Practical session I

Octopus I+II

Lecturers: Fernando Nogueira, Miguel Marques

August 29, 2004 ■ 14h30m-18h00m

Introduction

In the first practical session we will start by learning how to use OCTOPUS. We will achieve this through the calculation of some physical properties of small molecules: the total energy, the bond lengths and angles, the electron density, and the occupied and first unoccupied states. These calculations will also serve as the starting point for the TDDFT runs.

OCTOPUS is a pseudopotential real space package aimed at the simulation of the electron-ion dynamics of one, two, and three-dimensional finite systems subject to time-dependent electromagnetic fields. The program is based on time-dependent density-functional theory (TDDFT) in the Kohn-Sham scheme. All quantities are expanded in a regular mesh in real space, and the simulations are performed in real time. The program has been successfully used to calculate linear and non-linear absorption spectra, harmonic spectra, laser induced fragmentation, etc. of a variety of systems. All information about the OCTOPUS package can be found in its homepage, http://www.tddft.org/programs/octopus. The fundamentals of DFT and TDDFT can be found, e.g., in chapters 1 [1] and 4 [2] of vol. 620 of Springer's Lecture Notes in Physics series.

Although the Hohenberg-Kohn theorem states that DFT is exact, the Kohn-Sham method of reducing an interacting many-particle problem to a non-interacting single-particle problem introduces an approximation: the exchange and correlation term. Another approximation very often used in DFT calculations is the pseudopotential approximation.

- **The exchange and correlation functional**

  The Kohn-Sham [3] method of DFT assumes that, for each interacting ground state density $\rho(\vec{r})$, there exists a non-interacting electron system with the same ground state density. The interacting ground state is obtainable through the solution of the Kohn-Sham equations

$$\left[-\frac{1}{2}\nabla^2 + v_{\rm ext}(\vec{r}) + \int \frac{\rho(\vec{r})}{|\vec{r}-\vec{r'}|} d\vec{r'} + v_{\rm xc}\left([\rho];\vec{r}\right)\right]\varphi_i(\vec{r}) = \epsilon_i \varphi_i(\vec{r})\,, \tag{1}$$

  with the electronic density given by

$$\rho(\vec{r}) = \sum_{i=1}^{N} |\varphi_i(\vec{r})|^2\,, \tag{2}$$

  and the total energy of the system is

$$E_{\rm KS}[\rho(\vec{r})] = T_s[\rho(\vec{r})] + \int \rho(\vec{r}) v_{\rm ext}(\vec{r}) d\vec{r} + \frac{1}{2}\int \frac{\rho(\vec{r}_1)\rho(\vec{r}_2)}{|\vec{r}_1-\vec{r}_2|} d\vec{r}_1 d\vec{r}_2 + E_{xc}[\rho(\vec{r})]\,. \tag{3}$$

$\varphi_i(\vec{r})$ are the Kohn-Sham wavefunctions, $T_s[\rho(\vec{r})]$ is the non-interacting kinetic energy, $v_{\rm ext}$ is the external potential and $E_{xc}[\rho(\vec{r})]$ and $v_{xc}\left([\rho];\vec{r}\right)$ are, respectively, the exchange and correlation energy and potential. E_{xc} is an unknown object and includes all the non-trivial many-body effects required to make KS theory exact.

Several approximations to E_{xc} have been proposed. The most used is the local density approximation (LDA). In this approximation $E_{xc}[\rho(\vec{r})]$ is taken to be the exchange and correlation energy of a homogeneous electron gas with density $\rho = \rho(\vec{r})$. Although there exists an exact expression for the exchange energy in this model, the exact value of the correlation energy is known only at very high at very high densities. Ceperley and Alder [4] did a Monte Carlo simulation of the homogeneous electron gas at several densities, thus obtaining the correlation energy at arbitrary densities. Several parameterizations of the correlation energy for any density [5, 6] were then obtained interpolating the Monte Carlo results. One particularly simple parameterization was proposed by Perdew and Zunger [5], and this option may be used by OCTOPUS. You can, of course, choose other xc functionals...

