Week 2 Tutorial 1

(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 decide whether it is a minimum, maximum or saddle point.

The gradient of the function is given by

$$\nabla f(x, y) = \begin{pmatrix} 3x^2 + 3\alpha y \\ -3y^2 + 3\alpha x \end{pmatrix}$$

Setting it equal to zero, gives the following system

$$\begin{aligned} 3x^2 + 3\alpha y &= 0 \\ -3y^2 + 3\alpha x &= 0 \end{aligned}$$

which implies that $\alpha y = -x^2$ and $\alpha x = y^2$. Therefore $x = \frac{y^2}{\alpha}$, and $\alpha y + \frac{y^4}{\alpha} = y(y^3 + \alpha^3) = 0$. The solutions are therefore

$$y = 0 \text{ and } x = 0 \quad \text{or} \quad y = -\alpha \text{ and } x = \alpha.$$

The Hessian is

$$\nabla^2 f(x, y) = \begin{pmatrix} 6x & 3\alpha \\ 3\alpha & -6y \end{pmatrix}$$

Evaluating the Hessian at the two critical points gives

$$\begin{aligned} (0, 0) &: \begin{pmatrix} 0 & 3\alpha \\ 3\alpha & 0 \end{pmatrix} \\ (\alpha, -\alpha) &: \begin{pmatrix} 6\alpha & 3\alpha \\ 3\alpha & 6\alpha \end{pmatrix} \end{aligned}$$

In the first case the determinante is negative for all values of α considered, and therefore we have a saddle point.
In the second case the determinant is given by $25\alpha^2$. For $\alpha > 0$ we have that $f_{xx} > 0$ and therefore it's a local minimum, for $\alpha < 0$ we have that $f_{xx} < 0$ and therefore it's a local maximum.

For TAs Some stuff you can discuss before or while going through the example:

- Partial derivatives, gradients and Hessian
- 2nd partial derivative test
  https://en.wikipedia.org/wiki/Second_partial_derivative_test

(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 gradient of the Rosenbrock function is given by

$$\nabla f = \begin{pmatrix} -400x_1(x_2 - x_1^2) + 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix}$$

The Hessian by

$$\nabla^2 f = \begin{pmatrix} 1200x_1^2 - 400x_2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}$$

We see that for $(1, 1)$ the gradient vanishes, the eigenvalues of the Hessian are $\lambda_1 \approx 1001.6$ and $\lambda_2 = 0.399361$ hence it's positive definite.

For TAs: The Rosenbrock 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 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ 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**:

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

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

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

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

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

For TAs: This is a bit of an outlook (we will cover iterative algorithms in week 2). It could also be useful to write out the optimality conditions (time permitting) and calculate the minimiser explicitely. For information, here is a snippet from the lecture notes on iterative algorithms describing the derivation of the optimal step length and the resulting algorithm:

We add the factor $1/2$ because it makes the expressions a bit simpler down the line:

$$f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

The Hessian is symmetric and positive semidefinite, with the gradient given by

$$\nabla f(\boldsymbol{x}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

The method of gradient descent proceeds as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}).$$

To find the best α_k, we compute the minimum of the function

$$\alpha \mapsto \varphi(\alpha) = f(\boldsymbol{x}_k - \alpha \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b})). \tag{0.1}$$

If we set $\boldsymbol{r}_k := \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) = -\nabla f(\boldsymbol{x}_k)$ and compute the minimum of (0.1) by setting the derivative to zero,

$$\begin{aligned}
\varphi'(\alpha) &= \frac{d}{d\alpha} f(\boldsymbol{x}_k + \alpha \boldsymbol{r}_k) = \langle \nabla f(\boldsymbol{x}_k + \alpha \boldsymbol{r}_k), \boldsymbol{r}_k \rangle \\
&= \langle \boldsymbol{A}^\top(\boldsymbol{A}(\boldsymbol{x}_k + \alpha \boldsymbol{r}_k) - \boldsymbol{b}), \boldsymbol{r}_k \rangle \\
&= \langle \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b}), \boldsymbol{r}_k \rangle + \alpha^2 \langle \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k, \boldsymbol{r}_k \rangle \\
&= -\boldsymbol{r}_k^\top \boldsymbol{r}_k + \alpha \boldsymbol{r}_k^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k = 0,
\end{aligned}$$

we get the step length

$$\alpha_k = \frac{\boldsymbol{r}_k^\top \boldsymbol{r}_k}{\boldsymbol{r}_k^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k} = \frac{\|\boldsymbol{r}_k\|_2^2}{\|\boldsymbol{A}\boldsymbol{r}_k\|_2^2}.$$

Note also that when we have $\boldsymbol{r}_k$ and α_k, we can compute the next $\boldsymbol{r}_k$ as

$$\begin{aligned}
r_{k+1} &= \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1}) \\
&= \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}_k + \alpha_k \boldsymbol{r}_k)) \\
&= \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k - \alpha_k \boldsymbol{A}\boldsymbol{r}_k) = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k.
\end{aligned}$$

The gradient descent algorithm for the linear least squares problem proceeds by first computing $\boldsymbol{r}_0 = \boldsymbol{A}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0)$, and then at each step

$$\begin{aligned}
\alpha_k &= \frac{\boldsymbol{r}_k^\top \boldsymbol{r}_k}{\boldsymbol{r}_k^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k} \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{r}_k \\
\boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{r}_k.
\end{aligned}$$