

---

MAX School on Advanced Materials and Molecular Modelling  
with **QUANTUM ESPRESSO**

---

# **QE-2021: Hands-on session – Day-6**

## **TDDFPT**

**Iurii Timrov (EPFL, Switzerland)**

**Oscar Baseggio (SISSA, Italy)**

# Outline

1. **Part I** – Optical absorption in molecules
2. **Part II** – EELS in solids

(We will use HPC for this hands-on, because many examples are quite heavy)

# Part I – Optical absorption in molecules

1. **Example 1:** **turboDavidson** – from the Independent Particle Approximation (IPA) to interacting electrons
2. **Example 2:** **turboLanczos** – total spectra at low cost for interacting electrons

# turboDavidson program for calculations of absorption spectra

- The **turbo\_davidson.x** program allows us to calculate absorption spectra of molecules using time-dependent density functional perturbation theory (TDDFpT).
- The interactions of electrons (**Hartree** and **Exchange-Correlation effects**) are taken into account fully *ab initio* and self-consistently (or they can be neglected **if\_dft\_spectrum=.true.**).
- The electronic transitions from occupied to empty states can be analyzed by selecting a frequency range in which the transitions occur.
- However, calculations of overall absorption spectra in a wide frequency range are computationally rather expensive, because many eigenvalues of the Hamiltonian must be computed.

# turboDavidson program for calculations of absorption spectra

Coupled linear-response TDDFpT equations (**resonant** and **anti-resonant**):

$$\begin{aligned} (\hat{H}^o - \varepsilon_v - \hbar\omega) \tilde{\varphi}'_v(\mathbf{r}, \omega) + \hat{P}_c \hat{V}'_{\text{HXC}}[\tilde{\varphi}'_v(\mathbf{r}, \omega), \tilde{\varphi}'_v^*(\mathbf{r}, -\omega)] \varphi_v^o(\mathbf{r}) &= 0 \\ (\hat{H}^o - \varepsilon_v + \hbar\omega) \tilde{\varphi}'_v^*(\mathbf{r}, -\omega) + \hat{P}_c \hat{V}'_{\text{HXC}}[\tilde{\varphi}'_v(\mathbf{r}, \omega), \tilde{\varphi}'_v^*(\mathbf{r}, -\omega)] \varphi_v^o(\mathbf{r}) &= 0 \end{aligned}$$

These equations can be re-written as Casida's matrix equation:

$$\begin{pmatrix} 0 & \hat{D} \\ \hat{D} + \hat{K} & 0 \end{pmatrix} \begin{pmatrix} Q \\ P \end{pmatrix} = \omega \begin{pmatrix} Q \\ P \end{pmatrix}$$

**interaction terms**

where  $\hat{D} \longrightarrow (\hat{H}^o - \varepsilon_v)$  and  $\hat{K} \longrightarrow \hat{P}_c \hat{V}'_{\text{HXC}} \varphi_v^o(\mathbf{r})$

**Davidson algorithm** is used (the same algorithm as in the ground state SCF calculation) to solve the Casida's equations and to obtain the eigenvalues which are used for a calculation of the absorption coefficient.

# Independent Particle Approximation

**The simplest approximation:** Independent Particle Approximation (IPA) - a sum of independent excitations from occupied to empty states.

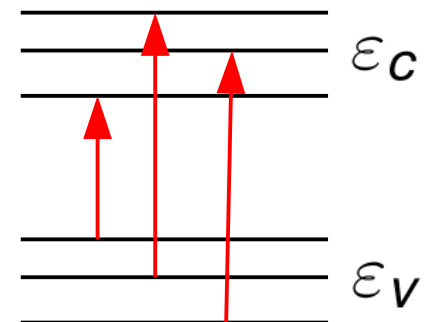
## Fermi's golden rule

The transition probability per unit time from occupied states  $|\varphi_v\rangle$  to empty states  $|\varphi_c\rangle$  reads:

$$T(\omega) = \frac{2\pi}{\hbar} \sum_{v,c} \delta(\varepsilon_c - \varepsilon_v - \hbar\omega) \left| \langle \varphi_c | V'_{ext} | \varphi_v \rangle \right|^2$$

$V'_{ext} = -e\mathbf{E} \cdot \mathbf{r}$  is the external potential induced by the electric field.

$\varepsilon_v, \varepsilon_c$  and  $|\varphi_v\rangle, |\varphi_c\rangle$  are the eigenvalues and the eigenfunctions of the ground-state Kohn-Sham equation

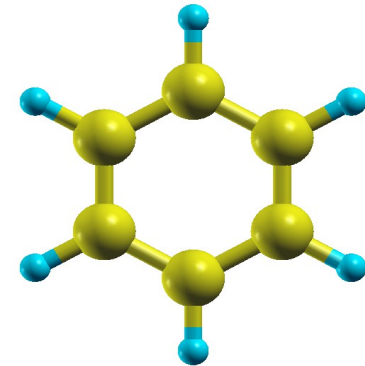
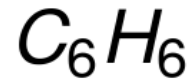


# Example 1: Input file for PWscf

**Step 1.** Perform a Self-Consistent Field ground-state calculation for benzene at the equilibrium structure using the **pw.x** program.

Specify the total number of Kohn-Sham bands (occupied + some empty states)

```
&CONTROL
  calculation = 'scf'
  restart_mode='from_scratch',
  prefix='Benzene'
  pseudo_dir = '../..pseudo'
  outdir='./tmp',
/
&SYSTEM
 ibrav = 6,
  celldm(1) = 32,
  celldm(3) = 0.83,
  nat = 12,
  ntyp = 2,
  ecutwfc = 30,
  nbnd = 20
/
&ELECTRONS
  conv_thr = 1.0d-8
  mixing_beta = 0.6,
/
ATOMIC_SPECIES
H 1.0 H.blyp-vbc.UPF
C 12.01 C.blyp-mt.UPF
ATOMIC_POSITIONS {Angstrom}
C 5.633200899 6.320861303 5.000000000
C 6.847051545 8.422621957 5.000000000
C 8.060751351 7.721904557 5.000000000
C 8.060707879 6.320636665 5.000000000
C 6.846898786 5.620067381 5.000000000
C 5.633279551 7.722134449 5.000000000
H 6.847254360 9.512254789 5.000000000
H 9.004364510 8.266639340 5.000000000
H 9.004297495 5.775895755 5.000000000
H 6.846845929 4.530522778 5.000000000
H 4.689556006 5.776237709 5.000000000
H 4.689791688 8.267023318 5.000000000
K_POINTS {gamma}
```



Input

Output

**pw.x < pw.benzene.scf.in > pw.benzene.scf.out**

# Example 1: Output file of PWscf

The output SCF file **pw.benzene.scf.out** :

```
End of self-consistent calculation
```

```
k = 0.0000 0.0000 0.0000 ( 37796 PWs) bands (ev):
```

```
-21.1410 -18.3897 -18.3884 -14.6742 -14.6729 -12.5121 -11.0334 -10.4379  
-9.8983 -9.8963 -8.5697 -7.8759 -7.8751 -5.8257 -5.8251 -0.7364  
-0.7362 -0.3332 0.2501 0.3207
```

```
highest occupied, lowest unoccupied level (ev): -5.8251 -0.7364
```

**HOMO** **LUMO**

**HOMO** – Highest Occupied Molecular Orbital

**LUMO** – Lowest Unoccupied Molecular Orbital

Energy gap = **LUMO** – **HOMO** = 5.1 (eV)



# Example 1: Calculation of absorption spectra using turboDavidson (IPA)

Perform the turboDavidson calculation without the electronic interaction using the **turbo\_davidson.x** program.

```
&lr_input  
  prefix = 'Benzene',  
  outdir = './tmp'  
/  
&lr_dav  
  if_dft_spectrum = .true.  
  num_init       = 30  
/
```

← Same as in PWscf

← This means we are using IPA

← Obsolete variable for the IPA case  
(will not be needed in future version of  
QE for the IPA case)

```
turbo_davidson.x < turbo_davidson.benzene.davidson.in >...  
...turbo_davidson.benzene.davidson.out
```

## Example 1: Post-processing step

Perform a spectrum calculation using the **turbo\_spectrum.x** program and using the eigenvalues computed in the previous step.

