

GWinput

GWinput sets the computational conditions for GW/QSGW calculations.

mkGWinputによる自動生成

- GWinput は 自動生成することができ、殆どのパラメータはデフォルト設定で利用できる。

自動生成コマンド

```
mkGWinput $target
```

[!INFO]

mkGWinput はpython スクリプトだが内部でecalj packageに含まれている実行ファイルを実行する。

そのため ~/bin ディレクトリ(ecalj package のバイナリinstall ディレクトリ)へのPATHが通っている必要がある。

これにより GWinput.tmp が生成される。

```
cp GWinput.tmp GWinput
```

とし, GWinput を用意する。

Format

- !で開始する行はコメント
- keyword value(s) の形式であり記載順序は問わないが, 同じkeywordを複数記述しないようにする

Parameters in GWinput

生成されたGWinputに, パラメータの意味は記載されていますが, 以下に重要な変数と記載がない変数について記述します。

$n_1 n_2 n_3$

\$GW\$自己エネルギー計算におけるk点の数。ctrl file に記載するk点とは異なる点を指定できる。計算コストは, このk点数の2乗に比例することに注意。

- **type** : integer list

We set number of k points for self-energy in GWinput. These can be 1/2 or 2/3 of k points specified in ctrl, in order to reduce computational time.

The 6x6x6 k points is feasible setting for ZB structure (2 atoms per cell). That is, $6 \times 6 \times 6 \times 2 = 432 \sim k$

`points \times atom number.`

For example, when we try 8 atoms per cell, 4x4x4 or 3x3x3

is fine because $4 \times 4 \times 4 \times 8 \sim 400$.

For metallic systems, larger is fine, but limited by computational

time. See takao kotani's papers, for example, <https://doi.org/10.1103/PhysRevB.93.075125>

Good news is that we don't need to use so many k points as in ctrl.

In cases, 4x4x4 for ZB is not so bad --- roughly speaking, this is a low limit for

publication. This means 3x3x3x8 (for 8 atom case) is not so bad. ($3 \times 3 \times 3 \times 8 > 4 \times 4 \times 4 \times 2$).

See our examinations shown in <https://doi.org/10.7566/JPSJ.83.094711> and

<http://doi.org/10.7567/JJAP.55.051201>

KeepEigen

波動関数をメモリに保持するかどうか。波動関数は**CPHI** および**GEIG**というファイルで出力される。それらが1MPIが使用できるメモリに対して, 同等もしくはそれ以上である場合は必ず**.false.**にする。

\$k\$点や軌道数が多い場合は**.false.**にする。

- **type:** boolean
- **default:** .true.

MEMnmbatch

バッチサイズ:単位GB。1MPIが使用できるメモリサイズの3分の1から4分の1程度を記載すると良い。

zmel_max_size とほぼ同様な意味をもつ。自己エネルギーの計算で使用される。大規模系では大きい方が計算は早くなるが、メモリ不足で計算が落ちる場合は小さくする。

- **type:** float
 - **default:** 2.0
- 系に対して値が小さいと"sxcf_fal2_count.sc. Too small memory for nmbatch mechanism. Enlarge GWinput MEMnmbatch" との出力が**stdout.xxxx.hsfp0.sc.mode.yyy**に出力され計算が終了する場合がある。その場合は大きくする。

zmel_max_size

PROF

バッチサイズ:単位GB。1MPIが使用できるメモリサイズの3分の1から4分の1程度を記載すると良い。

MEMnmbatch とほぼ同様な意味をもつが, \$W\$の計算で使用される。大規模系では大きい方が計算は早くなるが、メモリ不足で計算が落ちる場合は小さくする。

- **type:** float
- **default:** 1.0

KeepPpb

MT内の積基底とMT内波動関数基底の行列要素(**ppb**変数)をメモリに保持するかどうか。通常はデフォルト値**.true.**で良いが, 使用メモリ量を軽減させたいときに**.false.**とする。

- **type:** boolean
- **default:** .true.

