Assignment 1

Nonlinear Optimization WS2024/25

Alexander Falk, falk@tugraz.at
Clemens Krenn, clemens.krenn@student.tugraz.at
Stefanie Stoppacher, stoppacher@student.tugraz.at

October 22, 2024

Deadline: Nov 12^{th}, 2024 at 16:00

Submission: Upload your report and your implementation to the TeachCenter. Please use the provided framework-file for your implementation. Make sure that the total size of your submission does not exceed 50MB. Include **all** of the following files in your submission:

- `report.pdf`: This file includes your notes for the individual tasks. Keep in mind that we must be able to follow your *calculation process*. Thus, it is not sufficient to only present the final results. Please submit your findings in a compiled LaTeX document.
- `main.py`: This file includes your `python` code to solve the different tasks. Please only change the marked code sections. Also please follow the instructions defined in `main.py`.
- `figures.pdf`: This file is generated by running `main.py`. It includes a plot of all mandatory figures on separate pdf pages. Hence, you do not have to embed the plots in your report.

1 Characterization of Functions (8 P.)

Given $\mathbf{x} = (x_1\ x_2)^\top \in \mathbb{R}^2$ and

a) $f(\mathbf{x}) = (\mathbf{a}^\top \mathbf{x} - d)^2$ where $\mathbf{a} = (-1\ 3)^\top$, $d = 2.5$

b) $f(\mathbf{x}) = (x_1 - 2)^2 + x_1 x_2^2 - 2$

c) $f(\mathbf{x}) = x_1^2 + x_1 \|\mathbf{x}\|^2 + \|\mathbf{x}\|^2$

d) $f(\mathbf{x}) = \alpha x_1^2 - 2x_1 + \beta x_2^2$

and let $\|\cdot\|$ denote the ℓ_2-norm. For each of the given functions do:

In Python:

1. Plot the contour sets of the above functions using a contour plot[1]. For d) use α, β of your choice.
2. Add markers to the stationary points (computed with Pen & Paper).

With Pen & Paper:

3. Compute the gradient and the Hessian.
4. Determine the set of stationary points.
5. For a) to c) characterize every stationary point whether it is a saddle point, (strict) local/global minimum or maximum.
6. For d) denote the intervals for α and β for which minima/maxima and saddle points are attained.

[1] `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.contour.html`

2 Matrix Calculus (7.5 P.)

Given $\mathbf{x} \in \mathbb{R}^n$

a) $f(\mathbf{x}) = \frac{1}{4}\|\mathbf{x} - \mathbf{b}\|^4$ for $\mathbf{b} \in \mathbb{R}^n$

b) $f(\mathbf{x}) = \sum_{i=1}^{n} g\left((\mathbf{A}\mathbf{x})_i\right)$ with $g(z) = \frac{1}{2}z^2 + z$ for $z \in \mathbb{R}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, subscript i denoting the i-th element

c) $f(\mathbf{x}) = (\mathbf{x} \oslash \mathbf{b})^\top \mathbf{D}(\mathbf{x} \oslash \mathbf{b})$ for $\mathbf{b} \in \mathbb{R}^n, \mathbf{D} \in \mathbb{R}^{n \times n}$

The operator $\oslash$ denotes the Hadamard-Division which is the element-wise division of two vectors or matrices e.g.

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \oslash \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} \frac{a_{11}}{b_{11}} & \frac{a_{12}}{b_{12}} \\ \frac{a_{21}}{b_{21}} & \frac{a_{22}}{b_{22}} \end{pmatrix}.$$

For each of the given functions:

With Pen & Paper:

1. Compute the gradient of $f(\mathbf{x})$. Compute the derivative for each element using the summation formulation as shown in the exercise lecture. Convert your result back to multivariate notation.
2. Compute the Hessian of $f(\mathbf{x})$. Proceed similarly to the gradient computation step. Show all your steps in the report.

