Rehearsal Questions for the course "Numerical Optimization"

Moritz Diehl and teaching assistants

April 15, 2026

The following questions might help in rehearsing the contents of the course:

1. What is an optimization problem? Objective, degrees of freedom, constraints. Feasible set? Standard form of NLP.

   An optimization problem consists of the following three ingredients:

   - An objective function $x \mapsto f(x) \in \mathbb{R}$ that shall be maximized or minimized.
   - Some decision variables, typically lying in a vector space $\mathbb{R}^n$: $x \in \mathbb{R}^n$
   - A feasible set $\Omega \subset \mathbb{R}^n$, typically defined by equality and inequality constraints:

   $$\Omega = \left\{ x \in \mathbb{R}^n \;\middle|\; g(x) = 0 \text{ and } h(x) \geq 0 \right\} \tag{1}$$

   The problem is then written as follows:

   $$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) && \text{(2a)} \\ \text{subject to} \quad & g(x) = 0, && \text{(2b)} \\ & h(x) \geq 0. && \text{(2c)} \end{aligned}$$

2. Definition of global and local minimum.

   We say that $x^\star$ is a *global minimizer* when it is feasible, i.e. $x^\star \in \Omega$ and optimal, i.e. $\forall x \in \Omega : f(x) \geq f(x^\star)$.
   We say that it is the *strict global minimizer* when $x^\star \in \Omega$ and $\forall x \in \Omega \setminus \{x^\star\} : f(x) > f(x^\star)$.
   We say that it is a *local minimizer* when there exists a neighborhood $\mathcal{N}$ of $x^\star$ (e.g. an open ball around $x^\star$) such that it is optimal within that neighborhood, i.e. $\forall x \in \Omega \cap \mathcal{N} : f(x) \geq f(x^\star)$.

3. State a condition under which there exists at least one minimizers.

   If the feasible set is *compact*, i.e. bounded and closed, and the objective function is continuous, then there exists at least one *global* minimizer (this is called the Theorem of Weierstrass)

4. Types of optimization problems: Linear / Quadratic programming (LP/QP), convex, smooth, mixed-integer, ...

- LP:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^\top x && \text{(3a)} \\ \text{subject to} \quad & Ax - b = 0, && \text{(3b)} \\ & Cx - d \geq 0. && \text{(3c)} \end{aligned}$$

- QP:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^\top x + \frac{1}{2} x^\top B x && \text{(4a)} \\ \text{subject to} \quad & Ax - b = 0, && \text{(4b)} \\ & Cx - d \geq 0. && \text{(4c)} \end{aligned}$$

- **Convex problem:** in the following form: $f(\cdot)$ a convex function, each component $h_j(\cdot)$ has to be concave, and $g(\cdot)$ has to be affine

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) && \text{(5a)} \\ \text{subject to} \quad & g(x) = 0, && \text{(5b)} \\ & h(x) \geq 0. && \text{(5c)} \end{aligned}$$

- **Smooth problem:** the functions $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are continuously differentiable in the following problem

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) && \text{(6a)} \\ \text{subject to} \quad & g(x) = 0, && \text{(6b)} \\ & h(x) \geq 0. && \text{(6c)} \end{aligned}$$

- **Mixed-integer program:** some of the variables are restricted to integers:

$$\begin{aligned} \underset{x \in \mathbb{R}^n,\, z \in \mathbb{Z}^{n'}}{\text{minimize}} \quad & f(x, z) && \text{(7a)} \\ \text{subject to} \quad & g(x, z) = 0, && \text{(7b)} \\ & h(x, z) \geq 0. && \text{(7c)} \end{aligned}$$

5. When is a function convex? Gives the definition and a characterization when it is twice differentiable. Definition. If it is twice differentiable?

A function $f : \Omega \to \mathbb{R}$ is *convex*,if Ω is convex and if:

$$\forall x, y \in \Omega, t \in [0,1] : \quad f(x + t(y - x)) \leq f(x) + t(f(y) - f(x)) \tag{8}$$

i.e. all secants are above graph.
When $f(\cdot)$ is *twice continuously differentiable* and Ω convex, then it is convex *if and only if* the Hessian is positive semi-definite at all point, i.e.

$$\forall x \in \Omega : \qquad \nabla^2 f(x) \succcurlyeq 0 \tag{9}$$

6. When is a set convex? Definition.

A set $\Omega \subset \mathbb{R}^n$ is convex,if:

$$\forall x, y \in \Omega, t \in [0,1] : \quad x + t(y - x) \in \Omega \tag{10}$$

i.e. all connecting lines lie inside set.