GaussianFilterX0

密度応答関数の計算に用いられる振動数におけるスミアリングパラメータ: 単位 a.u.^2 (Hartree 2)。QSGW計算が不安定な場合(固有値が振動するなど)の際に用いる。

- **type:** float
- **default:** 0
- **examples:** GaussianFilterX0 0.0001

esmr

In (Ry). this is the smearing of poles of G for $G \times W$. To reduce the effect of the sharp cutoff of Fermi energy, use larger esmr (maybe stabilized).

GaussSmear on

We usually use Gaussian smearing. Rectangular smearing otherwise.

Poles of the Green function G are treated as if they have width esmr in $G \times W$.

If GaussSmear is on, each pole of G is smeared by a Gaussian function with $\sigma = \text{esmr}$ in the calculation of $h_{\text{sf}0}$. If GaussSmear is off, we assume rectangular smearing for the poles. Usually it is necessary to take rather smaller value than band gap for insulators. Try to use 0.003 or so in the case of Si and GaussSmear=on.

For metal, this esmr is somehow related to how we capture the Fermi surface; In principle, we have to take the limit $n_1 n_2 n_3 \rightarrow \infty$ and $\text{esmr} \rightarrow 0$. However, we may inevitably use some finite esmr to make calculations converged.

unit_2pioa

unit_2pioa off ! off \rightarrow a.u.; on \rightarrow unit of $Q_{\text{pGcut_}}$ are in $2\pi/\text{alat}$
普通offでよい.

alpha_OffG

1.000 ! (a.u.) Used in auxiliary function in the offset-Gamma method.
Not need to change.

QpGcut_psi

波動関数のIPW基底のカットオフエネルギー: 単位 Ryd. **lmf**計算で用いるPMT手法とは波動関数の基底関数が異なることに注意。原子間領域の波動関数はIPWで展開し直して記述される。IPWとはinterstitial plane wave.

- **type:** float
- **default:** 4.0

波動関数のIPWのカットオフctrl内のpwemaxとも関係する。pwemaxとしては2,3を試すことが多い(単位 Ryd)。

$|q+G| < Q_{\text{pGcut_psi}}$ 単位 $1/\text{a.u.}$

個数はqg4gwの結果 (lqg4gw)にかかっている。

QpGcut_cou

積基底のIPW部分のカットオフエネルギー: 単位 Ryd
 $|q+G| < QpGcut_cou$ 単位 1/a.u.

- **type:** float
- **default:** 3.0

個数はqg4gwの結果 (lqg4gw)にかかっている。

emax_sigm

フェルミ準位から測った, 計算する準粒子状態の自己エネルギーの最大値: 単位 Ryd。QSGW計算では自己エネルギーはポテンシャル構築に使用されるため小さくするとポテンシャルの精度も下がる。

- **type:** float
- **default:** 3.0

HistBin if High resolution energy mesh near Ef for metal

This defines histogram bins to accumulate weight (imaginary part) of the polarization functions.

For metal, it might be better to test

```
HistBin_dw      1d-5 ! 1d-5 is fine mesh (good for metal?) !(a.u.)
BinWidth along real axis at omega=0.
HistBin_ratio 1.03 ! 1.03 maybe safer. frhis(iw)= b*(exp(a*(iw-1))-1),
where a=ratio-1.0 and dw=b*a
```

m_freq.f90のgetfreqでこのメッシュを作っている。

- HistBin_dw and HistBin_ratio specify real space bins which we accumulate imaginary part weight of polarization functions. The bins are (see frhis in ecalj/fpgw/gwsc/m_freq.F)
$$\omega_i = b * (\exp(a * (i - 1)) - 1) / a$$

where $[\omega_i, \omega_{i+1}]$ $(i = 1, 2, \dots, nwhis+1)$ is the i-th bin. HistBin_dw is bin width at $\omega = 0$.
The ratio $\omega(i+1)/\omega(i)$ for large ω is $\exp(a) = \text{HistBin_ratio}$. This choice of getting coarser at high energy is because we think $W(\omega)$ around $\omega \sim 0$ gives most important contribution to the GW approximation. If histogram bins are too wide, dielectric function can be less accurate, but results may be not so much affected.
In GW calculation, the Plasmon pole is important. It is determined not by the Drude weight; it usually gives small contribution to the Plasmon pole (for example, Si is described well by a Plasmon pole model, but Si has no Fermi surface). We expect that the GW results are not so sensitive to the choice of HistBin_dw, HistBin_ratio usually. We may use fine mesh when we plot quantities such as $W(\omega)$ near $\omega = 0$.

niw

Number of frequencies along Im axis. Used for integration along imaginary axis. Probably 10 is good enough.

