octopus: a first-principles tool for excited electron-ion dynamics. ⋆

Miguel A. L. Marques [a], Alberto Castro [b,a,c], George F. Bertsch [c]
and Angel Rubio [a]

[a]*Departamento de Física de Materiales, Facultad de Químicas, Universidad del País Vasco, Centro Mixto CSIC-UPV/EHU and Donostia International Physics Center (DIPC), 20080 San Sebastián, Spain*

[b]*Departamento de Física Teórica, Universidad de Valladolid, E-47011 Valladolid, Spain*

[c]*Physics Department and Institute for Nuclear Theory, University of Washington, Seattle WA 98195 USA*

Abstract

We present a computer package aimed at the simulation of the electron-ion dynamics of finite systems, both in one and three dimensions, under the influence of time-dependent electromagnetic fields. The electronic degrees of freedom are treated quantum mechanically within the time-dependent Kohn-Sham formalism, while the ions are handled classically. All quantities are expanded in a regular mesh in real space, and the simulations are performed in real time. Although not optimized for that purpose, the program is also able to obtain static properties like ground-state geometries, or static polarizabilities. The method employed proved quite reliable and general, and has been successfully used to calculate linear and non-linear absorption spectra, harmonic spectra, laser induced fragmentation, etc. of a variety of systems, from small clusters to medium sized quantum dots.

Key words: Electronic structure, time-dependent, density-functional theory, non-linear optics, response functions
PACS: 33.20.-t, 78.67.-n, 82.53.-k

PROGRAM SUMMARY

Title of program: octopus

Catalogue identifier:

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland, or from the web page http://www.tddft.org/programs/octopus/.

⋆ octopus@tddft.org

Program summary URL:

Licensing provisions: This program is distributed under the GNU General Public License v2.0 (see `http://www.gnu.org/` for details).

Computers for which the program has been designed and others on which it has been operable: any computer architecture, running any flavor of UNIX.

Operating systems under which the program has been tested: GNU/Linux, AIX, Tru64 Unix, Irix.

Programming languages used: Fortran 90, C, bison, m4, sh.

Libraries required:

- BLAS (`http://www.netlib.org/blas/`),
- LAPACK (`http://www.netlib.org/lapack/`),
- FFTW (`http://www.fftw.org/`),
- GSL (`http://www.gnu.org/software/gsl/`)
- MPI (`http://www-unix.mcs.anl.gov/mpi/`)

All of these are available under open-source licencies.

Memory used to execute with typical data: 50-1000Mb.

CPU time required to execute test cases: For the benzene example (section 7.1), the ground-state calculation took around 15 minutes (in a single processor), while each of the time-evolutions took around 2 days (using 8 processors). These numbers refer to an IBM SP3.

No. of bits in a word: 32 or 64.

Has the code been vectorized or parallelized?: Parallelized with MPI.

Number of bytes in distributed program including test data, etc.: a lot.

Distribution format: tar gzipped.

Keywords: electronic-structure, linear response, non-linear response, non-adiabatic dynamics, density-functional theory, time-dependent density-functional theory, local-density approximation, generalized-gradient approximation, real-space methods.

Nature of physical problem
Interaction of quantum finite systems with classical electromagnetic fields. The electronic degrees of freedom are described within the Kohn-Sham form of the time-dependent density functional theory, while nuclei are treated as classical point particles.

Method of solution
The electronic wave-functions are discretized in real space using an uniform mesh, and are propagated in real time using nearly unitary propagation schemes. Pseudopotentials are normaly used to described the electron-ion interaction, although model interactions can also be employed. The electromagnetic fields are treated classically either in the length or the velocity gauge.

Restrictions on the complexity of the problem: The present version only handles finite systems and classical nuclei. In the near future the code will handle periodic structures.

Unusual features of the program: The program can be run in either one or three dimensions (we plan to support two dimensions and several cathegories of periodical systems in future versions). `octopus` makes use of a very sophisticated, but user-friendly input system.

LONG WRITE-UP

1 Introdution

In the past years, density functional theory (DFT) has become the method of choice for electronic structure calculations[1] due to its simplicity (it does not rely on the complete knowledge of the N-electron wave function, but only on the electronic density) and high accuracy (present DFT results have surpassed in quality those from standard *ab-initio* quantum chemistry techniques like Hartree-Fock, configuration-interaction, etc.). It is thus natural that a swarm of computer codes solving the Kohn-Sham equations are available to the community. [1] . Using a variety of approaches – e.g. plane waves, localized basis sets, real-space discretizations – these programs are able to calculate ground-state energies, equilibrium geometries, phase transitions, phonon modes, thermodynamic properties, etc. [1–4]. Nevertheless, electronic excited-state properties cannot be studied within the traditional ground-state DFT formulation. Several extensions to the basic formalism have been

[1] There are two rather complete and efficient pseudopotential based codes, which are freely available (as in free speech), ABINIT (`http://www.abinit.org`) and PWSCF (`http://www.sissa.it/cm/PWcodes`). Several other pseudopotential programs coexist in the market like VASP (`http://cms.mpi.univie.ac.at/vasp`), PARATEC (`http://www.nersc.gov/projects/paratec`), MIKA (`http://www.csc.fi/physics/mika`), DACAPO (`http://www.fysik.dtu.dk/CAMP/dacapo.html`), DMOL (`http://www.ccwp.ac.uk/ccwp/cg_dmol.html`), SIESTA (`http://www.uam.es/siesta`), etc.. In the quantum-chemistry realm we can find several all-electron codes like APW, WIEN, ADF, GAUSSIAN, GAMESS, etc.

proposed to overcome this defficiency, the most successful of which has been time-dependent density functional theory (TDDFT). TDDFT allows one to calculate excited-state energies and spectra involving neutral particle-hole excitations of a many-body system. This technique has been quite successful in describing optical spectra of finite and extended systems [5,6].

The advent of strong, femtosecond laser sources – now equipping modern laboratories – enabled the study of new and exciting non-linear phenomena in condensed matter systems[7]. These lasers can yield information about elementary electronic processes occurring on timescales from pico- to femtoseconds, which are nowadays becoming relevant for technological applications. As an example of exciting new problems currently under investigation by the laser community we can mention laser control of chemical reactivity[8], high harmonic generation (that can in turn be used to generate soft X-ray laser pulses)[9] and above-threshold dissociation[10]. Obviously, all these phenomena are beyond the scope of perturbative methods, such as linear response theory. In this context, TDDFT becomes a valuable tool: Fully solving the time-dependent Kohn-Sham equations is a numerically feasible task – even for moderate to large size systems, and opens the door to the simulation of these and other processes.

`octopus` is intended as a general-purpose tool for the study of time-dependent processes, both in the linear and non-linear regimes. It solves the time dependent Kohn-Sham equations in real time, by discretizing all quantities in real space[4,11–13]. This methodology has already proved to be a very powerful technique, yet its greatest attractiveness resides in its conceptual simplicity. Furthermore, `octopus` makes use of norm-conserving pseudopotentials to describe the electron-ion interaction, which are indispensable to reduce the computational burden. Electromagnetic fields are treated classically and the nuclei are considered to be classical point particles. The electronic wave functions are propagated with a nearly unitary scheme. The present version can be used to study both one and three dimensional finite systems. Modern workstations should be able to handle molecules of up to a hundred atoms, which is enough, for example, to treat chromophores in proteins (using QM/MM techniques [2]). Some physical properties obtainable with the program are:

- Ground state properties, such as ground-state energies, equilibrium geometries, molecular dynamics, static polarizabilities, etc.
- Linear optical absorption spectra.
- Harmonic spectra generated by strong (femto- to picosecond) laser fields.
- Ionization probabilities.
- Laser induced reactivity, such as photodissociation paths, isomerization.

Finally, we would like to briefly mention the history of `octopus` . It is based on a

[2] In this case, part of the system is treated within quantum mechanics whereas the rest of the protein and solvent are treated as classical fields [14].

program written by K. Yabana and one of the authors (GFB) to calculate the linear optical response of clusters[12]. This successful application together with the enormous potential of the time-evolution approach prompted us to completely rewrite the code. The language used was mainly Fortran 90, which enable us to reuse most of the previous source lines, and also to employ some modern programming techniques. The rationale was to implement a modular structure such that new features could be easily added. For example, we have already included nuclear dynamics, external time-dependent electromagnetic fields, several exchange-correlation functionals, non-collinear spin, etc. We have also parallelized the time-evolution using MPI.