7. What is a "stationary" point in the the case of unconstrained smooth optimization?

In the case of unconstrained optimization a stationary point is a point $\bar{x}$ where the gradient is zero: $\nabla f(\bar{x}) = 0$.

8. How are gradient and Hessian of a function $f : \mathbb{R}^n \to \mathbb{R}$ defined?

The *gradient* of $f(\cdot)$ is defined to be the *vector field* whose components are the *partial derivatives* of f, i.e.:

$$\nabla f(x) \coloneqq \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{11}$$

The *Hessian* of $f(\cdot)$ is the matrix containing the second derivatives of f:

$$\nabla^2 f(x) \coloneqq \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdot & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdot & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdot & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{12}$$

9. What are the first order necessary conditions for optimality (FONC) in the case of unconstrained optimization problems?

The theorem states that if $f(\cdot)$ is continuously differentiable, and $x^\star$ is a solution of the problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x). \tag{13}$$

Then $x^\star$ is a stationary point, i.e.

$$\nabla f(x^\star) = 0 \tag{14}$$

10. What are the second order necessary conditions for optimality (SONC) in the case of unconstrained optimization problems?

The theorem states that if $f(\cdot)$ is twice continuously differentiable, and $x^\star$ is a solution of the problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x). \tag{15}$$

Then $x^\star$ is a stationary point and the Hessian is positive semi-definite at this point, i.e.

$$\nabla f(x^\star) = 0, \nabla^2 f(x^\star) \succcurlyeq 0. \tag{16}$$

11. What are the second order sufficient conditions for optimality (SOSC) in the case of unconstrained optimization problems?

The theorem states that if $f(\cdot)$ is twice continuously differentiable, and $x^\star$ is a stationary point and the Hessian is positive definite at this point, i.e.

$$\nabla f(x^\star) = 0, \nabla^2 f(x^\star) \succ 0, \tag{17}$$

then $x^\star$ is a local minimizer to the problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x). \tag{18}$$

12. Basic idea of iterative descent methods?

It is an iterative algorithm where at iteration k, the iterate $x_k \in \mathbb{R}^n$ is the current best guess to the solution, and we look for the next iteration x_{k+1} to improve it. In optimization, we typically look for x_{k+1} such that a sufficient decrease of the value will be produced, which takes the form of $f(x_{k+1}) \leq f(x_k) - \ldots$.

13. Definition of local convergence rates: q/r-linear, superlinear, quadratic?

Let $(x_k)_{k\in\mathbb{N}}$ be a sequence in $\mathbb{R}^n$ that converges to $x^\star$. We say that the convergence is:

- **Q-linear:** if there is a constant $r \in (0,1)$ such that for k sufficiently large:

$$\|x_{k+1} - x^\star\| \leq r \|x_k - x^\star\|, \tag{19}$$

  e.g. $x^k = \frac{1}{2^k}$

- **R-linearly:** if it is upperbounded by another sequence itself convering Q-linearly:

$$y_{k+1} \leq r y_k, \tag{20a}$$
$$\|x_k - x^\star\| \leq y_k. \tag{20b}$$

- **Q-superlinear:** if there is a sequence r_k such that $r_k \to 0$ and:

$$\|x_{k+1} - x^\star\| \leq r_k \|x_k - x^\star\|, \tag{21}$$

  e.g. $x^k = \frac{1}{k!}$

- **Q-quadratic:** if there is a constant $M > 0$ such that:

$$\|x_{k+1} - x^\star\| \leq M \|x_k - x^\star\|^2. \tag{22}$$

14. What is a locally convergent algorithm? What is a globally convergent algorithm? What does the term "globalization" usually mean for optimizers?