See routines `wintzsg.f90`.

The integration points are $\omega'(n) = i(1/x(n) - 1)$, where $x(n)$ is the usual Gaussian-integration points for the interval $[0,1]$. In addition, we give the special analytical treatment for the peaky part at $\omega' = 0$. Our tests show $n_{iw}=6$ for Si is good enough for 0.01 eV accuracy. The convergence as for n_{iw} is quite good. This integration scheme has been developed by Ferdi Aryasetiawan. The number of points should be the one of 6, 10, 12, 16, 20, 24, 32, 40, or 48, because we use a subroutine `gauss` in `mate.f90` prepared by Ferdi.

Q0PChoice (This is not used anymore)

We now use offset Gamma Q0P which is ten times smaller than regular mesh. See `lqg4gw` and `Q0P` file.

`deltaw`

real (a.u.) only for one-shot case.

`deltaw` is the interval for the numerical derivative $\partial\Sigma(\omega)/\partial\omega$. We calculate $\Sigma(\omega), \Sigma(\omega \pm \text{deltaw})$.

From these values, we can calculate two Z (or second-derivative of $\Sigma(\omega)$).

`<QforGW>` section

This is only for one-shot/spectrum function mode.

In this section set `q` vector (in the unit of `2pi/alat`. See `BZ.html` and `syml` file to check).

If no `<QforGW>`, all irreducible `q` points are used. (see `m_getqforgw.f90` L64 for `ret=0`)

EMAXforGW

one-shot GW. Set this (eV, above the Fermi energy) to specify up to which bands you calculate. (we have `EMINforGW`).

PROF

`-- readQforGW` option

We can use any `q` in the `QforGW` section

Because of the shifted-mesh method, we can use any `q` point which is not on the mesh points.

If on the mesh point, we need to prepare less number of eigenfunctions for GW calculations.

`<QforEPS>`

This is for dielectric function mode. Set as

```
0 0 0.00050
0 0 0.00100
0 0 0.00200
```

Because of the shifted-mesh method, we can use any q point which is not on the mesh points.

QforEPSau

Whether q is in the unit of $2\pi/\text{alat}$ or in the unit of a.u.(= bohr^{-1}). I think QforEPSau on might be convenient.

Smaller lcutmx to reduce computational time

To reduce the computational time, we reduce number of MPB (mixed product basis).

One is lcutoff of Product Basis (PB) within MT. Use lcutmx=2 for oxygen or something (s,p block atoms).

Thus it is like

```
lcutmx(atom) = maximum l-cutoff for the product basis.  =4 is required
for atoms with valence d, like N
4 4 4 2 2 2
```

in the section of GWinput. Atom ID is in SiteInfo.foobar. The ording of SITE in ctrl (need check).

- NOTE: we need lcutmx =6 is required for 4f/5f atoms, while 2 is enough for Oxygen (probably other species on the same row).

Smaller IPW related part to reduce computational time

To reduce computational time, we may reduce MPB coming from IPS, reduce the size of IPW for psi, reduce emax_sigm and pwemax as follows.

- QpGcut_cou is for the Interstitial plane wave (IPW) for MPB.
- QpGcut_psi is for expansion of eigenfunctions.
- emax_sigm is the upper cutoff (relative to the Fermi energy) to calculate self energy.
- pwemax (in ctrl) is the APW basis cutoff for the eigenfunciton.

—
PROF

To reduce computational time, we may use

```
QpGcut_psi 3.0
QpGcut_cou 2.5
emax_sigm 2.0
```

In addition, we may use

```
pwemax=2 (in ctrl file).
```

This is for the APWs in the band calculation.

We sometimes use this setting. It is better to check how the numerical results are affected.