Excited states properties for SIESTA calculations: time-dependent density functional theory and beyond

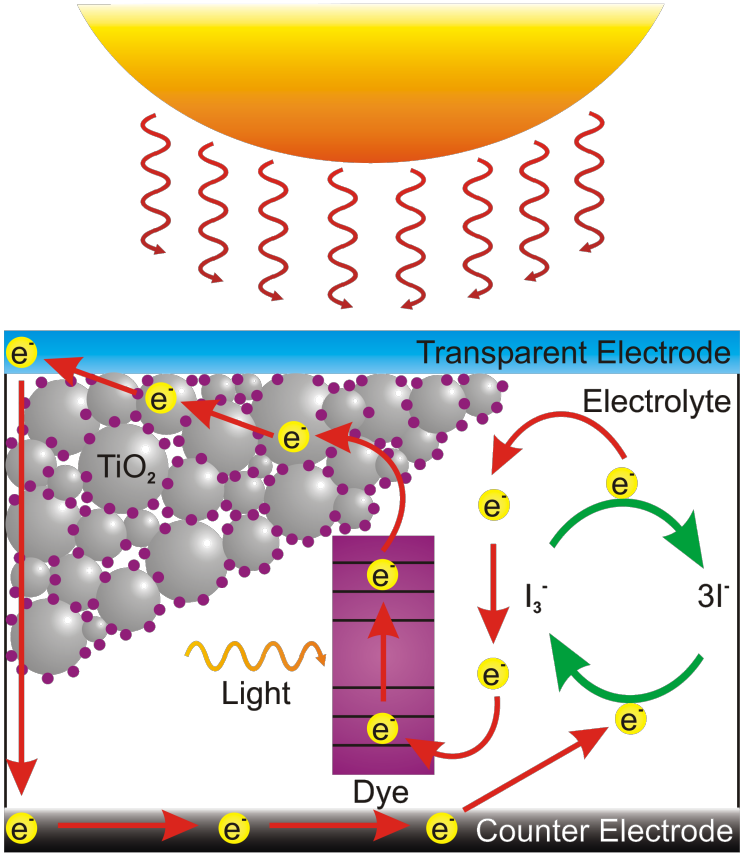
P. Koval, F. Marchesin, M. Barbry, and D. Sánchez-Portal

Imaginenano2018, Industrial Forum
Bilbao, Spain, March 13–15 2018 (March 13)
http://www.imaginenano.com/2018

Outline

- Why excited states are interesting?
- Generic workflow for excited states simulations
- A glimpse on the methods for excited states
- Programing solution to model excited states
- Applications: selected

Dye sensitized solar cells

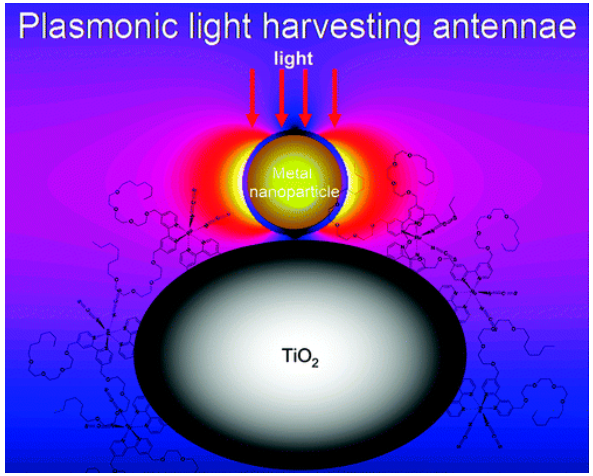


Michael Grätzel

- How effective a dye is absorbing: ??? for QM
- Does electrolyte modifies the absorption: ??? for QM&MM

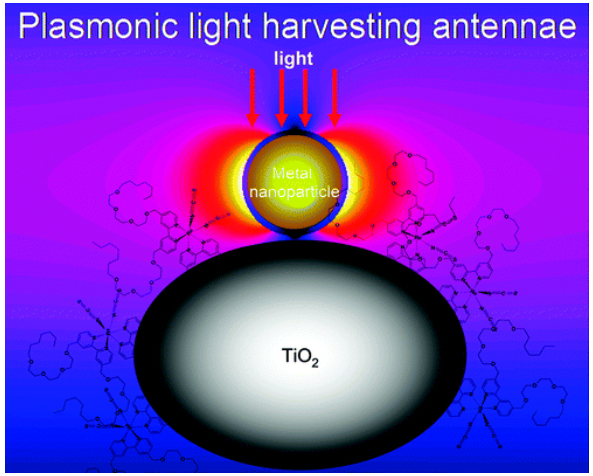
Plasmonics: applications of local field enhancement

Enhancement of efficiency in photo voltaics

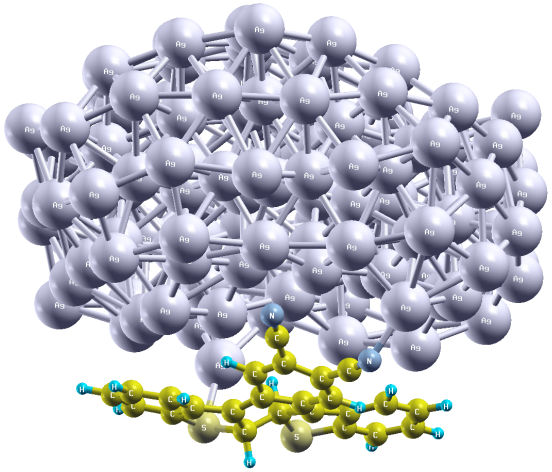


Plasmonics: applications of local field enhancement

Enhancement of efficiency in photo voltaics



Surface-enhanced Raman Spectroscopy SERS/TERS



Generic workflow for excited states simulations

- Density-functional theory (DFT) $E = E[n]$

Generic workflow for excited states simulations

- Density-functional theory (DFT) $E = E[n]$
- siesta Kohn-Sham scheme

$$\hat{H}_{\mathrm{KS}}[n]\Psi_n(\boldsymbol{r}) = E\Psi_n(\boldsymbol{r})$$

$$n(\boldsymbol{r}) = \sum_{n \in \mathrm{occ}} \Psi_n^*(\boldsymbol{r})\Psi_n(\boldsymbol{r})$$

Generic workflow for excited states simulations

- Density-functional theory (DFT) $E = E[n]$
- siesta Kohn-Sham scheme

$$\hat{H}_{\mathrm{KS}}[n]\Psi_n(\boldsymbol{r}) = E\Psi_n(\boldsymbol{r})$$

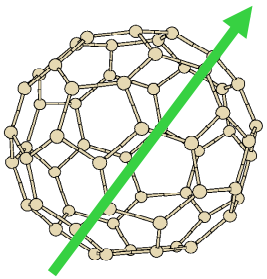
$$n(\boldsymbol{r}) = \sum_{n\in\mathrm{occ}} \Psi_n^*(\boldsymbol{r})\Psi_n(\boldsymbol{r})$$

- EXCITED STATES: Time-dependent DFT (TDDFT) or GW+BSE

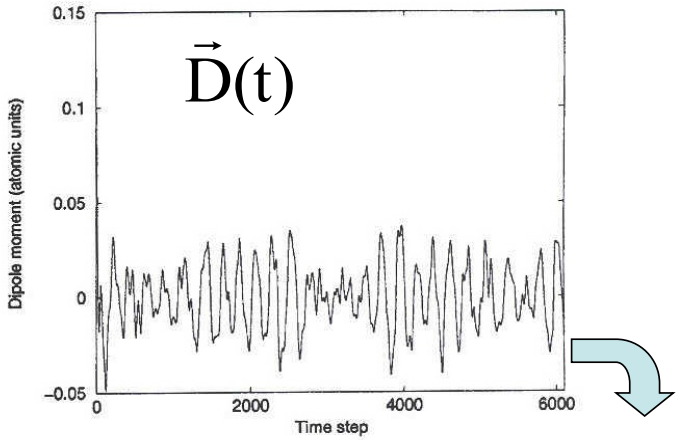
$$\hat{H}_{\mathrm{KS}}\Psi_n(\boldsymbol{r},t) = \mathrm{i}\frac{\partial}{\partial t}\Psi_n(\boldsymbol{r},t)$$

Ab-initio theory: real-time propagation of wave-packets

A. Tsolakidis, D. Sánchez-Portal, R. M. Martin, PRB(2002), adapting a recipe by Kazuhiro Yabana and George Bertsch, PRB (1996)

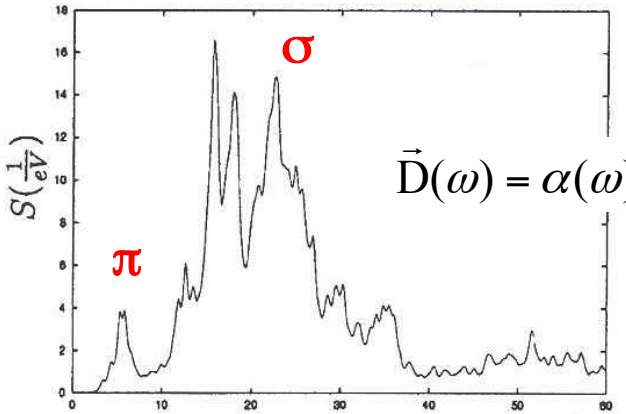


$\vec{E}$



Low energy peaks with TDLDA

SIESTA	Yabana	Exp.
3.5	3.4	3.8
4.4	4.3	
5.4	5.3	4.8
5.8	6.0	5.8



$$\vec{D}(\omega) = \alpha(\omega)\,\vec{E}(\omega)$$

RT TDDFT is capable to model a broad range of scenarious, but this is superfluous for spectroscopy

Ab-initio theory: linear response

- Casida equation: exchange/hybrid functionals ☺

Ab-initio theory: linear response

- Casida equation: exchange/hybrid functionals ☺
- Formalism of linear response functions $\chi(\omega)$: comput. cheap ☺

$$\delta n(\boldsymbol{r},\omega) \equiv \int \chi(\boldsymbol{r},\boldsymbol{r}',\omega)\delta V_{\text{ext}}(\boldsymbol{r}',\omega)dr', \text{ or } \Rightarrow \chi(\omega) = \frac{\delta n}{\delta V_{\text{ext}}}$$

$$V_{\text{eff}} = V_{\text{ext}} + V_{\text{Hxc}}[n] \Rightarrow \qquad \frac{\delta V_{\text{eff}}}{\delta n} = \frac{\delta V_{\text{ext}}}{\delta n} + \frac{\delta V_{\text{Hxc}}[n]}{\delta n}$$

$$\chi(\omega) = \chi_0(\omega) + \chi_0(\omega) K \chi(\omega),$$

$$\chi_0(\omega) = (f_n - f_m)\frac{\Psi_n(\boldsymbol{r})\Psi_m(\boldsymbol{r})\Psi_m(\boldsymbol{r}')\Psi_n(\boldsymbol{r}')}{\omega - (E_m - E_n)}$$

Products of wave-functions appear in $\chi_0(\omega)$
and interaction kernel K is known for semi-local functionals.