In the near future we intend to extend the code to one dimensional periodic systems (polymers), [3] current-density functional theory (CDFT), molecular transport and to multi-component TDDFT, so that the nuclei may be treated quantum mechanically. We expect the code to keep growing, as we gather contributions from different users. Some Articles have already been published based on results obtained by `octopus` (see Ref. [16]).

The code is freely available (as in free beer, but also as in free speech) from `http://www.tddft.org/programs/octopus/`. The reader is also referred to the web site to obtain more information, updated documentation, examples, new versions, etc. Since pseudopotentials are essencial to `octopus` , we also provide a web interface to J.L. Martins' pseudopotential generation code[17]. Some of those pseudopotentials have been tested in practical applications [4] . Other pseudopotentials already implemented in the code are the relativistic separable potentials of Hartwigsen, Goedecker and Hutter[18].

2 Theoretical background

In quantum mechanics the full Hamiltonian for a system of electrons and nuclei is:

$$\hat{H} = \hat{T}_{\rm e} + \hat{T}_{\rm n} + \hat{W} + \hat{V}_{nn} + \hat{V}_{en}\,, \tag{1}$$

where $\hat{T}_{\rm e}$ and $\hat{T}_{\rm n}$ are respectively the kinetic energy of the electrons and of the nuclei; $\hat{W}$ is the electron-electron interaction, while $\hat{V}_{nn}$ represents the nuclear-nuclear repulsion; finally, $\hat{V}_{en}$ accounts for the electron-nuclear attraction. Trying to directly solve this Hamiltonian is absolutely tantalizing. To make the problem manageable, we perform several (controlled) approximations. First, the quantum nature of the

[3] The formulation and implementation of the time-evolution technique in real space has been already proved reliable for describing the linear response of periodic systems[15].

[4] However this does not guarrantee the transferability to other systems. The user should go through a thorough check of the pseudopotential for his/her particular application.

nuclei is neglected: we will be dealing with point nuclei, obeying Newton's equations of motion. Then, we will make use of the pseudopotential approximation, which allows to explicitely treat much less electrons, and to avoid the fast oscillations produced by the core electrons[2]. Finally, TDDFT is employed to manage the electron-electron interaction (see Refs. [5,6] for details on the basic foundations of the TDDFT theory and for applications to finite and extended systems). The set of equations to be solved for the electron and ion dynamics are: [5]

$$-i\frac{\partial}{\partial t}\psi_i(\mathbf{r},t) = \Big[-\frac{\nabla^2}{2} + v_{\rm ext}(\mathbf{r},t) + v_{\rm Hartree}(n;\mathbf{r},t) + v_{\rm xc}(n;\mathbf{r},t) + v_{\rm laser}(\mathbf{r},t)\Big]\,\psi_i(\mathbf{r},t), \tag{2}$$

$$n(\mathbf{r},t) = \sum_{i=1}^{N} |\psi_i(\mathbf{r},t)|^2 ,$$

$$m_\alpha \frac{d^2\mathbf{R}_\alpha}{d^2t} = F_\alpha(\mathbf{R},t)\,, \tag{3}$$

where $v_{\rm laser}(\mathbf{r},t)$ describes the classical time-dependent external electromagnetic field acting on our system, $\{\psi_i\}$ $(i=1,\ldots,N)$ are the occupied Kohn-Sham orbitals – to simplify the notation, the spin coordinate is embodied in the i index – and n is the one-electron density. The potentials entering the first set of equations, the time-dependent Kohn-Sham equations, are defined as:

$$v_{\rm ext}(\mathbf{r}) = \sum_\alpha \left[v^\alpha_{\rm local}(\mathbf{r}-\mathbf{R}_\alpha) + \hat{v}^\alpha_{\rm non-local}(\mathbf{r}-\mathbf{R}_\alpha)\right] , \tag{4}$$

$v^\alpha_{\rm local}$ is the local part of the pseudopotential of the nucleus α at the position $\mathbf{R}_\alpha$, and $v^\alpha_{\rm non-local}$ denotes the non-local part of the pseudopotential which is used in the original Kleinman-Bylander form[19]:

$$\hat{v}_{\rm non-local} = \sum_{lm} \frac{|\phi_{lm}(\mathbf{r})\delta V_l(\mathbf{r})\rangle\,\langle\phi_{lm}(\mathbf{r})\delta V_l(\mathbf{r})|}{\langle\phi_{lm}(\mathbf{r})|\,\delta V_l(\mathbf{r})\,|\phi_{lm}(\mathbf{r})\rangle}\,, \tag{5}$$

where ϕ_{lm} are the atomic pseudo-wave functions of angular momentum l and m, and δV_l is the difference between the pseudopotential that generates ϕ_{lm}, and $v_{\rm local}$. The projectors may be calculated explicitly from a set of functions ϕ_{lm} and δV_l obtained previously – as in the procedure of Troullier and Martins [17], or parametrized with a small set of optimized parameters – as done by Hartwigsen, Goedecker and Hutter [18]. Both options are implemented in the program. In the latter case it is possible to include more than one projector per angular component[20]. The potential generated by a solvent or by a distribution of classical charges can be added to Eq. (4)

[5] Unless otherwise stated,atomic units throughout the paper, i.e. $e^2 = \hbar = m_e = 1$.

(this possibility is useful, e.g., to include the effects of the protein backbone when studying chromophores) [21]. Model potentials like harmonic oscillators (to study, e.g. quantum dots), or soft-Coulomb potentials (for 1D calculations) can be defined in the input file.

The Hartree potential, $v_{\rm Hartree}$ is written as:

$$v_{\rm Hartree}(\mathbf{r},t) = \int d^3r' \, \frac{n(\mathbf{r}',t)}{|\mathbf{r}-\mathbf{r}'|} \,, \tag{6}$$

and the exchange-correlation potential $v_{\rm xc}$ is defined by:

$$v_{\rm xc}(\mathbf{r},t) = \frac{\delta \mathcal{A}_{\rm xc}}{\delta n(\mathbf{r},t)} \,. \tag{7}$$

Here $\mathcal{A}_{\rm xc}$ is the exchange-correlation part of the quantum mechanical action of the electronic system,

$$\mathcal{A}_{\rm xc} = \int_{t_0}^{t_1} dt \, \langle \Psi(t) | \, \hat{W} \, | \Psi(t) \rangle - \frac{1}{2} \int_{t_0}^{t_1} dt \int d^3r \, n(\mathbf{r},t) v_{\rm Hartree}(\mathbf{r}) + S_W - S_0 \,, \tag{8}$$

where Ψ stands for the many-body wave-function, S_W is the functional defined by:

$$S_W = \int_{t_0}^{t_1} dt \, \langle \Psi(t) | \, i\frac{\partial}{\partial t} - \hat{T}_e \, | \Psi(t) \rangle \,, \tag{9}$$

and S_0 is equal to expression (9) for $\hat{W} = 0$. $\mathcal{A}_{\rm xc}$ is the term that needs to be approximated. In contrast to ground-state DFT, where rather good functionals are available, approximations to $v_{\rm xc}(\mathbf{r},t)$ are still in their infancy. In particular, the time dependence of $v_{\rm xc}(\mathbf{r},t)$ (or, in an equivalent way, the frequency dependence) is quite hard to model. Nevertheless, some work has been done to go beyond the simple adiabatic approximations[5,6].

Finally, in the Newton equation for the nuclei, Eq. (3), $\mathbf{R}_\alpha$ stands for the coordinate of the nucleus labeled α, m_α for its mass, and $\mathbf{F}_\alpha$ for the force exerted on it. This last quantity is calculated through Ehrenfest's theorem:

$$F_\alpha(\mathbf{R},t) = - \langle \Psi(t) | \, \frac{\partial}{\partial \mathbf{R}_\alpha} \hat{H} \, | \Psi(t) \rangle \,. \tag{10}$$

This is nothing else but the extension of the Helmann-Feynman theorem to the time-dependent domain [22]. We should note that, as we expand our Kohn-Sham wave-functions in a regular grid in space, there are no Pulay correction to the previous expression. This would not be the case if one uses localized basis sets to represent the wavefunctions.