An algorithm is locally convergent when it converges to a solution under the condition that it is initialized closed enough to a solution.
An algorithm is globally convergent when it converges to a solution no matter how it is initialized.
The term "globalization" is a component of an algorithm to ensure convergence even for bad initialization of the algotithm.

15. What is the Armijo condition? Why is it used in line-search algorithms?

The Armijo condition is a condition on the new iterate x_{k+1}, making sure that it induces a *sufficient decrease* of the objective function:

$$f(x_{k+1}) \leq f(x_k) + \gamma \nabla f(x_k)^\top (x_{k+1} - x_k). \tag{23}$$

Using this condition in a backtracking line-search algorithm often ensures global convergence of the optimization algorithm (given some condition on the chosen descent direction).

16. Why is satisfaction of Armijo condition alone not sufficient to guarantee convergence towards stationary points? Give a counterexample.

If one choses $x_{k+1} = x_k + t_k p_k$ with p_k being a descent direction, and the t_k a sequence that converges very quickly to zero, the algorithm might get stuck, even though the Armijo condition will be satisfied.
Example: $f(x) = x^2$ and $x_{k+1} = x_k - \frac{x_k}{(k+2)^2}$ satisfies the Armijo condition for k large enough, and yet it is does not converge to the solution:

$$x_k = \prod_{i=2}^{k} \left(1 - \frac{1}{i^2}\right) x_0, \tag{24a}$$

$$= \frac{1}{2}\frac{k+1}{k-1} x_0 \underset{k\to+\infty}{\longrightarrow} \frac{x_0}{2} \neq 0. \tag{24b}$$

17. What is backtracking?

In backtracking, the candidates for the next iterate is $x_{k+1} = x_k + t_k p_k$, and we shrink the step size $t_k \leftarrow \beta t_k$ with $\beta \in (0, 1)$ until a condition is satisfied.
Usually, the condition is the Armijo condition, we start at $t_k = 1$ and we choose $\beta = 0.8$.

18. Define the *the steepest descent method*. What is the local convergence rate?

The steepest descent method is the following iterative algorithm:

$$x_{k+1} = x_k - \alpha \nabla f(x_k), \tag{25}$$

with $\alpha > 0$ being the *learning rate*.
If the learning rate α is small enough to ensure convergence, the steepest descent method converges *Q-linearly* to a stationary point, i.e.:

$$\|x_{k+1} - x^\star\| \leq r \|x_k - x^\star\| \leq \cdots \leq r^{k+1} \|x_0 - x^\star\|, \tag{26}$$

where r is a constant in $(0, 1)$.

19. What is Newton's method for solution of nonlinear equations $F(x) = 0$? How does it iterate, what is the motivation for it. How does it converge locally?

It is a numerical method for finding a solution to some nonlinear equation $F(x) = 0$. It is an iterative method where at each step, we linearize the function $F(\cdot)$ at the current iterate x_k and solve the linearized equation. The iteration is given by:

$$x_{k+1} = x_k - \nabla F(x_k)^{-1} F(x_k). \tag{27}$$

This locally converges very fast for smooth functions, i.e. the convergence rate is *quadratic*, i.e.:

$$\|x_{k+1} - x^\star\| \leq M \|x_k - x^\star\|^2, \tag{28}$$

where M is a constant that depends on the function $F(\cdot)$ and the point $x^\star$.
A limitation is that the method is guranteed to converge *only if* the initial guess x_0 is close enough to some solution $x^\star$.

20. How works Newton's method for unconstrained optimization?

In Newton's method for unconstrained optimization, we apply Newton's method to the nonlinear root-finding equation:

$$F(x) \coloneqq \nabla f(x) = 0. \tag{29}$$

The resulting iteration is:

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k). \tag{30}$$

21. What are *Newton type / approximate Newton* methods?

In Newton type methods, we apply tyhe same idea, but replace the Hessian $\nabla^2 f(x_k)$ by some approximation B_k of the Hessian, i.e.:

$$x_{k+1} = x_k - B_k^{-1} \nabla f(x_k), \tag{31}$$

where B_k replaces the Hessian $\nabla^2 f(x_k)$.

22. What is the idea behind Quasi-Newton methods?

In Quasi-Newton methods, the Hessian is approximated in such a way that the approximation and the exact Hessian match eventually:

$$B_k \underset{k \to +\infty}{\longrightarrow} \nabla^2 f(x_k). \tag{32}$$

This is useful because it ensures *superlinear* convergence, often almost as good as the exact Newton method.

