MSc Data Science Ramp-Up Course in Mathematics
Part: Continuous Optimization

Prof. Dr. Christian Kirches

Summer 2024

The following is a summary of some essential topics that I would have treated in a one-semester BSc class in continuous optimization at TU Braunschweig. It is intentionally written in a brief and informal style. The intention is to transfer general concepts while, at certain times, glancing over the finer details to maintain simplicity and accessibility of the exposition. Almost every topic addressed here deserves a broader and much more detailed discussion. Textbooks on calculus and continuous optimization should be consulted in parallel to reading this summary.
I highly recommend the textbook *Numerical Optimization* by J. Nocedal and S.J. Wright, 2nd edition, Springer, 2006 (ISBN 978-0-387-40065-5). The university libraries has copies. A Springer ebook can be obtained from `https://link.springer.com/book/10.1007/978-0-387-40065-5`.

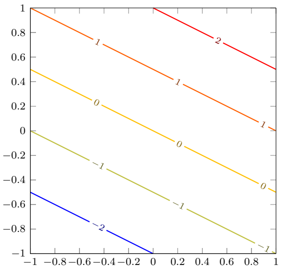
1 Calculus Prerequisites

Level Sets of Functions

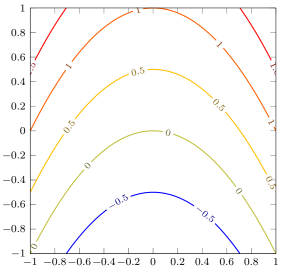
Given a function $f : \mathbb{R}^n \to \mathbb{R}$, a level set $N_c(f)$ of f contains all points in $\mathbb{R}^n$ at which the function has the same value c,

$$N_c(f) := \{x \in \mathbb{R}^n \mid f(x) = c\}.$$

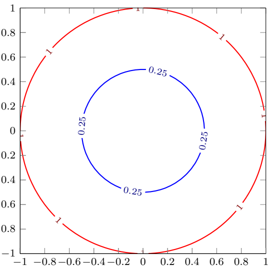
There is one level set for every function value $c \in \mathbb{R}$. Level sets are empty for values c that f never assumes.



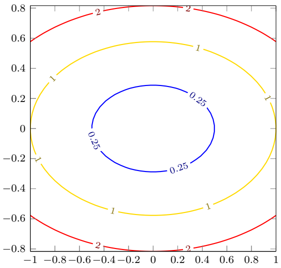
(a) $f(x) = x_1 + 2x_2$ is linear.



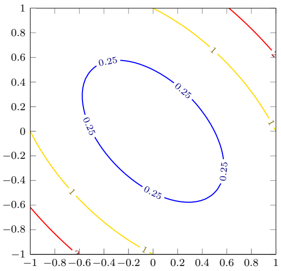
(b) $f(x) = x_1^2 + x_2$ is linear-quadratic.



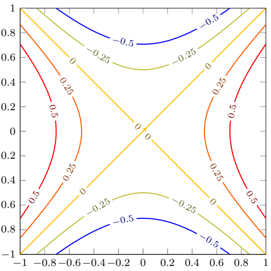
(c) $f(x) = x_1^2 + x_2^2$ is quadratic.



(d) $f(x) = x_1^2 + 3x_2^2$ is quadratic.



(e) $f(x) = x_1^2 + x_2^2 + x_1 x_2$ is quadratic.



(f) $f(x) = x_1^2 - x_2^2$ is quadratic but nonconvex.

Gradient, Jacobian, Directional Derivatives

The **gradient** of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ in a point $x \in \mathbb{R}^n$ is a column vector in $\mathbb{R}^n$ denoted by $\nabla f(x)$, or alternatively a row vector denoted by $f'(x)$. Its elements are the differential quotients

$$(f'(x))_i = (\nabla f(x))_i = \lim_{h\to 0} \frac{1}{h}(f(x + he_i) - f(x)),$$

which are the slopes of f along the n different coordinate axes e_i, $1 \leq i \leq n$.

In practice, the gradient is computed by applying the known rules of differential calculus (product rule, chain rule, etc.), which we assume you know. When visualized by an arrow, a gradient is perpendicular to the level set tangent and points into the direction of steepest ascent.

For a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ we have a gradient for every one of the m component functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $1 \leq j \leq m$. The gradient of f becomes an $n \times m$ matrix composed of columns,

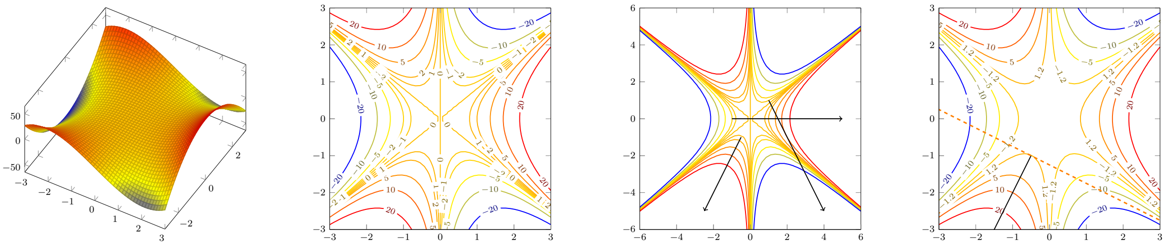
$$\nabla f(x) = \left(\begin{array}{c|c|c} \nabla f_1(x) & \cdots & \nabla f_m(x) \end{array} \right) \in \mathbb{R}^{n\times m},$$

and its transpose is called the **Jacobian**[1] $f'(x)$ of f at x, composed of rows

$$f'(x) = \begin{pmatrix} f_1'(x) \\ \hline \vdots \\ \hline f_m'(x) \end{pmatrix} \in \mathbb{R}^{m\times n}.$$

A **directional derivative** represents the slope of f at $x \in \mathbb{R}^n$ into a direction $d \in \mathbb{R}^n$ that is *not* necessarily one of the coordinate axes e_i. It is given by the dot product ($m = 1$) or matrix-vector product ($m > 1$)

$$\nabla f(x)^T d = \langle \nabla f(x), d\rangle = f'(x)d \in \mathbb{R}.$$



(a) The graph of $f(x) = 2x_1^3 - 3x_1x_2^2$.

(b) Some level sets of f.

(c) Gradients $\nabla f(x)$ in several locations x.

(d) The gradient is a direction of steepest ascent, and is perpendicular to the level set tangent.