- **Exchange and correlation functionals in TDDFT**

  The time-dependent Kohn-Sham equations are

$$\mathrm{i}\frac{\partial}{\partial t}\varphi_i(\vec{r},t) = \left[-\frac{\nabla^2}{2} + v_{\rm KS}(\vec{r},t)\right]\varphi_i(\vec{r},t)\,. \tag{4}$$

  Here the time-dependent Kohn-Sham potential is the sum of the external time-dependent potential with the Hartree and exchange-correlation potentials

$$v_{\rm KS}(\vec{r},t) = v_{\rm ext}(\vec{r},t) + \int \frac{\rho(\vec{r},t)}{|\vec{r}-\vec{r'}|}d\vec{r'} + v_{xc}\left([\rho];\vec{r},t\right)\,. \tag{5}$$

  As in the time-independent case, the density of the interacting system can be obtained from the time-dependent Kohn-Sham orbitals

$$\rho(\vec{r},t) = \sum_i^{\rm occ} |\varphi_i(\vec{r},t)|^2\,. \tag{6}$$

  The time-dependence of the exchange and correlation potential introduces the need for an approximation beyond the one made in the time-independent case. The simplest method of obtaining a time-dependent xc potential consists in assuming that the potential is the time-independent xc potential evaluated at the time-dependent density, i.e.,

$$v_{\rm xc}\left([\rho];\vec{r},t\right) \simeq v_{\rm xc}\left([\rho];\vec{r}\right)\big|_{\rho\equiv\rho(\vec{r},t)}\,. \tag{7}$$

  This is called the adiabatic approximation. If the time-independent xc potential chosen is the LDA, then we obtain the so-called adiabatic local density approximation (ALDA). This approximation gives remarkably good excitation energies but suffers from the same problems as the LDA, most notably the exponential fall-off of the xc potential. If a strong laser pushes the electrons to regions far from the nucleus, ALDA should not be expected to give an accurate description of the system.

Other options for the time-dependent xc potential are, e.g., the orbital-dependent potentials like the exact exchange functional (EXX) (usually in the Krieger-Li-Iafrate (KLI) approximation).

- **Pseudopotentials**

  The many-electron Schrödinger equation can be greatly simplified if electrons are divided in two groups: valence electrons and inner core electrons. The electrons in the inner shells are strongly bound and do not play a significant role in the chemical binding of atoms, thus forming with the nucleus an inert core. Binding properties are almost completely due to the valence electrons, specially in metals and semiconductors. This separation implies that inner electrons can be ignored, reducing the atom to an inert ionic core that interacts with the valence electrons. This suggests the use of an effective interaction, a **pseudopotential**, that gives an approximation to the potential felt by the valence electrons due to the nucleus and the core electrons. This can significantly reduce the number of electrons that have to be dealt with. Moreover, the pseudo-wavefunctions of these valence electrons are much smoother in the core region than the true valence wavefunctions, thus reducing the computational burden of the calculations.

  Modern pseudopotentials are obtained by inverting the free atom Schrödinger equation for a given reference electronic configuration, and forcing the pseudo-wavefunctions to coincide with the true valence wavefunctions beyond a certain cutoff distance. The pseudo-wavefunctions are also forced to have the same norm as the true valence wavefunctions, and the energy pseudo-eigenvalues are matched to the true valence eigenvalues. Different methods of obtaining a pseudo-eigenfunction that satisfies all these requirements lead to different non-local, angular momentum dependent pseudopotentials. Some widely used pseudopotentials are the Troullier and Martins [7] potentials, the Hamann [8] potentials, the Vanderbilt [9] potentials and the Hartwigsen-Goedecker-Hutter [10] potentials. The default potentials used by OCTOPUS are of the Troullier and Martins type, although you can also opt for the HGH potentials.

Besides these usual DFT approximations, the use of OCTOPUS implies another approximation, which stems from the particular calculation method used.

- **The choice of the mesh**

  In OCTOPUS the Kohn-Sham equations are discretized on a finite mesh in real space. This means that derivatives are evaluated using a finite differences method (OCTOPUS uses a 9-point rule for the laplacian, although other rules can be chosen) and integrals are weighted sums over the grid points. The mesh is uniform, but future versions of OCTOPUS will have other kinds of grids. The grid points play the role of the 'basis functions' of the methods where an expansion of the Kohn-Sham wavefunctions in some base is employed (e.g., plane-waves or Gaussian wavefunctions). But, contrary to, e.g., plane-waves, this 'basis set' is not variational. Care must be taken to ensure that the calculations are converged with respect to these

basis fuctions. This implies performing calculations with different grid spacings. As the combination of pseudopotentials and the LDA will always introduce a ∼0.1 eV error, the convergence with grid spacing is good enough when energy differences vary by less than 0.1 eV.