The ground-state of the electronic system in the nuclear equilibrium configuration, i.e. the solution of the ground-state Kohn-Sham equations:

$$\left[-\frac{\nabla^2}{2}+v_{\rm ext}(\mathbf{r})+v_{\rm Hartree}(n;\mathbf{r})+v_{\rm xc}(n;\mathbf{r})\right]\psi_i(\mathbf{r})=\varepsilon_i\psi_i(\mathbf{r})\,, \tag{11}$$

$$n(\mathbf{r})=\sum_{i=1}^{N}|\psi_i(\mathbf{r})|^2\,, \tag{12}$$

is usually used as the starting point for the time-dependent simulations. The Kohn-Sham orbitals are then propagated as (see next section for details):

$$\psi_i(\mathbf{r},t+\Delta t)=e^{-i\int_t^{t+\Delta t}dt\,\hat{H}_{\rm KS}(t)}\psi_i(\mathbf{r},t)\,. \tag{13}$$

Note that only occupied orbitals need to be propagated and there is no need of computing empty states.

To obtain the linear optical absoption spectrum of the system, we follow the scheme proposed by Yabana and Bertsch in Ref. [12], and excite all frequencies of the system by giving some small momemtum ($\mathcal{K}$) to the electrons. This is achieved by transforming the ground-state wave-functions according to:

$$\psi_i(\mathbf{r},\delta t)=e^{i\mathcal{K}z}\psi_i(\mathbf{r},0)\,, \tag{14}$$

and then propagating these wave-functions for some (finite) time. The spectrum can then be obtained from the expression of the dipole strength function $S(\omega)$:

$$S(\omega)=\frac{2\omega}{\pi}\Im\alpha(\omega)\,, \tag{15}$$

where the dynamical polarizability, $\alpha(\omega)$, is essentially the Fourier transform of the dipole moment of the system $d(t)$:

$$\alpha(\omega)=\frac{1}{\mathcal{K}}\int dt\;e^{i\omega t}\left[d(t)-d(0)\right]. \tag{16}$$

With this definition, the Thomas-Reiche-Kuhn f-sum rule is given by the integral:

$$N=\int d\omega\,S(\omega)\,, \tag{17}$$

This sum rule can be used to check the quality of the calculations. Another check is energy conservation, which the TDDFT respects when no external field is applied. For the purpose of obtaining nonlinear optical properties, we follow the evolution of the system under the influence of a laser field that is treated in the dipole approximation (although this constrain can be removed). The emitted harmonic spectra can then be calculated from the acceleration of the dipole moment[7]:

$$H(\omega) \propto \left| \int dt \, e^{i\omega t} \frac{d^2}{dt^2} d(t) \right|^2 . \tag{18}$$

During the propagation, charge density is absorbed at the boundaries of the simulation region, either by an imaginary absorbing potential[23] or a mask function. In the first case, we add to the Kohn-Sham potential:

$$V_{\text{eff}}(\mathbf{r},t) = V_{\text{KS}}(\mathbf{r},t) - iV_{\text{abs}}(\mathbf{r}) \, , \tag{19}$$

where V_{abs} is zero in the inner region of the simulation box, and rises smoothly till the edges. By adjusting both the height and the shape of the potential, we can select which momenta are absorbed and prevent the unwanted reflections at the boundary. When using a mask, the wave-function is multiplied in each time-step by a function which is 1 in the inner simulation region and gradually goes to 0 at the borders:

$$\psi(\mathbf{r},t) \to M(\mathbf{r})\psi(\mathbf{r},t) \, . \tag{20}$$

The absorbed charge can be interpreted as an ionization probability and can be used to estimate the photo-electron spectra. The box size has to be big enough so that the physical system is not perturbed by the absorbing boundaries. Note that the wavefunctions are no longer normalized as the system slowly gets charged.

3 Some numerical aspects

3.1 Input file

One of the authors (MALM) decided to develop an input system aimed at making the interaction with numerical applications powerful, yet user-friendly. These parsing utilities are completely independent of the rest of the package, and can easily be used in other projects.

All input options should be in a file called '`inp`', in the directory `octopus` is run from. For a fairly comprehensive example, just look at the file `samples/inp`. At the beginning of the program, the library `liboct` reads the `inp` file, parses it, and generates a list of variables that will be read by `octopus` . There are two kinds of variables, scalar values (strings or numbers), and blocks (that you may view as matrices). A scalar variable `var` can be defined by:

```
var = exp
```

`var` can contain any alphanumeric character plus "_", and `exp` can be a quote delimited string, a number (integer, real, or complex), a variable name, or a mathematical expression. In the expressions all arithmetic operators are supported, as well as the most common mathematical functions. Blocks are defined as a collection of values, organised in row and column format. The syntax is the following:

```
%var
  exp | exp | exp | ...
  exp | exp | exp | ...
  ...
%
```

Rows in a block are separated by a newline, while columns are separated by the character "`|`". There may be any number of lines and any number of columns in a block. Note also that each line can have a different number of columns.

If `octopus` tries to read a variable that is not defined in the `inp` file, it automatically assigns to it a default value. All variables read are output to the file '`out.oct`'. Everything following the character "#" until the end of the line is ignored.

In some particular cases, it is also possible to define functions in the input file. In this way the user may for example include a tailored external potential in the Hamiltonian.

3.2 Grids

All quantities are represented in an uniform grid in real space. Although more sophisticated discretizations have been proposed in the literature (like the adaptative grid method[24]), we think that the simplicity and intuitiveness of the regular mesh surpass the advantadges of those complex methods. The simulation region can take the shape of a sphere, a cylinder or a parellelepiped. In the latter case, the user can select to have different mesh spacings in each of the spatial directions (useful e.g. for quasi two dimensional quantum dots). Since the wave functions are sensitive to the boundaries, these must be placed at least some Angstroms away from the molecule or cluster.

To profit from the short range of the Kleynman-Bylander projectors, we store them in small spheres centered around each atom. The typical number of points contained in those spheres is a couple of hundreds. This number effectively determines the sparseness of the Hamiltonian matrix.

The Laplacian operator is discretized at the grid points $\mathbf{r}_i$ using a finite-order rule:

$$\nabla^2\psi(\mathbf{r}_i) \approx \sum_j c_j \psi(\mathbf{r}_j)\,. \tag{21}$$

Normally `octopus` uses a 9-point rule, although the user can change this setting in the input file. The expansion coefficients c_j are generated, to the desired order, following an algorithm developed in Ref. [25].

The electronic ground-state is obtained through the self-consistent solution of the Kohn-Sham equations. The starting trial wave-functions may be generated randomly or obtained through a simple LCAO calculation. In the latter case, the Hamiltonian is built using a minimum basis set of atomic pseudo-wave functions. This turns out to be a good starting point for the self-consistent (SC) cycle. The procedure could be refined by improving on the quality of the basis set. [6]

In each step of the SC cycle, the hamiltonian is diagonalized using a conjugate gradients method. The typical dimension of the eigensystem ranges from 10,000 to 250,000. All quantities are calculated in real space, except the Hartree potential (see below).

The electronic density used to start iteration i, $n_i^{\text{start}}(\mathbf{r})$, is obtained from the densities of previous iterations, either by using a linear or a Broyden scheme:

$$n_i^{\text{start}}(\mathbf{r}) = \sum_{j=0}^{i} a_j\, n_j(\mathbf{r})\,, \tag{22}$$

where n_j is the electronic density at the end of iteration j. In the first case, $n_i^{\text{start}}(\mathbf{r})$ is simply a linear combination of $n_{i-1}(\mathbf{r})$ and $n_i(\mathbf{r})$, while in the Broyden scheme $n_i^{\text{start}}(\mathbf{r})$ is extrapolated from the previous iterations by trying to minimize the total energy[26].

The Hartree potential is obtained as the solution of the Poisson equation:

$$\nabla^2 v_{\text{Hartree}}(\mathbf{r}) = -4\pi n(\mathbf{r})\,. \tag{23}$$

[6] As a curiosity, we should note that `octopus` , with very little changes, can be used either as an LCAO, a plane-wave or a real-space based code. However, only minimal basis are employed in the LCAO calculations – extensions to more complete basis sets are planned for the future.