23. What is the secant condition? How is it motivated?

The secant condition is a condition on the Hessian approximation that would justify the latest change in the gradient. More precisely, it is the following condition:

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k). \tag{33}$$

This formula can be retrieved by linearizing the gradient.

24. What is the BFGS formula? Under which condition does it preserve positive definiteness?

Write $s_k \coloneqq x_{k+1} - x_k$ the latest change in the solution point, and $y_k \coloneqq \nabla f(x_{k+1}) - \nabla f(x_k)$ the latest change in the gradient.
The BFGS update formula is given by:

$$B_{k+1} = B_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k}. \tag{34}$$

This preserves positive definiteness of the Hessian approximation B_k when $y_k^\top s_k > 0$.

25. Prove that the latter condition is necessary for any update formula satisfying the secant condition to stay positive definite.

Multiplying on both side of the secant condition by $s_k^\top$ gives:

$$0 < s_k^\top B_{k+1} s_k = s_k^\top y_k, \tag{35}$$

which proves that the condition was necessary.

26. What is a linear least squares problem (unconstrained)? What is a a nonlinear least squares problem (unconstrained)?

- **Linear least squares problem:**

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|Ax - b\|^2 . \tag{36}$$

- **Nonlinear least squares problem:**

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \|F(x)\|^2 . \tag{37}$$

  where $F : \mathbb{R}^n \to \mathbb{R}^m$ is a nonlinear function.

27. How does the Gauss-Newton method iterate? When is it applicable?

Gauss-Newton method is an iterative method for solving nonlinear least squares problems. It iterates by solving the linear least squares problem induced by a linearization of the inner function $F(\cdot)$ at each iteration:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|F(x_k) + \nabla F(x_k)(x - x_k)\|^2 . \tag{38}$$

28. When does the Gauss-Newton method perform well? What local convergence rate does it have?

In general, the Gauss-Newton method performs well when the nonlinear components of the function $F(\cdot)$ are small at the solution. In general, the convergence rate is *Q-linear*, but it might become *superlinear* when at the solution $x^\star$:

$$\text{for } j = 1, \ldots, m : \qquad \nabla^2 F_j(x^\star) = 0 \ \text{ or } F_j(x^\star) = 0. \tag{39}$$

29. Statistical motivation of least squares terms in estimation problems?

Assume that some measurements $\eta_j \in \mathbb{R}$ are available, and modeled as follows:

$$\eta_j \approx M_j(\bar{x}). \tag{40}$$

where the model $M_j : \mathbb{R}^n \to \mathbb{R}$ is a function that depends on some parameters $\bar{x} \in \mathbb{R}^n$. Then a good estimate of the parameters $\bar{x}$ is given by solving the least squares problem:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{j=1}^{m} (\eta_j - M_j(x))^2. \tag{41}$$

This is motivated statistically by assuming that the noise is normally distributed:

$$\eta_j = M_j(\bar{x}) + \varepsilon_j, \tag{42a}$$
$$\varepsilon_j \sim \mathcal{N}(0, \sigma^2). \tag{42b}$$

In this case, the maximum likelihood problem is equivalent to the least squares problem, i.e.:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad -\log p(\text{measurements} \mid x) = \sum_{j=1}^{m} \frac{1}{2\sigma^2} (\eta_j - M_j(\overline{x}))^2. \tag{43}$$

30. What are the differences and similarities between line search and trust region methods?

These two methods are for ensuring global convergence by preventing the algorithm from taking too large steps.
In trust-region algorithm, the next iterate is restricted to be in a *trust region* around the current iterate x_k, so the step is constrained *before* computing it.
In line-search algorithm, *after* computing a descent direction, the algorithm searches for the right size of step to take in this direction.

31. List two ways to compute derivatives with help of computers.

- Finite differences;
- Automatic differentiation;

32. What errors occur when computing derivatives with finite differences? Do you know a rule of thumb of how large to choose the perturbation?

