FYS-4096 Computational Physics, project 1 (max 70 XPs)

Read this entire document before you start working on your project.

Return your solution to project `project1` under your GitLab group for this course by March 26th 11:59 pm. **Late submissions will not be accepted.** ## 1. Introduction

This project will test your ability to

- write readable simulation and analysis code
- perform numerical experiments
- write a report describing the methods you've used and the numerical results you've obtained
- store and publish the results of your numerical experiments

Your grade for the project will be based on, **e.g.**, * the final report (understanding and description of the methods and experimental workflow, discussion of the findings, quality of figures, typography, . . .) * correctness of your implementation * readability of your code * **unit tests** (write your scripts in a modular way and provide unit tests for them) * **organization of your git repository** * git commits (amount, quality of messages) * your coding style * licensing

You are allowed to: * use and modify all the reference solutions for the exercises, especially the TDSE solver developed in `ex6`. The source code for the TDSE solver is available at ../ex6/solution and it can also be installed via pip: `pip install --user quantum_dynamics`. * use code from stackoverflow or internet provided that (1) you write comments explaining where the code is from and who wrote it, and (2) you adhere to their license agreement terms * use the Taito cluster for simulations * ask clarification to this problem sheet * ask hints regarding the project (answers not quaranteed)

2. Select your problem

Choose one of the following problems to work on.

I. Above-threshold ionization of 1D hydrogen

When an atom is exposed to a relatively weak laser pulse with carrier frequency ‘ω_0‘, the electron(s) interact with the field by absorbing one or more photons. Depending on the number of photons absorbed (‘n‘) and the ionization energy of the atom ‘I‘, we expect high-probability of measuring ionized electrons with kinetic energies

‘$E_{\mathrm{kin}} = n\omega_0 - I,$‘

i.e., the kinetic energies should be peaked with intervals ‘ω_0‘. In our model, the ionization energy is the negative of the ground state energy.

Your task is to verify this phenomenological model by conducting a numerical experiment as follows:

1. Set up a laser pulse with carrier frequency ‘$\omega_0 = 0.2$‘ a.u. (corresponding to ‘ ≈ 230‘ nm), electric field amplitude ‘$\epsilon_{\text{max}} = 0.06$‘ a.u., and duration ‘$T_{\text{max}} = 36$‘ fs.
2. Calculate a time-dependent quantum simulation with the software from `exercise 6` and save the wave-function ‘$\psi(x, t)$‘. You should use the reference implementation since it also saves the field-free Hamiltonian ‘$\mathbf{H}_0$‘.
3. Write a script that reads the datafile from step 2, and calculates the probability distribution for the energy of the ionized electron from ‘$P(E) = \gamma^4 \langle \psi(t = T_{\text{max}}) \mid \left[(\mathbf{H}_0 - E\mathbb{I})^4 + \gamma^4 \mathbb{I} \right]^{-1} \mid \psi(t = T_{\text{max}}) \rangle,$‘ where ‘$E$‘ is the energy we are interested in, ‘$\mathbf{H}_0$‘ is the time-independent Hamiltonian (*without* the complex absorbing potential), and ‘γ‘ is a small number, typically half the energy spacing of our analysis, and ‘$\mathbb{I}$‘ is the identity matrix.
4. Run a set of simulations with different time-steps, grid sizes, energy spacing, etc. until you get converged results.
5. Visualize ‘$P(E)$‘
6. Discuss your findings in final_report.pdf.

Note: The reference solution for `ex6` saves the field-free Hamiltonian ‘$\mathbf{H}_0$‘ to file, and you can load it like

```
from quantum_dynamics.utils import load_sparse_matrix
import h5py

f = h5py.File('myfile.h5', 'r')
H0 = load_sparse_matrix(f['tise_hamiltonian'])
```

II. High-harmonic generation with 1D hydrogen

When an atom is exposed to a strong ultrashort laser pulse with carrier frequency ‘ω_0‘, the electrons interact with the field by tunneling out of the atom. In addition, the electron density oscillates around the atomic nucleus forming an oscillating dipole.

