54. (a) Write a program in MATLAB or Octave which takes as input parameters:
    - a function $h : \mathbb{R}^n \to \mathbb{R}$,
    - its gradient $g = \nabla h : \mathbb{R}^n \to \mathbb{R}^n$,
    - a starting point $x_0$ (an $n \times 1$ vector) and
    - a tolerance `tol`

    and performs the **BFGS method** with Armijo line search for the optimization problem
    $$\min_{x \in \mathbb{R}^n} h(x)$$
    (up to the tolerance `tol` and starting at $x_0$).

    (b) Test your program on
    $$\min_{x \in \mathbb{R}^2} x_1^2 + x_2^2$$
    with $x_0 = \binom{1}{1}$ and the tolerance $\sqrt{\texttt{eps}}$.

55. Consider the **Rosenbrock function** $h : \mathbb{R}^2 \to \mathbb{R}$, $h(x) = (-10x_1^2 + 10x_2)^2 + (x_1 - 1)^2$.

    (a) Compute (by hand) the gradient $g(x) = \nabla h(x)$ of $h$.

    (b) Apply your program from example 54 to (minimizing) the Rosenbrock function with $x_0 = \binom{0}{0}$ and the tolerance $\sqrt{\texttt{eps}}$.

56. (a) Compute (by hand) the **Newton interpolation polynomial** $P(x)$ of degree 3 such that $P(-2) = 0$, $P(-1) = 1$, $P(1) = -3$ and $P(2) = 16$.

    (b) Evaluate (by hand) $P(0)$.

57. (a) Write a program in MATLAB or Octave which takes as input parameters:
    - a list of $n + 1$ points $x$ in $\mathbb{R}$ and
    - a list of $n + 1$ values $y$ in $\mathbb{R}$

    and computes the coefficients $a_0, \ldots, a_n$ of the **Newton interpolation polynomial** $P$ of degree $n$ which satisfies $P(x_i) = y_i$ $\forall i = 0, \ldots, n$. Test your program on example 56 (a), i.e. compare the result with the answer you computed by hand.
    Note: The coefficients should be the coefficients appearing in the generalized Horner scheme (the ones directly produced by the algorithm), i.e. **do not expand** your polynomial!

    (b) Write a program in MATLAB or Octave which takes as input parameters:
    - the list of $n + 1$ points $x$ in $\mathbb{R}$ and
    - the list of $n + 1$ coefficients $a$ in $\mathbb{R}$

    of a Newton interpolation polynomial $P$, as well as
    - an evaluation point $x_e$

    and evaluates $P(x_e)$ using the **generalized Horner scheme**. Test your program on example 56 (b), i.e. compare the result with the answer you computed by hand.

58. Let $f : \mathbb{R} \to \mathbb{R}$ be any function with $f(0) = 0$, $f(1) = 2$ and $f(2) = 1$. By hand, approximate the integral

$$\int_0^2 f(x)dx$$

(a) by the midpoint rule for $h = 2$ (i.e. using $f(1)$),

(b) by the trapezoid rule for $h = 2$ (i.e. using $f(0)$ and $f(2)$),

(c) by the trapezoid rule for $h = 1$ (i.e. using $f(0)$, $f(1)$ and $f(2)$) and

(d) by Simpson's rule for $h = 1$ (i.e. using $f(0)$, $f(1)$ and $f(2)$).

Note: The trapezoid rule is the Newton-Cotes formula of degree 1, Simpson's rule is the Newton-Cotes formula of degree 2.

59. Let $f(x) = \frac{1}{2}(7x - 3x^2)$. By hand:

(a) Compute the exact integral $\int_0^2 f(x)dx$.

(b) Evaluate $f(0)$, $f(1)$ and $f(2)$.

(c) Compare with example 58. In particular, what can you say about Simpson's rule? Why?

60. Let $f(x) = \left(2 - \frac{\sqrt{2}}{2}\right) \sin(\frac{\pi}{2}x) + \sin(\frac{\pi}{4}x)$. By hand:

(a) Compute the exact integral $\int_0^2 f(x)dx$, and evaluate your result numerically using a computer or a calculator.

(b) Evaluate $f(0)$, $f(1)$ and $f(2)$.

(c) Compare with example 58.