GWL tutorial

Paolo Umari, Università degli Studi di Padova, Italy
Democritos, Trieste

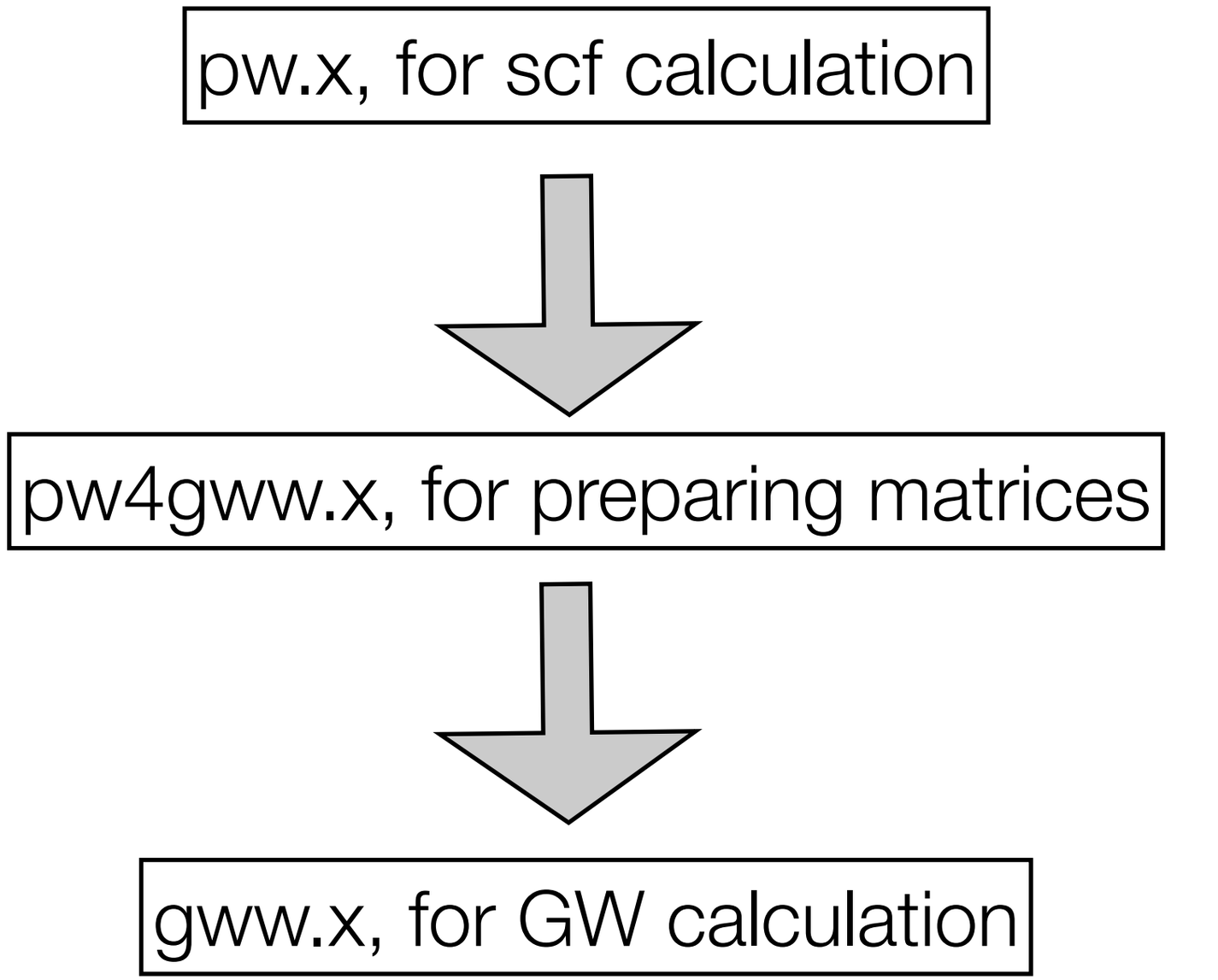
GW calculation with QE and GWL

Benfits:

- Optimal basis (reduced) for representing polarizabilty operators
- Full convergence on sums over empty states through a Lanczos chain algorithm
- Use analytic continuation no plasmon-pole

A minimal example: CH_4

3 steps:



CH_4: pw.x

methane_scf.in

Only : Gamma point calculations

```
&control
   calculation = 'scf',
   restart_mode='from_scratch',
   prefix='ch4',
   tprnfor = .true.,
   pseudo_dir = './',
 /
 &system
   ibrav=  1,
   celldm(1) =15.0,
   nat=5,
   ntyp= 2,
   ecutwfc =40.0,
   nbnd=5
 /
 &electrons
   diagonalization='cg'
   mixing_beta = 0.5,
   conv_thr =  1.0d-8
 /
ATOMIC_SPECIES
 H  1.0   H.pz-vbc.UPF
 C  12.0  C.pz-vbc.UPF
ATOMIC_POSITIONS {bohr}
H       1.198204546   1.198204546   1.198204546
H      -1.198204546  -1.198204546   1.198204546
H       1.198204546  -1.198204546  -1.198204546
H      -1.198204546   1.198204546  -1.198204546
C       0.000000000   0.000000000   0.000000000
```

CH_4: pw4gww.x

methane_pw4gww.in

valence states

total # of KS states

```
&inputpw4gww
   prefix='ch4'
   num_nbndv(1)=4
   num_nbnds=5
   l_truncated_coulomb=.true.
   truncation_radius=7.5d0
   numw_prod=50
/
```

Use truncated Coulomb interaction for isolated systems and define radius (bohr)

Dimension of optimal polarizability basis

CH_4:gww.x

$$\tau = \frac{2}{\Omega} n$$

methane_gww.in

```
&inputgww
ggwin%prefix='ch4'
ggwin%max_i=5,
ggwin%i_min=1
ggwin%i_max=5
ggwin%omega=20
ggwin%n=118,
ggwin%tau=11.8
ggwin%grid_freq=5
ggwin%second_grid_i=3
ggwin%second_grid_n=10
ggwin%omega_fit=20
ggwin%n_grid_fit=240
ggwin%n_fit=120,
ggwin%n_multipoles=2
ggwin%l_truncated_coulomb=.true.
/
```

tot # of states

from state

to state

Length of frequency grid

of *main* grid points

Length of time grid

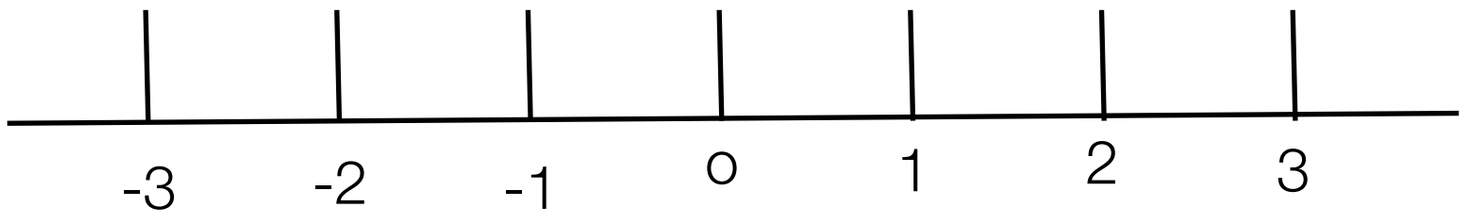
Length of frequency for fitting e.v. of self-energy, and # of points

