

# Session 3

## High intensity pulses: High harmonic generation, and electron photo-emission

### 1 High harmonic generation

The goal of this exercise is to reproduce Fig. 1 in Ref. [1]. Since you are now an expert in the use of octopus, we will not give you input files, but only some instructions. The first step is of course to read that reference, at least the first part (since we are not interested in optimization, we just want to learn how to compute the HHG spectrum of some atom or molecule).

Therefore, some hints:

- You need to build a model in 1D: use the variable “Dimensions”.
- The model is the one given in Eq. (13), for  $a = 1$ . You need to define it such a potential with the “Species” block, i.e.:

```
%Species
"Hydrogen1D" | species_user_defined | potential_formula | "-1/sqrt(1+(x)^2)" | valence | 1
%
```

- You must use absorbing boundaries when doing the td runs. (see “AbsorbingBoundaries”, “ABWidth”, and “ABHeight” variables). By looking at the total charge in the td.general/multipoles file, you will learn how much ionization you get. For this exercise it should be as small as possible.
- Be sure to find good values for the grid spacing, the simulation box, and later for the time-dependent discretization. A converged result can be obtained with a calculation of a few minutes, but only if those parameters are chosen wisely.

Remember that the smaller the grid spacing you use, the smaller the td time step you will need [Question: why?]

- Define the laser through the “TDExternalFields” and “TDFunctions” blocks. Be sure to choose the right amplitude for the field, so that it matches the one given in the paper (corresponding to a peak ponderomotive energy of

0.069 a.u., or  $3.158523 \cdot 10^{13}$  W/cm<sup>2</sup>. The output of the code gives these values, which may help you to figure this out.

The “shape” or “envelope” (the function given in the “TDFunctions” block) of the pulse should also match the one described in the paper if you want to get identical results. In order to see how your pulse looks like, octopus plots it when a td calculation is started, in the td.general/laser file (as long as you have added the “laser” keyword to the “TDOuput” variable).

- Once the td run is done, you must compute the HHG spectrum; for that purpose, you should use the “oct-harmonic-spectrum” utility. This program essentially computes Eq. (1). It can be done in two different manners:
  1. Reading the td dipole moment from the “multipoles” file, and taking the second time derivative numerically from it. This is the default mode, you just need to launch “oct-harmonic-spectrum” and it will do this. It will generate a file called “hs-mult.x” with the spectrum.
  2. If you run the td calculations with octopus adding the “dipole\_acceleration” keyword to the “TDOuput” variable (for example, “TDOuput = laser + multipoles + dipole\_acceleration”, octopus will generate an “acceleration” file in the “td.general” directory, with the acceleration of the dipole as given in Eq. (2). Then you can run “oct-harmonic-spectrum” with the “-m 2” option, and it will compute Eq. (1) using that, and generating a “hs-acc.x” file with the spectrum. Mathematically, this is equivalent to the former method; numerically they may differ.
- There are some variables related to the “oct-propagation-spectrum” that you may set, mainly “PropagationSpectrumMaxEnergy”, and “PropagationSpectrumEnergyStep”, of easy meaning: the spectrum will be plotted up to the maximum energy given by the former, and at intervals of energy given by the latter. It is good, numerically, to set the former to some multiple of the laser carrier frequency, and to set the latter at the Fourier transform base frequency  $2\pi/T$ , where  $T$  is the total propagation time.

“But this is not TDDFT because there is only one electron!” Yes, you are right. But once you have learnt to do this, it is straightforward to add one or more electrons, and get TDDFT HHG spectra (for example, the Helium case in the same paper).

## 2 Photo-emission

Now you will learn the basics of computing photo-emission spectra from finite systems with TDDFT. In fact, we will once again use a one-electron model system to make sure that calculations can be done in one afternoon, but the

ideas can be extended to larger systems, and you may try to set up calculations for those when you finish.

1. The first step is of course to understand the theory: you need to read Ref. [2], where most of the details behind this exercise are explained. In a nutshell, the idea will be to perform photo-emission calculations for a 2D atomic model, by using the “mask method” implemented in the octopus code. With this method the electrons absorbed at the edge of the simulation box are collected and propagated as plane-waves outside the simulation box. The technique allows to calculate  $P(\mathbf{k})$ , the probability to observe an electron with a given momentum  $\mathbf{k}$  (which is a vector), at a detector placed at infinite distance from the system. From this quantity, less resolved probabilities such as the energy-resolved photoelectron spectrum  $P(E)$  can be obtained by integration.
2. The next step is to define a model atom system to play with. We will work in 2D, and therefore we have to use the octopus option of setting “user-defined” external potentials to model systems. For example, an atom with a valence electron, that we will call “Sodium”, can be defined with:

```
%Species
"Na_2D" | species_user_defined | potential_formula | "-1/sqrt(7.45+(r)^2)" | valence | 1
%
```

This defines the “atom” through a potential:

$$v(r) = -\frac{1}{\sqrt{7.45 + r^2}}, \quad (1)$$

and it assumes there is only one valence electron, as in Sodium. The number “7.45” is somehow arbitrary, but it could be tuned to reproduce e.g. the ionization potential of Sodium.

