Introduction to the FlapwMBPT code

Andrey L. Kutepov
Brookhaven National Lab, Upton, NY

This work was supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences as a part of the Computational Materials Science Program.

COMSCOPE
from codes to spectroscopies

BROOKHAVEN
NATIONAL LABORATORY

Plan

- Hedin's set of equations as theoretical background
- Useful approximations:
- **scGW – self-consistent GW approximation**
- **QSGW – quasiparticle self-consistent GW**
- **sc(GW+G3W2) – first order vertex corrections, sc(BSE:P@GW+G3W2) – BSE for polarizability**
- A little bit of details (basis set, complexity…)
- Examples of the calculations
- Hands on training
- Homework (optional)

Hedin's equations

$$\Gamma^\alpha(123) = \delta(12)\delta(13) + \sum_\beta \frac{\delta\Sigma^\alpha(12)}{\delta G^\beta(45)} G^\beta(46)\Gamma^\beta(673)G^\beta(75)$$

$$P(12) = \sum_\alpha G^\alpha(13)\Gamma^\alpha(342)G^\alpha(41)$$

$$W(12) = V(12) + V(13)P(34)W(42)$$

$$\Sigma^\alpha(12) = -G^\alpha(14)\Gamma^\alpha(425)W(51)$$

$$G^\alpha(12) = G_0^\alpha(12) + G_0^\alpha(13)\Sigma^\alpha(34)G^\alpha(42)$$

This system of equations is supposed to be solved self-consistently. As written, it is exact (neglect e-ph). In practice, approximations are needed.

L.Hedin
PR 139, A796 (1965)

G.Strinati
Rivista del Nuovo Cimento
11,1 (1988)

GW approximation

$$\Gamma^{\alpha}(123) = \delta(12)\delta(13)$$

$$P(12;\tau) = -G(12;\tau)G(21;\beta-\tau)$$

$$W(12;\nu) = V(12) + \int d(34)V(13)P(34;\nu)W(42;\nu)$$

$$\Sigma(12;\tau) = -G(12;\tau)W(21;\tau),$$

$$G(12;\omega) = G_0(12;\omega) + \int d(34)G_0(13;\omega)\Sigma(34;\omega)G(42;\omega)$$

The approximation was first considered by L. Hedin (1965) (electron gas). Applications to real materials (non-self-consistent) first appeared in 1980's:

G. Strinati et al., Phys. Rev. B **25**, 2867 (1982).

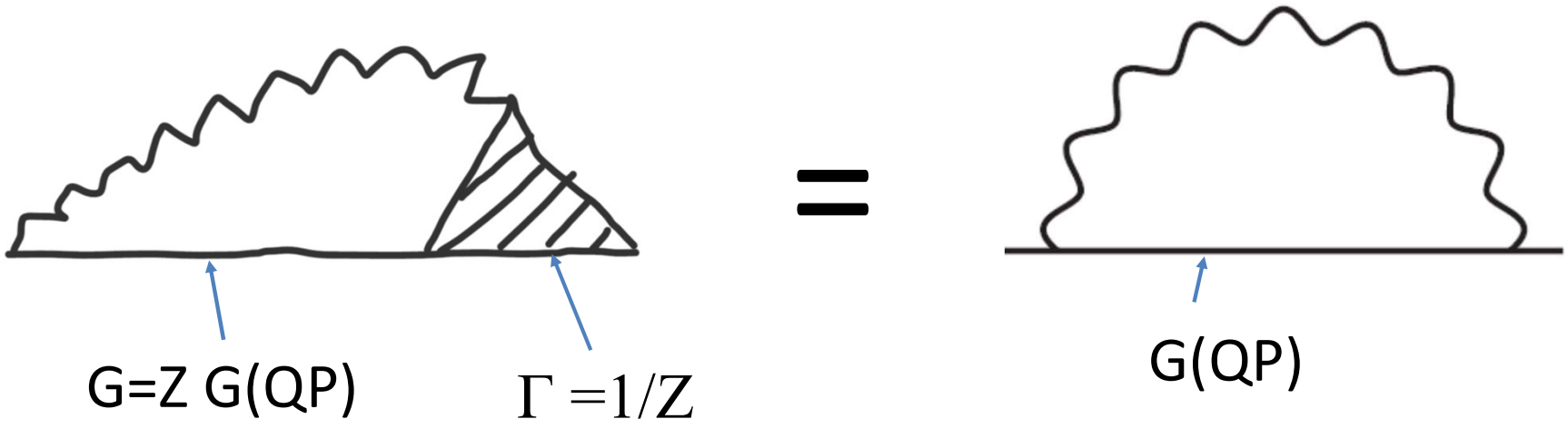
M. S. Hybertsen and S. G. Louie, Phys. Rev. B **34**, 5390 (1986).

R. W. Godby et al., Phys. Rev. B **37**, 10159 (1988).

Quasiparticle GW approximation (QSGW)

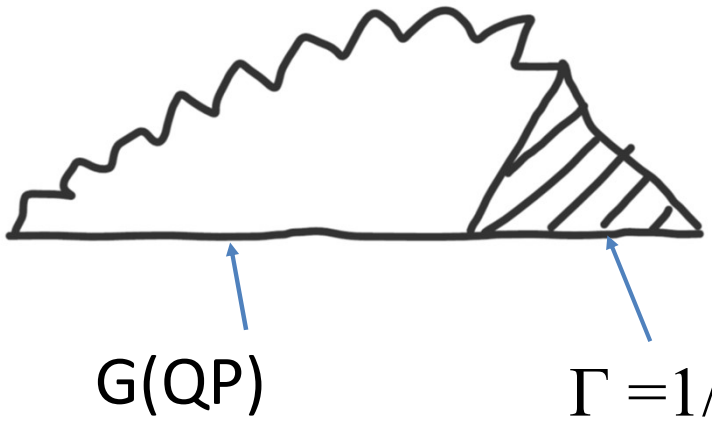
1) Based on the finite temperature (Matsubara) formalism
2) Different from the implementation by Kotani et al. [PRB 76, 165106]
3) Details: [PRB 85, 155129, CPC 219, 407]
4) Because of the so called Z-factor cancellation [PRB 76, 165106] (next slide), QSGW is often quite accurate, especially in simple metals and semiconductors.
5) In more complicated solids (especially where d or f electrons play an important role) QSGW approach is not necessary better than the scGW. Good example is metal americium, where both DFT and QSGW fail to describe the experimentally determined [PRL 52, 1834] position of the occupied 5f5/2 states whereas scGW describes them very well [PRB 85, 155129].

Z factor cancellation



PRB 76, 165106

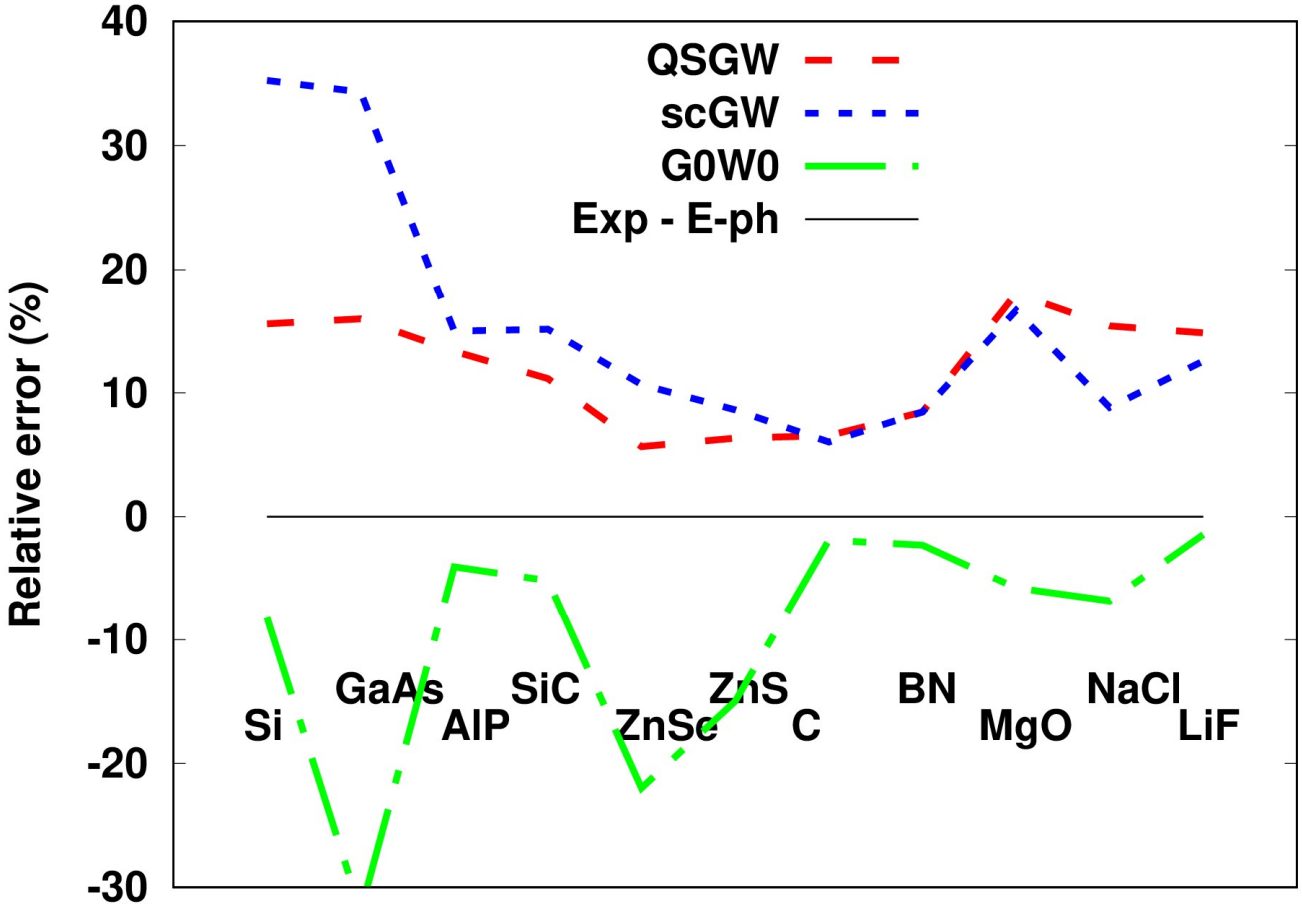
What about QSGW + Vertex?



