Numerical Optimization

Lecture Notes, Summer 2019

Ludwig Maximilian University of Munich

Julian Wagner

Contents

Chapter 1

Mathematical Foundations

In this chapter we focus on basic concepts and notations that are necessary for the general understanding of this lecture. More precisely, we

- review fundamentals of linear algebra and analysis,
- introduce the concept of convexity,
- introduce fundamental terms and characteristics of optimization problems,
- give a brief introduction to numerical differentiation with finite differences.

1.1 Review of Linear Algebra and Analysis

Vectors, Matrices and Norms

If not stated otherwise, we equip the $\mathbb{R}^n$ with the *euclidean norm*

$$\|x\| = \left(\sum_{i=0}^{n} x_i^2\right)^{1/2} \qquad \forall x \in \mathbb{R}^n,$$

which is induced by the *standard inner product*

$$\langle x, y\rangle = x^T y = \sum_{i=1}^{n} x_i y_i \qquad \forall x, y \in \mathbb{R}^n.$$

Especially, it is

$$\|x\| = \sqrt{\langle x, x\rangle}.$$

For $x \in \mathbb{R}^n$ and $\varepsilon > 0$ we call

$$\mathcal{B}_\varepsilon = \{y \in \mathbb{R}^n : \|x - y\| < \varepsilon\}$$

the *open ball* at x with radius ε and

$$\overline{\mathcal{B}}_\varepsilon = \{y \in \mathbb{R}^n : \|x - y\| \leq \varepsilon\}$$

the *closed ball* at x with radius ε.

For a sequence $(x^{(k)})_{k\in\mathbb{N}_0}$ we say that it *converges* to some $x^* \in \mathbb{R}^n$ if

$$\forall \varepsilon > 0 \quad \exists\, K_\varepsilon \quad \text{such that} \quad \|x^{(k)} - x^*\| < \varepsilon \quad \forall k > K_\varepsilon$$

and write

$$x^{(k)} \overset{k\to\infty}{\longrightarrow} x^* \quad \text{or} \quad \lim_{k\to\infty} x^{(k)} = x^*.$$

In this case we say that the *rate of convergence* is

1. *sublinear* if
$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 1,$$
2. *linear* if
$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} < 1,$$
3. *superlinear* if
$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 0,$$
4. *quadratic* if
$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|^2} < \infty.$$

For convergence rates we frequently use the *Landau-Symbols* (O-Notation), i.e.

- $f = o(g) \quad \text{if} \quad \lim_{x\to a} \left|\frac{f(x)}{g(x)}\right| = 0$
- $f = \mathcal{O}(g) \quad \text{if} \quad \lim_{x\to a} \left|\frac{f(x)}{g(x)}\right| < \infty$

for some functions f and g. For example, if $x^{(k)} \overset{k\to\infty}{\longrightarrow} x^*$ quadratically, then

$$\underbrace{\|x^{(k+1)} - x^*\|}_{f(k)} = \mathcal{O}(\underbrace{\|x^{(k)} - x^*\|^2}_{g(k)}).$$

On the space of quadratic matrices $\mathbb{R}^{n\times n}$ we use the *induced matrix norm*

$$\|A\| := \max_{\|x\|=1} \|Ax\| \quad \forall A \in \mathbb{R}^{n\times n}$$

for which it holds

$$\|Ax\| \leq \|A\| \cdot \|x\| \quad \forall x \in \mathbb{R}^n$$

and

$$\|A\| = \sqrt{\varrho(A^\top A)},$$

where $\varrho(\cdot)$ denotes the *spectral radius*, i.e.

$$\varrho(A) = \max\{|\lambda_1|, \ldots, |\lambda_n|\},$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A.

We call a matrix $A \in \mathbb{R}^{n\times n}$

- *symmetric* if $A^\top = A$,
- *positive semidefinite* if $x^\top Ax \geq 0 \quad \forall x \in \mathbb{R}^n$,
- *positive definite* if $x^\top Ax > 0 \quad \forall x \in \mathbb{R}^n,\ x \neq 0$.

For symmetric matrices all eigenvalues are real and it holds

$$\lambda_{\min} \|x\|^2 \ \leq \ x^T Ax \ \leq \ \lambda_{\max} \|x\|^2 \quad \forall x \in \mathbb{R}^n,$$

where $\lambda_{\max}$ and $\lambda_{\min}$ denote the largest and smallest eigenvalue of A, respectively. Thus, a symmetric matrix is positive definite (positive semidefinite) if $\lambda_{\min} > 0$ ($\lambda_{\min} \geq 0$).

If A is symmetric and positive definite it holds

$$\|A\| = \lambda_{\max} \quad \text{and} \quad \|A^{-1}\| = \frac{1}{\lambda_{\min}}.$$

Further, A defines an inner product

$$\langle x, y\rangle_A = x^\top Ay \quad \forall x, y \in \mathbb{R}^n$$

as well as a norm

$$\|x\|_A = \sqrt{\langle x,\, x\rangle_A} = \sqrt{x^\top Ax} \quad \forall x \in \mathbb{R}^n.$$

Differentiation in $\mathbb{R}^n$

A function

$$f\colon \mathbb{R}^n \to \mathbb{R}^m, \quad x = (x_1, \ldots, x_n)^\top \mapsto \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix}$$

is called

- *continuous*, if $f_i : \mathbb{R}^n \to \mathbb{R}$ is continuous for all $i = 1, ..., m$,
- *Lipschitz (continuous)*, if

$$\exists\ L > 0 :\ \|f(x) - f(y)\| \ \leq \ L\,\|x - y\| \quad \forall x, y \in \mathbb{R}^n,$$

- *locally Lipschitz* at $z \in \mathbb{R}^n$, if

$$\exists \varepsilon > 0 : f \text{ Lipschitz on } \mathcal{B}_\varepsilon(z),$$

- *differentiable* at $x \in \mathbb{R}^n$, if there exists a unique linear mapping $Df(x) : \mathbb{R}^n \to \mathbb{R}^m$ (i.e. a matrix $Df(x) \in \mathbb{R}^{m\times n}$) such that

$$\lim_{h\to 0} \frac{1}{\|h\|}(f(x + h) - f(x) - Df(x)h) = 0.$$

For a differentiable function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ the matrix $Df(x) \in \mathbb{R}^{m\times n}$ is called the *Jacobian* of f at x. If f is continuously differentiable, then

$$Df(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \dots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \dots & \frac{\partial f_m}{\partial x_n}(x) \end{bmatrix}.$$

If $m = 1$, the Jacobian is a row vector and the related column vector

$$\nabla f(x) = Df(x)^\top = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}$$

is referred to as the *gradient* of f at x. For arbitrary $d \in \mathbb{R}^n$ it is

$$\lim_{t\to 0} \frac{1}{t}(f(x+td) - f(x)) = \nabla f(x)^\top d$$

such that the term $\nabla f(x)^\top d$ denotes the slope of f at x in direction d (hence the name „gradient“).

If $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, the symmetric matrix

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix} \in \mathbb{R}^{n\times n}$$

is called *Hessian* of f at x. Note that the Hessian function is the Jacobian of the gradient function $\nabla f\colon \mathbb{R}^n \to \mathbb{R}^n$, i.e. $\nabla^2 f := D(\nabla f)$. A very important result in this context is the following theorem.

Theorem 1.1.1 (Taylors Theorem)
Let $f : \mathbb{R}^n \to \mathbb{R}$ and $x, y \in \mathbb{R}^n$ be arbitrary. It holds:

1. *If f is continuously differentiable, then there exists a vector $z \in [x, y]$ such that*

$$f(y) = f(x) + \nabla f(z)^\top (y - x).$$

2. *If f is twice continuously differentiable, then there exists a vector $z \in [x, y]$ such that*

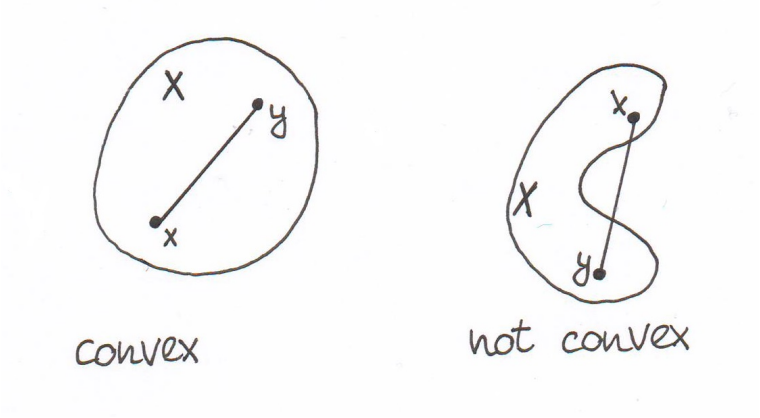
$$f(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top \nabla^2 f(z)(y - x).$$

1.2 Convexity

The concept of convexity is of particular importance in optimization, since it makes things „a lot easier“.

Definition 1.2.1
A set $X \subseteq \mathbb{R}^n$ *is called a* convex set *if for all* $x, y \in X$ *it holds*

$$(1-\lambda)x + \lambda y \in X \quad \forall \lambda \in]0,1[.$$



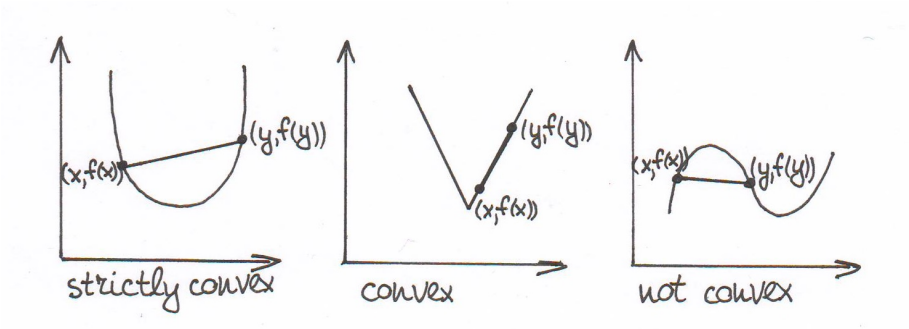
Definition 1.2.2
Let $X \subseteq \mathbb{R}^n$ *be a convex set and let* $f : X \to \mathbb{R}$*. The function* f *is called:*

- convex *on* X *if for all* $x, y \in X$ *it holds*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \qquad \forall \lambda \in]0,1[,$$

- strictly convex *on* X *if for all* $x, y \in X$ *with* $x \neq y$ *it holds*

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y) \qquad \forall \lambda \in]0,1[.$$



Theorem 1.2.3 (First order characterisation of convex functions)
Let $X \subseteq \mathbb{R}^n$ be a convex set and let $f : X \to \mathbb{R}$ be continuously differentiable. Then it holds:

1. *f is convex if and only if $\nabla f(x)^\top (y - x) \leq f(y) - f(x) \quad \forall x, y \in X$.*
2. *f is strictly convex if and only if $\nabla f(x)^\top (y - x) < f(y) - f(x) \quad \forall x, y \in X$ with $x \neq y$.*

Proof.
Part 1:
„$\Rightarrow$“ We assume that f is convex and want to show that $\nabla f(x)^\top (y - x) \leq f(y) - f(x)$ holds for all $x, y \in X$.
Let therefore $x, y \in X$ and $\lambda \in]0, 1[$ be arbitrary. The convexity of f yields

$$\begin{aligned} & f(x + \lambda(y - x)) \leq f(x) + \lambda(f(y) - f(x)) \\ \overset{\lambda > 0}{\Leftrightarrow} \; & \frac{1}{\lambda}[f(x + \lambda(y - x)) - f(x)] \leq f(y) - f(x) \\ \overset{\lambda \searrow 0}{\Leftrightarrow} \; & \nabla f(x)^\top (y - x) \leq f(y) - f(x). \end{aligned}$$

„$\Leftarrow$“ We assume that $\nabla f(x)^\top (y - x) \leq f(y) - f(x)$ holds for all $x, y \in X$ and want to show that f is convex.
Let therefore $x, y \in X$ and $\lambda \in]0, 1[$ be arbitrary. Since X is convex it holds

$$z := \lambda x + (1 - \lambda) y \in X.$$

Further, we have

$$\begin{aligned} \lambda f(x) + (1 - \lambda) f(y) - f(z) &= \lambda \underbrace{f(x) - f(z)}_{\geq f(x)^\top (x - z)} + (1 - \lambda) \underbrace{(f(y) - f(x))}_{\geq f(x)^\top (y - z)} \\ &\geq \lambda \nabla f(z)^\top (x - z) + (1 - \lambda) \nabla f(z)^\top (y - z) \\ &= \nabla f(z)^\top \underbrace{(\lambda x + (1 - \lambda) y - z)}_{=0} \\ &= 0, \end{aligned}$$

such that $\lambda f(x) + (1 - \lambda) f(y) \geq f(z)$, i.e. f is convex.

Part 2:
„$\Leftarrow$“ In analogy to „$\Leftarrow$“ in Part 1.
„$\Rightarrow$“ We assume that f is strictly convex and want to show that $\nabla f(x)^\top (y - x) < f(y) - f(x)$ holds for all $x, y \in X,\ x \neq y$.
Since X is a convex set it holds

$$z := \frac{1}{2}(x + y) \in X.$$

Further, we have

$$\nabla f(x)^\top (y - x) = 2 \nabla f(x)^\top (z - x) \overset{1.}{\leq} 2(f(z) - f(x)).$$

Since f is strictly convex it holds

$$f(z) < \frac{1}{2}(f(x) + f(y)),$$

such that

$$f(x)^\top(y-x) < 2\left(\frac{1}{2}\left(f(x)+f(y)\right)-f(x)\right) = f(y)-f(x).$$

□

Theorem 1.2.4 (Second order characterization of convex functions)
Let $X \subseteq \mathbb{R}^n$ be a convex set and let $f : X \to \mathbb{R}$ be twice continuously differentiable. Then it holds:

1. *If $\nabla^2 f(x) \geq 0 \ \forall x \in X$, then f is convex on X.*
2. *If $\nabla^2 f(x) > 0 \ \forall x \in X$, then f is strictly convex on X.*
3. *If X is an open set and if f is convex, then $\nabla^2 f(x) \geq 0 \ \forall x \in X$.*

Proof.
Part 1:
Let $x, y \in X$ be arbitrary. By Theorem 1.1.1 it exists a $z \in [x, y]$ such that

$$\begin{aligned} & f(y) = f(x) + \nabla f(x)^\top(y-x) + \frac{1}{2}(y-x)^\top \nabla^2 f(z)(y-x) \geq f(x) + \nabla f(x)^\top(y-x) \\ \Leftrightarrow\ & f(y) - f(x) \geq \nabla f(x)^\top(y-x). \end{aligned}$$

Hence, f is convex by 1.2.3.

Part 2:
In analogy to Part 1.

Part 3:
Let X be an open set, f be convex, and $x \in X$ and $d \in \mathbb{R}^n$ be arbitrary. Since X is open it holds $y_t := x + td \in X$ for all t sufficiently small (i.e. $t \leq \varepsilon_x / \|d\|$). By Theorem 1.1.1 there exists a $z \in [x, y_t]$ such that

$$\begin{aligned} & f(y_t) = f(x) + \nabla f(x)^\top(y_t - x) + \frac{1}{2}(y_t - x)^\top \nabla^2 f(z)(y_t - x) \\ \Leftrightarrow\ & f(x) - f(y_t) + \nabla f(x)^\top(y_t - x) = -\frac{1}{2} t^2 d^\top \nabla^2 f(z) d. \end{aligned}$$

By Theorem 1.2.3 we have

$$0 \geq f(x) - f(y_t) + \nabla f(x)^\top(y_t - x),$$

such that

$$d^\top \nabla^2 f(z_t) d \geq 0.$$

Since $\nabla^2 f$ is continuous it follows

$$0 \leq \lim_{t \searrow 0} d^\top \nabla^2 f(z_t) d = d^\top \nabla^2 f(x) d$$

and therefore

$$\nabla^2 f(x) \geq 0 \quad \forall x \in X.$$

□

1.3 Optimization Problems

Let a set $X \subseteq \mathbb{R}^n$ and a function $f\colon X \to \mathbb{R}$ be given. The task of finding a vector $x^* \in X$ such that

$$f(x^*) \leq f(x) \quad \forall x \in X$$

is referred to as *minimization problem*, denoted as

$$\min_{x \in X} f(x). \tag{P}$$

We call:

- f *cost* or *objective* function,
- x (decision) *variable*,
- x^* *solution* (vector),
- X *feasible set*,
- $f^* := f(x^*)$ *optimal value*,
- (P) *unconstrained* if $X = \mathbb{R}^n$,
- (P) *constrained* if $X \subsetneq \mathbb{R}^n$.

Typically, the feasible set X is given in functional form as

$$\begin{aligned} X := \{x \in \mathbb{R}^n : & c_i(x) = 0 \quad \forall i \ \in I_{eq} \\ & c_i(x) \leq 0 \quad \forall i \in I_{ineq}\} \end{aligned}$$

for some functions $c_i : \mathbb{R}^n \to \mathbb{R}$ and disjoint index sets I_{eq} and I_{ineq}. We refer to the conditions $i \in I_{eq}$ as *equality constraints* and to the conditions $i \in I_{ineq}$ as *inequality constraints*. In this case we rewrite (P) as

$$\begin{array}{lll} \min & f(x) & \\ \text{s.t.} & c_i(x) = 0 & \forall i \in I_{eq} \\ & c_i(x) \leq 0 & \forall i \in I_{ineq} \end{array}$$

Definition 1.3.1
A vector x^* is called

- local solution *of* (P), *if*

$$\exists \varepsilon > 0 : \ f(x^*) \leq f(x) \quad \forall x \in B_\varepsilon(x^*) \cap X,$$

- strict local solution *of* (P), *if*

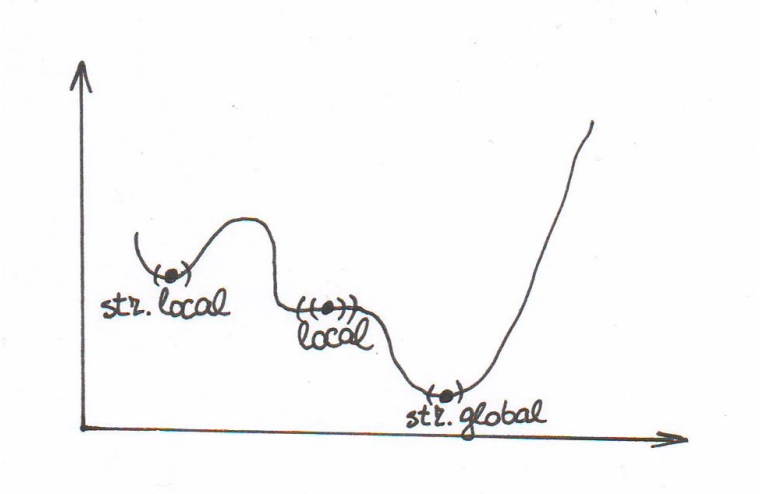
$$\exists \varepsilon > 0 : \ f(x^*) < f(x) \quad \forall x \in B_\varepsilon(x^*) \cap X, \ x \neq x^*,$$

- global solution *of* (P), *if*

$$f(x^*) \leq f(x) \quad x \in X,$$

- strict global solution *of* (P), *if*

$$f(x^*) < f(x) \quad x \in X,\ x \neq x^*.$$



Obviously, any global solution of (P) is also local solution, whereas the reversal is in general not valid. For convex functions, however, the reversal holds also true.

Theorem 1.3.2
Let $X \subseteq \mathbb{R}^n$ be a convex set and let $f : X \to \mathbb{R}$ be a convex function. Then it holds:

1. *Every local solution of (P) is a global solution.*
2. *The solution set $X^* = \{x^* \in X : \ f(x^*) \leq f(x)\ \forall x \in X\}$ is a convex set.*
3. *The problem (P) has at most one solution (which is then a strict global solution).*

Proof.
See Exercise 4. □

Besides local and global solutions, the simple example

$$\min_{x \in \mathbb{R}^n} e^x$$

shows that no (local or global) solution has to exist. Frequently, existence of solutions is shown by means of the following theorem.

