Programming Project – 2021-08-24

MATB21: Analysis in several variables 1

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 2 can be solved without the theoretical material covered).

The project follows the line of a previous exam in the course MATB21 (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: Tuesday, 19/10/2021**. 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 - Integration, Curves, Surface integrals, directional derivatives

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

Compute the length of the plane curve $\gamma : (x, y) = (t^2, t^3), -2 \leq t \leq 1$.

- Write your own python function `Rie_sum(f, a, b, n)` to compute a Riemann sum for calculating integrals of functions in one variable.
- Verify this python function: Take a simple function `f` and show that the difference of your function and the exact solution tends to zero for $n \to \infty$.

- Use `Rie_sum` to solve the above exam question. Compare your result with `scipy.integrate.quad`.

Task 2 - Optimization, Gradients

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

Determine local extreme values for the function $f(x, y) = 8xy - 4x^2y - 2xy^2 + x^2y^2$.

- **Graphically** determine local extreme values using a contour plot.
- Numerically compute local extreme values using `scipy.optimize.fmin`. Hint: Maxima of f are minima of $-f$.
- `fmin` is an iterative method. Trace the iterates in your contour plot. Hint: Check the documentation of `fmin`, some optional parameters might help here.

Task 3 Implicit functions, curves, surfaces

This is related to Task 6 in the above mentioned exam:

Show that the equation $x + 2y + z + e^{2z} = 1$ *has a smooth solution* $z = z(x, y)$ *defined in a neighbourhood of the origin* $x = y = z = 0$. *Find the Taylor polynomial of degree 2 of the function* $z(x, y)$ *about the point* $x = y = 0$.

- Illustrate the smoothness of $z(x, y)$ in a neighborhood of the origin using a 3D plot. Hint: For a given x and y you can solve the equation for z using `scipy.optimize.fsolve`.
- Determine the coefficients of the Taylor polynomial $P_2(x, y)$ of degree 2 of $z(x, y)$ about the point $x = y = 0$ using **numerical differentiation**. Hint: Check the Wikipedia article on numerical differentiation, which also helps on how to pick suitable h. **Do not compute the coefficients by hand!**
- Make a 3D plot of $P_2(x, y)$.
- Plot the absolute error $e(x, y) := |z(x, y) - P_2(x, y)|$. Is the error reasonable given your knowledge on Taylor approximations?