\psi_{\mathbf{k}n'}\rangle = 0$ for occupied n and unoccupied n' . However, core eigenfunction is not completely orthogonal to the valence eigenfunction (in our FP-LMTO scheme). Thus this behavior can not be so perfect. In order to keep the behavior, we orthogonalize the core eigenfunctions. It is by an optional switch `CoreOrth` in the input file `GWinput` (See the description in the explanation of it). (NOTE: `CoreOrth` is not maintained recently) However, we find that QPE usually affected little by this option.

4.1 Brillouin-zone integral for the self-energy; the smearing method and the offset- Γ method.

- Smearing method

[Note that this is not for not for Π —it is usually evaluated by the tetrahedron method.]

Our smearing method means, we replace δ function with $\bar{\delta}(\omega)$ as shown in EQ.41.

$$\bar{\delta}(\omega) = \frac{1}{E_{\text{smear}}} \text{ for } -\frac{E_{\text{smear}}}{2} < \omega < \frac{E_{\text{smear}}}{2}, \text{ otherwise zero (GaussSmear = off mode } E_{\text{smear}} = \text{esmr in GWinput)} \quad (25)$$

or

$$\bar{\delta}(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\omega^2}{2\sigma^2}) \quad (\text{GaussSmear} = \text{on mode}, \sigma = \text{esmr in GWinput})$$

As for insulator, E_{smear} is irrelevant (due to numerics, $E_{\text{smear}} = 0$ is not allowed. You need to set E_{smear} smaller than band gap. But not too small). However, you may need to pay attention to the size of E_{smear} in the case of metal. (The pole distribution around the Fermi energy is shown in a file `DOSACC.1da`). $\rho_{\mathbf{q}nm}(\mathbf{k})$ can have unsmooth behavior as a function of k in BZ due to the Fermi energy cutoff in the case of metal. Larger E_{smear} reduce the unsmoothness. With denser meshing in BZ, you are allowed to use smaller E_{smear} .

New offset Gamma method is introduced in paper. xxx (this is old document for previous offset-Gamma method (BZmesh=1))

XX

• **Offset gamma method** See Ref.I. We have to take the two limit $E_{\text{smer}} \rightarrow 0$ and $N_1 N_2 N_3 \rightarrow \infty$. There could be a convergence problem as for the states $\Psi_{\mathbf{qn}}$ whose $\epsilon_{\mathbf{qn}}$ are near E_F in the case of low DOS at E_F . In Fig.??, we showed the convergence test for $\langle \Psi_{\mathbf{qn}} | \Sigma_x | \Psi_{\mathbf{qn}} \rangle$ as a function of E_{smer} in the case of CaB₆. The DOS at E_{Fermi} for CaB₆ is quite small, therefore, it is a severe test for the smearing method. Through the comparison between 444 and 666 case, we can say a rapid change at $E_{\text{smer}} \rightarrow 0$ will be virtual because of the finite number of k points. Therefore we can use $E_{\text{smer}} \sim 0.05$ Ry in order to avoid such a finite number effect at $E_{\text{smer}} \rightarrow 0$. Then we can expect 0.1 eV level of accuracy under the assumption of the flat behavior at $E_{\text{smer}} \rightarrow 0$. Due to the cancellation effects, $\Sigma_x + \Sigma_c$ can give better convergences.

XX

=== How to calculate Spin Wave. (old memo) =====

The script calj_nlfc_metal summarize peak position and width of SW.
 It is in ecal/util/
 (it calls calj_interp_mat.F and calj_nlfc_mat.F in it).

We can calculate

chi^{0+-} for nolfc mode : epsPP_lmfh_chipm

chi^{0+-} full matrix mode: eps_lmfh_chipm

However, I now think eps_lmfh_chipm mode might be not so meaningful,
 because we have not find a way to determined U.

(1) So use epsPP_lmfh_chipm now.

You need sigm.*, ctrl.*, GWinput, rst.* files.

chi^{0+-} is a matrix whose dimension is the number of magnetic atoms in a cell.

epsPP_lmfh_chipm : Its main output is ChiPM*.nlfc.mat.

Set MagAtom section and q vector in GWinput.

E.g. MagAtom 1 -3

We treat two magnetic atoms 1st and 3rd.