Theorem 1.3.3 (Weierstrass extreme value theorem)
Let $X \subseteq \mathbb{R}^n$ be a compact set and let $f : X \to \mathbb{R}$ be a continuous function. Then, f attains its minimum and maximum on X, i.e.

$$\exists x_1, x_2 \in X : \ f(x_1) \leq f(x) \leq f(x_2)\ \forall x \in X.$$

If X is not compact, it suffices to show that the level set

$$X \cap L_0, \; L_0 := \{x \in \mathbb{R}^n : f(x) \leq f(x^{(0)})\}$$

to some level $x^{(0)} \in X$ is compact, which is for example true if

$$\lim_{\|x\| \to \infty} f(x) = \infty.$$

For simplification, we assume in the following that a solution of (P) always exists.

1.4 Excursus: Derivative Approximation with Finite Differences

The numerical solution of optimization problems frequently requires to deal with gradients ∇f and Hessians $\nabla^2 f$ of functions $f \colon \mathbb{R}^n \to \mathbb{R}$. Their computation can become very expensive or even impossible, especially if the number of variables n becomes large. It is therefore common practice to use appropriate approximations of this quantities, of which the finite-difference has become very popular and is therefore briefly introduced in the following.

For a function $f : \mathbb{R} \to \mathbb{R}$ the Taylor series at x around h yields

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2} f''(x) + ... \\ &= \sum_{h=0}^{\infty} \frac{h^n}{n!} f^{(n)}(x) \\ &= \sum_{h=0}^{d} \frac{h^n}{n!} f^{(n)}(x) + \mathcal{O}(h^d) \end{aligned}$$

and

$$f(x-h) = \sum_{n=0}^{d} (-1)^n \frac{h^n}{n!} f^{(n)}(x) + \mathcal{O}(h^d).$$

With $d = 1$ we get

- $f'(x) = \dfrac{f(x+h) - f(x)}{h} + \mathcal{O}(h),$ (forward difference)
- $f'(x) = \dfrac{f(x) - f(x-h)}{h} + \mathcal{O}(h).$ (backward difference)

With $d = 2$ we get

- $f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \mathcal{O}(h^2)$
- $f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) + \mathcal{O}(h^2)$

such that

$$\begin{aligned} & f(x+h) - f(x-h) = 2hf'(x) + \mathcal{O}(h^2) \\ \Leftrightarrow\ & f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2). \end{aligned} \qquad \text{(central difference)}$$

By means of this approximations we can we can approximate the partial derivatives of a continuous function $f : \mathbb{R}^n \to \mathbb{R}$ at x via

- $\frac{\partial f}{\partial x_i}(x) = \frac{f(x+h_i e_i) - f(x)}{h_i} + \mathcal{O}(h_i)$,
- $\frac{\partial f}{\partial x_i}(x) = \frac{f(x) - f(x-h_i e_i)}{h_i} + \mathcal{O}(h_i)$,
- $\frac{\partial f}{\partial x_i}(x) = \frac{f(x+h_i e_i) - f(x-h_i e_i)}{2h_i} + \mathcal{O}(h_i^2)$,

where $h_i > 0$ and e_i denotes the i-th uni vector, $i = 1, \ldots, n$. Therefore, we obtain the *forward gradient* approximation

$$\nabla f(x) \approx \begin{pmatrix} \dfrac{f(x+h_1 e_1) - f(x)}{h_1} \\ \vdots \\ \dfrac{f(x+h_n e_n) - f(x)}{h_n} \end{pmatrix}.$$

For simplification one frequently uses $h_i = h$ for all $i = 1, ..., n$. This yields Algorithm 1.4.1 to compute an approximation of the gradient. Note that this computation requires n additional function evaluations.

Algorithm 1.4.1: Forward gradient approximation

Input: $f,\ x \in \mathbb{R}^n,\ h \in \mathbb{R}^n,\ h > 0$

```
f_x ← f(x)
for i = 1, ..., n do
    f_i ← f(x + h_i e_i)
    g_i ← (f_i − f_x)/h_i
end
return g ≈ ∇f(x)
```

In analogy, we obtain the more exact central gradient approximation as

$$\nabla f(x) \approx \begin{pmatrix} \frac{f(x+h_1 e_1) - f(x-h_1 e_1)}{2h_1} \\ \vdots \\ \frac{f(x+h_n e_n) - f(x-h_n e_n)}{2h_n} \end{pmatrix}$$

leading to Algorithm 1.4.2. The central gradient is more accurate then the forward or backward approximation, but requires $2n$ additional function evaluations.

Algorithm 1.4.2: Central gradient approximation

Input: $f,\ x \in \mathbb{R}^n,\ h \in \mathbb{R}^n,\ h > 0$

for $i = 1, \ldots, n$ **do**

  $f_i^+ \leftarrow f(x + h_i e_i)$

  $f_i^- \leftarrow f(x - h_i e_i)$

  $g_i \leftarrow (f_i^+ - f_i^-)/2h_i$

end

return $g \approx \nabla f(x)$

For Hessian approximation we now want to approximate second order partial derivatives

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x).$$

For this purpose we can use forward difference twice. Once to compute $\frac{\partial f}{\partial x_j} =: g$ and then $\frac{\partial g}{\partial x_i}$.

$$\begin{aligned}
\frac{\partial^2 f}{\partial x_i \partial x_j}(x) &= \frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial x_j}(x)\right) \approx \frac{\partial}{\partial x_i}\left(\frac{f(x + h_j e_j) - f(x)}{h_j}\right) \\
&= \frac{1}{h_j}\left(\frac{\partial f}{\partial x_i}(x + h_j e_j) - \frac{\partial f}{\partial x_i}(x)\right) \\
&\approx \frac{1}{h_j}\left(\left(\frac{f(x + h_j e_j + h_i e_i)}{h_i} - \frac{f(x + h_j e_j)}{h_i}\right) - \left(\frac{f(x + h_i e_i)}{h_i} - \frac{f(x)}{h_i}\right)\right) \\
&= \frac{1}{h_j h_i}\left(f(x + h_j e_j + h_i e_i) - f(x + h_j e_j) - f(x + h_i e_i) + f(x)\right).
\end{aligned}$$

Especially for $i = j$ it holds

$$\frac{\partial^2 f}{\partial x_i^2} = \frac{1}{h_i^2}\left(f(x + h_i e_i) - 2f(x + h_i e_i) + f(x)\right).$$

In analogy, applying the central difference twice yields

$$\frac{\partial^2 f}{\partial x_i \partial x_j}(x) =$$
$$\frac{1}{4h_i h_j}\left[f(x + h_i e_i + h_j e_j) - f(x + h_i e_i - h_j e_j) - f(x - h_i e_i + h_j e_j) + f(x - h_i e_i - h_j e_j)\right]$$

and

$$\frac{\partial^2 f}{\partial x_i^2}(x) = \frac{1}{4h_i^2}[f(x + 2h_i e_i) - 2f(x) + f(x - 2h_i e_i)].$$

Chapter 2

Unconstrained Optimization

In this chapter we focus on the numerical solution of unconstrained optimization problems, i.e.

$$\min_{x \in \mathbb{R}^n} f(x), \tag{PU}$$

where f is at least continuously differentiable. We will:

- state conditions to identify (local) solutions of (PU),
- derive numerical methods to find (local) solutions of (PU).

More detail and comprehensive information are given by Nocedal and Wright (2006), Sun and Yuan (2006), and Alt (2013) among others.

2.1 Optimality Conditions

In order to state optimality conditions we make use of the following lemma, which will also be useful for the construction of numerical solution methods for the unconstrained problem (PU).

Lemma 2.1.1
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x^* \in \mathbb{R}^n$. If x^* is a solution of (PU), then it holds

$$\nabla f(x^*)^\top d \geq 0 \quad \forall\, d \in \mathbb{R}^n.$$

Proof.
Since x^* is a local solution there exists an $\varepsilon > 0$ such that

$$f(x^*) \leq f(x) \quad \forall\, x \in \mathcal{B}_\varepsilon(x^*).$$

For arbitrary $d \in \mathbb{R}^n$ it holds

$$\nabla f(x^*)^\top d = \lim_{t \searrow 0} \frac{1}{t} \left(f(x^* + td) - f(x^*) \right)$$

and since $x^* + td \in \mathcal{B}_\varepsilon(x^*)$ for $t < \varepsilon/\|d\|$ we have

$$f(x^* + td) - f(x^*) \geq 0.$$

With $t \searrow 0$ we conclude by the continuity of f that

$$\nabla f(x^*)^\top d \geq 0.$$

□

This directly yields the following more convenient optimality condition.

Theorem 2.1.2 (Necessary first-order optimality condition (NFOC))
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x^* \in \mathbb{R}^n$. If x^* is a solution of (PU), then it holds

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

Proof.
For arbitrary $d \in \mathbb{R}^n$ the Lemma 2.1.1 implies that $\nabla f(x^*)^T d \geq 0$ and since also $-d \in \mathbb{R}^n$ it further holds

$$-\nabla f(x^*)^\top d \geq 0,$$

which yields

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

since d is arbitrarily chosen. □

Note that the condition $\nabla f(x^*) = 0$ is not sufficient to indicate x^* as a solutions as the simple example

$$\min_{x \in \mathbb{R}} -x^2$$

shows ($x^* = 0$ is a maximum). However, points with a vanishing gradient are of major interest for the construction of solution algorithms.

Definition 2.1.3
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x^* \in \mathbb{R}^n$. If $\nabla f(x^*) = 0$, x^* is referred to as stationary point.

To distinguish between maxima and minima, we introduce the following theorem.

Theorem 2.1.4 (Necessary second-order optimality condition (NSOC))
Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable at $x^* \in \mathbb{R}^n$. If x^* is a solution of (PU), then it holds

$$d^\top \nabla^2 f(x^*) d \geq 0 \quad \forall\ d \in \mathbb{R}^n,$$

i.e. $\nabla^2 f(x^*)$ is positive semidefinite.

Proof.
Since x^* is a solution of (PU) there exists an $\varepsilon > 0$ such that

$$f(x^*) \leq f(x) \quad \forall\, x \in \mathcal{B}_\varepsilon(x^*).$$

For $t < \frac{\varepsilon}{\|d\|}$ it holds

$$x_t^* := x^* + td \in \mathcal{B}_\varepsilon(x^*)$$

such that $f(x^*) \leq f(x_t^*)$. By Theorem 1.1.1 there exists a $z_t \in [x^*, x_t^*]$ with

$$\begin{aligned} & f(x_t^*) = f(x^*) + \underbrace{\nabla f(x^*)^\top}_{=0}(x_t^* - x^*) + \frac{1}{2}\underbrace{(x_t^* - x^*)^\top}_{=td}\nabla^2 f(z_t)(x_t^* - x^*) \\ \Leftrightarrow\ & \underbrace{f(x_t^*) - f(x^*)}_{\geq 0} = \frac{1}{2}t^2 d^\top \nabla^2 f(z_t) d \\ \Rightarrow\ & d^\top \nabla^2 f(z_t) d \geq 0. \end{aligned}$$

Since $\nabla^2 f$ is continuous we conclude from $t \searrow 0$ that

$$d^\top \nabla^2 f(x^*) d \geq 0.$$

□

As for the NFOC the NSOL is not a sufficient condition to indicate a solution as the simple example

$$\min_{x \in \mathbb{R}} x^3$$

shows ($x^* = 0$ is not a minimum). To achieve sufficient optimality conditions we therefore need to make stronger assumptions.

Theorem 2.1.5 (Sufficient second-order optimality condition (SSOC))
Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable at $x^* \in \mathbb{R}^n$. If it holds

1. $\nabla f(x^*) = 0$ *and*
2. $\nabla^2 f(x^*) > 0$,

then x^* is a strict local solution of (PU).

Proof.
Let the assumptions 1. and 2. hold. Since $\nabla^2 f(x^*)$ is symmetric positive definite, there exists an $\alpha > 0$ such that

$$x^\top \nabla^2 f(x^*) x \geq \alpha \|x\|^2 \quad \forall\, x \in \mathbb{R}^n.$$

Since $\nabla^2 f$ is continuous in x^* there exists an $\varepsilon > 0$ such that

$$\|\nabla^2 f(x^*) - \nabla^2 f(x)\| \leq \frac{\alpha}{2} \quad \forall\, x \in \mathcal{B}_\varepsilon(x^*).$$

Let $x \in \mathcal{B}_\varepsilon(x^*)\backslash$, $x \neq x*$, be arbitrary. By Theorem 1.1.1, there exists a vector $y \in [x, x^*]$ such that

$$\begin{aligned} & f(x) = f(x^*) + \nabla f(x^*)^\top (x - x^*) + \frac{1}{2}(x - x^*)^\top \nabla^2 f(y)(x - x^*) \\ \Leftrightarrow\ & f(x) - f(x^*) = \frac{1}{2}(x - x^*)\nabla^2 f(y)(x - x^*). \end{aligned}$$

Thus, it is left to show that $(x - x^*)^\top \nabla^2 f(y)(x - x^*) > 0$ holds. It is

$$\begin{aligned} & (x - x^*)^\top \nabla^2 f(y)(x - x^*) \\ & = (x - x^*)^\top \nabla^2 f(x^*)(x - x^*) - (x - x^*)^\top [\nabla^2 f(x^*) - \nabla^2 f(y)](x - x^*) \\ & \geq \alpha \|x - x^*\|^2 - \|\nabla^2 f(x^*) - \nabla^2 f(y)\| \|x - x^*\|^2 \\ & \geq \alpha \|x - x^*\|^2 - \frac{\alpha}{2}\|x - x^*\|^2 && \text{(since } y \in \mathcal{B}_\varepsilon(x^*)\text{)} \\ & \geq \frac{\alpha}{2}\|x - x^*\|^2 \\ & > 0. \end{aligned}$$

Finally, $f(x) > f(x^*)$ for all $x \in \mathcal{B}_\varepsilon(x^*)$ with $x \neq x^*$. □

Note that the SSOC is not a necessary condition to indicate a solution of (PU) as the simple example

$$\min_{x \in \mathbb{R}^4} x^4$$

with $x^* = 0$ shows.

The introduced conditions are based on local informations, i.e. gradient and Hessian at x^*, and are therefore only able to characterize *local* solutions. For global solutions such „simple“ conditions can in general not be derived. An exception is given by convex optimization problems.

Theorem 2.1.6 (Optimality conditions for convex problems)
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function that is continuously differentiable at $x^* \in \mathbb{R}^n$. Then, x^* is a global solution of (PU) if and only if $\nabla f(x^*) = 0$.

Proof.
„⇒“ Let x^* be a global solution. Then, the NFOC directly yields $\nabla f(x^*) = 0$.
„⇐“ Let x^* be a stationary point of f. Since f is a convex function, Theorem 1.2.3 yields for all $x \in \mathbb{R}^n$ that

$$\begin{aligned} & f(x) - f(x^*) \geq \nabla f(x^*)^\top (x - x^*) \\ \Leftrightarrow\ & f(x) \geq f(x^*), && \text{(since } \nabla f(x^*) = 0\text{)} \end{aligned}$$

i.e. x^* is a global solution of (PU). □

2.2 Line Search Methods

In general, there is no analytical (closed-form) solution of the non-linear equation

$$\nabla f(x) \stackrel{!}{=} 0,$$

such that numerical methods have to be applied in order to find stationary points. The idea of *descent methods* is to start with an initial guess $x^{(0)} \in \mathbb{R}^n$ and to generate a sequence $(x^{(k)})_{k\in\mathbb{N}_0}$ with

$$f(x^{(k+1)}) < f(x^{(k)}) \quad \forall\, k \in \mathbb{N}_0.$$

In this context, *line search methods* choose a search direction $d^{(k)} \in \mathbb{R}^n$ and a step length $t_k > 0$ and use the update

$$x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}.$$

The success of the line search strategy, as presented in Algorithm 2.2.1, depends on both $d^{(k)}$ and t_k. The major part of this chapter is devoted to the question of how to efficiently choose those quantities.

Algorithm 2.2.1: General line search method

Input: $x^{(0)}$, $\varepsilon \geq 0$, $k \leftarrow 0$
while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
  Find $d^{(k)}$ and t_k such that $f(x^{(k)} + t_k d^{(k)}) < f(x^{(k)})$
  $x^{(k)} \leftarrow x^{(k)} + t_k d^{(k)}$
  $k \leftarrow k + 1$
end

2.2.1 Descent Directions

Definition 2.2.2
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x \in \mathbb{R}^n$. A vector $d \in \mathbb{R}^n$ is called descent direction *of f at x if*

$$\nabla f(x)^\top d < 0.$$

Lemma 2.2.3
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x \in \mathbb{R}^n$ and let $d \in \mathbb{R}^n$ be a descent direction of f at x. Then there exists a $\bar{t} > 0$ such that

$$f(x + td) < f(x) \quad \forall t \in]0, \bar{t}[.$$

Proof.
Since f is continuously differentiable at x it holds

$$0 > \nabla f(x)^\top d = Df(x)d = \lim_{t \searrow 0} \frac{1}{t}(f(x + td) - f(x)),$$

such that, since f is continuous, there exists a $\bar{t} > 0$ with

$$f(x + td) - f(x) < 0 \quad t \in]0, \bar{t}[.$$

□

Note that, due to Lemma 2.1.1, there exists always a descent direction of f at x if x is not a solution of (PU).

Lemma 2.2.4
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x \in \mathbb{R}^n$ with $\nabla f(x) \neq 0$. Then $-\nabla f(x)$ provides a descent direction of f at x.

Proof.
Since $\nabla f(x) \neq 0$ it holds $\nabla f(x)^\top(-\nabla f(x)) = -\|\nabla f(x)\|^2 < 0$. □

Lemma 2.2.5
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x \in \mathbb{R}^n$ with $\nabla f(x) \neq 0$ and let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite. Then

$$-A\nabla f(x)$$

provides a descent direction of f at x.

Proof.
It holds

$$\begin{aligned}\nabla f(x)^\top(-A\nabla f(x)) &= -\nabla f(x)^\top A\nabla f(x)\\ &\leq -\|A\| \cdot \|\nabla f(x)\|^2\\ &< 0.\end{aligned}$$

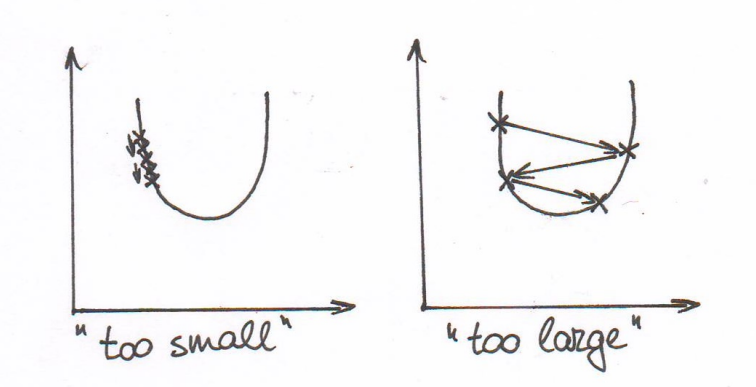
□

2.2.2 Step Length

For $x^{(k)} \in \mathbb{R}^n$ let $d^{(k)}$ be a descent direction of f at $x^{(k)}$. A line search finds the new iterate $x^{(k+1)}$ along the half-line $\{x^{(k)} + td^{(k)} : t \geq 0\}$ in such a way that

$$f(x^{(k+1)}) = f(x^{(k)} + td^{(k)}) < f(x^{(k)}).$$

Note that, by Lemma 2.2.3, such an iterate always exists. The challenges are in avoiding both, a too small and a too large step length.



Optimal / exact step length

Let $\varphi(t) := f(x^{(k)} + td^{(k)})$. Then, the *optimal step length* is a global solution of

$$\min_{t>0} \varphi(t) \ .$$

However, finding the exact step length requires the solution of an optimization problem itself, which is (in general) too expensive or even impossible. It is therefore introduced only for theoretical investigations.

