Ehrenfest Molecular Dynamics: one simple example of photo-dissociation with a laser pulse

1 Introduction

In these exercises, we will do some simulations on laser matter irradiation. Due to the limited time, we will work with a one-electron system, but the level of theory we will be aiming at is that of non-adiabatic Molecular Dynamics (MD) based on time-dependent density-functional theory (TDDFT) and the Ehrenfest equations. In fact, if time permits and your computer can, it will be an easy exercise to add one electron and do a "real" TDDFT calculation. If you don't know much about these topics: Refs [1, 2, 3, 5] are some example references for the combination of TDDFT and non-adiabatic first principles MD.

2 A molecule irradiated with a laser pulse.

We will simulate the irradiation of a molecule with a laser field. Since we cannot do heavy calculations in little time, let us use a simple one, the Na_2^+ molecule, which only has one valence electron. `octopus` uses pseudopotentials, and therefore we will only be dealing with that one valence electron (all other inner electrons are supposed to be "frozen" and do not participate in the chemistry).

First, we need to get the ground state of the molecule, which can be done by making use of file `inp.1`:

```
##################################################
# Ground state of the Na2+ molecule
CalculationMode = gs
FromScratch = yes

BoxShape = sphere
```

```
Spacing = 0.7
Radius = 20.0

TheoryLevel = independent_particles

ExcessCharge = 1

%Coordinates
"Na" | -3.282843 | 0 | 0
"Na" |  3.282843 | 0 | 0
%

EigenSolver = cg
EigenSolverMaxIter = 250
EigenSolverTolerance = 1.0e-6
ConvRelDens = 1.0e-6
################################################
```

Note the presence of the `ExcessCharge` variable. The code computes the number of electron that it needs according to the type of atoms. In this case, since Na only has one valence electron, and we have two Na atoms, in principle the code would use two electrons. However, if we want to have a charged system, we need to specify it by setting this variable. The `EigenSolver...` and `ConvRelDens` variables refer to the degree of numerical convergence that should be achieved in the diagonalization of the Hamiltonian for these ground state calculations. You can learn about them in the variables description of the web page, or by running the `oct-help` command.

- One should perform a convergence analysis to be sure that the `Spacing` and `Radius` are correct. If your are too lazy to to this by hand, you can find a sample bash script to to this job for you on the tutorial page (section Nitrogen atom).

- By changing the bond-length given in the block `Coordinates`, you can check whether or not the system is at the equilibrium geometry. You can vary the bond length in several consecutive calculations, and plot the total energy and forces on the ions as a function of bond-length.

- In fact, to be even more sure, you can perform a geometry optimization run (`CalculationMode = go`), and find the right equilibrium geometry.

We will need to know which are the excitation energies of the system, and therefore we will need once again to perform a calculation in the `CalculationMode = unocc` mode with the `inp.2` file:

```
##################################################
# Calculation of excited states of Na2+ molecule.
CalculationMode = unocc
ExtraStates = 4
FromScratch = yes

BoxShape = sphere
Spacing = 0.7
Radius = 20.0

TheoryLevel = independent_particles

ExcessCharge = 1

%Coordinates
"Na" | -3.282843 | 0 | 0
"Na" |  3.282843 | 0 | 0
%

EigenSolver = cg
EigenSolverMaxIter = 250
EigenSolverTolerance = 1.0e-6
ConvRelDens = 1.0e-6
##################################################
```

Now we will perform time-dependent simulations. Take a look at file `inp.3`:

```
##################################################
# Ground state of the Na2+ molecule
CalculationMode = td
FromScratch = yes

BoxShape = sphere
Spacing = 0.7
Radius = 20.0

TheoryLevel = independent_particles

ExcessCharge = 1

%Coordinates
"Na" | -3.282843 | 0 | 0
"Na" |  3.282843 | 0 | 0
%

TDEnergyUpdateIter = 1

TDPropagator = exp_mid
```

```
TDExponentialMethod = lanczos
TDExpOrder = 20

AbsorbingBoundaries = mask
AbWidth = 4.0

MoveIons = no

omega = 0.1
electric_amplitude = 0.05
tau0 = 2*(2*pi)/omega
t0 = 2*(2*pi)/omega
totaltime = t0+tau0

TDTimeStep = totaltime / 250
TDMaximumIter = totaltime/TDTimeStep

%TDExternalFields
electric_field   | i | 0 | 0 | omega | "envelope_function"
%

%TDFunctions
"envelope_function" | tdf_cosinoidal | electric_amplitude | tau0 | t0
%

TDOutput = multipoles + laser + energy
#################################################
```

Here are the novelties:

- The equations are propagated by *discretizing* the time interval that is going to be simulated in smaller pieces or time steps. `TDTimeStep` determines the size of this time step. If this value is too large, the propagations will be unstable. But the larger it is, the faster we will simulate the system. Change it and see how your total simulation times vary.