**turbo\_spectrum.Benzene.tddfpt\_pp.in :**

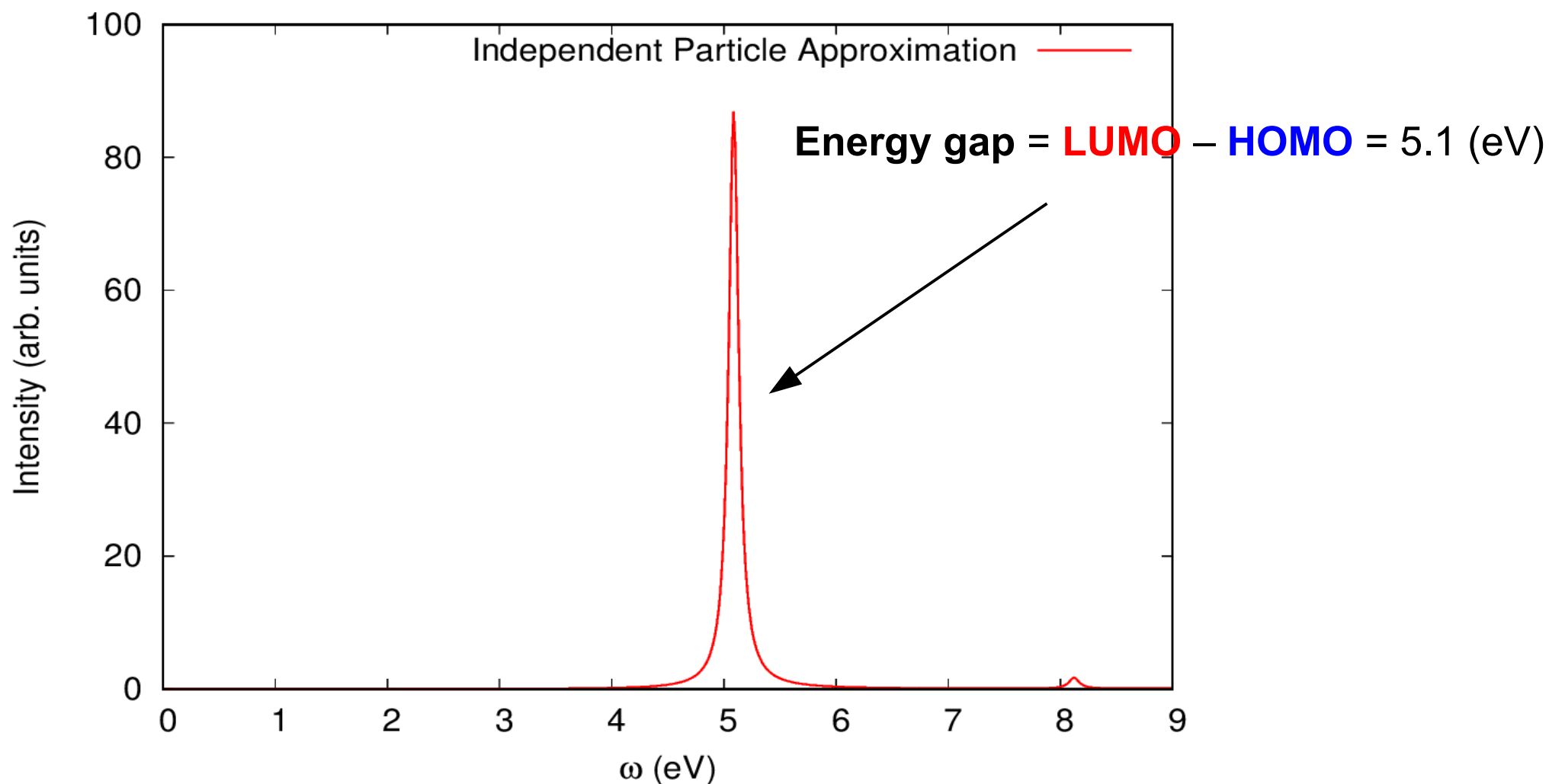
<pre>&amp;lr_input</pre>	
<pre>  prefix = 'Benzene',</pre>	← The same prefix as in the SCF calculation
<pre>  outdir = './tmp',</pre>	← Directory for temporary files
<pre>  td = 'davidson',</pre>	← Type of previous calculation
<pre>  epsil = 0.004,</pre>	← The value of Lorentzian smearing in Ry
<pre>  start = 0.0d0,</pre>	← Minimum value of frequencies for a plot in Ry
<pre>  end    = 1.0d0,</pre>	← Maximum value of frequencies for a plot in Ry
<pre>  increment = 0.0001d0,</pre>	← Frequency step in Ry
<pre>  eign_file = 'Benzene-dft.eigen'</pre>	← Frequency with Davidson eigenvalues
<pre>/</pre>	

**turbo\_spectrum.x < turbo\_spectrum.benzene.davidson.in >...**  
**...turbo\_spectrum.benzene.davidson.out**

# Example 1: Calculation of absorption spectra in IPA

Absorption spectrum of benzene in the Independent Particle Approximation (file `Benzene_spectrum.eps`):

Use the script `plot_spectrum.gp` to produce this figure.



# Example 1: Spectra including interactions

Perform the turboDavidson calculation without the electronic interaction using the **turbo\_davidson.x** program.

```
&lr_input
  prefix = 'Benzene',
  outdir='./tmp'
/
&lr_dav
  if_dft_spectrum = .false.
  num_eign = 15
  num_init = 30
  num_basis_max = 90
  residue_conv_thr = 1.0E-4
  start = 0.0
  finish = 1.0
  step = 0.001
  broadening = 0.005
  reference = 0.3
/
```

← The same prefix as in the SCF calculation

← Directory for temporary files

← **Switch on the interactions**

← Number of eigenvalues to be calculated

← Number of initial vectors

← Maximum number of basis allowed for the sub-basis

← Convergence threshold

← Minimum value of frequencies for a plot in Ry

← Maximum value of frequencies for a plot in Ry

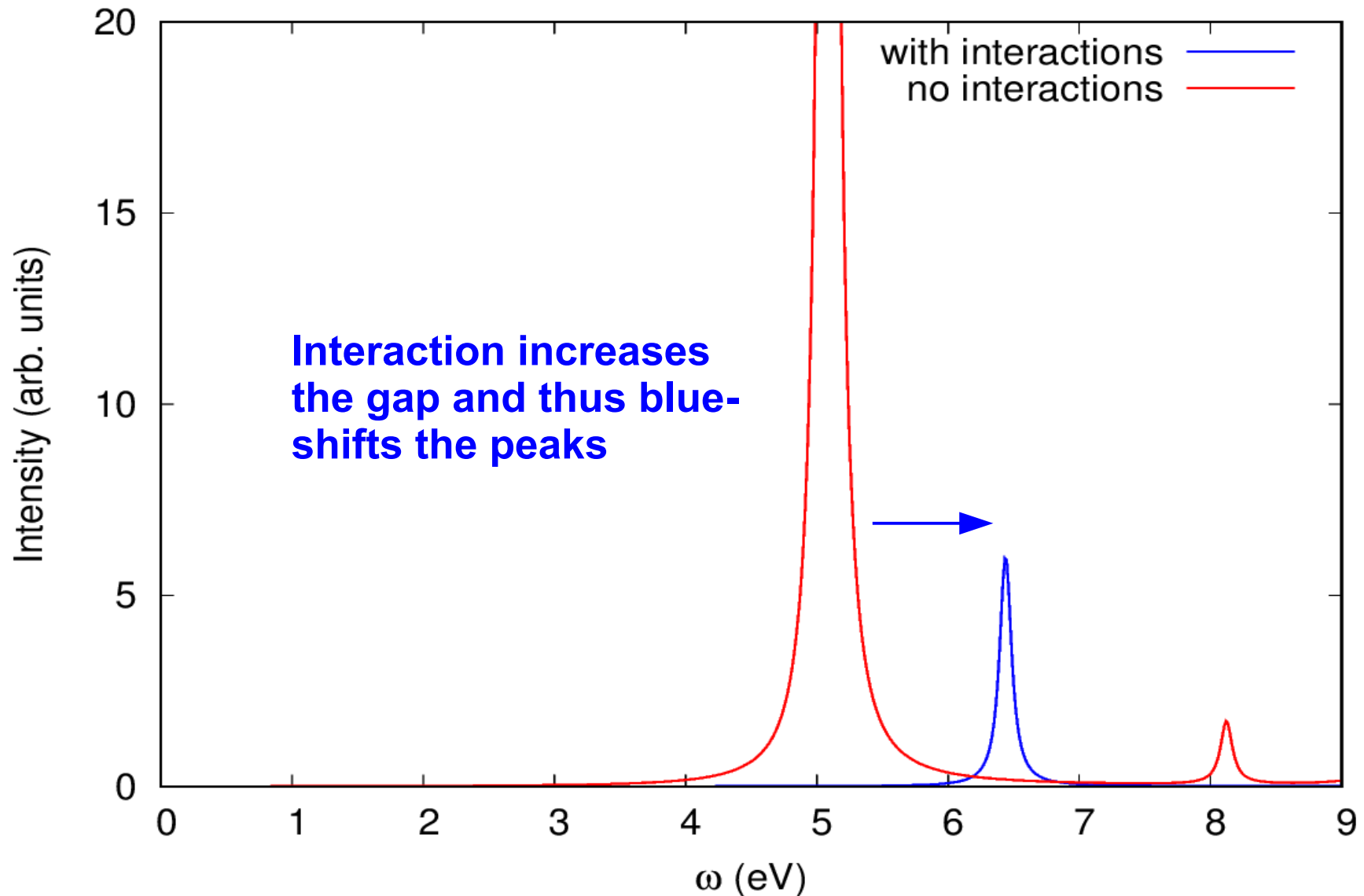
← Frequency step in Ry

← Lorentzian broadening parameter in Ry

← Reference frequency in Ry where the peak is expected

**turbo\_davidson.x < turbo\_davidson.benzene.davidson.in >...**  
**...turbo\_davidson.benzene.davidson.out**

# Example 1: Comparison of spectra with and without interactions



# turboLanczos program for calculation of absorption spectra

- The **turbo\_lanczos.x** program allows us to calculate absorption spectra of molecules using time-dependent density functional perturbation theory (TDDFpT) **without computing empty states!**
- The interactions of electrons (**Hartree** and **Exchange-Correlation effects**) are taken into account fully *ab initio* and self-consistently.
- The electronic transitions from occupied to empty states cannot be analyzed (use **turbo\_davidson.x** for this purpose).
- The overall absorption spectrum in a wide frequency range **can be calculated at once!**

# turboLanczos program for calculation of absorption spectra

Coupled linear-response TDDFpT equations (**resonant** and **anti-resonant**):

$$\begin{aligned}
 (\hat{H}^o - \varepsilon_v - \hbar\omega) \tilde{\varphi}'_v(\mathbf{r}, \omega) + \hat{P}_c \hat{V}'_{\text{HXC}}[\tilde{\varphi}'_v(\mathbf{r}, \omega), \tilde{\varphi}'_v^*(\mathbf{r}, -\omega)] \varphi_v^o(\mathbf{r}) &= -\hat{P}_c \hat{V}'_{\text{ext}} \varphi_v^o(\mathbf{r}) \\
 (\hat{H}^o - \varepsilon_v + \hbar\omega) \tilde{\varphi}'_v^*(\mathbf{r}, -\omega) + \hat{P}_c \hat{V}'_{\text{HXC}}[\tilde{\varphi}'_v(\mathbf{r}, \omega), \tilde{\varphi}'_v^*(\mathbf{r}, -\omega)] \varphi_v^o(\mathbf{r}) &= -\hat{P}_c \hat{V}'_{\text{ext}} \varphi_v^o(\mathbf{r})
 \end{aligned}$$

**perturbation**

These equations can be re-written as a quantum Liouville equation: **interaction terms**

$$\begin{aligned}
 (\omega - \mathcal{L}) \cdot \rho'(\omega) &= [\hat{V}'_{\text{ext}}, \rho^o] \\
 \mathcal{L} \cdot \rho' &\equiv [\hat{H}^o, \rho'] + [\hat{V}'_{\text{HXC}}, \rho^o]
 \end{aligned}$$

**Absorption coefficient** is computed as:

$$\alpha(\omega) = -(\mathbf{r}, (\omega - \mathcal{L})^{-1} [\mathbf{r}, \rho^o])$$

**Lanczos algorithm** is used to solve recursively quantum Liouville equation in the standard batch representation. This allows us to avoid inversions and multiplications of large matrices.

