Programming Project – 2021-08-24

MATB22: Linear Algebra

C. Führer, L. Versbach, N. Kotarsky

This assignment has 3 tasks.

The goal of this project is to gain a better understanding of theoretical concepts with the help of practical implementations. This requires both programming skills and theoretical knowledge. **The main computations have to be done using Python!** Make sure to study the documentations of any Python functions we ask you to use and that you understand the mathematics of the problem before you start programming. It is recommended to start with each task after the theoretical material is covered in the lectures, but it is even possible to start programming parts before that (task 3 can be solved without the theoretical material covered).

The projects follows the line of a previous exam in the course MATB22 (Aug, 2017).

*For this project you should work in **groups of two, three or four**. Solve the project tasks during the course and upload your group's code as a single file having one of the file types* `*.py` *or* `*.ipynb` *in Canvas.*

***Deadline: Monday, 18/10/2021**. Appointments for oral presentations will be scheduled via Canvas, pay attention to any announcements. This group presentation is mandatory.*

All questions and discussion with regards to the tasks should be done using Canvas. Due to the current circumstances, we will have all contact using zoom (questions as well as the oral presentations).

Task 1 - Least squares, normal equation, minimization

This is related to Task 1 in the above mentioned exam:

1. Find the vector $\boldsymbol{x} \in \mathbb{R}^3$ that minimises $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|$ where

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}.$$

- Solve this task by setting up and solving the normal equations in Python.

- Use `scipy.optimize.fmin` to directly solve the minimization problem. Compare results.
- Replace $\boldsymbol{b}$ by $\boldsymbol{b}(a) = (1, a, 1, a)^{\mathrm{T}}$ and plot the residual for each $r(a) = \|\boldsymbol{A}\boldsymbol{x}(a) - \boldsymbol{b}(a)\|$ versus a. Note the dependence of $\boldsymbol{x}$ on a in this subtask.
- Does this curve have a zero? If yes, what does this mean in the context of solving over-determined linear systems?

Task 2 - Eigenvalues, eigenvectors, recurrence relations

This is related to Task 2 in the above mentioned exam:

$$\begin{cases} a_{n+1} = a_n + 3b_n + 2c_n \\ b_{n+1} = -3a_n + 4b_n + 3c_n \\ c_{n+1} = 2a_n + 3b_n + c_n \end{cases}, \qquad \begin{cases} a_0 = 8 \\ b_0 = 3 \\ c_0 = 12 \end{cases}.$$

- Do the iterates $\boldsymbol{z}_n = [a_n, b_n, c_n]$ converge for $n \to \infty$? If so, determine the limit $\boldsymbol{z}$.
- Do the normalized iterates $\boldsymbol{v}_n := \boldsymbol{z}_n / \|\boldsymbol{z}_n\|$ converge[1]? If so, determine the limit $\boldsymbol{v}$.
- Which property do the limits $\boldsymbol{z}$ and $\boldsymbol{v}$ have (if they exist)? Hint: It might be helpful to investigate how you would solve this on paper.
- Compute for every normalized iterate the quantity $q_n = \boldsymbol{v}_n^{\mathrm{T}} \boldsymbol{A} \boldsymbol{v}_n$ ($\boldsymbol{A}$ defined by the above problem). Determine the limit q.
- What quantity does q represent with respect to the matrix $\boldsymbol{A}$?
- How many iterates do you need to fulfil $\|\boldsymbol{v}_n - \boldsymbol{v}\| < \varepsilon$ for $\varepsilon = 10^{-8}$?
- Vary ε between 0.1 and 10^{-14}, plot the number of iterates versus ε in a semilog plot. Do the same for q_n and its limit q. Which sequence converges faster?

Task 3 - Quadratic curves and surfaces

This is related to Task 3 in the above mentioned exam:

A surface has, with respect to an orthonormal coordinate system for 3-space, the equation

$$2x_1^2 - x_2^2 + 2x_3^2 - 10x_1x_2 - 4x_1x_3 + 10x_2x_3 = 1.$$

Identify its type and specify the points on the surface closest to the origin.

[1] Normalize in every step.

Make a 3D plot of this surface for $x_1, x_2 \in [-1, 1]$ without simplifying the above equation by hand. There are two possible options a) and b) to solve this task. Pick one of them.

a) Determine the solutions of x_3 in dependence of x_1 and x_2 using `sympy`.

b)
- Without computing them, how many solutions x_3 are there for a given x_1, x_2? What type of equation is this?
- Write a function to determine a solution x_3 for a given x_1 and x_2 using `scipy.optimize.fsolve`. You will need to find **suitable initial values** for `fsolve` to find all solutions.