Objectives

- Get a working knowledge of OCTOPUS input file;
- Calculate ground state for some small molecules (Na_2, C_2H_2, and SiH_4) using different exchange and correlation functionals (LDA, GGA, and EXX);
- Study the convergence of the calculations;
- Plot the density, potential, and wavefunctions;
- Calculate some unoccupied states;
- Start linear response calculation.

Tasks

1. **Convergence of the total energy with respect to the grid spacing**

   Move to the directory ∼`/molecules/`. There you will find three input files (`c2h2.inp`, `na2.inp`, and `sih4.inp`), for three different systems. Pick the file corresponding to the system assigned to you and take a look at the input variables. Information on the input variables is available by typing `info octopus` and also on the file `/usr/share/doc/octopus-1.4a/octopus.ps.gz`. You will notice that the input file is rather short. This means that all the missing input variables will be given some reasonable default values. In particular, you will not find in the input files any information neither on the exchange and correlation functional to be used nor on the pseudopotentials. If you were assigned an exchange and correlation functional different from the default Perdew-Zunger LDA you will have to add some variables to the input file. Find out which variables to add and type them in. Fill in the following table.

Molecule being studied:	
Geometry	
Box type	
Box size	
Grid spacing	
XC functional	
Pseudopotentials	

Do a test run of OCTOPUS by copying `xxx.inp` to `inp` and typing `octopus > xxx.out`, where `xxx` represents the system you will be studying. This should take a couple of minutes. At the end of the run you will have the following new files: `out.oct`, `xxx.out`, and `static/info`. The first one of these files is the list of ALL the input variables of OCTOPUS and the values they assumed in this run, the second file is the information about every stage of the run, and the third file is a summary of the final results. Take a good look at these files to familiarize yourself with them.

You are now going to determine a reasonable grid spacing that ensures that your calculations are converged. To help you, a sample script is provided: look for `xxx_convergence.sh` and edit it to suit your needs. Run the script and plot the total energy and some eigenvalues as a function of the grid spacing (use, e.g., `xmgrace` or `gnuplot`). Check also that the simulation box you use is of the convenient type and size.

Can you answer the following questions?

Q1. What was the criterion for convergence of the SCF cycle?

Q2. How many SCF cycles were needed to meet this criterion?

Q3. Which grid spacing assures an error for the total energy smaller than 10 meV?

Q4. What can you say about the convergence of the eigenvalues differences?

Q5. Does your curve of the total energy as a function of the grid spacing look monotonous? Why?

2. **Plot the density, potential and wavefunctions for the converged run**

Replace the grid spacing in the input file for the one you have just found. The input file as it is now will not output the wavefunctions or the density. In order to plot them you will have to add some variables to the input file. Try adding

```
OutputKSPotential = yes
OutputDensity = yes
OutputWfs = yes
OutputWfsSqMod = yes
OutputGeometry = yes
OutputDX= yes
```

and run octopus with the new input file. In the `static` directory you will now have some files with extension `.dx`. These are input files for IBM's *Data Explorer* visualization program. OCTOPUS provides an applet to ease the plotting of these files with *Data Explorer*: simply copy the `mf.cfg` and `mf.net` files in `/usr/share/octopus/util` to your current directory and type `dx`. Then, select *Run Visual Programs...* from the menu and open `mf.net`. Take a deep breath and start playing with the program. The applet instructions will appear as a separate window on your terminal if you select *Application Comment* from the *Help* menu (NOT the *Help* button on the initial window). Plot the density, the wavefunctions squared, and the Kohn-Sham potential. You can superimpose these plots on a ball and stick model of the molecule....

3. **Calculate some unoccupied states**

   Until now, you have just computed occupied states. This means that the number of Kohn-Sham wavefunctions used in the SCF cycle was exactly half the number of valence electrons. You could have added some extra wavefunctions to the static ground-state calculation by including, e.g.,

   ```
   ExtraStates = 3
   ```

   in the input file. This would instruct OCTOPUS to use three additional wavefunctions in the SCF cycle. Of course these wavefunctions would correspond to unoccupied states in the cases we are studying, but could be important if you had some degeneracy or quasi-degeneracy at the HOMO level. In that case you could also fix the occupation numbers and distribute the top-lying electrons among the degenerate orbitals. For example, to compute the ground-state of the carbon atom you could resort to

   ```
   ExtraStates = 2
   %Occupations
     2 | 2/3 | 2/3 | 2/3
   %
   ```

   These extra wavefunctions and eigenvalues would of course be updated during the SCF cycle and would contribute to the ground-state density.