Constant step length

On the other hand, the most simple approach is a *constant step length* for each iteration, i.e.

$$t_k = t \quad \forall k \in \mathbb{N}.$$

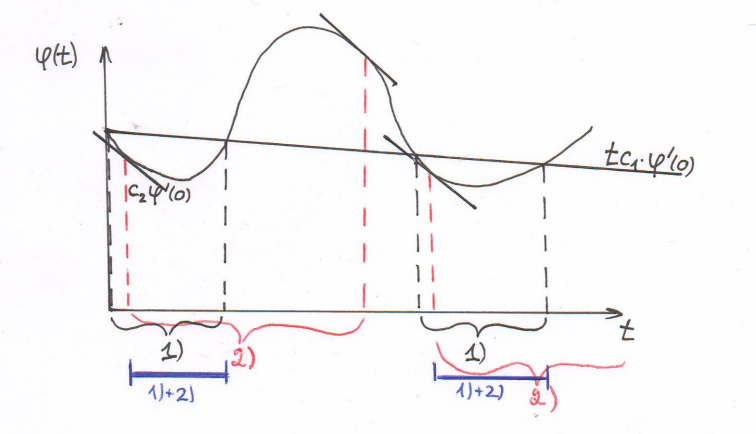
However, the step length t has then to be „very small“ resulting in a very slow convergence rate.

Wolfe condition

More practical line searches make a compromise between accuracy and usability. Commonly used conditions to achieve an adequate step length are the *Wolfe conditions*:

1. $\varphi(t) \ \leq \ \phi(0) + c_1 t\varphi'(0) \quad \Leftrightarrow f(x^{(k)} + td^{(k)}) \leq f(x^{(k)} + c_1 t\nabla f(x^{(k)})^\top d^{(k)}$
2. $\varphi'(t) \ \geq \ c_2\varphi(0)' \quad \Leftrightarrow \nabla f(x^{(k)} + td^{(k)})^\top d^{(k)} \geq c_2 \nabla f(x^{(k)})^\top d^{(k)}$

with some constant $0 < c_1 < \frac{1}{2} < c_2 < 1$. Typical choices are for example $c_1 = 10^{-2}$ and $c_2 = 0.9$. The first condition asks f to decrease enough (t not too large), whereas the second condition asks the derivative to increase enough (t not too small). The first condition is frequently referred to as *sufficient decrease condition* (or Armijo condition) and the second one as *curvature condition*.



Lemma 2.2.6
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable. Let $d^{(k)}$ be a descent direction of f at $x^{(k)}$ and let f be bounded below along the ray $\{x^{(k)} + td^{(k)} : t \geq 0\}$. Then, for all $0 < c_1 < c_2 < 1$ there exists a step length $\tilde{t}$ satisfying the Wolfe condition.

Proof.
We define the function

$$\begin{aligned}\Phi(t) :&= \varphi(t) - (\varphi(0) + c_1 t \cdot \varphi'(0)) \\ &= f(x^{(k)} + td^{(k)}) - (f(x^{(k)}) + c_1 t \cdot \nabla f(x^{(k)})^\top d^{(k)}) \ .\end{aligned}$$

Since

$$\begin{aligned}\Phi(0) &= 0 \\ \Phi'(0) &= \nabla f(x^{(k)} + 0d^{(k)})^\top d^{(k)} - c_1 \nabla f(x^{(k)})^\top d^{(k)} \\ &= (1 - c_1)\nabla f(x^{(k)})^\top d^{(k)} < 0\end{aligned}$$

it holds $\Phi(t) < 0$ for all t sufficiently close to zero. Since f is bounded below on $\{x^{(k)} + td^{(k)}\}$ it holds

$$\lim_{t\to\infty} \Phi(t) = \lim_{t\to\infty} (f(x^{(k)} + td^{(k)}) - f(x^{(k)}) - c_1 t \cdot \nabla f(x^{(k)})^\top d^{(k)}) = \infty$$

and since Φ is continuous (because f is continuously differentiable), there exists a $\bar{t} > 0$ such that

1. $\Phi(\bar{t}) = 0$,
2. $\Phi(t) \leq 0 \quad \forall t \in [0, \bar{t}]$,

i.e. the sufficient decrease condition is met for all $t \in [0, \bar{t}]$. By Theorem 1.1.1, there exists a $\tilde{t} \in [0, \bar{t}]$ such that

$$\begin{aligned}&\Phi(\bar{t}) = \Phi(0) + \Phi(\tilde{t})' \cdot (\bar{t} - 0) \\ \Leftrightarrow\ &\bar{t} \cdot [\nabla f(x^{(k)} + td^{(k)})^\top d^{(k)} - c_1 \nabla f(x^{(k)})^\top d^{(k)}] = 0 \\ \Leftrightarrow\ &\nabla f(x^{(k)} + \tilde{t}d^{(k)})^\top d^{(k)} = c_1 \nabla f(x^{(k)})^\top d^{(k)} > c_2 \nabla f(x^{(k)})^\top d^{(k)},\end{aligned}$$

i.e. $\tilde{t}$ satisfies the Wolfe condition. □

Note that, since f is continuous, there exists a whole interval around $\tilde{t}$ satisfying the Wolfe condition. In practice, a step length that satisfies the Wolfe condition is achieved by the *backtracking* approach.

Algorithm 2.2.7: Backtracking line search

Input: $t_{\max} > 0,\ \rho,\ c \in]0, 1[$

$t \leftarrow t_{\max}$

while $f(x^{(k)} + td^{(k)}) > f(x^{(k)}) + ct \cdot \nabla f(x^{(k)})^\top d^{(k)}$ **do**

  $t \leftarrow \rho \cdot t$

end

Note that the backtracking line search only checks for the sufficient decrease condition since the curvature condition is automatically met if $t_{\max}$ is not too small.

2.2.3 Convergence

So far we have obtained a whole class of optimization algorithms by combining the backtracking line search with any kind of descent direction in Algorithm 2.2.1. We now turn to the question of convergence, i.e.

- does the sequence $\left(x^{(k)}\right)_{k\in\mathbb{N}_0}$ converge to some x^*?
- is x^* a stationary point of f?

To establish convergence we have to ensure that the slope of f at $x^{(k)}$ in direction $d^{(k)}$, i.e. $\nabla f(x^{(k)})^\top d^{(k)}$, will not become too small (only if $x^{(k)}$ is already a stationary point). That is, we claim

$$\cos(\theta_k) = \frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|\nabla f(x^{(k)})\|\|d^{(k)}\|} \geq \omega > 0 \quad \forall k \in \mathbb{N}_0, \tag{angle condition}$$

where θ_k denotes the angle between $-\nabla f(x^{(k)})$ and $d^{(k)}$. If $d^{(k)}$ meets the angle condition and it further holds

$$\frac{-\nabla f(x^{(k)})d^{(k)}}{\|d^{(k)}\|} = \underbrace{\cos(\theta_k)}_{\geq\omega}\|\nabla f(x^{(k)})\| \to 0 \tag{Zoutendijk's condition}$$

we conclude $\|\nabla f(x^{(k)})\| \to 0$, i.e. $\nabla f(x^{(k)}) \to 0$.

Lemma 2.2.8
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and bounded below and let the sequence $\left(x^{(k)}\right)_{k\in\mathbb{N}_0}$ be generated by Algorithm 2.2.1, where the t_k meet the Wolfe condition. If ∇f is Lipschitz on the level set $L_0 = \{x : f(x) \leq f(x^{(0)})\}$, then the Zoutendijk condition is met for all descent directions $d^{(k)}$.

Proof.
Since $f(x^{(k+1)}) < f(x^{(k)})$ for all $k \in \mathbb{N}_0$ it holds $x^{(k)} \in L_0$ for all $k \in \mathbb{N}_0$ such that, since f

is bounded below, the sequence $f(x^{(k)})_{k\in\mathbb{N}_0}$ converges. Assume that $\left(\cos(\theta_k)\|\nabla f(x^{(k)})\|\right)_{k\in\mathbb{N}_0}$ does not converge to 0, such that there exists an $\varepsilon > 0$ and an infinite index set I_∞ with

$$\frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|d^{(k)}\|} \geq \varepsilon \quad \forall k \in I_\infty.$$

The sufficient decrease condition implies

$$\begin{aligned} & f(x^{(k+1)}) - f(x^{(k)}) \leq c_1 t_k \nabla f(x^{(k)})^\top d^{(k)} \\ \Leftrightarrow\ & f(x^{(k)}) - f(x^{(k+1)}) \geq c_1 t_k \|d^{(k)}\| \left(\frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|d^{(k)}\|}\right) \geq c_1 t_k \|d^{(k)}\| \varepsilon \quad \forall k \in I_\infty \end{aligned}$$

such that

$$\begin{aligned} & 0 \leq c_1 t_k \|d^{(k)}\| \varepsilon \leq f(x^{(k)}) - f(x^{(k+1)}) \to 0 \\ \Rightarrow\ & t_k \|d^{(k)}\| \to 0. \end{aligned}$$

By the curvature condition, we have

$$\begin{aligned} & \nabla f(x^{(k+1)})^\top d^{(k)} \geq c_2 \nabla f(x^{(k)})^\top d^{(k)} \\ \Leftrightarrow\ & [\nabla f(x^{(k+1)}) - \nabla f(x^{(k)})]^\top d^{(k)} \geq (c_2 - 1) \nabla f(x^{(k)})^\top d^{(k)} \\ \Leftrightarrow\ & \frac{1}{1-c_2} [\nabla f(x^{(k+1)}) - \nabla f(x^{(k)})]^\top d^{(k)} \geq -\nabla f(x^{(k)})^\top d^{(k)} \\ \Rightarrow\ & \varepsilon \leq -\frac{\nabla f(x^{(k)})^\top d^{(k)}}{\|d^{(k)}\|} \leq \frac{1}{1-c_2} \cdot \frac{1}{\|d^{(k)}\|} [\nabla f(x^{(k+1)}) - \nabla f(x^{(k)})]^\top d^{(k)} \end{aligned}$$

such that

$$\begin{aligned} \varepsilon &\leq \frac{1}{1-c_2} \cdot \frac{1}{\|d^{(k)}\|} \cdot \|\nabla f(x^{(k+1)}) - \nabla f(x^{(k)})\| \|d^{(k)}\| \\ &= \frac{1}{1-c_2} \|\nabla f(x^{(k)} + t_k d^{(k)}) - \nabla f(x^{(k)})\| \\ &\overset{\nabla f \text{ Lipschitz}}{\leq} \frac{L}{1-c_2} \|x^{(k)} + t_k d^{(k)} - x^{(k)}\| \\ &= \frac{L}{1-c_2} t_k \|d^{(k)}\| \to 0, \end{aligned}$$

i.e. $\varepsilon = 0$, which is a contradiction. □

Theorem 2.2.9
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and bounded below and let $(x^{(k)})_{k\in\mathbb{N}_0}$ be generated by Algorithm 2.2.1 where the t_k meet the Wolfe condition and the $d^{(k)}$ meet the angle condition. If ∇f Lipschitz on L_0, then either $\nabla f(x^{(k)}) = 0$ or $f(x^{(k)}) \to 0$.

Proof.
Let $f(x^{(k)}) \neq 0$ for all $k \in \mathbb{N}_0$. By Lemma 2.2.8 it follows that $(x^{(k)})_{k\in\mathbb{N}_0}$ meets the Zoutendijk condition, i.e.

$$0 \leq \cos(\theta_k)\|\nabla f(x^{(k)})\| = \frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|d^{(k)}\|} \to 0$$

and, since $\cos(\theta_k) \geq w > 0$, we conclude $\|\nabla f(x^{(k)})\| \to 0$, i.e. $\nabla f(x^{(k)}) \to 0$. □

Note that the convergence of $\left(f(x^{(k)})\right)_{k\in\mathbb{N}_0}$ coes not yield the convergence of $\left(x^{(k)}\right)_{k\in\mathbb{N}_0}$, but:

Theorem 2.2.10
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and bounded below and let $(x^{(k)})_{k\in\mathbb{N}_0}$ be as in Theorem 2.2.9. If L_0 is compact, then either $\nabla f(x^{(k)}) = 0$ for some k or there exists at least one accumulation point x^* of the sequence $(x^{(k)})_{k\in\mathbb{N}_0}$. For each of the accumulation points it holds $\nabla f(x^*) = 0$.

Proof.
Let $\nabla f(x^{(k)}) \neq 0$ for all $k \in \mathbb{N}_0$. Since $f(x^{(k+1)}) < f(x^{(k)})$ it holds $x^{(k)} \in L_0$ and, since L_0 is compact, the sequence $(x^{(k)})_{k\in\mathbb{N}_0}$ is bounded. The Theorem of Bolzano-Weierstrass yields the existence of an accumulation point x^*. Without loss of generality we assume $x^{(k)} \to x^*$. Since ∇f is continuous it holds $\nabla f(x^{(k)}) \to \nabla f(x^*)$ and, since L_0 is compact, ∇f is uniformly continuous on L_0 such that, by Theorem 2.2.9, $\nabla f(x^{(k)}) \to 0$, i.e. $\nabla f(x^*) = 0$. □

In Lemma 2.2.8 and Theorem 2.2.9 we can weaken the condition of ∇f being Lipschitz to being uniformly continuous (Lipschitz ⇒ uniformly continuous ⇒ continuous). Note that if a function g is continuous on a compact set X, then g is already uniformly continuous on X.

To derive and analyse numerical methods we make the following assumption throughout this chapter:

$$f \in C^1(\mathbb{R}^n) \text{ and } L_0 \text{ compact for some } x^{(0)}.$$

This ensures the existence of a global solution (i.e. a stationary point of f) and further, by Theorem 2.2.10, $x^{(k)} \to x^*$ if the descent directions $d^{(k)}$ meet the angle condition and the step lengths t_k meet the Wolfe condition.

2.3 The Steepest Descent Method

If x is not a stationary point, Lemma 2.2.4 yields that $-\nabla f(x)$ is a descent direction of f at x.

Lemma 2.3.1
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable at $x \in \mathbb{R}^n$ with $\nabla f(x) \neq 0$. Then

$$d^* := \frac{-\nabla f(x)}{\|\nabla f(x)\|}$$

is the steepest descent *direction of f at x, i.e. the solution of*

$$\min_{\|d\|=1} \nabla f(x)^\top d.$$

Proof.
By the Cauchy-Schwarz inequality we have for all $d \in \mathbb{R}^n$ that

$$0 > \nabla f(x)^\top d = -|\nabla f(x)^\top d| = -|\langle \nabla f(x), d\rangle| \geq -\|\nabla f(x)\|\|d\|$$

such that for all d with $\|d\| = 1$ it holds

$$\nabla f(x)^\top d \geq -\|\nabla f(x)\| = \nabla f(x)^\top d^*,$$

i.e. d^* is a global solution. □

Algorithm 2.3.2: Steepest descent method

Input: $x^0 \in \mathbb{R}^n,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
- $d^{(k)} \leftarrow -\nabla f(x^{(k)})$
- find t_k that meets wolfe conditions
- $x^{(k+1)} \leftarrow x^{(k)} + t d^{(k)}$
- $k \leftarrow k + 1$

end

Theorem 2.3.3 (global convergence)
Let $f \in C^1(\mathbb{R}^n)$, L_0 be compact, and $(x^{(k)})_{k\in\mathbb{N}_0}$ the iterates of the steepest descent method. Then, either $\nabla f(x^{(k)}) = 0$ for some $k \in \mathbb{N}_0$, or there exists at least one accumulation point of the sequence. Further, each accumulation point is a stationary point of f.

Proof.
Assume $\nabla f(x^{(k)}) \neq 0$ for all $k \in \mathbb{N}_0$. Due to Theorem 2.2.10 it suffices to show that $-\nabla f(x^{(k)})$ meets the angle condition for all $k \in \mathbb{N}_0$, i.e.

$$\frac{-\nabla f(x^{(k)})^\top(-\nabla f(x^{(k)}))}{\|\nabla f(x^{(k)})\| \cdot \|-\nabla f(x^{(k)})\|} \geq \omega > 0.$$

This is obviously true since

$$\frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla f(x^{(k)})\|^2} = 1 > 0.$$

□

Unfortunately, the global convergence of the steepest descent method does not mean that it is efficient. In fact, the steepest descent direction is only a local property and the method becomes actually very slow for many problems. This behaviour is explained by the following lemma.

Lemma 2.3.4 (Zigzagging of the steepest descent directions)
Let $(x^{(k)})_{k\in\mathbb{N}_0}$ be generated by the steepest descent method with exact step length. Then it holds

$$\langle -\nabla f(x^{(k+1)}), -\nabla f(x^{(k)})\rangle = 0,$$

i.e. consecutive descent directions are orthogonal.

Proof.
For fixed $k \in \mathbb{N}_0$ we define

$$\phi(t) = f(x^{(k)} - t\nabla f(x^{(k)})).$$

Since t_k is the optimal step length it holds

$$\begin{aligned} 0 = \phi'(t_k) &= Df(x^{(k)} - t_k \nabla f(x^{(k)}))(-\nabla f(x^{(k)})) \\ &= \nabla f(x^{(k)} - t_k f(x^{(k)})^\top (-\nabla f(x^{(k)}) \\ &= -\nabla f(x^{(k+1)})^\top \nabla f(x^{(k)}) \\ &= -\langle \nabla f(x^{(k+1)}), \nabla f(x^{(k)}) \rangle \ . \end{aligned}$$

Multiplication with -1 yields the desired equality. □

Even for the „ideal“ case of a strictly convex quadratic objective function

$$f(x) = \frac{1}{2} x^\top A x - b^\top x$$

with symmetric and positive definite matrix A the convergence rate of the steepest descent method can become very small.

Lemma 2.3.5 (Kantorovitch's inequality)
Let $A \in \mathbb{R}^{n \times n}$ be symmetric and positive definite with largest eigenvalue $\lambda_{max} > 0$ and smallest $\lambda_{min} > 0$. Then it holds

$$\frac{(x^\top x)^2}{(x^\top A x)(x^\top A^{-1} x)} \geq 4 \cdot \frac{\lambda_{min} \cdot \lambda_{max}}{(\lambda_{min} + \lambda_{max})^2} \quad \forall x \in \mathbb{R}^n.$$

Theorem 2.3.6 (Steepest descent for convex quadratic functions)
Consider the problem

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top A x - b^\top x$$

with symmetric and positive definite matrix A and let $x^* := A^{-1}b$. denote its unique solution. Let $(x^{(k)})_{k \in \mathbb{N}_0}$ denote the iterates of the steepest descent method with optimal step length. Then it holds

1. $f(x^{(k)}) - f(x^*) \leq \left(\frac{\kappa(A) - 1)}{\kappa(A) + 1} \right)^2 (f(x^{(k)}) - f(x^*))$
2. $\|x^{(k+1)} - x^*\|_A \leq \frac{\kappa(A) - 1}{\kappa(A) + 1} \ \|x^{(k)} - x^*\|_A$
3. $\|x^{(k+1)} - x^*\| \leq \sqrt{\kappa(A)} \ \frac{\kappa(A) - 1}{\kappa(A) + 1} \ \|x^{(k)} - x^*\|$

where $\kappa(A) := \lambda_{\min}/\lambda_{\max}$ denotes the condition number of A.

Proof.
Part 1: As shown in Exercise 10 it holds

$$f(x^{(k+1)}) - f(x^{(k)}) = \left(1 - \frac{(g_k^\top g_k)^2}{(g_k^\top A g_k)(g_k^\top A^{-1} g_k)}\right) \cdot (f(x^{(k)}) - f(x^*)),$$

where $g_k := \nabla f(x^k) = Ax^{(k)} - b$. Lemma 2.3.5 yields

$$\begin{aligned} 1 - \frac{(g_k^\top g_k)^2}{(g_k^\top A g_k)} &\leq 1 - 4\frac{\lambda_{\max}\lambda_{\min}}{(\lambda_{\max} + \lambda_{\min})^2} \\ &= \frac{\lambda_{\max}^2 - 2\lambda_{\max}\lambda_{\min} + \lambda_{\min}^2}{(\lambda_{\max} + \lambda_{\min})^2} \\ &= \frac{(\lambda_{\max} - \lambda_{\min})^2}{(\lambda_{\max} + \lambda_{\min})^2} = \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right) \end{aligned}$$

which proves part 1.

Part 2: For all $x \in \mathbb{R}^n$ there exists, by Theorem 1.1.1, a $z \in [x, x^*]$ such that

$$\begin{aligned} f(x) - f(x^*) &= \nabla f(x^*)^\top (x - x^*) + \frac{1}{2}(x - x^*)\nabla^2 f(z)(x - x^*) \\ &= \frac{1}{2}(x - x^*)^\top A(x - x*) \\ &= \frac{1}{2}\|x - x^*\|_A^2. \end{aligned}$$

From part 1 it follows

$$\frac{1}{2}\|x^{(k+1)} - x^*\|_A^2 \leq \left(\frac{\kappa(A) - 1)}{\kappa(A) + 1}\right)^2 \frac{1}{2}\|x^{(k)} - x^*\|_A^2.$$

Multiplication by 2 and root extraction proves part 2.