This equation can be solved either by a conjugate-gradients minimation (where boundary conditions are handled through a multipole expansion of the charge inside the simulation box), or by fast Fourier transforms. In the latter case, care has to be taken not to include fictitious cell-cell interactions. This problem has been solved in the literature using two basic methodologies, either by imposing a cut-off to the Coulomb potential [27] or by computing the corrections analyticaly[28]. `octopus` implements the first of these options [29].

Finally, we have implemented a set of exchange and correlations functionals: LDA in the parametrization of Perdew and Zunger [30], GGA in the recipe of Ref. [31], the van Leeuwen and Baerends potential [32], the self-interaction corrected (SIC) LDA functional [30], and the exact exchange [33]. The last two functionals were implemented within the Krieger, Li and Iafrate approximation [34]. The exchange and correlation term is computed in real space. with a proper discretization of the gradient operator in the case of the GGA[35].

3.3 Time evolution

The choice of the propagator algorithm is obviously crucial to any real-time simulation computer code. `octopus` provides several choices.

The preferred and default method is to propagate the electronic wave-functions in real time using a scheme specially designed to enforce time-reversal symmetry. The method is based on the assumption that the wave function forward propagated from t to $t+\Delta t/2$ must be equal to the wave function backward propagated from $t+\Delta t$ to $t+\Delta t/2$:

$$e^{-i\hat{H}(t)\frac{\Delta t}{2}}\psi(t) = e^{i\hat{H}(t+\Delta t)\frac{\Delta t}{2}}\psi(t+\Delta t)\,, \tag{24}$$

which yields:

$$\psi(t+\Delta t) = e^{-i\hat{H}(t+\Delta t)\frac{\Delta t}{2}}e^{-i\hat{H}(t)\frac{\Delta t}{2}}\psi(t)\,. \tag{25}$$

We now have two options:

- We can start by estimating $\psi(t+\Delta t) \approx e^{-i\hat{H}(t)\Delta t}\psi(t)$ by expanding the exponential to fourth order in Δt. This expansion order has especially favorable numerical properties, permitting stable time integration for Δt satisfying $||H||\Delta t < \sqrt{8}$ [36]. This estimate is used in turn to obtain an approximation to $\hat{H}(t+\Delta t)$. The expression (25) is then computed, once more by expanding the exponentials to fourth order.
- We can simplify the process by extrapolating $H(t+\Delta t)$ from a given number of previous iterations Hamiltonians.

It is also possible to use the scheme devised by Flocard, Koonin and Weiss [37] for nuclear simulation processes, or the split operator method. Investigations are underway to raise the level of sophistication of this important component of the code; for a recent compilation of the state-of-the-art algorithms, see Ref. [38].

The maximum time step is determined by the grid spacing, and typical values are of the order of 0.001 fs. [7] A required 0.1 eV accuracy in energy is obtained for total simulation times of 15 fs.

For the matrix-vector multiplication we take advantage of the sparseness of the Hamiltonian matrix in real space by using iterative methods that never require the explicit construction of the full matrix. The sparseness is determined by the finite difference formula for the kinetic energy and by the nonlocal-projectors of the pseudopotential. Typically, more than 99% of the matrix elements are zero.

The Newton's equations for the ions are solved using Verlet algorithms. Both normal Verlet,

$$\mathbf{r}(t+\Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\Delta t) + \mathbf{a}(t)\Delta t^2, \tag{26}$$

and the velocity Verlet,

$$\begin{aligned} \mathbf{r}(t+\Delta t) &= \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 \\ \mathbf{v}(t+\Delta t) &= \mathbf{v}(t) + \frac{1}{2}[\mathbf{a}(t) + \mathbf{a}(t+\Delta t)]\Delta t \end{aligned} \tag{27}$$

were implemented[39]. As propagating the electronic wave functions is the most time consuming part, we simply use the same time-step for the electrons and the ions. This time-step is so small compared with nuclear time-scales that any energy-conserving algorithm would yield the same nuclear trajectories.

Usually propagating the system takes between one and two orders of magnitude more time than obtaining the ground-state wave-functions. We therefore parallelized the time evolution using MPI, by distributing the wave-functions among different processors. For optimal efficiency, the number of electronic states should be a multiple of the number of processors. This procedure scales almost linearly until the number of states per processor approaches 4-5. To improve the scaling behaviour, one could further parallelize the calculation of the Hartree and the exchange-correlation potentials. This, was nevertheless found unnecessary for the systems and computers we investigated, but can become relevant for very large systems.

[7] For an uniform spatial grid with $\Delta x = 0.3$ Å spacing, a stable time-step to perform the time-evolution is $\Delta t = 0.001\,\mathrm{fs} \ll m(\Delta x)^2/\hbar$.

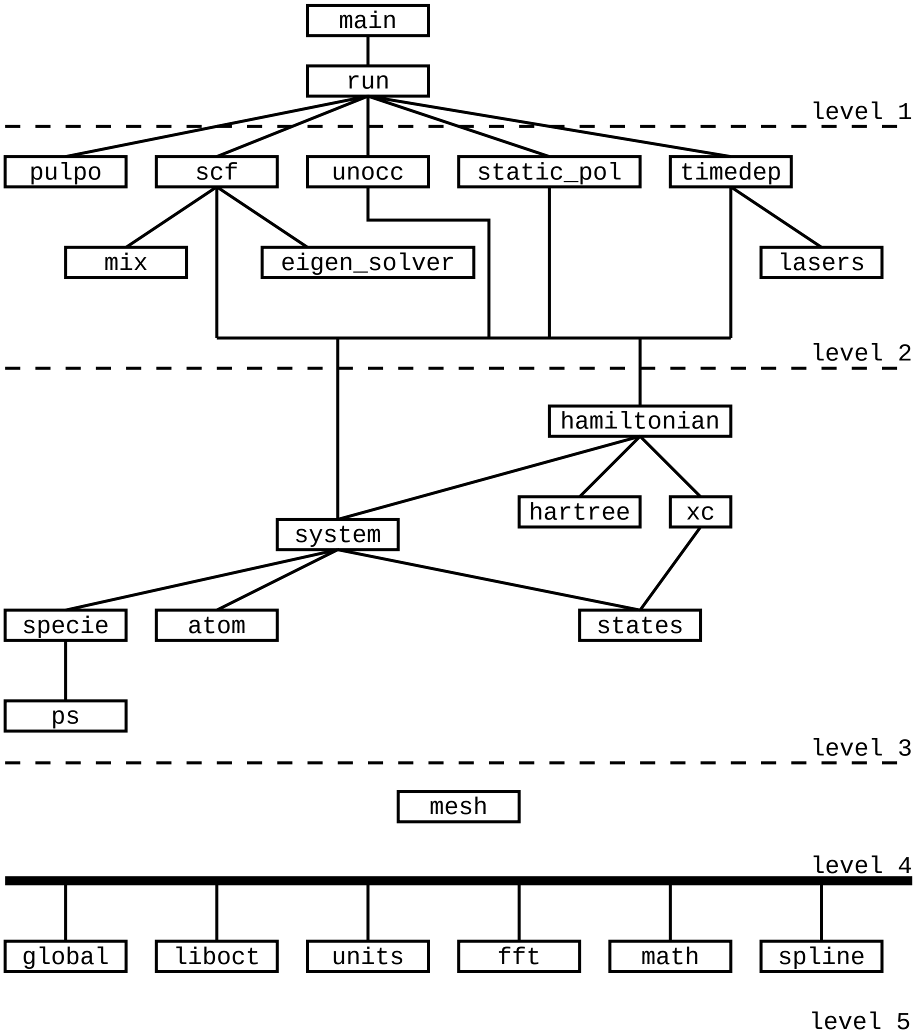

Fig. 1. Simplified schematic chart of the program.

4 Structure of the program

The structure of the program is schematically depicted in Figure 1. Each one of the squares denotes a Fortran 90 module, and the lines represent module dependencies. For clarification purposes, we have considerably simplified the diagram, stressing only the main relationships and modules. In the following, we will only provide a short description - the interested reader should refer to the source code for further insight. One can divide the modules into 5 levels. The top-most level, number 1, comprises the `main` program unit, and the module `run`.