Hessian

The **Hessian**[2] of a function $f : \mathbb{R}^n \to \mathbb{R}$ in a point $x \in \mathbb{R}^n$ is the $n \times n$ matrix $\nabla^2 f(x)$ of mixed second order partial differentials

$$(\nabla^2 f(x))_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

If the second derivative of f is continuous, the matrix $(\nabla^2 f(x))$ is symmetric (Schwarz'[3] theorem). The **curvature** of f along the coordinate axes e_i is found on the diagonal,

$$e_i^T\ \nabla^2 f(x)\ e_i = (\nabla^2 f(x))_{ii}.$$

The curvature into a direction $d \in \mathbb{R}^n$ that is *not* necessarily one of the coordinate axes e_i is found by computing the bilinear form

$$d^T\ \nabla^2 f(x)\ d \in \mathbb{R}.$$

This requires computing the matrix-vector product $v = \nabla^2 f(x)\ d$ and then the dot product $d^T v$.

[1] Carl Gustav Jacob Jacobi, German mathematician

[2] Otto Hesse, German mathematician

[3] Hermann Amandus Schwarz, German mathematician

Taylor Expansion, Tangents, Quadratic Models

A twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ admits a TAYLOR[4] expansion

$$f(x+d) = f(x) + \nabla f(x)^T d + \tfrac{1}{2} d^T \nabla^2 f(x) d + o(||d||^2).$$

This involves the function's value, directional derivative, and curvature into a direction $d \in \mathbb{R}^n$. As $d \to 0$, the error term $o(||d||^2)$ approches zero faster than $||d||^2$ does.

The function

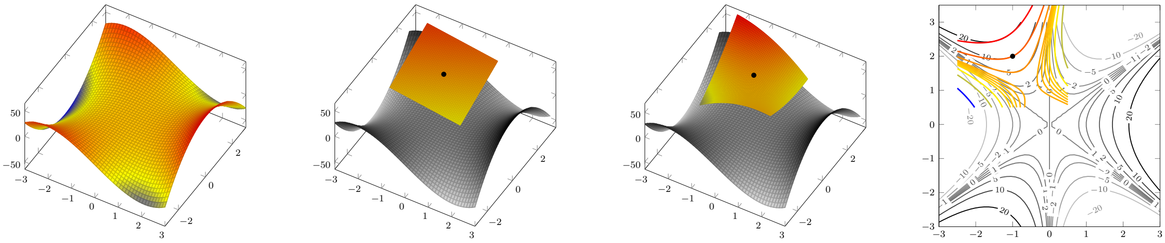
$$\ell(d) = f(x) + \nabla f(x)^T d$$

is a **linear model** of the behavior of the function f, taken in a fixed point $x \in \mathbb{R}^n$, and evaluated when leaving x into direction $d \in \mathbb{R}^n$. It is frequently called a **tangent** to f in x.

The function

$$q(d) = f(x) + \nabla f(x)^T d + \tfrac{1}{2} d^T \nabla^2 f(x) d$$

is a **quadratic model** of the behavior of the function f, taken in a fixed point $x \in \mathbb{R}^n$, and evaluated when leaving x into direction $d \in \mathbb{R}^n$.



(a) The graph of $f(x) = 2x_1^3 - 3x_1x_2^2$.

(b) A linear model of f in a location x.

(c) A quadratic model of f in a location x.

(d) Overlay of level sets of f and the same quadratic model.

A twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$ is vector-valued and its TAYLOR expansion

$$f(x+d) = f(x) + \nabla f(x)^T d + \tfrac{1}{2} d^T \nabla^2 f(x) d + o(||d||^2)$$

works with the gradient matrix $\nabla f(x) \in \mathbb{R}^{n\times m}$ and the Hessian is a third-order tensor $\nabla^2 f(x) \in \mathbb{R}^{n\times n\times m}$. The bilinear form $d^T \nabla^2 f(x) d \in \mathbb{R}^m$ is vector-valued. It is sometimes easier to work with Taylor expansions of the component functions $f_j : \mathbb{R}^n \to \mathbb{R}$, $1 \le j \le m$.

Convex Functions

A function $f : D \to \mathbb{R}$ is convex on a subset $D \subset \mathbb{R}^n$ if one of the following holds:

1. f has nonnegative curvature everywhere on D,

$$\nabla^2 f(x) \text{ positive semidefinite } \forall x \in D.$$

2. *Any* tangent to f never become greater than f on D,

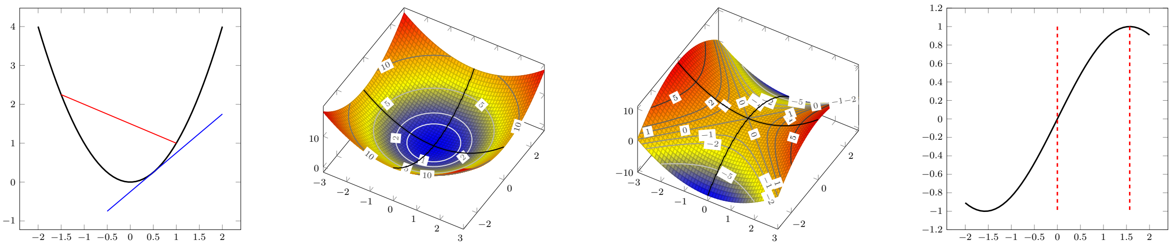
$$f(x) + \nabla f(x)^T (y-x) \le f(y) \ \forall x,y \in D.$$

3. Secants between *any* two points on the graph of f never become smaller than f on D,

$$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y) \ \forall x,y \in D, \lambda \in (0,1).$$

If the inequalities are strict (equiv. if $\nabla^2 f(x)$ is positive definite) on D, the function f is called *strictly convex* on D.

[4] BROOK TAYLOR, British mathematician



(a) A convex function (black) with a tangent (blue) and a secant between two points (red).

(b) $f(x) = x_1^2 + x_2^2$ is convex. The Hessian $\nabla^2 f$ has a double eigenvalue 2.

(c) $f(x) = x_1^2 - x_2^2$ is convex along x_1 but not along x_2, so not a convex function. Eigenvalues of $\nabla^2 f$ are 2 and -2.

(d) $f(x) = \sin(x)$ is convex on $D = [0, \pi]$ but not on $D = \mathbb{R}$.