$\sim$ **G(QP)*W / Z ???**

Direct addition of diagrams works only with scGW but not with QSGW.
By the same reason QSGW+DMFT can only be considered on the model level (similar to DFT+DMFT).

scGW versus QSGW: sp semiconductors



G0W0:
PRB 93, 115203 (2016)
scGW, QSGW:
PRB 95, 195120 (2017)

Variants of GW (brief summary)

G0W0 – one shot GW, is used in most of the cases. Least expensive but depends on starting point. Success is mostly based on error cancellation (no self-consistency and no vertex correction)

GW0 – G-only self-consistent GW: still inexpensive but often accurate (not diagrammatic). Also depends on starting point.

QSGW – consistently most accurate GW variant, but rather inconvenient for further improvements (not diagrammatic)

scGW – the variant most often criticized, but the most appropriate for further diagrammatic improvements

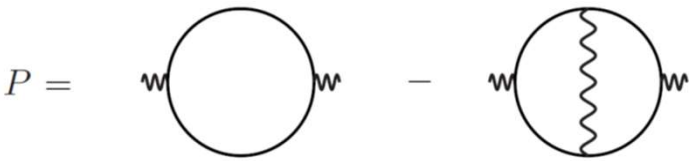
Vertex corrected schemes: pros and cons

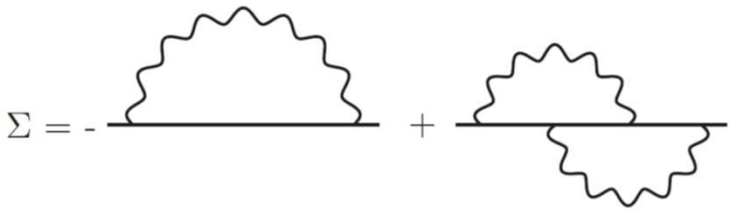
Advantages:

Systematically improve scGW
Diagrammatic – further improvements mean just adding diagrams (no double counting). This is contrary to DFT or QSGW which are non-diagrammatic and are difficult to improve.
No dependence on starting point (contrary to G0W0 or GW0)
No adjustable parameters (contrary to hybrids or DFT+U)

Disadvantage: High computational cost

sc(GW+G3W2)

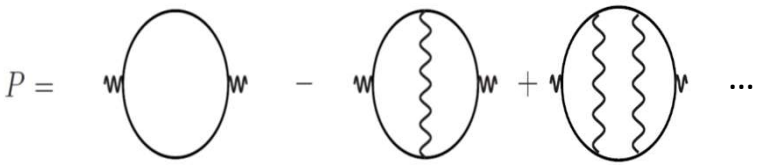


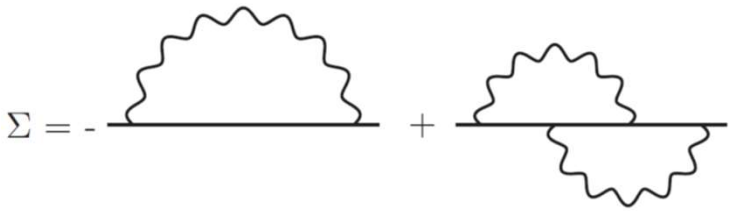


Represents natural extension of scGW by including all next order diagrams to polarizability and self energy. Always better than scGW. In weakly correlated materials it is usually all one needs to get accurate electronic structure.

Details:
PRB 94, 155101 (2016) (scheme B)
arXiv 2105.03770 (2021)

sc(BSE:P@GW+G3W2)





Systematically includes (through the Bethe-Salpeter Equation, BSE) the effects of electron-hole interaction. Noticeably better than scGW or sc(GW+G3W2) in electron gas and alkali metals. In insulators, provides improvements when excitonic effects are large (LiF as an example).

Details:
PRB 94, 155101 (2016) (scheme G)

Basis set selection

Quick overview of the basis sets

Exact

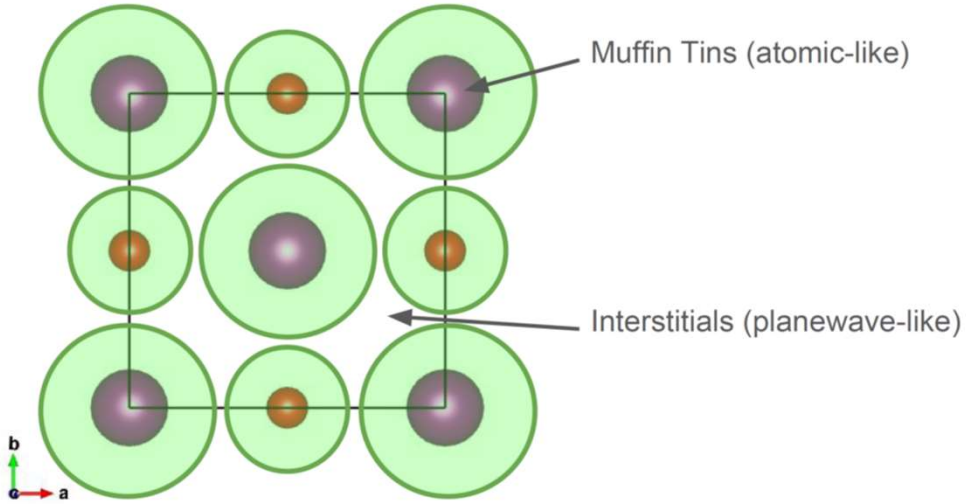
- Plane wave (PW) expansion
- Real space grid (RSG) based methods

Adapted to physical problem

- LCAO (numerical, analytical)
- LMTO
- PAW
- LAPW

LAPW basis set

$$\Psi_{\lambda}^{\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{G}} A_{\boldsymbol{G}\lambda}^{\boldsymbol{k}} \Pi_{\boldsymbol{G}}^{\boldsymbol{k}}(\boldsymbol{r})$$



$$\Pi_{\boldsymbol{G}}^{\boldsymbol{k}}(\boldsymbol{r}) = \begin{cases} e^{i(\boldsymbol{k}+\boldsymbol{G})\boldsymbol{r}}, \boldsymbol{r} \in Interstitial \\ \sum_{l,m} \left(a_{\boldsymbol{tlm}}^{\boldsymbol{G}}(\boldsymbol{k}) u_{tl}(r) + b_{\boldsymbol{tlm}}^{\boldsymbol{G}}(\boldsymbol{k}) \dot{u}_{\boldsymbol{tl}}(r) \right) Y_{lm}(\boldsymbol{r}), \boldsymbol{r} \in \boldsymbol{t} \text{ (MT)} \end{cases}$$

The coefficients (a, b) are defined by the requirement of the continuity of the value and slope of each (L, m) component at the MT spheres.

Extensions of LAPW: LAPW+LO

$$\Psi_{\lambda}^{\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{G}} A_{\boldsymbol{G};\lambda}^{\boldsymbol{k}} \Pi_{\boldsymbol{G}}^{\boldsymbol{k}}(\boldsymbol{r}) + \sum_{\boldsymbol{tnL}} B_{\boldsymbol{tnL};\lambda}^{\boldsymbol{k}} \Lambda_{\boldsymbol{tnL}}^{\boldsymbol{k}}(\boldsymbol{r})$$

$$\Lambda_{\boldsymbol{tnL}}^{\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}(\boldsymbol{t}+\boldsymbol{R})} \left(a_{lnm}^{\boldsymbol{t}} u_{\boldsymbol{t}nl}(r) + b_{lnm}^{\boldsymbol{t}} \dot{u}_{\boldsymbol{t}nl}(r) + c_{lnm}^{\boldsymbol{t}} \ddot{u}_{\boldsymbol{t}nl}(r) \right) Y_{lm}(\boldsymbol{r})$$

Local orbital extension (LO) is specifically designed to describe semi-core states and high energy excited states. But it also enhances the variational freedom in the valence region.