This dipole emits radiation with a distinctive cut-off energy predicted at ‘$E_{\text{cutoff}} = I + 3.17 U_p$‘ where ‘$I$‘ is the ionization energy of the atom (in our model the ionization energy is the negative of the ground state energy), and you can estimate the ponderomotive energy ‘U_p‘ as ‘$U_p \approx \left(\frac{\epsilon_{\text{max}}}{2\omega_0} \right)^2$‘. Here ‘$\epsilon_{\text{max}}$‘ is the maximum laser electric field amplitude and ‘ω_0‘ is the central frequency of the laser pulse.

From a quantum simulation, the dipole acceleration can be calculated as

‘$\ddot{d}(t) \propto \int \psi^*(x,t) \frac{-\partial V}{\partial x} \psi(x,t)\,\mathrm{d}x,$ ‘

where ‘$V(x)$‘ is the atomic potential.

From the dipole acceleration we can calculate the energy spectrum of the emitted radiation as

‘$S(E) \propto |FFT[\ddot{d}(t), t \to E]|^2/E^2.$‘

Your task is the verify the cutoff-energy formula with the following numerical experiment:

1. Set up a laser electric field with central frequency ‘$\omega_0 = 0.0569$‘ a.u. (corresponding to 800 nm), electric field amplitude ‘$\epsilon_{\mathrm{max}} = 0.1$‘ a.u., and duration ‘$T_{\mathrm{max}} = 25$‘ fs.
2. Calculate a time-dependent quantum simulation with the code from exercise 6 and save the wave-function ‘$\psi(x,t)$‘.
3. Write a script that reads the datafile from step 2, and calculates the energy spectrum of the emitted radiation.
4. Visualize ‘$S(E)$‘.
5. Discuss your findings in final_report.pdf

III. Wave-packet tunneling through 1D barrier

A typical problem in an introductory quantum mechanics course is the tunneling of an electron wave-packet through a rectangular potential barrier in 1D. Your task is to numerically investigate how the tunneling probability ‘T‘ depends on the barrier characteristics and the properties of the initial wave-packet. In contrast to previous two problems, there is no driving laser field in this setup.

The experiment setup is as follows:

The initial state is a wave-packet with momentum ‘p‘, i.e.,

‘$\psi(x, t=0) = \frac{1}{\sqrt[4]{2\pi\sigma^2}} \exp[-(x-x_0)^2/(4\sigma^2)] \exp(\mathrm{i}px),$ ‘

where ‘x_0‘ is the initial position, ‘σ‘ the initial wave-packet width, and ‘$p = \sqrt{2E_{\mathrm{kin}}}$‘ the initial momentum. **Use ‘$\sigma = 10$‘ in your simulations.**

A smooth potential barrier is given by

‘$V(x) = \frac{V_0}{1+\exp[(|x|-r)/w]},$ ‘

where ‘V_0‘ is the barrier height, ‘r‘ half the barrier width, and ‘w‘ the barrier softness. **Use ‘$V_0 = 1$‘ in your simulations.** The initial setup is illustrated in the following figure.

