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 $f(\mathbf{x}) = a(x_2 - x_1^2)^2 + (1 - x_1)^2$. Compute the gradient $\nabla f(\mathbf{x})$ and the Hessian $\nabla^2 f(\mathbf{x})$. 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):

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

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

- Sketch the contours of f around its stationary point.
- Compute the steepest descent direction and the Newton direction at $\mathbf{x}$. Verify the steepest descent direction is a descent direction for any $\mathbf{x}$. For what points $\mathbf{x}$ is the Newton direction a descent direction?
- Compute explicitly $\mathbf{x}_{k+1}$ given $\mathbf{x}_0$ 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 $\mathbf{x}_0$: $\binom{0}{1}$, $\binom{1}{1}$, $\binom{2}{1}$, $\binom{1}{0}$. What happens with $\mathbf{x}_0 = \mathbf{0}$? What happens if using an exact line search?

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

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 $\mathbf{x}_0, \dots, \mathbf{x}_K \in \mathbb{R}^n$ produced by an optimization method where the optimizer is $\mathbf{x}^*$. 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 $d_k = \|\mathbf{x}_k - \mathbf{x}^*\|$; 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 $\mathbf{X}$ (containing the sequence, rowwise) and a $1 \times n$ vector $\mathbf{x}^*$ 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 $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x}$ (use the result of exercise 3.3). Test it for $\mathbf{x} \in \mathbb{R}^2$ with matrices $\mathbf{A}$ having condition numbers $\kappa(\mathbf{A}) = 2,\ 10, 100$ 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 $y = 1 + \texttt{eps}/2$, $y = 1 + \texttt{eps}$, $y = 1 + \sqrt{\texttt{eps}}$, $y = 1 + 10^3\sqrt{\texttt{eps}}$.
   (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 $x_\pm = (-b \pm \sqrt{b^2 - 4ac})/2a$ and then compute their difference $d = x_+ - x_-$. Tabulate the relative error in d for $\epsilon = 10^0, 10^{-1}, 10^{-2}, \dots, 10^{-17}$.
2. Ill-conditioning:
   (a) Compute the solution to $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} = \left(\begin{smallmatrix} 1+\epsilon & 1 \\ 1 & 1 \end{smallmatrix}\right)$ and $\mathbf{b} = \left(\begin{smallmatrix} 2+\epsilon \\ 2 \end{smallmatrix}\right)$, for $\epsilon \in \{5\texttt{eps}, 2\texttt{eps}, \texttt{eps}, \texttt{eps}/2\}$.

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) $\mathbf{B}_k = \nabla^2 f(\mathbf{x}_k) + \lambda_k \mathbf{I}$, so that the search direction is $\mathbf{p}_k = -\mathbf{B}_k^{-1}\nabla f(\mathbf{x}_k)$, and has as extreme cases the pure Newton step for $\lambda = 0$ and the steepest descent direction for $\lambda \to \infty$. Assume that $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}$ and $\mathbf{x}_k = (1, -3)$. Plot the contours of f, the gradient at $\mathbf{x}_k$ and the search direction $\mathbf{p}_k$ as a function of $\lambda \geq 0$ for the following three cases: $\mathbf{A} = \left(\begin{smallmatrix} 4 & 0 \\ 0 & -1 \end{smallmatrix}\right)$; $\mathbf{A} = \left(\begin{smallmatrix} 4 & 0 \\ 0 & 1 \end{smallmatrix}\right)$; $\mathbf{A} = \left(\begin{smallmatrix} -4 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. What is the relation with theorem 4.1 (p. 70 in the book) and with exercise 4.1?