Extensions of LAPW: APW+lo

$$\Psi_{\lambda}^{\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{G}} A_{\boldsymbol{G}\lambda}^{\boldsymbol{k}} \widetilde{\Pi}_{\boldsymbol{G}}^{\boldsymbol{k}}(\boldsymbol{r}) + \sum_{\boldsymbol{t}L} B_{\boldsymbol{t}L\lambda}^{\boldsymbol{k}} \widetilde{\Lambda}_{\boldsymbol{t}L}^{\boldsymbol{k}}(\boldsymbol{r})$$

$$\widetilde{\Pi}_{\boldsymbol{G}}^{\boldsymbol{k}}(\boldsymbol{r}) = \begin{cases} e^{i(\boldsymbol{k}+\boldsymbol{G})\boldsymbol{r}}, \boldsymbol{r} \in Interstitial \\ \sum_{l,m} \tilde{a}_{\boldsymbol{t}lm}^{\boldsymbol{G}}(\boldsymbol{k}) u_{tl}(r)\, Y_{lm}(\boldsymbol{r}), \boldsymbol{r} \in \boldsymbol{t}\ (\mathrm{MT}) \end{cases}$$

$$\widetilde{\Lambda}_{\boldsymbol{t}nL}^{\boldsymbol{k}}(\boldsymbol{r}) = \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}(\boldsymbol{t}+\boldsymbol{R})} \left(\tilde{a}_{lnm}^{\boldsymbol{t}} u_{\boldsymbol{t}nl}(r) + \tilde{b}_{lnm}^{\boldsymbol{t}} \dot{u}_{\boldsymbol{t}nl}(r) \right) Y_{lm}(\boldsymbol{r})$$

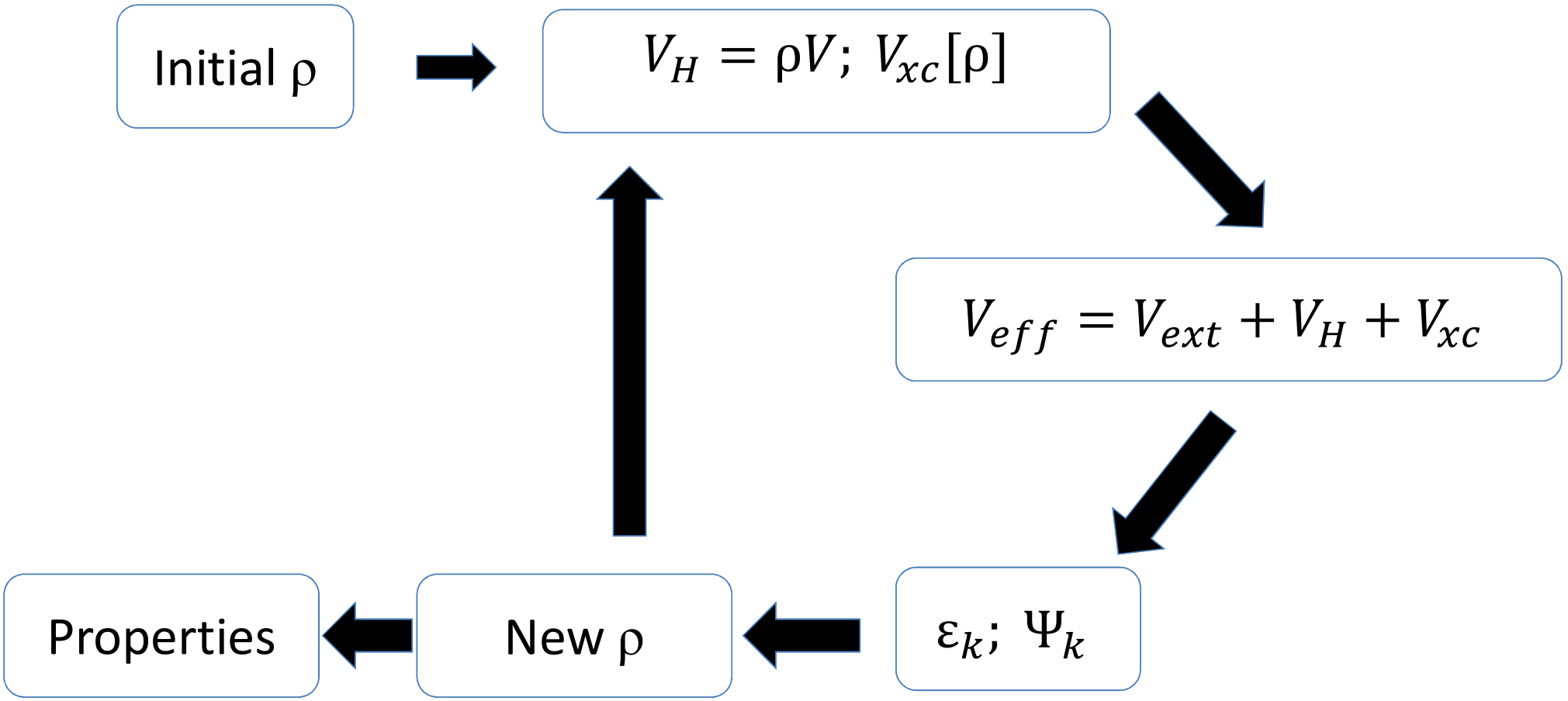
Efficiency issues

Principal advance in efficiency of scGW is based on the observation that some of the algorithmic steps are faster to do in momentum+frequency space whereas other steps in real-space+time representation.
This was originally used in the context of plane-waves [PRL 74, 1827].
LAPW-based implementation in FlapwMBPT: PRB 85,155129.

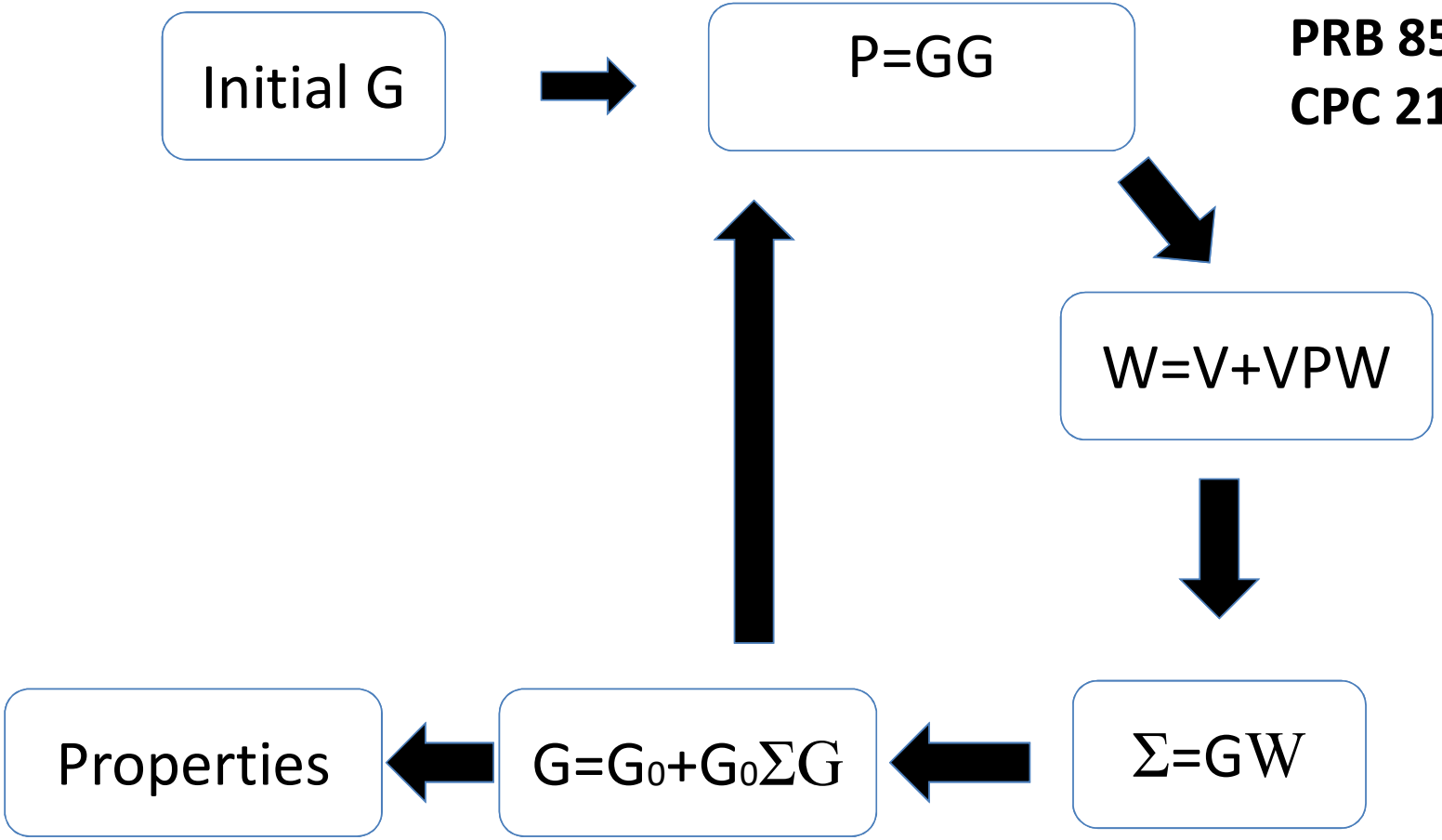
Evaluation of polarizability and self energy in sc(GW+G3W2) and sc(BSE:P@GW+G3W2) uses exactly the same idea.
Details one can find in PRB 94, 155101.