Subsets of $\mathbb{R}^n$

A point $x \in \mathbb{R}^n$ is called **feasible** with respect to a set $\mathcal{F} \subset \mathbb{R}^n$ if $x \in \mathcal{F}$. It is called **infeasible** with respect to $\mathcal{F}$ otherwise.
Testing feasibility is not a computational concept unless an **outer description** of $\mathcal{F}$ is accessible. We frequently assume that functions $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^k$ exist such that

$$\mathcal{F} := \{x \in \mathbb{R}^n \mid g(x) = 0 \in \mathbb{R}^n,\ h(x) \geq 0 \in \mathbb{R}^m\},$$

i.e., membership of x in $\mathcal{F}$ can be tested by evaluating g and h and checking the signs of their vector return values.

The **boundary** of a set $\mathcal{F}$ is denoted by $\partial\mathcal{F}$ and consists of all point $x \in \mathcal{F}$ with neighborhoods that contain points $y \notin \mathcal{F}$. If $\partial\mathcal{F} \in \mathcal{F}$, then $\mathcal{F}$ is called **closed**. Subsets of $\mathbb{R}^n$ that are bounded and closed are called **compact**. The **interior** $\operatorname{int} \mathcal{F}$ is the set $\mathcal{F} \setminus \partial\mathcal{F}$. If $x \in \partial\mathcal{F}$ implies $x \notin \mathcal{F}$, then $\mathcal{F}$ is called **open**.

A set $\mathcal{C}$ satisfying $\lambda x \in \mathcal{C}$ for all $x \in \mathcal{C}$ and all $\lambda > 0$ is called a **cone**. It is **pointed** if $0 \in \mathcal{C}$. Finite unions and intersections of cones are cones. Finite intersections of convex cones are convex, but unions need not be.

The cone $\mathcal{C}^* = \{y \in \mathbb{R}^n \mid \langle x, y \rangle \geq 0\ \forall x \in \mathcal{F}\}$ is called the **dual cone** of a set $\mathcal{F}$. The cone $\mathcal{C}^\circ = \{y \in \mathbb{R}^n \mid \langle x, y \rangle \leq 0\ \forall x \in \mathcal{F}\}$ is called the **polar cone** of a set $\mathcal{F}$. Both are closed and convex.

2 Unconstrained Optimization Theory

Definition 1. (Unconstrained Nonlinear Program)
The unconstrained minimization problem

$$\boxed{\min_{x \in \mathbb{R}^n} \ f(x)} \tag{MIN}$$

for an *objective function* $f : \mathbb{R}^n \to \mathbb{R}$ is called an *unconstrained nonlinear program.*

Definition 2. (Minimizers)
Let $x^* \in \mathbb{R}^n$. If there exists a number $\varepsilon > 0$ such that

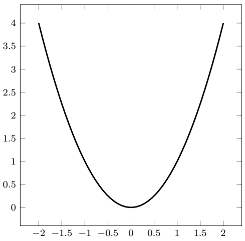
$$f(x^*) \leq f(y) \text{ for all } y \in U_\varepsilon(x) := \{y \in \mathbb{R}^n \mid ||y - x^*|| < \varepsilon\}, \tag{1}$$

then x^* is called a **local minimizer** of (MIN). If the inequality holds strictly on $U_\varepsilon(x) \setminus \{x^*\}$, then x^* is called a **strict local minimizer**. If ε can be chosen arbitrarily large, then x^* is called a **global minimizer**. The value $f(x^*)$ is called a (strict) local/global **minimum**.

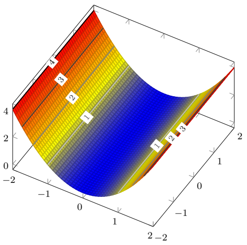
Identifying *global* minimizers is an $\mathcal{NP}$-hard task in general. Assuming that the $\mathcal{P} \neq \mathcal{NP}$ hypothesis is true, this roughly means that on deterministic computers there is no algorithm that finds a global minimizer within a runtime that is "fast", i.e. a polynomial in the number n of unknowns. The problem isn't "unsolvable", though, as there are "slow" algorithms that solve it within a runtime that is exponential in n.

n	$O(\log n)$	$O(n)$	$O(n \log n)$	$O(n^2)$	$O(n^3)$	$O(1.001^n)$	$O(2^{n/100})$
10^2	5 ns	100 ns	461 ns	10 μs	1 ms	1 ns	2 ns
10^3	7 ns	1 μs	7 μs	1 ms	1 s	2.7 ns	1 μs
10^4	9 ns	10 μs	92 μs	100 ms	14 ms	22 μs	–
10^5	12 ns	100 μs	1 ms	10 s	11.5 d	–	–
10^6	14 ns	1 ms	14 ms	17 min	31.7 y	–	–
10^9	21 ns	1 s	21 s	31.7 y	–	–	–

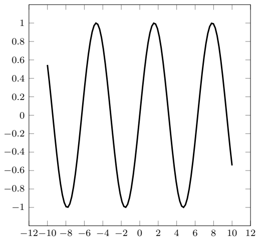
Fast versus slow algorithms. The table shows runtimes depending on n for algorithms of different runtime complexity. It is assumed that one operation takes one nanosecond (10^{-9} seconds, i.e. a machine at 1 GHz clock speed) and that the constant hidden by O is 1. Other choices would introduce a constant factor but would not change the general picture. "–" indicates runtimes certainly in excess of your lifetime.



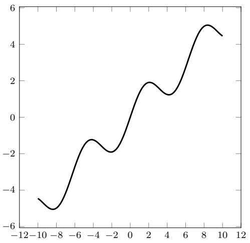
(a) $f(x) = x^2$ has one strict local and global minimizer.



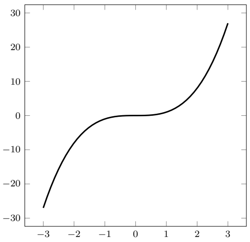
(b) $f(x) = x_1^2$ on $\mathbb{R}^2$ has a global minimizer in $x^* = 0$ that is not strict.



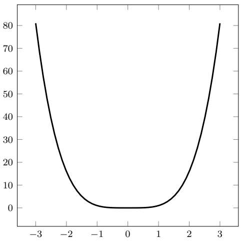
(c) $\sin(x)$ has countably infinitely many strict local minimizers on $\mathbb{R}$, all are strict global.



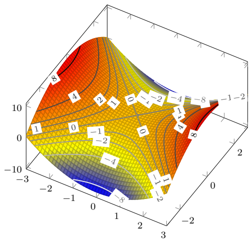
(d) $\sin(x) + \frac{1}{2}x$ has countably infinitely many strict local minimizers on $\mathbb{R}$, none is global.