- The `TDPropagator` variable sets which propagation algorithm is used. You may read about this topic in Ref. [6], and also consult the various options in the variable description of the web page. The exponential midpoint rule, which is the option chosen in the input file, is given by:

$$\varphi(t + \Delta t) = \exp\{-i\Delta t \hat{H}(t + \Delta t/2)\}\varphi(t)\,. \tag{1}$$

- Even though the propagation scheme is now fixed, it requires of an algorithm to compute the action of the exponential of a matrix on a

wave function, which is a non-trivial task when the dimension is large. The algorithm to do this is chosen by the `TDExponentialMethod` and `TDExpOrder` variables.

- When the electrons are propagated in time with an intense laser field, some ionization may occur. In order to account for this, one has to add absorbing boundaries to the simulation box. This is done, in this case, by applying a "mask function" (`AbsorbingBoundaries = mask`) that at each time step cancels a part of the wave function that is close to the simulation box boundary.

- `MoveIons = no` means that we will perform the calculation with the nuclei fixed to their original position. We will relax this condition in the next section.

- Finally, the external field (that simulates the electric field created by a laser pulse) is described through the blocks `TDExternalFields` and `TDFunctions`. These are reasonably well described in the code web page, so we will not repeat the explanations here. Note only that the applied electric field has the form:

$$\mathbf{E}(t) = \mathrm{Re}[f(t)e^{i\omega t}\mathbf{p}]\,, \tag{2}$$

  where $f(t)$ is the envelope funciton, ω is the carrier frequency (`omega` in the `inp` file), and $\mathbf{p}$ is the polarization vector, which can be complex.

- The total simulation time is specified through the line `TDMaxSteps = totaltime/TDTimeStep`: the total time `totaltime` is divided by the time step, to yield the number of time steps.

Now it is time to analyze the results. Take a look at the variable `TDOutput`: it orders the code to create some output files, with information about the multipoles of the system (monopole – i.e. the total electronic charge present in the simulation box – and the dipole), the laser field, or the energy as the system evolves in time. You will find the files in the directory `td.general`.

You should try to plot these files with some plotting program (`gnuplot` should be installed in your computers). For example, the second column of the file `multipoles` contains the time at each time step, whereas the third column contains the electronic charge in the simulatin box. One of the most relevant information to learn from this simulation is the ionization yield, as a function of time.

You can also plot the laser field that you have used in this simulation: column number two of the file `laser` is the time, whereas column number three is the x component of the electric field.

- The total ionization is strongly dependent on the peak electric amplitude of the laser pulse. You can analyze this effect by performing various runs at varying values of this value.
- The polarization of the laser pulse is also relevant: how does this aspect affect the total ionization for this system? The effect of this parameter will be specially visible in the evolution of the dipole of the system. You can check this by looking at the `multipoles` file.
- Are we well converged with respect to the simulation box? This is especially difficult in cases where one studies ionization (in fact, it is impossible to obtain perfect convergence).

3 Photo-dissocation – or not.

In the previous calculation, the nuclei were frozen at their equilibrium position. In reality, they should move. To simulate this, we will utilize the Ehrenfest model, which is simply turned on by setting `MoveIons = yes`. The goals of this section are the following:

- The `inp.4` files contains the specification of a *non-resonant* calculation: the frequency of the laser field is not tuned to any of the excitation energies of the system (transition from the ground state to any of the excited states). What are the effects of this pulse on the system? Specifically, does it lead to the ionization of the system or not? And finally, how is the movement of the two nuclei? You may look at it in the `coordinates` file.
- Tune the laser frequency to the first resonance of the system, and check now whether or not the system dissociates. Is this dissociation accompanied of simultaneous ionization of the system?
- What happens if you keep the resonant frequency, but change the polarization direction of the laser field?

References

[1] U. Saalmann, and R. Schmidt, Z. Phys. D **38**, 153 (1996).

[2] E. K. U. Gross, J. F. Dobson, and M. Petersilka, in "Density Functional Theory" (Topics in Current Chemistry **181**), edited by R. F. Nalewajski, p. 81 (Springer-Verlag, Berlin-Heidelberg, 1996).

[3] T. Kunert, and R. Schmidt, Phys. Rev. Lett. **86**, 5258 (2001).

[4] C. Brif, R. Chakrabarti, and H. Rabitz, New J. Phys. **12**, 075008 (2010).

[5] A. Castro, M. A. L. Marques, J. A. Alonso, G. F. Bertsch, and Angel Rubio, Eur. Phys. J. D **28**, 211 (2004).

[6] A. Castro, M. A. L. Marques, and A. Rubio, J. Chem. Phys. **121**, 3425 (2004).