Do ground state calculations with such potential to find out an appropriate grid spacing.

3. Next, let us perform some td calculations with a laser field. We will choose a laser pulse defined in the velocity gauge, i.e. through an external potential, with a cosinoidal envelope phase, i.e. something like:

$$A(t) = A_0 \cos\left(\frac{\pi}{2} \frac{t - t_0 - 2\tau_0}{\tau_0}\right) \cos(\omega t) \quad (2)$$

for  $|t - t_0| < \tau_0$  (otherwise  $A(t) = 0$ ). You will need to set up the “TDExternalFields” and “TDFunctions” accordingly. For a choice of the parameters, try  $\omega = 2.214088a.u.$ ,  $\tau = \frac{2\pi}{\omega}$ , and  $\tau_0 = t_0 = 10\tau$  (this is a XUV pulse centered at  $t_0 = 10\tau$  that has a duration of  $20\tau$ , i.e. starts at zero and finishes 20 oscillation periods later). You can plot a function like this with gnuplot, to see how it looks, and then you can compare with the output of octopus, to see if you defined the laser pulse properly.

Regarding the amplitude  $A_0$ , we will change it during the exercise. You can start with 0.5 a.u. [Question: how can one relate this value to the intensity of the pulse in  $\text{W}/\text{cm}^2$  as it is commonly used in the literature?]

You must then choose the total propagation time (that can be set to match the duration of the pulse), and the time step. This last parameter, as the grid spacing, must be small enough to get converged results, while at the same time not so small that the calculations are too long.

Set absorbing boundaries (see “AbsorbingBoundaries”, “ABWidth”, and “ABHeight” variables). Your task now is to get a feeling of how large you need the simulation box to be, by looking at how many electrons are ionized during the duration of the pulse. You may see how this depends on the intensity (change the amplitude  $A_0$ ), the length of the pulse (set  $\tau_0$  and  $t_0$  to different values), etc.

4. Time now to do photo-emission calculations. Set

```
PhotoElectronSpectrum = pes_mask
%PEMaskSize
inner_radius | outer_radius
%
PEMaskMode = mask_mode
```

where `inner_radius` and `outer_radius` must be chosen carefully. They are the parameters defining the regions A, B, and C in Fig. 2 of Ref. [2]. Set the outer radius to exactly match the radius of your simulation box, and the inner radius to something smaller.

The bigger your regions, the better the results, but the calculations may become unfeasible. Therefore, start playing with those values to find appropriate values for fast yet accurate calculations. The information you gained in the previous points about the ionization for different laser pulses, boxes, etc. will be a hint. Also, a necessary requirement is that you have to make sure that the ground state wavefunction is negligible in the mask region in order to avoid spurious absorption. This condition is however no sufficient to have converged results.

[Do not forget to eliminate the “AbsorbingBoundaries” you used in the previous point! The photo-emission mode uses its own absorption setup]

The variable `PEMaskMode = mask_mode` specifies that we want to use the simple mask method where the electrons can only move from real space to momentum space. The full method triggered by `fullmask_mode` allows electrons to pass from one space representation to the other.

After the time propagation we will find three new files in the directory `td.general`: `PESM_power.sum`, `PESM_map.z=0`, and `PESM_map.obf`. The first file contains the energy resolved spectrum  $P(E)$  the second contains the velocity map  $P(k_x, k_y, 0)$  calculated on a plane perpendicular to  $k_z$

and the last one contains the full  $P(\mathbf{k})$ . This last file is in binary format and must be post-processed by the appropriate utility.

5. Let us first take a look the energy resolved photoelectron probability. To this end we should plot the file `PESM_power.sum`.