   However, you can also, given some previously calculated ground-state density, determine the unoccupied states **without** including them in the SCF cycle. These states are calculated in a single run using a fixed density. This method of obtaining the unoccupied states is more adequate than the previous one.

   With these calculations one can get a first approximation to the excitation energies of the system by simply taking the differences between the eigenvalues. Although not entirely justifiable, this approach does give a rough idea of the excitation spectrum.

   Now, you will calculate some unoccupied states of the system. This is done by replacing

   ```
   CalculationMode = gs_start
   ```

   with

   ```
   CalculationMode = unocc_start
   ```

   in the input file. Upon running OCTOPUS with this input you will find the list of eigenvalues in `static/eigenvalues`. Please note that convergence is very slow. If you use the default values

   ```
   UnoccMaximumIter = 200
   UnoccConv = 1e-4
   ```

chances are that your unoccupied eigenvalues will be very poorly converged at the end of the run. Either change the default values or simply restart two or three times the unoccupied states calculation taking care of changing from

```
CalculationMode = unocc_start
```

to

```
CalculationMode = unocc
```

(this will restart the unoccupied states calculation from the point where the previous calculation ended). To perform a simple calculation of the excitation spectrum, just include

```
LinEigenvalues = yes
```

in the input file and run `oct-excite`. This external tool supplied with OCTOPUS will simply calculate all the eigenvalue differences and put them, sorted, in the file `linear/eps-diff`. There is another external tool, `oct-broad`, that will compute a spectrum by broadening the excitation energy peaks and place it in `linear/spectrum.eps-diff`. The parameters of the broadening are controlled by `LinBroadening`, `LinMinEnergy`, `LinMaxEnergy`, and `LinEnergyStep`. The first of these parameters is the width of the Lorentzian used to broaden the spectrum; the spectrum is calculated between `LinMinEnergy` and `LinMaxEnergy` in a regularly spaced grid with steps of `LinEnergyStep`. The output file will contain 5 columns: energy, dipole strength in the x direction, dipole strength in the y direction, dipole strength in the z direction, and arithmetical average of the three dipole strengths. Plot the 5th column vs. the 1st one.

Q6. How do these values compare to the experimental ones?

4. **Determine the optimal time-step for the propagation of the TDKS equations**

While the time-independent Kohn-Sham equations constitute a boundary value problem, the solution of the time-dependent Kohn-Sham equations is an initial value problem. At $t = t_0$ the system is in some initial state described by the Kohn-Sham orbitals $\varphi_i(\vec{r}, t_0)$. In most cases the initial state will be the ground state of the system (i.e., $\varphi_i(\vec{r}, t_0)$ will be the solution of the ground-state Kohn-Sham equations). To solve the TDKS equations amounts to to propagate this initial state until some final time, t_f.

The time-dependent Kohn-Sham equations can be rewritten in the integral form

$$\varphi_i(\vec{r}, t_f) = \hat{U}(t_f, t_0)\varphi_i(\vec{r}, t_0)\,, \tag{8}$$

where the time-evolution operator, $\hat{U}$, is defined by

$$\hat{U}(t', t) = \hat{T} \exp\left[-\mathrm{i}\int_t^{t'} \mathrm{d}\tau\ \hat{H}_{\mathrm{KS}}(\tau)\right]. \tag{9}$$

Note that $\hat{H}_{\rm KS}$ is explicitly time-dependent due to the Hartree and xc potentials. It is therefore important to retain the time-ordering product $\hat{T}$, in the definition of the operator $\hat{U}$. The exponential in expression (9) is clearly too complex to be applied directly, and needs to be approximated in some suitable manner.

OCTOPUS provides several approximation methods: the split-operator, Suzuki-Trotter (a higher order split-operator method), Enforced Time-Reversal Symmetry, Approximated Enforced Time-Reversal Symmetry, Exponential Midpoint Rule, and Magnus Expansion methods. The method used in the propagation is chosen putting in the input file the keyword

```
TDEvolutionMethod = xxx
```

where `xxx` is either `split`, `suzuki-trotter`, `etrs`, `aetrs`, `exp_mid`, or `magnus`. Some of these methods require also a numerical approximation to the exponential of the Hamiltonian. The methods supplied by OCTOPUS with the input variable

```
TDExponentialMethod = xxx
```

are the split-operator, Suzuki-Trotter, Lanczos, Chebyshev methods and a standard n-th order expansion of the exponential. The value of the above variable is either `split`, `suzuki-trotter`, `lanczos`, `chebyshev`, or `standard`.