Structure of the code

DFT

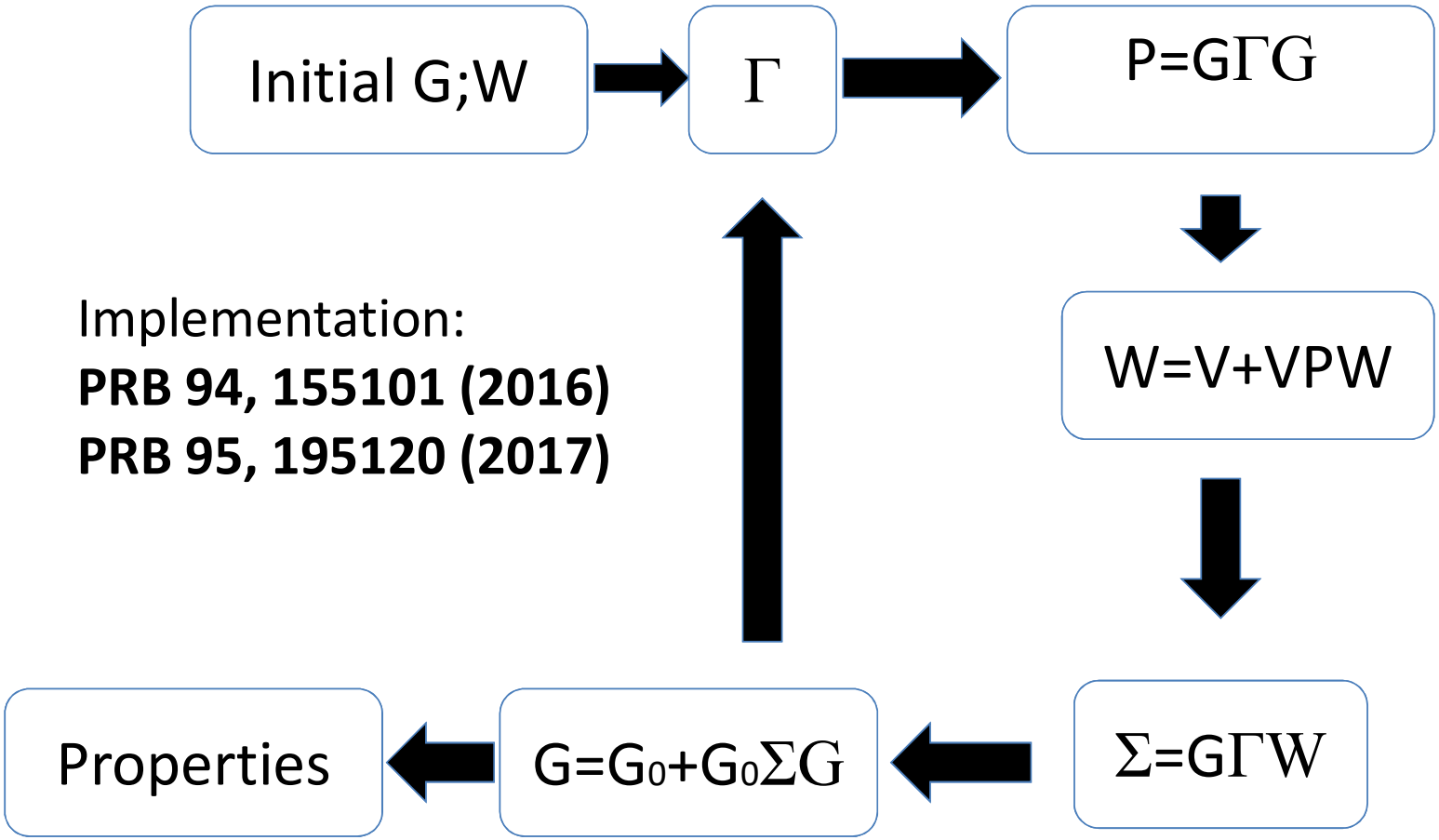

Initial ρ
$V_H = \rho V;\ V_{xc}[\rho]$
$V_{eff} = V_{ext} + V_H + V_{xc}$
$\varepsilon_k;\ \Psi_k$
New ρ
Properties

GW



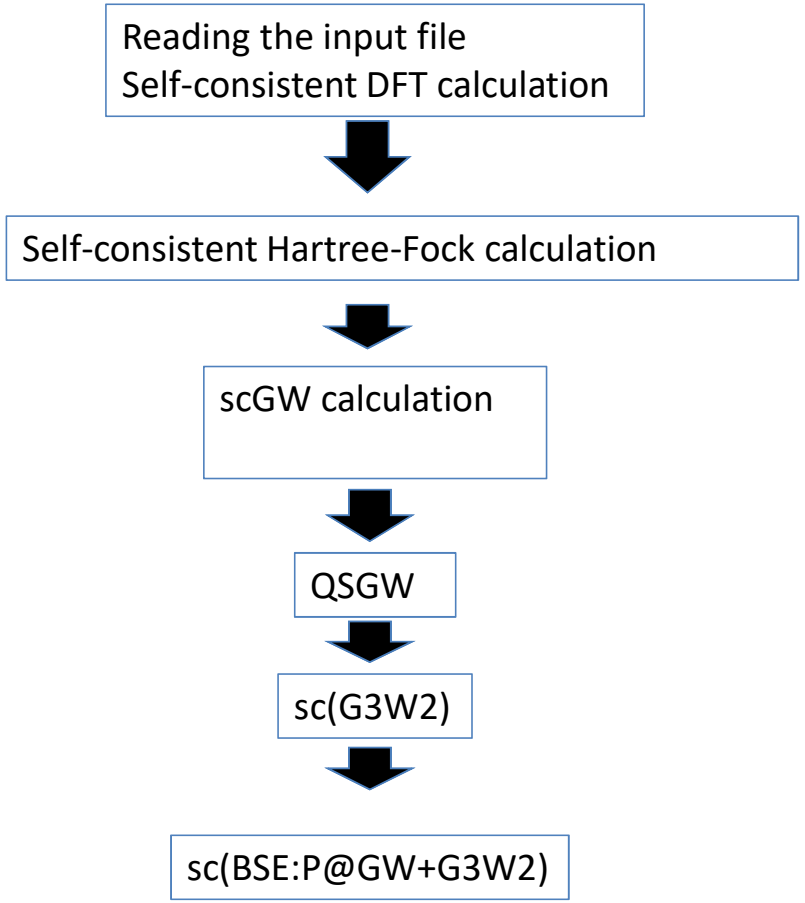
Implementation:
PRB 85, 155129 (2012)
CPC 219, 407 (2017)

Vertex corrected GW



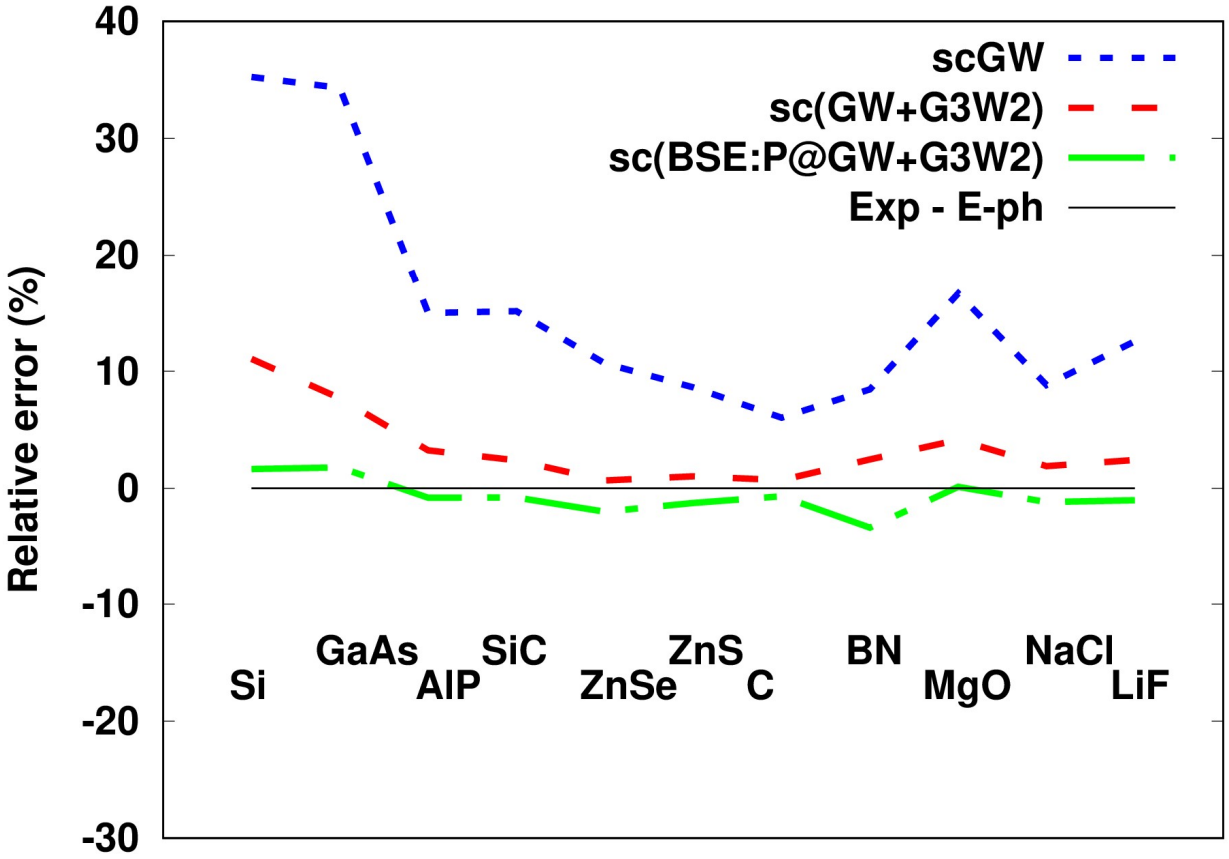
Implementation:
PRB 94, 155101 (2016)
PRB 95, 195120 (2017)

