# ecalj note

https://github.com/tkotani/ecalj

August 9, 2025

This note becomes rather obsolete. I have to revise things step by step.

## 1 Wannier function

We can generate Wannier functions (maximally localized Wannier Functions or similar) by a script `genMLWF`. It automatically perform cRPA calculation sucessively. (If it is not necessary, insert 'exit' in `genMLWF`, after it performs `lmaxloc2`).

Try to run examples in `ecalj/MATERIALS/Sample_MLWF/`. Read `README` in it. To run the script `genMLWF`, we need to get `GWinput` by editing `GWinput.tmp`. (`mkGWin_lmf2` contains default Wannier section). In addition, we have some settings (energy windows and so on). This is the example of the initial conditions for Cu case. 5 is the number of Wannier function. The most left one means $\phi$ index and the right one of it is $\dot{\phi}$ index. They are written in the **@MNLA_CPHI** file.

Then we can run `genMLWF`. After it finished, we can analyze it results. (if you don't need Wannier funciton plot, You can skip a line of wanplot in genMLWF. Then we don't need to set `vis_wan_*` options.)

### 1.1 lwmatK1 and lwmatK2

If you input the following command

```
>grep Wan lwmatK*
```

You will get the following results. (This case : Cu cases)

```
lwmatK1:  Wannier    1    1   24.644475    0.000000 eV
lwmatK1:  Wannier    1    2   24.644576    0.000000 eV
lwmatK1:  Wannier    1    3   25.471361    0.000000 eV
lwmatK1:  Wannier    1    4   24.644575    0.000000 eV
lwmatK1:  Wannier    1    5   25.470946    0.000000 eV
lwmatK2:  Wannier    1    1    0.000000 eV   -21.263759   -0.000000 eV
lwmatK2:  Wannier    1    2    0.000000 eV   -21.263839    0.000000 eV
lwmatK2:  Wannier    1    3    0.000000 eV   -21.931033   -0.000000 eV
lwmatK2:  Wannier    1    4    0.000000 eV   -21.263839   -0.000000 eV
lwmatK2:  Wannier    1    5    0.000000 eV   -21.930702   -0.000000 eV
```

### Wanneir Branch now under developing (imported from T.Miyake's Wannier and H.Kino's).

A. make at ecalj/fpgw/Wannier/ directory, and do make, and make install.
   (need to check Makefile first). You first have to install fpgw/exec/ in advance.

B. Samples are at these directories.
   MATERIALS/CuMLWFs (small samples),
   MATERIALS/CuMLWF/
   MATERIALS/CuMLWFs/
   MATERIALS/FeMLWF/
   MATERIALS/NiOMLWF/
   MATERIALS/SrVO3MLWF/

C. With GWinput and ctrl.*, run
   >genMLWF
   at these directories.
   In GWinput, we supply settings to generate Wannier funcitons. (Sorry,not documentet yet..

D. After genMLWF, do
   >grep Wan lwmatK*
   then compare these with Result.grepWanlwmatK
   These are onsite effective interactions (diagonal part only shown).
   *.xsf are for plotting the Maximally localized Wannier funcitons.

Anyway, documentaion on Wannier is on the way half.
Time consuming part (and also the advantage) is for effective interaction in RPA.
Look into the shell script genMLWF; you can skip last part if you don't need the effective inte