Ab-initio theory: linear response

- Formalism of linear response functions $\chi(\omega)$: comput. cheap ☺

$$\delta n(\boldsymbol{r},\omega) \equiv \int \chi(\boldsymbol{r},\boldsymbol{r}',\omega)\delta V_{\text{ext}}(\boldsymbol{r}',\omega)dr', \text{ or } \Rightarrow \chi(\omega) = \frac{\delta n}{\delta V_{\text{ext}}}$$

$$V_{\text{eff}} = V_{\text{ext}} + V_{\text{Hxc}}[n] \Rightarrow \qquad \frac{\delta V_{\text{eff}}}{\delta n} = \frac{\delta V_{\text{ext}}}{\delta n} + \frac{\delta V_{\text{Hxc}}[n]}{\delta n}$$

$$\chi(\omega) = \chi_0(\omega) + \chi_0(\omega) K \chi(\omega),$$

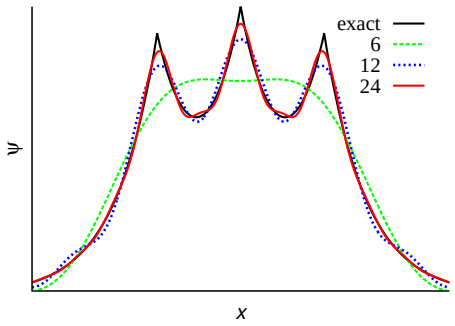
$$\chi_0(\omega) = (f_n - f_m)\frac{\Psi_n(\boldsymbol{r})\Psi_m(\boldsymbol{r})\Psi_m(\boldsymbol{r}')\Psi_n(\boldsymbol{r}')}{\omega - (E_m - E_n)}$$

Products of wave-functions appear in $\chi_0(\omega)$
and interaction kernel K is known for semi-local functionals.

Basis sets

$$\Psi_n(\boldsymbol{r}) = \sum_a X_a^n f^a(\boldsymbol{r})$$

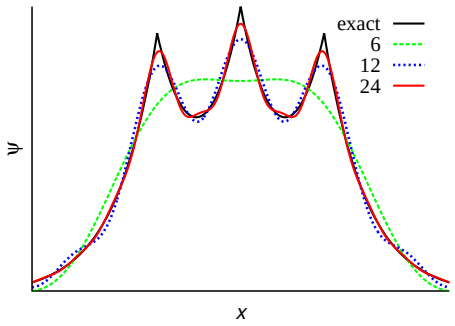
Plane-wave (PW) basis sets: $\exp(\mathrm{i}\boldsymbol{G}\boldsymbol{r})$



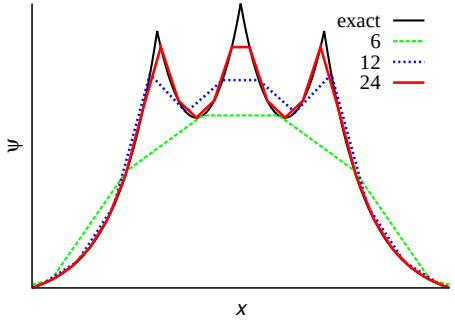
Basis sets

$$\Psi_n(\boldsymbol{r}) = \sum_a X_a^n f^a(\boldsymbol{r})$$

Plane-wave (PW) basis sets: $\exp(\mathrm{i}\boldsymbol{G}\boldsymbol{r})$



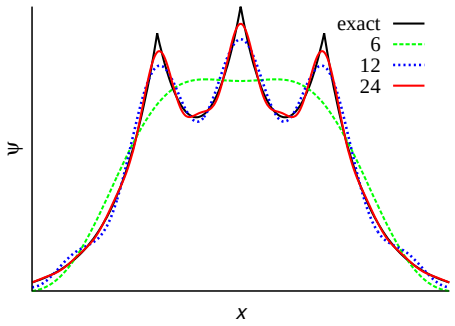
Real-space grids (RSG): $\{\boldsymbol{r}_i\}, i = 1 \ldots N$



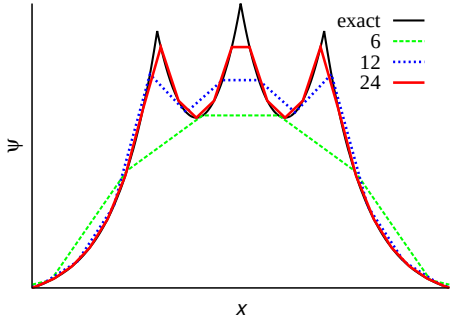
Basis sets

$$\Psi_n(\boldsymbol{r}) = \sum_a X_a^n f^a(\boldsymbol{r})$$

Plane-wave (PW) basis sets: $\exp(\mathrm{i}\boldsymbol{G}\boldsymbol{r})$



Real-space grids (RSG): $\{\boldsymbol{r}_i\}, i = 1 \ldots N$



- Numerical atomic orbitals (NAO): $f^a(r)\mathrm{Y}_{lm}(\boldsymbol{r})$
  - Parsimonious for atomic systems
  - Need density-fitting basis

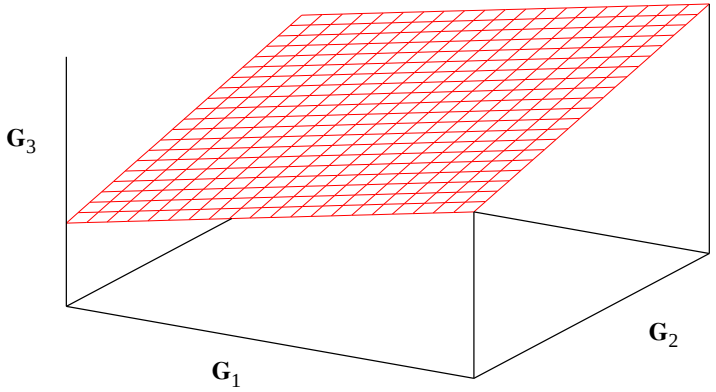
Product basis: PW vs RSG

For NAO $f^a(\boldsymbol{r})$ we need an auxiliary basis $f^a(\boldsymbol{r})f^b(\boldsymbol{r}) = V_\mu^{ab} F^\mu(\boldsymbol{r})$

Product basis: PW vs RSG

For NAO $f^a(\boldsymbol{r})$ we need an auxiliary basis $f^a(\boldsymbol{r})f^b(\boldsymbol{r}) = V^{ab}_{\mu} F^{\mu}(\boldsymbol{r})$

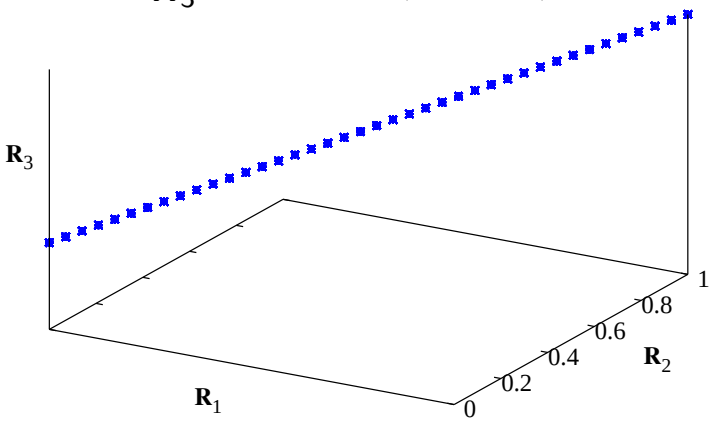
$$V^{\boldsymbol{G}_1\boldsymbol{G}_2}_{\boldsymbol{G}_3} = \delta_{\boldsymbol{G}_1+\boldsymbol{G}_2,\boldsymbol{G}_3}$$



$O(N^2)$

Sparse

$$V^{\boldsymbol{R}_1\boldsymbol{R}_2}_{\boldsymbol{R}_3} = \delta_{\boldsymbol{R}_1,\boldsymbol{R}_3}\delta_{\boldsymbol{R}_2,\boldsymbol{R}_3}$$



$O(N)$

Double sparse

$\Rightarrow$ Potential advantage of localized basis sets

Iterative computation of induced density

- Induced density $\delta n(\omega)$ due to an external perturbation $\delta V^{\mathrm{ext}}(\omega)$

$$\delta n(\omega) = \chi_0(\omega)\delta V^{\mathrm{eff}}(\omega)$$

Iterative computation of induced density

- Induced density $\delta n(\omega)$ due to an external perturbation $\delta V^{\mathrm{ext}}(\omega)$

$$\delta n(\omega) = \chi_0(\omega)\delta V^{\mathrm{eff}}(\omega)$$

- The effective perturbation $\delta V^{\mathrm{eff}}(\omega)$ obeys SLE

$$(\delta - K\chi_0(\omega))\delta V^{\mathrm{eff}}(\omega) = \delta V^{\mathrm{ext}}(\omega)$$

Iterative computation of induced density

- Induced density $\delta n(\omega)$ due to an external perturbation $\delta V^{\mathrm{ext}}(\omega)$

$$\delta n(\omega) = \chi_0(\omega)\delta V^{\mathrm{eff}}(\omega)$$

- The effective perturbation $\delta V^{\mathrm{eff}}(\omega)$ obeys SLE

$$(\delta - K\chi_0(\omega))\delta V^{\mathrm{eff}}(\omega) = \delta V^{\mathrm{ext}}(\omega)$$

- We solve the SLE above with Krylov subspace methods

$$\chi^0_{\mu\nu} z^\nu = V^{ab}_\mu X^n_a X^m_b X^m_c V^{cd}_\nu z^\nu X^n_d$$

- $O(N^3)$ operations or less, where N is number of atoms

Programming solutions to realize TDDFT for SIESTA

- SIESTA RT https://launchpad.net/siesta

Programming solutions to realize TDDFT for SIESTA

- SIESTA RT https://launchpad.net/siesta
- FAST LR https://gforge.inria.fr/frs/?group_id=1179
- MBPT-LCAO LR http://mbpt-domiprod.wikidot.com
- PySCF-NAO LR https://github.com/cfm-mpc/pyscf/tree/nao
- ASE/PySCF-NAO LR cloud-computing solution