How different methods are combined in one code



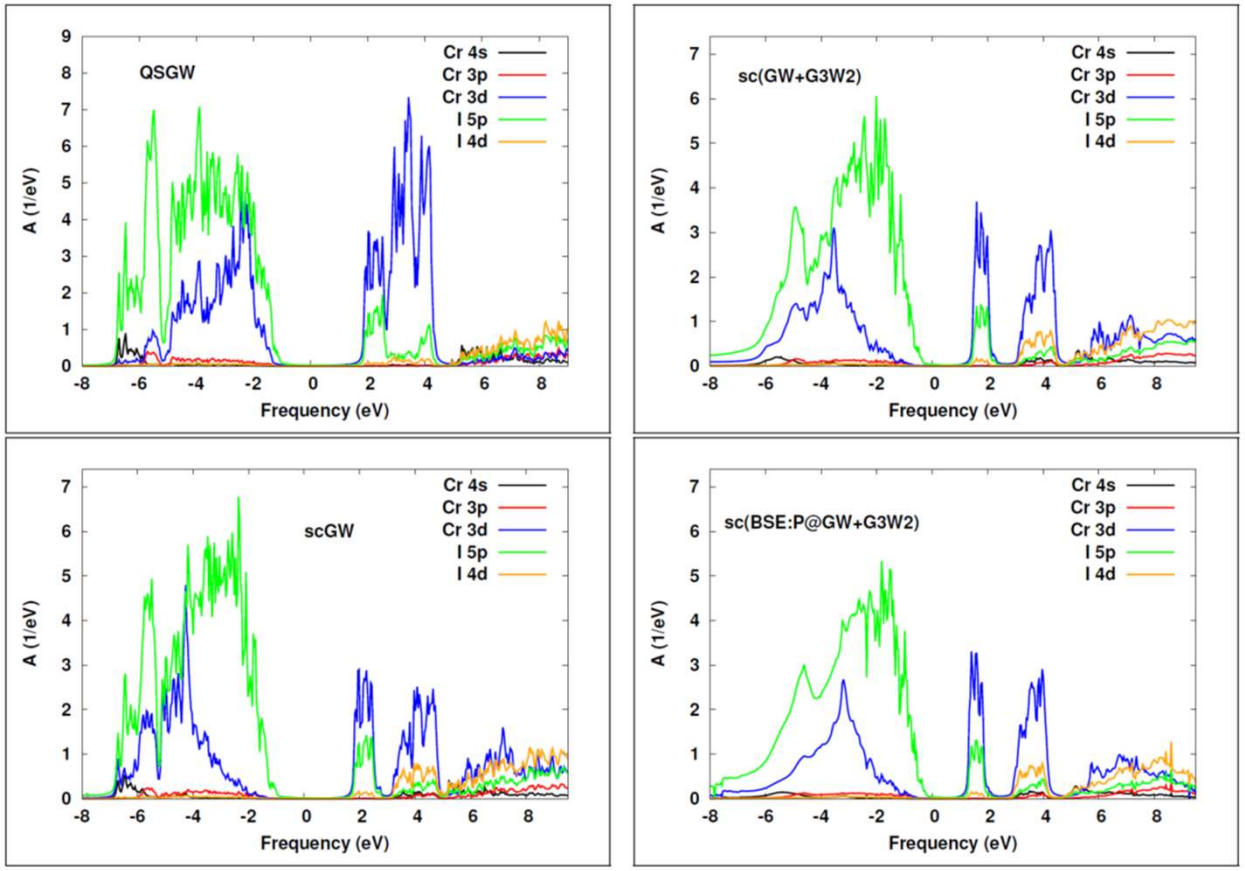
Examples of calculations

Examples: band gaps of sp semiconductors



PRB 95, 195120 (2017)

Other examples: CrI3

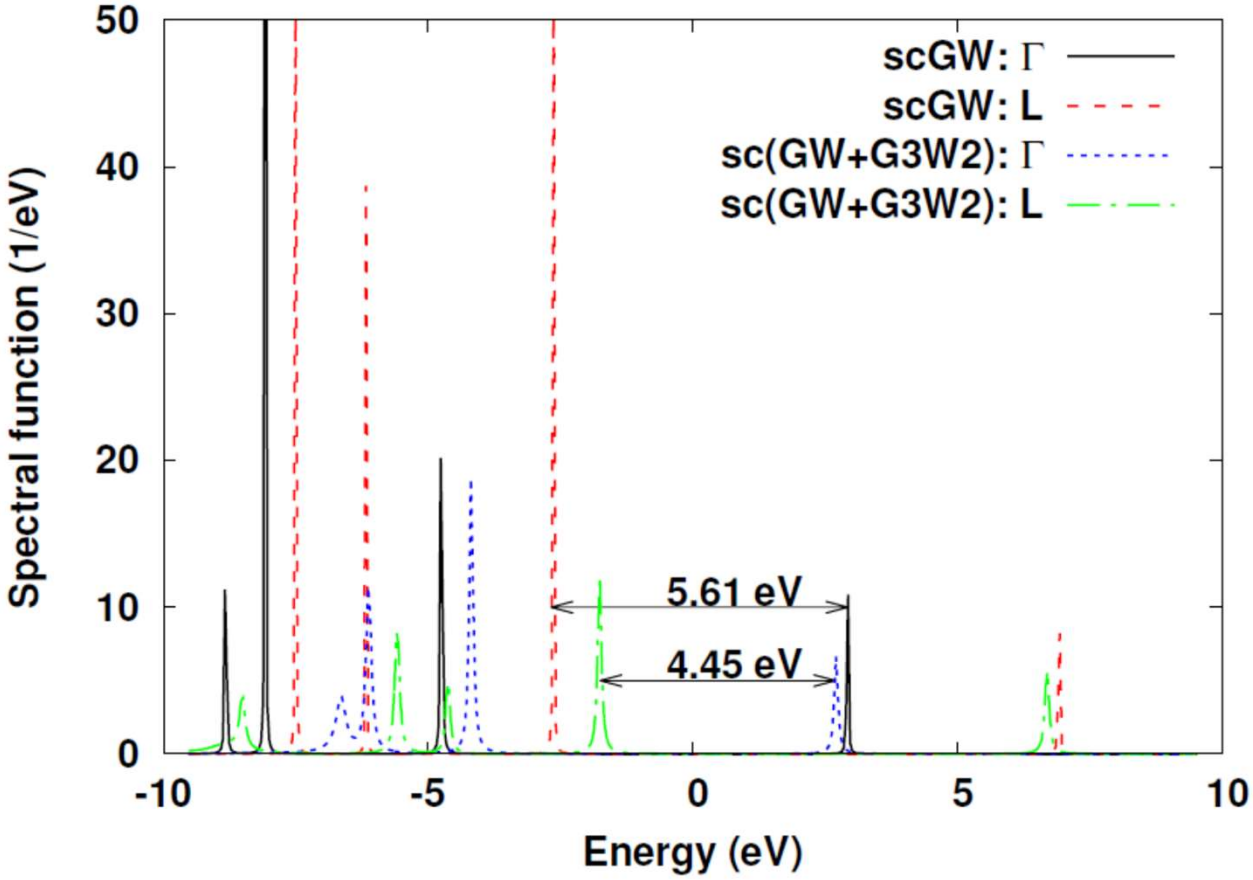


arXiv:2105.07798

Experimental band gap is ~ 1.3 eV,

A. K. Kundu, Y. Liu, C. Petrovic and T. Valla,
B. Sci. Rep. 10, 15602 (2020)

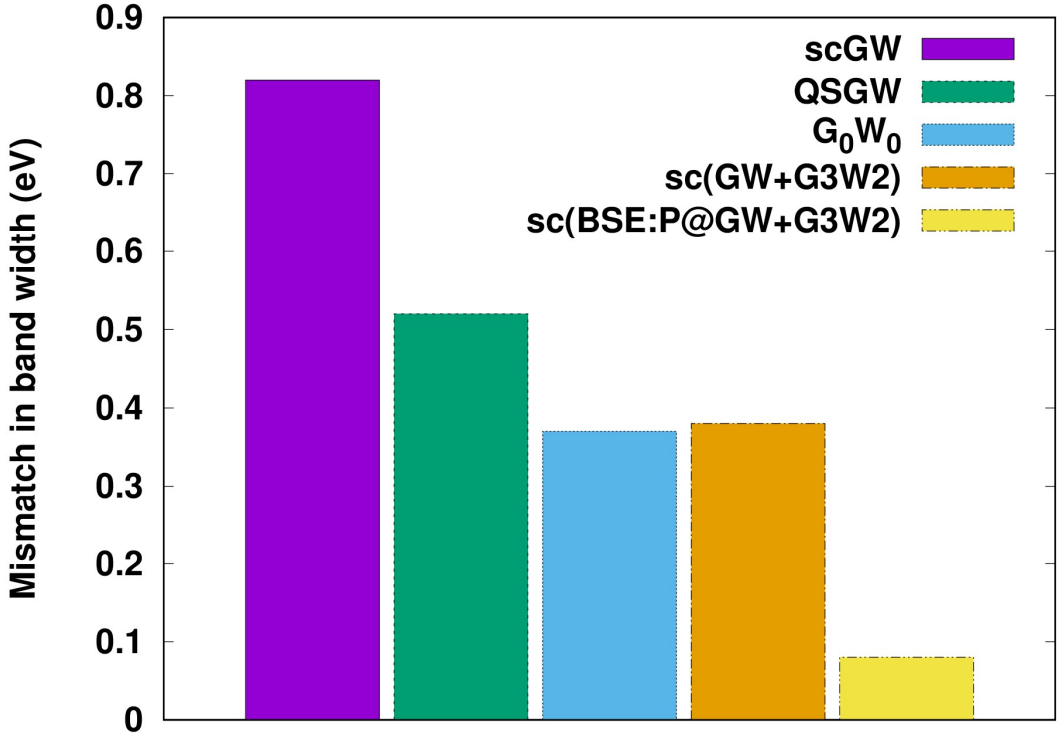
Other examples: NiO



arXiv:2106.03800

Experimental band gap is ~ 4.3 eV,
G. A. Savatzky, and J. W. Allen,
Phys. Rev. Lett. 53 (1984) 2339