actually fitted

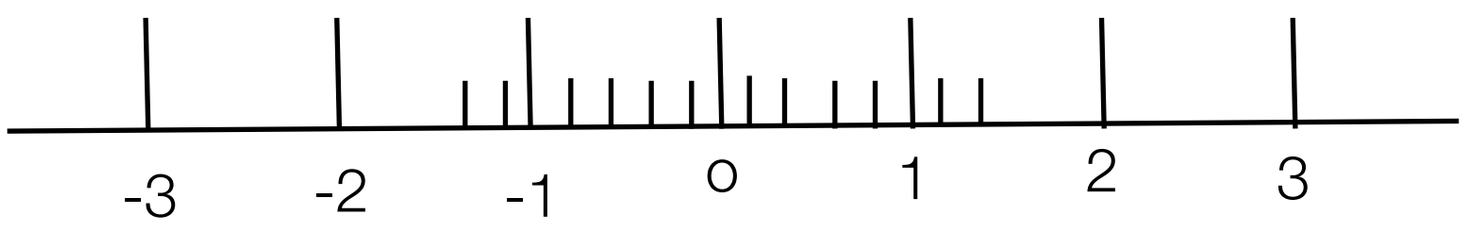
# of poles in multipole expansion

gww.x: Frequency grids

Equally spaces grid: ⟶ ggwin%grid_freq=3



Augmented at origin: ⟶ ggwin%grid_freq=5



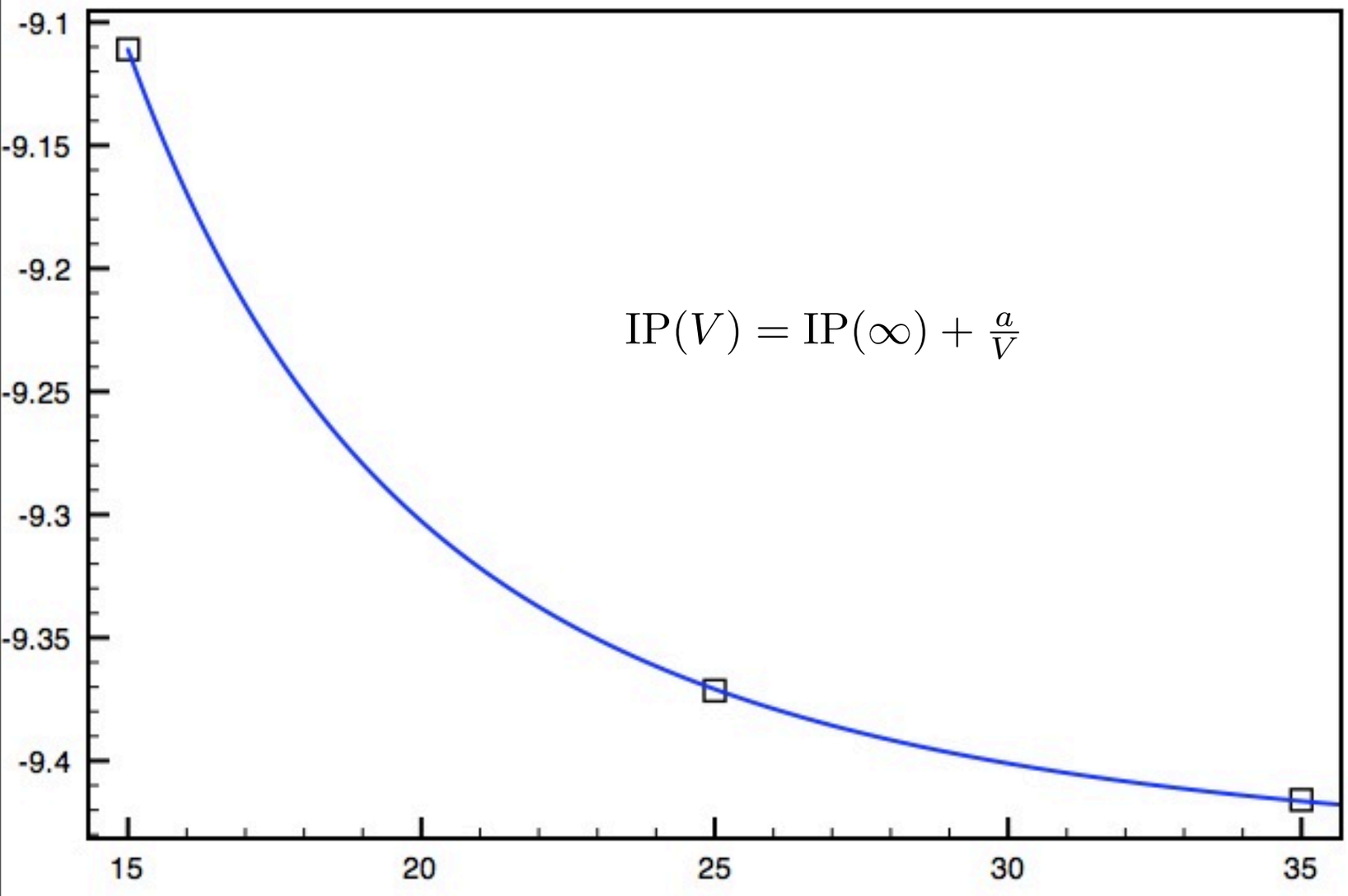
Main points to be augmented: ⟶ ggwin%second_grid_i=1

augmented points in half-interval ⟶ ggwin%second_grid_n=2

gww.x: results

QUASI-PARTICLES ENERGIES IN Ev, Spin: 1 1
State: 1DFT : -16.61246 GW-PERT : -20.35596 GW : -20.12100 HF-pert : -25.31687
State: 2DFT : -9.11161 GW-PERT : -13.67554 GW : -13.58892 HF-pert : -14.42308
State: 3DFT : -9.11161 GW-PERT : -13.67151 GW : -13.58390 HF-pert : -14.42308
State: 4DFT : -9.11161 GW-PERT : -13.67654 GW : -13.58950 HF-pert : -14.42308
State: 5DFT : -0.56517 GW-PERT : 0.21436 GW : 0.21384 HF-pert : 0.90049
IMAGINARY ENERGIES IN Ev:
State: 1 GW (Im) : 0.51158
State: 2 GW (Im) : -0.04833
State: 3 GW (Im) : -0.05202
State: 4 GW (Im) : -0.04985
State: 5 GW (Im) : 0.02240