Programming solutions to realize TDDFT for SIESTA

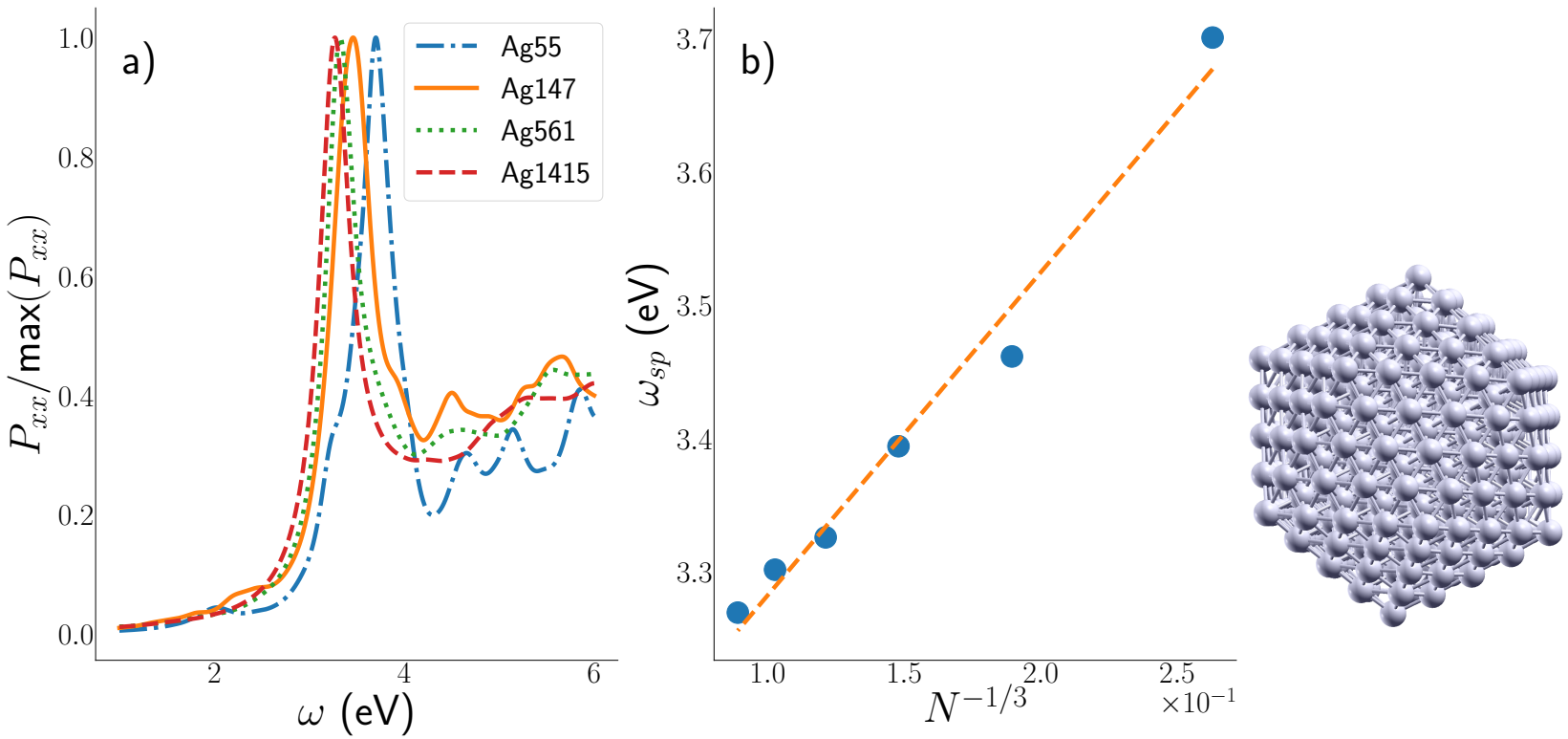
- Berkeley*GW* LR — NAO to PW converter

Programming solutions to realize TDDFT for SIESTA

- PySCF-NAO LR https://github.com/cfm-mpc/pyscf/tree/nao
- ASE/PySCF-NAO LR cloud-computing solution

PySCF-NAO: rather fledged iterative TDDFT

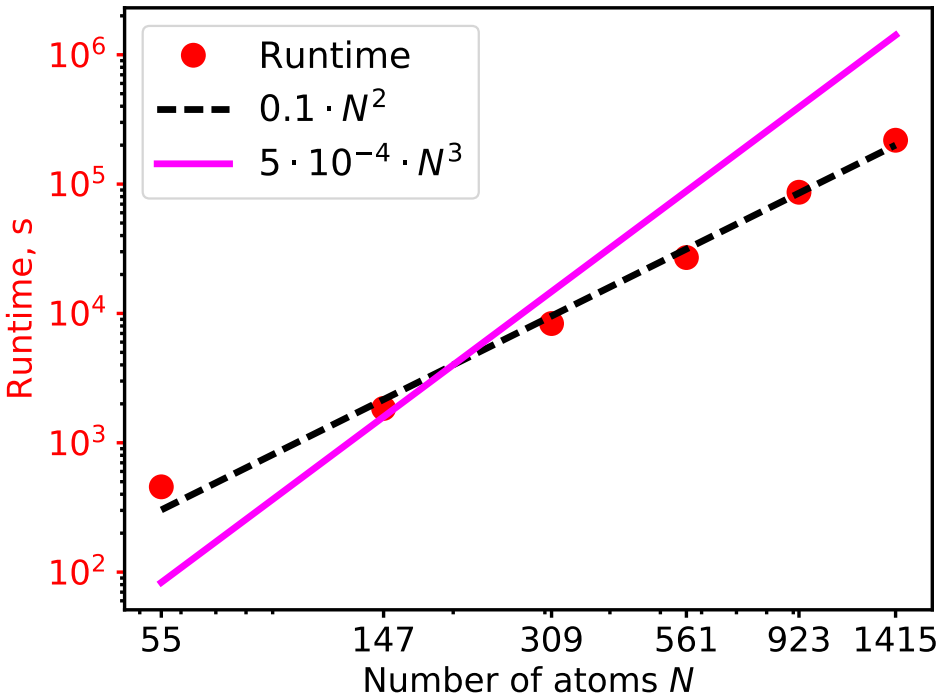
- Open-source, Python-based, Compact code[1]