- `main`: initializes some external libraries, and passes control to the run module.
- `run`: the program can be run in different run modes, as requested by the user. The purpose of this module is to break each mode into smaller tasks and execute them.

In level 2, we find the modules responsible for executing the main tasks.

- `pulpo`: prints an octopus recipe in the users language of choice. In some of the authors opinion, this is the most important problem the program is able to solve: What to cook for dinner?
- `scf`: calculates the ground-state of the electronic system, by solving self-consistently the static Kohn-Sham equations. The wave-functions thus obtained can then be used as an input for the time-dependent calculations. This module makes use of:
  - `mix`: calculates the electronic density for the next self-consistent iteration. Both linear and Broyden mixing are available.
  - `eigen_solver`: diagonalizes the hamiltonian using a conjugate gradients method.
- `unocc`: after performing the self-consistent cycle, one can calculate unoccupied states using this module.
- `static_pol`: it calculates the static polarizability of the system using a finite field method.
- `timedep`: performs the evolution in time of the system. An external time-dependent field can be included in the simulation. The output from this run mode can then be processed by external utilities in order to calculate spectra. This module uses:
  - `lasers`: handles the laser fields applied during the simulation.

The next level contains modules that define the physical system under study and its Hamiltonian:

- `system`: it serves as a container for the modules that define the system under study:
  - `specie`: defines the chemical species or model potentials. These can be specified through non-local pseudopotentials, or through a user defined function. In the first case, it uses the module:
    - `ps`: generates all information relative to the pseudopotentials.
  - `atom`: contains all routines related to the geometry of the system.
  - `states`: this module describes the electronic states. It is one of the most important modules for it handles the wave-functions, density, occupation numbers, etc.
- `hamiltonian`: Contain all routines necessary to generate the Hamiltonian of the system. It uses the modules:
  - `system`: to obtain the electron-ion and ion-ion interaction.
  - `hartree`: to generate the Hartree potential. It can be calculated either using a conjugate-gradients method, or by using fast Fourier transforms.
  - `xc`: to generate the exchange-correlation potential. Several flavors are included, from the local density approximation, passing by the generalized gradient functional, to the more esoteric orbital dependent functional in the Krieger, Li and Iafrate approximation.

We consider the module `mesh` to be in a level by itself, as it defines the regular grid of points in which our functions are discretized. It is used in almost all other upper level modules.

- `mesh`: It contains functions to generate the grid, to integrate, differentiate, etc.

Finally, level 5 has utility modules, essencially independent of each other and from the upper level, that are used for specific auxiliary tasks.

- `global`: includes error handling, debugging, definition of constants, and string functions. It is included in all other modules.
- `liboct`: interfaces to the `liboct` library, which will be described below.
- `units`: handles input and output units. We should note that internally the code uses solely Hartree atomic units.
- `fft`: defines contants and routines used to interface the FFTW library.
- `math`: some mathematical functions (other mathematical functions are included in the `liboct` library.
- `spline`: routines necessary to interface the spline functions from the `liboct` library (which in turn uses the GSL interpolation functions).

Almost all modules have the same structure: first one encounters the definition of data strutures; there is a routine (named `modulename_init`) that initializes the structures, another that destroys them (`modulename_end`), and others to perform the necessary operations related to the structures (usually named `modulename_routinename`). The reader can notice that this is very much in the spirit of object-oriented programming.

As Fortran is really not well suited for some programming tasks, we have used C to implement some basic utility functions. These are contained in the `liboct` library. Essencially it includes the input parser (the grammar is written in bison), and interfaces to the GSL library.

The code is complemented with utilities to post-process the generated data files. We provide small programs to calculate the linear absoption spectrum and the harmonic spectrum and to link the data to the open source visualization package OpenDX [40], in order to plot 3D functions like the density or the wave-functions (this utility is still in an experimental stage). Such pictures (and animations) may help us to understand the physical processes.

5 Installation

As `octopus` makes use of the GNU autotools (automake/autoconf/libtool), installation should be straightforward: in some standard computer systems, typing

```
./configure
make
```

```
make install
```

should be sufficient. Nevertheless the user can find detailed instructions, including troubleshooting, in the manual.

6 Running `octopus`

After setting up the input file, the program can be run by simply typing the command `octopus`.

`octopus` has a large number of options, that we will not enumerate here for they are extensively described in the manual. As an example, and also because it gives a good overview on the program's capabilities, we explain the option "CalculationMode". From the manual:

`CalculationMode` (integer, default=1): It defines the type of simulation to perform. Modes are:

- 1: Start ground-state calculation.
- 2: Resume ground-state calculation.
- 3: Calculate unoccupied states.
- 4: Resume calculation of unoccupied states.
- 5: Start time-dependent simulation.
- 6: Resume time-dependent simulation.
- 7: Start static polarizability calculation.
- 8: Resume static polarizability calculation.
- 99: Prints out an octopus recipe.

7 Some examples

7.1 Linear absorption spectra

Calculation of linear optical absorption spectra within TDDFT is today a very well-tested technique. The framework can, in fact, be traced back to early works on atomic polarizabilities back in 1980 [41]. Applications to molecules and clusters were performed much more recently [42,16,13,5,6,11]. A particularly successful example is the benzene molecule, calculated by Yabana and Bertsch with the predecessor of the present code [43]. In this case, the comparison of the experimental and calculated optical absorption spectrum is excellent. Therefore, we decided to replicate those calculations as a first example. The results are shown in Fig. 2. The

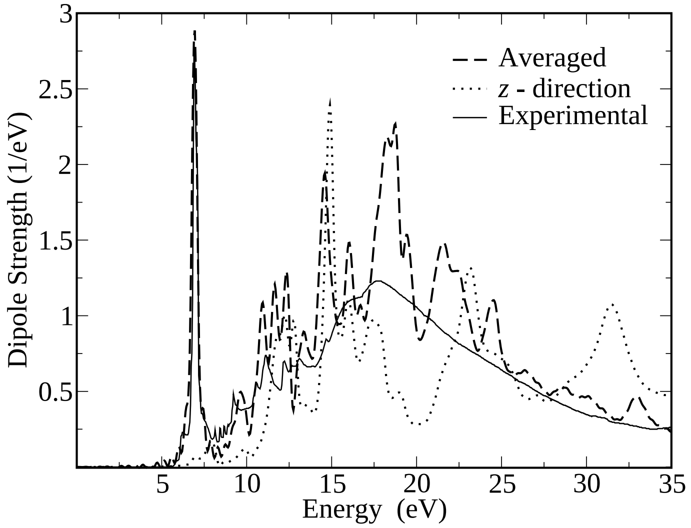

Fig. 2. Optical absorption of the benzene molecule, from Eq. 15. Experimental results from Ref. [44].

main features of the plot are: (i) the narrow peak at about 7 eV; it corresponds to the $\pi \to \pi^*$ transition – a characteristic of carbon conjugate compounds, and (ii) the broad feature above 9 eV, which corresponds to the $\sigma \to \sigma^+$ transition.

Regarding technical details, we fixed the benzene geometry to its D_6 symmetric configuration, with a CC bond length of 1.396 Å, and CH bond length of 1.083 Å. Calculations were made using a cylindrical box of 6 Å of radius and 7 Å height, and the grid spacing was 0.22 Å – which determine ≈ 72,000 mesh points [8]. The exchange and correlation functional employed was LDA, but these results are in fact rather independent of any (reasonable) choice of the xc functional. The evolution was performed with a time step of 0.002 eV^{-1}, ensuring energy conservation to better than one part in a million.

Using the above-mentioned OpenDX program, we have plotted in Fig. 3 the π and π^* orbitals of the benzene molecule; oscillations between these (doubly degenerate) states are responsible for the strong absorption at 7 eV.

At the end of this article, we attach the input file for the static calculation, as well as the resulting "`C6H6.static`" file.

[8] Minimization of this experimental geometry leads to almost the same configuration. However, although 0.22 Å is enough to converge the absorption spectrum, a finer grid would be needed to perform the minimization with present code in real space.