Other OCTOPUS variables that are relevant for the propagation are, obviously, the duration of the propagation (`TDMaximumIter`) and the time-step for the propagation (`TDTimeStep`). If we had infinite speed computers the latter could be chosen as small as numerical accuracy would allow. But, in the real world, we have to compromise and use a large time-step. However, the discretization of the evolution operator imposes an upper limit on the allowed time-step. This upper limit depends on the grid-spacing, becoming smaller for finer grids. You will now determine the relation between the maximum time-step that allows for an energy-conserving propagation and the grid-spacing. To do this, you will have to perform several TDDFT runs for each grid spacing with different time-steps, and monitor the evolution of the system and the variation of the total energy. You will notice that an energy-conserving well-behaved propagation will occur only below a certain $\Delta t_{\rm max}$.

To start a simple TDDFT run with OCTOPUS you replace

```
CalculationMode = gs_start
```

with

```
CalculationMode = td_start
```

in the input file and include, e.g., the following lines

```
TDEvolutionMethod = etrs
TDExponentialMethod = standard
TDMaximumIter = 100
TDTimeStep = 1.0
```

This will propagate the ground state that you had previously calculated in the same external time-indepedent potential (there is no perturbation added to the ground state, the external potential is just the nuclear potential!). Please note that the above time step is huge and will lead to an enormously silly total energy just after the first step. Take care to redirect the output to a file (type `octopus > td.out`). Besides this file, which should be self-explanatory, you will find a new directory, `td.general`, and a `multipoles` file in it. Disregard this file for now.

Plot the $\Delta t_{\max}$ vs. Δx curve (Δx is the grid spacing).

Q7. Can you guess the analytical dependence of $\Delta t_{\max}$ on Δx?

5. **Start a TDDFT run**

   Having found all the optimal parameters for a TD simulation, you will now start a real TDDFT run. Replace the grid spacing and time-step in the input file with your ideal parameters, put `TDMaximumIter = 10000`, add the following lines to the input file

```
TDDeltaStrength = 0.05
%TDPolarization
 1 | 0 | 0
%
```

   start OCTOPUS and go out and have a beer! You will analyze the spectrum tomorrow.

Comments

Practical session III

Octopus III

Lecturers: Fernando Nogueira, Miguel Marques

August 30, 2004 ■ 14h30m-16h00m

Objectives

- Obtain an optical absorption spectrum from a TDDFT run;
- Obtain the excitation energies from linear response theory;
- Start a TDDFT run with an external strong perturbation: a laser pulse.

Tasks

1. **Obtain the dipole strength function from yesterday's TDDFT run**

   Your run from yesterday should have finished by now. You should have a big `td.general/multipoles` file. This file will be used by yet another external tool, `oct-sf`, to generate the dipole strength function of the system. Running this tool will produce a `spectrum` file with two columns: the energy and the dipole strength function. (As your perturbation is in one direction only, you get just one strength function.) Units are those specified in the input file. You can use the following parameters in the input file to control the calculation of the spectrum: `SpecMinEnergy`, `SpecMaxEnergy`, and `SpecEnergyStep`. The meaning of these parameters is the same as the similarly named parameters for the linear response spectrum calculation.

   Q1. Compare this spectrum with the one obtained yesterday and with the experimental results.

2. **Obtain excitation energies from linear response theory**

   When a molecule is subject to a weak external time-dependent potential, it may not be necessary to solve the full time-dependent Kohn-Sham equations to determine the behavior of the system: perturbation theory allows us to calculate easily the linear change of the density, and that in turn is all that we need to get, e.g., the optical absorption spectrum. This approximation is known as **linear-response theory**. However, it is quite difficult to solve numerically the linear density response equation [2], which is an integral equation that requires the non-interacting response function as an input. To obtain this quantity it is usually necessary to perform a summation over all states, both occupied and unoccupied. Such summations are sometimes slowly convergent and require the inclusion of many unoccupied states. There are however frameworks that circumvent the full solution of the response equation. Two of these approximations are implemented in OCTOPUS: the single-pole approximation of Petersilka [11] and the full matrix solution of Casida [12, 13].