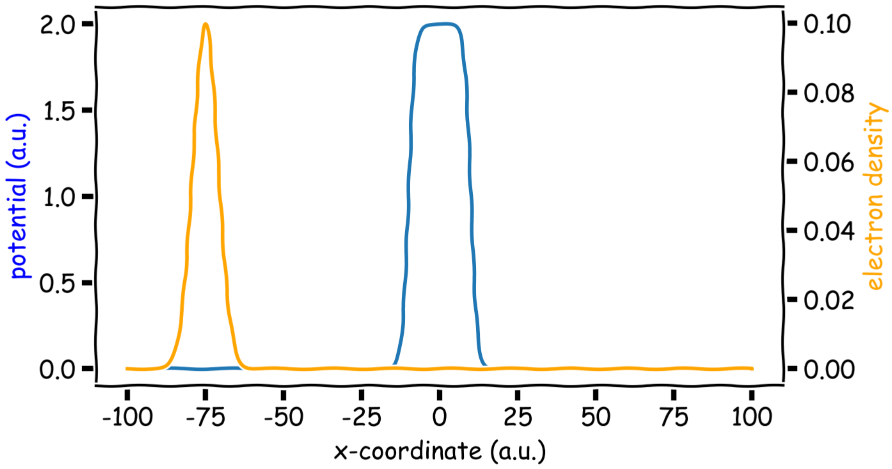

Figure 1: barrier

How to conduct the experiment (using the code developed in `exercise6`)

1. Set up the initial state as the wave-packet described above. This is not an eigenstate of the Hamiltonian so you need to calculate the discretized initial wavefunction manually. Set up the potential from the above expression.

2. Run the simulation. Note that you need a suitable grid length and simulation time so that (1) the electron density is not absorbed by the boundaries or reflected from them and

(2) the electron wavepacket has enough time to reach and tunnel through the barrier.

3. Write a script that evaluates the tunneling probability, i.e., the amount of electron density that has passed the barrier at the end of the simulation, ‘$T = \int\limits_r^\infty \mathrm{d}x\, |\psi(x,T)|^2$‘ (if you set up the wavepacket initially at ‘$x < 0$‘)

4. Investigate and visualize the dependence of the tunneling probability ‘T‘ on the parameters ‘E_{kin}, r‘, and ‘w‘.

5. Discuss your findings in final_report.pdf

3. Final report

The final report should be relatively short (~ a few pages), but it should contain

- description of the numerical methods used (including the code for quantum mechanics)
- description of the workflow/setup of the numerical experiment including illustrative figures where appropriate
- description of the convergence checking you've conducted
- your results
- discussion of your results

A reader should be able to *easily* reproduce your numerical experiment based on the information you provide in the final report and scripts/README-files you provide in your git repository.

You can write your final report with any tool you like, but LaTeX is naturally preferred. **Include all the source files and the final pdf of the report in your git repository.**

4. Submitting to Zenodo, optional (extra + 10 XPs)

Finally, you can submit your work (including the code and the final report) to an EU-funded public data-repository called Zenodo. This will give your work a Digital Object Identifier (DOI). You should login to Zenodo with your ORCID credentials (create ORCID if you don't have it).

Please note that submissions to Zenodo cannot be removed. This means that your submission will stay in Zenodo for all eternity. While submissions to Zenodo can't be removed, you can (if you wish) restrict the access level of the submission to `closed` so no-one can access the data without your approval.

Make a note to `README.rst` and final_report.pdf of the DOI you get from Zenodo, commit, and tag the commit with `final`.

Do not submit the *entire* git repository, but only the final version of the files. You can get an archive of the files in your git repository with the command

```
$ git archive master | bzip2 > project_final.tar.bz2
```

Note that it's possible (and preferred) to get the DOI from Zenodo, then add it to `README.rst`, commit and tag with `final`. Then you can submit the tagged final version to Zenodo.

Make sure that the files you upload to Zenodo do not contain any sensitive information such as you student number.

You should upload via https://zenodo.org/deposit/new?c=tampere_university_computational_physics_101 to ensure that the submission ends up in the course's community within Zenodo.

5. Pushing your Git repository to GitLab

Return your solution to project `project1` under your GitLab group for this course by March 19th 11:59 pm. **Late submissions will not be accepted.**

Tag the final version with `final` keyword.

6. Hints

- Try visualizing with linear and logarithmic scales for the axis
- You can clone the reference code for `ex6` via `git clone https://gitlab.com/solanpaa/quantum_dynami`
- You can **install** the reference code for `ex6` via `pip3 install --user quantum_dynamics`