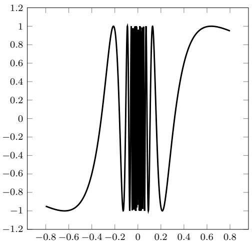
(e) $f(x) = x^3$ has a saddle point in $x^* = 0$, but no minimizers. Still $\nabla f(0) = 0$.



(f) $f(x) = x^4$ has a local and global minimizer in $x^* = 0$ but $\nabla^2 f(0) = 0$.



(g) $f(x) = x_1^2 - x_2^2$ has a saddle point in $x^* = 0$, but no minimizers.



(h) $\sin(1/x)$ has countably infinitely many strict local and global minimizers in the finite interval $[-1, 1]$.

Hence we only discuss local minimizers. Even the definition of a *local* minimizer is not suitable for computation as it requires verifying the condition $f(x^*) \leq f(y)$ *for uncountably infinitely many* choices of y in a certain set. In its place, the following necessary condition is used to characterize local minimizers.

Proposition 3. (Necessary Optimality Condition of First Order)
Let x^* be a local minimizer of (MIN). Then the following holds:

$$\nabla f(x^*) = 0.$$

The condition is not sufficient, which motivates the following definition.

Definition 4. (Stationary Point)
If a point $x \in \mathbb{R}^n$ satisfies $\nabla f(x) = 0$, then x is called a **stationary point** of (MIN).

As a consequence, all local minimizers are stationary points. Vice versa, stationary points are the *candidates* for local minimizers, but some of them may not actually be minimizers (cf. examples (a) and (e)). In general, a gap remains between stationarity and local minimizers. The situation is much better for convex functions f.

Proposition 5. (Necessary and Sufficient Optimality Condition of First Order for Convex Functions)
Let f be convex. If and only if $\nabla f(x^*) = 0$ then x^* is a local and global minimizer of (MIN).

Compare examples (a) and (f). Moreover, the second order necessary optimality condition is always satisfied for convex functions. By convexity, a local minimizer also is a global one. If f is even *strictly convex*, the sufficient optimality condition of second order is always satisfied and the local minimizer is both unique and global. None of this however means that convex optimization problems can be solved exactly in polynomial time.

To try and tell local minimizer apart from stationary points, curvature information from the Hessian of f may be used to state second order conditions.

Proposition 6. (Necessary Optimality Condition of Second Order)
Let x^* be a local minimizer of (MIN). Then the following holds:

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \text{ is positive semidefinite.}$$

Again, all local minimizer satisfy these conditions. However, there may be additional points that satisfy them without being local minimizers (compare examples (a), (e) and (f)). Curvature information also allows to formulate a sufficient condition.

Proposition 7. (Sufficient Optimality Condition of Second Order)
If a point $x^* \in \mathbb{R}^n$ satisfies

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) \text{ is positive definite,}$$

then x^* is a strict local minimum of (MIN).

Sufficient conditions, if satisfied, certify the presence of a local minimizer. If not satisfied, no statement is made, i.e. we *must not conclude* that the point in question is *not* a local minimizer. There are local minimizers that don't satisfy sufficient conditions, cf. example (f).

Unconstrained Optimization Algorithms

We cannot generally expect to describe local minimizers of (MIN) in terms of an algebraic expression (this is possible for fourth-order polynomials only). Instead, we can only hope to approximate them using iterative methods. The most basic one is gradient descent.

Definition 8. (Steepest Descent Direction)
The unit direction $d \in \mathbb{R}^n$ that minimizes the directional derivative

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \nabla f(x)^T d \\ \text{s.t.} \quad & \|d\| = 1 \end{aligned} \tag{2}$$

is called the direction of *steepest descent* of f in the point x w.r.t. the norm $||\cdot||$. For the Euclidean norm $||\cdot||_2$, it is given by

$$d = -\frac{\nabla f(x)}{||\nabla f(x)||_2}. \tag{3}$$

This approach minimizes the first order TAYLOR term. The result shows that the gradient of a function is a direction of steepest ascent, and the antigradient is a direction of steepest descent. As we're looking for minimizers, this motivates gradient descent.

Definition 9. (Gradient Descent)
An algorithm producing a sequence of iterates $\{x^{(k)}\} \subset \mathbb{R}^n$ by starting in $x^{(0)} \in \mathbb{R}^n$ and letting

$$x^{(k+1)} \leftarrow x^{(k)} - \alpha^{(k)} \nabla f(x^{(k)}) \tag{4}$$

with step sizes $\alpha^{(k)} \in (0, 1]$ is called a *gradient descent* algorithm.

Gradient descent can be understood as replacing f (which we can't minimize exactly in one step) by its linear model $\ell(d)$ in $x^{(k)}$. Step sizes (called learning rates in ML) may be determined using line search.

Definition 10. (ARMIJO Line Search)
For a point $x \in \mathbb{R}^n$ and a descent direction $d \in \mathbb{R}^n$ with $\nabla f(x)^T d < 0$, a step size $\alpha \in (0, 1]$ satisfying

$$f(x + \alpha d) < f(x) \tag{5}$$

gives montonically decresing functions values $\{f(x^{(k)})\}$. A step size satisfying

$$f(x + \alpha d) < f(x) + \alpha\gamma \nabla f(x)^T d \tag{6}$$

for a fixed tuning factor $\gamma \in (0, 1]$ is called an ARMIJO *step size.* The procedure of finding one is called ARMIJO *Line Search*, and may, for example, proceed by trying a monotonically descending sequence of step sizes $\alpha \in \{\beta^\ell\}$ for $\ell \geq 0$ and $\beta \in (0, 1)$, e.g. $\beta = 0.5$.

We are interested in the speed of continuous optimization algorithms, which is measured in both the number of iterations required and the CPU time per iteration required. The number of iterations depends on the rate of convergence of an algorithm.

Definition 11. (Rates of Convergence)
For a convergent sequence $\{x^{(k)}\} \subset \mathbb{R}^n$ with limit $x^* \in \mathbb{R}^n$, we say that it is *linearly convergent* if there is an iteration index $\ell \in \mathbb{N}$ such that for all later iteration indices $k \geq \ell$ the following holds:

$$\frac{||x^{(k+1)} - x^*||}{||x^{(k)} - x^*||} \leq \kappa \text{ for some constant } \kappa \in [0, 1). \tag{7}$$

We say that is *superlinearly convergent* if

$$\frac{||x^{(k+1)} - x^*||}{||x^{(k)} - x^*||} \leq \kappa^{(k)} \text{ for a sequence } \{\kappa^{(k)}\} \subset [0, 1). \tag{8}$$

and $\kappa^{(k)} \to 0$ as $k \to \infty$. We say that is *quadratically convergent* if

$$\frac{||x^{(k+1)} - x^*||}{||x^{(k)} - x^*||^2} \leq \omega \text{ for some constant } \omega \in [0, \infty). \tag{9}$$

Proposition 12. (Global Convergence of Line Search Gradient Descent)
The gradient descent algorithm with Armijo line search is linearly convergent from any starting point $x^{(0)} \in \mathbb{R}^n$.

