Excited state properties of low dimensional systems

Ground state

All the initial information is stored in an input file called `inp;` octopus will always look for this file in the directory we are working in.
In our case:

CalculationMode=gs

Units = eV_Angstrom
radius = 3.5
spacing = 0.25
XYZCoordinates='Methane.xyz'

The calculation mode we want to perform is the groud state:

********** Calculation Mode **********

Input: [CalculationMode = gs]

****

Octopus uses the preloaded pseudopotentials for the atomic species we have defined:

********** Species **********

Reading pseudopotential from file:

'/home/giuseppe/octopus/instalacion//share/octopus/PP/PSF/C.psf'

Calculating atomic pseudo-eigenfunctions for specie C ....

Done.

Info: l = 0 component used as local potential

Reading pseudopotential from file:

'/home/giuseppe/octopus/instalacion//share/octopus/PP/PSF/H.psf'

Calculating atomic pseudo-eigenfunctions for specie H ....

Done.

Info: l = 0 component used as local potential

****

Then we have some information about the simulation box:

********** Grid **********
Simulation Box:

  Type = around nuclei
  Radius [A] =   3.500
  Octopus will run in 3 dimension(s).
  Octopus will treat the system as periodic in 0 dimension(s).

Main mesh:

  Spacing [A] = (0.250, 0.250, 0.250)   volume/point [A^3] = 0.01563
  # inner mesh =   21577
  # total mesh =   51297

  Grid Cutoff [eV] =   601.653

****

The smaller is the spacing, the better is the accuracy of the calculation; obviously it is important to find a trade off between the accuracy and the computational cost of the calculation.
The calculation will be performed in three dimensions and for a non-periodic system.

By default, the theory level used is the DFT:

********** Theory Level **********

Input: [TheoryLevel = dft]

Exchange and correlation:

  Exchange

    Slater exchange (LDA)

    [1] PAM Dirac, Proceedings of the Cambridge Philosophical Society 26, 376 (1930)

    [2] F Bloch, Zeitschrift fuer Physik 57, 545 (1929)

  Correlation

    Perdew & Zunger (Modified) (LDA)

    [1] Perdew and Zunger, Phys. Rev. B 23, 5048 (1981)

[2] Modified to improve the matching between the low and high rs parts

Input: [SICCorrection = sic_none]

****

The approximations used for the the exchange and the correlation terms are also shown.
Then there are other default settings:

****

Input: [RelativisticCorrection = non_relativistic]
Input: [TDGauge = length]
Input: [AbsorbingBoundaries = no_absorbing]
Info: Setting up Hamiltonian.
Info: SCF using real wavefunctions.
Input: [What2Mix = density] (what to mix during SCF cycles)
Input: [TypeOfMixing = broyden]
Input: [EigenSolver = cg]
Input: [Preconditioner = pre_filter]

****

The calculation we did converged in eight iterations:

********** SCF CYCLE ITER # 8 **********

etot = -2.18221621E+02 abs_ev = 2.30E-03 rel_ev = 2.66E-05

abs_dens = 4.61E-06 rel_dens = 5.76E-07

Matrix vector products: 37

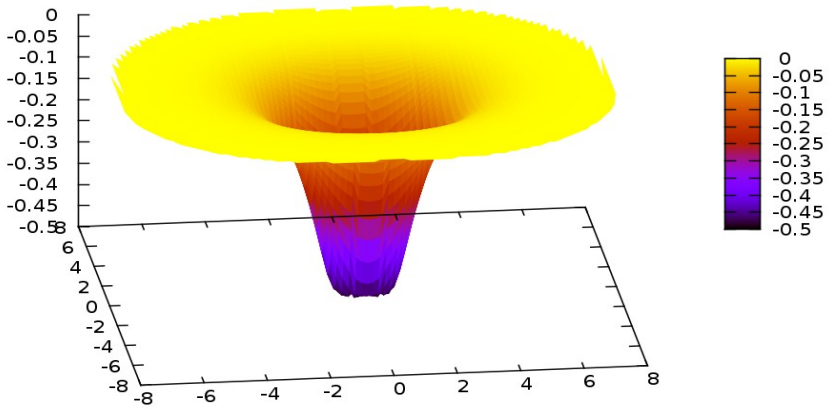
Converged eigenvectors: 4

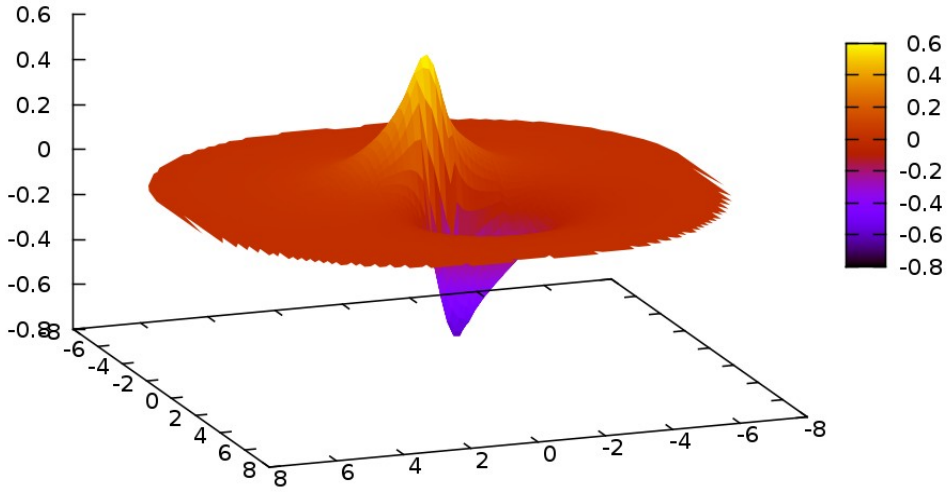
Eigenvalues [eV]

#st	Spin	Eigenvalue	Occupation	Error
1	--	-16.159271	2.000000	(7.9E-07)
2	--	-9.060725	2.000000	(8.0E-07)
3	--	-9.060725	2.000000	(8.0E-07)
4	--	-9.060725	2.000000	(8.0E-07)

Elapsed time for SCF step: 1.30

Info: SCF converged in 8 iterations





Optical spectra from Casida equation

The Casida equation is a (pseudo-)eigenvalue equation written in the basis of particle-hole states. This means that we need both the occupied states -- computed in the ground-state calculation -- as well as the unoccupied states, that we will now obtain.
So, after the ground state calculation, we need the unoccupied state calculation:

CalculationMode=unocc

Units = eV_Angstrom
radius = 4
spacing = 0.175
XYZCoordinates='Methane.xyz'

Output = wfs
OutputHow = plane_z + gnuplot

NumberUnoccStates = 10
EigenSolverMaxIter = 500

Now we can run a “casida” calculation to obtain the absorption spectrum. The file “casida” contains the dipole moments:

$$\langle x\rangle=\langle \Phi_0|x|\Phi_I\rangle \qquad \langle y\rangle=\langle \Phi_0|y|\Phi_I\rangle \qquad \langle z\rangle=\langle \Phi_0|z|\Phi_I\rangle$$

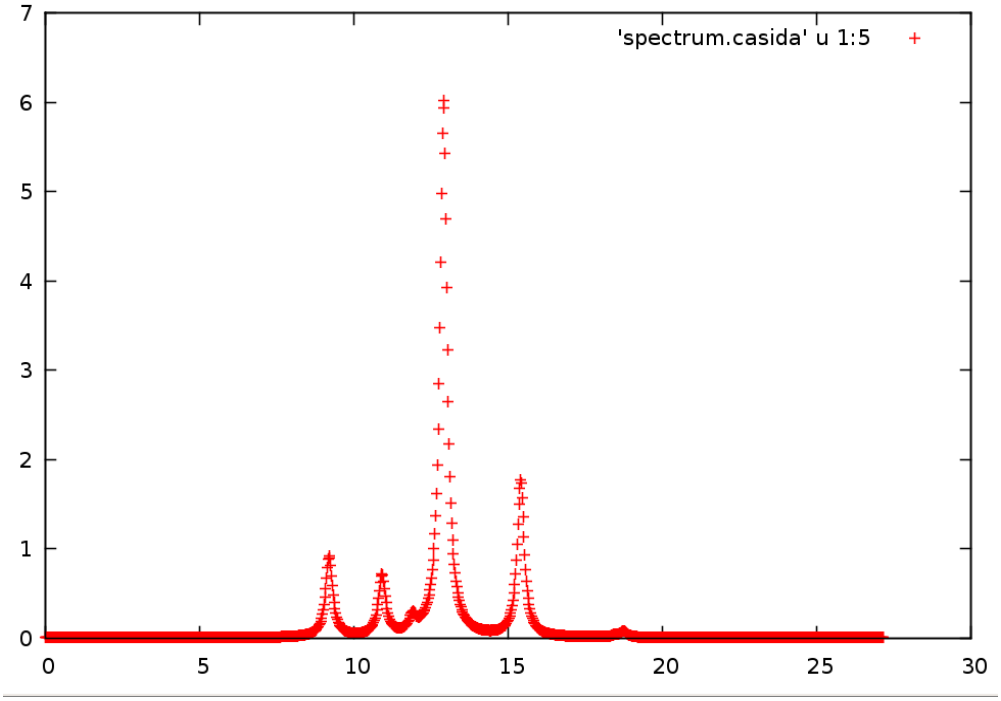
And the oscillator strengths:

$$f_I[x]=2\,\pi\omega_I*|\langle \Phi_0|x|\Phi_I\rangle|^2$$

The optical absorption is given by:

$$S(\omega)=\sum_I f_I*\delta(\omega-\omega_I)$$

Now we can plot the absorption spectrum saved in “spectrum.casida”:



TD calculation

Also in this case, before running the TD calculation, we have to do the ground state calculation. In order to run a time-dependent simulation we have to change the input file:

```
CalculationMode=td

Units = eV_Angstrom

radius = 4
spacing = 0.175
XYZCoordinates='Methane.xyz'

T  = 0.1
dt = 0.002

TDEvolutionMethod = aetrs
TDMaximumIter = T/dt
TDTimeStep = dt
```

As we are calculating the time dependent response of the system under no external perturbation, the electronic system doesn't evolve; also the total energy does not change, nor any other observable change.
TDEvolutionMethod establishes which algorithm will be used to approximate the evolution operator; TDMaximumIter tells the code how may time steps to perform; and TDTimeStep fixes the length of each time step.
A key parameter is, of course, the time step. Before making long-scale calculations, it is worthwhile spending some time choosing the largest time-step possible. This time-step depends crucially on the system under consideration, on the applied perturbation, and on the algorithm chosen to approximate the evolution operator.