## Example 2: Calculation of absorption spectra using turboLanczos

**Step 1.** Perform a SCF ground-state calculation (no **nbnd** in the input):

```
pw.x < pw.benzene.scf.in > pw.benzene.scf.out
```

**Step 2.** Perform Lanczos recursions using the **turbo\_lanczos.x** program.

```
&lr_input  
  prefix = 'Benzene',  
  outdir='./tmp',  
  restart_step = 100,  
  restart = .false.  
/  
&lr_control  
  itermax = 500,  
  ipol = 1  
/
```

← The same prefix as in the SCF calculation

← Directory for temporary files

← The code writes restart files every restart\_step iterations

← Restart iterations after previous calculation

← Number of Lanczos iterations

← Polarization direction of incoming light, 1=x

```
turbo_lanczos.x < turbo_lanczos.benzene.lanczos.in >...  
...turbo_lanczos.benzene.lanczos.out
```



## Example 2: Post-processing step

Perform a spectrum calculation using the post-processing program **turbo\_spectrum.x** and using the Lanczos coefficients computed in the previous step.

```
&lr_input  
  prefix = 'Benzene',  
  outdir = './tmp',  
  itermax0 = 500,  
  itermax = 500,  
  extrapolation = 'no',  
  epsil = 0.004,  
  start = 0.0d0,  
  end = 1.0d0,  
  increment = 0.0001d0,  
  ipol = 1  
/
```

- ← The same prefix as in the SCF calculation
- ← Directory for temporary files
- ← Number of calculated Lanczos coefficient
- ← Number of extrapolated Lanczos coefficient
- ← Type of extrapolation
- ← The value of Lorentzian smearing in Ry
- ← Minimum value of frequencies for a plot in Ry
- ← Maximum value of frequencies for a plot in Ry
- ← Frequency step in Ry
- ← Polarization direction (same as in turbo\_lanczos.x)

```
turbo_spectrum.x < turbo_spectrum.benzene.in >...  
...turbo_spectrum.benzene.out
```

## Example 2: Post-processing step

### No extrapolation of Lanczos coefficients

turbo\_lanczos.benzene.lanczos.in

turbo\_spectrum.benzene.lanczos.in

```
&lr_input
  prefix = 'Benzene',
  outdir = './tmp',
  restart_step = 100,
  restart = .false.
/
&lr_control
  itermax = 500,
  ipol = 1
/
```

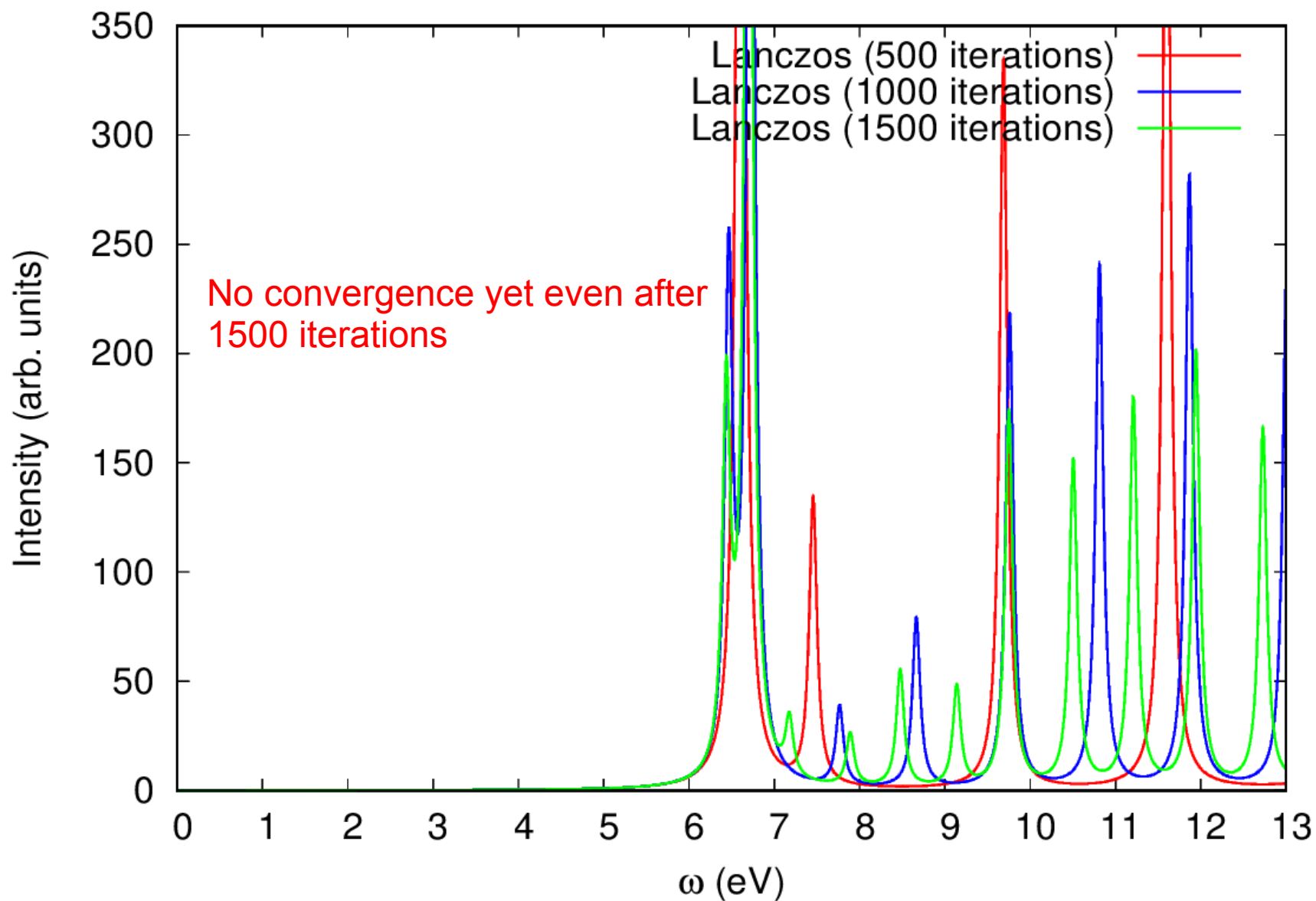
```
&lr_input
  prefix = 'Benzene',
  outdir = './tmp',
  itermax0 = 500,
  itermax = 500,
  extrapolation = 'no',
  epsil = 0.004,
  start = 0.0d0,
  end = 1.0d0,
  increment = 0.0001d0,
  ipol = 1
/
```

Must be equal !

You can use **restart = .true.** option in order not to start from scratch, when you want to increase the number of Lanczos iterations.

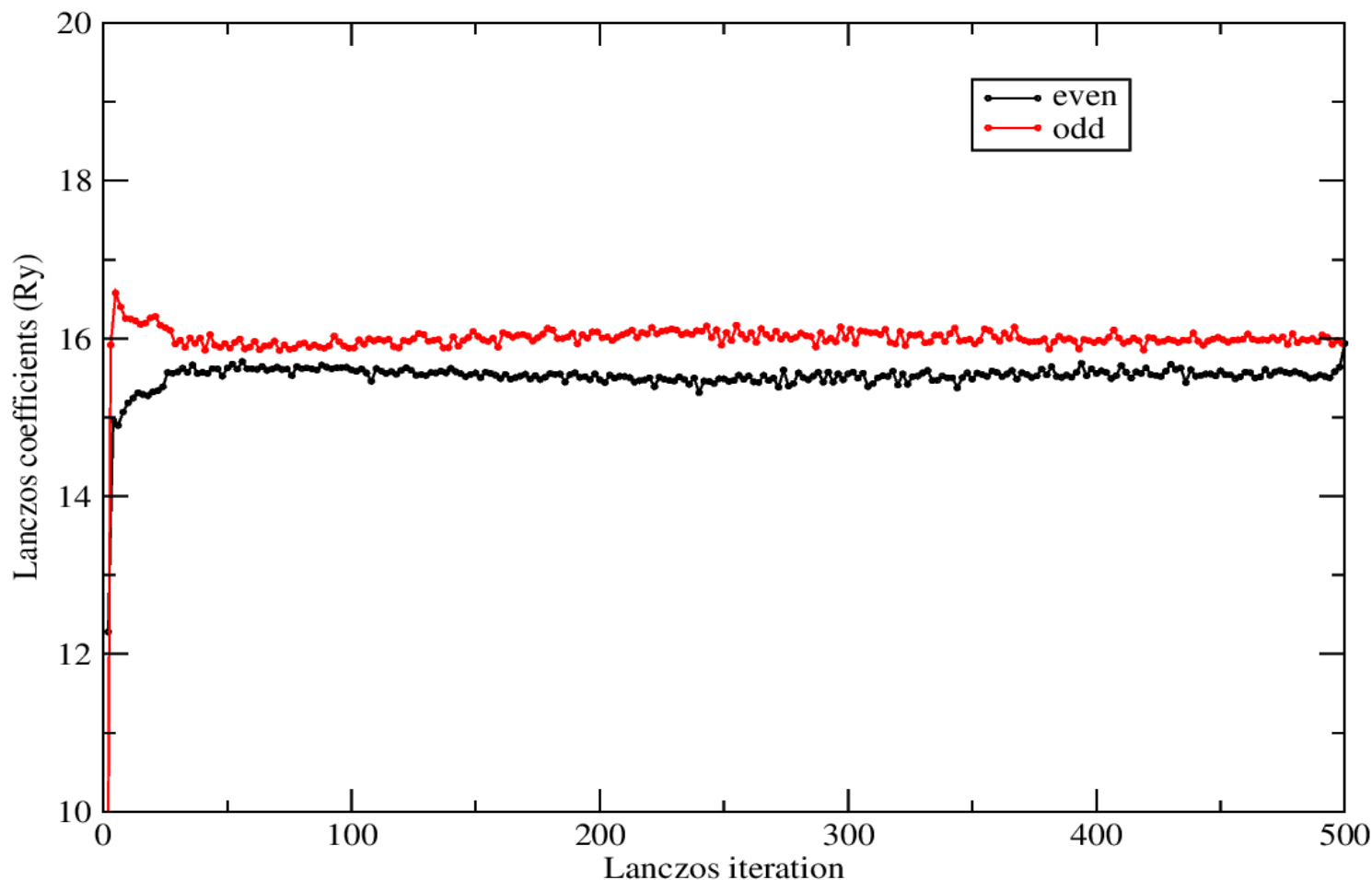
## Example 2: Post-processing step

### No extrapolation of Lanczos coefficients



## Example 2: Lanczos coefficients

Behavior of  $\beta$  Lanczos coefficients



Lanczos coefficients oscillate around half of the kinetic-energy cutoff ( $\sim 30/2 = 15$  Ry in this case). Hence, we can extrapolate these coefficients.