The third atom point opposite direction; AF case or so.

Note; atoms can be re-ordered by lmf. See a file names ad LMT0.

format of ChiPM*.nlfc.mat

q(1:3)	omega(Ry)	<eiqr chipm eiqr>	<eiqr chipm eiqr> ^{-1}
3*real	real	complex	complex

but Need to check normlizaion for eiqr= e^{{i {\bf q} \bfr}}

(ChiPM*.nlfc.dat is now deleted; it was a date file of <eiqr|chipm0|eiqr> but
 for \omega-independent U)

format of ChiPM*.nlfc.dat (no \omega-dependent U---so a little wrong).

q(1:3)	omega(Ry)	<eiqr chipm eiqr>	<eiqr chipm eiqr> ^{-1}
3*real	real	complex	complex

but Need to check normlizaion for eiqr= e^{{i {\bf q} \bfr}})

(2) Perform calj_nlfc_metal (MnAs) or calj_summary_mat(NiO,MnO)

Then you will have SWE, FMHF JMAT MMAT. uu0uu1 is generated (U matrix).

Also generate SW spectrum.

These script calls calj_nlfc_mat.F and calj_interp_mat.F internally.

But these scrips are imperfect yet.

NEED FIXING!!! In anyway, it is necessary to learn these fortran codes
 (and my SW paper) to calculate spin susceptibility.

You see static moment and I by echo ChiPM0001.nlfc.mat|calj_nlfc_xxx.

(memo: for olde version;

x calj_det.F : Calculate spin wave from ChiPM*.mat matrix file.

x calj_detc.F : modified version of calj_det.F

x calj_search0.F : SW (pole search) for ChiPM*.dat ChiPM*.nlfc.dat files.

x calj_interp.F : make spectrum function by interpolation (for metal).

x

x calj_summary_ferro : for ferro

x calj_summary_aferro: for aferro

```

x calj_interp: for metal
)

-----

ErAs case: How to get sigma without MZ?
=== need to re-implement this again if necessary ===

lmf eras --wsig:fbz
cp sigm2.eras sigm.eras
Change ctrl.eras as
    nk=3
    Remove MZ
    RDSIG =10012
lmf eras --wsig:newkp
mv sigm2.eras sigm.eras
Change ctrl.eras backs up origial except MZ.
    nk=8      (move back to original)
    RDSIG=12 (move back to original)

-----

syml.coo
41 -.5 .5 .5 0 0 0    X G
41 0 0 0 -.25 .75 -.25    G L
41 -.75 .25 .25 -.0625 .875 -.0625    L U
41 -.0625 .875 -.0625 .25 .25 .25    U T
41 .25 .25 .25 0 0 0    T G
0 0 0 0 0 0 0

lmf --band:fn=syml coo >llmf_band
plbnds -fplot -ef=0 -scl=13.605 -spin2 eras
fplot -f plot.plbnds

-----

iarg = iargc()
call getarg(0,str)
call getarg(1,str) !1st argument
call system('ls') !system call

```

4.2 eqs. for Fourier transformation in fpwgw program

(I think this section is still meananigful for my code).

Any site-dependent functions $A(\mathbf{R})$ are written as

$$\bar{A}(\mathbf{k}) = \sum_{\mathbf{R}} A(\mathbf{R}) e^{-i\mathbf{k}\mathbf{R}} \quad (26)$$

$$A(\mathbf{R}) = \frac{1}{N} \sum_{\mathbf{k}} \bar{A}(\mathbf{k}) e^{i\mathbf{k}\mathbf{R}} \rightarrow \int \frac{\Omega d^3k}{(2\pi)^3} \bar{A}(\mathbf{k}) e^{i\mathbf{k}\mathbf{R}} \quad (N \rightarrow \infty.) \quad (27)$$

$$(\text{For any } \alpha(\mathbf{k}) \text{ and } \bar{A}(\mathbf{k}), \frac{1}{N} \sum_{\mathbf{k}} \alpha(\mathbf{k}) \bar{A}(\mathbf{k}) \rightarrow \int \frac{\Omega d^3k}{(2\pi)^3} \alpha(\mathbf{k}) \bar{A}(\mathbf{k}) \quad (N \rightarrow \infty).)$$