Part 3: Since for all $x \in \mathbb{R}^n$ it holds

$$\lambda_{\min}\|x - x^*\| \leq (x - x^*)^\top A(x - x^*) \leq \lambda_{\max}\|x - x^*\|^2$$

we have

$$\begin{aligned} \|x^{(k+1)} - x^*\|^2 &\leq \frac{1}{\lambda_{\min}}\|x^{(k+1)} - x^*\|_A^2 \\ &\leq \frac{1}{\lambda_{\min}}\left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^2 \|x^{(k)} - x^*\|_A^2 \\ &\leq \frac{\lambda_{\max}}{\lambda_{\min}}\left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^2 \|x^{(k)} - x^*\|^2, \end{aligned}$$

which proves part 3. □

Theorem 2.3.6 states linear convergence of the steepest descent method for convex quadratic functions. This result can be extended to arbitrary functions by the idea that, because of Theorem 1.1.1,

$$f(x) - f(x^*) \approx \frac{1}{2}(x - x^*)^\top \nabla^2 f(x^*)(x - x^*)$$

if x is „close enough“ to x^* (cf. Nocedal and Wright, 2006).

Theorem 2.3.7
Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable and let $(x^{(k)})_{k\in\mathbb{N}_0}$ denote the steepest descent sequence with exact step length. Assume $x^{(k)} \to x^*$ with $\nabla^2 f(x^*) > 0$, then it holds

$$f(x^{(k+1)} - f(x^*) \leq \left(\frac{\kappa(\nabla^2 f(x^*)) - 1}{\kappa(\nabla^2 f(x^*)) + 1}\right)^2 (f(x^{(k)}) - f(x^*)) \ .$$

An example a very simple but ill-conditioned problem is

$$\min \frac{1}{2} x^\top A x$$

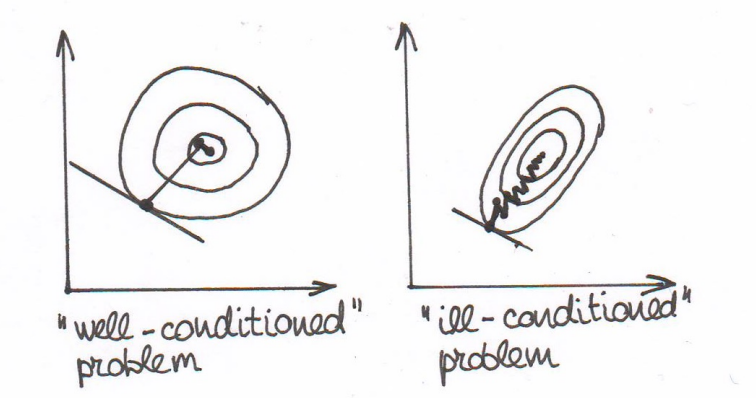
with

$$A = \begin{bmatrix} 1000 & 0 \\ 0 & 1 \end{bmatrix}, \ b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

In this case it is obviously $x^* = (0, 0)^\top$ with $f^* = 0$. Due to $\kappa(A) = 1000$ we obtain with $x^{(0)} = (1, 0)^\top$ that

$$f(x^{(500)}) \approx 0.1$$

after 500 steepest descent iterations.



2.4 The Conjugate Gradient Method

For a symmetric and positive definite matrix A, solving the problem

$$\min_{x\in\mathbb{R}^n} \frac{1}{2} x^\top A x - b^\top x \tag{QP}$$

via the steepest descent method has shown to be very inefficient. The *conjugate gradient* (CG) method overcomes this slow convergence by modifying the steepest descent direction by means of directions from previous iterations.

Definition 2.4.1
Let $A \in \mathbb{R}^{n\times n}$ be symmetric and positive definite. The vectors $d^{(1)}, ..., d^{(m)} \in \mathbb{R}^n \setminus \{0\}$, with $m \leq n$, are called A-conjugate *if $\langle d^{(i)}, d^{(j)}\rangle_A = (d^{(i)})^\top A d^{(j)} = 0$ for all $i \neq j$.*

Lemma 2.4.2
Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite and let $d^{(1)}, \dots, d^{(m)}$ be A-conjugate. Then, the vectors $d^{(1)}, ..., d^{(m)}$ are linearly independent. Especially for $m = n$ they form a basis of $\mathbb{R}^n$.

Proof.
Let

$$\sum_{i=1}^{n} \lambda_i d^{(i)} = 0$$

and let $j \in \{1, ..., n\}$ be arbitrary. Then it holds

$$\begin{aligned} 0 &= (d^{(j)})^\top A \left(\sum_{i=1}^{n} \lambda_i d^{(i)} \right) = \sum_{i=1}^{n} \lambda_i (d^{(j)})^\top A d^{(i)} = \sum_{i=1}^{n} \lambda_i \langle d^{(j)}, d^{(i)}\rangle_A \\ &= \lambda_j \langle d^{(j)}, d^{(i)}\rangle_A = \lambda_j \underbrace{\|d^{(j)}\|_A^2}_{>0}, \end{aligned}$$

such that $\lambda_j = 0$. □

If the vectors $d^{(0)}, ..., d^{(n-1)}$ are A-conjugate, then it holds for all $v \in \mathbb{R}^n$ that

$$v = \sum_{i=0}^{n-1} \lambda_i d^{(i)}$$

for some $\lambda_i \in \mathbb{R}$. Multiplication with $(d^{(k)})^\top A$ yields

$$\begin{aligned} (d^{(k)})^\top A v = \sum_{i=0}^{n-1} \lambda_i \langle d^{(k)}, d^{(i)}\rangle_A = \lambda_k \|d^{(k)}\|_A^2 \\ \Leftrightarrow \lambda_k = \frac{\langle d^{(k)}, v\rangle_A}{\|d^{(k)}\|_A^2}, \end{aligned}$$

such that

$$v = \sum_{k=0}^{n-1} \frac{\langle d^{(k)}, v\rangle_A}{\|d^{(k)}\|_A^2} d(k) \quad \forall v \in \mathbb{R}^n.$$

For $x^* = A^{-1}b$, i.e. the unique solution of (QP), and arbitrary $x \in \mathbb{R}^n$ we obtain with

$v := x^* - x$ that

$$x^* = x + \sum_{k=0}^{n-1} \frac{\langle d^{(k)}, x^* - x\rangle_A}{\|d^{(k)}\|_A^2} d^{(k)} = x + \sum_{k=0}^{n-1} \frac{(d^{(k)})^\top A(x^* - x)}{\|d^{(k)}\|_A^2} d^{(k)}$$
$$= x + \sum_{k=0}^{n-1} \frac{(d^{(k)})^\top \underbrace{(b - Ax)}_{=-\nabla f(x)}}{\|d^{(k)}\|_A^2} d^{(k)} = x + \sum_{k=0}^{n-1} \frac{-\nabla f(x)^\top d^{(k)}}{\|d^{(k)}\|_A^2} d^{(k)}$$

and the right-hand side does not depend on x^*!

This leads to a general conjugate direction method:

Algorithm 2.4.3: Conjugate direction method

Input: $x^{(0)} \in \mathbb{R}^n,\ d^{(0)} \in \mathbb{R}^n, \varepsilon \geq 0\ k \leftarrow 0$
while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
  $t_k \leftarrow \dfrac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|d^{(k)}\|_A^2}$
  $x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}$
  find $d^{(k+1)}$ such that $\langle d^{(k+1)}, d^{(j)}\rangle_A = 0$ for all $j = 0, ..., k$
  $k \leftarrow k + 1$
end

Theorem 2.4.4
For any $x^{(0)} \in \mathbb{R}^n$ the sequence $(x^{(k)})_{k\in\mathbb{N}_0}$ generated by Algorithm 2.4.3 yields the exact solution of (QP) in at most n iterations, i.e. $x^{(n)} = A^{-1}b$.

Proof.
We already know that

$$A^{-1}b = x^* = x^{(0)} + \sum_{k=0}^{n-1} \underbrace{\frac{-\nabla f(x^{(0)})^\top d^{(k)}}{\|d^{(k)}\|_A^2}}_{\overset{!}{=} t_k} d^{(k)}.$$

Since for all $k = 0, ..., n-1$ it holds

$$x^{(k+1)} = x^{(k)} + t_k d^{(k)} = ... = x^{(0)} + \sum_{j=0}^{k} t_j d^{(j)}$$

it follows

$$t_k = \frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|d^{(k)}\|_A^2} = \frac{-(Ax^{(k)} - b)^\top d^{(k)}}{\|d^{(k)}\|_A^2} = \frac{-1}{\|d^{(k)}\|_A^2}[A(x^{(0)} + \sum_{j=0}^{k-1} t_j d^{(j)}) - b]^\top d^{(k)}$$
$$= \frac{-1}{\|d^{(k)}\|_A^2}[(Ax^{(0)} - b)^\top d^{(k)} + \sum_{j=0}^{k-1} t_j \underbrace{\langle d^{(j)}, d^{(k)}\rangle_A}_{=0}]$$
$$= \frac{-1}{\|d^{(k)}\|_A^2}(Ax^{(0)} - b)^\top d^{(k)} = \frac{-\nabla f(x^{(0)})^\top d^{(k)}}{\|d^{(k)}\|_A^2}$$

and finally

$$x^{(n)} = x^{(0)} + \sum_{j=0}^{n-1} \frac{-f(x^{(0)})^\top d^{(k)}}{\|d^{(k)}\|_A^2} d^{(k)} = x^*.$$

□

In order to find A-conjugate vectors, we state the following helpful result:

Lemma 2.4.5
Let $x^{(0)}, ..., x^{(n)}$ be generated by Algorithm 2.4.3. Then it holds

$$\nabla f(x^{(k)})^\top d^{(j)} = 0 \quad \forall 0 \leq j < k,\ k \in \{1, ..., n\}.$$

Proof.
Let $k \in \{1, ..., n\}$ and $j \in \{0, ..., k-1\}$ be arbitrary. Since

$$x^{(k)} = x^{(k-1)} + t_{k-1} d^{(k-1)} = ... = x^{(j)} + \sum_{i=j}^{k-1} t_i d^{(i)}$$

it holds

$$\nabla f(x^{(k)}) = Ax^{(k)} - b = A(x^{(j)} + \sum_{i=j}^{k-1} t_i d^{(i)}) - b = \nabla f(x^{(j)}) + \sum_{i=j}^{k-1} t_i A d^{(i)}$$

such that

$$\nabla f(x^{(k)})^\top d^{(j)} = \nabla f(x^{(j)})^\top d^{(j)} + \sum_{i=j}^{k-1} t_i \langle d^{(i)}, d^{(j)}\rangle_A$$
$$= \nabla f(x^{(j)})^\top d^{(j)} + t_j \|d^{(j)}\|_A^2$$
$$= \nabla f(x^{(j)})^\top d^{(j)} + \frac{-\nabla f(x^{(j)})^\top d^{(j)}}{\|d^{(j)}\|_A^2} \|d^{(j)}\|_A^2$$
$$= 0.$$

□

In order to find A-conjugate vectors we try

$$d^{(0)} = -\nabla f(x^{(0)})$$
$$d^{(k)} = -\nabla f(x^{(k)}) + \sum_{i=0}^{k-1} \beta_i d^{(i)}.$$

We assume that $d^{(0)}, ..., d^{(k-1)}$ are already A-conjugate and we want to choose $\beta_0, ..., \beta_{k-1}$ such that

$$\langle d^{(k)}, d^{(j)} \rangle_A = 0 \quad \forall j = 0, ..., k-1.$$

For $0 \leq j < k$ it holds

$$\begin{aligned}
\langle d^{(k)}, d^{(j)} \rangle_A &= \langle -\nabla f(x^{(k)}) + \sum_{i=0}^{k-1} \beta_j d^{(i)}, d^{(j)} \rangle_A \\
&= -\langle \nabla f(x^{(k)}), d^{(j)} \rangle_A + \sum_{i=0}^{k-1} \beta_i \langle d^{(i)}, d^{(j)} \rangle_A \\
&= -\langle \nabla f(x^{(k)}), d^{(j)} \rangle_A + \beta_j \|d^{(j)}\|_A^2
\end{aligned}$$

such that

$$0 = \langle d^{(k)}, d^{(j)} \rangle_A \Leftrightarrow \beta_j = \frac{\langle \nabla f(x^{(k)}), d^{(j)} \rangle_A}{\|d^{(j)}\|_A^2} = \frac{\nabla f(x^{(k)})^\top A d^{(j)}}{(d^{(j)})^\top A d^{(j)}}.$$

Since $d^{(j)} = \frac{1}{t_j}(x^{(j+1)} - x^{(j)})$ it follows

$$\beta_j = \frac{\nabla f(x^{(k)})^\top A(x^{(j+1)} - x^{(j)})}{(d^{(j)})^\top A(x^{(j+1)} - x^{(j)})} = \frac{\nabla f(x^{(k)})^\top (\nabla f(x^{(j+1)}) - \nabla f(x^{(j)}))}{(d^{(j)})^\top (\nabla f(x^{(j+1)}) - \nabla f(x^{(j)}))}.$$

For $j < k$ we have by Lemma 2.4.5 that

$$\begin{aligned}
0 = \nabla f(x^{(k)})^\top d^{(j)} &= \nabla f(x^{(k)})^\top (-\nabla f(x^{(j)}) + \sum_{i=0}^{j-1} \beta_i d^{(i)}) \\
&= -\nabla f(x^{(k)})^\top \nabla f(x^{(j)}) + \sum_{i=0}^{j-1} \beta_i \nabla f(x^{(k)})^\top d^{(i)} \\
&= -\nabla f(x^{(k)})^\top \nabla f(x^{(j)})
\end{aligned}$$

and we conclude

$$\begin{aligned}
\beta_j &= 0 \quad \forall j = 0, ..., k-2 \\
\beta_{k-1} &= \frac{\|\nabla f(x^{(k)})\|^2}{(d^{(k-1)})^\top (\nabla f(x^{(k)}) - \nabla f(x^{(k-1)}))} \ .
\end{aligned} \tag{Dixon}$$

Since

$$d^{(k-1)} = -\nabla f(x^{(k-1)}) + \underbrace{\sum_{i=0}^{k-2} \beta_i d^{(i)}}_{=0, \text{since } \beta_i = 0} = -\nabla f(x^{(k-1)})$$

we also have

$$\beta_{k-1} = \frac{\|\nabla f(x^{(k)})\|^2}{\underbrace{-\nabla f(x^{(k-1)})^\top \nabla f(x^{(k)})}_{=0} + \|\nabla f(x^{(k-1)})\|^2} = \frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla f(x^{(k-1)})\|^2} \ . \qquad \text{(Fletcher-Reeves)}$$

Finally, we obtain A-conjugate vectors via

$$\begin{aligned} d^{(0)} &= -\nabla f(x^{(0)}), \\ d^{(k)} &= -\nabla f(x^{(k)}) - \frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla f(x^{(k-1)})\|^2} d^{(k-1)} \quad k > 0 \ . \end{aligned}$$

Summarizing these results yields the *conjugate gradient* (CG) method:

Algorithm 2.4.6: CG-method

Input: $x^{(0)} \in \mathbb{R}^n,\ \varepsilon \geq 0$

Set $d^{(0)} \leftarrow r^{(0)} \leftarrow b - Ax^{(0)}\ (= -\nabla f(x^{(0)}))$

while $\|r^{(k)}\| > \varepsilon$ **do**

  $t_k \leftarrow \dfrac{\|r^{(k)}\|^2}{\|d^{(k)}\|_A^2}$

  $x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}$

  $r^{(k+1)} \leftarrow r^{(k)} - t_k A d^{(k)}$

  $\beta_k \leftarrow \dfrac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$

  $d^{(k+1)} \leftarrow r^{(k+1)} + \beta_k d^{(k)}$

  $k \leftarrow k + 1$

end

Remark 2.4.7

1. *By construction, the CG-method yields the exact solution in at most n iterations. If A has only $m \leq n$ distinct eigenvalues, then $x^{(m)} = A^{-1}b = x^*$, i.e. the CG-iteration terminates in at most m iterations.*

2. *Despite that the CG-method finds the exact solution, in practice it is used as iterative method due to rounding impacts and possible large dimension n.*

3. *The computational cost of one CG-iteration corresponds mainly to the cost of computing the product $Ad^{(k)}$. Therefore, the structure of A can be exploited to drastically reduce memory requirements and to speed up the computation.*

4. *The CG-method computes, $x^* = A^{-1}b$, i.e. the solution of the linear system $Ax \stackrel{!}{=} b$ and is therefore frequently used in this context.*

5. *The CG-method is indeed a line-search method:*

$$\begin{aligned}\nabla f(x^{(k)})^\top d^{(k)} &= \nabla f(x^{(k)})^\top(-\nabla f(x^{(k)}) + \sum_{i=0}^{k-1}\beta_i d^{(i)}) \\ &= -\|\nabla f(x^{(k)})\|^2 - \sum_{i=0}^{k-1}\beta_i\langle\nabla f(x^{(k)}), d^{(i)}\rangle \\ &= -\|\nabla f(x^{(k)})\|^2 < 0,\end{aligned}$$

i.e. $d^{(k)}$ is a descent direction of f in $x^{(k)}$. Further, since for

$$\phi(t) := f(x^{(k)} + td^{(k)})$$

it holds

$$\begin{aligned}\phi'(t) &= \nabla f(x^{(k)} + td^{(k)})^\top d^{(k)} = (A(x^{(k)} + td^{(k)}) - b)\top d^{(k)} \\ &= \nabla f(x^{(k)})^\top d^{(k)} + t\|d^{(k)}\|_A^2 \overset{!}{=} 0 \\ \Leftrightarrow \qquad t &= \frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|d^{(k)}\|_A^2} =: t_k\end{aligned}$$

and since

$$\phi''(t) = \|d^{(k)}\|_A^2 > 0$$

the step length t_k is the optimal step length.

If the CG-method is used as an iterative method we obtain linear convergence (cf. Nocedal and Wright, 2006, Chapter 5):

Theorem 2.4.8
For the iterates of the CG-method $x^{(0)}, ..., x^{(n)}$ it holds

$$\|x^{(k+1)} - x^*\|_A \leq \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)\|x^{(k)} - x^*\|_A \ ,$$

where $\kappa(A) = \lambda_{\max}/\lambda_{\min}$ denotes the condition number of A.

This shows that the CG-method has a linear rate of convergence, just as steppest descent method. The convergence factor of the CG-method, however, is significantly smaller since $\sqrt{\kappa(A)} \ll \kappa(A)$, especially if $\kappa(A)$ is large (cf. Theorem 2.3.6). Further, keep in mind that the CG-method will terminate in at most n iterations, which is not true for the steepest descent method. The computational costs of both methods, however, are similar. In analogy to the steepest descent, the CG-method can be applied to arbitrary objective function f, where $\nabla^2 f(x^*) > 0$. The finite termination property is lost in this case but the convergence is still much faster than it is in the steepest descent method.

Since the CG-method solves the linear system $Ax \stackrel{!}{=} b$, we can improve the convergence rate by means of *preconditioning* techniques. That is, we find a „simple“ matrix P such that

$$\kappa(P^{-1}A) \ll \kappa(A)$$

and solve the preconditioned system

$$(P^{-1}A)x = P^{-1}b \qquad (\Leftrightarrow P^{-1}(Ax - b) \stackrel{!}{=} 0)$$

by means of the CG-method. Note that the preconditioner P is of course not explicitly inverted but only used implicitly as a solution of a linear system leading to the *preconditioned CG-method* (PCG-method), presented in Algorithm 2.4.9. Widely used preconditioners are for example

- Jacobi-method and (symmetric) Gauss-Seidel method,
- Incomplete Fatorization (ILU),
- Multigrid methods.