- The spectrum should present a marked peak around  $E = 2$  au. Can you tell why photoelectrons comes out with this kinetic energy?  
*Hint:* try to change the laser frequency.
- The spectrum looks somehow *noisy*. This is due to the fact that `octopus` always calculates  $P(\mathbf{k})$  with  $\mathbf{k}$  distributed on a cartesian grid while  $P(E)$  should result from an integration in polar coordinates with  $E = k^2/2$ . The *noise* is a result of the different distributions of the volume elements between cartesian and polar grids. To solve this problem we can use the `oct-photoelectron-spectrum` utility which interpolates  $P(k)$  on a polar grid before performing the integration. The utility generates the file `PES_power.sum` in the root directory which contains a much smoother spectrum. The utility can do different kind of post-processing besides the generation of  $P(E)$ . You can have an idea by running the command "`oct-photoelectron-spectrum -help`". In the present two-dimensional case, however, its use is limited.
- We can look at the velocity map, i.e. the probability to have electrons emerging with a given velocity, by plotting `PESM_map.z=0`. This is a two-dimensional quantity that can be easily visualized with `gnuplot` as a density plot. Just use this commands in the `gnuplot` terminal:

```
gnuplot> set pm3d map
gnuplot> sp "td.general/PESM_map.z=0"
```

Can you understand why photoelectrons emerge with this angular distribution? *Hint:* try to change the laser polarization direction.

- Note: Since we are restricted to two dimensions `PESM_map.z=0` is equivalent to the full momentum-resolved probability. In three dimensions this is no longer the case and `oct-photoelectron-spectrum` can help the visualization of  $P(\mathbf{k})$ , for instance, by slicing it along different planes.
6. Let us now change the laser frequency and intensity to `omega = 1.0` and `amplitude = 15`. This amplitude corresponds to the choice of a laser with intensity  $I = 3.9 \times 10^{14}$  W/cm<sup>2</sup>. You can read a summary of the laser properties in the output of the code in the section "**Time-dependent external fields**". In this section it also gives you the value of the ponderomotive energy  $U_p$  that you can use to estimate the Keldish parameter  $\gamma = \sqrt{IP/2U_p}$  (IP is the ionization potential of our system) which tells us that we are deep in the multi-photon regime.

The photoelectron spectrum plotted in log-scale displays a series of peaks positioned at energies that correspond to the absorption of  $s$  photons according to  $E = s\omega - IP - U_p$ .

7. Since  $\omega > IP$  electrons are ionized by a single photon we cannot talk of above threshold ionization. To test ATI we have to choose lower values for  $\omega$ . For our next example we thus modify `omega = 0.057` (corresponding to 800 nm) and `amplitude = 20 a.u.` Electrons will now have to absorb at least 4 photons in order to be released into the continuum. You will see a series of peaks separated by  $\omega$  in the lower part of the spectrum.

If you now change the amplitude (say from 20 to 27 a.u.), you will see a shift in the peaks. Can you explain why?

8. For the “pros”: if you compile octopus with the NFFT library, you may improve the resolution of the plots by making use of the technique explained in Section 7 of Ref. [3], the “zero padding with NFFT”. With this approach we can compress the entire momentum space by a factor  $1/\alpha$ . To do that we have to tell octopus to use NFFT by setting `PESMaskPlaneWaveProjection = nfft_map` and specify the enlargement level `PESMask2PEnlargeFactor = 2` which means  $\alpha = 2^{lev}$ .

9. In all cases, you can use smaller boxes if you use the full mask-method setting `PESMaskMode = fullmask_mode`. This way electrons can go back and forth from real-space to momentum-space representation when the field changes sign. With this method you should be able to converge the angular distribution with a much smaller simulation box.

Try repeating some of the previous calculations with this method, and compare.

10. If you have been using a fixed length laser pulse all the time, perhaps you could investigate how results change if you use a larger number of cycles.

*Hint:* you should increase the simulation box together with the number of cycles to obtain stable results, because electrons may make longer excursions.

11. More about photo-emission with TDDFT and real-space real-time methods: Ref. [4].

## References

- [1] Alberto Castro, Angel Rubio, and E. K. U. Gross. Enhancing and controlling single-atom high-harmonic generation spectra: a time-dependent density-functional scheme. *Eur. Phys. J. B*, 88:191, 2015.
- [2] Umberto De Giovannini, Daniele Varsano, M A L Marques, H Appel, E K U Gross, and A Rubio. *Ab initio* angle- and energy-resolved photoelectron

spectroscopy with time-dependent density-functional theory. *Phys. Rev. A*, 85:062515, June 2012.

- [3] Xavier Andrade, David Strubbe, Umberto De Giovannini, Ask Hjorth Larsen, Micael J. T. Oliveira, Joseba Alberdi-Rodriguez, Alejandro Varas, Iris Theophilou, Nicole Helbig, Matthieu J. Verstraete, Lorenzo Stella, Fernando Nogueira, Alan Aspuru-Guzik, Alberto Castro, Miguel A. L. Marques, and Angel Rubio. Real-space grids and the octopus code as tools for the development of new simulation approaches for electronic systems. *Phys. Chem. Chem. Phys.*, 17:31371–31396, 2015.
- [4] Philipp Wopperer, Umberto De Giovannini, and Angel Rubio. Efficient and accurate modeling of electron photoemission in nanostructures with tddft. *arXiv preprint: 1608.02818v2*, 2016.