FYS-4096 Computational Physics: Exercise 3

Ilkka Kylänpää

Week 4

General:

- You should have access to help files on teacher's repo. If not, please provide your username and associated email to the teacher.
- There, in folder `ex3_help`, you will find files `linear_interp.py`, `spline_class.py`, and `matrix_eigs.py` that are needed or can be useful in these exercises.
- **Deadline always before noon on Monday the following week!** That is, for these problems before noon on Monday February 1.
- Better to send/push unfinished in time, than finished, but past deadline.

Problem 1: 2D integral revisited

- Calculate the integral below **using scipy's Simpson rule**:

$$\int_{-2}^{2}\int_{x_0}^{x_1} f(x,y)dxdy,$$

  where $f(x,y) = (x+y)^2\exp(-\sqrt{x^2+y^2})$ with $x_0 = 0$ and $x_1 = 2$.
- Note: `from scipy.integrate import simps`.
- Estimate the accuracy by using a couple of different grids.

Problem 2: 2D interpolation revisited

- Generate your "experimental data" using the function

$$f(x,y) = (x+y)\exp(-\sqrt{x^2+y^2})$$

  on a 30 by 30 linear grid with $x \in [-2,2]$ and $y \in [-2,2]$.
- Use the "experimental data" in estimating values along a path given by $y = 2x^2$, where $x \geq 0$. In the path there should be 100 uniformly spaced grid points for the x-values.
- Plot the 1D values from your interpolation as well as from the exact function. Use different line styles, legend, etc. to straightforwardly distinct the interpolated and exact solutions.

Problem 3: Power method

- Code a function called `largest_eig` and apply it to find the largest eigenvalue and corresponding eigenvector of the matrix given in `matrix_eigs.py`
- Compare the eigenvalue and the eigenvector with scipy's solver (provided in the example file).

Extra If time permits after problem 4 (on next page), read about inverse power method as well as Arnoldi iteration from, e.g., wikipedia.

Problem 4: Electric field

- Consider a charged 1D rod that is on x-axis from $-L/2$ to $L/2$, where L is the length of the rod.
- A positive charge Q is evenly distributed along the rod, and thus, its line charge density is given as $\lambda = Q/L$.
- The differential electric field at any point in space is given as

$$d\mathbf{E} = \frac{1}{4\pi\varepsilon_0}\frac{\lambda dx}{r^2}\hat{\mathbf{r}}, \tag{1}$$

  where r is the distance from a point (i.e., charge element) in the rod to a point in space, and $\hat{\mathbf{r}}$ is a corresponding unit vector.

- Since the net field is given by integration, i.e., $\mathbf{E} = \int d\mathbf{E}$, **use Simpson rule for the integration**.
- First, implement a code that calculates the electric field by the charged rod at any point in space. Test your numerical implementation against the electric field at $\mathbf{r} = (L/2 + d, 0) = (x, y)$ for which the analytical answer is

$$\mathbf{E} = \frac{\lambda}{4\pi\varepsilon_0}\left[\frac{1}{d} - \frac{1}{d+L}\right]\hat{\mathbf{i}}. \tag{2}$$

- Second, calculate numerically the electric field in xy-plane, and visualize your vector field result with matplotlib's quiver function (or something similar). (Notes: Avoid $y = 0$ in your grid. A 20 by 20 grid should be good enough.)

Returning your exercise

1. Make a new folder called "exercise3" into your existing Computational Physics repo.
2. Create a file solvedProblems.txt at the root of your "exercise3" folder. Inside it, write a comma separated list of problems you have solved, e.g., 1,2,3x.
3. Make sure all your source files are under version control and push them to GitLab.
4. Push your commits (and possible tags) to GitLab before noon on Monday February 1:
   ```
   git push --all && git push --tags
   ```