Energy position of vacuum level

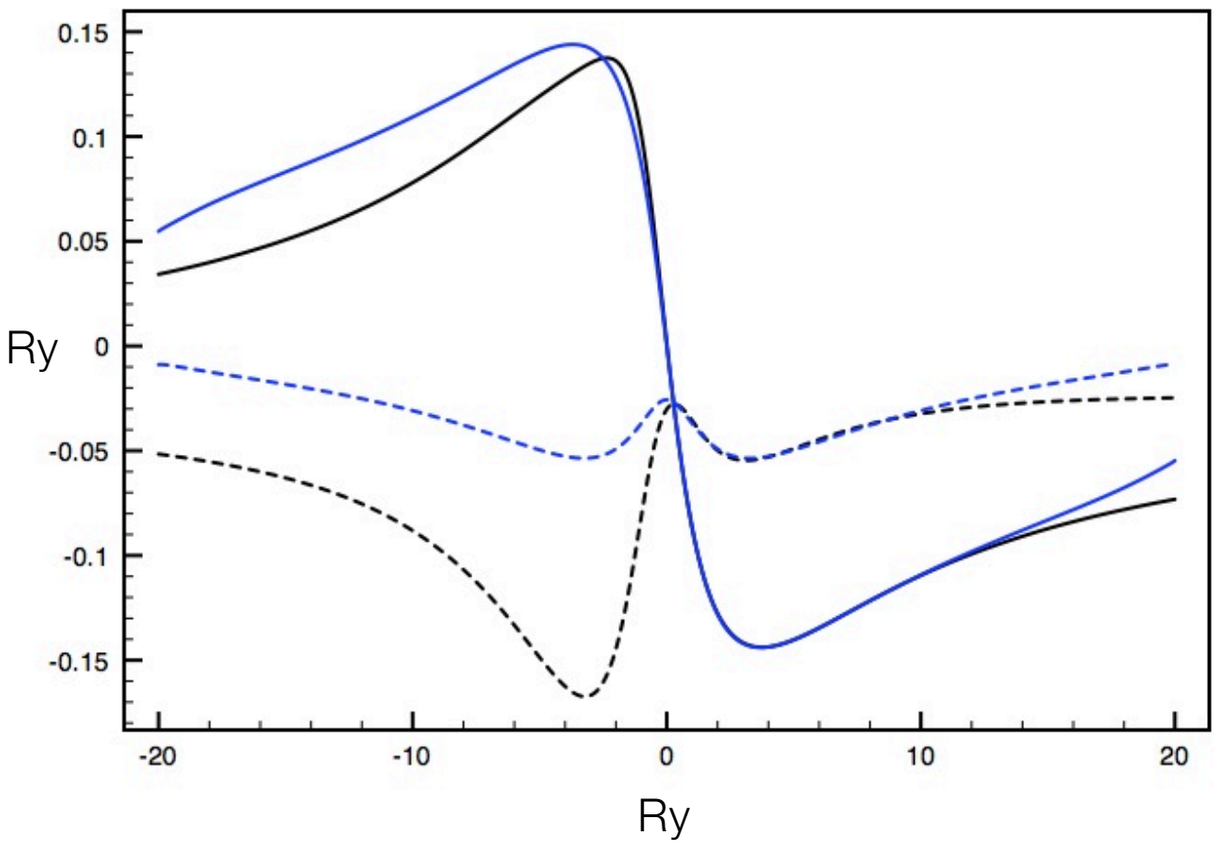


- IP_{LDA}=9.44eV
- IP_{GW}=13.91eV
- IP_{ext}=13.6eV

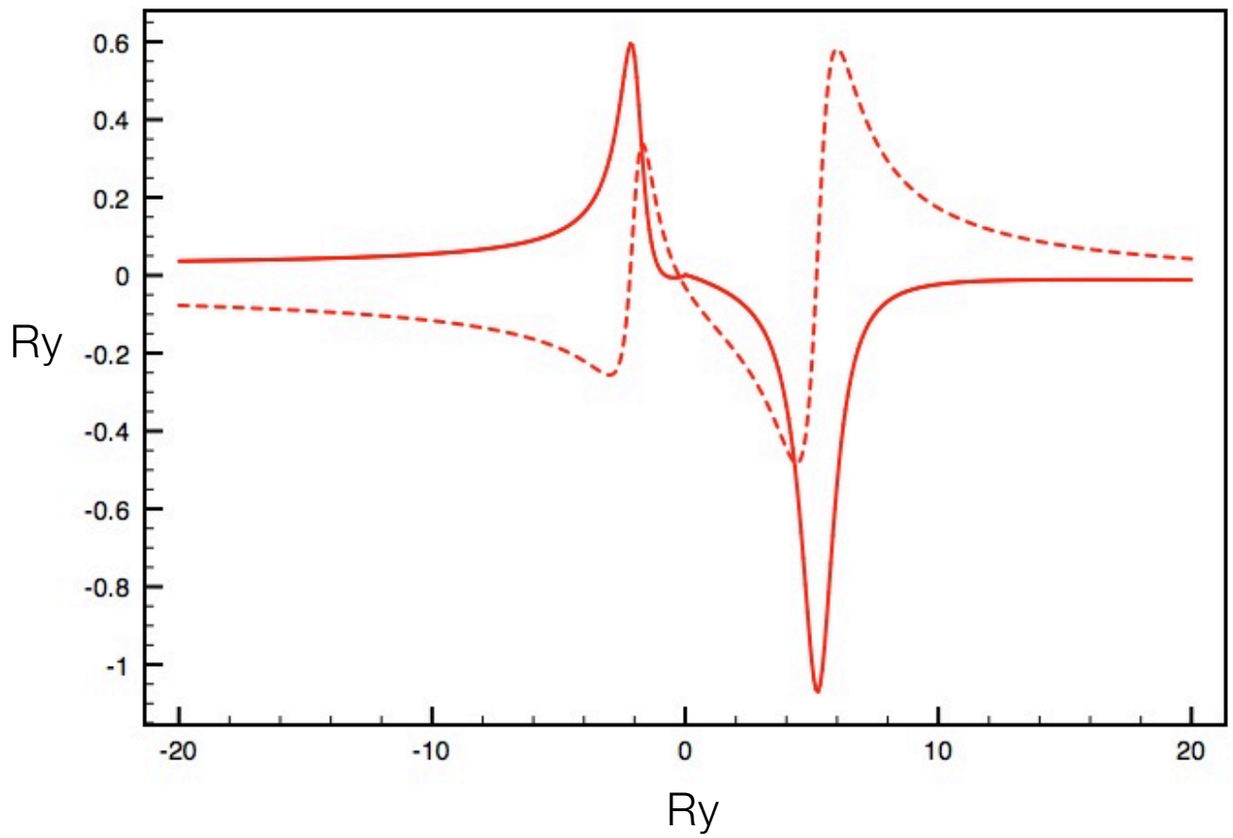
Self-energy files:

Expectation values of the self-energy operator: real (dashed) *re_on_im000XX* and imaginary *im_on_im000XX* parts:

On imaginary energy:



On real energy:



- calculated
- fitted
- analytically continued

The rules of the game: G_0W_0 Approximation

M.S. Hybertsen and S.G. Louie, Phys. Rev. Lett 55, 1418 (1985)

$$E_n \simeq \epsilon_n + \langle \Sigma_{G^\circ W^\circ}(E_n) \rangle_n - \langle V_{xc} \rangle_n$$

$$\Sigma_{G^\circ W^\circ}(\mathbf{r}, \mathbf{r}'; \omega) = \frac{i}{2\pi} \int d\omega' G^\circ(\mathbf{r}, \mathbf{r}'; \omega - \omega') W^\circ(\mathbf{r}, \mathbf{r}'; \omega')$$

$$\mathrm{W}^\circ \quad = \quad v + v \cdot \Pi^\circ \cdot v \quad \text{where} \quad \Pi^\circ = P^\circ \cdot (1 - v \cdot P^\circ)^{-1}$$

$$P^\circ(\mathbf{r}, \mathbf{r}'; \omega) = \frac{1}{2\pi} \int d\omega' G^\circ(\mathbf{r}, \mathbf{r}'; \omega - \omega') G^\circ(\mathbf{r}, \mathbf{r}'; \omega')$$

$$G^\circ(\mathbf{r}, \mathbf{r}'; \omega) = \sum_i \frac{\psi_i(\mathbf{r}) \psi_i^*(\mathbf{r}')}{\omega - \epsilon_i \pm i\delta}$$

For accurate calculations: analytic continuation method

M.M. Rieger, L. Steinbeck, I.D. White, H.N. Rojas and R.W. Godby, Comp. Phys. Comm. 117 211 (1999)

Optimal polarizability basis

If an optimal representation of P° can be found:

$$P^\circ(\mathbf{r},\mathbf{r}';\omega) \simeq \sum_{\alpha\beta} \Phi_\alpha(\mathbf{r}) {P^\circ}_{\alpha\beta}(\omega) \Phi_\beta(\mathbf{r}')$$

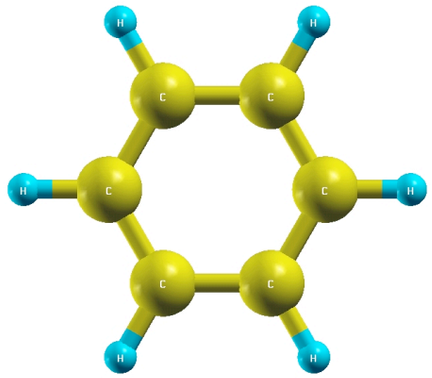
$$\Pi^\circ(\mathbf{r},\mathbf{r}';\omega) \simeq \sum_{\alpha\beta} \Phi_\alpha(\mathbf{r}) {\Pi^\circ}_{\alpha\beta}(\omega) \Phi_\beta(\mathbf{r}')$$

$$W^\circ(\mathbf{r},\mathbf{r}';\omega) \simeq \int d\mathbf{r}'' d\mathbf{r}''' \sum_{\alpha\beta} v(\mathbf{r},\mathbf{r}'') \Phi_\alpha(\mathbf{r}'') {\Pi^\circ}_{\alpha\beta}(\omega) \Phi_\beta(\mathbf{r}''') v(\mathbf{r}''',\mathbf{r}')$$

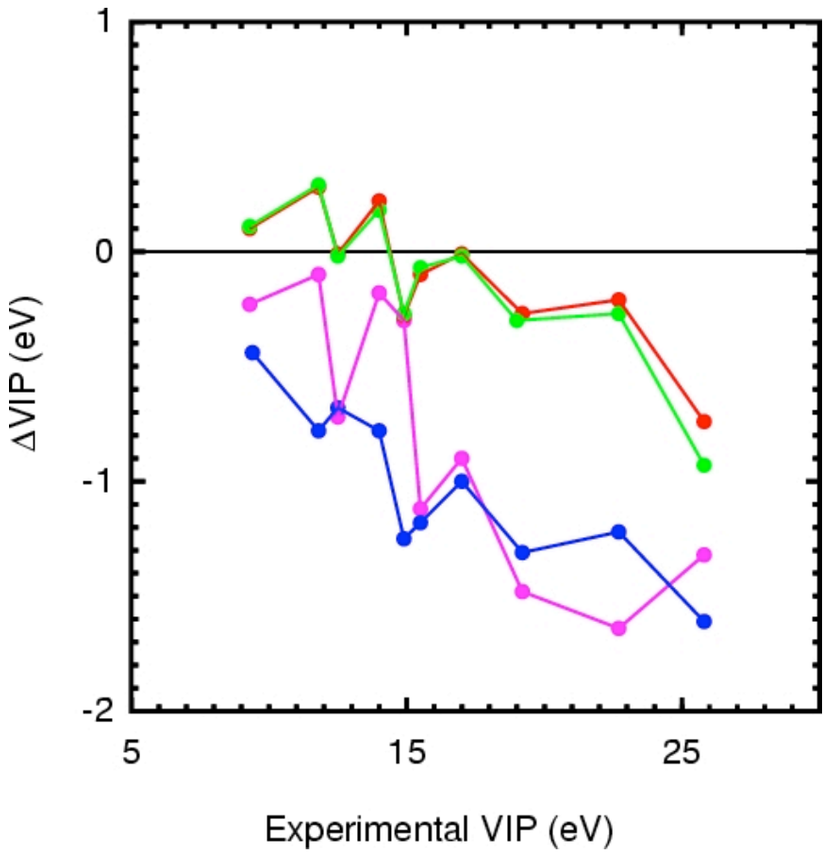
then a high speed-up can be achieved

We take the most important eigenvectors of P(E*, t = 0)