Algorithm 2.4.9: PCG

Input: $x^{(0)}$, $\varepsilon \geq 0$

$r^{(0)} \leftarrow b - Ax^{(0)},\ d^{(0)} \leftarrow z^{(0)} \leftarrow P^{-1}r^{(0)},\ k \leftarrow 0$

while $\|r^{(k)}\| > \varepsilon$ **do**

  $t_k \leftarrow \dfrac{\langle z^{(k)}, r^{(k)} \rangle}{\|d^{(k)}\|_A^2}$

  $x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}$

  $r^{(k+1)} \leftarrow r^{(k)} - t_k A d^{(k)}$

  $z^{(k+1)} \leftarrow P^{-1}r^{(k)}$

  $\beta_{k+1} \leftarrow \frac{\langle z^{(k+1)}, r^{(k+1)} \rangle}{\langle z^{(k)}, r^{(k)} \rangle}$

  $d^{(k+1)} \leftarrow z^{(k+1)} + \beta_k d^{(k)}$

  $k \leftarrow k + 1$

end

2.5 Newton Method

We now turn back to the general minimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{PU}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. If $\nabla f(x) \neq 0$ and $\nabla^2 f(x) > 0$ for some $x \in \mathbb{R}^n$, then Lemma 2.2.5 yields that

$$-\nabla^2 f(x)^{-1} \nabla f(x)$$

provides a descent direction of f at x, which is referred to as *Newton direction*. The motivation is based on the Taylor expansion for ∇f (cf. Theorem 1.1.1), i.e.

$$\nabla f(x+h) \approx \nabla f(x) + \nabla^2 f(x)h =: m(h).$$

If $\nabla^2 f(x)$ is nonsingular, it holds

$$m(h) = 0 \Leftrightarrow h = -\nabla^2 f(x)^{-1}\nabla f(x)$$

leading to the new approximation

$$x^{new} = x + h = x + \nabla^2 f(x)^{-1}\nabla f(x)$$

such that the Newton direction provides a natural step length of 1.

Algorithm 2.5.1: general Newton method

Input: $x^{(0)} \in \mathbb{R}^n,\ \varepsilon \geq 0,\ k \leftarrow 0$

1 **while** $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
2 $\quad d^{(k)} \leftarrow -\nabla^2 f(x^{(k)})^{-1}\nabla f(x^{(k)})$ $\quad$ || solve linear system
3 $\quad x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}$
4 $\quad k \leftarrow k+1$
5 **end**

In order to analyse the convergence behaviour of the Newton method we state the following results.

Lemma 2.5.2
Let $g : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and $\nabla g : \mathbb{R}^n \to \mathbb{R}^n$ be Lipschitz at some $z \in \mathbb{R}^n$, i.e.

$$\exists \varepsilon > 0, L > 0 : \|\nabla g(x) - \nabla g(y)\| \leq L\|x - y\| \quad \forall x, y \in B_\varepsilon(z).$$

Then it holds

$$\|g(x) - g(y) - \nabla g(y)^\top (x-y)\| \leq \frac{L}{2}\|x-y\|^2 \quad \forall x, y \in B_\varepsilon(z).$$

Lemma 2.5.3 (Banach-Lemma)
Let $A, B \in \mathbb{R}^{n\times n}$, A nonsingular, and $\|A^{-1}\|\|B\| < 1$. Then, $A + B$ is nonsingular and

$$\|(A+B)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\|B\|}.$$

Theorem 2.5.4 (Local quadratic convergence of Newtons method)
Let $f \in C^2(\mathbb{R}^n)$, $x^* \in \mathbb{R}^n$ with $\nabla f(x^*) = 0$, and $\nabla^2 f(x^*) > 0$ (i.e. x^* is a strict local solution of (PU)), and let $\nabla^2 f$ be locally Lipschitz at x^*. Then there exists $\varepsilon > 0$ such that for all $x^{(0)} \in B_\varepsilon(x^*)$ it holds $x^{(k)} \to x^*$ quadratically, where $(x^{(k)})_{k\in\mathbb{N}_0}$ is generated by the Newton method.

Proof.
Since $\nabla^2 f$ is locally Lipschitz at x^*

$$\exists \tilde{\varepsilon} > 0 : \|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq L\|x - x^*\| \quad \forall x \in B_{\tilde{\varepsilon}}(x^*)$$

such that for

$$\varepsilon := \min\left\{\tilde{\varepsilon},\ \frac{1}{2L\|\nabla^2 f(x^*)^{-1}\|}\right\}$$

it holds

$$\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq \frac{1}{2\|\nabla^2 f(x^*)^{-1}\|} \quad \forall x \in B_\varepsilon(x^*).$$

Hence

$$\|\underbrace{\nabla^2 f(x^*)^{-1}}_{=:A^{-1}}\|\|\underbrace{\nabla^2 f(x) - \nabla^2 f(x^*)}_{=:B}\| \leq \frac{1}{2} < 1$$

such that Lemma 2.5.3 yields that $\nabla^2 f(x)$ is nonsingular for all $x \in B_\varepsilon(x^*)$ and

$$\|\nabla^2 f(x)^{-1}\| \leq \frac{\|\nabla^2 f(x^*)^{-1}\|}{1 - \|\nabla f(x^*)^{-1}\|\|\nabla^2 f(x) - \nabla^2 f(x^*)\|} \leq 2\|\nabla^2 f(x^*)^{-1}\|. \tag{1}$$

Further, by Lemma 2.5.2 it holds

$$\begin{aligned}
&\|\nabla f(x^*) - \nabla f(x) - \nabla^2 f(x)(x^* - x)\| \leq \frac{L}{2}\|x^* - x\|^2 \quad \forall x \in B_\varepsilon(x^*) \\
\Leftrightarrow\ &\|\nabla^2 f(x)(x - x^*) - \nabla f(x)\| \leq \frac{L}{2}\|x^* - x\|^2 \quad \forall x \in B_\varepsilon(x^*)
\end{aligned} \tag{2}$$

such that for $x^{(0)} \in B_\varepsilon(x^*)$ we have

$$\begin{aligned}
\|x^{(1)} - x^*\| &= \|x^{(0)} - \nabla^2 f(x^{(0)})^{-1}\nabla f(x^{(0)}) - x^*\| \\
&\leq \|\nabla^2 f(x^{(0)})^{-1}\|\|\nabla^2 f(x^{(0)})(x^{(0)} - x^*) - \nabla f(x^{(0)})\| \\
&\overset{(2)}{\leq} \|\nabla^2 f(x^{(0)})^{-1}\|\frac{L}{2}\|x^* - x^{(0)}\|^2 \\
&\leq \|\nabla^2 f(x^{(0)})^{-1}\|\frac{L}{2}\|x^* - x^{(0)}\|\varepsilon \\
&\overset{(1)}{\leq} 2\|\nabla^2 f(x^*)^{-1}\|\frac{L}{2}\|x^* - x^{(0)}\|\varepsilon \\
&\overset{\text{def. of } \varepsilon}{\leq} \frac{1}{2} \cdot \varepsilon \\
&< \varepsilon
\end{aligned}$$

and therefore $x^{(1)} \in B_\varepsilon(x^*)$. By induction we conclude $x^{(k)} \in B_\varepsilon(x^*)$ $\forall k \in \mathbb{N}_0$ and

$$\|x^{(k)} - x^*\| \leq \left(\frac{1}{2}\right)^k \cdot \varepsilon$$

such that

$$\lim_{k\to\infty} x^{(k)} = x^*.$$

Further,

$$\begin{aligned}
\|x^{(k+1)} - x^*\| &= \|x^{(k)} - \nabla^2 f(x^{(k)})^{-1}\nabla f(x^{(k)}) - x^*\| \\
&\leq \|\nabla^2 f(x^{(k)})^{-1}\| \|\nabla^2 f(x^{(k)})(x^{(k)} - x^*) - \nabla f(x^{(k)})\| \\
&= \|\nabla^2 f(x^{(k)})^{-1}\| \|\nabla f(x^{(*)}) - \nabla f(x^{(k)}) - \nabla^2 f(x^{(k)})(x^* - x^{(k)})\| \\
&\overset{(2)}{\leq} \|\nabla^2 f(x^{(k)})^{-1}\| \frac{L}{2} \|x^* - x^{(k)}\|^2 \\
&= \underbrace{\|\nabla^2 f(x^*)^{-1}\| L}_{<c<\infty} \|x^{(k)} - x^*\|^2 \ ,
\end{aligned}$$

such that

$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^*|}{\|x^{(k)} - x^*\|^2} \leq c < \infty,$$

i.e. $x^{(k)} \to x^*$ quadratically for all $x^{(0)} \in B_\varepsilon(x^*)$. □

Remark 2.5.5

1. *A small modification of the proof shows that, if $\nabla^2 f$ is only continuous, the same result holds true with superlinear convergence rate.*
2. *The Newton method is computationally expensive (especially if n is large) since in every iteration the Hessian $\nabla^2 f(x^{(k)})$ has to be computed and a linear system has to be solved. This effort is rewarded with a very fast rate of convergence.*
3. *The Newton method is only local convergent, i.e. it converges to x^* only if $x^{(0)} \in B_\varepsilon(x^*)$.*
4. *The computational costs can be drastically reduced by replacing $\nabla^2 f(x^{(k)})$ with $\nabla^2 f(x^{(0)})$ or by updating $\nabla^2 f(x^{(j)})$ every j steps (*simplified Newton method*). We then obtain linear convergence but with a fast rate of convergence.*
5. *To reduce the computational cost we can solve the Newton system $\nabla^2 f(x^{(k)})d^{(k)} = -\nabla f(x^{(k)})$ only approximately, i.e. $d^{(k)} \approx -\nabla^2 f(x^{(k)})^{-1}\nabla f(x^{(k)})$, for example by only using a few iterations of the CG-method (*inexact Newton method*).*
6. *In order to not explicitly compute the derivates ∇f and $\nabla^2 f$ one can use finite-difference approximations (*finite difference Newton method*) and still obtain superlinear convergence.*
7. *The Newton method solves in general a nonlinear system $F(x) \overset{!}{=} 0$, where $F : \mathbb{R}^n \to \mathbb{R}^n$ (here $F = \nabla f$).*

For an arbitrary initial guess $x^{(0)}$ the Newton method finds a stationary point of f (i.e. a solution of $\nabla f(x) \overset{!}{=} 0$) but we cannot guarantee that this is a minimizer of f (only if $x^{(0)}$ is „close“ to a minimizer). To overcome this problem and to prevent the Newton method to converge to a maximizer of f, we can apply a line-search strategy that ensures $f(x^{(k+1)}) < f(x^{(k)})$. In order to maintain the fast convergence rate of the Newton method we can use a backtracking line-search with $t_{\max} = 1$ and stopping criterion $(-\nabla f(x^{(k)})^\top d^{(k)} < 0)$. This idea leads to the *damped Newton method*:

Algorithm 2.5.6: Damped Newton method

Input: $x^{(0)} \in \mathbb{R}^n,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**

  $d^{(k)} \leftarrow -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$; `// solve linear system`

  t_k via backtracking with $t_{\max} = 1$ and stopping criterion

  $x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}$

  $k \leftarrow k + 1$

end

Theorem 2.5.7
Let $f \in C^2(\mathbb{R}^n)$ be strictly convex on the level set L_0 and let x^* be a stationary point of f, i.e. a strict local solution of (PU). Then it holds for the iterates of the damped Newton method that $x^{(k)} \to x^*$.

Proof.
Since f is strictly convex the level set L_0 is compact and the Newton direction is a descent direction. Therefore, $x^{(k)} \in L_0$ for all $k \in \mathbb{N}_0$ and since $\nabla^2 f(x) > 0$ for all $x \in L_0$ it holds

$$\begin{aligned}\frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|\nabla f(x^{(k)})\|\|d^{(k)}\|} &= \frac{\nabla f(x^{(k)})^\top \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|\|\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})\|} \\ &\geq \frac{\|\nabla^2 f(x^{(k)})^{-1}\|\|\nabla f(x^{(k)})\|}{\|\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})\|} \geq 1 > 0\end{aligned}$$

which is the angle condition. Theorem 2.2.10 yields that x^* is an accumulation point of $(x^{(k)})_{k \in \mathbb{N}_0}$. Since f is strictly convex on L_0, x^* is the only stationary point, such that $x^{(k)} \to x^*$. □

If f is not strictly convex on L_0, the Newton method direction might not be a descent direction. In that case, we can try to combine the Newton method with a globally convergent method, such as the steepest descent method. However, this comes along with a slow overall convergence rate. Very popular alternatives are

- quasi-Newton method (cf. Section 2.6),
- trust-region Newton method (cf. Section 2.7.1).

2.6 Quasi-Newton Method

The success of the Newton method is based on the use of second order information, i.e. Hessians. In practice, computing the Hessian and solving the Newton system can be very difficult and expensive. Quasi-Newton methods generate a series of Hessian approximations without computing the Hessians while, at the same time, maintain a fast rate of convergence.

The basic idea is to find a matrix $B_k \approx \nabla^2 f(x^{(k)})$ in some sense and, in analogy to the Newton method, to update

$$x^{(k+1)} \leftarrow x^{(k)} - t_k B_k^{-1} \nabla f(x^{(k)}) \ .$$

For the matrix B_k we want

- $B_k \approx \nabla^2 f(x^{(k)})$ to maintain the fast convergence rate of Newton method,
- $B_k > 0$, such that $-B_k^{-1}\nabla f(x^{(k)})$ is a descent direction,
- B_k should be of simple structure, such that the linear system $B_k d^{(k)} = -\nabla f(x^{(k)})$ is efficiently to solve,
- B_{k+1} should be updated from B_k, i.e. $B_{k+1} = B_k + U_k$, where U_k is a matrix of simple structure.

Assume we have computed the iterate $x^{(k+1)}$. What requirements should be imposed on B_{k+1}? By Theorem 1.1.1 we have

$$\begin{aligned} &\nabla f(x^{(k)}) \approx \nabla f(x^{(k+1)}) + \nabla^2 f(x^{(k+1)})(x^{(k)} - x^{(k+1)}) \\ \Rightarrow\ &\nabla^2 f(x^{(k+1)}) \underbrace{(x^{(k+1)} - x^{(k)})}_{=:s^{(k)}} \approx \underbrace{\nabla f(x^{(k+1)}) - \nabla f(x^{(k)})}_{=:y^{(k)}} \end{aligned}$$

and in order to make B_{k+1} „behave like“ $\nabla^2 f(x^{(k+1)})$, we claim

$$B_{k+1} s^{(k)} = y^{(k)} c \ . \qquad \text{(quasi-Newton condition / secant equation)}$$

Multiplication with $(s^{(k)})^\top$ yields that

$$(s^{(k)})^\top B_{k+1} s^{(k)} = (s^{(k)})^\top y^{(k)} = \langle s^{(k)}, y^{(k)} \rangle > 0 \qquad \text{(curvature condition)}$$

is a necessary condition for B_{k+1} being positive definite. Note that, if t_k meets the Wolfe conditions, we have

$$\begin{aligned} \langle s^{(k)}, y^{(k)} \rangle &= t_k (d^{(k)})^\top y_k = t_k (\nabla f(x^{(k+1)})^\top d^{(k)} - \nabla f(x^{(k)})^\top d^{(k)}) \\ &\geq t_k (c_2 - 1) \nabla f(x^{(k+1)})^\top d^{(k)} \\ &> 0, \end{aligned}$$

i.e. the curvature condition holds automatically true.

Algorithm 2.6.1: General quasi-Newton method

Input: $x^{(0)} \in \mathbb{R}^n,\ \mathbb{R}^{n\times n} \ni B_0 > 0,\ \varepsilon > 0,\ k \leftarrow 0$

1 **while** $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
2 $\quad d^{(k)} \leftarrow -B_k^{-1}\nabla f(x^{(k)})$; `// solve linear system`
3 $\quad t_k$ that meets Wolfe condition
4 $\quad x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}$
5 $\quad$ find $B_{k+1} > 0$ that meets quasi-Newton condition
6 $\quad k \leftarrow k + 1$
7 **end**

2.6.1 BFGS-update

For an efficient algorithm we need an update formula for B_{k+1}, i.e.

$$B_{k+1} = B_k + U_k,$$

such that B_{k+1} is symmetric positive definite and meets the quasi-Newton condition and U_k does not require any additional evaluations of f or ∇f.

Since the quasi-Newton condition represents n conditions but there are $n(n+1)/2$ free entries of a symmetric $n \times n$ matrix, there are many possible update rules to obtain B_{k+1} from B_k. The most popular approaches are symmetric rank-2 updates, i.e.

$$B_{\text{new}} = B_{\text{old}} + \alpha u u^\top + \beta v v^\top$$

with $\alpha, \beta \in \mathbb{R}$ and $u, v \in \mathbb{R}^n$. The quasi-Newton condition yields

$$y^{(k)} = B_{k+1} s^{(k)} = B_k s^{(k)} + \alpha u u^\top s^{(k)} + \beta v v^\top s^{(k)}$$

such that

$$\alpha = \frac{1}{u^\top s^{(k)}}, \ \beta = -\frac{1}{v^\top s^{(k)}} \ .$$

Broyden, Fletcher, Goldfarb, and Shanno (1970) proposed $u = y^{(k)}$ and $v = B_k s^{(k)}$ leading to the update formula

$$B_{k+1}^{\text{BFGS}} = B_k + \frac{1}{(y^{(k)})^\top s^{(k)}} y^{(k)} (y^{(k)})^\top - \frac{1}{\langle s^{(k)}, s^{(k)} \rangle_{B_k}} B_k s^{(k)} (B_k s^{(k)})^\top. \qquad \text{(BFGS-update)}$$

Lemma 2.6.2
Let B_k be symmetric positive definite and let $\langle y^{(k)}, s^{(k)} \rangle > 0$. Then, B_{k+1}^{BFGS} is also symmetric positive definite.

Proof.
The symmetry of B_{k+1}^{BFGS} is obvious by construction. For simplicity we omit indices for the rest of the proof. To show the positive definiteness of B^{BFGS} we recall the Cauchy-Schwarz inequality

$$\langle u, v \rangle_B^2 \leq \langle u, u \rangle_B \cdot \langle v, v \rangle_B \quad \forall u, v \in \mathbb{R}^n, \ B > 0$$

with strict inequality if u, v are linearly independent. For arbitrary $u \in \mathbb{R}^n \setminus \{0\}$ it holds

$$\begin{aligned} u^\top B^{\text{BFGS}} u &= \langle u, u \rangle_B + \frac{1}{\langle y, s \rangle} u^\top y y^\top u - \frac{1}{\langle s, s \rangle_B} u^\top B s (B s)^\top u \\ &= \langle u, u \rangle_B + \frac{\langle u, y \rangle^2}{\langle y, s \rangle} - \frac{\langle u, s \rangle_B^2}{\langle s, s \rangle_B}. \end{aligned}$$

If u and s are linearly independent, the Cauchy-Schwarz inequality yields

$$\begin{aligned} u^\top B^{BFGS} u &> \langle u, u \rangle_B + \frac{\langle u, y \rangle^2}{\langle y, s \rangle} - \frac{\langle u, u \rangle_B \langle s, s \rangle_B}{\langle s, s \rangle_B} \\ &= \frac{\langle u, y \rangle^2}{\langle y, s \rangle} \geq 0. \end{aligned}$$

Otherwise, if $u = \lambda s$ for some $\lambda \neq 0$ it holds

$$\langle u, y\rangle = \lambda\langle s, y\rangle \neq 0$$

such that

$$\begin{aligned} u^\top B^{\mathrm{BFGS}} u &= \langle u, u\rangle_B + \frac{\lambda^2\langle s, y\rangle^2}{\langle s, y\rangle} - \underbrace{\frac{\lambda^2\langle s, s\rangle_B^2}{\langle s, s\rangle_B}}_{=\langle u,u\rangle_B} \\ &= \lambda^2\langle s, y\rangle > 0 \end{aligned}$$

□

2.6.2 Inverse BFGS-update

The inverse BFGS-update for Hessian approximation directly yields an approximation for its inverse, which is computationally much more convenient (matrix-vector product instead of solving a linear system).