This means the method is comparably slow and will typically take many thousands of iterations, the rate κ depending on both the nonlinearity and the anisotropy (eigenvalues of the Hessian) of the problem. Faster methods make use of second order information.

By working with the quadratic model $q(d)$ obtained from a second order TAYLOR expansion we obtain

$$f(x^{(k)} + d) - f(x^{(k)}) \approx \nabla f(x^{(k)})d + \tfrac{1}{2}d^T \nabla^2 f(x^{(k)})d$$

This motivates minimizing the right-hand side to find a direction d that will, in general, be different from the steepest descent direction:

$$\min_{d \in \mathbb{R}^n} \; \nabla f(x^{(k)})d + \tfrac{1}{2}d^T \nabla^2 f(x^{(k)})d$$

If $\nabla^2 f(x^{(k)})$ is positive definite (c.f. the sufficient condition of second order), the result is

$$d^{(k)} = -\left(\nabla^2 f(x^{(k)})\right)^{-1} \nabla f(x^{(k)}).$$

The difference to gradient descent is obvious: The inverse of the Hessian applys a basis transformation to the antigradient. Only if the curvature information happens to be identity will the two approaches yield identical steps.

Definition 13. (NEWTON[5] Type Directions)
Given (an approximation of) the Hessian of the objective $B \approx \nabla^2 f(x)$ and assuming that B is invertible, the direction $d \in \mathbb{R}^n$ that minimizes the directional derivative for a direction of unit length in the B-norm

$$\begin{aligned} &\min_{d \in \mathbb{R}^n} \; \nabla f(x)^T d \\ &\text{s.t. } ||d||_{B,2} = 1 \end{aligned} \tag{10}$$

is called the **Newton-type direction** of f in the point x for (the approximation) B (with respect to the Euclidean norm). It is given by

$$d = -B^{-1} \frac{\nabla f(x)}{||\nabla f(x)||_2}. \tag{11}$$

[5] ISAAC NEWTON, British mathematician

Definition 14. (Newton-Type Method)
An algorithm producing a sequence of iterates $\{x^{(k)}\}$ by letting

$$x^{(k+1)} \leftarrow x^{(k)} - \alpha^{(k)} {B^{(k)}}^{-1} \nabla f(x^{(k)}) \tag{12}$$

with step sizes $\alpha^{(k)} \in (0, 1]$ is called a Newton-*Type* algorithm.

If $B = \nabla^2 f(x)$, we call this the *exact or classical* Newton *method*. Depending on the source and type of an approximation $B \approx \nabla^2 f(x)$, one speaks of *Quasi*-Newton or Newton-Type methods. Popular approximations include the BFGS update, the SR-1 update, or the Gauss-Newton approximation in case of least-squares objectives f.

Proposition 15. (Global Convergence of Newton-Type Methods)
A Newton-Type algorithm with Armijo line search is linearly convergent from any starting point $x^{(0)} \in \mathbb{R}^n$ if $B = Id$ is choosen whenever the Newton-type direction $d_N = -{B^{(k)}}^{-1} \nabla f(x^{(k)})$ is not a descent direction. In a suitable (possibly small) neighborhood of a minimizer x^*, it is superlinearly convergent if the $\{B^{(k)}\}$ are certain approximations of the exact Hessian, and it is quadratically convergent if the $\{B^{(k)}\}$ are the exact Hessians of the objective f.

Constrained Optimization

Definition 16. Constrained Nonlinear Program
The constrained minimization problem

$$\boxed{\begin{aligned} \min_{x\in\mathbb{R}^n} \ & f(x) \\ \text{s.t. } & g(x) = 0 \\ & h(x) \geq 0 \end{aligned}} \tag{NLP}$$

for an *objective function* $f : \mathbb{R}^n \to \mathbb{R}$, *equality constraints* $g : \mathbb{R}^n \to \mathbb{R}^m$ and *inequality constraints* $h : \mathbb{R}^n \to \mathbb{R}^k$ is called an *(constrained) nonlinear program*, short NLP.

Definition 17. (Feasible Point, Feasible Set)
A point $\bar{x} \in \mathbb{R}^n$ satisfying $g(\bar{x}) = 0$ and $h(\bar{x}) \geq 0$ (component-wise) is called **feasible point** of (NLP). The set of all feasible points

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid g(x) = 0,\ h(x) \geq 0\} \tag{13}$$

is called the **feasible set** of (NLP).

Definition 18. (Active Inequality, Active Set)
Let $\bar{x} \in \mathbb{R}^n$ be a feasible point. If an inequality indexed by $i \in \{1, \dots, k\}$ is satisfied with equality,

$$h_i(\bar{x}) = 0, \tag{14}$$

then we call inequality h_i **active in** $\bar{x}$. We call it **inactive in** $\bar{x}$ otherwise. The set of indices of inequalities active in $\bar{x}$,

$$\mathcal{A}(\bar{x}) = \{1 \leq i \leq k \mid h_i(\bar{x}) = 0\} \subseteq \{1, \dots, k\} \tag{15}$$

is called the **active set** of $\bar{x}$. Its complement in the index set $\{1, \dots, k\}$ is called the **inactive set**.

Definition 19. (Local Minimum)
If for a feasible point $\bar{x} \in \mathcal{F}$ there exists a number $\varepsilon > 0$ such that

$$f(\bar{x}) \leq f(y) \text{ for all } y \in U_\varepsilon(\bar{x}) \cap \mathcal{F} := \{y \in \mathcal{F} \mid ||y - \bar{x}|| < \varepsilon\}, \tag{16}$$

then $\bar{x}$ is called a *local minimum* of (NLP). If the inequality is even strict, $\bar{x}$ is called a *strict local minimum.*

If f is once continuously differentiable, then a first order necessary condition characterizes local minimizers. The complicating issue are the equalities, and the inequalities active in x^*. We can ask for nonnegative gradients of the objective f into only those directions d that point into the interior of the feasible set. These directions are captured by the tangent cone.