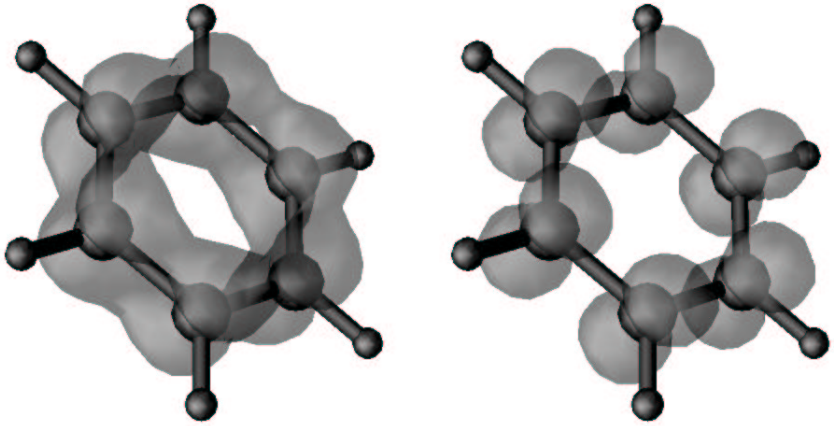

Fig. 3. π (left) and π^* (right) Kohn-Sham orbitals of benzene, as calculated by `octopus` . Transitions between these two states are responsible for the large absorption peak at 7 eV.

7.2 Non-linear harmonic spectra

When strong laser pulses interact with atoms or molecules, high order harmonics are generated [7,45]. This process has been widely investigated in the past years – one of the reasons being that harmonics may be used as a source of coherent radiation in the extreme ultraviolet or soft X-ray region [9]. Most of the work, however, has focused on atoms. The shape of the spectra is qualitatively similar for all atomic gases: an initial decrease followed by a plateau, and a more or less sharp cutoff beyond which there is no emission. This behaviour is usually explained by invoking the recollision mechanism [46]. Investigations regarding molecules, despite less numerous, have also been reported, both experimentally [47] and theoretically [48]. One conclusion that can be drawn is that the harmonic spectra of diatomic molecular gases are in general very similar to those in rare gases, with the characteristic plateau/cutoff behaviour.

To exemplify the use of `octopus` in the calculation of this strongly non-linear phenomenon we use the H_2 molecule. In this case, we chose to study the problem in one dimension, which considerably reduced the computational resources needed [9] . The exchange-correlation was approximated by the exact exchange functional, and the system was confined in a box of length 40 Å and grid spacing 0.12 Å. We modelled the interactions (both the electron-electron and electron-nucleus) by soft-Coulomb potentials of the form $1/\sqrt{1+x^2}$. The laser pulse applied had a frequency of $\omega = 1.6$ eV, peak intensity 10^{14} W/cm^2, and envelope formed by a linear ramp until the tenth optical cycle and a constant function for another twenty cycles. For the evolution, the time step was 0.001 eV^{-1}. Fig. 4 shows the final outcome of the calculation. The maximum conversion efficiency is observed betweent the ninth

[9] Although 3D calculations for this example are also feasible with `octopus` , this example serves to illustrate the use of the program in 1D.

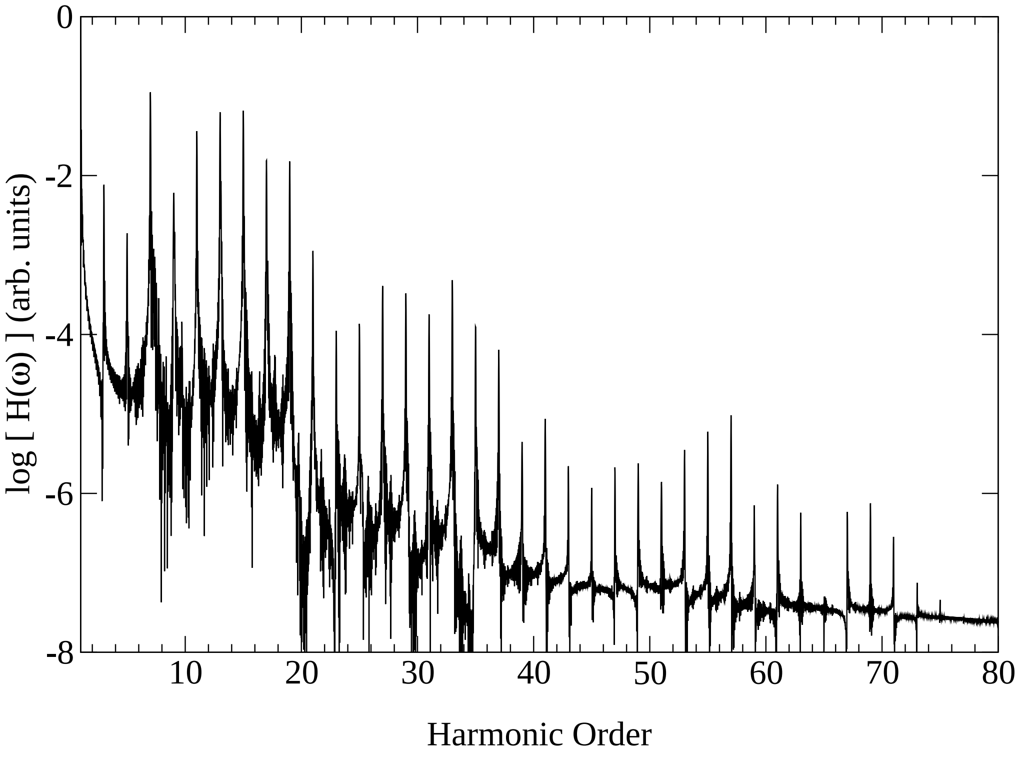

Fig. 4. Harmonic spectrum of H_2 for an applied laser of frequency ω = 1.6 eV, in arbitrary units.

and the nineteenth harmonic order; the final cutoff for this intensity approaches the 75th order.

Generation of high harmonics is subject to selection rules that may forbid certain peaks [49]. For example, parity symmetry leads to the oblivion of even harmonics. In a recent paper, Kreibich *et al.*[50] have demonstrated how non-adiabatic effects break this symmetry and lead to the appearance of even harmonics for homopolar but isotopically assymetric dimers. Since our evolution is also non-adiabatic, we decided to address this problem [51]. We concluded that neglection of the quantum nature of the hydrogen nuclei drastically reduces this non-adiabatic effect – although it is indeed present.

7.3 Electron-ion dynamics

Almost all the applications of TDDFT in the field of laser physics have only involved electronic dynamics. But TDDFT may be used to describe the coupled nuclear and electronic motion, either (i) with a classical treatment of the nuclei, by doing Ehrenfest molecular dynamis – as described in Section 2 – or (ii) with a full quantum treatment of the nuclei, by using multi-component TDDFT. Up to now, octopus only implements option (i), which is nevertheless valid for a wide range of physical situations.

As an example of the combined electron-ion dynamics, we have studied the femtosecond-laser induced dissociation of the singly ionised sodium dimer, Na_2^+. This is a particularly simple test-case, but it can nevertheless show some interesting features. Also, dissociation of this system after irradiation with a femtosecond laser has been recently investigated experimentally [52]. In Fig. 5 we depict the results of three different runs, in which the laser frequency and intensity were varied – the rest of the calculation variables kept constant: the laser envelope was a semicycle cosine of 80 fs length; the radius of the spherical simulation box was 10 Å, and the grid spacing was 0.3 Å. A Troullier-Martins norm conserving nonlocal pseudopotential was used to model the core-electron interaction. We can interpret the obtained results as follows:

- $\omega = 2.52$ eV (solid line) is a resonant frequency; it corresponds to the transition between the ground and first excited electronic states, which is anti-bonding [53]. As such, a weak intensity ($\approx 10^{10}$ W/cm^2) produces photodissociation on the light induced potential.
- $\omega = 3.2$ eV (dotted line) is also a resonant frequency, which corresponds to the transition between the ground and the second excited electronic state. In contrast to previous case this state is bonding, and therefore one should not expect photodissociation – this is the result observed in Fig. 5.
- Using a non-resonant frequency, $\omega = 1.57$ eV (dashed line), one cannot induce the photodissociation unless ionising the system (the so-called Coulomb explosion channel). As for all three cases the intensity was too low to produce significative ionisation, the system remains bound.