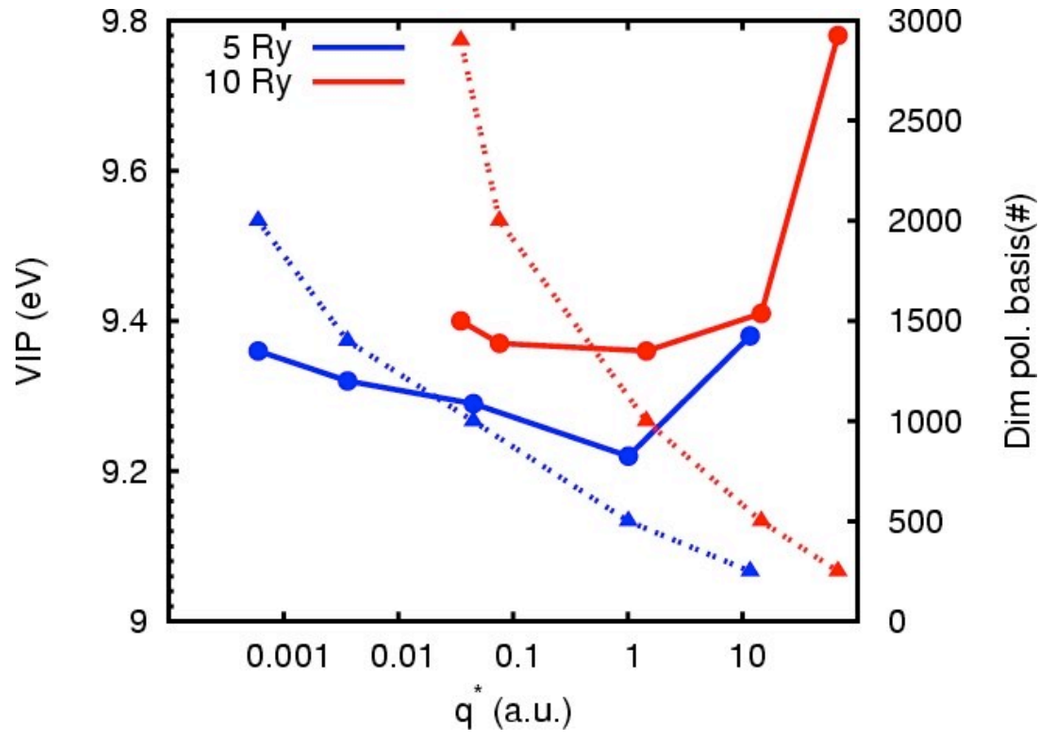
Optimal polarizability basis: Benzene



convergence of IPs



convergence of first IP



- E=10Ry q_* =0.035a.u. N = 2900
- E=10Ry q_* =14.5a.u. N = 500
- Extrapolations
- Plane waves E = 5 Ry N = 1500

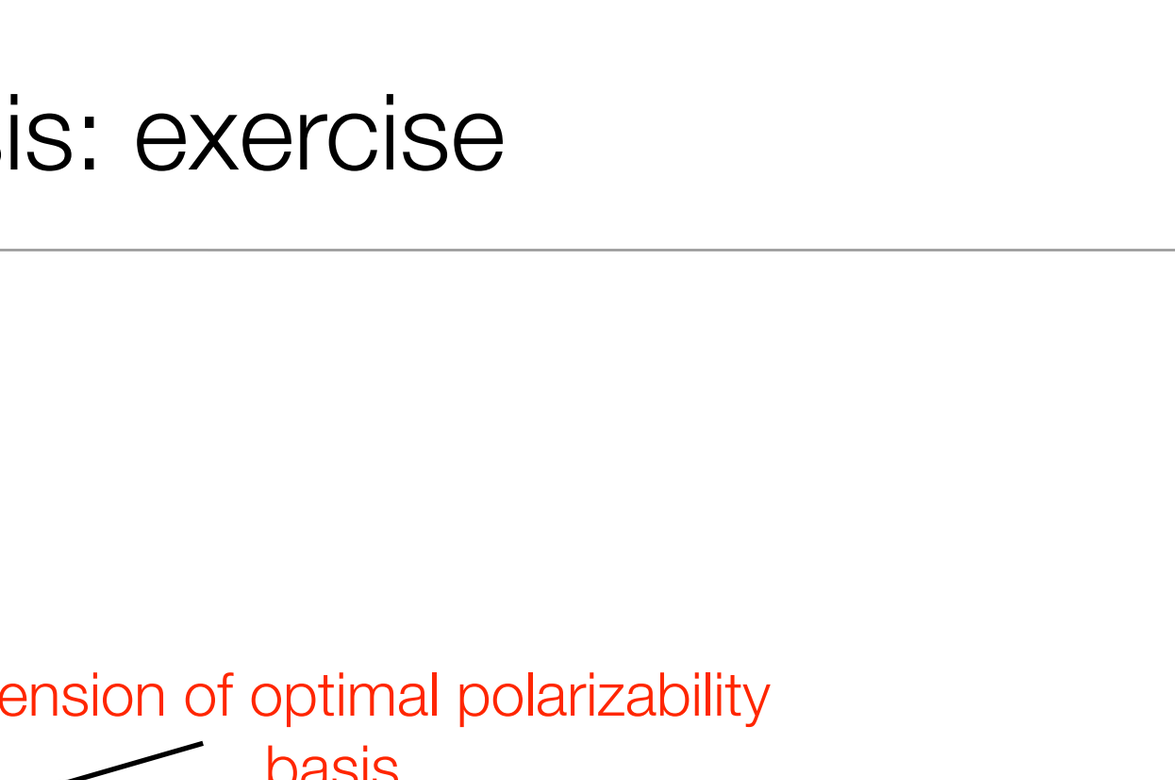
Optimal polarizability basis: exercise

methane_pw4gww_basis.in

```
&inputpw4gww
   prefix='ch4'
   num_nbndv(1)=4
   num_nbnds=5
   l_truncated_coulomb=.true.
   truncation_radius=7.5d0
   numw_prod=100
   pmat_cutoff=3d0
/
```

Dimension of optimal polarizability basis

E^*



Lanczos chains

Sternheimer approach for P

The polarizability matrix $P^{\circ}_{\mu\nu}(i\omega)$:

$$P^{\circ}_{\mu\nu}(i\omega) = -4\Re \sum_{v,c} \frac{\int d\mathbf{r} d\mathbf{r}' \Phi_\mu(\mathbf{r})\psi_v(\mathbf{r})\psi_c(\mathbf{r})\psi_v(\mathbf{r}')\psi_c(\mathbf{r}')\Phi_\nu(\mathbf{r}')}{\epsilon_c - \epsilon_v + i\omega}.$$

the projector over the conduction manifold Q_c:

$$Q_c(\mathbf{r},\mathbf{r}') = \sum_c \psi_c(\mathbf{r})\psi_c(\mathbf{r}') = \delta(\mathbf{r}-\mathbf{r}') - \sum_v \psi_v(\mathbf{r})\psi_v(\mathbf{r}'),$$

with the notation:

$$\langle \mathbf{r}|\psi_i\Phi_\nu\rangle = \psi_i(\mathbf{r})\Phi_\nu(\mathbf{r}).$$

We can now eliminate the sum over c :

$$P^{\circ}_{\mu\nu}(i\omega) = -4\Re \sum_v \langle \Phi_\mu\psi_v|Q_c(H-\epsilon_v+i\omega)^{-1}Q_c|\psi_v\Phi_\nu\rangle,$$

See also: F. Giustino,M.L.Cohen, and S.G Louie PRB **81**,115105 (2019).

Self-energy

$$\langle\psi_i|\Sigma_c(i\omega)|\psi_i\rangle = \frac{1}{2\pi}\int d\omega' \sum_{\mu,\nu}\langle\psi_i(v\Phi_\mu)|(H^0 - i(\omega-\omega'))^{-1}|(v\Phi_\nu)\psi_i\rangle\Pi_{\mu\nu}(i\omega')$$

Equivalent Lanczos approach for the self-energy:

$$\langle\mathbf{r}|\psi_n(v\Phi_\mu)\rangle \approx \sum_\alpha s^0_\alpha(\mathbf{r})S_{\alpha,n\mu},$$

with:

$$\langle\mathbf{r}|(v\Phi_\mu)\rangle = \int d\mathbf{r}' v(\mathbf{r},\mathbf{r}')\Phi_\mu(\mathbf{r}')$$

For constructing *s* vectors: block algorithm starting from KS states

Lanczos chains

The computational load can be hugely reduced:

with an optimal basis:

$$\langle r|Q_c|\psi_v\Phi_\mu\rangle \approx \sum_\alpha t_\alpha(\mathbf{r})T_{\alpha,v\mu},$$

We can easily solve:

$$\langle t_\alpha|(H-\epsilon_v+i\omega)^{-1}|t_\beta\rangle$$

for every ϵ_v and every ω using Lanczos chains.