Definition 20. (Tangent Cone)
For a feasible point $\bar{x} \in \mathcal{F}$, the **tangent cone** of $\mathcal{F}$ in the point $\bar{x}$ is given by

$$\begin{aligned} \mathcal{T}(\mathcal{F}, \bar{x}) := \{d \in \mathbb{R}^n \mid \exists\ \{t^{(k)}\} \to \infty \text{ and } \{x^{(k)}\} \subset \mathcal{F} \text{ with } x^{(k)} \to \bar{x} \\ \text{such that } t^{(k)} \cdot (x^{(k)} - \bar{x}) \to d \text{ as } k \to \infty.\}. \end{aligned} \tag{17}$$

The tangent cone is difficult to work with, as it is defined by asking for existence of two infinite sequences. The linearized cone may replace it under certain conditions, and is easier to work with as it makes use of gradients of the constraint functions instead.

Definition 21. (Linearized Cone)
For a feasible point $\bar{x} \in \mathcal{F}$, the **linearized cone** of $\mathcal{F}$ in $\bar{x}$ is given by

$$\mathcal{L}(\mathcal{F}, \bar{x}) := \{d \in \mathbb{R}^n \mid \nabla g(\bar{x})^T d = 0,\ \nabla h(\bar{x})^T d \geq 0\}. \tag{18}$$

One can show $\mathcal{T}(\mathcal{F}, \bar{x}) \subset \mathcal{L}(\mathcal{F}, \bar{x})$, but the linearized cone may be larger. It is a pointed convex cone. We work under the assumption that it is possible to replace the tangent cone by the linearized cone, i.e. they are equivalent. Whenever this situation is present, this is called **to have a constraint qualification** at $\bar{x}$.

Definition 22. (Constraint Qualification)
We say that a **constraint qualification** holds in a feasible point $\bar{x} \in \mathcal{F}$ of (NLP) if

$$\mathcal{L}(\mathcal{F}, \bar{x}) = \mathcal{T}(\mathcal{F}, \bar{x}).$$

We can now formulate a first order necessary optimality condition for (NLP).

Proposition 23. (Necessary Optimality Condition of First Order)
Let x^* be a local minimizer of (NLP) and let a constraint qualification hold in x^*. Then the following holds:

$$\begin{aligned} &1.\ g(x^*) = 0,\ h(x^*) \geq 0, \\ &2.\ \nabla f(x^*)^T d \geq 0 \qquad \text{for all } d \in \mathcal{L}(\mathcal{F}, x^*), \\ &\qquad\qquad i.e. \text{ for all } d \in \mathbb{R}^n \text{ with } \nabla g(x^*)^T d = 0,\ \nabla h(x^*)^T d \geq 0. \end{aligned} \tag{19}$$

As before, the condition is not sufficient. Hence the following definition.

Definition 24. (Stationary Point)
If a constraint qualification holds in a feasible point $\bar{x} \in \mathbb{R}^n$ and

$$\nabla f(\bar{x})^T d \geq 0 \text{ for all } d \in \mathcal{L}(\mathcal{F}, \bar{x})$$

then $\bar{x}$ is called a *stationary point* of (NLP).

As before, necessary and sufficient conditions of second order may be stated using curvature information from the Hessian.

Proposition 25. (Necessary Optimality Condition of Second Order)
Let x^* be a local minimizer of (NLP) and let a constraint qualification hold at x^*. Then the following holds:

$$\begin{aligned} &1.\ x^* \text{ is a stationary point,} \\ &2.\ \nabla^2 f(x^*) \text{ is positive semidefinite on the null-space of the active constraints, i.e.} \\ &\quad d^T \nabla^2 f(x^*) d \geq 0 \text{ for all } d \in \mathcal{L}(\mathcal{F}, x^*) \end{aligned} \tag{20}$$

Proposition 26. (Sufficient Optimality Condition of Second Order)
If a stationary point $x \in \mathbb{R}^n$ of (NLP) satisfies

$$\begin{aligned} &\nabla^2 f(x^*) \text{ is positive definite on the null-space of the active constraints, i.e.} \\ &d^T \nabla^2 f(x^*) d > 0 \text{ for all } d \in \mathcal{L}(\mathcal{F}, x^*) \end{aligned} \tag{21}$$

then it is a strict local minimum of (NLP).

The linearized cone is computationally difficult to work with as one has to certify a property *for all* directions d in a set. An *equivalent* necessary optimality condition is given by the Karush-Kuhn-Tucker[6] theorem, which works using *existence*, i.e. we only have to compute *a single object* with a given property. This is one of the most important theorems in continuous optimization and you need to know it by heart.

Theorem 27. (Karush-Kuhn-Tucker)
Let x^* be a local minimum of (NLP) and let a constraint qualification hold in x^*. Then, there exist Lagrange[7] multiplier vectors $\lambda^* \in \mathbb{R}^m$ and $\mu^* \in \mathbb{R}^k$ such that the following four conditions hold:

1. Stationarity: $\nabla f(x^*) = \nabla g(x^*)\lambda^* = \nabla h(x^*)\mu^*$,
2. (Primal) Feasibility: $g(x^*) = 0$, $h(x^*) \geq 0$,
3. Optimality: $\mu^* \geq 0$,
4. Complementary slackness: $h(x^*)^T \mu^* = 0$.

[6] William Karush, U.S. mathematician; Harold W. Kuhn, U.S. mathematician; Albert W. Tucker, Canadian mathematician

[7] Giuseppe Lodovico (or Luigi) Lagrangia (or De La Grange), later Joseph-Louis Lagrange, Italian-French mathematician