## Example 2: Extrapolation of Lanczos coefficients

In the Liouville-Lanczos approach, the absorption coefficient is computed as:

$$\alpha(\omega) = (\zeta_N, (\omega - \hat{T}_N)^{-1} \cdot \mathbf{e}_1^N)$$

where  $\hat{T}_N$  is the tridiagonal matrix composed of the Lanczos coefficients.

$$\hat{T}_N = \begin{pmatrix} 0 & \gamma_2 & 0 & \cdots & 0 \\ \beta_2 & 0 & \gamma_3 & 0 & 0 \\ 0 & \beta_3 & 0 & \ddots & 0 \\ \vdots & 0 & \ddots & \ddots & \gamma_N \\ 0 & \cdots & 0 & \beta_N & 0 \end{pmatrix} \xrightarrow{\text{Extrapolation}} \hat{T}_{N'} = \begin{pmatrix} 0 & \gamma_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \beta_2 & 0 & \gamma_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta_3 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \gamma_N & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_N & 0 & \langle \gamma \rangle & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \langle \beta \rangle & 0 & \langle \gamma \rangle & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddots & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \langle \beta \rangle & 0 & \langle \gamma \rangle \end{pmatrix}$$

$$\langle \beta \rangle = \frac{\beta_1 + \beta_2 + \dots + \beta_N}{N}$$

$$\langle \gamma \rangle = \frac{\gamma_1 + \gamma_2 + \dots + \gamma_N}{N}$$

$$N = \text{itermax0}, \quad N' = \text{itermax}$$

## Example 2: Post-processing step

### With extrapolation of Lanczos coefficients

turbo\_lanczos.benzene.lanczos.in

turbo\_spectrum.benzene.lanczos.in

```
&lr_input
  prefix = 'Benzene',
  outdir = './tmp',
  restart_step = 100,
  restart = .false.
/
&lr_control
  itermax = 500,
  ipol = 1
/
```

```
&lr_input
  prefix = 'Benzene',
  outdir = './tmp',
  itermax0 = 500,
  itermax = 20000,
  extrapolation = 'osc',
  epsil = 0.004,
  start = 0.0d0,
  end = 1.0d0,
  increment = 0.0001d0,
  ipol = 
/
```

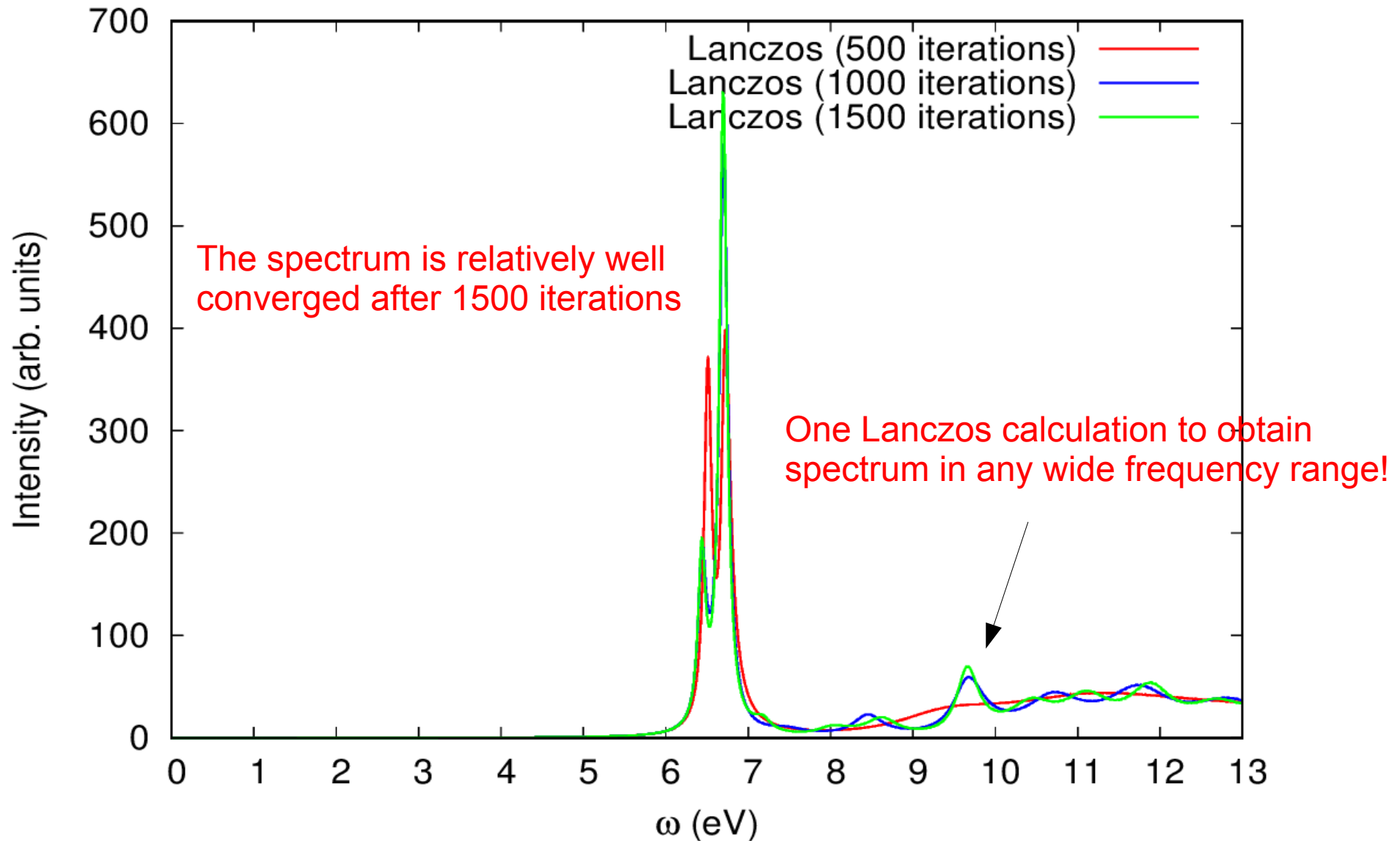
Must be equal !

Extrapolation until 20000

You can use `restart = .true.` option in order not to start from scratch, when you want to increase the number of Lanczos iterations.

## Example 2: Post-processing step

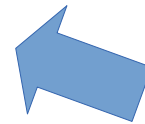
With extrapolation of Lanczos coefficients



# Part II – EELS in solids

1. **Example 3:** EELS spectrum of Si using the **turboEELS** code (**Lanczos algorithm**)

2. **Example 4:** EELS spectrum of Si using the **turboEELS** code (**Sternheimer algorithm**)



**OPTIONAL**  
**(very slow)**



# Basic equations

Coupled (**resonant** and **anti-resonant**) frequency-dependent **Sternheimer** equations:

$$\begin{aligned}
 (\hat{H}_{\mathbf{k}+\mathbf{q}}^{\circ} - \varepsilon_{n,\mathbf{k}}^{\circ} - \omega) \tilde{u}'_{n,\mathbf{k}+\mathbf{q}}(\mathbf{r}, \omega) + \boxed{\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{HXC},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})} &= \boxed{-\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{ext},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})} \\
 (\hat{H}_{\mathbf{k}+\mathbf{q}}^{\circ} - \varepsilon_{n,\mathbf{k}}^{\circ} + \omega) \tilde{u}'_{n,-\mathbf{k}-\mathbf{q}}(\mathbf{r}, -\omega) + \boxed{\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{HXC},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})} &= \boxed{-\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{ext},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})}
 \end{aligned}$$

**interaction terms**
**perturbation**

These equations can be re-written as a quantum Liouville equation and solved using the **Lanczos** algorithm:

$$\begin{aligned}
 (\omega - \mathcal{L}) \cdot \hat{\rho}'_{\mathbf{q}}(\omega) &= [\tilde{V}'_{\text{ext},\mathbf{q}}(\omega), \hat{\rho}^{\circ}] \\
 \mathcal{L} \cdot \hat{\rho}'_{\mathbf{q}} &\equiv [\hat{H}^{\circ}, \hat{\rho}'_{\mathbf{q}}] + [\tilde{V}'_{\text{HXC},\mathbf{q}}, \hat{\rho}^{\circ}]
 \end{aligned}$$

$\chi(\mathbf{q}, \mathbf{q}; \omega)$   
**Susceptibility**

$$-\text{Im}[\varepsilon^{-1}(\mathbf{q}, \omega)] = -\frac{4\pi e^2}{|\mathbf{q}|^2} \text{Im}[\chi(\mathbf{q}, \mathbf{q}; \omega)]$$

**Loss function**

# Example 3

Coupled (**resonant** and **anti-resonant**) frequency-dependent **Sternheimer** equations:

$$\begin{aligned}
 (\hat{H}_{\mathbf{k}+\mathbf{q}}^{\circ} - \varepsilon_{n,\mathbf{k}}^{\circ} - \omega) \tilde{u}'_{n,\mathbf{k}+\mathbf{q}}(\mathbf{r}, \omega) + \hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{HXC},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r}) &= -\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{ext},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r}) \\
 (\hat{H}_{\mathbf{k}+\mathbf{q}}^{\circ} - \varepsilon_{n,\mathbf{k}}^{\circ} + \omega) \tilde{u}'_{n,-\mathbf{k}-\mathbf{q}}(\mathbf{r}, -\omega) + \hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{HXC},\mathbf{q}}(\mathbf{r}, -\omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r}) &= -\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{ext},\mathbf{q}}(\mathbf{r}, -\omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})
 \end{aligned}$$

Let's use the Lanczos algorithm




interaction terms

perturbation


These equations can be re-written as a quantum Liouville equation and solved using the **Lanczos** algorithm:

$$\begin{aligned}
 (\omega - \mathcal{L}) \cdot \hat{\rho}'_{\mathbf{q}}(\omega) &= [\tilde{V}'_{\text{ext},\mathbf{q}}(\omega), \hat{\rho}^{\circ}] \\
 \mathcal{L} \cdot \hat{\rho}'_{\mathbf{q}} &\equiv [\hat{H}^{\circ}, \hat{\rho}'_{\mathbf{q}}] + [\tilde{V}'_{\text{HXC},\mathbf{q}}, \hat{\rho}^{\circ}]
 \end{aligned}$$



$\chi(\mathbf{q}, \mathbf{q}; \omega)$

Susceptibility



Loss function

$$-\text{Im}[\varepsilon^{-1}(\mathbf{q}, \omega)] = -\frac{4\pi e^2}{|\mathbf{q}|^2} \text{Im}[\chi(\mathbf{q}, \mathbf{q}; \omega)]$$

## Example 3: Input file for PWscf

```
&CONTROL
  calculation = 'scf'
  restart_mode='from_scratch',
  prefix='silicon'
  pseudo_dir = '../..../pseudo',
  outdir='./tmp'
/
&SYSTEM
 ibrav = 2,
celldm(1) = 10.26,
nat = 2,
ntyp = 1,
ecutwfc = 16.0
/
&ELECTRONS
  conv_thr = 1.0d-8
  mixing_beta = 0.7,
/
ATOMIC_SPECIES {alat}
Si 28.08 Si.pz-vbc.UPF
ATOMIC_POSITIONS {alat}
Si 0.00 0.00 0.00
Si 0.25 0.25 0.25
K_POINTS {automatic}
10 10 10 1 1 1
```

Step 1. Perform a Self-Consistent Field ground-state calculation for silicon using the **pw.x** program.