## 2 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.
============================================================
```

# 3 Linear response calculations

With these scripts for linear response calculations, `eps*`, we can calculate **q**-dependent dielectric funciton $\epsilon(\omega, \mathbf{q})$ (and $v$, $W$) (and $\chi$ for spin fluctuation). But (because of numerical reason), we can not use $\mathbf{q} = 0$ limit. (if $|\mathbf{q}|$ is too small, we have numerical problem, zero divided by zero, because we have not implemented the version to use $\mathbf{q} = 0$.)

- `eps_lmfh`
  Dielectric function epsilon with local field correction. Expensive calculation (we may need to reduce number of wing parts in future...).

- `epsPP_lmfh`
  epsilon without local field correction. $1 - \langle e^{i\mathbf{qr}}|v|e^{i\mathbf{qr}}\rangle\langle e^{i\mathbf{qr}}|\left(\chi^0\right)|e^{i\mathbf{qr}}\rangle$

- `epsPP_lmfh_chipm`

  For spin susceptibility. This essentially calculate non-interacting spin susceptibility. Then it is used for the calculation of full spin susceptibility with `util/calj_*.F` programs (small quick programs). See spin wave paper. See spin susceptibility section Sec.**??**.

- (not maintained now; we will recover this)`eps_lmfh_chipm`

  This gives full non-interacting spin susceptibility. Testing. We have to determine $U$ (Stoner $I$) for the determination of full spin susceptibility. TDLDA? or so?

- `(This is old mode --- not maintained) epsPP_lmfh_chipm_q`

  For spin susceptibility, spin susceptibility $\langle e^{iqr}|\chi(q,\omega)|e^{iqr}\rangle$ In this script, You have to assign that isp=1 is majority, isp=2 is minority. This is with long wave approximation.

---

• We use the histogram method (the Hilbert transformation method); we first calculate its imaginary parts with the tetrahedron technique for dielectric functions. Then we get its real part by the Hilbert transformation.
You need to choose `HisBin_dw,HisBin_ratio`. The width of histogram bins are getting larger when omega gets larger. dw is the size of histogram-bin width at omega=0. At omega=omg_c, its width gets twiced.

To plot dielectric function with reasonable resolution, it might be better to set `dw 0.001` and `omg_c 0.1` for example. You may have to choose small enough omega for spin wave mode as 0.001 Ry (Or smaller). omg_c is given like 0.05 Ry or so. But sometimes it can be like 1Ry.
• `epsPP_lmfh` only calculation an matrix element of dielectric funciton for $exp(i\mathbf{qr})$. Thus very faster than `eps_lmfh` mode. It uses a a special product basis set for cases without inversion (problem is in how to expand $\exp(i\mathbf{qr})$ in the MPB; the product basis is not from phi and phidot, but from spherical Bessel functions).

In *_lmfh_* modes( I now use little for *_lmf_* modes), you can use small enough delta.
Use small enough delta (=-1e-8 a.u.) for spin wave modes (also you can use it for
dielectric function and GW). This is necessary because pole is too smeared
if you use larger delta.

### 3.1 eps_lmfh, epsPP_lmfh: the dielectric functions

You can invoke the script, e.g. as "`eps_lmfh` si".

---

Specify **q** point in `<QforEPS>` or so. Mesh for $\omega$ is specified by `dw, omg_c`.

The obtained data are in EPS*.dat and EPS*.nlfc.dat. EPS*.nlfc.dat contains the result without local-field correction EPS*.dat contains the result with local-field correction (this is generated only for `eps_lmfh`. Both of them contains

**q**(1:3), $\omega$, Re($\epsilon$) Im($\epsilon$), Re($1/\epsilon$), In($1/\epsilon$)

in each line.

### 3.2 epsPP_lmfh: the dielectric function(No LFC— faster)

You can calculate $\epsilon$ without LFC by **epsPP_lmfh**. It is very faster than `eps_lmfh`.

To calculate $\epsilon(\mathbf{q}, \omega)$ without LFC accurately, the best basis set for the expansion of the Coulomb matrix within MT is apparently not the product basis, but the Bessel functions corresponding to the plane waves $\exp(i\mathbf{qr})$. We use such a basis in this mode. However, our experience shows that the changes are little even with the usual product basis (we don't describe this here).

### 3.3 How to calculate correct dielectric funciton?

(this subsection is essentially OK... but need to clean it up. dec2014)

```
There are prolems to calculate correct epsilon.
At first, we talk about epsPP_lmfh, which is No LFC. Main problem are

