FYS-4096 Computational Physics, project 2 (max 70 XPs)

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

Return your solution to project `project2` under your GitLab group for this course by May 14th 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 if applicable) * **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 and examples from the lecture slides * use code from stackoverflow (or anywhere on the 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 (available till May 14th) * 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.

1. Nanoplasmonics: Plasmon response of gold nanoparticles in the quasi-static limit

A nanoparticle whose size is in the nanometer scale changes locally the characteristics of an incoming laser electric field. Near the nanostructure, the electric field is significantly enhanced due to plasmonic effects, i.e., the electrons of the nanoparticle reposition themselves in such a way that they increase the local electric field at some places around the nanoparticle.

When the nanoparticle size is a lot less than the wavelength of the laser electric field, we can model the system as **quasi-static**, i.e., we decouple the spatial and temporal part of the laser electric field near the nanostructure:

‘$\overline{\mathbf{E}}(\bar{\mathbf{r}}, t) = f(t)\overline{\mathbf{E}}(\bar{\mathbf{r}})$‘

In the quasistatic approximation the spatial part of the electric field ‘$\overline{\mathbf{E}}(\bar{\mathbf{r}})$‘ can be obtained from a time-independent scalar potential ‘$V(\bar{\mathbf{r}})$‘ via ‘$\overline{\mathbf{E}}(\bar{\mathbf{r}}) = -\nabla V(\bar{\mathbf{r}})$‘. The scalar potential obeys the macroscopic Gauss' law

‘$-\nabla \cdot [\epsilon(\mathbf{r})\nabla V(\mathbf{r})] = 0$‘

both inside the nanoparticle ‘Ω_{NP}‘ and outside ‘Ω_{vacuum}‘. Here ‘$\epsilon(\mathbf{r})$‘ is the complex relative permittivity of the material at position ‘$\mathbf{r}$‘. You can obtain its values from Refractive index database; there ‘ϵ_1‘ is the real part and ‘ϵ_2‘ the imaginary part.

At the nanoparticle surface, the interface condition for the electic displacement field yields

‘$[-\epsilon_{\text{vacuum}}\nabla V_{\text{vacuum}} + \epsilon_{NP}\nabla V_{NP}] \cdot n_{NP\to\text{vacuum}} = 0,$‘

where ‘$\epsilon_{\text{vacuum}/NP}$‘ are the relative permittivities of vacuum and the nanoparticle for the incoming laser's wavelength.

Due to complex relative permittivity, the scalar potential ‘V‘ becomes a *complex valued quantity*, where the *real* part corresponds to measured values. This means that also your test and trial functions become complex quantities.

Far from the nanostructure, the boundary conditions should give you the desired electric field polarized, e.g., in the ‘z‘-direction, i.e., ‘$\overline{\mathbf{E}}(\bar{\mathbf{r}}) = E_0\hat{\mathbf{z}}$‘.

Your job is to calculate and visualize the electric field near a gold nanosphere of radius 50 nm irradiated by a 800 nm laser with peak electric field (without the nanosphere) ‘$1\,\mathrm{V/nm}$‘. **Solve the PDE numerically!** What is the enhancement factor (‘$\max(E/E_0)$‘) of the nanosphere at this wavelength?

To run *FEniCS* on Taito, you can use my (temporary) installation by

```
$ export SOLANPAA_USERAPPL=/homeappl/home/solanpaa/appl_taito
$ module use $SOLANPAA_USERAPPL/modules/
$ ml purge
$ ml solanpaa fenics
$ python3 my_fenics_script.py
```

Please note that this installation doesn't include working PETSc, SLEPc nor MSHR.

2. Quantum control 101

In the following, all quantities are in atomic units.

Quantum Optimal Control Theory is a framework for designing laser pulses that drive a quantum system to a predefined goal. In this project you'll consider again the 1D hydrogen of exercises 4 - 7 and project 1, but this time we want to optimize a certain transition in the system.

The electron starts in its ground-state, $\psi(x, t = 0) = \psi_{\mathrm{GS}}(x)$ and interacts with a laser pulse parametrized as

$\epsilon(t) = E_{\max} \sin^2\left(\frac{\pi}{T} t\right) \left[A \cos(\omega_A t + \phi_A) + B \cos(\omega_B t + \phi_B)\right] \Theta(t)\Theta(T - t),$

where $\omega_A = 0.07$, $\omega_B = 0.03$, and $T = 1033$ are fixed. Adjustable parameters are $E_{\max}$, A, B, ϕ_A, and ϕ_B.

Your job is to find a set of parameters $E_{\max}$, A, B, ϕ_A, and ϕ_B that maximizes the population of the 1st excited state at the end of the laser pulse.

1. Use the example TDSE code developed earlier (link in project 1)
2. Write a function that (A) takes $E_{\max}$, A, B, ϕ_A and ϕ_B as input parameters, and (B) returns $|\langle\psi_1(x)|\psi(x, t = T)\rangle|^2$
3. Write a script that passes the function to a **gradient-free** optimization routine (e.g., PRAXIS) to obtain the optimal parameters.

Constrain the variables so that $E_{\max} \leq 0.1$, $\phi_A \in [0, 2\pi]$, $\phi_B \in [0, 2\pi]$, and $A + B = 1$.

3. Your own problem

- Must get an approval before starting to work on your project.
- Is either implementation of a numerical method or a computational study of a phenomenon

3. Final report

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

- description of the numerical methods used
- 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 + 5 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 your 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 `project2` under your GitLab group for this course.

Tag the final version with `final` keyword.