In finite differences, for computing the derivative w.r.t. one direction, we use:

$$\nabla f(x)^\top p \approx \frac{f(x+\varepsilon p) - f(x)}{\varepsilon}. \tag{44}$$

If ε is too small the derivative will suffer from **numerical noise (round-off error)**. On the other hand, if ε is too large the **linearization error** will be dominant.
A good rule of thum is to use $t = \sqrt{\varepsilon_{\text{mach}}}$ which ensures an error on the scale of $\sqrt{\varepsilon_{\text{mach}}}$.
For example, if the computer computes $f(x)$ with a precision of $\pm 10^{-10}$ (i.e. 10 digits of accuracy), then the derivatives will be computed with an error of about $\pm 10^{-5}$ (i.e. 5 digits of accuracy.)

33. What is the idea behind Automatic Differentiation (AD)? What is its main advantage?

Automatic Differentiation takes a function as a list of elementary operations to computes, where each of the elementary operations has a known derivative. Then, it computes the derivative of the function by applying the chain rule to each elementary operation, and combining the results.
The advantage is that it computes derivatives with machine precision and that the computational cost scales nicely with the dimension of the function.

34. Can AD be applied to compute second order derivatives?

Yes, by applying twice the AD principle, second-derivatives are seamlessly computed.

35. There are two ways of AD. Describe briefly. What are the advantages / disadvantages of the two?

There are two modes of AD:

- **Forward mode:** Applying the chain rule from the input to the output. This can be done while computing the value of the function itself. The cost of this algorithm scales nicely with the number of outputs. Hence this is useful when the function has a small number of inputs and a large number of outputs.
- **Reverse/Backward mode:** Applying the chain rule from the output to the input. For this, the algorithm needs the value of all of the intermediate variables, so a forward sweep is needed first (i.e. what we do when we evaluate the function). The disadvantage is that one needs to save in memory all of the intermediate variables before computing the derivative. The advantage is that the computational cost of this algorithm scales nicely with the number of inputs. Hence this is useful when the function has a large number of inputs and a small number of outputs. Note that this is typically the case when one is required to compute the gradient of the objective function.

36. Write a nonlinear program (NLP) in its standard form. How is the lagrangian function defined?

The standard form of a nonlinear program (NLP) is typically:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) && \text{(45a)} \\ \text{subject to} \quad & g(x) = 0, && \text{(45b)} \\ & h_j(x) \geq 0. && \text{(45c)} \end{aligned}$$

The Lagrangian function of this NLP is defined as:

$$\mathcal{L}(x, \lambda, \mu) = f(x) - \lambda^\top g(x) - \mu^\top h(x), \tag{46}$$

where λ is a vector with the same dimension as the number of equality constraints, and μ is a vector with the same dimension as the number of inequality constraints.

37. What is the constraint qualification (CQ)? What is the linear independence constraint qualification (LICQ) at some point $\bar{x}$?

CQ refers to some regularity conditions on the active constraints at some point $\bar{x}$. One of the most common CQ is the *linear independence constraint qualification* (LICQ), which states that the gradients of the active constraints at x are linearly independent, i.e. the vectors $\nabla g_j(\bar{x})$ (for all index j) and $\nabla h_k(\bar{x})$ (for indices k such that $h_k(\bar{x}) = 0$) are linearly independent.

38. What are the Karush-Kuhn-Tucker (KKT) conditions for optimality? Why is it useful?

The KKT conditions are necessary conditions for a point to be a minimizer when LICQ holds. More formally, if $x \in \mathbb{R}^n$ is such that LICQ holds, and is a local minimizer, then there exists some Lagrange multiplier vectors λ and μ,such the following conditions hold:

$$\begin{aligned} \nabla f(x) - \lambda^\top \nabla g(x) - \mu^\top \nabla h(x) &= 0, && \text{(47a)} \\ g(x) &= 0, && \text{(47b)} \\ \begin{cases} h(x) \geq 0, \\ \mu = 0, \end{cases} \quad \text{or} \quad & \begin{cases} h(x) = 0, \\ \mu \geq 0. \end{cases} && \text{(47c)} \end{aligned}$$