-----------------
1.Convergence for number of k point(specified by n1n2n3).
  Roughly speaking, 20x20x20 is required for not-so-bad results for Fe and Ni.
  It is better to do 30x30x30 to see convergence check.
  However, in the case of ZB-MnAs (maybe because of simple structure around Ef),
  it requires less q points.

  figs are for GaAs.
  fig001: n1n2n3 convergence for Chi_RegQbz = on  case.
  fig002: n1n2n3 convergence for Chi_RegQbz = off case.
  (Chi_RegQbz in explained in General section in this manual).
```

As you see, k points convergence looks a little better in Chi_RegQbz=off
(mesh not including gamma). However a little ploblem is that its thereshold around
0.5eV is too high and slowly changing.

fig003: Alouanis'(from Arnaud) vs. ``Chi_RegQbz = on'' vs. ``Chi_RegQbz = off''
As you see, the threshold of the Red line (20x20x20 Chi_RegQbz=on) and Alouani's
are almost the same, but the red line is too oscilating at the low energy part.
On the other hand, ``Chi_RegQbz = off'' in Green broken line is not so satisfactory
at the low energy part.

fig.gas_eps_kconf.pdf shows the convergence behevior of epsilon for

2.$q \to 0$ convergence (this is related to whether Chi_RegQbz=on or off).
If you use very small q like q=0.001 is GaAs, it can cause a problem.
Use q=0.01 or larger (maybe q=0.02 or more is safer).
Very small q can give numerical error for high-energy region.

In fig004, we show the high energy tail part of Im $\epsilon(\omega)$ for GaAs case.
At q=0.01 (this means q= 2*pi/alat * (0 0 0.01)), the imaginary part
is a little too large . Less than 80eV, q=0.02 gives good results when compared with
other high q results, though it still has noise above 80eV.