In light of these results, we expect that laser induced orientation, alignment, fragmentation, and isomerization, as well as excited-state activated chemical reativity can in principle be addressed with the present code. Due to its good scaling properties, the size of the systems may range from the simple test-cases presented in present communication, to medium-sized molecular systems.

8 Acknowledgements

We would like to thank K. Yabana for making available the source code of his program and F. Nogueira for useful discussing and beta-testing. This work was partially supported by the RTN program of the European Union NANOPHASE (contract HPRN-CT-2000-00167), DGESIC (PB98-0345) and JCyL (VA28/99). G.B. acknowledges support by the U.S. Department of Energy under Contract No. E-FG-

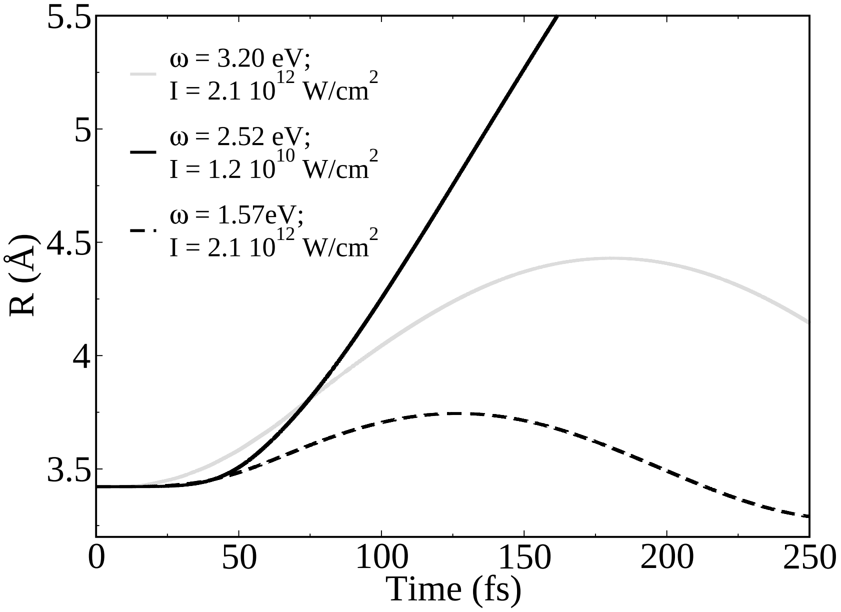

Fig. 5. Internuclear distance of Na_2^+, as a function of time, for three different laser pulses of the frequency and intensity given.

06-90ER-41132. Some of the calculations present in the examples were perform in the computers of CESCA and CEPBA.

References

[1] See, for example, the following compilations of DFT results in molecules, surfaces and solids: Recent Advances in Density Functional Methods, World Scientific, Singapore (2002) and (1995); *Recent Developments and Applications of Modern Density Functional Theory* Ed. J.M. Seminario, (Elsevier, Amsterdam) (1996); Dobson, J , G. Vignale and M.P. Das (Eds), *Electronic Density Functional Theory: Recent Progress and New Directions* (Plenum, New York) (1997).

[2] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992); W. E. Pickett, Comput. Phys. Rep. **9**, 115 (1989); M. L. Cohen, Solid Stat. Commun. **92**, 45 (1994); Phys. Scri. **1**, 5 (1982).

[3] S. Baroni, S. de Gironcoli, A. Dal Corso, and P. Giannozzi, Rev. Mod. Phys. **73**, 515 (2001).

[4] T. L. Beck, Rev. Mod. Phys. **72**, 1041 (2000).

[5] See for example the review: E. K. U. Gross, F. J. Dobson, and M. Petersilka, *Density Functional Theory* (Springer, New York, 1996); K. Burke, M. Petersilka and E. K. U. Gross, in *Recent Advances in Density Functional Methods*, edited by P. Fantucci and A. Bencini (World Scientific, Singapure) (2001); M. E. Casida, 1995, in Recent Advances in Density Functional Methods, Part I, ed. D. P. Chong (World Scientific, Singapore), p.155; R. van Leeuwen, Int. J. Mod. Phys. B **15** , 1969 (2001).

[6] G. Onida, L. Reining and A. Rubio, Rev. Mod. Phys. (in press 2002). A review comparing many-body based technique and TDDFT technques pointing out the practical advantages and limitations of each approach.

[7] T. Brabec and F. Krausz, Rev. Mod. Phys. **72**, 545 (2000); M. Protopapas, C.H. Keitel and P.L. Knight, Rep. Prog. Phys. **60**, 389 (1997).

[8] H. Ihee, V.A. Lobastov, U.M. Gomez, B.M. Godson, R. Srinivasan, C.-Y. Ruan and A.H. Zewail, Science **291**, 458 (2001).

[9] R. Bartels, S. Backus, E. Zeek, L. Misoguti, G. Vdovin, I.P. Christov, M.M. Murnane and H.C. Kapteyn, Nature **406**, 164 (2000).

[10] J.H. Ebberly, J. Javanainen and R. Rzazewski, Phye. Rep. **204**, 331 (1991).

[11] J.R. Chelikowsky, N., Troullier J.R. Chelikowsky, N. Troullier and Y. Saad, Phys. Rev. Lett. **72**, 1240 (1994); Phys. Rev. B **50**, 11355 (2994) I. Vasiliev, S. Öğüt , and J.R. Chelikowsky, Phys. Rev. Lett. **82** 1919 (1999); Phys. Rev. B. **65**, 115416 (2002); and references therein.

[12] K. Yabana, and G.F. Bertsch, Phys. Rev. B **54**, 4484 (1996); Z. Phys. D (Atom. Mol. Cl.) 42, 219 (1997); Phys. Rev. A **58**, 2604 (1999); Phys. Rev. A **60**, 1271 (1999); Phys. Rev. A **60**, 3809 (1999); Int. J. Quant. Chem. **75**, 55 (1999); J.I. Iwata., K. Yabana, and G.F. Bertsch, J. Chem. Phys. **115**, 8773 (2001).

[13] A. Rubio, J.A. Alonso, X. Blase, L.C. Balbás, and S.G. Louie, Phys. Rev. Lett., 77, 247-250 (1996); E5442 (1996); Int. J. Mod. Phys. B **11**, 2727 (1997).

[14] G. Monard and K. M. Herz, Acc. Chem. Res. **32**, 404 (1999).

[15] G.F. Bertsch, J.-I. Iwata, A. Rubio and K. Yabana, Phys. Rev. B **62**, 7998 (2000).

[16] M. A. L. Marques, A. Castro and A. Rubio, J. Chem. Phys. **115** 3006 (2001); A. Castro, M. A. L. Marques, J. A Alonso, G. F. Bertsch, K. Yabana and A. Rubio, J. Chem. Phys. **116**, 1930 (2002).

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

[18] S. Goedecker, M. Teter and J. Hutter, Phys. Rev. B **54**, 1703 (1996). C. Hartwigsen, S. Goedecker and J. Hutter, Phys. Rev. B **58**, 3641 (1998).

[19] L. Kleinman and D.M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982).

[20] P. E. Blöch, Phys. Rev. B **41**, 5414 (1990).

[21] A. Castro, M. A. L. Marques, X. Lopez, G. Bertsch and A. Rubio (work in progress).

[22] H. Hellmann, Z. Phys. **85**, 180 (1933); R. P. Feynman, Phys. Rev. **56**, 340 (1939). See also M. Di Ventra and S. T. Pantelides, Phys. Rev. A **61**, 16207 (2000).

[23] See, e.g. T. Nakatsukatsa and K. Yabana, J. Chem. Phys. **114**, 2550 (2000), and references therein.

[24] F. Gygi, Phys. Rev. B **48**, 11692 (1993); F. Gygi and G. Galli, Phys. Rev. B. **52**, R2229 (1995).

[25] B. Fornberg and D. M. Sloan, Acta Numerica 203-267 (1994).

[26] P. Bendt, and A. Zunger, Phys. Rev. B **r**26, 3114 (1992), C. G. Broyden, Math. Comput. **19**, 577 (1965).

[27] G. Makov and M. C. Payne, Phys. Rev. B **51**, 4014 (1995); M. R. Jarvis, I. D. White, R. W. Godby and M. C. Payne, Phys. Rev. B **56**, 14972 (1997); G. Onida, L. Reining, R.W. Godby, R. Del Sole and W. Andreoni, Phys. Rev. Lett. **75**, 818 (1995).