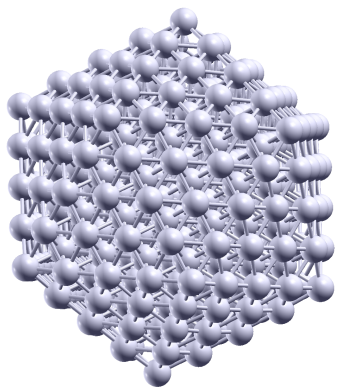


- The plasmonic resonance of silver clusters blue-shifts for smaller sizes

[1] PK, MB, DSP, **submitted**, preprint on ResearchGate

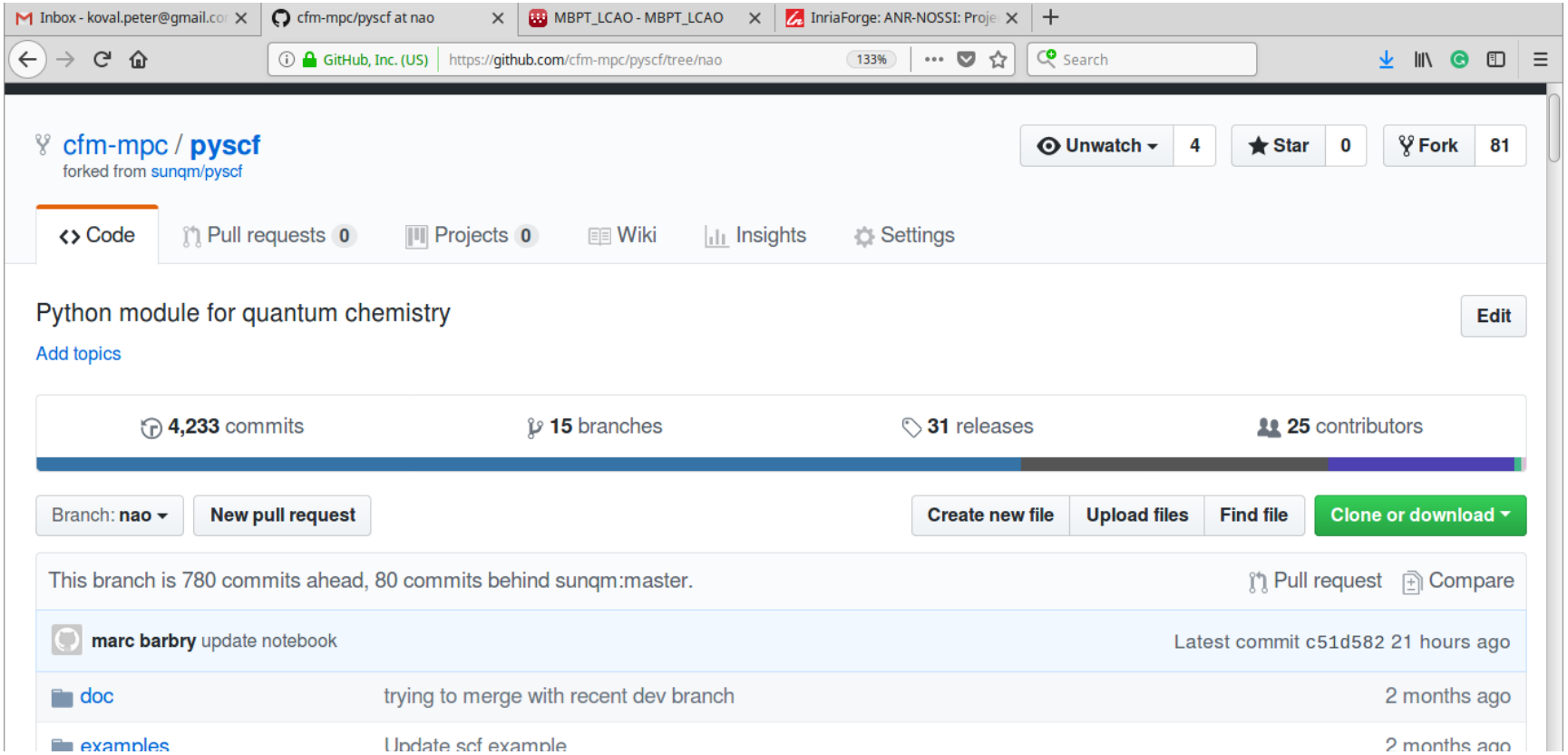
PySCF-NAO: walltime scaling





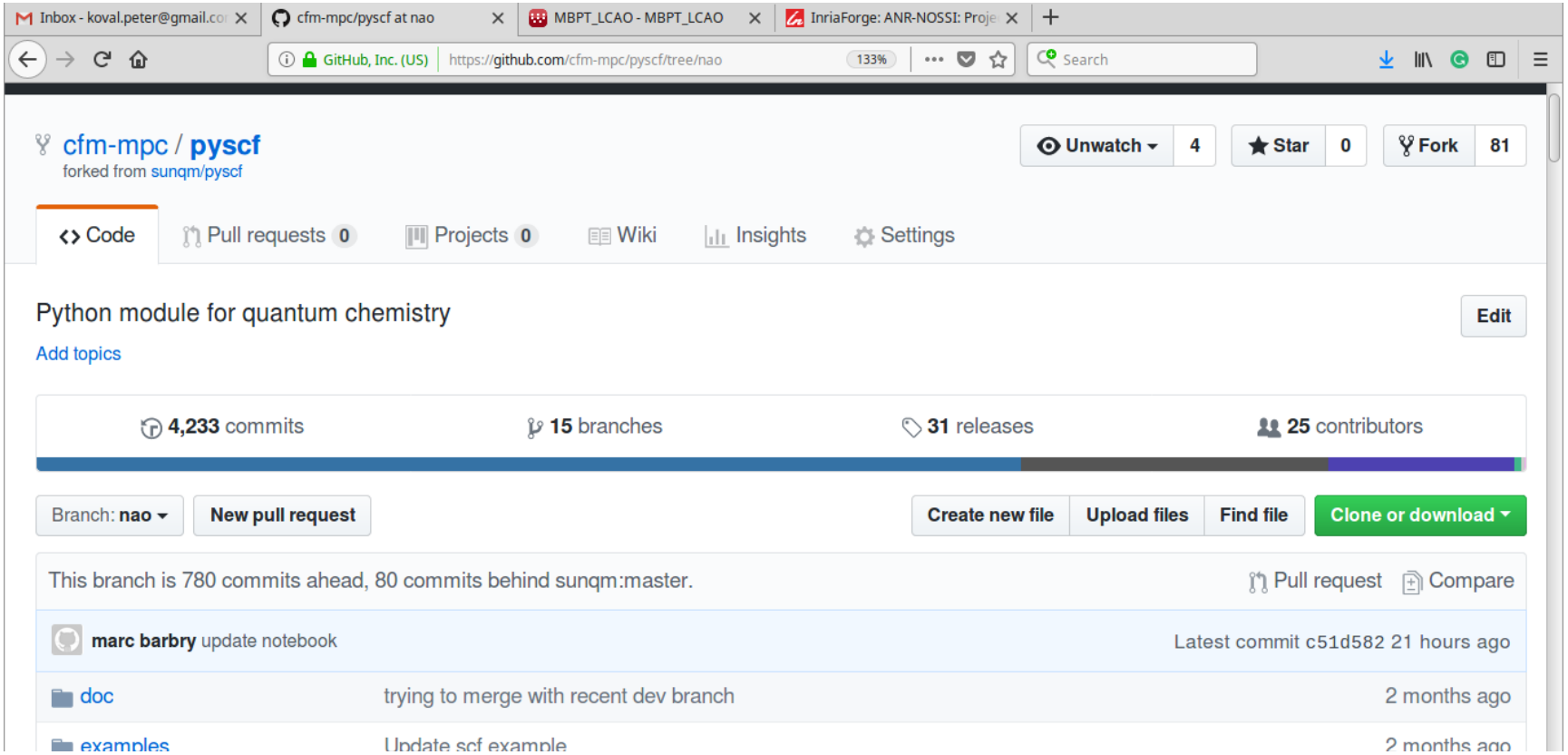
- The runtime on 12 cores of Intel® Xeon® Processor E5-2680 v3
- Largest calculation for Ag_{1415} lasts 28 hours

PySCF-NAO: download



pyscf/wiki/Git-workflow

PySCF-NAO: download



pyscf/wiki/Git-workflow

```
git clone https://github.com/cfm-mpc/pyscf.git
cd pyscf
git checkout -b nao origin/nao
```

PySCF-NAO: installation

README.me at `pyscf/tree/nao/pyscf/lib/nao`

PySCF-NAO: installation

README.me at pyscf/tree/nao/pyscf/lib/nao

```
cd pyscf/lib
cp cmake_arch_config/cmake.arch.inc-nao-gnu cmake.arch.inc
mkdir build
cd build
export FC=gfortran
cmake ..
make export PYTHONPATH=/path/to/pyscf
```

PySCF-NAO: testing

`pyscf/pyscf/nao/tests`

PySCF-NAO: testing

pyscf/pyscf/nao/tests

```
cd pyscf/nao/tests
for f in test_*.py; do python $f; done
```

PySCF-NAO: testing

pyscf/pyscf/nao/tests

```
cd pyscf/nao/tests
for f in test_*.py; do python $f; done
```

```
ImportError: No module named numpy
kovalp:test$ . /opt/intel/intelpython2/bin/activate
(root) kovalp:test$ for f in test_*.py ; do python $f; done
.....
----------------------------------------------------------------------
Ran 5 tests in 1.407s

OK
..
----------------------------------------------------------------------
Ran 2 tests in 1.616s

OK
.
----------------------------------------------------------------------
Ran 1 test in 6.049s

OK
...
----------------------------------------------------------------------
Ran 3 tests in 0.052s
```

Ab-initio Toolkit @ Singularity

siesta™ + pyscf + ASE

↓

Ab-Initio Toolkit

Ab-initio Toolkit @ Singularity

- All popular OS. Desktops, clusters and **cloud platforms**

- Work within common shells, scripts or Jupyter notebook
- Documentation at: mbarbry.website.fr.to/Ab-initioToolkit
- **No cumbersome intallation necessary**[2]

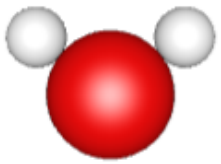
[2] Once the Singularity image is working

Ab-initio Toolkit: Example

We first import the necessary libraries and define the system using ASE

In [1]:
```
# import libraries and set up the molecule geometry

from ase.units import Ry, eV, Ha
from ase.calculators.siesta import Siesta
from ase import Atoms
import numpy as np
import matplotlib.pyplot as plt

H2O = Atoms('H2O', positions = [[-0.757,  0.586,  0.000],
                                [0.757,  0.586,  0.000],
                                [0.0, 0.0, 0.0]],
            cell=[20, 20, 20])

# visualization of the particle
from ase.visualize import view
view(H2O, viewer='x3d')
```

Out[1]:

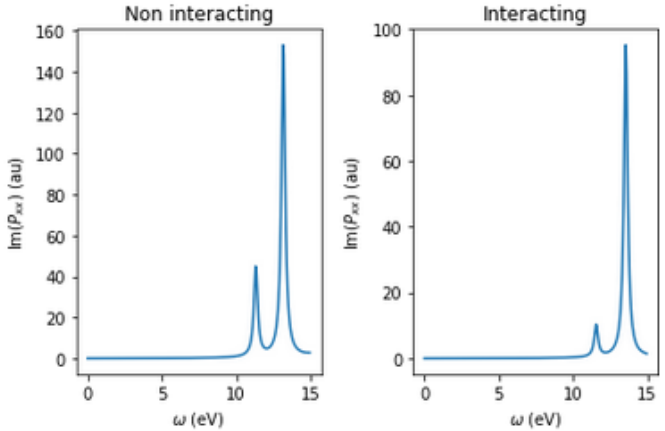


Ab-initio Toolkit: Example

The TDDFT calculations with PySCF-NAO

In [3]:
```
# compute polarizability using pyscf-nao

siesta.pyscf_tddft(label="siesta", jcutoff=7, iter_broadening=0.15/Ha,
        xc_code='LDA,PZ', tol_loc=1e-6, tol_biloc=1e-7, freq = np.arange(0.0, 15.0, 0.05))
```

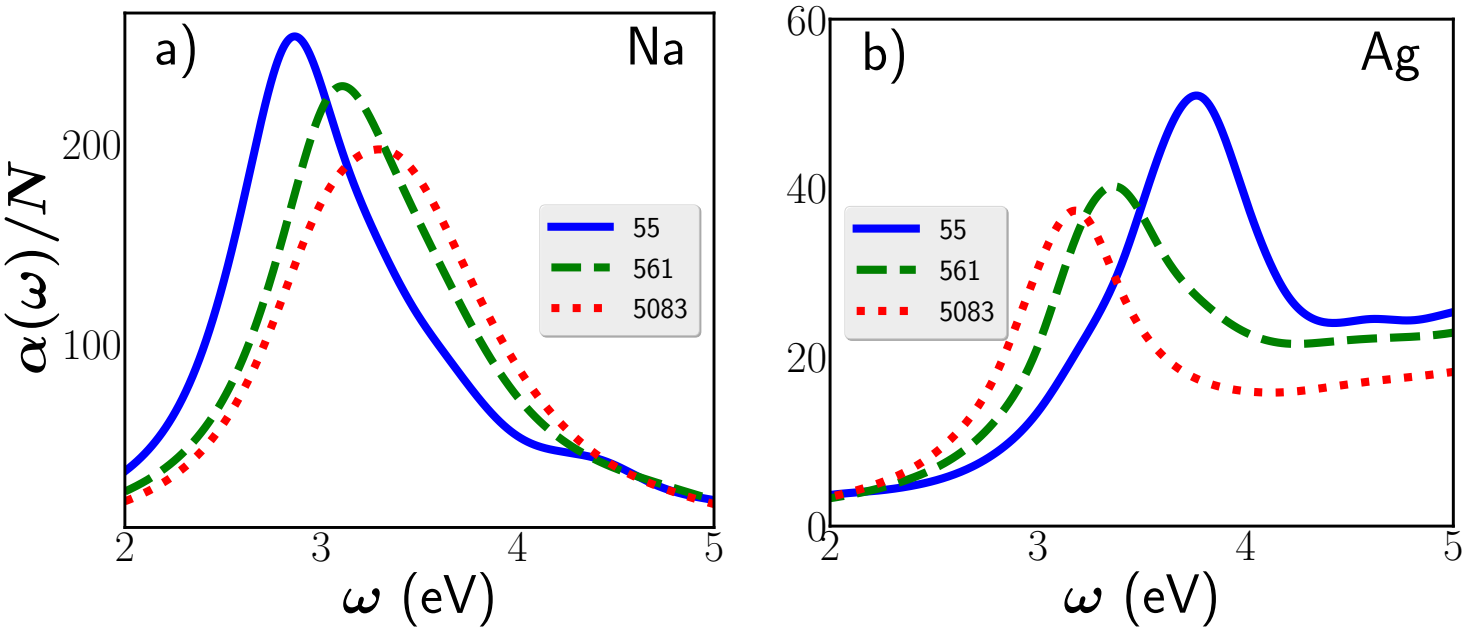
In [4]:
```
# plot polarizability with matplotlib
%matplotlib inline

fig = plt.figure(1)
ax1 = fig.add_subplot(121)
ax2 = fig.add_subplot(122)
ax1.plot(siesta.results["freq range"], siesta.results["polarizability nonin"][:, 0, 0].imag)
ax2.plot(siesta.results["freq range"], siesta.results["polarizability inter"][:, 0, 0].imag)

ax1.set_xlabel(r"$\omega$ (eV)")
ax2.set_xlabel(r"$\omega$ (eV)")

ax1.set_ylabel(r"Im($P_{xx}$) (au)")
ax2.set_ylabel(r"Im($P_{xx}$) (au)")

ax1.set_title(r"Non interacting")
ax2.set_title(r"Interacting")

fig.tight_layout()
```



Size/material dependence: Na vs Ag

Being ab-initio theory TDDFT allows to see straight away

- dependence on the size N — number of atoms
- ...material: Na 3s; Ag $4d^{10}$ 5s