Lemma 2.6.3 (Sherman-Morrison-Woodburry)
Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. Let $U, V \in \mathbb{R}^{n\times p}$ be arbitrary and let $(I_p + V^\top A^{-1}U)$ be nonsingular. Then $(A + UV^\top)$ is nonsingular and

$$(A + UV^\top)^{-1} = A^{-1} - A^{-1}U\underbrace{[I_p + V^\top A^{-1}U]^{-1}}_{\in p\times p}V^\top A^{-1}.$$

Especially, if $p = 1$ it holds

$$(A + UV^\top)^{-1} = A^{-1} - \frac{1}{1 + V^\top A^{-1}U}A^{-1}UV^\top A.$$

With $H_k := B_k^{-1}$ the SMW-formula applied to the BFGS-update yields

$$\begin{aligned} H_{k+1}^{\mathrm{BFGS}} &= H_k - \frac{1}{\langle y^{(k)}, s^{(k)}\rangle}(s^{(k)}(y^{(k)})^\top H_k + H_k y^{(k)}(s^{(k)})^\top) \\ &\quad + \frac{1}{\langle y^{(k)}, s^{(k)}\rangle^2}(\langle s^{(k)}, y^{(k)}\rangle + \langle y^{(k)}, y^{(k)}\rangle_{H_k})s^{(k)}(s^{(k)})^\top \\ H_{k+1}^{\mathrm{BFGS}} &= H_k + U_k^{\mathrm{BFGS}}\ , \end{aligned}$$

where

$$\begin{aligned} U_k^{\mathrm{BFGS}} := &- \frac{1}{\langle y^{(k)}, s^{(k)}\rangle}(s^{(k)}(y^{(k)})^\top H_k + H_k y^{(k)}(s^{(k)})^\top) \\ &+ \frac{1}{\langle y^{(k)}, s^{(k)}\rangle^2}(\langle s^{(k)}, y^{(k)}\rangle + \langle y^{(k)}, y^{(k)}\rangle_{H_k})s^{(k)}(s^{(k)})^\top \end{aligned}$$

denotes the *inverse BFGS-update*.

Algorithm 2.6.4: Quasi-Newton method with inverse BFGS-update

Input: $x^{(k)} \in \mathbb{R}^n,\ H_0 = I,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
- $d^{(k)} \leftarrow -H_k \nabla f(x^{(k)})$; `// matrix-vector product`
- t_k that meets Wolfe condition
- $x^{k+1} \leftarrow x^{(k)} + t_k d^{(k)}$
- $H_{k+1} \leftarrow H_k + U_k^{\text{BFGS}}$
- $k \leftarrow k + 1$

end

Since $H_k > 0$ we obtain, in analogy to the proof of Theorem 2.5.7, that $d^{(k)}$ meets the angle condition. Therefore, if there exists a vector $x^* \in \mathbb{R}^n$ with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) > 0$, we obtain $x^{(k)} \to x^*$ for the iterates of Algorithm 2.6.4.

Analysis of the convergence rate, however, is very technical and requires much more preparation. Therefore we only cite the following result.

Theorem 2.6.5
Let $f \in C^2(\mathbb{R}^n)$ and let $x^* \in \mathbb{R}^n$ be such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) > 0$. Then the iterates of the inverse BFGS-method fulfil

$$x^{(k)} \to x^*$$

globally superlinear.

Sketch of proof.
Dennis-More characterization of superlinear convergence:

$$\begin{aligned} x^{(k)} \to x^* \text{ superlinear} &\Leftrightarrow \lim_{k\to\infty} \frac{\|B_k s^{(k)} - y^{(k)}\|}{\|s^{(k)}\|} = 0 \\ &\Leftrightarrow \underbrace{t_k \to 1,\ B_k \to \nabla^2 f(x^*)}_{\text{to show}} . \end{aligned}$$

□

2.6.3 Limited Memory BFGS-update

The BFGS-method is one of the most efficient methods to solve (PU) and can even be applied to large-scale problems ($n \approx 10^{-5} - 10^{-6}$) after some modifications. This variant is called *limited memory BFGS-method* (L-BFGS method). Instead of computing and storing the $n \times n$ matrix H_k the L-BFGS only requires storage of a few n-dimensional vectors to compute the matrix-vector product $d^{(k)} \leftarrow -H_k \nabla f(x^{(k)})$.

We define

$$\rho_k := \frac{1}{\langle y^{(k)}, s^{(k)} \rangle} \in \mathbb{R}, \quad V_k := I - \rho_k y^{(k)} (s^{(k)})^\top \in \mathbb{R}^{n \times n},$$

such that

$$H_{k+1}^{\text{BFGS}} = V_k^\top H_k V_k + \rho_k s^{(k)} (s^{(k)})^\top.$$

Using the last m vector pairs $(s^{(i)}, y^{(i)}),\ i = k-m, ..., k-1$ it follows

$$\begin{aligned} H_k^{\text{BFGS}} &= (V_{k-1}^\top V_{k-2}^\top \cdot ... \cdot V_{k-m}^\top) H_{k-m} (V_{k-m} V_{k-m+1} \cdot ... \cdot V_{k-1}) \\ &+ \rho_{k-m} (V_{k-1}^\top \cdot ... \cdot V_{k-m+1}^\top) s^{(k-m)} (s^{(k-m)})^\top (V_{k-m+1} \cdot ... \cdot V_{k-1}) \\ &+ \rho_{k-m+1} (V_{k-1}^\top \cdot ... \cdot V_{k-m+2}^\top) s^{(k-m+1)} (s^{(k-m+1)})^\top (V_{k-m+2} \cdot ... \cdot V_{k-1}) \\ &+ ... \\ &+ \rho_{k-1} s^{(k-1)} (s^{(k-1)})^\top, \end{aligned}$$

where

$$H_k^{(0)} = \gamma_k I_n,\ \gamma_k = \frac{\langle s^{(k-1)}, y^{(k-1)} \rangle}{\langle y^{k-1}, y^{(k-1)} \rangle}\ .$$

For the BFGS-method we do not need to explicitly know the matrix H_k^{BFGS} but only its action on the vector $-\nabla f(x^{(k)})$, i.e.

$$d^{(k)} \leftarrow -H_k \nabla f(x^{(k)})\ .$$

Since

$$\begin{aligned} V_i \nabla f(x^{(k)}) &= (I - \rho_i y^{(i)} (s^{(i)})^\top) \nabla f(x^{(k)}) \\ &= \nabla f(x^{(k)}) - \rho_i \langle s^{(i)}, \nabla f(x^{(k)}) \rangle y^{(i)} \end{aligned}$$

for $i = k-1, ..., k-m$ we obtain the following recursion to compute the desired matrix-vector product.

Algorithm 2.6.6: Two-loop recursion to compute $H_k \nabla f(x^{(k)})$

Input: $\nabla f(x^{(k)}),\ (s^{(k)}, y^{(k)})$ for $k = k-1, ..., k-m$

$q \leftarrow \nabla f(x^{(k)})$
for $i = k-1, ..., k-m$ **do**
  $\rho_i \leftarrow \langle y^{(i)}, s^{(i)} \rangle$
  $\alpha_i \leftarrow \rho_i \langle s^{(i)}, q \rangle$
  $q \leftarrow q - \alpha_i y^{(i)}$
end
$r \leftarrow H_k^{(0)} q \quad (= \gamma_k q)$
for $i = k-m, ..., k-1$ **do**
  $\beta_i \leftarrow \rho_i \langle y^{(i)}, r \rangle$
  $r \leftarrow r + (\alpha_i - \beta_i) s^{(i)}$
end
return $r\ (= H_k \nabla f(x^{(k)}))$

Using this matrix vector product, we obtain the *L-BFGS method*. Note that the choice of m depends strongly on the problem dimension n. Usually $3 \leq m \leq 30$.

Algorithm 2.6.7: L-BFGS method

Input: $x^{(0)} \in \mathbb{R}^n,\ \varepsilon \geq 0,\ m \in \mathbb{N},\ k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
  $d^{(k)} \leftarrow -H_k \nabla f(x^{(k)})$; // Algorithm 2.6.6
  t_k that meets Wolfe condition
  $x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}$
  if $k > m$ **then**
    delete $s^{(k-m)}, y^{(k-m)}$
    $s^{(k)} \leftarrow x^{(k+1)} - x^{(k)}$
    $y^{(k)} \leftarrow \nabla f(x^{(k+1)} - \nabla f(x^{(k)})$
  end
  $k \leftarrow k + 1$
end

2.7 Trust-Region Methods

In Theorem 2.5.7 we have shown that the damped Newton method is globally convergent, provided $\nabla^2 f(x^{(k)}) > 0$ for all $k \in \mathbb{N}_0$. If this condition is not met, the globalization via a line-search is not possible. In this case trust-region (TR) methods provide a valuable alternative.

As in the (quasi-) Newton method we consider the quadratic approximation

$$f(x^{(k)} + d) \approx f(x^{(k)}) + \nabla f(x^{(k)})^\top d + \frac{1}{2} d^\top B_k d =: m_k(d)$$

This approximation is accurate provide $\|d\| \leq \Delta_k$ such that we determine $d^{(k)}$ as solution of

$$\min_{\|d\| \leq \Delta_k} m_k(d) \tag{TR}$$

and update the new iterate

$$x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}.$$

Note that (TR) always has a solution since m_k is continuous and the set

$$\{d \in \mathbb{R}^n : \|d\| \leq \Delta_k\}$$

is compact. The set

$$\{x \in \mathbb{R}^n : \|x - x^{(k)}\| \leq \Delta_k\}$$

where we believe that the quadratic model is accurate is called *trust-region* and the number $\Delta_k > 0$ the *trust-region radius*. The main problems are:

- How to solve the TR-subproblem (TR)?
- How to choose the TR-radius Δ_k?

2.7.1 The TR-Newton Method

For the TR-Newton method we set

$$B_k = \nabla^2 f(x^{(k)}).$$

In order to set the TR-radius Δ_k, we consider the actual reduction

$$f(x^{(k)}) - f(x^{(k)} + d^{(k)}),$$

i.e. the reduction of the objective function as well as the predicted reduction

$$m_k(0) - m_k(d^{(k)})$$

and consider the ratio

$$\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + d^{(k)})}{m_k(0) - m_k(d^{(k)})}$$

which measures the agreement between objective function f and model function m_k. We distinguish between:

1. $\rho_k \ll 1$ (or even < 0): the model is not a good approximation such that we reject the update (i.e. $x^{(k+1)} \leftarrow x^{(k)}$) and reduce the TR-radius ($\Delta_{k+1} \leftarrow c_1 \Delta_k,\ c_1 \in (0,1)$) [unsuccessful iteration].

2. $\rho_k \approx 1$ (> 1): the agreement is very accurate and we use the update $x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}$ and further increase the TR-radius $\Delta_{k+1} \leftarrow c_2 \Delta_k,\ c_2 > 1$ [very successful iteration].

3. $0 < \eta_1 < \rho_k < \eta_2 < 1$: the agreement is acceptable such that we use the update $x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}$ but do not alter the TR-radius ($\Delta_{k+1} \leftarrow \Delta_k$) [successful iteration].

Algorithm 2.7.1: TR-Newton method

Input: $x^{(0)} \in \mathbb{R}^n,\ 0 < \eta_1 < \eta_2 < 1,\ 0 < c_1 < 1 < c_2,\ \varepsilon \geq 0,\ \Delta_0 > 0,\ k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
  $d^{(k)} \leftarrow \underset{\|d\| \leq \Delta_k}{\operatorname{argmin}}\ m_k(d)$
  $\rho_k = \dfrac{f(x^{(k)}) - f(x^{(k)} + d^{(k)})}{m_k(0) - m_k(d^{(k)})}$
  if $\rho_k < \eta_1$ **then**
    $x^{(k+1)} \leftarrow x^{(k)}$
    $\Delta_{k+1} \leftarrow c_1 \Delta_k$
  end
  else if $\eta_1 < \rho_k < \eta_2$ **then**
    $x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}$
    $\Delta_{k+1} \leftarrow \Delta_k$
  end
  else
    $x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}$
    $\Delta_{k+1} \leftarrow c_2 \Delta_k$
  end
  $k \leftarrow k + 1$
end

Some typical parameter choices are for example

$$\eta_1 = 0.01,\ \eta_2 = 0.75,\ c_1 = 0.5,\ c_2 = 2,\ \Delta_0 = 1\ .$$

To show the practicability of Algorithm 2.7.1 it is left to show that $m_k(0) - m_k(d^{(k)}) \neq 0$, i.e. $m_k(0) - m_k(d^{(k)}) > 0$.

Lemma 2.7.2
It holds

$$m_k(0) - m_k(d^{(k)}) \geq \frac{1}{2}\|\nabla f(x^{(k)})\| \min\left\{\Delta_k, \frac{\|\nabla f(x^{(k)})\|}{\|\nabla^2 f(x^{(k)})\|}\right\} \geq 0\ .$$

Especially $m_k(0) - m_k(d^{(k)}) = 0$ *only if* $\nabla f(x^{(k)}) = 0$.

Proof.
Since $d^{(k)}$ is a global minimizer of m_k over the set $\{d : \|d\| \leq \Delta_k\}$ it holds for all d with $\|d\| \leq \Delta_k$ that

$$\begin{aligned} m_k(0) - m_k(d^{(k)}) &\geq m_k(0) - m_k(d) \\ &= f(x^{(k)}) - (f(x^{(k)}) + \nabla f(x^{(k)})^\top d + \frac{1}{2} d^\top \nabla^2 f(x^{(k)}) d) \\ &= -\nabla f(x^{(k)})^\top d - \frac{1}{2} d^\top \nabla^2 f(x^{(k)}) d \\ &\geq -\nabla f(x^{(k)})^\top d - \frac{1}{2}\|f(x^{(k)})\|\|d\|^2\ . \end{aligned}$$

1. First, we assume that $\|\nabla f(x^{(k)})\| \leq \Delta_k \|\nabla^2 f(x^{(k)})\|$. If $\|\nabla^2 f(x^{(k)})\| = 0$, this directly implies $\|\nabla f(x^{(k)})\| = 0$. If $\|\nabla^2 f(x^{(k)})\| \neq 0$ we set

   $$d := \frac{-1}{\|\nabla^2 f(x^{(k)})\|} \nabla f(x^{(k)})$$

   such that $\|d\| \leq \Delta_k$ and therefore

   $$\begin{aligned} m_k(0) - m_k(d^{(k)}) &\geq \frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla^2 f(x^{(k)})\|} - \frac{1}{2}\frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla^2 f(x^{(k)})\|^2}\|\nabla^2 f(x^{(k)})\| \\ &= \frac{1}{2}\frac{\|\nabla f(x^{(k)})\|}{\|\nabla^2 f(x^{(k)})\|}\|\nabla f(x^{(k)})\|\ . \end{aligned}$$

2. We now assume $\|\nabla f(x^{(k)})\| > \Delta_k \|\nabla^2 f(x^{(k)})\|$. Then we set

   $$d := -\frac{\Delta_k}{\|\nabla f(x^{(k)})\|} \nabla f(x^{(k)})$$

such that $\|d\| \leq \Delta_k$ and therefore

$$\begin{aligned} m_k(0) - m_k(d^{(k)}) &\geq \Delta_k \frac{\|\nabla f(x^{(k)})\|^2}{\|\nabla^2 f(x^{(k)})\|} - \frac{1}{2}\Delta_k^2 \|\nabla^2 f(x^{(k)})\| \\ &= \Delta_k(\|\nabla f(x^{(k)})\| - \frac{1}{2}\Delta_k \|\nabla^2 f(x^{(k)})\|) \\ &\geq \Delta_k(\|\nabla f(x^{(k)})\| - \frac{1}{2}\|\nabla^2 f(x^{(k)})\|) \\ &= \frac{1}{2}\|\nabla f(x^{(k)})\|\Delta_k \ . \end{aligned}$$

Taking the minimum over both inequalities proves the statement. □

This directly yields the practicability of the TR-Newton method and its convergence to a stationary point (provided one exists).

Theorem 2.7.3
Let $f \in C^2(\mathbb{R}^n)$, $(x^{(k)})_{k\in\mathbb{N}_0}$ the TR-Newton iterates, and let $L_0 := \{x : f(x) \leq f(x^{(0)})\}$ be compact. Then there exists a stationary point x^* of f such that $x^{(k)} \to x^*$, i.e. global convergence. Further, if $\nabla^2 f(x^*) > 0$, there exists a $K \in \mathbb{N}_0$ such that the TR-Newton iterates coincide with the Newton iterates for all $k \geq K$, i.e. the TR-Newton method converges locally superlinear/quadratic.

Proof. Cf. Nocedal and Wright (2006, Chapter 4). □

2.7.2 Solving the TR-subproblem

So far we have only stated that the TR-subproblem

$$\min_{\|d\|\leq\Delta_k} m_k(d) \quad \Leftrightarrow \quad \min_{\|d\|\leq\Delta_k} \Delta f(x^{(k)})^\top d + \frac{1}{2} d^\top \nabla^2 f(x^{(k)}) d \tag{TR}$$

has a solution but not how to find it. Obviously, an exact solution is ineffective such that we aim for an approximated solution.

The Cauchy-point

The simplest approach is to employ a steepest descent step, i.e.

$$d^{(k)} = -\lambda \nabla f(x^{(k)})$$

for some $\lambda \geq 0$. To determine λ we solve

$$\begin{aligned} \min \quad & m_k(\lambda \nabla f(x^{(k)})) \\ \text{s.t.} \quad & \lambda \geq 0 \\ & \lambda \|\nabla f(x^{(k)})\| \leq \Delta_k \end{aligned} \tag{2.1}$$

which is equivalently written as

$$\begin{aligned} \min \quad & -\lambda\|\nabla f(x^{(k)}))\|^2 + \frac{\lambda^2}{2}\nabla f(x^{(k)}))^\top \nabla^2 f(x^{(k)}))\nabla f(x^{(k)})) \\ \text{s.t.} \quad & \lambda \geq 0 \\ & \lambda\|\nabla f(x^{(k)})\| \leq \Delta_k \end{aligned} \tag{2.2}$$

such that

$$\lambda_k := \begin{cases} \dfrac{\Delta_k}{\|\nabla f(x^{(k)})\|}, & \text{if } \nabla f(x^{(k)})^\top \nabla^2 f(x^{(k)}) \nabla f(x^{(k)}) \leq 0 \\ \min\left\{ \dfrac{\Delta_k}{\|\nabla f(x^{(k)})\|}, \dfrac{\|\nabla f(x^{(k)})\|^2}{\nabla f(x^{(k)})^\top \nabla^2 f(x^{(k)}) \nabla f(x^{(k)})} \right\}, & \text{else} \end{cases} .$$

The point

$$x_c := x^{(k)} - \lambda_k \nabla f(x^{(k)})$$

is called *Cauchy-point*. We could take $x^{(k+1)} \leftarrow x_c$ which is easy to implement but corresponds to a (damped) steepest descent step, such that the convergence can become slow.

The Dogleg Method

To maintain the fast convergence of the Newton method (cf. Theorem 2.7.3) we need to take the Newton direction into account. The optimal (constrained) steepest descent direction reads

$$d_G = \frac{\|\nabla f(x^{(k)})\|^2}{\nabla f(x^{(k)})^\top \nabla^2 f(x^{(k)}) \nabla f(x^{(k)})} \nabla f(x^{(k)}) ,$$

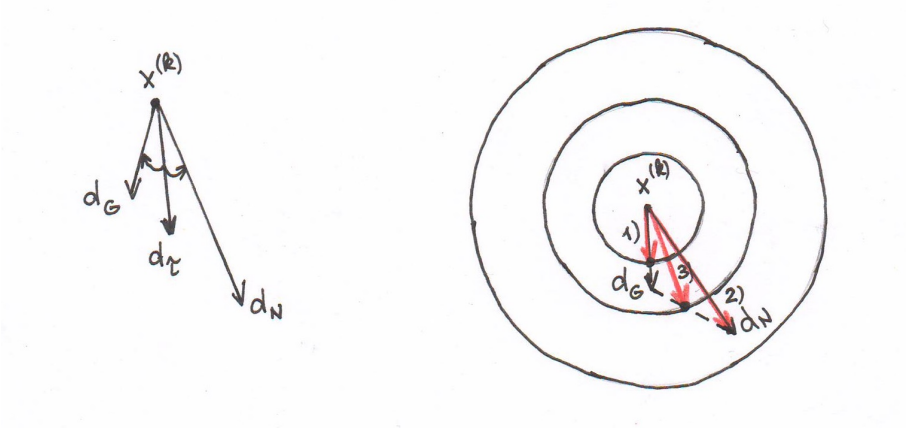
whereas the Newton direction is

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

Then we define $d^{(k)}$ as minimizer of m_k along the *Dogleg-path*

$$d(\tau) := \begin{cases} \tau d_G, & \tau \in [0,1] \\ d_G + (1-\tau)(d_N - d_G), & \in [1,2] \end{cases}$$

with the restriction $\|d(\tau)\| \leq \Delta_k$.