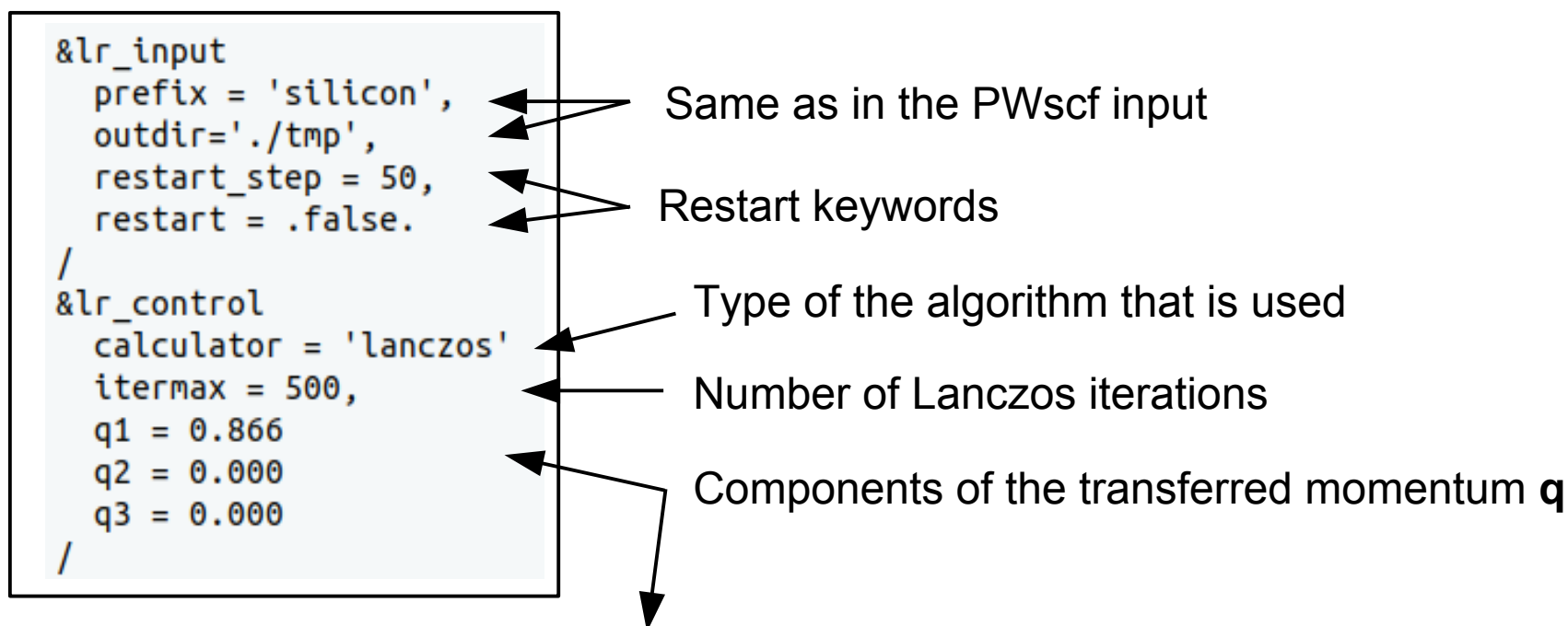
Input

Output

**pw.x < pw.si.scf.in > pw.si.scf.out**

## Example 3 : Input file for turboEELS

**Step 2.** Perform Lanczos recursions using the **turbo\_eels.x** program in order to compute Lanczos coefficients, which will be needed in Step 3.



How to specify q1, q2, q3 ? -> See next slide

**turbo\_eels.x < turbo\_eels.si.lanczos.in > turbo\_eels.si.lanczos.out**

## Example 3: Transferred momentum specification

$$\mathbf{q} = \frac{2\pi}{a_0} (q_1, q_2, q_3)$$

$$a_0 = \text{celldm}(1) = 10.26 \text{ Bohr} \quad - \text{ lattice parameter}$$

**Example:**  $|\mathbf{q}| = 0.53 \text{ Bohr}^{-1}$  and  $\mathbf{q} \parallel [100]$  .

Therefore, we obtain:

$$q_1 = \frac{|\mathbf{q}| a_0}{2\pi} = 0.866$$

$$q_2 = 0$$

$$q_3 = 0$$

## Example 3: Post-processing step

**Step 3.** Perform a spectrum calculation using the post-processing program **turbo\_spectrum.x** and using the Lanczos coefficients computed in the previous step.

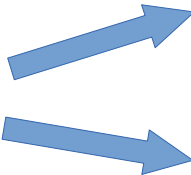
```
&lr_input
  prefix = 'silicon',
  outdir = './tmp',
  eels = .true.
  itermax0 = 500,
  itermax = 500,
  extrapolation = 'no',
  epsil = 0.035,
  units = 1,
  start = 0.0d0,
  end = 50.0d0,
  increment = 0.01d0
/
```

- ← The same prefix as in the SCF calculation
- ← Directory for temporary files
- ← **Must be .true. for EELS, otherwise absorption**
- ← Number of calculated Lanczos coefficient
- ← Number of extrapolated Lanczos coefficient
- ← Type of extrapolation (bi-constant)
- ← The value of Lorentzian smearing in Ry
- ← The units for start, end and increment
- ← Minimum value of frequencies for a plot in eV
- ← Maximum value of frequencies for a plot in eV
- ← Frequency step in eV

**turbo\_spectrum.x < turbo\_spectrum.si.in > turbo\_spectrum.si.out**

## Example 3: Computed quantities

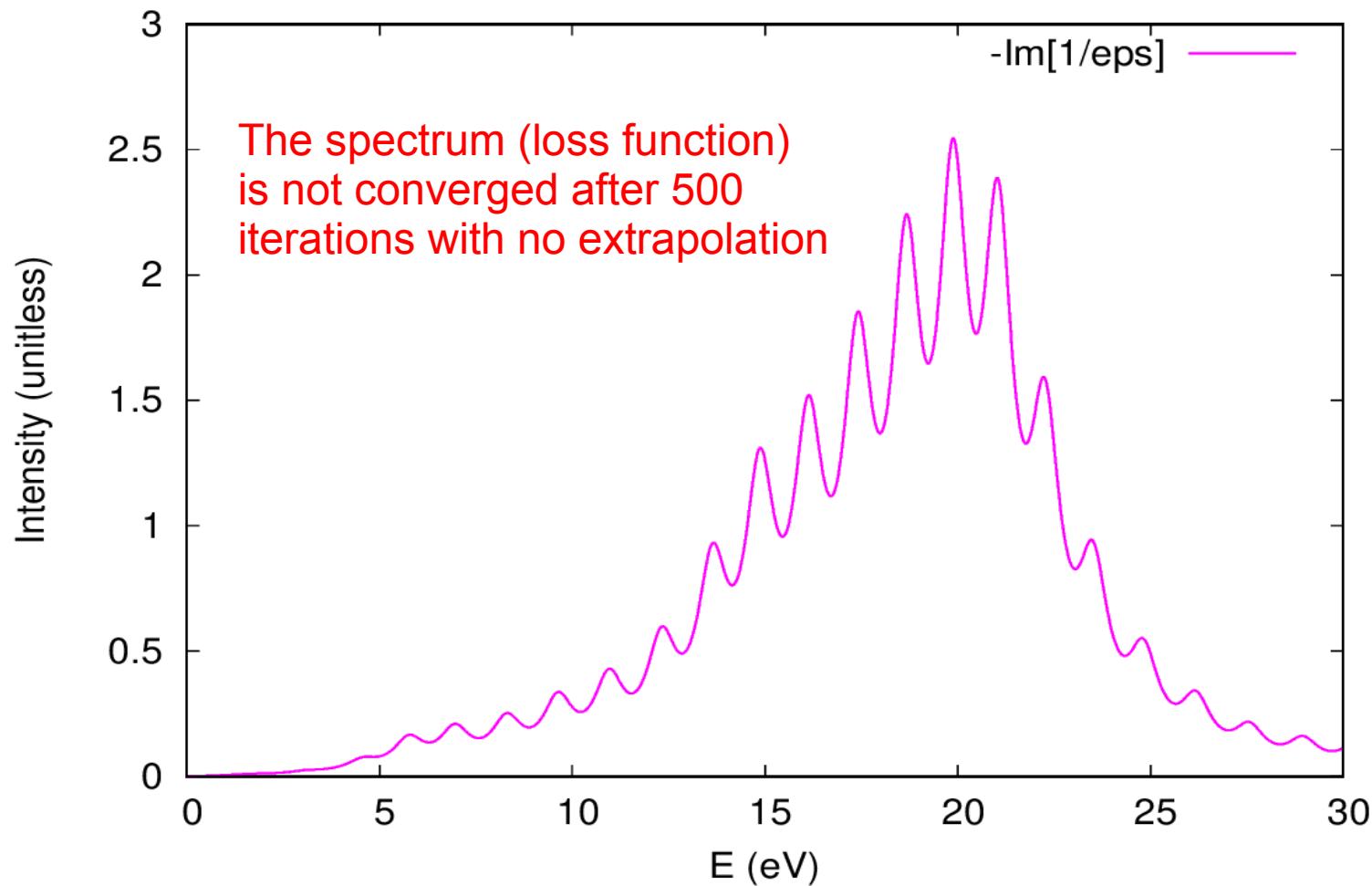
The code `turbo_spectrum.x` produces a file `silicon.plot_eps.dat` which contains the real and imaginary parts of the dielectric function and its inverse:

$$\varepsilon^{-1}(\mathbf{q}, \omega) = 1 + \frac{4\pi e^2}{|\mathbf{q}|^2} \chi(\mathbf{q}, \mathbf{q}; \omega)$$

$$\begin{aligned}\operatorname{Re}[\varepsilon^{-1}(\mathbf{q}, \omega)] &= 1 + \frac{4\pi e^2}{|\mathbf{q}|^2} \operatorname{Re}[\chi(\mathbf{q}, \mathbf{q}; \omega)] \\ -\operatorname{Im}[\varepsilon^{-1}(\mathbf{q}, \omega)] &= -\frac{4\pi e^2}{|\mathbf{q}|^2} \operatorname{Im}[\chi(\mathbf{q}, \mathbf{q}; \omega)]\end{aligned}$$

$$\varepsilon(\mathbf{q}, \omega) = \frac{1}{\varepsilon^{-1}(\mathbf{q}, \omega)} = \underbrace{\frac{\operatorname{Re}[\varepsilon^{-1}]}{(\operatorname{Re}[\varepsilon^{-1}])^2 + (\operatorname{Im}[\varepsilon^{-1}])^2}}_{\operatorname{Re}[\varepsilon(\mathbf{q}, \omega)]} + i \underbrace{\frac{-\operatorname{Im}[\varepsilon^{-1}]}{(\operatorname{Re}[\varepsilon^{-1}])^2 + (\operatorname{Im}[\varepsilon^{-1}])^2}}_{\operatorname{Im}[\varepsilon(\mathbf{q}, \omega)]}$$