Also, there is another input variable that we did not set explicitly, relying on its default value, TDExponentialMethod. Since most propagators rely on algorithms to calculate the action of the exponential of the Hamiltonian, one can specify which algorithm can be used for this purpose.

Now we can use a laser field as an external time-dependent perturbation:

```
CalculationMode=td
Units = eV_Angstrom

radius = 4
spacing = 0.175
XYZCoordinates='Methane.xyz'

T  =1
dt = 0.002

TDEvolutionMethod = aetrs
TDMaximumIter = T/dt
TDTimeStep = dt
```

```
amplitude = 1
omega = 18.0
tau0 = 0.5
t0 = tau0
%TDExternalFields
  electric_field | 1 | 0 | 0 | envelope_cosinusoidal | amplitude | omega | tau0 | t0
%

TDOutput = laser + multipoles
```

Optical spectra from TD calculation

To calculate absorption, we excite the system with a very short electric pulse, and then propagate the time-dependent Kohn-Sham equations for a certain time *T*. The spectrum can then be evaluated from the time-dependent dipole moment.

```
CalculationMode = td
Units = eV_angstrom
fromScratch = yes

radius = 4
spacing = 0.175

XYZCoordinates='Methane.xyz'

TDDeltaStrength = 0.01
TDPolarizationDirection = 1
TDPolarizationEquivAxis = 3

tmax = 10
TDEvolutionMethod = aetrs
TDTimeStep = 0.002
TDMaximumIter = tmax/TDTimeStep
```

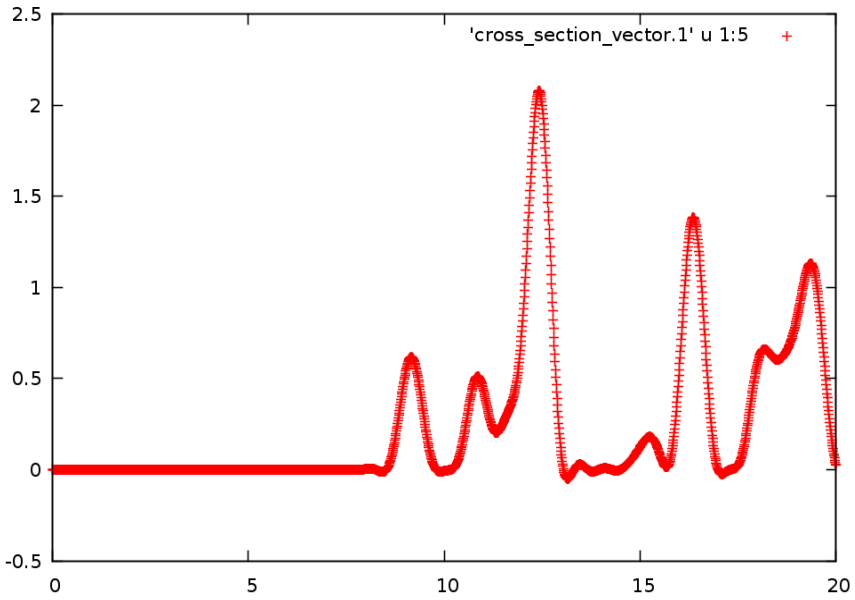
The dynamic polarizability is, in its most general form, a 3x3 tensor. The reason is that we can shine light on the system polarized in any of the three Cartesian axes, and for each of these three cases measure how the dipole of the molecule oscillates along the three Cartesian axes. This usually means that to obtain the full dynamic polarizability of the molecule we usually need to apply 3 different perturbations along x, y, z by setting TDPolarizationDirection to 1, 2, or 3. However, if the molecule has some symmetries, it is in general possible to reduce the total number of calculations from 3 to 2, or even 1.

The methane molecule has T_d symmetry, which means that the response is identical for all directions. This means that we only need one propagation to obtain the whole tensor. This propagation can be performed in any direction we wish. So we could use the input

```
%TDPolarization
 1 | 0 | 0
 0 | 1 | 0
 0 | 0 | 1
%
TDPolarizationDirection = 1
```

```
TDPolarizationEquivAxis = 3
%TDPolarizationWprime
 0 | 0 | 1
%
```

The optical absorption spectrum we calculated is:



As we can see from the figure the spectrum, for low energies, is similar to that one calculated with the Casida's equation.