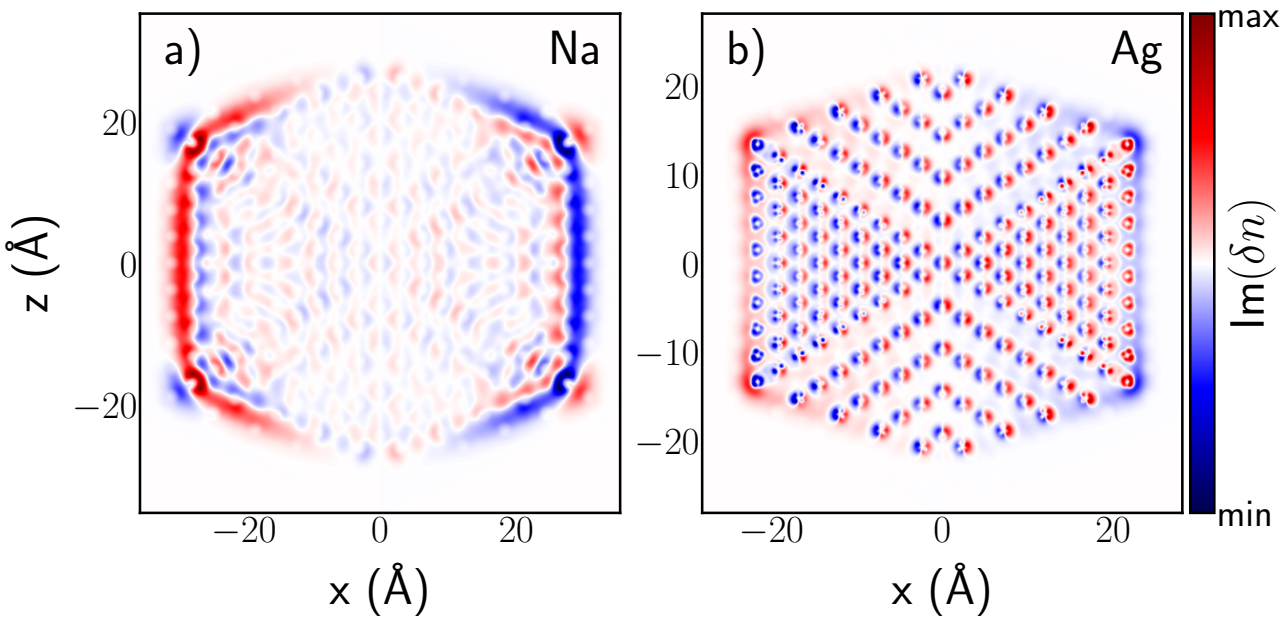


- Why that?

Size/material dependence: reason

Reason for the opposite size dispersion

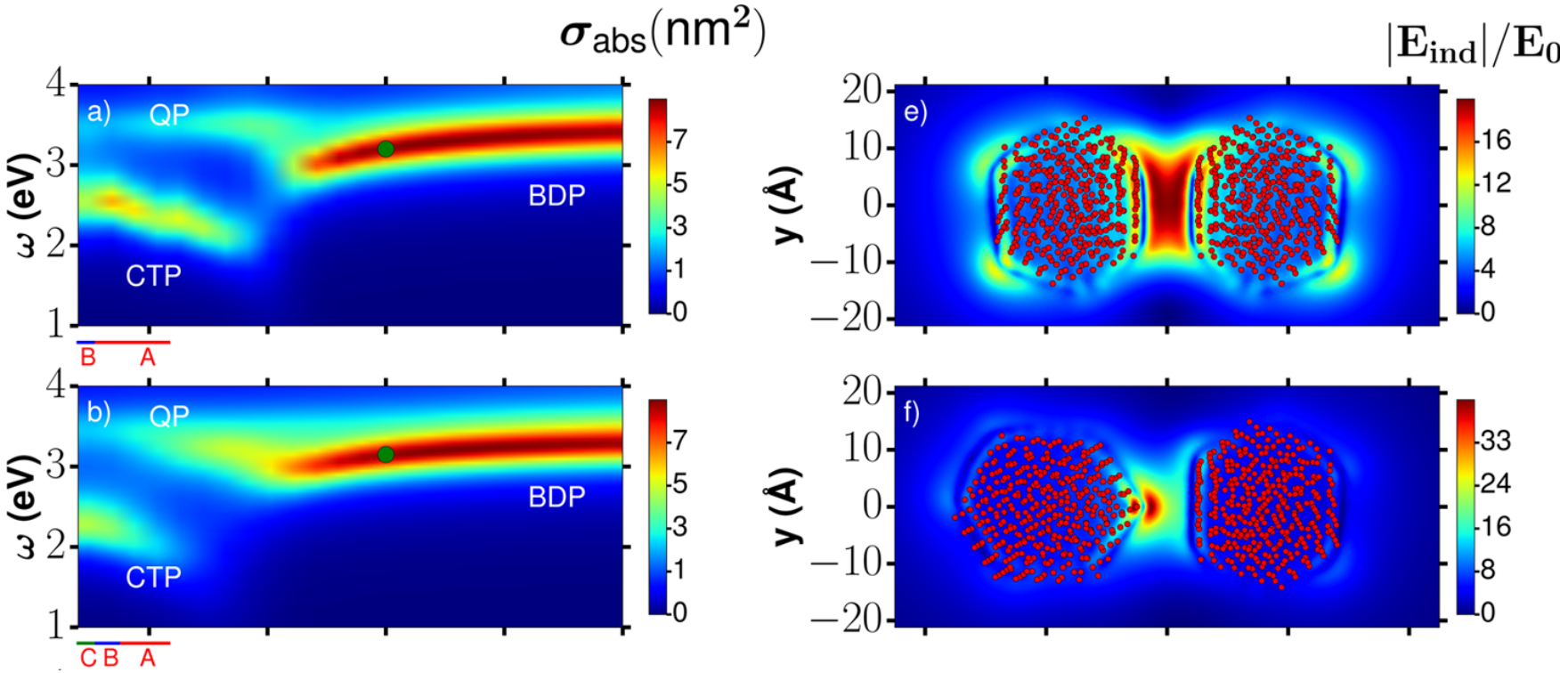
- Without the Coulomb interaction Na and Ag clusters blue shift — quantum confinement
- Without d-electrons Ag dispersion flattens: screening



- Dependence on the shape

Shape dependence for a plasmonic cavity

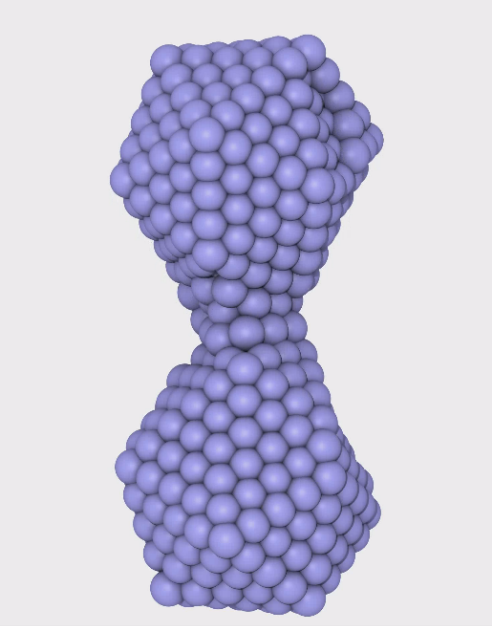
Na_{380} dimer is a model of plasmonic cavity (EM antenna)[3]



[3] Barbry *etal* Nano Lett. 15 (2015) 3410

Relaxations in the Na_{380} cavity

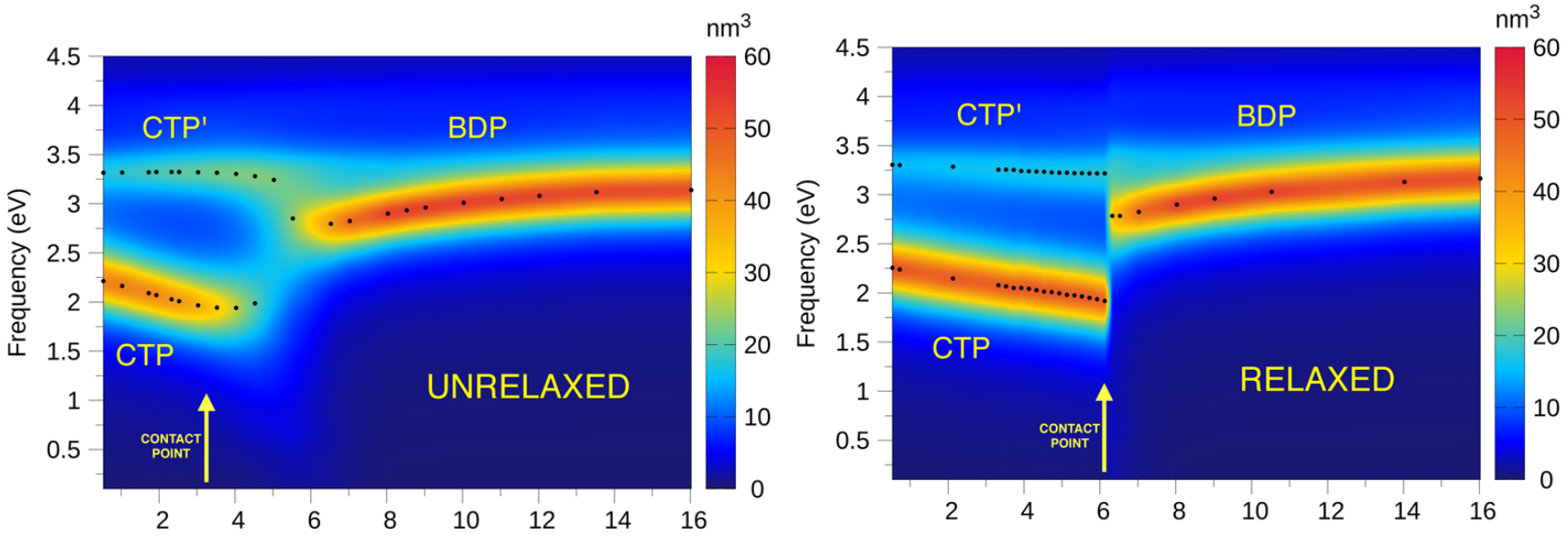
What happens if we allow the atoms to move?