Band width of Na

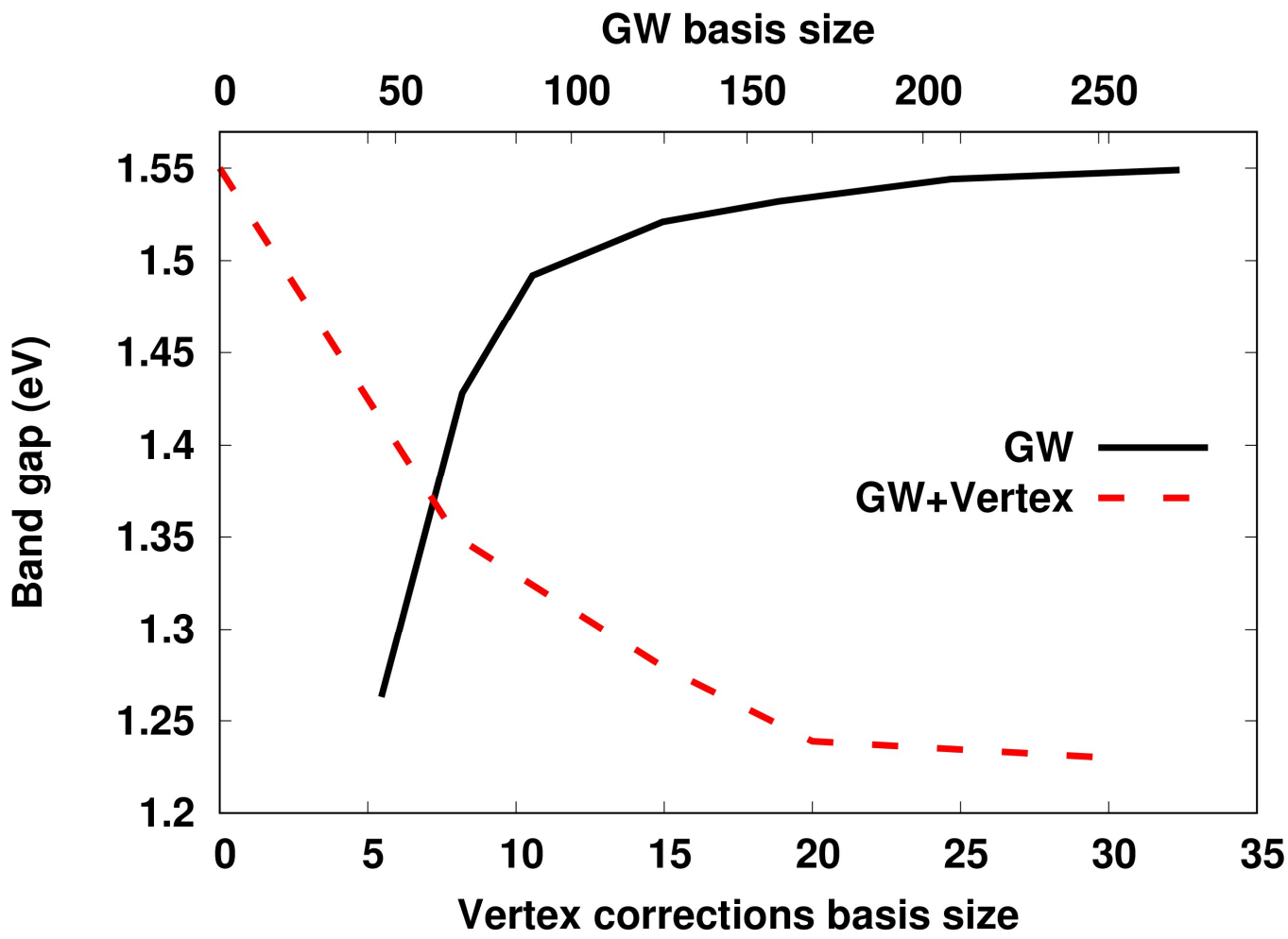


Details:
PRB 94, 155101 (2016)

Relative to the experimental band width 2.65 eV from:
PRL 60, 1558 (1988),
PRB 41, 8075 (1990)

Vertex part convergence issues

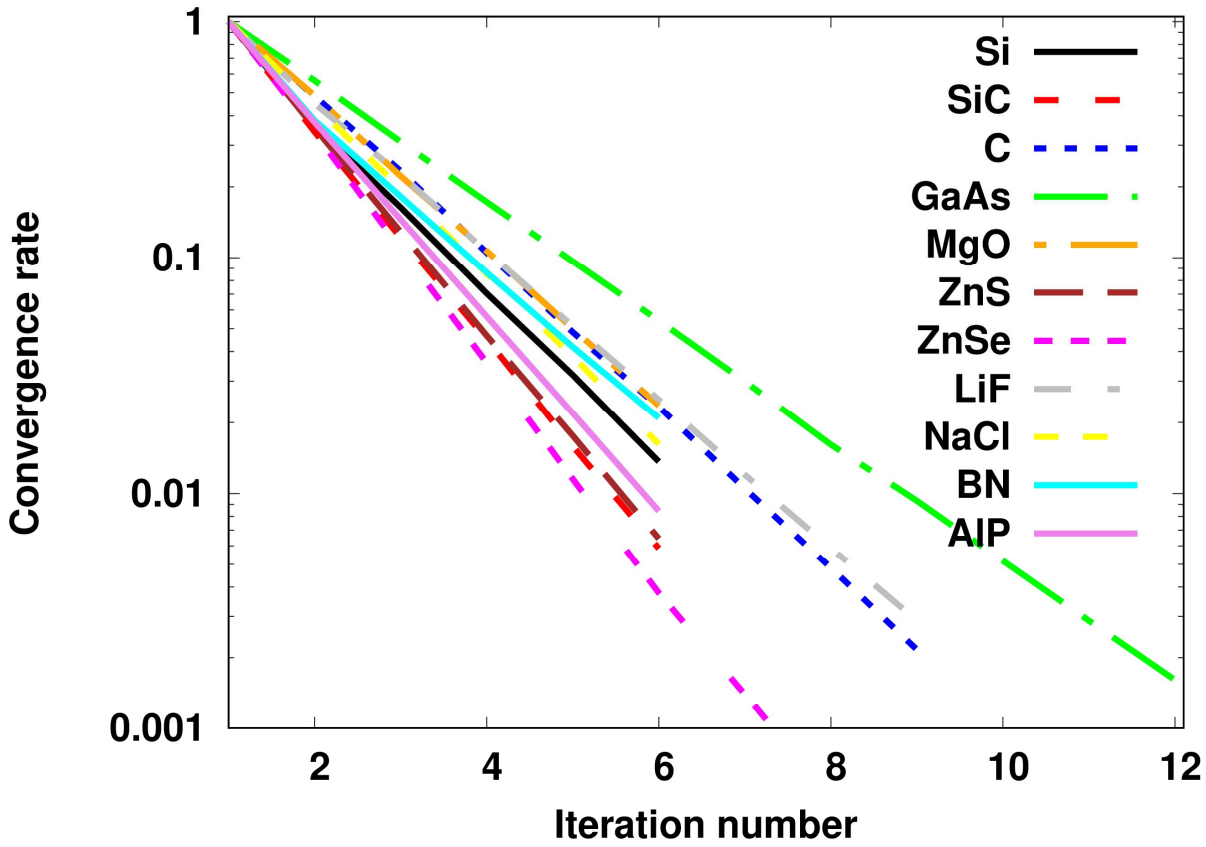
Convergence with respect to the basis size (Si)



Convergence of MgO band gap (GW+Vertex)

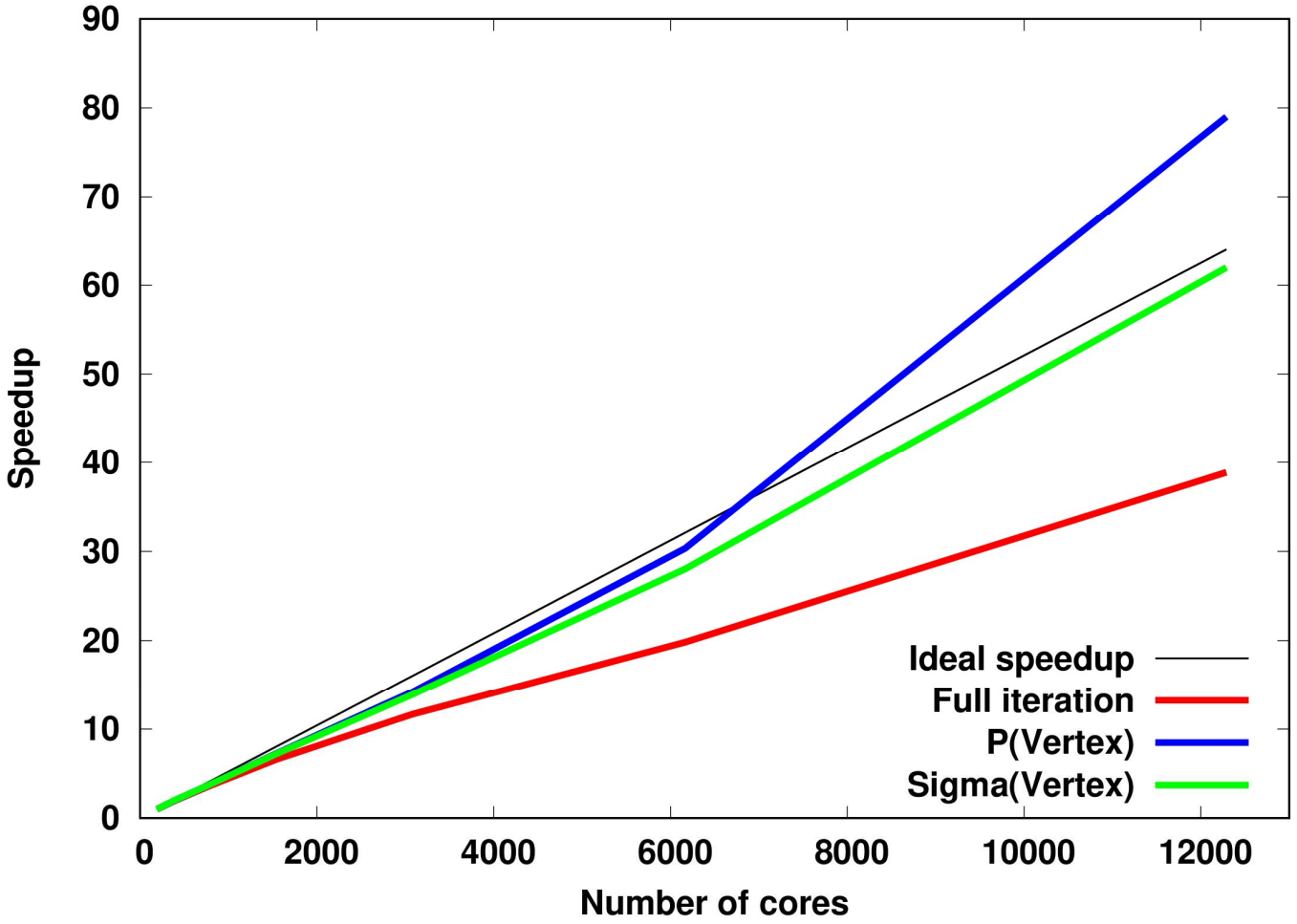
Parameter	Setup	Band gap
N_{bnd}	0	9.31
	5	8.81
	10	8.40
	20	8.28
	30	8.29
ϕ_{nl}	3s(Mg)/2p(O)	8.56
	3s3p(Mg)/3s2p(O)	8.40
	3s3p3d(Mg)/3s2p3d(O)	8.28
	3s3p3d4f(Mg)/3s2p3d4f(O)	8.27
$N_{\mathbf{G}}$	26	8.19
	59	8.25
	92	8.28
$N_{\mathbf{k}}$	2^3	8.28
	3^3	8.24
	4^3	8.27
n_τ, n_ω, n_v	46	8.28
	62	8.29
	94	8.29