# Example 3: The computed EELS

## No extrapolation of Lanczos coefficients

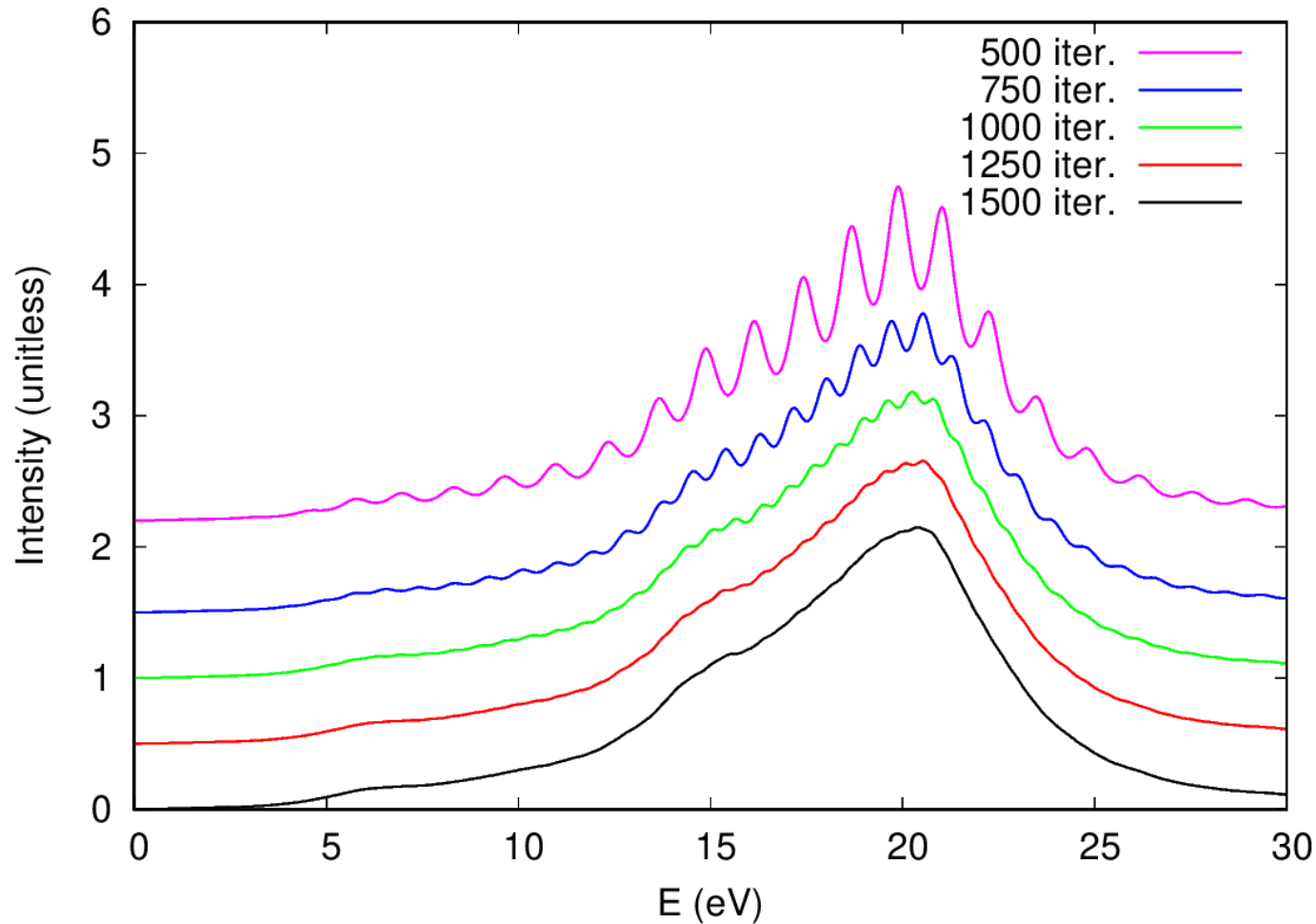


Here, **k** points mesh 10x10x10 1 1 1 is used.



# Example 3: Convergence of EELS

**No extrapolation of Lanczos coefficients**



Here, **k** points mesh 10x10x10 1 1 1 is used.

## Example 3: Post-processing step

**Step 3.** Perform a spectrum calculation using the post-processing program **turbo\_spectrum.x** and using the Lanczos coefficients computed in the previous step.

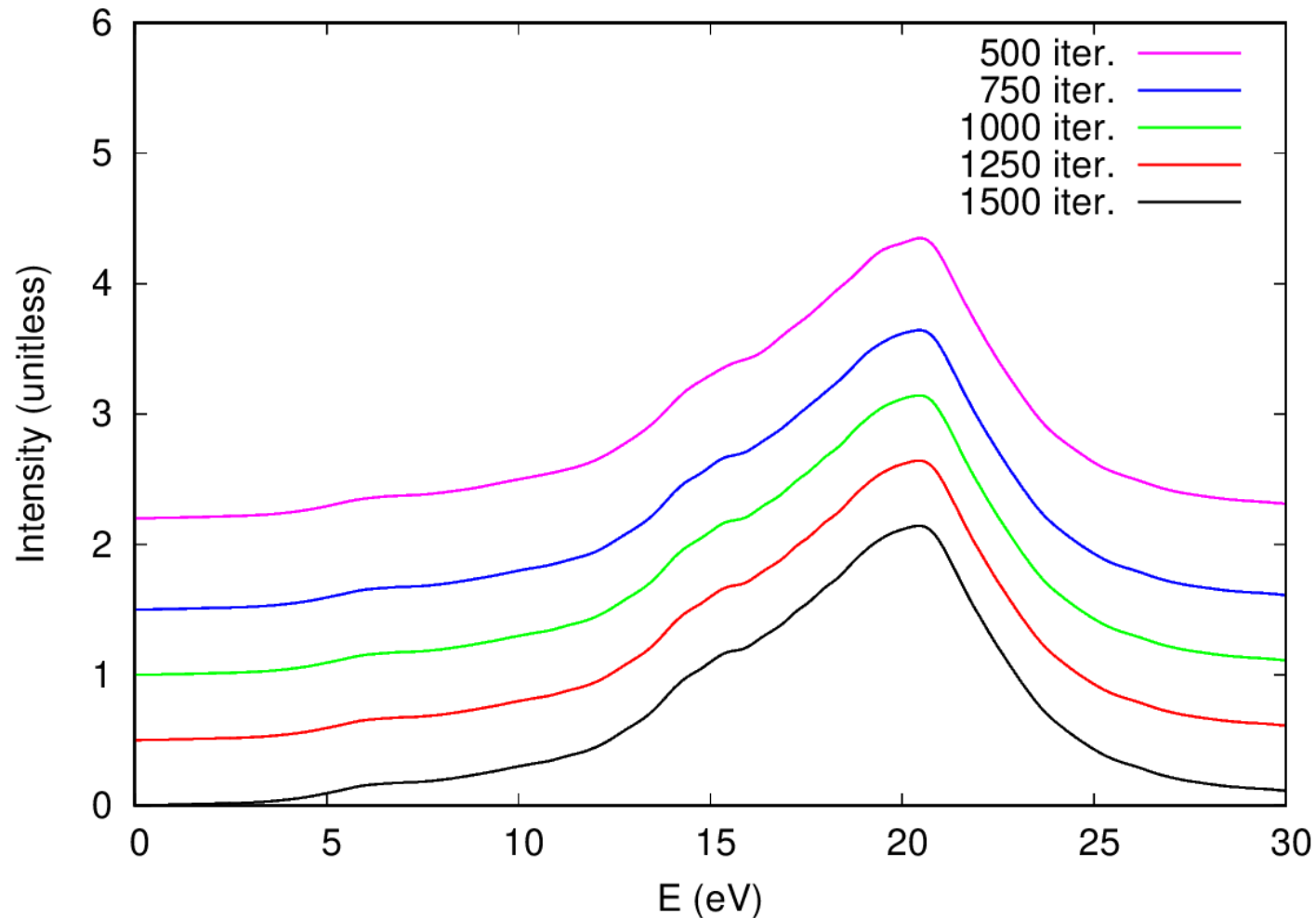
```
&lr_input
  prefix = 'silicon',
  outdir = './tmp',
  eels = .true.
  itermax0 = 500,
  itermax = 20000
  extrapolation = 'osc'
  epsil = 0.035,
  units = 1,
  start = 0.0d0,
  end = 50.0d0,
  increment = 0.01d0
/
```

- ← The same prefix as in the SCF calculation
- ← Directory for temporary files
- ← **Must be .true. for EELS, otherwise absorption**
- ← Number of calculated Lanczos coefficient
- ← Number of extrapolated Lanczos coefficient
- ← Type of extrapolation (bi-constant)
- ← The value of Lorentzian smearing in Ry
- ← The units for start, end and increment
- ← Minimum value of frequencies for a plot in eV
- ← Maximum value of frequencies for a plot in eV
- ← Frequency step in eV