[28] P. A. Shultz, Phys. Rev. B **60**, 1551 (1999); Phys. Rev. Lett. **84**, 1942 (2000); M. C. Strain, G. E. Scuseria , M. J. Frisch, Science **271**, 51 (1996); G. Lauritsch and P.-G. Reinhard, Int. J. Mod. Phys. C **5**, 65 (1994).

[29] A. Castro, A. Rubio and M. J. Stott, physics/0012024.

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

[31] J. P. Perdew, K. Burke and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[32] R. van Leeuwen and E. J. Baerends, Phys. Rev. A **49**, 2421 (1994).

[33] R. T. Sharp, and G. K. Horton, Phys. Rev. **90**, 317 (1953); J. D. Talman, and W. F. Shadwick, Phys. Rev. A **14**, 36 (1976); C. A. Ullrich, U. J. Gossmann, and E. K. U. Gross, Phys. Rev. Lett. **74**, 872 (1995).

[34] J. B. Krieger, Y. Li, and G. J. Iafrate, Phys. Rev. A **45**, 101 (1992).

[35] J. A. White and D. M. Bird, Phys. Rev. B **50**, 4954 (1994).

[36] J. Giansiracusa, private communication.

[37] H. Flocard, S. E. Koonin and M. S. Weiss, Phys. Rev. C **17**, 1682 (1978).

[38] C. Lubich, *Integrators for Quantum Dynamics: A Numerical Analyst's Brief Review*, in *Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms*, edited by J. Grotendorts, D. Marx and A. Muramatsu, (John von Neumann Institute for Computing, Jülich).

[39] M. P. Allen, and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon Press, Oxford, 1987).

[40] http://www.opendx.org/

[41] M. J. Stott and E. Zaremba, Phys. Rev. A **21**, 12 (1980); A. Zangwill and P. Soven, Phys. Rev. A **21**, 1561 (1980).

[42] K. Yabana and G. F. Bertsch, Phys. Rev. B **54**, 4484 (1996); Int. J. Quantum Chem. **75**, 55 (1999).

[43] K. Yabana and G. F. Bertsch, Int. J. Quantum Chem. **75**, 55 (1999).

[44] E. E. Koch and A. Otto, Chem. Phys. Lett. **12**, 476 (1972).

[45] A. McPherson, G. Gibson, H. Jara, U. Johann, T. S. Luk, I. A. McIntyre, K. Boyer, and C. K. Rhodes, J. Opt. Soc. Am. B **4**, 595 (1987).

[46] P. B. Corkum, Phys. Rev. Lett. **71**, 1994 (1993).

[47] Y. Liang, S. Augst, S. L. Chin, Y. Beaudoin and M. Chacker, J. Phys. B **27**, 5119 (1994); D. J. Fraser, M. H. R. Hutchinson, J. P. Marangos, Y. L. Shao, J. W. G. Tisch and M. Castillejo, J. Phys. B **28** L739; H. Sakai and K. Miyasazi, Appl. Phys. B **61** 493; C. Lyngå, A. L'Huillier and C.-G. Wahlström, J. Phys. B **29**, 3293 (1996).

[48] M. Yu. Ivanov and P. B. Corkum, Phys. Rev. A **48**, 580 (1993); T. Zuo, S. Chelkowski and A. D. Baudrauk, *ibid*. **48**, 3837 (1993); M. Lein, N. Hay, R. Velotta, J. P. Marangos and P. L. Knight, Phys. Rev. Lett. **88**, 183903 (2002).

[49] O. E. Alon, V. Averbukh and N. Moiseyev, Phys. Rev. Lett. **80**, 3743 (1998).

[50] T. Kreibich, M. Lein, V. Engel and E. K. U. Gross, Phys. Rev. Lett. **87**, 103901 (2001).

[51] A. Castro, M. A. L. Marques, J. A. Alonso, G. F. Bertsch and A. Rubio, submitted for publication.

[52] A. Assion, T. Baumert, U. Weichmann and G. Gerber, Phys. Rev. Lett. **86**, 5695 (2001).

[53] S. Magnier, F. Oise and M. Seeuws, Mol. Phys. **89**, 711 (1996).

TEST RUN OUTPUT

Input file:

```
CalculationMode = 1
LocalPotentialSpace = no
LCAOStart = yes
TypeOfMixing = 2

BoxShape = cylinder
radius   = 6
zlength  = 3.5
spacing  =  0.22

SystemName = "C6H6"
Units = "eVA"

ConvAbsDens = 1e-05

%Species
 "H"  |  1.0079   |  1 | "tm" | 0 | 0
 "C"  | 12.011    |  6 | "tm" | 1 | 1
%
```

```
%Coordinates
 "C" |  0.000 |  1.396 | 0.000 | no
 "C" |  1.209 |  0.698 | 0.000 | no
 "C" |  1.209 | -0.698 | 0.000 | no
 "C" |  0.000 | -1.396 | 0.000 | no
 "C" | -1.209 |  0.698 | 0.000 | no
 "C" | -1.209 | -0.698 | 0.000 | no
 "H" |  0.000 |  2.479 | 0.000 | no
 "H" |  2.147 |  1.240 | 0.000 | no
 "H" |  2.147 | -1.240 | 0.000 | no
 "H" |  0.000 | -2.479 | 0.000 | no
 "H" | -2.147 |  1.240 | 0.000 | no
 "H" | -2.147 | -1.240 | 0.000 | no
%
```

Output file (“`C6H6.static`”):

```
System name: C6H6

Mesh:
  Type = cylinder         Radius [A] =   6.000, zlength [A] =   3.500
  Spacing [A] = ( 0.220, 0.220, 0.220)    volume/point [A^3] =  0.01065
  # inner mesh =  72447   # outer mesh =  38784

Exchange and correlation functionals:
      Exchange    family    : LDA
                  functional: non-relativistic
      Correlation family    : LDA
                  functional: Perdew-Zunger

SCF converged in   12 iterations

Eigenvalues [eV]
   #  Eigenvalue   Occupation      Error (1)
   1   -21.043553     2.000000 (  0.00000280)
   2   -18.266178     2.000000 (  0.00000392)
   3   -18.242553     2.000000 (  0.00000394)
   4   -14.773587     2.000000 (  0.00000574)
   5   -14.737536     2.000000 (  0.00000586)
   6   -12.947971     2.000000 (  0.00000458)
   7   -11.095505     2.000000 (  0.00002115)
   8   -11.085865     2.000000 (  0.00000574)
   9   -10.263398     2.000000 (  0.00001544)
  10   -10.220461     2.000000 (  0.00000630)
  11    -9.273797     2.000000 (  0.00000404)
  12    -8.297391     2.000000 (  0.00000720)
  13    -8.270419     2.000000 (  0.00000717)
  14    -6.550202     2.000000 (  0.00000458)
  15    -6.544706     2.000000 (  0.00000464)

Energy [eV]:
      Ion-ion     =   2805.78929284
      Eigenvalues =   -363.22624270
      Potentials  =  -3133.07876456
      Exchange    =   -292.99706076
      Correlation =    -43.44935609
      Total       =  -1026.96213126

Convergence:
      abs_dens = 8.22590990E-06 (1.00000000E-05)
      rel_dens = 2.74196997E-07 (0.00000000E+00)
      abs_ener = 7.20987714E-11 (0.00000000E+00) [eV]
      rel_ener = 1.91041591E-12 (0.00000000E+00) [eV]

Forces on the ions [eV/A]
 Ion                        x              y              z
   1         C       0.000000      -1.472136       0.000000
   2         C      -0.839306      -1.337475       0.000000
   3         C      -0.839306       1.337475       0.000000
   4         C       0.000000       1.472136       0.000000
   5         C       0.839306      -1.337475       0.000000
   6         C       0.839306       1.337475       0.000000
   7         H       0.000000       1.825650       0.000000
   8         H      -0.447528      -0.066911       0.000000
   9         H      -0.447528       0.066911       0.000000
  10         H       0.000000      -1.825650       0.000000
  11         H       0.447528      -0.066911       0.000000
  12         H       0.447528       0.066911       0.000000
```