Implemented in the *quantum-Espresso* package
www.quantum-espresso.org
See: P.Umari, G Stenuit and S. Baroni PRB **79** 201014R (2009)
See: P.Umari, G Stenuit and S. Baroni PRB **81** 115104 (2010)

t basis construction: recipe

1. For each Wannier valence function: construct local orthonormal basis
2. With a block algorithm construct global orthonormal basis
3. Construct Lanczos chains and store overlap terms with global basis

Lanczos steps for self-energy: exercise

methane_pw4gww_steps.in

```
&inputpw4gww
   prefix='ch4'
   num_nbndv(1)=4
   num_nbnds=5
   l_truncated_coulomb=.true.
   truncation_radius=7.5d0
   numw_prod=100
   nsteps_lanczos_self=2
   restart_gww=2
   lanczos_restart=3
   s_first_state=4
   s_last_state=4
/
```

we can restart from a previous calculation

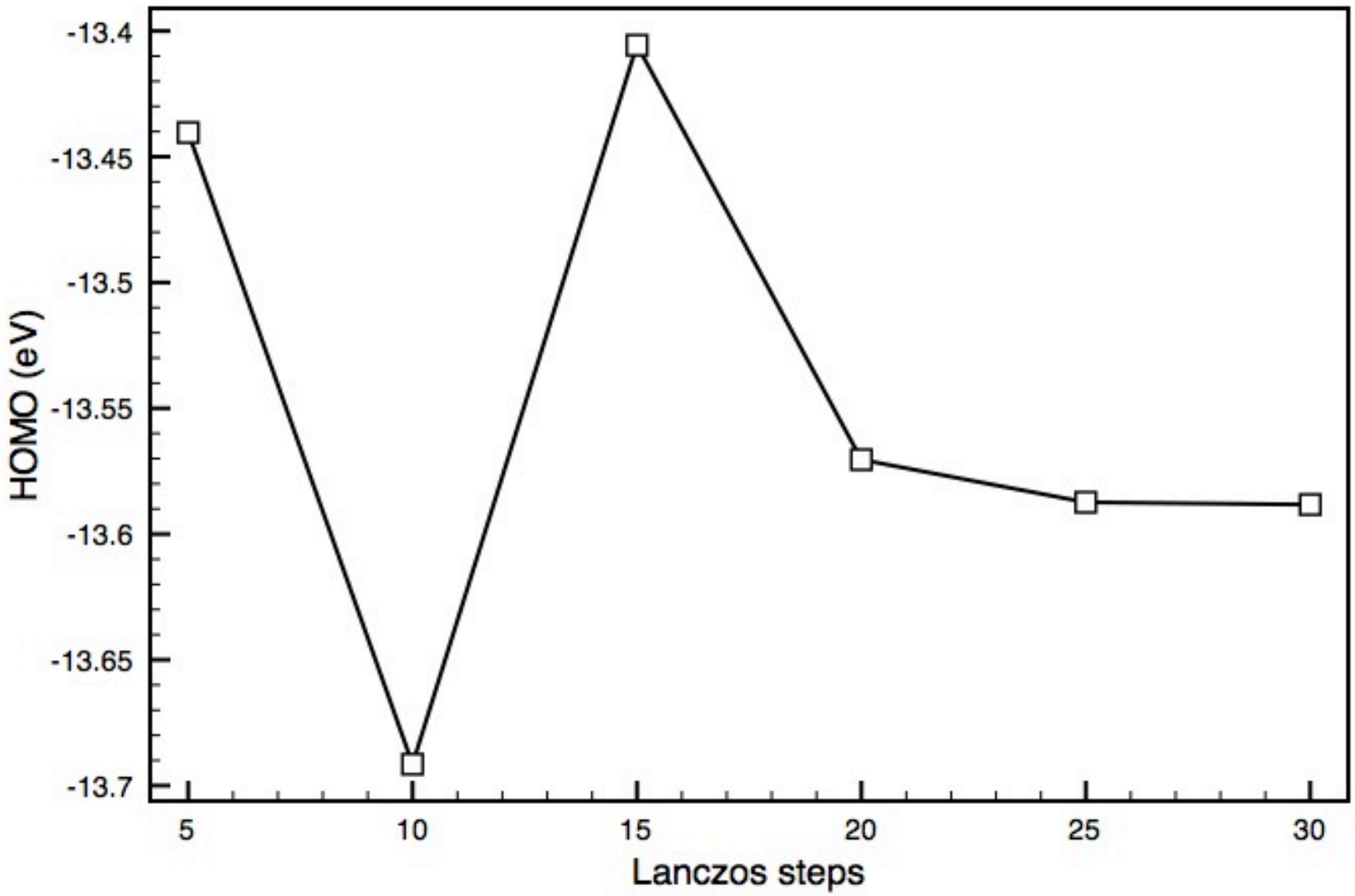
number of Lanczos steps for self-energy

start from calculated *local s,* vectors

KS states considered in *global s* basis

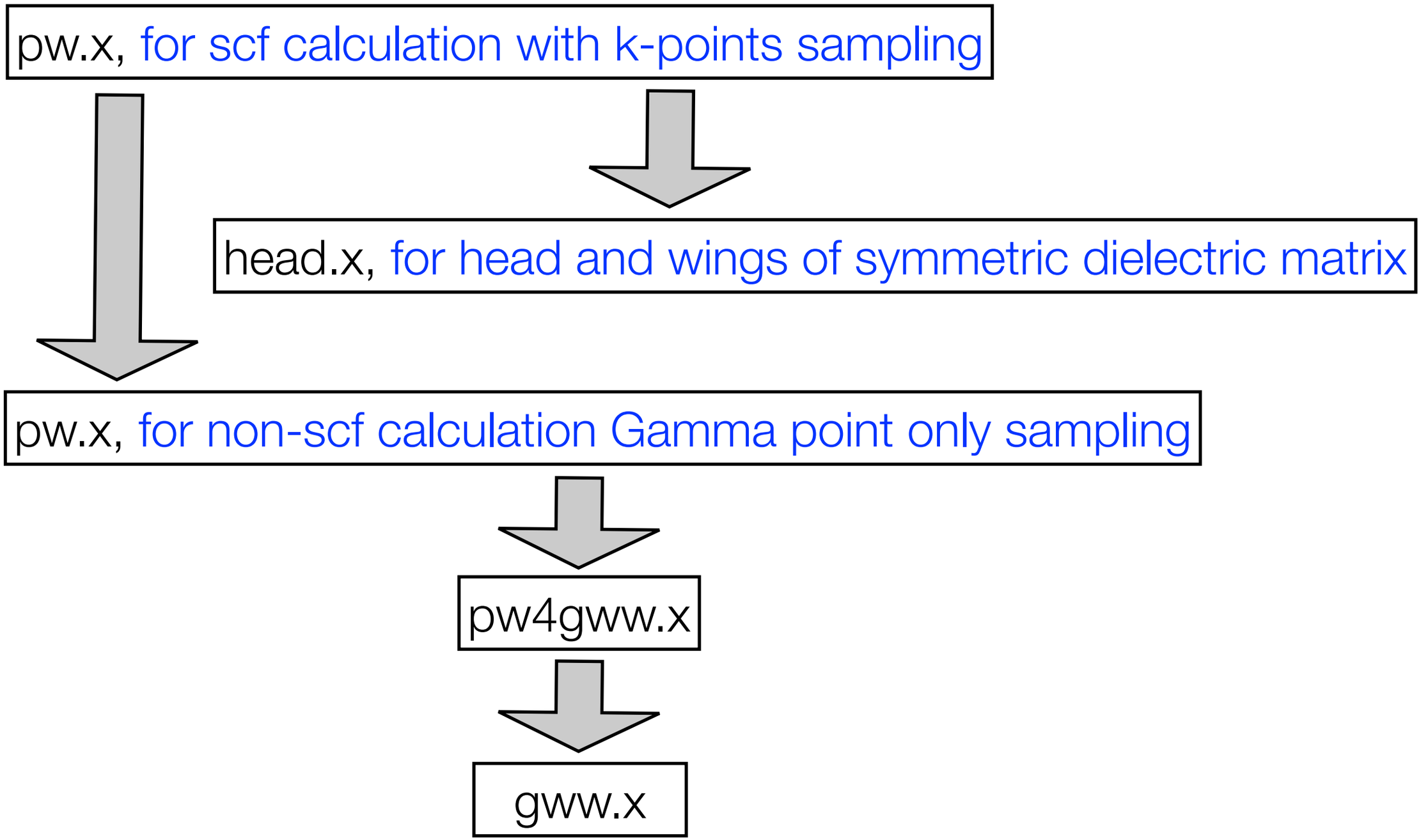
in methane_gww.in, we set: ggwin%starting_point=6

Lanczos steps...



extended systems:

4 steps:



Example: bulk silicon

Note: a 8 Si atoms cell, is not enough for giving converged results, use at least 64 atoms cells for production calculations!

si_scf_k.in

```
&control
 calculation='scf'
 restart_mode='from_scratch',
 prefix='si'
 pseudo_dir='./'
/
&system
 ibrav= 8, celldm(1)= 10.26,celldm(2)= 1, celldm(3)=1,
 nat=  8, ntyp= 1,
 ecutwfc = 15.0
/
&electrons
 diagonalization='david',
 conv_thr =  1.0d-10,
 mixing_beta = 0.5,
 startingwfc='random',
/
ATOMIC_SPECIES
Si  1. Si.pz-vbc.UPF
ATOMIC_POSITIONS (crystal)
Si 0.00000 0.00000 0.00000
Si 0.50000 0.50000 0.00000
Si 0.00000 0.50000 0.50000
Si 0.50000 0.00000 0.50000
Si 0.25000 0.25000 0.25000
Si 0.75000 0.75000 0.25000
Si 0.75000 0.25000 0.75000
Si 0.25000 0.75000 0.75000
K_POINTS (automatic)