In the case of $A(\mathbf{R}) = 1$ (constant function), $\bar{A}(\mathbf{k} \neq 0) = 0$ and $\bar{A}(\mathbf{k} = 0) = N$. At $N \rightarrow \infty$, $\bar{A}(\mathbf{k}) = \frac{(2\pi)^3}{\Omega} \delta(\mathbf{k})$. Here \mathbf{k} takes discrete values as $\mathbf{k}_{n_1 n_2 n_3} = \frac{2\pi}{a} \left(\frac{n_1}{N} \mathbf{b}_1 + \frac{n_2}{N} \mathbf{b}_2 + \frac{n_3}{N} \mathbf{b}_3 \right)$, where $N = n_1 n_2 n_3$. a is the given scale of the system (given in `alat`). \mathbf{b}_i are reciprocal lattice vector (given in `qlat` or `qbas`). $\mathbf{a}_i \cdot \mathbf{b}_j = \delta_{ij}$. (\mathbf{a}_i is given in `plat`). Note $\int \frac{\Omega d^3k}{(2\pi)^3} = 1$.

At first, we assume $\Psi_{\mathbf{k}n}$ is normalized in the macroscopic volume V . In `fpwgw` program, we use $\bar{\Psi}_{\mathbf{k}n}$ as $\bar{\Psi}_{\mathbf{k}n} = \sqrt{\frac{V}{\Omega}} \Psi_{\mathbf{k}n}$, where $\Omega = V/N$ denote the volume of primitive cell. Thus the normalization is

$$\int d^3r |\bar{\Psi}_{\mathbf{k}n}(\mathbf{r})|^2 = 1 \quad (28)$$

1. Then

$$\sum_{\mathbf{k}} \Psi_{\mathbf{k}}^*(\mathbf{r}) \Psi_{\mathbf{k}}(\mathbf{r}') = \int \frac{V d^3k}{(2\pi)^3} \Psi_{\mathbf{k}}^*(\mathbf{r}) \Psi_{\mathbf{k}}(\mathbf{r}') = \int \frac{\Omega d^3k}{(2\pi)^3} \bar{\Psi}_{\mathbf{k}}^*(\mathbf{r}) \bar{\Psi}_{\mathbf{k}}(\mathbf{r}'). \quad (29)$$

2. The Fourier transformation for Bravais lattice.

$$\delta_{\mathbf{T}\mathbf{T}'} = \int \frac{\Omega d^3k}{(2\pi)^3} e^{i\mathbf{k}(\mathbf{T}-\mathbf{T}')}, \quad \sum_{\mathbf{T}} e^{i\mathbf{k}(\mathbf{T}-\mathbf{T}')} = \frac{(2\pi)^3}{\Omega} \delta(\mathbf{k}) \quad (30)$$

\mathbf{T} denotes Bravais lattice as $\mathbf{T}_{i_1 i_2 i_3} = a (i_1 \mathbf{a}_1 + i_2 \mathbf{a}_2 + i_3 \mathbf{a}_3)$.

3. At first, we express $\chi(\mathbf{r}, \mathbf{r}')$ as the superposions of $\chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}')$ components;

$$\chi(\mathbf{r}, \mathbf{r}') = \frac{1}{N} \sum_{\mathbf{q}} \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') = \int \frac{\Omega d^3q}{(2\pi)^3} \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}'). \quad (31)$$

Here $\chi_{\mathbf{q}}$ is written as

$$\chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') = \sum_{\mathbf{T}} \chi(\mathbf{r} + \mathbf{T}, \mathbf{r}') e^{-i\mathbf{q}\mathbf{T}}, \quad (32)$$

which satisfy

$$\chi_{\mathbf{q}}(\mathbf{r} + \mathbf{T}, \mathbf{r}') = \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') e^{i\mathbf{q}\mathbf{T}}, \quad \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}' + \mathbf{T}) = \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') e^{-i\mathbf{q}\mathbf{T}}. \quad (33)$$

