AI505/AI801, Optimization – Exercise Sheet 02

2026-02-22

Exercises with the symbol $^+$ are to be done at home before the class. Exercises with the symbol * will be tackled in class. The remaining exercises are left for self training after the exercise class. Some exercises are from the text book and the number is reported. They have the solution at the end of the book.

Exercises

Exercise 1 * (3.6)

Suppose we have a unimodal function defined on the interval $[1, 32]$. After three function evaluations of our choice, will we be able to narrow the optimum to an interval of at most length 10? Why or why not? How much more can we reduce with one further evaluation?

Exercise 2 $^+$ (3.1)

Give an example of a problem when Fibonacci search can be applied while the bisection method not.

Exercise 3 $^+$ (3.4)

Suppose we have $f(x) = x^2/2 - x$. Apply the bisection method to find an interval containing the minimizer of f starting with the interval $[0, 1000]$. Execute three steps of the algorithm (you can do this by hand or in Python).

Exercise 4 $^+$ (3.5)

Suppose we have a function $f(x) = (x + 2)^2$ on the interval $[0, 1]$. Is 2 a valid Lipschitz constant for f on that interval?

Exercise 5 $^+$ (4.2)

The first Wolfe condition requires

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) \leq f(\boldsymbol{x}_k) + \beta\alpha\nabla \boldsymbol{d}_k f(\boldsymbol{x}_k)$$

What is the maximum step length α that satisfies this condition, given that $f(\boldsymbol{x}) = 5 + x_1^2 + x_2^2$, $\boldsymbol{x}_k = [-1, -1]$, $\boldsymbol{d} = [1, 0]$, and $\beta = 10^{-4}$?

Exercise 6 $^+$

Consider the following function $f(x_1, x_2)$:

$$f(x_1, x_2) = x_1^4 - 5x_1^2 + x_2^2$$

We are at $\boldsymbol{x}_0 = [0, 1]$ and we want to move in the descent direction $\boldsymbol{d} = [1, -1]$.

- Plot $f(x_1, x_2)$.
- Define the line search problem to solve.
- Solve the line search problem using the Strong Backtracking Line Search Algorithm to find a value for the learning rate that satisfies the Strong Wolfe conditions.
- Plot the function of the line search and the iterates found in the process.

Note, you find the algorithm in Python in the slides. The focus of this exercise is on visualisation of the search process and on making sense of it.

Exercise 7 *

The steepest descent algorithm is a Descent Direction Iteration method that moves along $d_k = -\nabla f(\boldsymbol{x}_k)$ at every step. Program steepest descent algorithms using the backtracking line search. Use them to minimize the Rosenbrock function. Set the initial step length $\alpha_0 = 1$ and print the step length used by each method at each iteration. First, try the initial point $x_0 = [1.2, 1.2]$ and then the more difficult starting point $x_0 = [-1.2, 1]$.

Consider implementing and comparing also other ways for solving the line search problem and the conjugate gradient.

Exercise 8 *

Descent direction methods may use search directions other than the steepest descent mentioned in the previous exercise. In general, which descent direction guarantees to produce a decrease in f?

Exercise 9 $^+$ (5.1)

Compute the gradient of $\boldsymbol{x}^T A \boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$ when A is symmetric. You have the result in the slides of lectures, here you are asked to show that the result is correct.

Exercise 10 * (5.2)

Apply one step of gradient descent to $f(x) = x^4$ from $x_0 = 1$ with both a unit step factor and with exact line search.

Exercise 11 $^{+}$

Recall that a way to measure rate of convergence is by the *limit of the ratio of successive errors*,

$$\lim_{k\to\infty} \frac{f(\boldsymbol{w}_{k+1}) - f(\boldsymbol{w}^*)}{f(\boldsymbol{w}_k) - f(\boldsymbol{w}^*)} = r$$

Different r values of give us different rates of convergence:

- If $r = 1$ we call it a sublinear rate.
- If $r \in (0, 1)$ we call it a linear rate.
- If $r = 0$ we call it a superlinear rate.

Consider the following sequences, which represent the error $e_k = \|F(\boldsymbol{w}_k) - F^*\|$ at iteration k of an optimization algorithm:

1. $e_k = 0.5^k$
2. $e_k = \frac{1}{k+1}$
3. $e_k = 0.1^k$
4. $e_k = \frac{1}{(k+1)^2}$
5. $e_k = \frac{1}{2^{2^k}}$

Tasks:

a) Classify the convergence rate of each sequence as linear, sublinear, superlinear, or quadratic.

b) Provide a justification for each classification by computing the ratio e_{k+1}/e_k or by using the definition of order of convergence.

Exercise 12 *

Consider applying gradient descent to the one-dimensional quadratic function

$$f(x) = \frac{1}{2}x^2$$

with the update rule:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k).$$

where $\nabla f(x) = x$.

Tasks:

a) Derive the update formula for $\boldsymbol{x}_k$.

b) Show that the error $e_k = \|\boldsymbol{x}_k\|$ follows an exponential decay when $0 < \alpha < 2$.

c) Compute $\frac{e_{k+1}}{e_k}$ and determine the rate of convergence for different values of α.

d) Set up a Python experiment where gradient descent is applied with different step sizes (α) and verify the theoretical convergence rate numerically.

Hint: Try $\alpha = 0.1, 0.5, 1, 1.5$ and observe how quickly the errors decrease.

(5.7)

In conjugate gradient descent, what is the descent direction at the first iteration for the function $f(x, y) = x^2 + xy + y^2 + 5$ when initialized at $[x, y] = [1, 1]$? What is the resulting point after two steps of the conjugate gradient method?