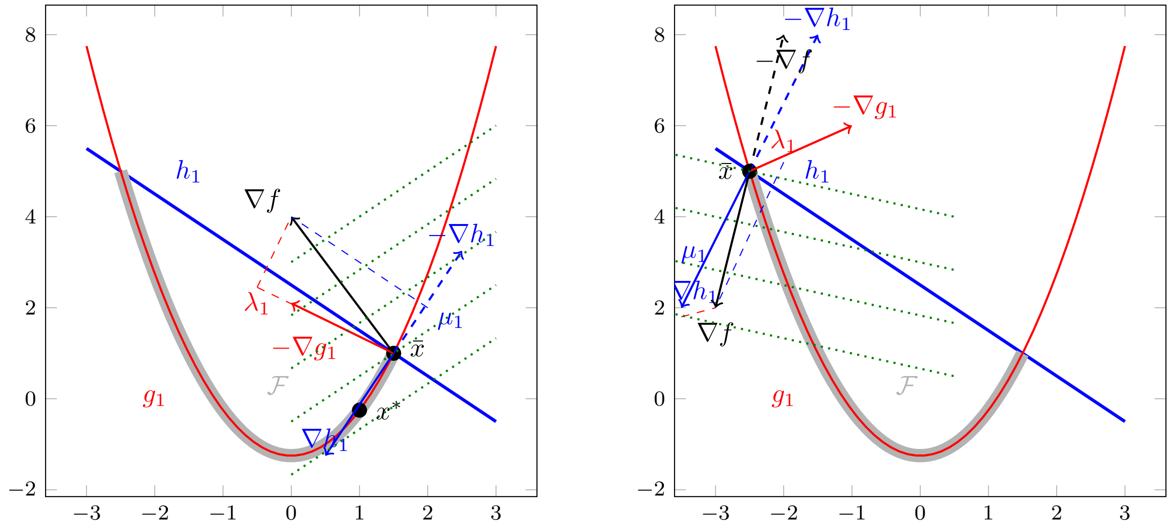
Complementary slackness is sometimes called “complementarity” for brevity and may be read as an “inclusive or” condition. For every inequality $i \in \{1, \dots, k\}$ one of the following possibilities holds: $h_i(x^*) = 0$ (active) and $\mu_i^* \geq 0$, or $h_i(x^*) > 0$ (inactive) and $\mu_i^* = 0$. Optimality is sometimes called “dual feasibility”, the LAGRANGE multipliers are referred to as “dual vectors” or “dual solution”.

The expression that shows up in the stationarity condition is frequently abbreviated as the gradient of the LAGRANGIAN of (NLP),

$$\mathcal{L}(x, \lambda, \mu) := f(x) - \lambda^T g(x) - \mu^T \nabla h(x). \tag{22}$$

Note the flip of signs such that condition 1. reads $\nabla_x \mathcal{L}(x, \lambda, \mu) = 0$. Feasibility reads $\nabla_\lambda \mathcal{L}(x, \lambda, \mu) = 0$, $\nabla_\mu \mathcal{L}(x, \lambda, \mu) \leq 0$. The latter is the reason why the LAGRANGIAN is sometimes defined with a “+” instead of a “−”. This would flip the signs of λ and μ, and optimality in the KKT theorem would read $\mu \leq 0$.

Geometrically, condition 1. in the KKT theorem asks to represent ∇f as a linear combination of ∇g and ∇h, where the linear factors of ∇h must be nonnegative (condition 3.). This is what we show in the pictures below. Equivalently, condition 1. asks use ∇f, $-\nabla g$ and $-\nabla h$ to form a closed loop where the linear factors of $-\nabla h$ must be nonnegative (condition 3.).



(a) The point $\bar{x}$ is feasible and $\nabla f(\bar{x})$ has a representation in terms of the active constraint normals ∇g_1 and ∇h_1. We can see $\lambda_1 > 1$ and $\mu_1 < 0$. Hence $\bar{x}$ is **not** a KKT point. Indeed x^*, for example, is feasible and has a smaller objective function value.

(b) The point $\bar{x}$ is feasible and $\nabla f(\bar{x})$ is represented in terms of the active constraint normals with a $\mu_1 \geq 0$. Hence $\bar{x}$ is a KKT point.

Constrained Optimization Algorithms

Penalty Methods

Constrained nonlinear problems can be transformed to unconstrained ones by introducing a penalty function,

$$\phi_1(x) = ||g(x)||_1 + ||[h(x)]^-||_1,$$

wherein $[\,\cdot\,]^-$ clamps the positive components of $h(x)$ (corresponding to feasibility) to zero and keeps the negative ones. On the feasible set $\mathcal{F}$, we have $\phi_1(x) = 0$. Outside of $\mathcal{F}$, it is positive and measures the amount of infeasibility. One can now solve the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} \; f(x) + \theta \cdot \phi_1(x)$$

for a penalty parameter $\theta > 0$. The solution can be shown to be exact for (NLP) if θ is chosen sufficiently large. However, the function ϕ_1 and hence also the objective function are not differentiable.

One can also introduce the differentiable penalty function

$$\phi_2(x) = ||g(x)||_2^2 + ||[h(x)]^-||_2^2,$$

and solve $\min_{x\in\mathbb{R}^n} \ f(x) + \theta \cdot \phi_2(x)$. Now, however, the solution will only be an approximate one, no matter how big $\theta > 0$ is chosen.

Sequential Quadratic Programming

NEWTON-Type methods are easily generalized to NLPs with only equality constraints, as follows. The necessary optimality conditions of first order read

$$\begin{aligned} 0 &= \nabla f(x) - \nabla g(x)\lambda \\ 0 &= g(x). \end{aligned} \tag{23}$$

This is a root finding problem $F(x, \lambda) = 0$ with gradient

$$\nabla_{(x,\lambda)} F(x, \lambda) = \begin{pmatrix} \nabla^2 f(x) - \nabla^2 g(x)\lambda & -\nabla g(x) \\ \nabla g(x)^T & 0 \end{pmatrix}. \tag{24}$$

An exact NEWTON iteration reads

$$\begin{pmatrix} x^{(k+1)} \\ \lambda^{(k+1)} \end{pmatrix} \leftarrow \begin{pmatrix} x^{(k)} \\ \lambda^{(k)} \end{pmatrix} - \alpha^{(k)} \underbrace{\begin{pmatrix} \nabla^2 f(x) - \nabla^2 g(x)\lambda & -\nabla g(x) \\ \nabla g(x)^T & 0 \end{pmatrix}^{-1}}_{=\nabla_{(x,\lambda)} F(x,\lambda)} \underbrace{\begin{pmatrix} \nabla f(x) - \nabla g(x)\lambda \\ g(x) \end{pmatrix}}_{=F(x,\lambda)} \tag{25}$$

and enjoys the same convergence properties as in the unconstrained case.

One easily proves that the minimizer $d \in \mathbb{R}^n$ of the following **quadratic minimization problem (QP)**

$$\begin{aligned} \min_d \ & \tfrac{1}{2} d^T \ H(x^{(k)}, \lambda^{(k)}) \ d + d^T c(x^{(k)}, \lambda^{(k)}) \\ \text{s.t.} \ & g(x^{(k)}) + \nabla g(x^{(k)})^T d = 0 \qquad \mid \Delta\lambda \end{aligned} \tag{EQP}$$

with quadratic objective part $H(x, \lambda) = \nabla^2 f(x) - \nabla^2 g(x)\lambda$ and linear objective part $c(x, \lambda) = \nabla f(x) - \nabla g(x)\lambda$ is the exact NEWTON step we just computed above. The optimal LAGRANGE multipliers $\Delta\lambda$ for the equality constraint in (EQP) provide the exact NEWTON step for the LAGRANGE multipliers $\lambda^{(k)}$, just as we computed above. As we solve a sequence of QPs instead of a sequence of linear systems of equations, this approach is called **sequential quadratic programming (SQP)** and is one of the most important algorithmic approaches to solving NLPs.