39. What are the first order necessary conditions for optimality (FONC) (constrained)?

If $x^\star$ is a local minimizer of the NLP and LICQ holds at $x^\star$, then there exists a $\lambda^\star \in \mathbb{R}^m$ and $\mu^\star \in \mathbb{R}^q$ such that:

$$\begin{aligned} \nabla f(x^\star) - \nabla g(x^\star)\lambda^\star - \nabla h(x^\star)\mu^\star &= 0, && \text{(48a)} \\ g(x^\star) &= 0, && \text{(48b)} \\ h(x^\star) &\geq 0, && \text{(48c)} \\ \mu^\star &\geq 0, && \text{(48d)} \\ \mu_i^\star h_i(x^\star) &= 0, \quad i = 1, \ldots, q. && \text{(48e)} \end{aligned}$$

40. What are the second order necessary conditions for optimality (SONC) (constrained)?

Regard $x^\star$ with LICQ. If $x^\star$ is a local minimizer of the NLP, then:

i. $\exists \lambda^\star, \mu^\star$ so that KKT conditions hold;

ii. $\forall p \in C(x^\star, \mu^\star)$ holds that $p^\top \nabla_x^2 \mathcal{L}(x^\star, \lambda^\star, \mu^\star) p \geq 0$.

41. What are the second order sufficient conditions for optimality (SOSC) (constrained)?

If $x^\star$ satisfies LICQ and:

i. $\exists \lambda^\star, \mu^\star$ so that KKT conditions hold;

ii. $\forall p \in C(x^\star, \mu^\star),\ p \neq 0$, holds that $p^\top \nabla_x^2 \mathcal{L}(x^\star, \lambda^\star, \mu^\star) p > 0$,

then $x^\star$ is a strict local minimizer.

42. What is the "active set of constraints"?

This is the set of indices of inequality constraints such that the inequality is asctually an equality. More precisely:

$$\mathbb{A}(x) \coloneqq \left\{ k \in \{, \dots, m\} \;\middle|\; h_k(x) = 0 \right\}. \tag{49}$$

43. Give a standardform of a QP.

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^\top x + \frac{1}{2} x^\top B x && \text{(50a)} \\ \text{subject to} \quad & Ax - b = 0, && \text{(50b)} \\ & Cx - d \geq 0. && \text{(50c)} \end{aligned}$$

44. When is a QP convex?

A QP is convex whenever its Hessian B is positive semi-definite, i.e. $B \succcurlyeq 0$.

45. What is the main idea of an active set strategy?

An active set strategy tries to identify the active set of constraints at each iteration such that thez problem that is solved is equivalent to an *equality constrained optimization problem* (instead of an *inequality constrained optimization problem*).

46. What is the main idea behind an SQP method (for inequality constrained problems)?

In SQP, at each iteration, the constaints are linearized, and the objective function is approximated by a quadratic function.

47. What is the L_1-penalty method for equality-constrained problems? Under which condition is it "exact", i.e. has the same local minima as the original NLP?

The L_1-penalty method approximates the constrained problem:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) && \text{(51a)} \\ \text{subject to} \quad & g(x) = 0, && \text{(51b)} \end{aligned}$$

with:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \rho \left\| g(x) \right\|_1 \quad = f(x) + \rho \sum_{i=1}^{m} \left| g_i(x) \right|. \tag{52}$$

The approximation yields the exact solution whenever:

$$\rho \geq \left\| \lambda^\star \right\|_\infty = \max_{i=1,\ldots,m} \left| \lambda_i^\star \right|, \tag{53}$$

where $\lambda^\star$ is the vector of Lagrange multipliers of the original problem at the solution $x^\star$.

48. How works Newton's method for equality constrained optimization?

Newton's method for equality constrained optimization involves iteratively solving a system of equations derived from the first-order optimality conditions:

$$\begin{aligned} \nabla f(x_k) + \lambda_k^\top \nabla g(x_k) = 0, && \text{(54a)} \\ g(x_k) = 0 && \text{(54b)} \end{aligned}$$

At each iteration, the method updates the current solution x_k together with the associated multipliers λ_k by solving the linearized version of the KKT conditions:

$$\begin{bmatrix} \nabla^2 f(x_k) - \lambda_k^\top \nabla^2 g(x_k) & \nabla g(x_k)^\top \\ \nabla g(x_k) & 0 \end{bmatrix} \begin{bmatrix} x - x_k \\ \lambda - \lambda_k \end{bmatrix} + \begin{bmatrix} \nabla f(x_k) + \lambda_k^\top \nabla g(x_k) \\ g(x_k) \end{bmatrix} = 0. \tag{55}$$

49. What local convergence rate does an SQP method with exact Hessian usually have?

It usually has a quadratic convergence rate under reasonable assumptions (i.e. LICQ, second order sufficient conditions).