In fig005, I showed the same results compared with Alouani's (his is up to 40eV).
Both gives rather good agreements. As you see, q=0.06 or above might be necessary
to get reasonable convergence for high energy part abouve 40eV.

We have to be careful for this poorness in high energy part--- it may effect
low-energy Re[$\epsion$] through KK relation. However this can be very small
ehough.
In fig.gas_eps_qconv.jpg, we checked the convergence of eps (\omega=0,q) for q \to 0.
As you see, it gives convergence, however, q=0.01 is a little out of
curve---this should be because of the poorness in the high energy part.
so q=0.02 or q=0.03 is safer, and you can get eps within 1 percenr accuracy.

3. Including Core for dielectric constant is dangerous.
It can cause very poor results if you include core part in GWinput.
You need to include core just as valence (with local orbital).

In fig008, we showed core effects. It starts from \approx 16eV

(this is core to conduction transition).
fig007 showd the check about the q point dependence---even with large q,
it would not change.
These shows that the core excitation can have larger energy range.
This is in contrast to the valence case
(then the most of excitaion is limited to less than 10eV).
We have to be careful for such high-energy exciation... The LMTO basis might
be not so good for high energy.

4. basis set.
Use QpGcut_psi \approx 3.0 a.u. or so (as same as GW calculation).
In the case of epsPP* mode,
QpGcut_cou can be very small--- In our codes now,
ngc>=1 should be for all q vector shown in lqg4gw02 (output of echo 2|qg4gw).
[In principle, it should be only for the q vector for which we calculate epsilon.
But there is a technical poorness in our code---
(maybe) a problem here; the plane-wave part of the eigenfunction generated