Convergence of the ladder sums



$$\Gamma = \mathbf{1} + \boldsymbol{WGG} + (\boldsymbol{WGG})^{2} + (\boldsymbol{WGG})^{3} + \cdots$$

Scalability (PbTe)



Hands on training (Plan)

- Basic DFT run – LiF
- QSGW – LiF
- scGW – LiF
- LDA – Fe (spin-polarized, FM)
- LDA – d-Pu (fully relativistic)
- LDA – Cr (AFM)

How to run the code

1) place the following two files in the same directory:

   ini FlapwMBPT.exe

2) do one of the following three:

   FlapwMBPT.exe (no MPI)

   mpirun ….. (for MPI application)

   use batch.file to submit the job (usually required for clusters)

Input file (ini) – LiF in LDA

```
iter_dft 14  post_dft 0
space_group 225
a_lat 4.026
Li  '0.0,0.0,0.0‘
F   '1/2,1/2,1/2‘
-------- End of mandatory input ----------------
acc_level 0
```

Basic DFT – output files

- **allout** – general output (basis set sizes, timings,…)
- dft.eig – energies for the bands near the Fermi level
- **dft.dos** – DOS (one needs more k-points to make it smooth)
- **dft.dosk** – k-resolved DOS
- **Li1___sum_dft.pdos** – PDOS for Li
- **F_1___sum_dft.pdos** – PDOS for F
- **dft_bands.dat** – data file for band plotting
- **dft_bands.gnu** – GNUPLOT file for band plotting

Input file (ini) – LiF QSGW

```
iter_dft 14  post_dft 1
QP 6
space_group 225
a_lat 4.026
Li  '0.0,0.0,0.0'
F   '1/2,1/2,1/2'
-------- End of mandatory input ----------------
acc_level 0
temperature 900.0
```

QSGW example – output files

- **……..**
- **qp.dos** – DOS for QSGW
- **qp.dosk** – k-resolved DOS for QSGW
- **Li1___sum_qp.pdos** – QSGW PDOS for Li
- **F_1___sum_qp.pdos** – QSGW PDOS for F
- **Nu_Q_Pol.qp** – $P^{\mathbf{q}}_{\mathbf{G}=\mathbf{G}'=0}$ as a function of Matsubara freq-cy
  Im_sigc_w_band_k.qp – Im part of Sigma_corr
- **Re_sigc_w_band_k.qp** – Re part of Sigma_corr
- **qp_qp.eig** – QSGW one electron energies
- **Z_factor_band_k_x.qp** – Renorm Z-factor

Input file (ini) – LiF scGW

```
iter_dft 14  post_dft 1
GW 6
space_group 225
a_lat 4.026
Li  '0.0,0.0,0.0'
F   '1/2,1/2,1/2'
-------- End of mandatory input -----------------
acc_level 0
temperature 900.0
```

scGW example – output files

- **…….**
- **gw.dos** – DOS for scGW
- **gw.dosk** – k-resolved DOS for scGW
- **Li1_Chain_gw.pdos** – scGW PDOS for Li
- **F_1_Chain_gw.pdos** – scGW PDOS for F
- **Nu_Q_Pol.gw** – $P^{\mathbf{q}}_{\mathbf{G}=\mathbf{G}'=0}$ as a function of Matsubara freq-cy
  Im_sigc_w_band_k.gw – Im part of Sigma_corr
- **Re_sigc_w_band_k.gw** – Re part of Sigma_corr
- **gw_qp.eig** – scGW one electron energies (linearization)
- **Z_factor_band_k_x.gw** – Renorm Z-factor

Input file (ini) – Fe LDA (FM)

```
iter_dft 10  post_dft 0
symgen 'I_R4Z_R3D_' bravais_lat cI
a_lat 2.8669
Fe  '0.0,0.0,0.0'
---------- End of mandatory input ----------
acc_level 1
iexch 205
temperature 600.0
admix 0.2
b_ext 0.0 0.0 1.0
iter_h_ext 1000000
magn_shift 1 0.6
```

Input file (ini) – d-Pu LDA (full relat.)

```
iter_dft 20  post_dft 0
symgen 'I_R4Z_R3D_' bravais_lat cF
a_lat 4.6347
Pu  '0.0,0.0,0.0'
------- End of mandatory input ----------
acc_level 1
ndiv 6 6 6
rkmax 9.0
lmb 1 6
admix 0.25
emindos -0.4
emaxdos  0.4
e_small 0.005
core_lim -6.0
core_bcs 0.03
```

Input file (ini) – Cr LDA (AFM)

```
iter_dft 20  post_dft 0
symgen 'I_R4Z_-E:T(1/2,1/2,1/2)_' bravais_lat cP
a_lat 2.91
Cr '0.0,0.0,0.0'
-------- End of mandatory input ----------------
acc_level 1
b_ext 0.0 0.0 1.0
magn_shift 1 0.5
iter_h_ext 1000000
rkmax 9.0
lmb 1 6
iexch 205
```

Homework (optional)

- Download file
- Compile the source files to get FlapwMBPT.exe
- Run examples

How to compile FlapwMBPT?

- Edit make.sys
- In ‘src’ directory:

  - make sure that reference to make.sys in makefile corresponds to the actual place where make.sys is. By default the reference is: ‘../make.sys’

  - execute the command ‘make’

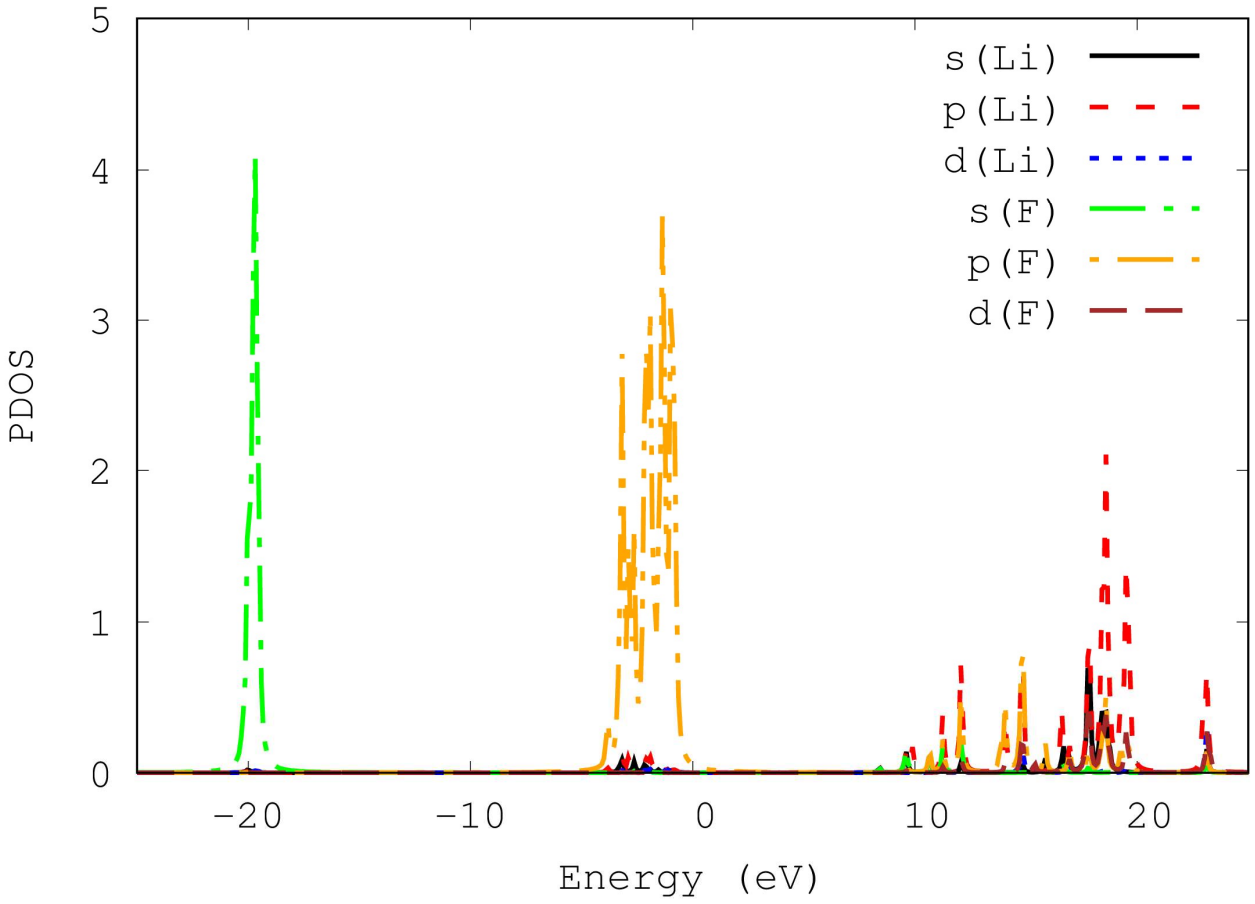
What to edit in make.sys?