Using the unoccupied states calculated yesterday you will now generate spectra within these approximations. Just proceed as yesterday (in the calculation of unoccupied states task), but include

```
LinPetersilka = yes
LinCasida = yes
```

in the input file.

Q2. Compare these spectra with the previous ones.

3. **Start a TDDFT run with a strong laser pulse**

   Now you will start a run in which your system will be perturbed by a strong laser pulse. The analysis of the output will be done in the next session. The key parameters are described in section 6.10 of the OCTOPUS manual. A good starting point is obtained replacing the previous TD input variables with

```
%TDLasers
 1 | 0 | 0 | 1 | 2.5 | 1 | 1 | 2.5
%
TDMaximumIter = 10000
TDTimeStep = <whatever time step you found out before>
TDOutputAcceleration = yes
TDOutputLaser = yes
TDOutputElEnergy = yes
TDOutputOccAnalysis = yes
```

   Depending on the laser intensity, part of the electronic charge may wish to abandon the simulation box, resulting in ionization of the system. There is no exact way of taking care of this. One solution is to consider a boundary region which absorbs the density that arrives there (otherwise the electrons would bounce back from the border of the simulation region, which is unacceptable). For this purpose, you can add the following variables:

```
AbsorbingBoundaries = mask
AbWidth = 1.0
```

   This setting applies a mask function to the density at each time-step. This mask is 1.0 in the inner region (i.e., it does nothing there), and decays smoothly to zero in the boundary region.

   Please ensure that the different groups working an a particular system use different laser strengths and frequencies. This will allow us to better explore the results.

Comments

Practical session V

Octopus IV

Lecturers: Fernando Nogueira, Miguel Marques

September 1, 2004 ■ 14h30m-16h00m

Objectives

- Analyze the output resulting from a strong perturbation of the system.

Tasks

1. **Analyze variation of total energy**

   Your TDDFT run should have created the files `acceleration`, `laser` and `el_energy` in the `td.general` directory. Using the data in the `el_energy`, plot the total energy of the system as a function of time and compare it to the plot obtained by the other groups working on the same system. How different are they?

2. **Plot the laser pulse**

   Using the `laser` plot the laser pulse and its Fourier transform. Give another look at the energy plot you did before...

3. **Calculate the emission spectrum**

   An approximation to the emission spectrum may be obtained from the power spectrum of the acceleration signal. For that purpose, you need to run the auxiliary program `oct-hs-acc`. You already know the variables regarding the characteristics of the spectrum (`SpecMaxEnergy`, etc). Once that you obtain it, you may see where the emission peaks.

Comments

Bibliography

[1] J. P. Perdew and S. Kurth, in *A Primer in Density Functional Theory*, Vol. 620 of *Lecture Notes in Physics*, edited by C. Fiolhais, F. Nogueira, and M. Marques (Springer, Berlin, 2003), Chap. 1, pp. 1–55.

[2] M. A. L. Marques and E. K. U. Gross, in *A Primer in Density Functional Theory*, Vol. 620 of *Lecture Notes in Physics*, edited by C. Fiolhais, F. Nogueira, and M. Marques (Springer, Berlin, 2003), Chap. 4, pp. 144–184.

[3] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[4] D. M. Ceperley and B. J. Alder, Phys. Rev. Lett. **45**, 566 (1980).

[5] J. P. Perdew and A. Zunger, Phys. Rev. B **23**, 5048 (1981).

[6] J. P. Perdew and Y. Wang, Phys. Rev. B **45**, 13244 (1992).

[7] N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991).

[8] D. R. Hamann, Phys. Rev. B **40**, 2980 (1989).

[9] D. Vanderbilt, Phys. Rev. B **41**, 7892 (1990).

[10] C. Hartwigsen, S. Goedecker, and J. Hutter, Phys. Rev. B **58**, 3641 (1998).

[11] M. Petersilka, U. J. Gossmann, and E. K. U. Gross, Phys. Rev. Lett. **76**, 1212 (1996).

[12] M. Casida, in *Recent developments and applications in density functional theory*, edited by J. M. Seminario (Elsevier, Amsterdam, 1996), p. 391.

[13] R. Bauernschmitt and R. Ahlrichs, Chem. Phys. Lett. **256**, 454 (1996).