Play movie

Effect of relaxations on polarizability

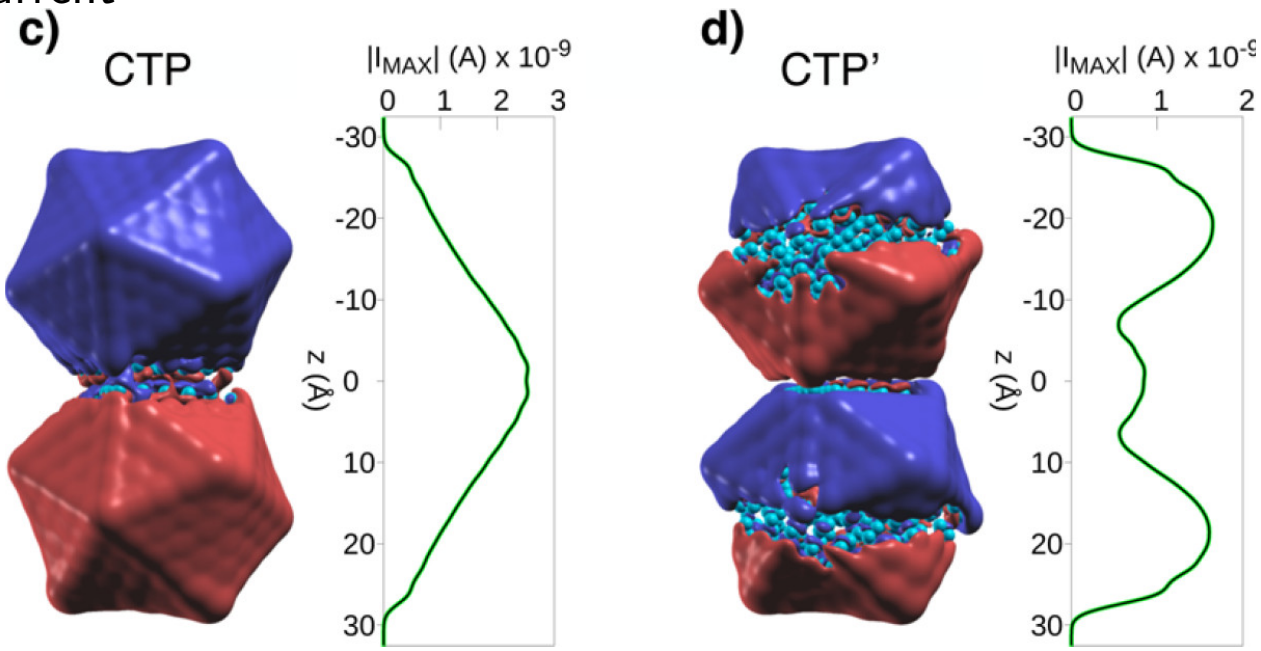
Since we use atomistic theory, we can model inelastic microscopic rearrangements [4]



[4] FM, PK, MB, JA, DSP, *ACS Photonics* **3** (2016) 269

Induced density and current at different frequencies

Because we have the induced density in real space, it is also possible to find the induced current

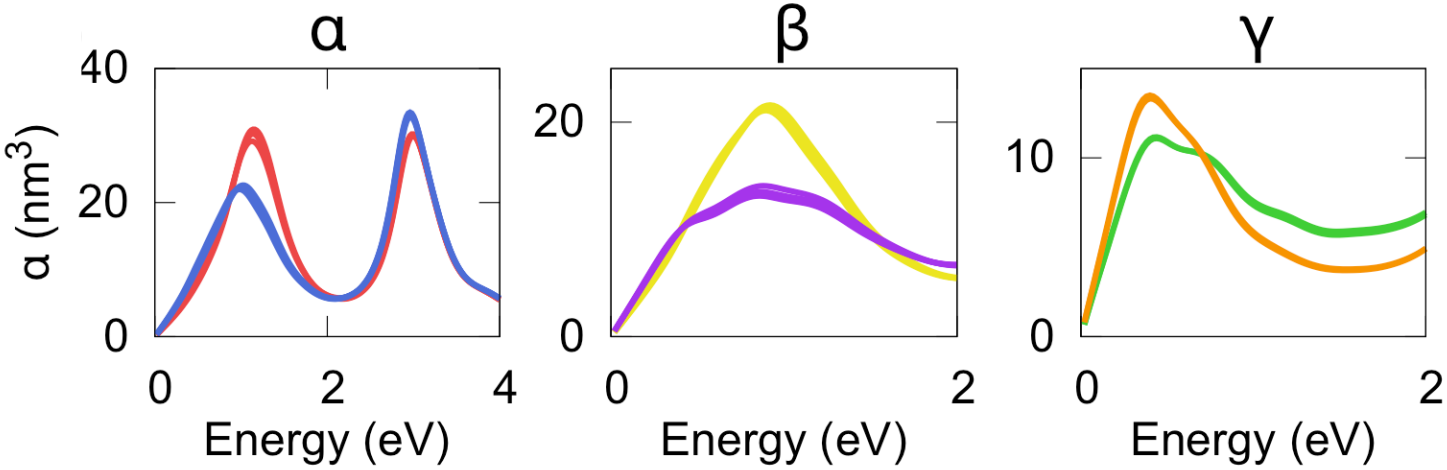


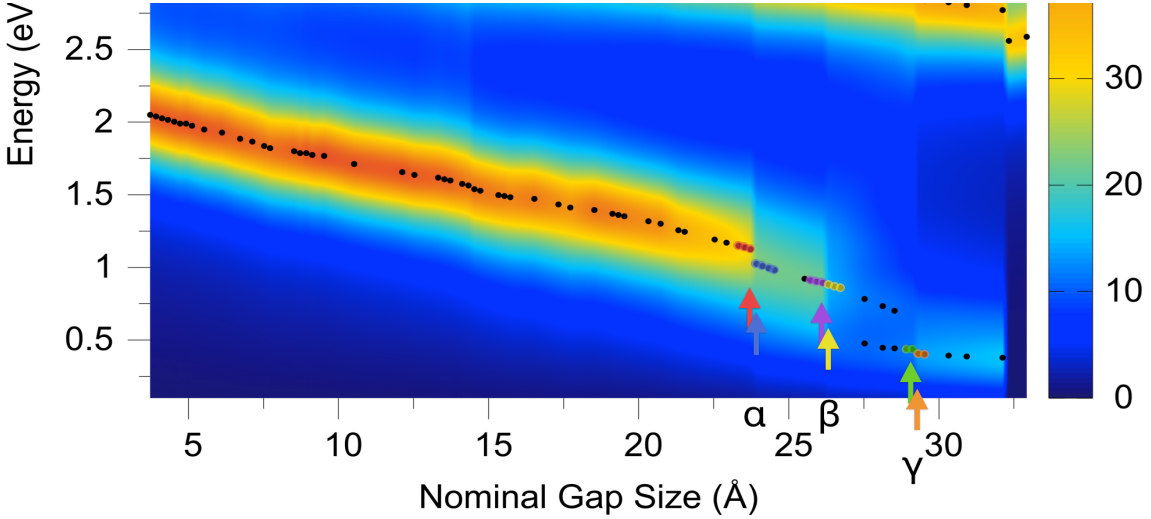
Imaginary part of the induced density and the corresponding modulus of the electron current flowing through the middle of the dimer 2 x Na_{380}[5]

[5] FM, PK, MB, JA, DSP, *ACS Photonics* **3** (2016) 269

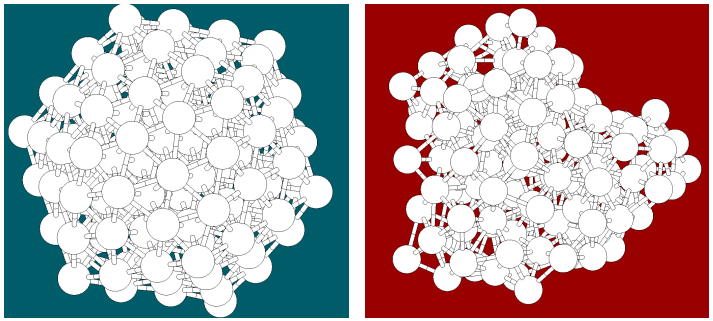
Butterfly effect at large strain

Polarizability $\alpha(\omega)$ as common graphs

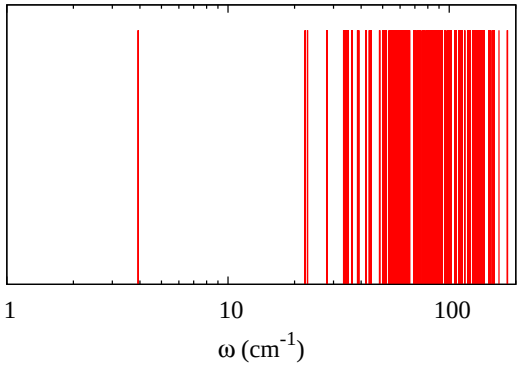




Semi-empirical molecular dynamics (MD) for Ag_{147}



- Atomistic Simulation Environment (ASE) to organize SEMD
- Langevin dynamics for $T_{\text{ion}} = 300$, 600 and 900 K
- Atomistic effective potentials used[6]

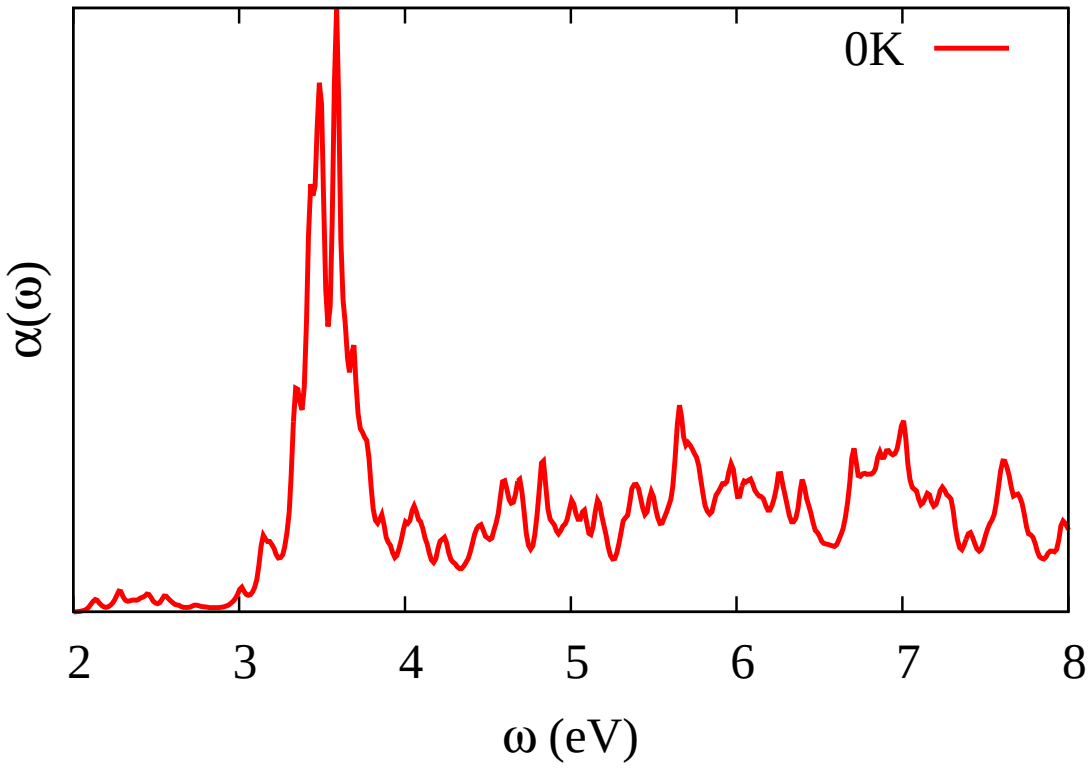


Vibration spectrum of Ag_{147}

- Vibration spectrum was computed with VIBRA utility
- Vibration spectrum was used to justify the MD parameters Δt and T_{total}
- 800 snapshots in TDDFT