- F90 = mpiifort
- PREPROC = -cpp -DMPI -DFFTMKL -DINTEL
- LDFLAG = -O3
- LIB = -mkl=sequential
- PROG = FlapwMBPT.exe

LiF ‘ini’ file for sc(GW+G3W2)

- iter_dft 14 post_dft 2
- GW 8 PSI 6
- space_group 225
- a_lat 4.026
- Li '0.0,0.0,0.0'
- F   '1/2,1/2,1/2'
- -------- End of mandatory input ----------
- ndiv 4 4 4
- n_tau 62
- psi_p 60000
- psi_sig 60000
- correlated 2 0 1 C
- correlated 1 1 1 C
- correlated 2 0 2 C
- correlated 1 1 2 C
- ndiv_c 2 2 2
- c0_bnd 2
- nproc_t 32
- nproc_k 3
- nproc_b 1

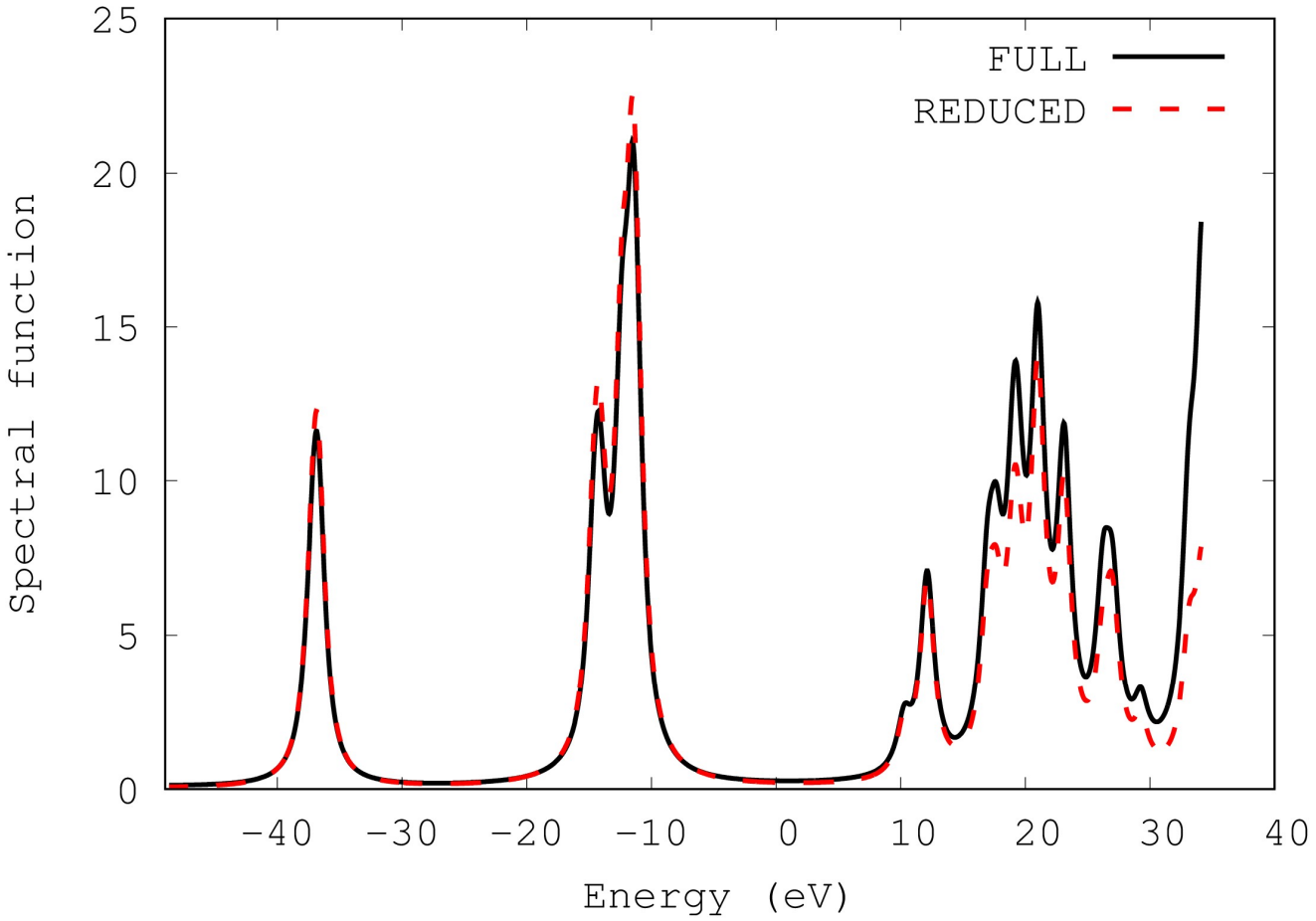
Assessing the quality of the reduced basis set



Lif_F___sum_dft.pdos
lifLi___sum_dft.pdos

-20.771	0.043	0.000	0.000	0.000
-20.680	0.053	0.000	0.000	0.000
-20.589	0.068	0.000	0.000	0.000
-20.499	0.091	0.000	0.000	0.000
-20.408	0.132	0.000	0.000	0.000
-20.317	0.231	0.000	0.000	0.000
-20.227	0.389	0.000	0.000	0.000
-20.136	0.620	0.001	0.000	0.000
-20.045	1.550	0.001	0.000	0.000
-19.955	1.690	0.001	0.000	0.000
-19.864	1.865	0.001	0.000	0.000
-19.773	3.293	0.001	0.000	0.000
-19.682	4.099	0.001	0.000	0.000
-19.592	2.622	0.001	0.000	0.000
-19.501	1.583	0.001	0.000	0.000
-19.410	0.594	0.000	0.000	0.000
-19.320	0.299	0.000	0.000	0.000
-19.229	0.184	0.000	0.000	0.000
-19.138	0.125	0.000	0.000	0.000
-19.047	0.092	0.000	0.000	0.000
-18.957	0.070	0.000	0.000	0.000
-18.866	0.055	0.000	0.000	0.000
-18.775	0.045	0.000	0.000	0.000
-18.685	0.037	0.000	0.000	0.000
-18.594	0.031	0.000	0.000	0.000

Assessing the quality of the reduced basis set



lifX_FULL_RED.dos

-16.73566	1.49960	1.58143
-16.64486	1.58879	1.67938
-16.55406	1.68721	1.78747
-16.46326	1.79620	1.90719
-16.37246	1.91734	2.04029
-16.28166	2.05252	2.18885
-16.19086	2.20400	2.35536
-16.10006	2.37448	2.54278
-16.00926	2.56718	2.75468
-15.91846	2.78598	2.99534
-15.82766	3.03551	3.26987
-15.73686	3.32131	3.58436
-15.64606	3.64988	3.94600
-15.55526	4.02874	4.36307
-15.46446	4.46631	4.84486
-15.37366	4.97147	5.40113
-15.28286	5.55254	6.04101
-15.19206	6.21542	6.77093
-15.10126	6.96050	7.59116
-15.01046	7.77840	8.49109
-14.91966	8.64545	9.44427
-14.82886	9.52160	10.40605
-14.73806	10.35398	11.31764
-14.64726	11.08721	12.11773
-14.55646	11.67552	12.75601
-14.46566	12.08762	13.19876
-14.37486	12.30271	13.42425
-14.28406	12.30718	13.41886
-14.19326	12.10181	13.18445
-14.10246	11.71431	12.75190
-14.01166	11.20228	12.18434

sc(GW+G3W2) example – output files

- **…….**
- **psi.dos** – Spectral function (SF) for sc(GW+G3W2)
- **psi.dosk** – k-resolved SF for sc(GW+G3W2)
- **Li1_Chain_psi.pdos** – sc(GW+G3W2) PSF for Li
- **F_1_Chain_psi.pdos** – sc(GW+G3W2) PSF for F
- **Nu_Q_Pol.psi** – P as a function of Matsubara freq-cy
- **Im_sigc_w_band_k.psi** – Im part of Sigma_corr
- **Re_sigc_w_band_k.psi** – Re part of Sigma_corr
  Z_factor_band_k_x.psi – Renorm Z-factor

For Bethe-Salpeter based calculations “psi” is replaced with “bsp”