Thus we can construct $\chi(\mathbf{r}, \mathbf{r}')$ from $\chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}')$ where \mathbf{r} and \mathbf{r}' is limited in unit cell. From the above equation, we have

$$\chi(\mathbf{r}, \mathbf{r}' + \mathbf{T}) = \frac{1}{N} \sum_{\mathbf{q}} \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') e^{-i\mathbf{q}\mathbf{T}} = \int \frac{\Omega d^3q}{(2\pi)^3} \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') e^{-i\mathbf{q}\mathbf{T}}. \quad (34)$$

4. Then $\chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}')$ is given as

$$\chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') = \sum_n \sum_{n'} \int \frac{\Omega d^3k'}{(2\pi)^3} \frac{\bar{\Psi}_{\mathbf{k}n}^*(\mathbf{r}) \bar{\Psi}_{\mathbf{k}'n'}(\mathbf{r}) \bar{\Psi}_{\mathbf{k}'n'}^*(\mathbf{r}') \bar{\Psi}_{\mathbf{k}n}(\mathbf{r}')}{\dots} + \dots, \quad (35)$$

where $\mathbf{k}' = \mathbf{q} + \mathbf{k}$.

5. Bloch basis (mixed basis) $M_{\mathbf{q}}(\mathbf{r})$ satisfy $M_{\mathbf{q}}(\mathbf{r} + \mathbf{T}) = M_{\mathbf{q}}(\mathbf{r}) e^{i\mathbf{q}\mathbf{T}}$. $M_{\mathbf{q}}(\mathbf{r})$ are is normalized in Ω .

$$\chi_{\mathbf{q}}(I, J) = \int_{\Omega} d^3r \int_{\Omega} d^3r' M_{\mathbf{q}I}^*(\mathbf{r}) \chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') M_{\mathbf{q}J}(\mathbf{r}') \quad (36)$$

$$\chi_{\mathbf{q}}(\mathbf{r}, \mathbf{r}') = \sum_{I,J} \int \frac{\Omega d^3k}{(2\pi)^3} \sum_{I',J'} M_{\mathbf{q}I'}(\mathbf{r}) O_{\mathbf{q}I'I}^{-1} \chi_{\mathbf{q}}(I, J) O_{\mathbf{q}JJ'}^{-1} M_{\mathbf{q}J'}^*(\mathbf{r}') \quad (37)$$

4.3 χ^{0+-}

In \mathbf{q} space, χ^{0+-} is written as

$$\begin{aligned} -\chi_{\mathbf{q}}^{0+-}(\mathbf{r}, \mathbf{r}', \omega) &= \sum_{\mathbf{kn}\downarrow}^{\text{occ}} \sum_{\mathbf{k}'n'\uparrow}^{\text{unocc}} \frac{\Psi_{\mathbf{kn}\downarrow}^*(\mathbf{r}) \Psi_{\mathbf{k}'n'\uparrow}(\mathbf{r}) \Psi_{\mathbf{k}'n'\uparrow}^*(\mathbf{r}') \Psi_{\mathbf{kn}\downarrow}(\mathbf{r}')}{\omega - (\epsilon_{\mathbf{k}'n'\uparrow} - \epsilon_{\mathbf{kn}\downarrow}) + i\delta} \\ &+ \sum_{\mathbf{kn}\downarrow}^{\text{unocc}} \sum_{\mathbf{k}'n'\uparrow}^{\text{occ}} \frac{\Psi_{\mathbf{kn}\downarrow}^*(\mathbf{r}) \Psi_{\mathbf{k}'n'\uparrow}(\mathbf{r}) \Psi_{\mathbf{k}'n'\uparrow}^*(\mathbf{r}') \Psi_{\mathbf{kn}\downarrow}(\mathbf{r}')}{-\omega - (\epsilon_{\mathbf{kn}\downarrow} - \epsilon_{\mathbf{k}'n'\uparrow}) + i\delta}, \end{aligned} \quad (38)$$

where $\mathbf{k}' = \mathbf{q} + \mathbf{k}$.