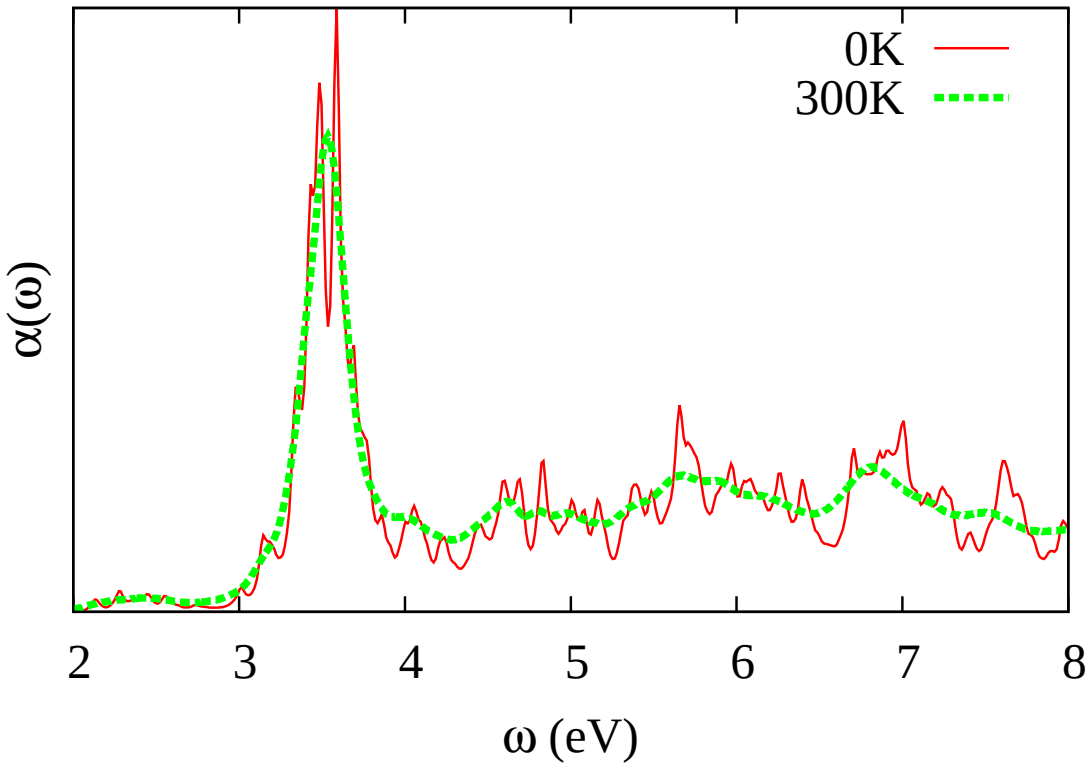
[6] Hale, Wong, Zimmerman and Zhou, *Modelling Simul. Mater. Sci. Eng.* 21 (2013) 045005

SEMD for Ag_{147}



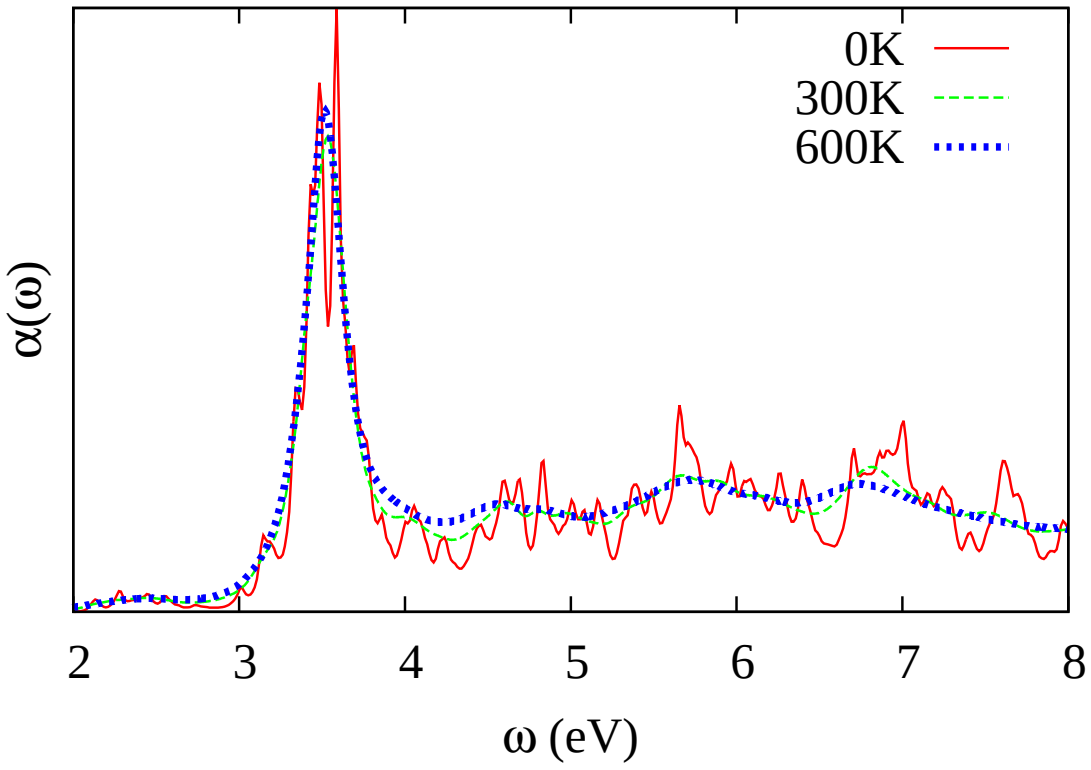
- Icosahedral modes at 0K

SEMD for Ag_{147}



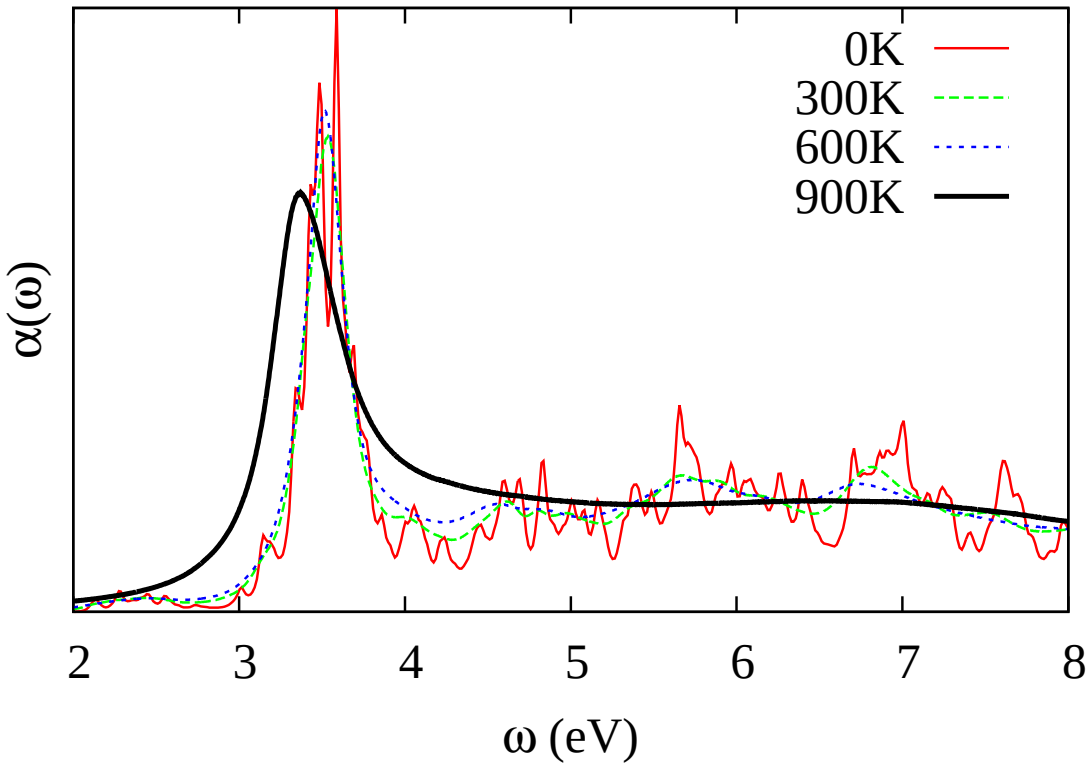
- Icosahedral modes at 0K
- merge at room temp 300K

SEMD for Ag_{147}



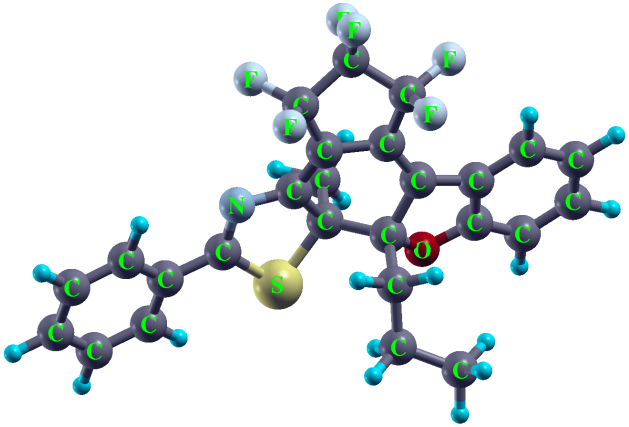
- Icosahedral modes at 0K
- merge at room temp 300K
- and stay unaltered between 300 and 600K

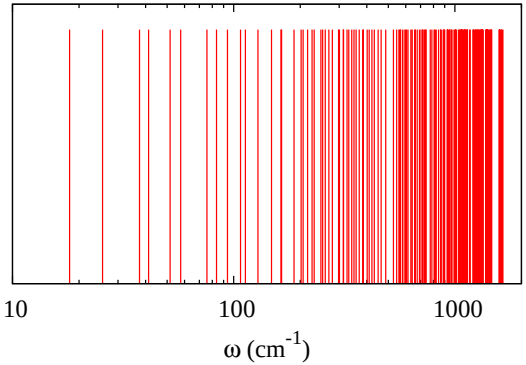
SEMD for Ag_{147}



- Icosahedral modes at 0K
- merge at room temp 300K
- and stay unaltered between 300 and 600K
- Melting affects the optical polarizability