**turbo\_spectrum.x < turbo\_spectrum.si.in > turbo\_spectrum.si.out**

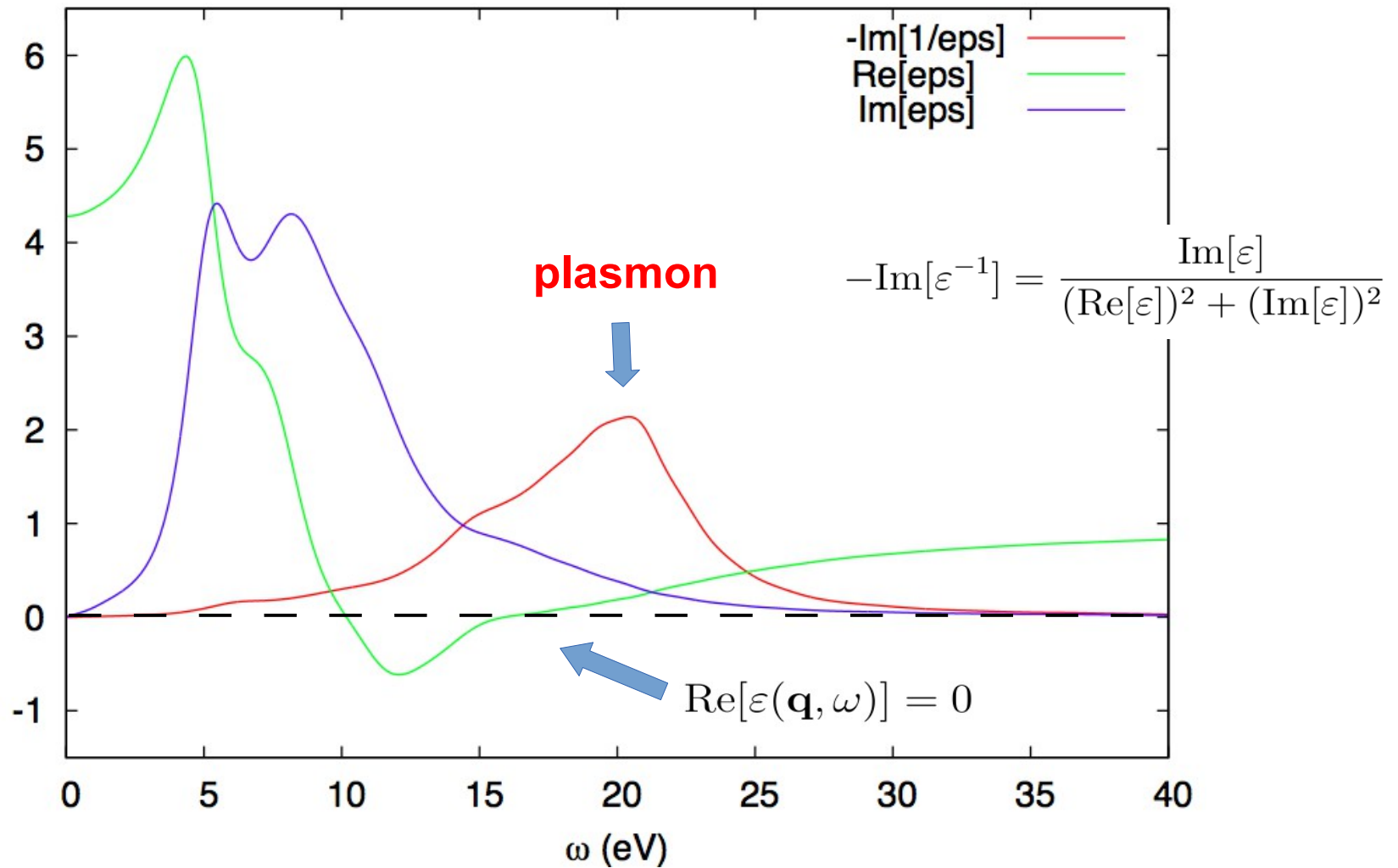
# Example 3: Convergence of EELS

**With extrapolation of Lanczos coefficients**



Here, k points mesh 10x10x10 1 1 1 was used.

## Example 3: Where is the plasmon peak?



## Example 4

Coupled (**resonant** and **anti-resonant**) frequency-dependent **Sternheimer** equations:

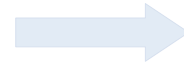
$$\begin{aligned}
 (\hat{H}_{\mathbf{k}+\mathbf{q}}^{\circ} - \varepsilon_{n,\mathbf{k}}^{\circ} - \omega) \tilde{u}'_{n,\mathbf{k}+\mathbf{q}}(\mathbf{r}, \omega) + \boxed{\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{HXC},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})} &= \boxed{-\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{ext},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})} \\
 (\hat{H}_{\mathbf{k}+\mathbf{q}}^{\circ} - \varepsilon_{n,\mathbf{k}}^{\circ} + \omega) \tilde{u}'_{n,-\mathbf{k}-\mathbf{q}}(\mathbf{r}, -\omega) + \boxed{\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{HXC},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})} &= \boxed{-\hat{P}_c^{\mathbf{k}+\mathbf{q}} \tilde{v}'_{\text{ext},\mathbf{q}}(\mathbf{r}, \omega) u_{n,\mathbf{k}}^{\circ}(\mathbf{r})}
 \end{aligned}$$

**interaction terms**
**perturbation**

These equations can be re-written as a quantum Liouville equation and solved using the **Lanczos** algorithm:

**Let's use the Sternheimer algorithm**

$$\begin{aligned}
 (\omega - \mathcal{L}) \cdot \hat{\rho}'_{\mathbf{q}}(\omega) &= [\tilde{V}'_{\text{HXC},\mathbf{q}}(\omega), \hat{\rho}^{\circ}] \\
 \mathcal{L} \cdot \hat{\rho}'_{\mathbf{q}} &\equiv [\hat{H}^{\circ}, \hat{\rho}'_{\mathbf{q}}] + [\tilde{V}'_{\text{HXC},\mathbf{q}}, \hat{\rho}^{\circ}]
 \end{aligned}$$



$\chi(\mathbf{q}, \mathbf{q}; \omega)$   
**Susceptibility**



$$-\text{Im}[\varepsilon^{-1}(\mathbf{q}, \omega)] = -\frac{4\pi e^2}{|\mathbf{q}|^2} \text{Im}[\chi(\mathbf{q}, \mathbf{q}; \omega)]$$

**Loss function**

## Example 4 : Input file for turboEELS

Step 1. Perform a SCF ground-state calculation for Si.

Step 2. Solve the Sternheimer equations using **turbo\_eels.x**

```
&lr_input
  prefix = 'silicon',
  outdir='./tmp'
/
&lr_control
  calculator = 'sternheimer',
  q1 = 0.866,
  q2 = 0.000,
  q3 = 0.000,
  epsil = 0.035,
  units = 1,
  start = 0.0d0,
  end   = 30.0d0,
  increment = 1.0d0
/
```

Same as in the PWscf input

Type of the algorithm that is used

Components of the transferred momentum  $\mathbf{q}$

The value of Lorentzian broadening in Ry

The units (0  $\rightarrow$  Ry, 1  $\rightarrow$  eV)

Keywords that control the range in which the spectrum is plotted

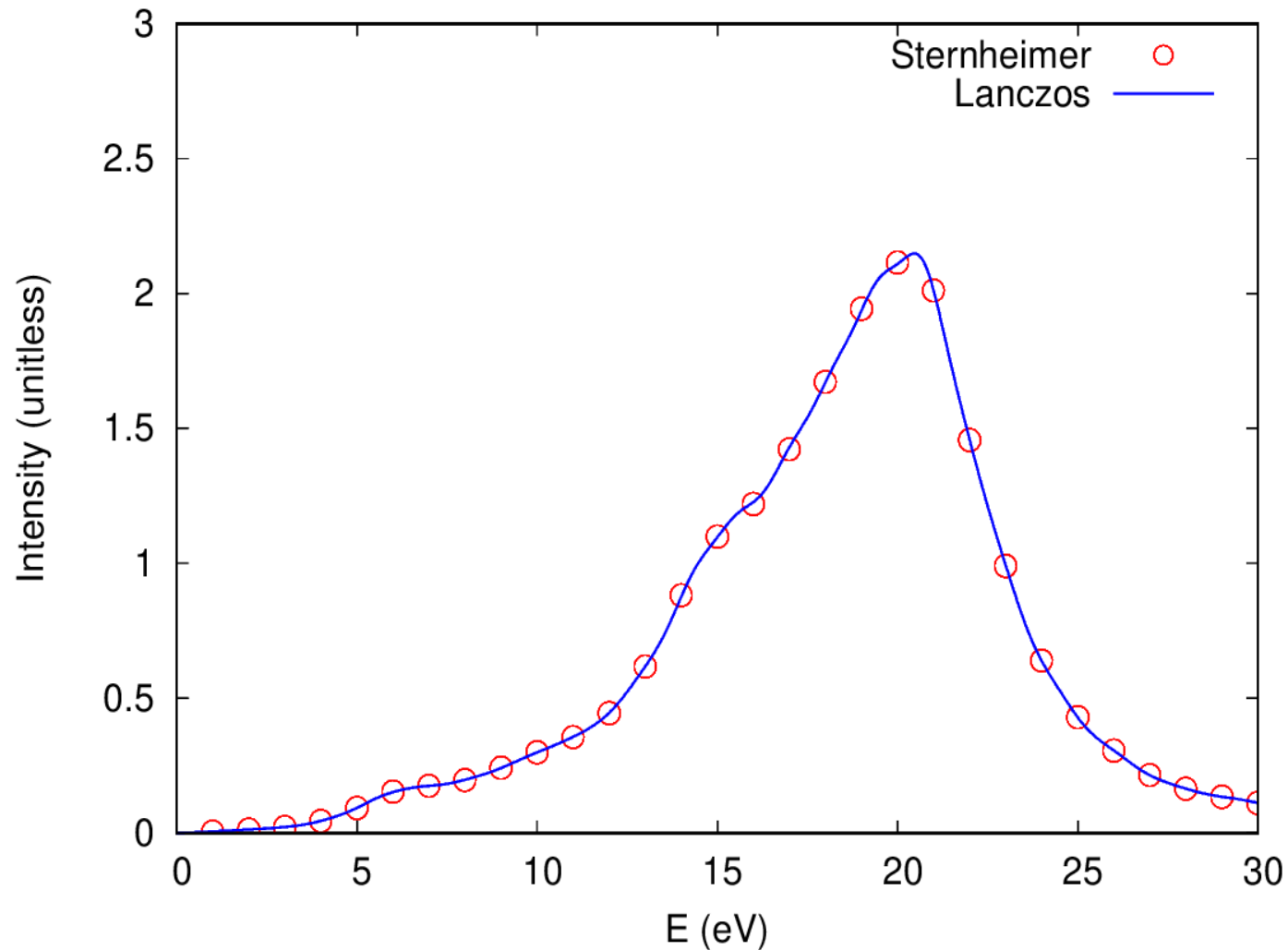
**turbo\_eels.x < turbo\_eels.si.sternheimer.in > ...**