In the case with the time-reversal symmetry, we have $\Psi_{\mathbf{kn}\downarrow}(\mathbf{r}) = \Psi_{-\mathbf{kn}\downarrow}^*(\mathbf{r})$ and $\Psi_{\mathbf{kn}\uparrow}(\mathbf{r}) = \Psi_{-\mathbf{kn}\uparrow}^*(\mathbf{r})$. Then Eq.(??) is reduced to be

$$\begin{aligned} -\chi_{\mathbf{q}}^{0+-}(\mathbf{r}, \mathbf{r}', \omega) &= \sum_{\mathbf{kn}\downarrow}^{\text{occ}} \sum_{\mathbf{k}'n'\uparrow}^{\text{unocc}} \frac{\Psi_{\mathbf{kn}\downarrow}^*(\mathbf{r}) \Psi_{\mathbf{k}'n'\uparrow}(\mathbf{r}) \Psi_{\mathbf{k}'n'\uparrow}^*(\mathbf{r}') \Psi_{\mathbf{kn}\downarrow}(\mathbf{r}')}{\omega - (\epsilon_{\mathbf{k}'n'\uparrow} - \epsilon_{\mathbf{kn}\downarrow}) + i\delta} \\ &+ \sum_{\mathbf{kn}\uparrow}^{\text{occ}} \sum_{\mathbf{k}'n'\downarrow}^{\text{unocc}} \frac{\Psi_{\mathbf{kn}\uparrow}^*(\mathbf{r}) \Psi_{\mathbf{k}'n'\downarrow}(\mathbf{r}) \Psi_{\mathbf{k}'n'\downarrow}^*(\mathbf{r}') \Psi_{\mathbf{kn}\uparrow}(\mathbf{r}')}{-\omega - (\epsilon_{\mathbf{k}'n'\downarrow} - \epsilon_{\mathbf{kn}\uparrow}) + i\delta}. \end{aligned} \quad (39)$$

(in the second term of Eq.(??), we need to rename \mathbf{kn} as $-\mathbf{k}'n'$, and $\mathbf{k}'n'$ as $-\mathbf{kn}$. Then we need to call \mathbf{k}' as $-\mathbf{k}'$.) In the current version **fpgw032f06f.tar.gz**, we use this formula in **hx0fp0** program. (I have not yet implemented the case of non-time reversal symmetry). If you assume paramagnetic case, this agree with the minus half of the usual charge-channel polarization function for the case of time-reversal symmetry, see e.g. Ferdi's GW review Eq.7.3. This Eq.(??) can be expanded by the mixed basis $|M_{\mathbf{q}I}\rangle$ as

$$\begin{aligned} -\langle I | \chi^{0+-}(\mathbf{q}, \omega) | J \rangle = -\chi_{\mathbf{q}}^{0+-}(I, J, \omega) &= \sum_{\mathbf{kn}\downarrow}^{\text{occ}} \sum_{\mathbf{k}'n'\uparrow}^{\text{unocc}} \frac{\langle M_{\mathbf{q}I} | \Psi_{\mathbf{kn}\downarrow}^* \Psi_{\mathbf{k}'n'\uparrow} \rangle \langle \Psi_{\mathbf{k}'n'\uparrow}^* \Psi_{\mathbf{kn}\downarrow} | M_{\mathbf{q}J} \rangle}{\omega - (\epsilon_{\mathbf{k}'n'\uparrow} - \epsilon_{\mathbf{kn}\downarrow}) + i\delta} \\ &+ \sum_{\mathbf{kn}\uparrow}^{\text{occ}} \sum_{\mathbf{k}'n'\downarrow}^{\text{unocc}} \frac{\langle M_{\mathbf{q}I} | \Psi_{\mathbf{kn}\uparrow}^* \Psi_{\mathbf{k}'n'\downarrow} \rangle \langle \Psi_{\mathbf{k}'n'\downarrow}^* \Psi_{\mathbf{kn}\uparrow} | M_{\mathbf{q}J} \rangle}{-\omega - (\epsilon_{\mathbf{k}'n'\downarrow} - \epsilon_{\mathbf{kn}\uparrow}) + i\delta} \end{aligned} \quad (40)$$

To calculate required quantities, we need $\langle m | M_{\mathbf{q}J} \rangle$. This is stored in **MixSpin.*** files. In mode 222 of **hx0fp0**, we directly calculate $\langle m | \chi^{0+-}(\mathbf{q}, \omega) | m \rangle$. In mode 223, we calculate full matrix of $\langle I | \chi^{0+-}(\mathbf{q}, \omega) | J \rangle$ and take its inverse.