50. What is the basic idea of interior point methods? Gives two views this class of method.

Interior point methods are method to solve problems with inequality constraints. They basically gives guideline to treat these inequalities in order to use tools from unconstrained or equality-constrained optimization. In this method, the iterates should always be feasible for the inequality constraints, and the inequalities have to be fulfilled in a strict sense.
There are two similar kinds of interior point methods (they are kind of equivalent, even though they produce different iterates):

- In the primal-dual interior point method, we modify slightly the KKT conditions to make them them less illed-conditionned, then we apply Newton's method to iteratively solve them. More precisely, in the KKT conditinos, the complementarity equations $\mu_j h_j(x) = 0$ are replaced with $\mu_j h_j(x) = \tau$ with $\tau > 0$ quite small. To ensure consistency of the method, the parameter τ decreases and converges to 0 (or to some tolerance value).
- In the logarithmic barrier interior point method: we add a barrier function $-\tau \log h(x)$ to the objective that pushes the solution to stay within the inequality constraints. More precisely, we solve the following problem using the Newton's method:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) - \tau \sum_{j=1}^{m} \log h_j(x) && \text{(56a)} \\ \text{subject to} \quad & g(x) = 0. && \text{(56b)} \end{aligned}$$

  The parameter $\tau > 0$ is chosen as above.

In both of the variations, the parameter $\tau > 0$ is chosen quite small. To ensure consistency of the method, the parameter τ decreases and converges to 0 (or to some tolerance value).

51. What is the Lagrangian dual function of a general NLP?

The Lagrangian dual function is defined as follows:

$$q(\lambda, \mu) \coloneqq \min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \mu) = \min_{x \in \mathbb{R}^n} f(x) - \lambda^\top g(x) - \mu^\top h(x). \tag{57}$$

52. What is the dual problem of a general NLP?

$$\begin{aligned} \underset{\lambda \in \mathbb{R}^p, \mu \in \mathbb{R}^q}{\text{maximize}} \quad & q(\lambda, \mu) \quad = \min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \mu) && \text{(58a)} \\ \text{subject to} \quad & \mu \geq 0. && \text{(58b)} \end{aligned}$$

53. What is weak duality? To which problems does it apply?

Weak duality always hold, and it states that the optimal value of the dual problem is a lower than the optimal value of the original (or primal) problem.

54. What is strong duality? Under which sufficient conditions does it apply?

Strong duality states that the optimal value of the dual problem is equal to the optimal value of the original (or primal) problem.
It holds for convex problems under Slater's condition, which states that there exists a feasible point for which the *nonlinear inequality constraints are inactive*.

55. What is a semidefinite program (SDP)? Give a standardform.

An SDP is an optimization problem where the constraints are affine are Linear Matrix Inequalities (LMI), and the objective is linear. More precisely, it takes the following form:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & c^\top x && \text{(59a)} \\ \text{subject to} \quad & Ax - b = 0, && \text{(59b)} \\ & B_0 + \sum_{i=1}^{n} B_i x_i \succcurlyeq 0. && \text{(59c)} \end{aligned}$$

56. How would you reformulate the following eigenvalue optimization problem into an SDP for A_0, A_1, A_2 three symmetric matrices?

$$\underset{x \in \mathbb{R}^2}{\text{minimize}} \quad \lambda_{\max}(A_0 + x_1 A_1 + x_2 A_2). \tag{60}$$

$$\begin{aligned} \underset{s \in \mathbb{R}, x \in \mathbb{R}^2}{\text{minimize}} \quad & s && \text{(61a)} \\ \text{subject to} \quad & A_0 + x_1 A_1 + x_2 A_2 \preccurlyeq s\mathbb{I}. && \text{(61b)} \end{aligned}$$