Week 2 Tutorial 1

You may want to use the following test to determine wether a point is a local minimum, maximum or saddle point of a function $f\colon \mathbb{R}^2 \to \mathbb{R}$ is the so-called 2^{nd} derivative test (this works only for functions on $\mathbb{R}^2$. Assume that all partial derivatives exist and recall that the Hessian is given by

$$\nabla^2 f(x,y) = \begin{pmatrix} \partial_{xx} f & \partial_{xy} f \\ \partial_{yz} f & \partial_{yy} f \end{pmatrix}$$

Let (a,b) be a critical point, that is $\partial_x f(a,b) = \partial_y f(a,b) = 0$. Then

- $\det\nabla^2 f(a,b) > 0$ and $\partial_{xx} f(a,b) > 0$, then (a,b) is a local minimum.
- $\det\nabla^2 f(a,b) > 0$ and $\partial_{xx} f(a,b) < 0$ then (a,b) is a local maximum.
- $\det \nabla^2 f(a,b) < 0$ then (a,b) is a saddle point.
- $\det \nabla^2 f(a,b) = 0$ the test is inconclusive. The point (a,b) could be a minimum, maximum or saddle point.

(1.1) Determine the critical points of the function

$$f(x,y) = x^3 + y^3 - 3\alpha xy$$

with respect to $\alpha \in \mathbb{R}\backslash\{0\}$ and decided whether it is a minimum, maximum or saddle point.

(1.2) Consider the **Rosenbrock function** in $\mathbb{R}^2$,

$$f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Compute the gradient ∇f and the Hessian $\nabla^2 f$. Show that $\boldsymbol{x}^* = (1,1)^\top$ is the only local minimizer of this function, and that the Hessian at this point is positive definite.

Using Python or another computing system, draw a contour plot of the Rosenbrock function.

The Rosenbrok function is one of the benchmark problems in global optimisation. You can show them how it looks like (contour lines, ...) and maybe have a look at other benchmark problems
https://en.wikipedia.org/wiki/Rosenbrock_function
https://www.sfu.ca/~ssurjano/optimization.html

(1.3) Problems in optimization are often solved by iteration: if we want to minimise a function $f(\boldsymbol{x})$, we start with a point $\boldsymbol{x}_0$ and generate a sequence of points $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots$ such that the $\boldsymbol{x}_i$ approach a minimizer of f. One method of generating auch a sequence for a differentiable function is by **gradient descent**:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \eta \nabla_i f(\boldsymbol{x}_i), \quad i = 1, 2, \dots$$

The parameter ∇_i can be constant or change in time, and is called the **step length** or **learning rate** in machine learning.

For tutors (and students if they want) Using Python or another computing system, compute and plot the sequence of points $\boldsymbol{x}_k$, starting with $\boldsymbol{x}_0 = (0,0)^\top$, for the gradient descent algorithm for the problem

$$\text{minimize} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$$

with data

$$\boldsymbol{A} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ -1 & 0 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{b} = \begin{pmatrix} 10 \\ -1 \\ 0 \end{pmatrix}.$$

Experiment with different step lengths and try to find an optimal one.

For students: Calculate the closed form solution of this problem. Is this solution unique? How does the situation change if you have a 2×2 matrix, for example

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