We distinguish the cases:

1. If $\|d_G\| \geq \Delta_k$, we take $x^{(k+1)} \leftarrow x_c \qquad (= x^{(k)} - \lambda_k \nabla f(x^{(k)}))$.
2. If $\|d_N\| \leq \Delta_k$, we take $x^{(k+1)} \leftarrow x^{(k)} + d_N$.
3. If $\|d_G\| \leq \Delta_k$ and $\|d_N\| > \Delta_k$, we take $x^{(k+1)} \leftarrow x^{(k)} + d(\tau^*)$, where τ^* is such that

$$\|d(\tau^*)\| = \Delta_k \qquad \Leftrightarrow \qquad \|d_G + (1-\tau^*)(d_G - d_N)\|^2 = \Delta_k^2 \ .$$

This can be solved by common root finding methods.

2.8 Nonlinear Least-Squares Problems

A problem of the form

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|r(x)\|^2 \qquad \Leftrightarrow \qquad \min_{x \in \mathbb{R}^n} \frac{1}{2}\sum_{j=1}^{m} r_j(x)^2 \ , \tag{NLS}$$

where $r : \mathbb{R}^n \to \mathbb{R}^m$ and $m \gg n$ is called a *nonlinear least-squares* problem. Those problems have wide applications in data fitting, parameter estimation, function approximation, and many others. Since (NLS) is an unconstrained problem with objective function

$$f : \mathbb{R}^n \to \mathbb{R}, \quad x \mapsto \frac{1}{2}\|r(x)\|^2$$

we can apply the already introduced unconstrained optimization methods to solve (NLS). However, it will be much more efficient to exploit the special structure of the objective function f.

Let

$$J(x) := Dr(x) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1}(x) & \dots & \frac{\partial r_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial x_1}(x) & \dots & \frac{\partial r_m}{\partial x_n}(x) \end{bmatrix} \in \mathbb{R}^{m \times n} \quad (\mathbb{R}^n \to \mathbb{R}^m)$$

denote the Jacobi matrix of r, such that

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x) \nabla r_j(x) = J(x)^\top r(x)$$

and

$$\begin{aligned} \nabla^2 f(x) &= \sum_{j=1}^{m} \nabla r_j(x) \nabla r_j(x)^\top + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \\ &= J(x)^\top J(x) + \underbrace{\sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x)}_{:= S(x)} \\ &= J(x)^\top J(x) + S(x) \ . \end{aligned}$$

2.8.1 Gauss-Newton Method

The Newton direction d_N for (NLS) is the solution of the linear system

$$(J(x)^\top J(x) + S(x))d_N \stackrel{!}{=} -J(x)^\top r(x).$$

The main disadvantage of this approach is the computation of the term $S(x)$, which is expensive since we need all Hessians $\nabla^2 r_j(x)$ for $j = 1, \dots, m$. The idea of the Gauss-Newton method is simply to exclude the expensive term $S(x)$ for the computation of the *Gauss-Newton direction* d_{GN}, i.e.

$$J(x)^\top J(x) d_{GN} \stackrel{!}{=} J(x)^\top r(x) \ .$$

This is for example reasonable if $r(x^*) \approx 0$ (small residual case) or if $\nabla^2 r_j(x^*) \approx 0$ (r is close to linear).

If $J(x)$ has full rank, then the matrix $J(x)^\top J(x)$ is symmetric and positive definite such that

$$d_{GN} = -(J(x)^\top J(x))^{-1} J(x)^\top r(x)$$

provides a descent direction of f in x, provided $\nabla f(x) \neq 0$.

Algorithm 2.8.1: Damped Gauss-Newton Method

Input: $x^{(0)} \in \mathbb{R}^n,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
- $d^{(k)} \leftarrow -(J(x^{(k)})^\top J(x^{(k)}))^{-1} J(x^{(k)})^\top r(x^{(k)})$; `// solve linear system`
- t_k that meets Wolfe condition
- $x^{(k+1)} \leftarrow x^{(k)} + t_k d^{(k)}$
- $k \leftarrow k + 1$

end

Theorem 2.8.2
Let $(x^{(k)})_{k\in\mathbb{N}_0}$ denote the Gauss-Newton-iterates, let L_0 be compact, and let $J(x)$ be of full rank for all $x \in L_0$. Then there exists at least one accumulation point of the sequence $(x^{(k)})_{k\in\mathbb{N}_0}$ and each accumulation point is a stationary point of f.

Proof.
Due to Theorem 2.2.10 it is left to show that $d^{(k)}$ meets the angle condition, i.e.

$$\frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|\nabla f(x^{(k)})\| \|d^{(k)}\|} > 0 \quad \forall k \in \mathbb{N}_0 \ .$$

For fixed $k \in \mathbb{N}_0$ there exists, since $J(x^{(k)})^\top J(x^{(k)}) > 0$, constants $c_1,\ c_2 > 0$ with

$$\begin{aligned}
-\nabla f(x^{(k)})^\top d^{(k)} &= \nabla f(x^{(k)})^\top (J(x^{(k)})^\top J(x^{(k)}))^{-1} \nabla f(x^{(k)}) \\
&\geq c_1 \|\nabla f(x^{(k)})\|^2 \\
&= c_1 \|J(x^{(k)})^\top J(x^{(k)}) (J(x^{(k)})^\top J(x^{(k)}))^{-1} \nabla f(x^{(k)})\| \|\nabla f(x^{(k)})\| \\
&= c_1 \| - J(x^{(k)})^\top J(x^{(k)}) d^{(k)}\| \|\nabla f(x^{(k)})\| \\
&\geq c_1 c_2 \|d^{(k)}\| \|\nabla f(x^{(k)})\|
\end{aligned}$$

such that

$$\frac{-\nabla f(x^{(k)})^\top d^{(k)}}{\|\nabla f(x^{(k)})\|\|d^{(k)}\|} \geq c_1 c_2 > 0 \ .$$

□

The rate of convergence strongly depends on the influence of the excluded term $S(x)$:

1. If $S(x^*) = 0$, then $d_{GN} = d_N$ and we obtain locally superlinear/quadratic and global convergence.
2. If $S(x^*) \ll J(x^*)^\top J(x^*)$ then we obtain linear convergence, but the convergence rate is very fast.
3. Otherwise, the Gauss-Newton method can converge arbitrary slow.

2.8.2 Levenberg-Marquardt Method

If $J(x)^\top J(x)$ is of ill-condition or even only positive semidefinite, i.e. if $J(x)$ is not of full rank, the idea is to add a regularization term such that

$$d_{LM} = -(J(x)^\top J(x) + \lambda I)^{-1} J(x)^\top r(x)$$

for some $\lambda > 0$. It holds

$$d_{LM}(\lambda) \overset{\lambda \to 0}{\longrightarrow} d_{GN}, \qquad d_{LM}(\lambda) \overset{\lambda \to \infty}{\longrightarrow} d_{SD}$$

and since $J(x)^\top J(x) + \lambda I$ is symmetric and positive definite for some $\lambda > 0$, d_{LM} is a descent direction and we obtain global convergence. For the practical choice of λ a simple updating rule is applied.

Algorithm 2.8.3: Levenberg-Marquardt method

Input: $x^{(0)} \in \mathbb{R}^n,\ \varepsilon \geq 0,\ \lambda_0 > 0,\ k \leftarrow 0$
while $\|\nabla f(x^{(k)})\| > \varepsilon$ **do**
  $d^{(k)} \leftarrow -(J(x^{(k)})^\top J(x^{(k)}) + \lambda_k I)^{-1} J(x^{(k)})^\top r(x^{(k)})$
  if $f(x^{(k)} + d^{(k)}) \geq f(x^{(k)})$ **then**
    $x^{(k+1)} \leftarrow x^{(k)}$
    $\lambda_{k+1} \leftarrow 2\lambda_k$
  end
  else
    $x^{(k+1)} \leftarrow x^{(k)} + d^{(k)}$
    $\lambda_{k+1} \leftarrow \frac{1}{2}\lambda_k$
  end
  $k \leftarrow k + 1$
end

As for the Gauss-Newton method, the convergence strongly depends on the neglected term $S(x)$. However, the convergence rates are quite similar to the Gauss-Newton method.

Chapter 3

(Nonlinear) Constrained Optimization

In this chapter we consider the minimization of an objective function subject to constraints on the variables, that is (cf. Section 1.3)

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \qquad i = 1, ..., m \\ & h_j(x) = 0, \qquad j = 1, ..., p \end{aligned} \tag{PC}$$

where the objective function f, the inequality constraints g_i, and the equality constraints h_j are in $C^1(\mathbb{R}^n, \mathbb{R})$. For more convenience we define the vector valued functions

$$g(x) := \begin{pmatrix} g_1(x) \\ \vdots \\ g_m(x) \end{pmatrix}, \qquad h(x) := \begin{pmatrix} h_1(x) \\ \vdots \\ h_p(x) \end{pmatrix}$$

such that (PC) reads

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0, \\ & h(x) = 0. \end{aligned}$$

We define the *feasible set* of (PC) as

$$X := \{x \in \mathbb{R}^n : g(x) \leq 0, h(x) = 0\}$$

and can rewrite (PC) in the more compact form

$$\min_{x\in X} f(x) \ .$$

Obviously, if $X = \mathbb{R}^n$, this is an unconstrained problem such that we assume $X \subsetneq \mathbb{R}^n$ and $X \neq \emptyset$. Further, we assume that at least one of the occurring functions is nonlinear.

3.1 Optimality Conditions

In Section 2.1 we have shown that the condition

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

is a characteristic of x^* being a solution of (PU). For (PC) this holds no longer true,as the example

$$\min_{x \leq -1} x^2$$

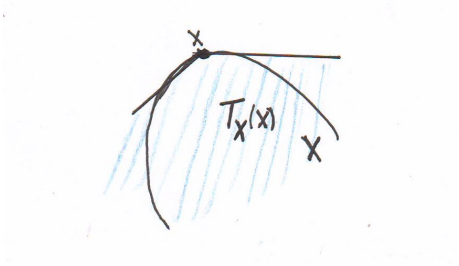
shows. Optimality conditions for (PC) require much more effort and are frequently characterized by means of the *tangent cone*:

Definition 3.1.1
For $x \in X$ we call the set

$$T_X(x) := \left\{ d \in \mathbb{R}^n : \exists (x^{(k)})_{k \in \mathbb{N}} \subseteq X, \exists t_k \searrow 0 : x^{(k)} \to x, \frac{x^{(k)} - x}{t_k} \to d \right\}$$

the tangent cone *of X at x.*

Geometrically, the tangent cone describes the set of all feasible directions of X at x, i.e. $d \in T_X(x)$ if and only if $\exists \bar{t} > 0$ such that $x + td \in X \ \forall t \in [0, \bar{t}[$.



For the tangent cone it holds

1. $x \in int(X) \Rightarrow T_X(x) = \mathbb{R}$,
2. $0 \in T_X(x)$ and thus $T_X(x) \neq \emptyset$,
3. $d \in T_X(x),\ \alpha > 0 \Rightarrow \alpha d \in T_X(x)$,
4. $T_X(x)$ is a closed set.

In analogy to Lemma 2.1.1 we obtain a very basic necessary characterization of optimality.

Lemma 3.1.2
Let $f \in C^1(X)$ and let x^* be a solution of (PC). Then it holds

$$\nabla f(x^*)^\top d \geq 0 \qquad \forall d \in T_X(x^*) \ .$$

Proof.
Let $d \in T_X(x^*)$ be arbitrary, such that there exists $t_k \searrow 0$ and $x^{(k)} \to x^*$ with

$$\lim_{k\to\infty} \frac{x^{(k)} - x^*}{t_k} = d.$$

Since f is continuous, Theorem 1.1.1 yields

$$f(x^{(k)}) - f(x^*) = \nabla f(z^{(k)})^\top (x^{(k)} - x^*)$$

for some $z^{(k)} \in [x^{(k)}, x^*]$. Especially, $z^{(k)} \to x^*$ and, since ∇f is continuous, it also holds $\nabla f(z^{(k)}) \to \nabla f(x^*)$. For k sufficiently large, we have

$$\nabla f(z^{(k)})^\top (x^{(k)} - x^*) = f(x^{(k)}) - f(x^*) \geq 0$$

since x^* is a solution of (PC). This finally yields

$$0 \leq \frac{\nabla f(z^{(k)})(x^{(k)} - x^*)}{t_k} \to \nabla f(x^*)^\top d.$$

□

In analogy to the unconstrained case, a point $x^* \in X$ with

$$\nabla f(x^*)^\top d \geq 0 \quad \forall d \in T_X(x^*)$$

is called a *stationary point* of (PC).

Lemma 3.1.2 says that, if x^* is a solution of (PC), there is no descent direction (of f at x) in $T_X(x^*)$. If there is a descent direction $\widetilde{d} \notin T_X(x^*)$, then $\widetilde{d}$ is not a feasible direction of X at x such that

$$x^* + t\widetilde{d} \notin X \quad \forall t > 0.$$

Since the tangent cone can become very complicated, Lemma 3.1.2 is rather of theoretical nature. In the following we derive more practical conditions.

Definition 3.1.3
Let $x \in X$ be feasible for (PC). We call

$$A_X(x) := \{i \in \{1, ..., m\} : g_i(x) = 0\}$$

the set of active inequalities (active set) *at x and*

$$L_X(x) := \{d \in \mathbb{R}^n : \nabla g_i(x)^\top d \leq 0 \ \forall i \in A_X(x), \nabla h_j(x)^\top d = 0 \ \forall j = 1, ..., p\}$$

the linearized cone *of X at x.*

Definition 3.1.4
The function

$$\begin{aligned} L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}, \quad L(x, \lambda, \mu) &= f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \mu_j h_j(x) \\ &= f(x) + \lambda^\top g(x) + \mu^\top h(x) \end{aligned}$$

is called Lagrangian (function) *of* (PC).

Definition 3.1.5
The conditions

1. $\nabla_x L(x, \lambda, \mu) = \nabla f(x) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x) + \sum_{j=1}^{p} \mu_j \nabla h_j(x) = 0$
2. $h(x) = 0$
3. $g(x) \leq 0$
4. $\lambda \geq 0$
5. $\lambda^\top g(x) = 0 \qquad (\Leftrightarrow \lambda_i g_i(x) \text{ for all } i = 1, ..., m)$

are called Karush-Kuhn-Tucher (KKT) *conditions of* (PC). *Each triple* (x^*, λ^*, μ^*) *that satisfies the KKT-conditions is called* KKT-point. *The vectors* λ^*, μ^* *are called* Lagrangian multipliers *(or dual variables).*

The KKT-conditions, under appropriate regularity conditions, play the role of the NFOC.

Lemma 3.1.6 (Farkas Lemma)
Let $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^n$. *The following statements are equivalent:*

1. $\exists\ y \geq 0 : A^\top y = b,$
2. $b \top d \geq 0 :\ \forall d$ *with* $Ad \geq 0.$

Theorem 3.1.7 (NFOC under ACQ)
Let $x^* \in X$ *be a solution of* (PC) *and let*

$$T_X(x^*) = L_X(x^*).$$

Then there exists Lagrangian multipliers $\lambda^* \in \mathbb{R}^m, \mu^* \in \mathbb{R}^p$ *such that* (x^*, λ^*, μ^*) *is a KKT-point of* (PC).

Proof.
Since $x^* \in X$ it holds $h(x^*) = 0$ and $g(x^*) \leq 0$ and by Lemma 3.1.2 it holds

$$\nabla f(x^*)^\top d \geq 0 \quad \forall d \in T_X(x^*) = L_X(x^*)$$

and therefore

$$\nabla f(x^*)^\top d \geq 0 \quad \forall d : Ad \leq 0\ ,$$

where

$$A = \begin{bmatrix} \nabla g_i(x^*)^\top & i \in A_X(x^*) \\ \nabla h_j(x^*)^\top & j = 1, ..., p \\ -\nabla h_j(x^*)^\top & j = 1, ..., p \end{bmatrix}\ .$$

This is equivalent to

$$-\nabla f(x^*)^\top \geq 0 \quad \forall d : Ad \geq 0$$

and Lemma 3.1.6 yields the existence of a vector $y \geq 0$ such that

$$A^\top y = -\nabla f(x^*).$$

Let the first $\#A_X(x^*)$ components of y be denoted as $\lambda_i^*,\ i \in A_X(x^*)$, the next p components as μ_j^+ and the last components as μ_j^-. Further, for $i \in \{1,...,n\} \setminus A_X(x^*)$ we set $\lambda_i^* = 0$ and $\mu_j^* = \mu_j^+ - \mu_j^-,\ j = 1,...,p$. Then it holds $\lambda^* \geq 0, (\lambda^*)^\top g(x^*) = 0$, and

$$\begin{aligned} -\nabla f(x^*) = A^\top y &= \sum_{i \in A_X(x^*)} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^+ \nabla h_j(x^*) + \sum_{j=1}^{p} \mu_j^- (-\nabla h_j(x^*)) \\ &= \sum_{i \in A_X(x^*)} \lambda_i^* \nabla g_i(x^*) + \sum_{i \in \{1,...,m\} \setminus A_X(x^*)} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*), \end{aligned}$$

i.e. $\nabla_X L(x^*, \lambda^*, \mu^*) = 0$. □

The assumption $T_X(x^*) = L_X(x^*)$ is of major importance for the proof of Theorem 3.1.7 and is referred to as *Abadie constraint qualification (ACQ)*. It always holds $T_X(x^*) \subseteq L_X(x^*)$, such that $ACQ \Leftrightarrow L_X(x) \subseteq T_X(x)$. This is easy to proof for special types of problems, like linear or convex problems. For more general problems, however, the ACQ might be difficult to verify (if even possible).

Definition 3.1.8
Let $x \in X$ be a feasible point of (PC). If

$$\{\nabla g_i(x), \nabla h_j(x) : i \in A_X(x),\ j = 1,...,p\}$$

are linearly independent, we say that x satisfies the linear independence constraint qualification *(LICQ).*

Theorem 3.1.9
If $x \in X$ satisfies the LICQ, then it also satisfies the ACQ.

Proof. Cf. Alt (2013, Korollar 7.2.30). □

Theorem 3.1.10 (NFOC under LICQ)
Let $x^* \in X$ be a local solution of (PC) that meets the LICQ. Then there exist unique Lagrangian multipliers λ^*, μ^* such that (x^*, λ^*, μ^*) is a KKT-point.

Proof.
Since by Theorem 3.1.9 the LICQ implies the ACQ, Theorem 3.1.7 yields the existence of Lagrangian multipliers λ^*, μ^* such that (x^*, λ^*, μ^*) is a KKT-point. Since $\lambda_i^* = 0$ for all $i \in A_X(x^*)$ by the KKT-conditions, it follows from $\nabla_X L(x^*, \lambda^*, \mu^*) = 0$ that

$$-\nabla f(x^*) = \sum_{i \in A_X(x^*)} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*)$$

and by the LICQ we have that the Lagrangian multipliers are unique. □

In order to obtain necessary and sufficient second order conditions we partition the active set of a given KKT-point (x^*, λ^*, μ^*) as

$$A_X(x^*) = A_0(x^*, \lambda^*) \cup A_>(x^*, \lambda^*),$$

where

$$\begin{aligned} A_0(x^*, \lambda^*) &= \{i \in A_X(x^*) : \lambda^* = 0\}, \\ A_>(x^*, \lambda^*) &= \{i \in A_X(x^*) : \lambda^* > 0\} \end{aligned}$$

and define the *critical cone*

$$C_X(x^*, \lambda^*) = \left\{ d \in \mathbb{R}^n : \begin{array}{ll} \nabla g_i(x^*)^\top d = 0 & \forall i \in A_>(x^*, \lambda^*) \\ \nabla g_i(x^*)^\top d \leq 0 & \forall i \in A_0(x^*, \lambda^*) \\ \nabla h_j(x^*)^\top d = 0 & \forall j = 1, ..., p. \end{array} \right. .$$

This cone is frequently used to define second order optimality conditions.

Theorem 3.1.11 (NSOC)
Let x^* be a local solution of (PC) that meets the LICQ and let λ^*, μ^* denote the unique Lagrangian multipliers associated with x^*. Then it holds

$$\nabla^2_{xx} L(x^*, \lambda^*, \mu^*) \geq 0 \text{ on } C_X(x^*, \lambda^*).$$

Proof. Cf. Geiger and Kanzow (2013, Theorem 2.54). □

Theorem 3.1.12 (SSOC)
Let (x^*, λ^*, μ^*) be a KKT-point of (PC) and let $\nabla_x L(x^*, \lambda^*, \mu^*) > 0$ on $C_X(x^*, \lambda^*) \setminus \{0\}$. The x^* is a strict (local) solution of (PC).