in lmfgw is not correctly passed to lmf2gw when ngc=0].

-- eps_lmfh: including LFC ----------
To include eps with LFC, do eps_lmfh.
But lcutmx=2 seems to be good enough to get 0.5 percent error (maybe better than this).
Test it 10x10x10 or so. (I need to repeat if necessary).
Further you can use smaller QpGcut_cou like 2.2 or so,
with rather smaller product basis (up to p timed d, not including f).

Note: epsPP_lmfh is designed to use good basis to calculate eps
without LFC. This is usually in agreement with what you obtained by eps_lmfh;
however it can give slight difference when you use small product basis.

---Summary ----------
So in conclusion, I think a best way to do is

1. set q=0.02 [q=2pi/alat(0 0 0.02)] or so for GaAs case.
If you want to check, do q=0.03 and q=0.06 also.

``Chi_RegQbz = off'' is better for matrials like GaAs with direct gap.

2. You can use small QpGcut_cou but all ngc should be one or more.

3. As for the Product basis setting in epsPP* scripts, only
   lcutmx and tolerance (this can be like 0.001 or so) are relevant.to determine eps(omega=0, q
   E.g. set lcutmx=4 or so.

   5. To get eps with LFC, set QpGcut_cut as xxx, and set lcutmx=2 where
   4. Do nk=20 18 16 and take interpolarion

   (occupied sp) \timex (unoccupied spd) are included.
   But correct EPS*.nolfc.d is rather from epsPP_lmfh script.

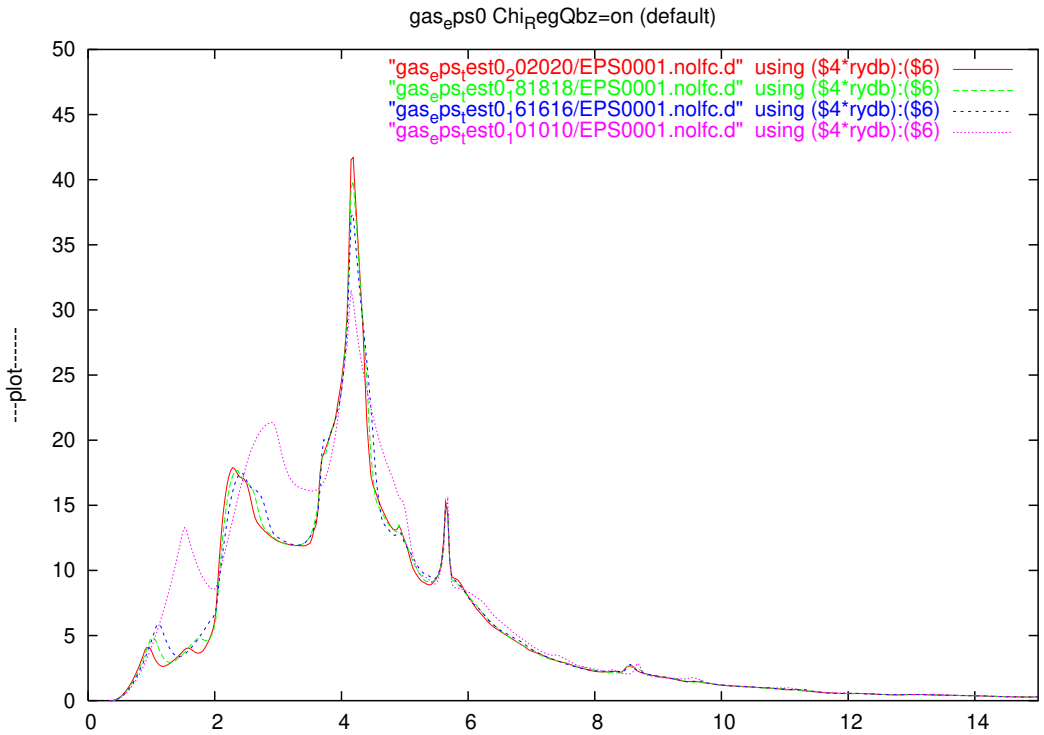

fig001

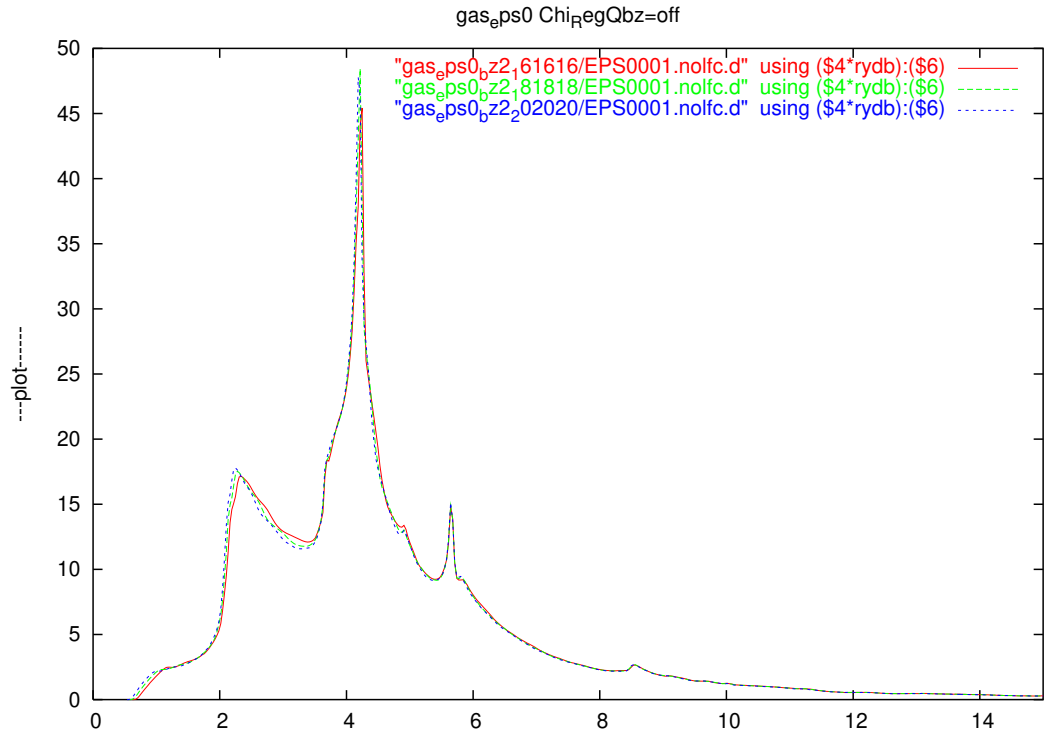


fig002

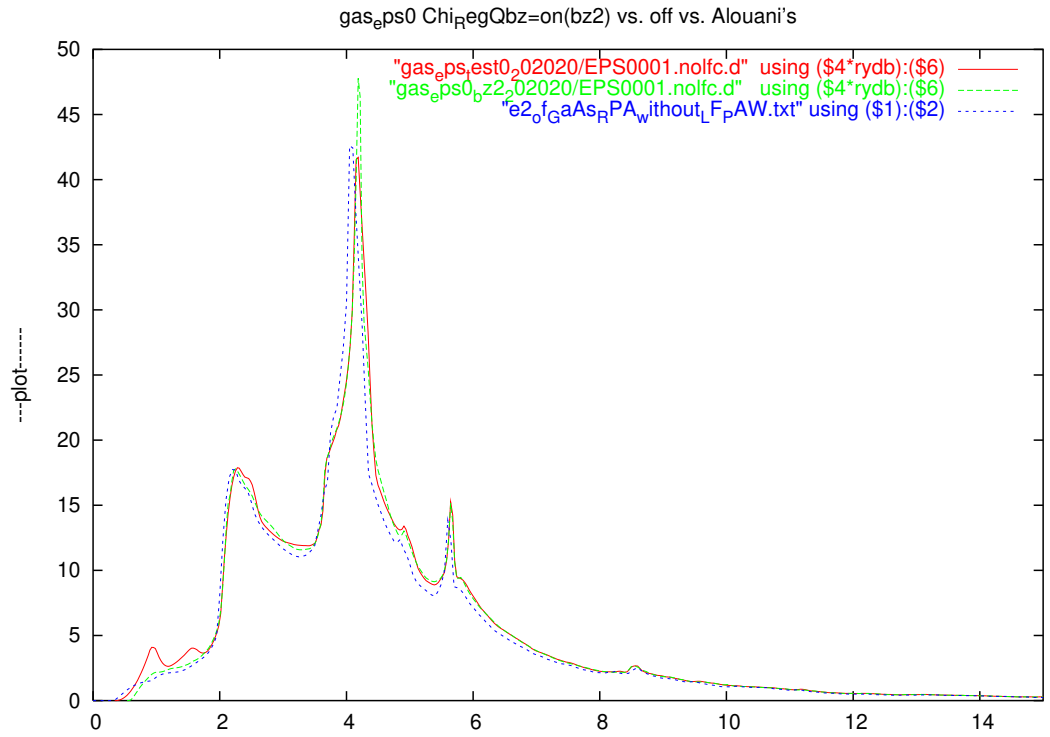


fig003

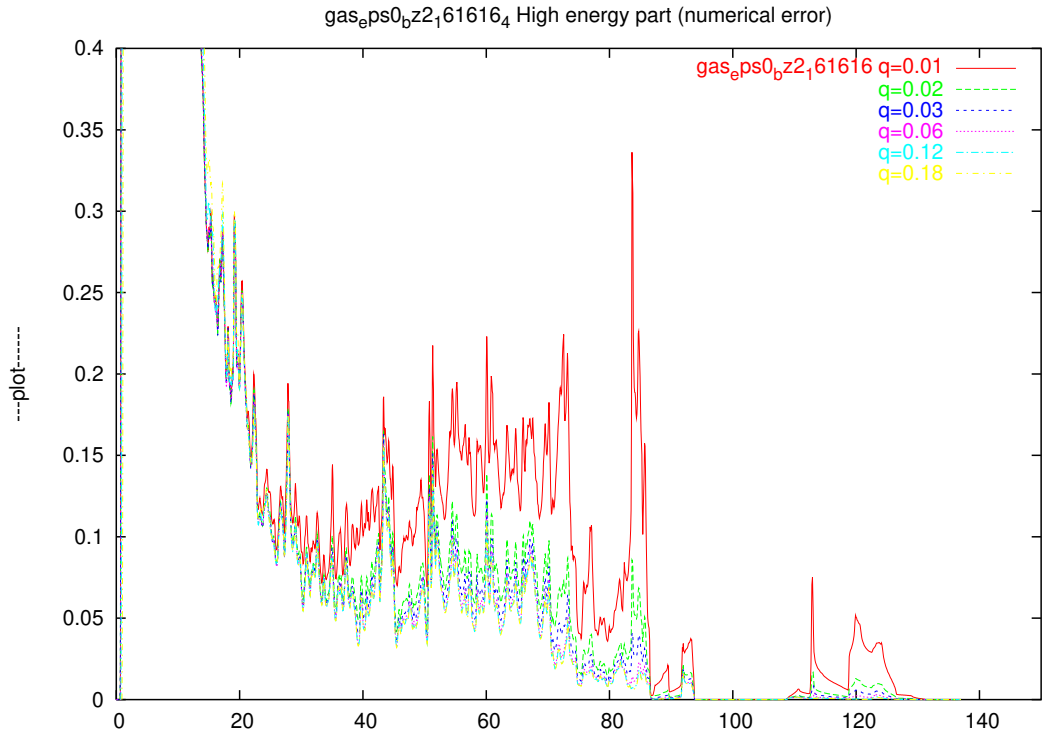


fig004

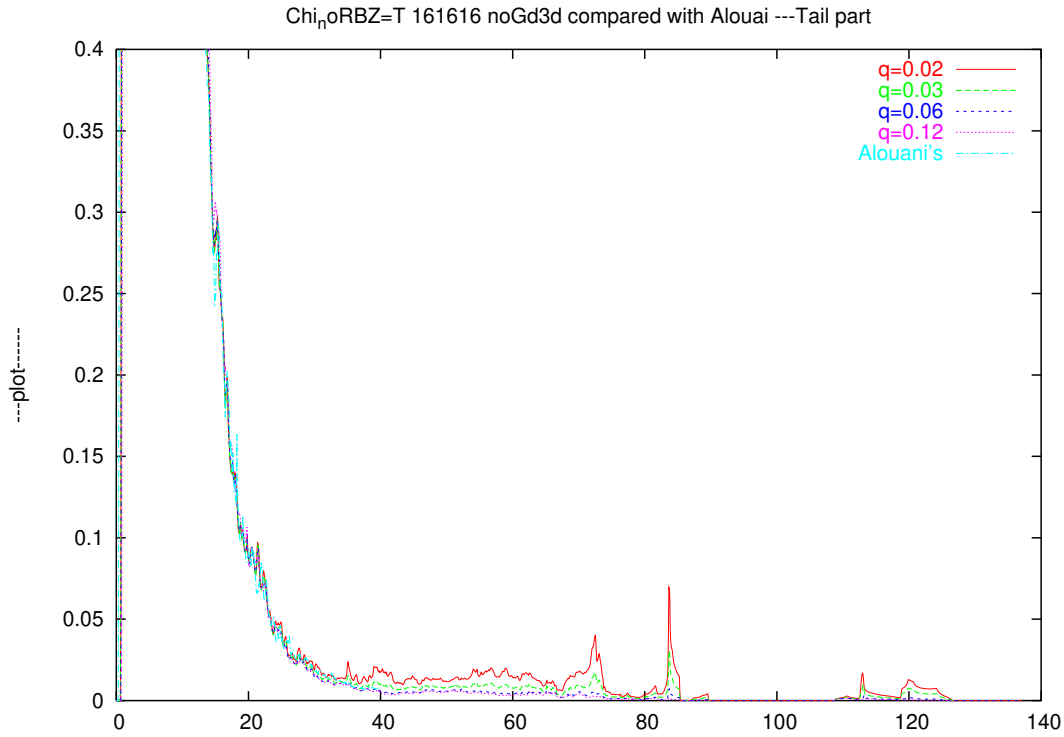


fig005

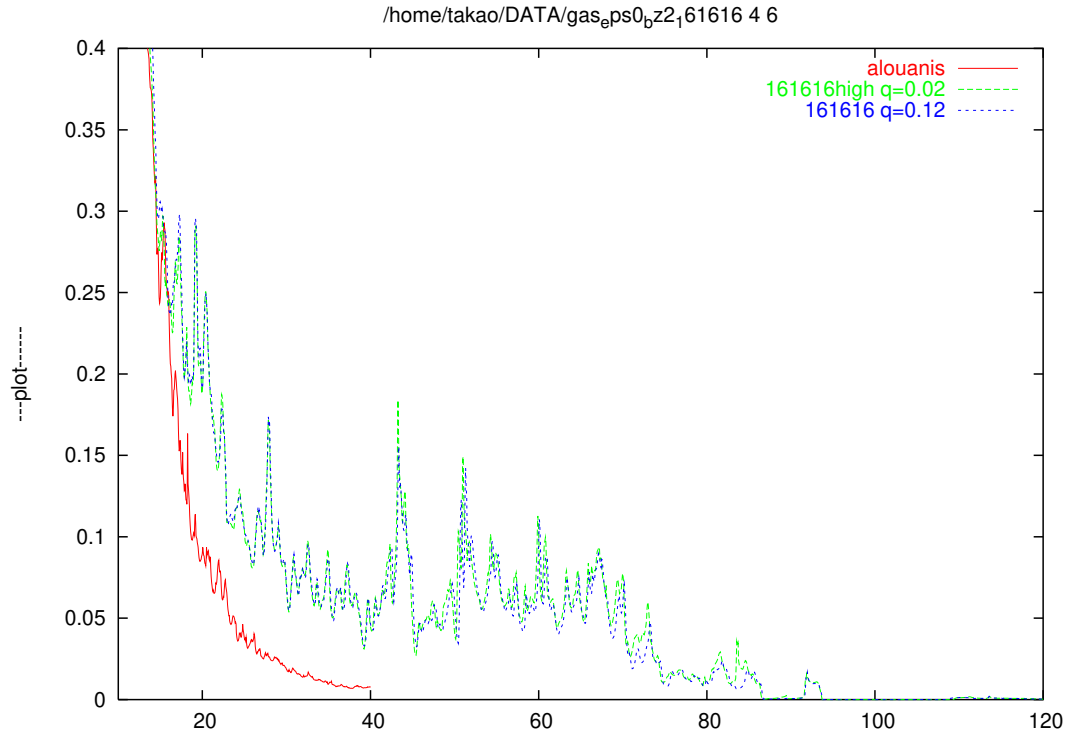


fig007

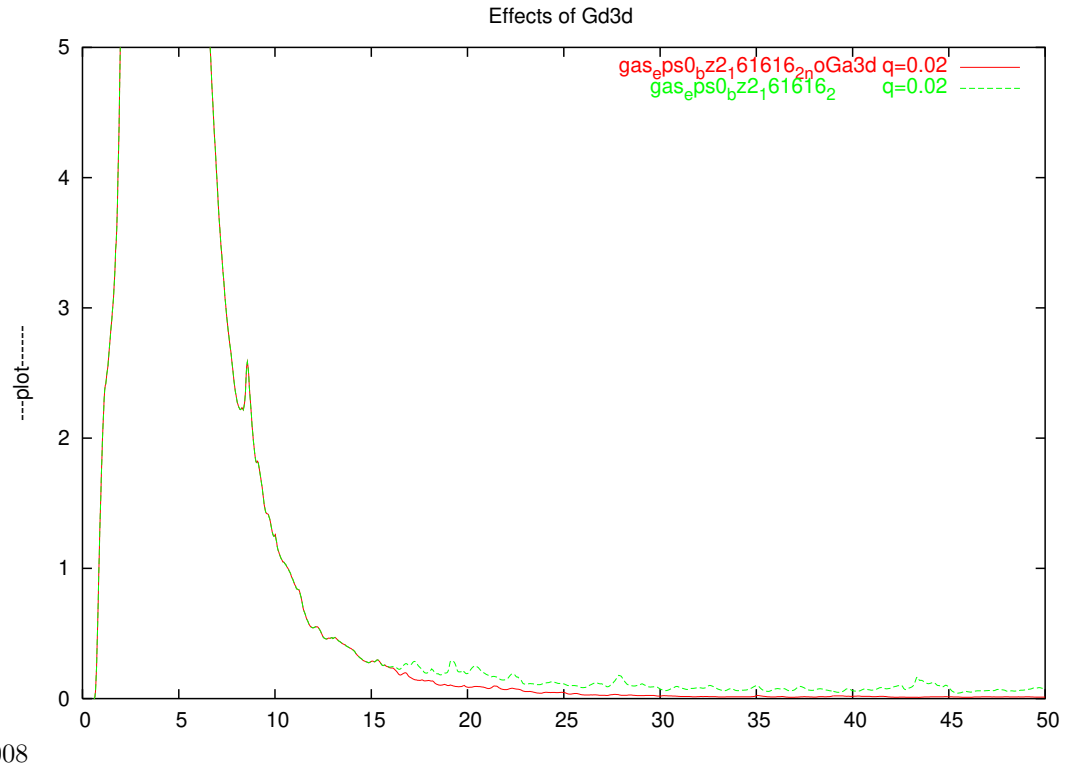


fig008