AIMD for a diarylethene as a comparison





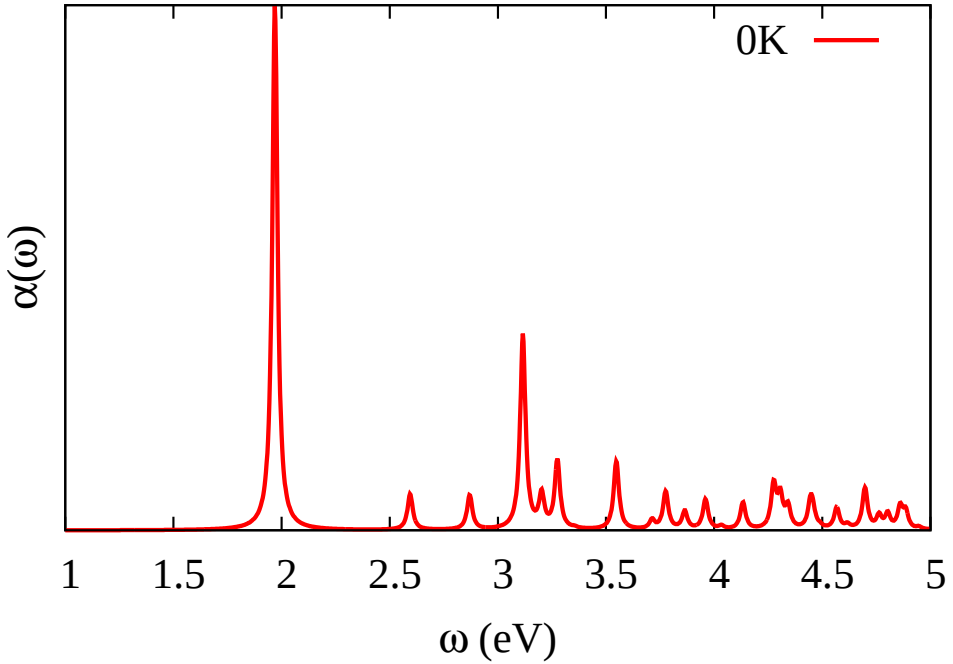
Vibration spectrum

- DFT+MD package SIESTA[7]
- Nosé thermostat $T_{\text{ion}} = 100$, 300 K
- PBE GGA

- PySCF-NAO: optical gap 1.97 eV
- PySCF-GTO: optical gap PBE 2.03 eV, B3LYP 2.33 eV
- VIBRAtion spectrum was used to justify the MD parameters Δt and T_{total}
- 800 snapshots in TDDFT

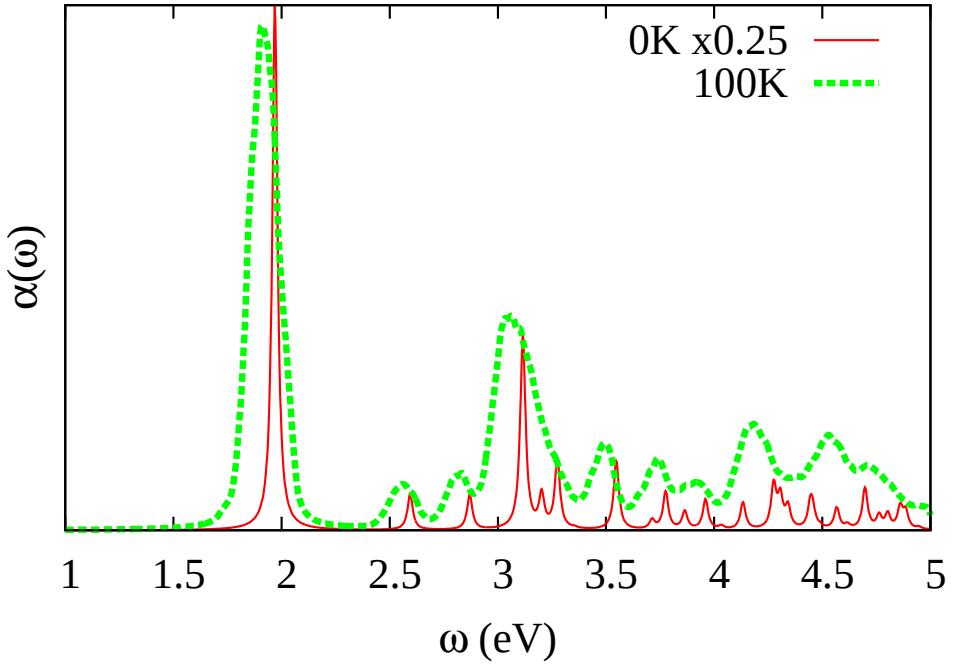
[7] José M Soler *et al* 2002 J. Phys.: Condens. Matter 14 2745

AIMD+TDDFT for a diarylethene compound



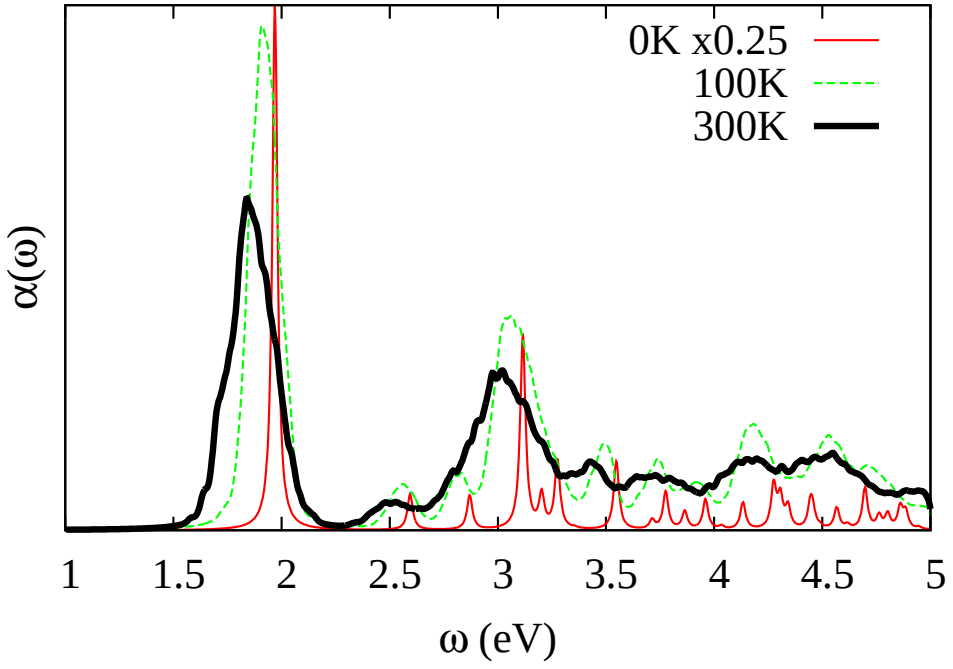
- HOMO-LUMO well-separated

AIMD+TDDFT for a diarylethene compound



- HOMO-LUMO well-separated
- In contrast to Ag_{147}, the resonance frequencies are affected at low $T_{\text{ion}} = 100K$

AIMD+TDDFT for a diarylethene compound



- HOMO-LUMO well-separated
- In contrast to Ag_{147}, the resonance frequencies are affected at low $T_{\text{ion}} = 100K$
- Similarly to Ag_{147}, the resonance frequencies $\omega_i \searrow$ are red-shifted while $T_{\text{ion}} \nearrow$

Valence electron energy loss spectra

Other stimuli is well possible[8]

$$\delta V_{\text{ext}}(\boldsymbol{r},\omega) = \int \mathrm{e}^{\mathrm{i}\omega t}|\boldsymbol{r} - \boldsymbol{r}_{\text{probe}}(t)|^{-1} dt$$

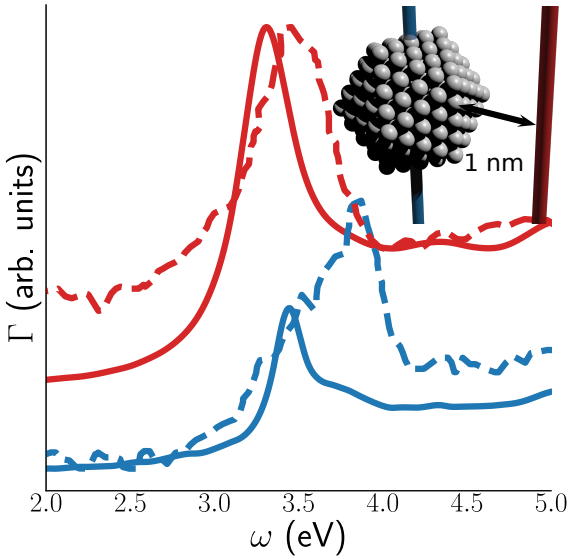


FIG. 4. Experimental [23] (dashed lines) and *ab initio* (full lines) EELS for silver icosahedral cluster. Two distincts trajectories of the electron beam are represented, one crossing the cluster at its center (blue lines) and the other near the surface of the cluster (red lines). The cluster geometry used for the calculations together with the beam trajectories are represented in the figure. The colors of the beams are corresponding to the colors of the lines.

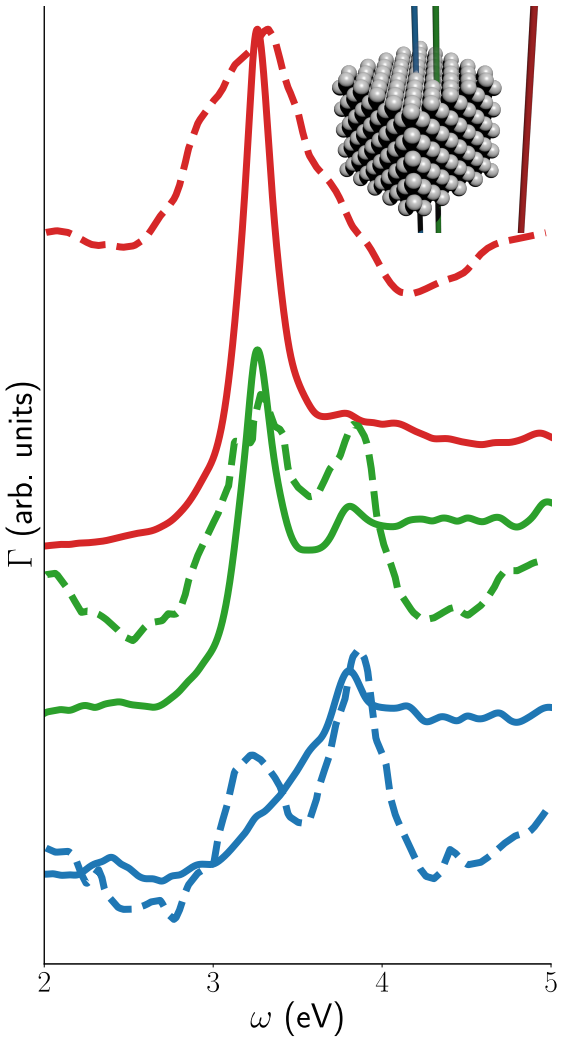


FIG. 5. Experimental [22] (dashed lines) and *ab initio* (full lines) EELS for a silver cube composed of 500 atoms and presenting a fcc lattice. Three distincts trajectories of the electron beam are represented, one crossing the cluster at its cen-

[8] MB, PK, DSP, Ab-initio theory of EELS, in preparation (2018)

Conclusions & Outlook

- Linear response iterative TDDFT is useful for plasmonics
- PySCF-NAO – available online already
- Atomistic approach allows to address a wealth of phenomena

Conclusions & Outlook

- Linear response iterative TDDFT is useful for plasmonics
- PySCF-NAO – available online already
- Atomistic approach allows to address a wealth of phenomena

Underway

- Other methods with Fock-like operators: GW/BSE
- Other observables: EELS and Raman

Conclusions & Outlook

- Linear response iterative TDDFT is useful for plasmonics
- PySCF-NAO – available online already
- Atomistic approach allows to address a wealth of phenomena

Underway

- Other methods with Fock-like operators: GW/BSE
- Other observables: EELS and Raman

Thank you for your kind attention!