This set covers chapters 1–4 and A of the book *Numerical Optimization* by Nocedal and Wright, 2nd ed.

From the book, the following exercises: 2.2–2.3, 2.6–2.9, 2.13–2.15, 3.1, 3.3–3.4, 4.1, 4.10. In addition, the exercises below. The Matlab programming exercises are 3.1 and I.4–I.9. For exercises 2.2 and 2.9, plot your results with Matlab (for your own information); for exercise 3.1, estimate the convergence rate with the `convseq` function of exercise I.4.

I.1. **Extended Rosenbrock function.** This is an extension of exercise 2.1. Consider the function $f(x) = a(x_2 - x_1^2)^2 + (1 - x_1)^2$. Compute the gradient $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$. Find and classify (as maxima, minima and saddle points) the stationary points of $f$. Compute the condition number of the Hessian at the stationary points. Plot the contours of $f$ in the rectangle $[-2, 2] \times [-1, 3]$ for $a = 4, 10, 0, -1, -4$ (you may need to select the contours manually to view the stationary point).

Repeat for the extended Rosenbrock function in $n$ variables (where $n$ is even):

$$f(x) = \sum_{i=1}^{[n/2]} \left(a(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2\right).$$

I.2. **Steepest descent & Newton directions in quadratic forms.** Consider the function $f(x) = \frac{1}{2}(x_1^2 - x_2^2)$.

- Sketch the contours of $f$ around its stationary point.
- Compute the steepest descent direction and the Newton direction at $x$. Verify the steepest descent direction is a descent direction for any $x$. For what points $x$ is the Newton direction a descent direction?
- Compute explicitly $x_{k+1}$ given $x_0$ for the steepest descent method using constant step size $\alpha_k = \alpha > 0$. What does it tend to for $k \gg 1$? Study the geometry of the problem for $\alpha < 1$, $\alpha = 0$ and $\alpha > 1$. Test the following starting points $x_0$: $(\frac{9}{1})$, $(\frac{1}{1})$, $(\frac{7}{1})$, $(\frac{1}{0})$. What happens with $x_0 = 0$? What happens if using an exact line search?

Repeat for the function $f(x) = x_1^2 + 2x_2^2$.

I.3. **Convergence rate.** Prove that the convergence of the sequences $(\frac{1}{k})$, $(2^{-k})$, $(k^{-k})$ and $(2^{-2^k})$ is sublinear, linear, superlinear and quadratic, respectively. Tabulate them for a few values of $k$.

I.4. **Empirical convergence rate.** Consider a sequence $x_0, \ldots, x_K \in \mathbb{R}^n$ produced by an optimization method where the optimizer is $x^*$. We can empirically estimate the rate of the method by fitting a line to the consecutive distances: $\log d_{k+1} = a + b \log d_k$ where $d_k = \|x_k - x^*\|$. $b$ will be the order of the method and $M = e^a$ the rate constant. Write a Matlab function `convseq` that takes as input $a$ ($K+1 \times n$ matrix $X$ (containing the sequence, rowwise) and a $1 \times n$ vector $x^*$ and plots the $K$ pairs of consecutive log-distances and the least-squares line, and gives the value of the order $b$ and the constant $M$. Apply it to the sequences of the previous exercise. What happens with the sub- and superlinear cases?

I.5. **Coordinate descent.** Program in Matlab the coordinate descent method using backtracking line search. Test it as in exercise 3.1. Estimate the convergence rate with the `convseq` function of exercise I.4.

I.6. **Exact line search in quadratic forms.** Program in Matlab the steepest descent method using exact line search for a quadratic function $f(x) = \frac{1}{2}x^T A x + b^T x$ (use the result of exercise 3.3). Test it for $x \in \mathbb{R}^2$ with matrices $A$ having condition numbers $\kappa(A) = 2, 10, 100$ and plot the sequence of iterates as in fig. 3.7. Estimate the convergence rate with `convseq`.

Repeat but for the coordinate descent method.
I.7. Floating-point computations. Compare the true result with the numerical Matlab answer in the following computations.

1. Cancellation:
   
   (a) Let $x = 1$. Evaluate in Matlab the following expressions: $y - x$, $\sqrt{(y-x)^2}$, $\sqrt{y^2 + x^2 - 2xy}$ when $y = 1 + \frac{\text{eps}}{2}$, $y = 1 + \text{eps}$, $y = 1 + \sqrt{\text{eps}}$, $y = 1 + 10^3 \sqrt{\text{eps}}$.

   (b) Consider the equation $ax^2 + bx + c = 0$ with $a = 1$, $b = 2(1 + \epsilon)$, $c = 1 + 2\epsilon$. Compute in Matlab the two roots using the formula $x_{\pm} = (\frac{-b \pm \sqrt{b^2 - 4ac}}{2a})$ and then compute their difference $d = x_+ - x_-$. Tabulate the relative error in $d$ for $\epsilon = 10^0, 10^{-1}, 10^{-2}, \ldots, 10^{-17}$.

2. Ill-conditioning:

   (a) Compute the solution to $Ax = b$ with $A = \begin{pmatrix} 1 + \epsilon & 1 \\ 1 & 1 \end{pmatrix}$ and $b = \begin{pmatrix} 2 + \epsilon \\ 2 \end{pmatrix}$, for $\epsilon \in \{5\text{eps}, 2\text{eps}, \text{eps}, \text{eps}/2\}$.

I.8. Non-isolated strict local minimisers. Plot $x^4 \cos \frac{1}{x} + 2x^4$ around $x = 0$.

I.9. Modified Newton’s method. Consider Newton’s method with a Hessian modification (algorithm 3.2 in p. 48 of the book) $B_k = \nabla^2 f(x_k) + \lambda_k I$, so that the search direction is $p_k = -B_k^{-1}\nabla f(x_k)$, and has as extreme cases the pure Newton step for $\lambda = 0$ and the steepest descent direction for $\lambda \to \infty$. Assume that $f(x) = \frac{1}{2}x^T Ax$ and $x_k = (1, -3)$. Plot the contours of $f$, the gradient at $x_k$ and the search direction $p_k$ as a function of $\lambda \geq 0$ for the following three cases: $A = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$; $A = \begin{pmatrix} 4 & 0 \\ 0 & -1 \end{pmatrix}$; $A = \begin{pmatrix} -4 & 0 \\ 0 & 1 \end{pmatrix}$). What is the relation with theorem 4.1 (p. 70 in the book) and with exercise 4.1?