**...turbo\_eels.si.sternheimer.out**

## Example 4: The computed EELS

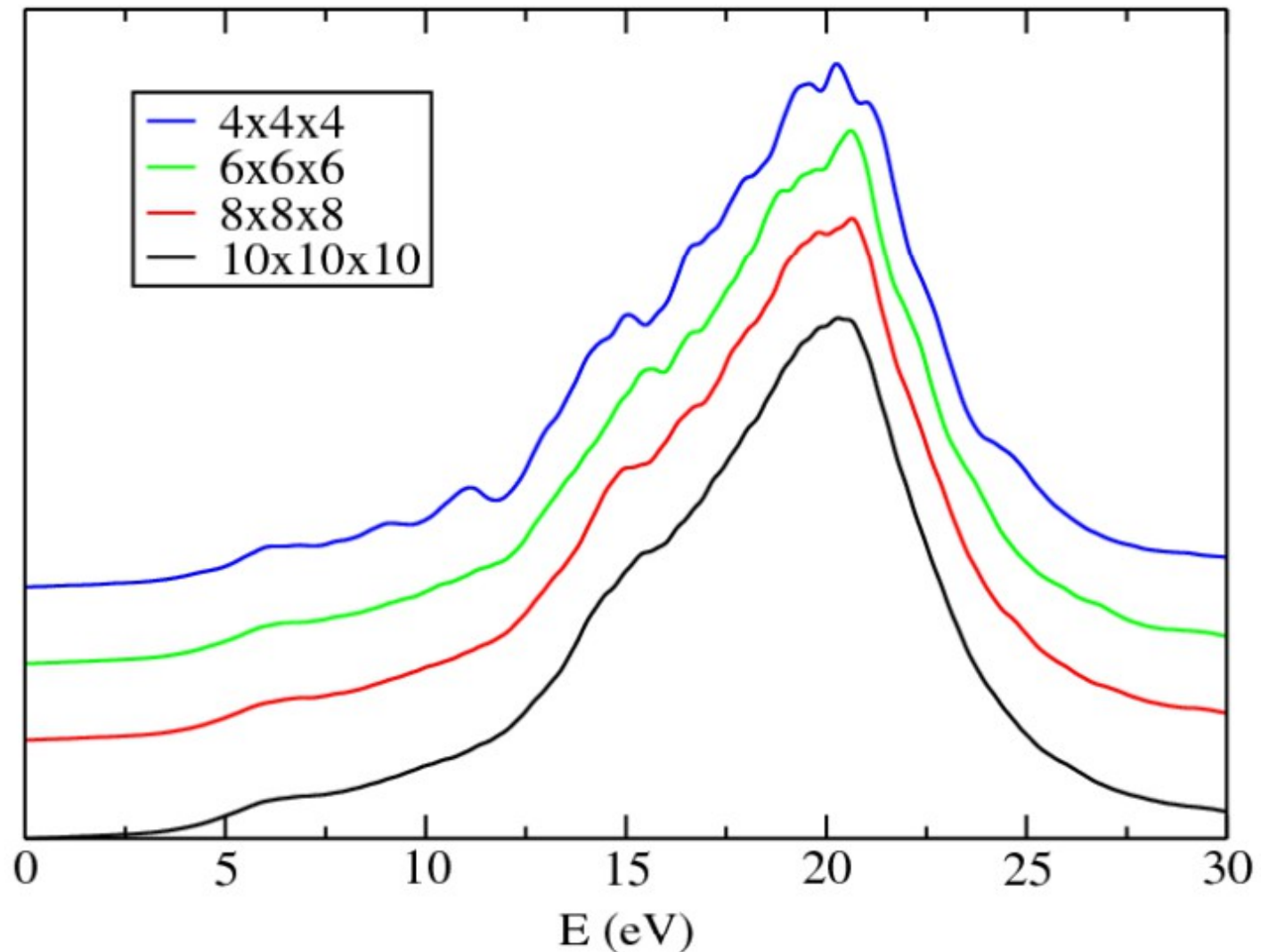
The Sternheimer and Lanczos algorithms give exactly the same spectrum.

Sternheimer is much slower than Lanczos!



## Example 4: Convergence w.r.t. **k** mesh

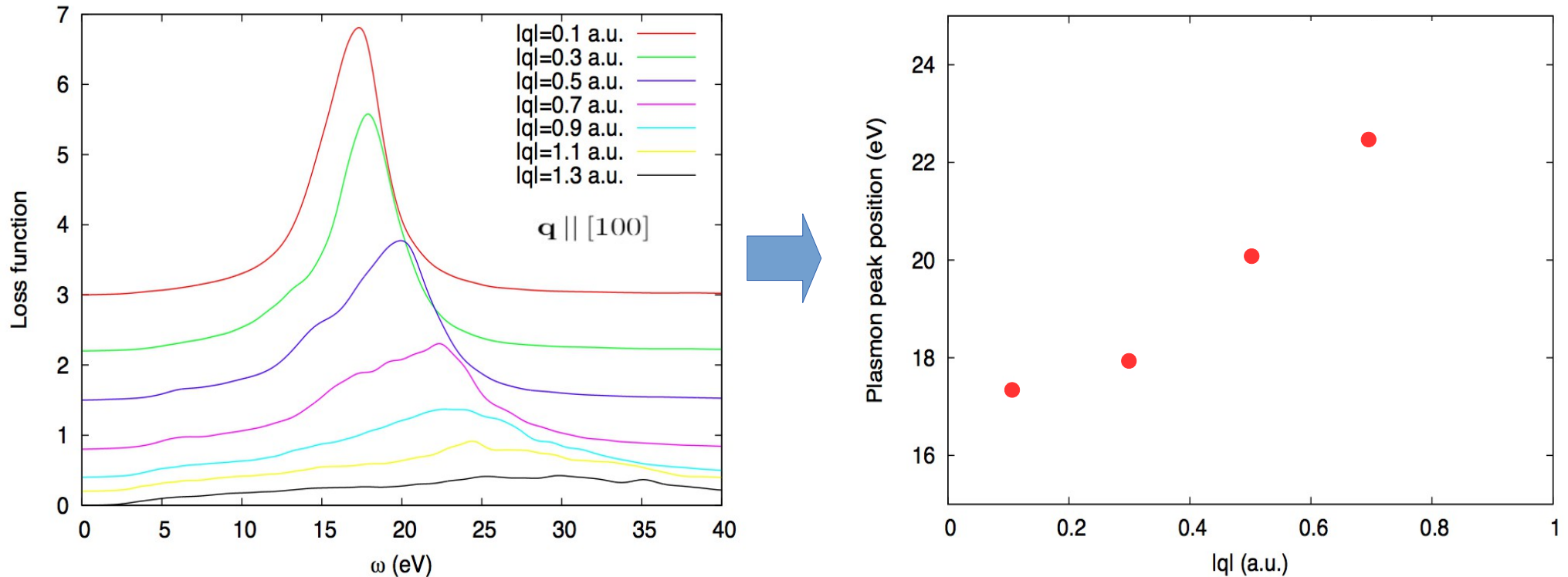
Convergence of EELS with respect to the size of the **k** points mesh





# Example 4: Plasmon dispersion

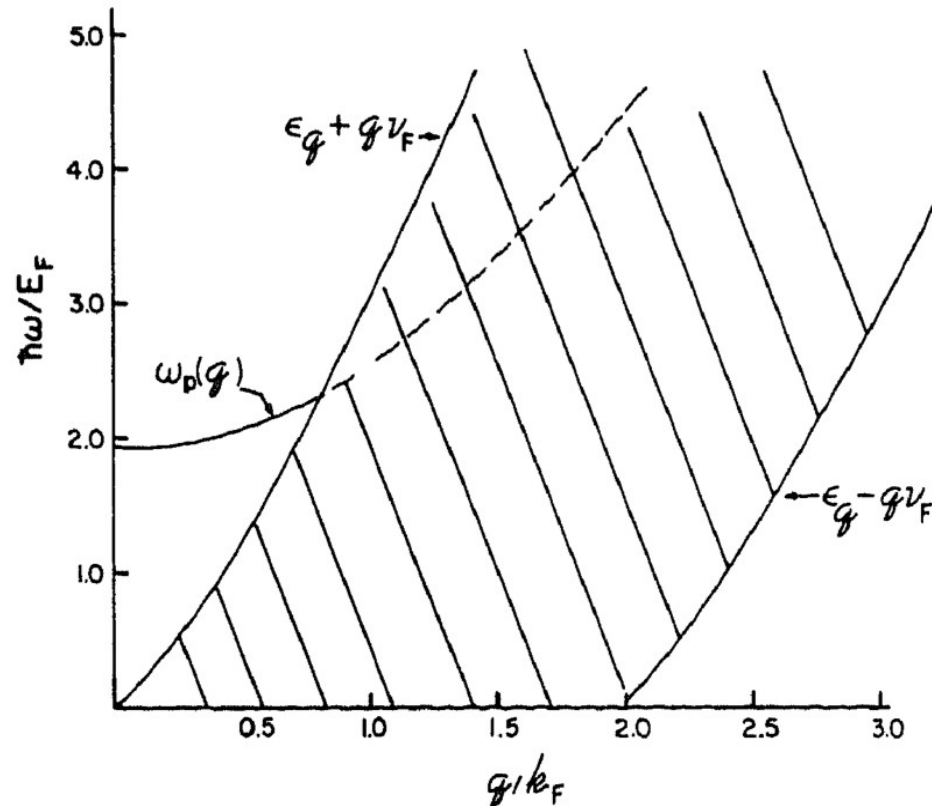
## Plasmon dispersion in bulk silicon



By increasing  $q$ , the plasmon peak blue-shifts and broadens, because it enters in the electron-hole continuum.

## Example 4: Plasmon dispersion and e-h continuum

The excitation region of the electron gas in  $(\mathbf{q}, \omega)$  space



The plasmon line becomes strongly damped in a region of electron-hole pairs.

*G. D. Mahan, Many-Particle Physics, Plenum Press, 2nd ed. New York, (1975)*