The merit of this equivalence lies with the fact that we can easily add inequality constraints to (EQP), but not to the root finding system in NEWTON's method. Hence, one solves

$$\begin{aligned} \min_d \ & \tfrac{1}{2} d^T \ H(x^{(k)}, \lambda^{(k)}) \ d + d^T c(x^{(k)}, \lambda^{(k)}) \\ \text{s.t.} \ & g(x^{(k)}) + \nabla g(x^{(k)})^T d = 0 \qquad \mid \Delta\lambda \\ & h(x^{(k)}) + \nabla h(x^{(k)})^T d \geq 0 \qquad \mid \Delta\mu \end{aligned} \tag{IQP}$$

to compute a step $(d, \Delta\lambda, \Delta\mu)$ in every iteration using, for example, an **active-set method**.

The primal active set method is one of a number of active set approaches to solve the convex IQPs

$$\begin{aligned} \min_d \ & \tfrac{1}{2} d^T H d + d^T c \\ \text{s.t.} \ & b_i + a_i^T d = 0, \quad i \in \mathcal{E} \mid \lambda_{\mathcal{E}} \\ & b_i + a_i^T d \geq 0, \quad i \in \mathcal{I} \mid \lambda_{\mathcal{I}} \end{aligned}$$

with H positive semidefinite. They arise in the SQP method: We dispose of iteration indices k and write $H = H(x^{(k)}, \lambda^{(k)})$, $c = c(x^{(k)}, \lambda^{(k)})$. For the constraint matrix A with columns vector a_i and for the vector b, we let

$$b = \begin{pmatrix} g(x^{(k)}) \\ h(x^{(k)}) \end{pmatrix} \in \mathbb{R}^{m+k}, \quad A^T = \begin{pmatrix} a_1 \\ \vdots \\ a_{k+m} \end{pmatrix} = \begin{pmatrix} \nabla g(x^{(k)})^T \\ \nabla h(x^{(k)})^T \end{pmatrix} \in \mathbb{R}^{(m+k)\times n}.$$

LAGRANGE multipliers for equalities and inequalities in (IQP) are denoted by $\lambda_{\mathcal{E}}$ and $\lambda_{\mathcal{I}}$, i.e. $\lambda \in \mathbb{R}^{m+k}$, such that we can write λ instead of (λ, μ). When we restrict A^T to contain the rows associated with equalities and *active* inequalities only, we write $A_{\mathcal{W}}^T$ if $\mathcal{W}$ is the active set.

The (Primal) Active Set Method for Convex IQPs

1. Pick a starting guess $x^{(0)}$ feasible for (IQP) and a working set $\mathcal{W}^{(0)} \subseteq \mathcal{A}(x^{(0)})$ such that the row rank of the active constraints' gradient $A^T_{\mathcal{W}^{(0)}}$ is full.
2. For $\ell \geq 0$ (QP iterations):
   (a) Solve the EQP associated with the working set $\mathcal{W}^{(\ell)}$:

   $$\begin{aligned} d^{(\ell)} = \text{argmin } & \tfrac{1}{2} d^T H d + d^T (H x^{(\ell)} + c) \\ \text{s.t. } & a_i^T d = 0 \text{ for all } i \in \mathcal{E} \cup \mathcal{W}^{(\ell)} \end{aligned}$$

   (As shown above, this is just a linear system of equations)

   (b) If $d^{(\ell)} \neq 0$:
      i. Determine the step length $\alpha^{(\ell)} \in [0,1]$ that hits the closest inactive inequality, $a^T_{j*}(x^{(\ell)} + \alpha^{(\ell)} d^{(\ell)}) = b_{j^*}$, $j^* \notin \mathcal{E} \cup \mathcal{W}^{(\ell)}$ (if $\alpha^{(\ell)} = 1$ an index j^* need not exist).
      ii. Let $x^{(\ell+1)} \leftarrow x^{(\ell)} + \alpha^{(\ell)} d^{(\ell)}$, $\mathcal{W}^{(\ell+1)} \leftarrow \mathcal{W}^{(\ell)} \cup \{j^*\}$.

   (c) Otherwise:
      i. Determine LAGRANGE multipliers $\lambda^{(\ell)}$ zu $\mathcal{W}^{(\ell)}$
      ii. Stop if $\lambda_j^{(\ell)} \geq 0$ for all $j \in \mathcal{W}^{(\ell)}$. (x^*, λ^*) is a KKT point of (QP).
      iii. Let $j^* = \text{argmin}\{\lambda_j^{(\ell)} \mid j \in \mathcal{W}^{(\ell)}\}$. This ensures $\lambda_{j^*}^{(\ell)} < 0$.
      iv. Let $x^{(\ell+1)} \leftarrow x^{(\ell)}$, $\mathcal{W}^{(\ell+1)} \leftarrow \mathcal{W}^{(\ell)} \setminus \{j^*\}$.

For this to be effective across the potentially large number of possible active sets, a lot of linear algebra technology goes into step 2(a) and makes implementations of the active set method quite involved in practice.

Global Convergence of SQP Methods
Step sizes $\alpha^{(k)}$ in SQP iterations must be computed to achieve descent with respect to *both* the objective f *and* the infeasibility of the constraints g and h using. This can be done by again using a penalty function for some sufficiently large $\theta > 0$, and by performing line search on it.

Further Reading
Many variants of SQP exist. Other approaches to solving constrained nonlinear optimization problems comprise *Interior Point* methods and *Augmented Lagrangian* methods, developed after SQP. There are globalization approaches besides line search, for example *trust regions*, and *filters*. Approximating Hessians (curvature) is important for speed, and *Quasi*-NEWTON methods such as BFGS, SR-1, or GAUSS-NEWTON do so. Using exact Hessians may give rise to nonconvex QPs, which require extra care. Many feasible sets have a particular geometry that can be exploited using *projections*, giving rise to projected gradient descent and projected NEWTON methods. Lack of differentiability can sometimes be overcome using *smoothing* or using *semi-smooth* methods.