3 Numerical Gradient Verification (4.5 P.)

To see whether the computed gradient is correct one can easily verify this by computing a numerical approximation of the gradient. For the case that $\mathbf{x} \in \mathbb{R}^2$ this can be achieved using central differences

$$\nabla f(\mathbf{x}) \approx \frac{1}{2\epsilon} \begin{pmatrix} f((x_1 + \epsilon,\ x_2)^\top) - f((x_1 - \epsilon,\ x_2)^\top) \\ f((x_1,\ x_2 + \epsilon)^\top) - f((x_1,\ x_2 - \epsilon)^\top) \end{pmatrix}$$

In Python:

1. Write a function to check the gradients for all the functions in Task 1 numerically as shown above. To do so, choose a random point for $\mathbf{x}$. For this point compare the result of your analytically computed gradient with the numerically approximation for different levels of $\epsilon \in [1 \times 10^{-7}, 1]$. Visualize the approximation error for the different levels of ϵ in terms of the ℓ_1, ℓ_2, and ℓ_∞ norm. For the plot, use a logarithmic [2] scale on the abscissa [3]. Use the provided functions and variables to compute the gradient, function value and gradient approximation.
2. In a general setting for $\mathbf{x} \in \mathbb{R}^n$ choose $n = 5$ and again perform the verification but for your results from Task 2. This time use scipy's `approx_fprime` [4] function. Again, visualize the approximation error for the different ϵ-levels in terms of the ℓ_1, ℓ_2, and ℓ_∞ norm. Use the provided functions and variables to compute the gradient, function value and gradient approximation.
3. Comment on your findings. What would be a good way to determine ϵ?

4 The Diet Problem (5 P.)

As the dietitian for a boarding school, your task is to ensure that students receive balanced nutrition while minimizing costs. To simplify the process of selecting from the available food options (see Tab. 1), you decide to frame the problem as a linear program (LP). The goal is to provide meals that meet the daily nutritional needs of an average person, including an energy intake of (8000 ± 150)kJ, with lipids ranging from (35 to 70)g, carbohydrates from (180 to 300)g, and proteins from (40 to 160)g. Additionally, proper fiber intake of at least 30g is crucial for gut health, while a minimum of 80mg of vitamin C is necessary to support the immune system. At the same time, salt intake should be limited to a maximum of 3000mg to avoid health risks associated with excessive amounts. By formulating this diet problem as an LP, you can determine the most cost-effective combination of foods that satisfies these nutritional requirements.

[2] `https://numpy.org/devdocs/reference/generated/numpy.logspace.html`
[3] `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.semilogx.html`
[4] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.approx_fprime.html`

Table 1: Nutritional data for servings of 100g.

Food [100g]	Cost [€]	Energy [kJ]	Lipids [g]	Carbs [g]	Proteins [g]	Fiber [g]	Vit. C [mg]	NaCl [mg]
Milk	0.2	266	3.5	4.6	3.4	–	13.7	130
Tomatoes	0.6	80	–	3.9	–	1.2	13.7	–
Bananas	0.25	373	–	23	1.1	2.6	8.7	–
Apples	0.3	217	–	14	–	2.4	4.6	–
Noodles	0.3	1521	2	71	13	3	–	–
Lettuce	0.1	49	0.2	1.1	1.3	1.3	9.2	–
Bread	0.4	937	0.8	46	4.7	2.7	–	2200
Eggs	0.6	648	11	0.7	13	–	–	–
Meat	0.9	485	3.1	–	22	–	–	150
Fish	1.2	932	16	–	20	–	–	100

Tasks:

(a) State the decision variables.

(b) Formulate the objective function of a linear program to compute a diet that minimizes the total cost.

(c) Formulate the corresponding constraints and state them in your report.

(d) Use `scipy`'s linear program solver[5] to solve the linear program.

(e) Report the resulting diet and check whether your solution fulfills all constraints.

(f) Compute and state the total cost of the diet.

(g) Is the resulting diet healthy? Why/why not?

(h) To ensure a proper supplementation with omega-3 fatty acids, set the minimal intake of fish to 30g. What do you observe?

[5] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html`