4 4 4 1 1 1
```

si_head.in

```
&inputph
 trans=.false.
 l_head=.true.
 tr2_ph=1.d-4,
 prefix='si',
 omega_gauss=20.0
 n_gauss=97
 grid_type=5
 second_grid_i=1
 second_grid_n=10
 niter_ph=1
 nsteps_lanczos=30
 outdir='./'
/
0.0 0.0 0.0
```

frequency grid parameters must be the same as in gww.x

# of Lanczos steps

Example: bulk Si

si_nscf.in

```
&control
 calculation='nscf'
 restart_mode='from_scratch',
 prefix='si'
 pseudo_dir='./'
/
&system
 ibrav= 8, celldm(1)= 10.26,
celldm(2)= 1, celldm(3)=1,
 nat=  8, ntyp= 1,
 ecutwfc = 15.0, nosym=.true.
    nbnd=32
/
&electrons
 diagonalization='cg',
 conv_thr =  1.0d-10,
 mixing_beta = 0.5,
 startingwfc='random',
/
ATOMIC_SPECIES
Si  1. Si.pz-vbc.UPF
ATOMIC_POSITIONS (crystal)
Si 0.00000 0.00000 0.00000
Si 0.50000 0.50000 0.00000
Si 0.00000 0.50000 0.50000
Si 0.50000 0.00000 0.50000
Si 0.25000 0.25000 0.25000
Si 0.75000 0.75000 0.25000
Si 0.75000 0.25000 0.75000
Si 0.25000 0.75000 0.75000
```

si_pw4gww.in

```
&inputpw4gww
    prefix='si'
    num_nbndv(1)=16
    num_nbnds=32
    l_truncated_coulomb=.false.
    numw_prod=100
    pmat_cutoff=3d0
    s_self_lanczos=1d-8

/
```

extended system

accuracy of *global s* basis, default 1d-12

Bulk Si

si_gww.in

```
&inputgww
ggwin%prefix='si'
ggwin%n=97,
ggwin%n_fit=120,
ggwin%max_i=32,
ggwin%i_min=1
ggwin%i_max=32
ggwin%l_truncated_coulomb=.false.
ggwin%grid_time=3
ggwin%grid_freq=5
ggwin%second_grid_i=1
ggwin%second_grid_n=10
ggwin%omega=20
ggwin%omega_fit=20
ggwin%n_grid_fit=240
ggwin%tau=9.8
ggwin%n_set_pola=16
/
```

extended system

how many valence states to save in memory at the same time while calculating the irreducible polarizability, it affects only the performance and memory requirements, default: 4

Comparison with plane waves basis sets

We use a plane-waves basis set for representing polarizability operators:

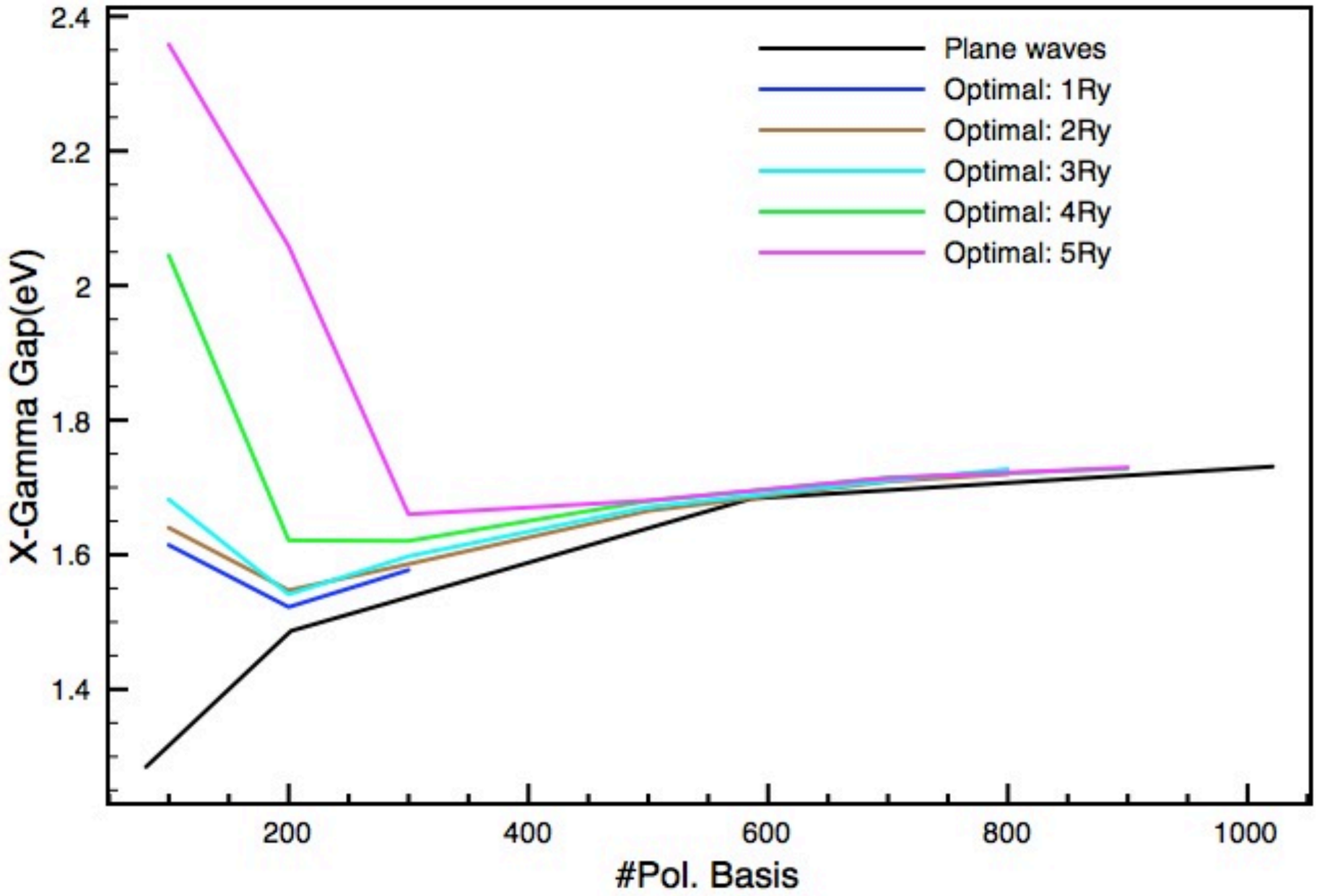
si_pw4gww_planewaves.in

```
&inputpw4gww
    prefix='si'
    num_nbndv(1)=16
    num_nbnds=32
    l_truncated_coulomb=.false.
    numw_prod=100
    pmat_cutoff=3d0
    pmat_type=5
    s_self_lanczos=1d-8
/
```

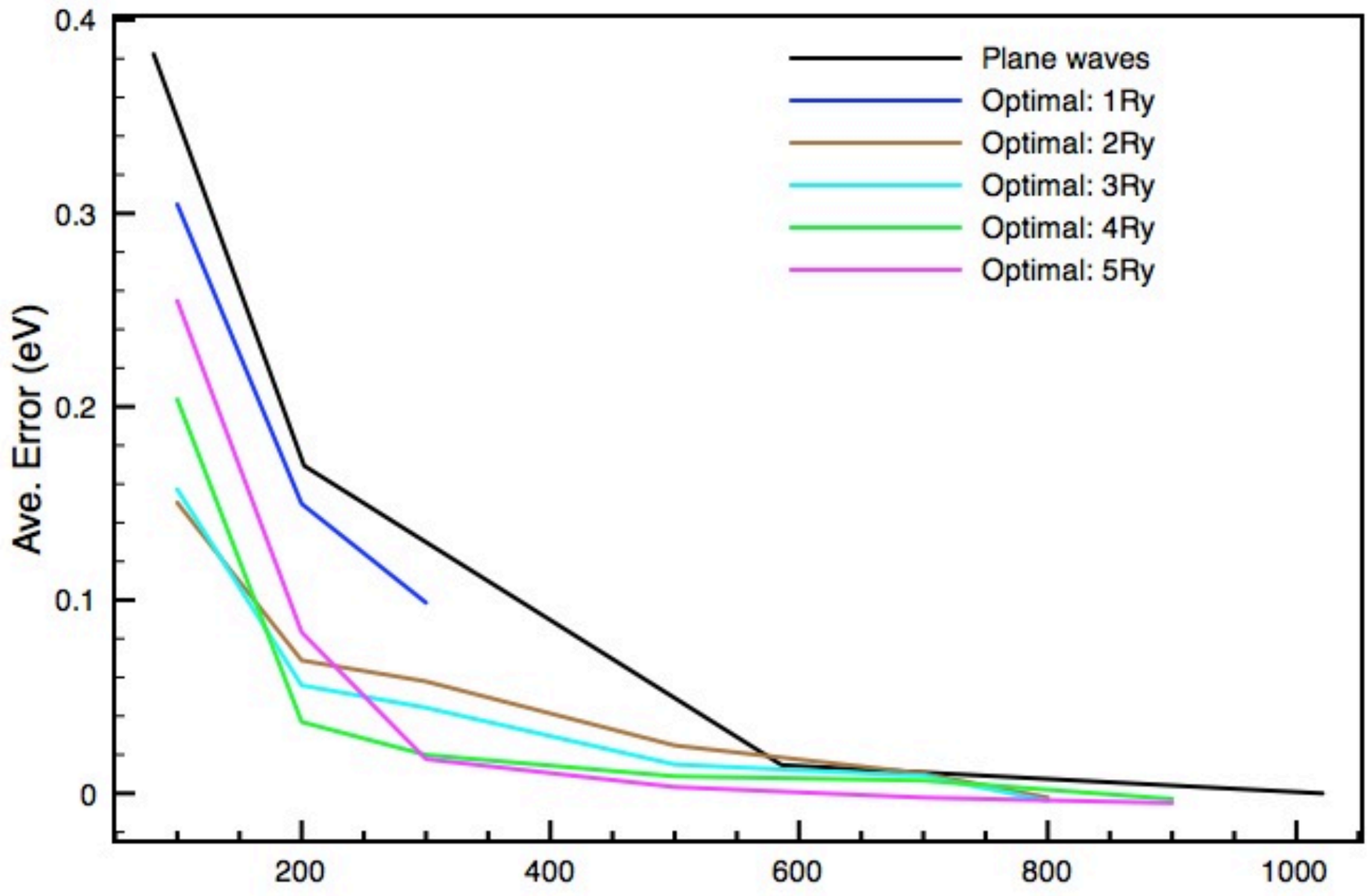
cutoff for plane-waves in Ry

select plane-waves basis set

Optimal basis vs Plane-waves: results



Optimal basis vs Plane-waves: results



Conclusions:

Always check the convergence of your results!!