Proof.
Suppose that x^* is not a strict local solution of (PC). Then we can find a sequence $(x^{(k)})_{k \in \mathbb{N}}$ in X such that $x^{(k)} \to x^*$, $x^{(k)} \neq x^*$ for all $k \in \mathbb{N}$, and $f(x^{(k)}) \leq f(x^*)$. Since

$$\left\| \frac{x^{(k)} - x^*}{\|x^{(k)} - x^*\|} \right\| = 1$$

we assume without loss of generality that

$$\frac{x^{(k)} - x^*}{\|x^{(k)} - x^*\|} \to d^* \in \mathbb{R}^n.$$

Since h_j in continuously differentiable for all $j = 1, ..., p$, Theorem 1.1.1 yields the existence of $z^{(k)} \in [x^{(k)}, x^*]$ such that

$$\begin{aligned} & h_j(x^{(k)}) = h_j(x^*) + \nabla h_j(z^{(k)})^\top (x^{(k)} - x^*) \\ \Leftrightarrow\ & 0 = \nabla h_j(z^{(k)})^\top (x^{(k)} - x^*) \\ \Leftrightarrow\ & 0 = \nabla h_j(z^{(k)})^\top \frac{x^{(k)} - x^*}{\|x^{(k)} - x^*\|} \\ \overset{k \to \infty}{\Rightarrow}\ & 0 = \nabla h_j(x^{(*)})^\top d^* \qquad \forall j = 1, ..., p. \end{aligned}$$

In analogy, since $g_i(x^*) = 0$ and $g_i(x^{(k)}) \leq 0$ for all $i \in A_X(x^*)$ we can show

$$\nabla g_i(x^*)^\top d^* \leq 0 \quad \forall i \in A_X(x^*)$$

and, since $f(x^{(k)}) \leq f(x^*)$, we can show

$$\nabla f(x^*)^\top d^* \leq 0.$$

We now distinguish the cases:

1. $d^* \in C_X(x^*, \lambda^*)$,
2. $d^* \notin C_X(x^*, \lambda^*)$.

1. Let $d^* \in C_X(x^*, \lambda^*)$. Since $x^{(k)} \in X$ it holds

$$f(x^*) \geq f(x^{(k)}) \geq f(x^{(k)}) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} \underbrace{g_i(x^*)}_{\leq 0} + \sum_{j=1}^{p} \mu_j^* \underbrace{h_j(x^*)}_{=0} = L(x^{(k)}, \lambda^*, \mu^*).$$

By Theorem 1.1.1 there exists $z^{(k)} \in [x^{(k)}, x^*]$:

$$\begin{aligned} f(x^*) &\geq L(x^{(k)}, \lambda^*, \mu^*) \\ &= \underbrace{L(x^*, \lambda^*, \mu^*)}_{=f(x^*)} + \underbrace{\nabla_X L(x^*, \lambda^*, \mu^*)}_{=0}(x^{(k)} - x^*) \\ &\quad + \frac{1}{2}(x^{(k)} - x^*)^\top \nabla_{xx}^2 L(z^{(k)}, \lambda^*, \mu^*)(x^{(k)} - x^*) \\ &= f(x^*) + \frac{1}{2}(x^{(k)} - x^*)^\top \nabla_{xx}^2 L(z^{(k)}, \lambda^*, \mu^*)(x^{(k)} - x^*) \ . \end{aligned}$$

This yields

$$\begin{aligned} &\frac{1}{\|x^{(k)} - x^*\|}\left[(x^{(k)} - x^*)^\top \nabla_{xx}^2 L(z^{(k)}, \lambda^*, \mu^*)(x^{(k)} - x^*)\right] \leq 0 \\ \overset{k\to\infty}{\Rightarrow}\ & (d^*)^\top \nabla_{xx}^2 L(z^{(k)}, \lambda^*, \mu^*) d^* \leq 0 \ , \end{aligned}$$

which is a contradiction to $\nabla_{xx}^2 L(z^{(k)}, \lambda^*, \mu^*) > 0$ on $C_X(x^*, \lambda^*) \setminus \{0\}$.

2. Let $d^* \notin C_X(x^*, \lambda^*)$. Then there exists $i_> \in A_>(x^*, \lambda^*)$ such that $\nabla g_{i>}(x^*)^\top d^* < 0$. It holds

$$\begin{aligned} 0 \geq \nabla f(x^*)^\top d^* &= -\sum_{i=1}^{m} \underbrace{\lambda_i^*}_{\geq 0} \underbrace{\nabla g_i(x^*)^\top d^*}_{\leq 0} - \sum_{j=1}^{p} \mu_j^\top \underbrace{\nabla h_j(x^*)^\top}_{=0} d^* \\ &\geq -\underbrace{\lambda_{i>}^*}_{>0} \underbrace{\nabla g_{i>}(x^*)^\top}_{<0} d^* > 0 \ , \end{aligned}$$

which is obviously a contradiction.

□

3.2 Penalty- and Barrier Methods

The most classical approach to solve a constrained optimization problem (PC) is to replace the problem by a sequence of unconstrained optimization problems (PU) where the new objective function penalizes violations of the constraints of (PC) with increasing severity. For that purpose, a *penalization* or *penalty function*

$$\pi : \mathbb{R}^n \to [0, \infty[, \qquad \pi(x) \begin{cases} = 0, & \text{if } x \in X \\ > 0, & \text{if } x \notin X \end{cases}$$

is added to the objective function f, weighted by a penalty parameter $\alpha > 0$. That is, we consider the unconstrained problem

$$\min_{x \in \mathbb{R}^n} \; f(x) + \alpha \pi(x)$$

and hope that the sequence of solutions $x^*(\alpha)$ converges to x^* for $\alpha \to \infty$, where x^* is a solution of (PC) and $x^*(\alpha)$ is a solution of the penalized problem with given α. In this context, there are basically two approaches, namely

1. *penalty methods*: allow $x^*(\alpha) \notin X$, such that the solution x^* is approximated from the outside of X.
2. *barrier methods*: claim $x^*(\alpha) \in X$ for all $\alpha > 0$ such that the solution x^* is approximated from the interior of X (interior-point methods, feasible-point methods).

3.2.1 Quadratic Penalty Method

To penalize the violation of constraints the quadratic penalty function is widely used. That is, we set

$$\begin{aligned} \pi_2(x) &:= \frac{1}{2} \sum_{i=1}^{m} \max\{0, g_i(x)\}^2 + \frac{1}{2} \sum_{j=1}^{p} h_j(x)^2 \\ &= \frac{1}{2} \Big(\sum_{i=1}^{m} (g_i(x)^+)^2 + \sum_{j=1}^{p} (h_j(x))^2 \Big) \\ &= \frac{1}{2} \|g(x)^+\|^2 + \frac{1}{2} \|h(x)\|^2 \end{aligned}$$

and instead of (PC) we focus on the unconstrained problem

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \; f(x) + \alpha \pi_2(x) \\ \Leftrightarrow &\min_{x \in \mathbb{R}^n} \; P_2(x; \alpha) \; , \end{aligned}$$

where $P_2(x; \alpha) := f(x) + \alpha \pi_2(x)$. This can be achieved by means of the previously presented methods for unconstrained optimization.

Algorithm 3.2.1: Quadratic penalty method

Input: $x^{(0)} \in \mathbb{R}^n,\ \alpha_0 = 0,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|\pi_2(x^{(k)}\| > \varepsilon$ **do**

  $x^{(k+1)} \leftarrow \operatorname{argmin}_{x \in \mathbb{R}^n} P_2(x, \alpha_k)$; `// warm start with` $x^{(k)}$

  $\alpha_{k+1} \leftarrow \max\{10\alpha_k, 10\}$

  $k \leftarrow k + 1$

end

Lemma 3.2.2
Let $f \in C^1(\mathbb{R}^n)$ and let $(x^{(k)})_{k \in \mathbb{N}_0}$ denote the iterates of the quadratic penalty method. Then it holds

1. *$(P_2(x^{(k)}, \alpha_{k-1}))_{k \in \mathbb{N}_0}$ is monotonically increasing,*
2. *$(\pi_2(x^{(k)}))_{k \in \mathbb{N}_0}$ is monotonically decreasing,*
3. *$(f(x^{(k)}))_{k \in \mathbb{N}_0}$ is monotonically increasing,*
4. *$g_i(x^{(k)}) \to 0$ for all $i = 1, ..., m$ and $h_j(x^{(k)}) \to 0$ for all $j = 1, ..., p$, i.e. $\pi_2(x^{(k)}) \to 0$.*

Proof.

1. Since $x^{(k)}$ is a minimizer of $P_2(\cdot, \alpha_{k-1})$ and since $\alpha_{k-1} < \alpha_k$ it holds

$$\begin{aligned} P_2(x^{(k)}, \alpha_{k-1}) &\leq P_2(x^{(k+1)}, \alpha_{k-1}) = f(x^{(k+1)}) + \alpha_{k-1}\pi_2(x^{(k+1)}) \\ &\leq f(x^{(k+1)}) + \alpha_k \pi_2(x^{(k+1)}) \\ &= P_2(x^{(k+1)}, \alpha_k). \end{aligned}$$

2. Since $P_2(x^{(k)}, \alpha_{k-1}) \leq P_2(x^{(k+1)}, \alpha_{k-1})$ and $P_2(x^{(k+1)}, \alpha_k) \leq P_2(x^{(k)}, \alpha_k)$ we have

$$\begin{aligned} & P_2(x^{(k)}, \alpha_{k-1}) + P_2(x^{(k+1)}, \alpha_k) \leq P_2(x^{(k+1)}, \alpha_{k-1}) + P_2(x^{(k)}, \alpha_k) \\ \Leftrightarrow\ & f(x^{(k)}) + \alpha_{k-1}\pi_2(x^{(k)}) + f(x^{(k+1)}) + \alpha_k \pi_2(x^{(k+1)}) \\ & \qquad \leq f(x^{(k+1)}) + \alpha_{k-1}\pi_2(x^{(k+1)}) + f(x^{(k)}) + \alpha_k \pi_2(x^{(k)}) \\ \Leftrightarrow\ & \alpha_{k-1}\pi_2(x^{(k)}) - \alpha_k \pi_2(x^{(k)}) + \alpha_k \pi_2(x^{(k+1)}) - \alpha_{k-1}\pi_2(x^{(k+1)}) \leq 0 \\ \Leftrightarrow\ & \underbrace{(\alpha_{k-1} - \alpha_k)}_{<0}(\pi_2(x^{(k)}) - \pi_2(x^{(k+1)}) \leq 0 \\ \Leftrightarrow\ & \pi_2(x^{(k)}) \geq \pi_2(x^{(k+1)}). \end{aligned}$$

3. It holds

$$\begin{aligned} & P_2(x^{(k)}, \alpha_{k-1}) \leq P_2(x^{(k+1)}, \alpha_{k-1}) \\ \Leftrightarrow\ & f(x^{(k)}) + \alpha_{k-1}\pi_2(x^{(k)}) \leq f(x^{(k+1)}) + \alpha_{k-1}\pi_2(x^{(k+1)}) \overset{2)}{\leq} f(x^{(k+1)}) + \alpha_{k-1}\pi_2(x^{(k)}) \\ \Leftrightarrow\ & f(x^{(k)}) \geq f(x^{(k+1)}). \end{aligned}$$

4. Let $\bar{x} \in X$ be arbitrary. Then it holds

$$f(\bar{x}) = \underbrace{f(\bar{x}) + \alpha_{k-1} \underbrace{\pi_2(\bar{x})}_{=0}}_{=P_2(\bar{x},\alpha_{k-1})} \geq \underbrace{f(x^{(k)}) + \alpha_{k-1}\pi_2(x^{(k)})}_{=P_2(x^{(k)},\alpha_{k-1})} \overset{3)}{\geq} f(x^{(0)}) + \alpha_{k-1}\pi_2(x^{(k)})$$

$$\Leftrightarrow \underbrace{\pi_2(x^{(k)})}_{\geq 0} \leq \frac{1}{\alpha_{k-1}}(f(\bar{x} - f(x^{(0)}) \to 0$$

$$\Rightarrow \pi_2(x^{(k)}) \to 0.$$

□

Theorem 3.2.3
Let $(x^{(k)})_{k\in\mathbb{N}_0}$ be the sequence of the quadratic penalty method. Then, each accumulation point of the sequence is a global solution of (PC).

Proof.
Let x^* denote an accumulation point and without loss of generality we assume $x^{(k)} \to x^*$. By Lemma 3.2.2 it holds $\pi_2(x^{(k)}) \to 0$ such that $\pi(x^*) = 0$, i.e. $x^* \in X$. Further,

$$P_2(x^{(k)}) \leq \inf_{x\in X} P_2(x, \alpha_{k-1}) = \inf_{x\in X} f(x) =: f^*.$$

This yields

$$\begin{aligned} f(x^*) &= \lim_{k\to\infty} f(x^{(k)}) \leq \lim_{k\to\infty} f(x^{(k)}) + \alpha_{k-1}\pi_2(x^{(k)}) \\ &= \lim_{k\to\infty} P_2(x^{(k)}, \alpha_{k-1}) \leq f^*, \end{aligned}$$

such that x^* is global solution of (PC). □

To minimize $P_2(\cdot, \alpha)$ we want to use methods of unconstrained optimization. Therefore we require that $P_2(\cdot, \alpha)$ is at least continuously differentiable, which is only problematic for the inequality constraints. One can show (exercise) that

$$\frac{\partial}{\partial t}\big(\frac{1}{2}\max\{0, t\}\big) = \max\{0, t\} = t^+ \ ,$$

such that

$$\nabla P_2(\cdot, \alpha) = \nabla f(x) + \alpha \sum_{i=1}^{m} g_i(x)^+ \nabla g_i(x) + \alpha \sum_{j=1}^{p} h_j(x)\nabla h_j(x).$$

Comparison to the Lagrangian yields

$$\nabla P_2(x, \alpha) = \nabla_x L(x, \lambda, \mu) \ ,$$

where

$$\lambda_i = \alpha g_i(x)^+ \quad \text{and} \quad \mu_j = \alpha h_j(x) \ .$$

Indeed, this observation can be used to estimate Lagrangian multipliers.

Theorem 3.2.4
Let $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^m \to \mathbb{R}$, and $h : \mathbb{R}^p \to \mathbb{R}$ be continuous differentiable and let $(x^{(k)})_{k\in\mathbb{N}}$ be generated by Algorithm 3.2.1. Let $x^{(k)} \to x^*$ and let x^* meet the LICQ. Then it holds

$$\lambda^{(k)} := \alpha_{k-1} g(x^{(k)})^+ \to \lambda^*$$
$$\mu^{(k)} := \alpha_{k-1} h(x^{(k)})^+ \to \mu^*$$

and (x^*, λ^*, μ^*) is a KKT-point of (PC).

Proof.
We first prove the convergence of the sequences $(\lambda^{(k)})_{k\in\mathbb{N}}$ and $(\mu^{(k)})_{k\in\mathbb{N}}$. For the inactive inequalities $i \notin A_X(x^*)$ it holds $g_i(x^*) < 0$ such that for k sufficiently large we have $g_i(x^{(k)}) < 0$ and hence $g_i(x^{(k)})^+ = 0$ such that $\lambda_i^{(k)} = \alpha_{k-1} g_i(x^{(k)})^+ = 0$ for all $i \in A_X(x^*)$. For the remaining components we use the LICQ. We define A_k as the matrix consisting of the columns $\nabla g_i(x^{(k)})$, $i \in A_X(x^*)$ and $\nabla h_j(x^{(k)})$, $j = 1, ..., p$, such that

$$A_k \to A_*.$$

Due to the LICQ, A_* is of full rank such that $A_*^\top A_*$ is nonsingular. For k sufficiently large, the Banach-Lemma (Lemma 2.5.3) with $A := A_*^\top A_*$ and $B := A_k^\top A_k - A_*^\top A_*$ yields that $A_k^\top A_k$ is nonsingular as well. Further,

$$(A_k^\top A_k)^{-1} \to (A_*^\top A_*)^{-1} .$$

Since $x^{(k)}$ is a minimizer of $P_2(\cdot, \alpha_{k-1})$ we have

$$\begin{aligned}
0 = \nabla P_2(x^{(k)}, \alpha_{k-1}) &= \nabla f(x^{(k)}) + \alpha_{k-1} \sum_{i=1}^{m} g_i(x^{(k)}) \nabla g_i(x^{(k)}) + \alpha_{k-1} \sum_{j=1}^{p} h_j(x^{(k)}) \nabla h_j(x^{(k)}) \\
&= \nabla f(x^{(k)}) + \sum_{i\in A_X(x^*)} g_i(x^{(k)}) \nabla g_i(x^{(k)}) + \sum_{j=1}^{p} \mu_j^{(k)} \nabla h_j(x^{(k)}) \\
&= \nabla f(x^{(k)}) + \sum_{i\in A_X(x^*)} \lambda_i^{(k)} \nabla g_i(x^{(k)}) + \sum_{j=1}^{p} \mu_j^{(k)} \nabla h_j(x^{(k)}) \\
&= \nabla f(x^{(k)}) + A_k \begin{pmatrix} \tilde{\lambda}^{(k)} \\ \mu^{(k)} \end{pmatrix} ,
\end{aligned}$$

where $\mu^{(k)} = (\mu_1^{(k)}, ..., \mu_p^{(k)})^\top$ and $\tilde{\lambda}^{(k)} = (\lambda_i^{(k)},\ i \in A_X(x^*))^\top$ such that

$$\begin{aligned}
0 = A_k^\top \nabla P_2(x^{(k)}, \alpha_{k-1}) = A_k^\top \nabla f(x^{(k)}) + A_k^\top A_k \begin{pmatrix} \tilde{\lambda}^{(k)} \\ \mu^{(k)} \end{pmatrix} \\
\Leftrightarrow \begin{pmatrix} \tilde{\lambda}^{(k)} \\ \mu^{(k)} \end{pmatrix} = (A_k \top A_k)^{-1} (-A_k^\top \nabla f(x^{(k)})) \to -(A_*^\top A_*)^{-1} A_*^\top \nabla f(x^*) =: \begin{pmatrix} \tilde{\lambda}^* \\ \mu^* \end{pmatrix},
\end{aligned}$$

which finally yields the convergence of the sequences.

We now check for the KKT-conditions. Due to the definition of $\lambda^{(k)}$ and $\mu^{(k)}$ we have

$$\nabla_x L(x^*, \lambda^*, \mu^*) = \lim_{k\to\infty} \nabla_x L(x^{(k)}, \lambda^{(k)}, \mu^{(k)}) = \lim_{k\to\infty} \nabla P_2(x^{(k)}, \alpha_{k-1}) = 0.$$

By Theorem 3.2.3 we know that $x^* \in X$ such that $g(x^*) \leq 0$ and $h(x^*) = 0$. Further it holds

$$\lambda_i^* = \lim_{k \to \infty} \alpha_{k-1} g_i(x^{(k)})^+ \geq 0 \quad \forall i = 1, ..., m$$

and

$$\left.\begin{array}{ll} \lambda_i^* g_i(x^*)^+ = 0 & \forall i \in A_X(x^*) \\ \lambda_i^* g_i(x^*)^+ = 0 & \forall i \notin A_X(x^*) \end{array}\right\} \quad \text{i.e. } \forall i = 1, ..., m.$$

□

For the iterates of the quadratic penalty methods it holds $x^{(k)} \notin X$. [If $x^{(k)} \in X$ it holds $0 = \nabla P_2(x^{(k)}, \alpha_{k-1}) = \nabla f(x^{(k)})$ such that $x^{(k)}$ is a stationary point of f, which is in general not true.] Therefore we need to drive $\alpha \to \infty$. This is problematic since the minimization of $P_2(\cdot, \alpha)$ becomes more difficult for increasing α.

3.2.2 Exact Penalty Method

We do not want to force $\alpha \to \infty$ but get similar results.

Definition 3.2.5
Let $x^* \in X$ be a solution of (PC). A penalty function π is called exact *(in x^*) if there exists an $\bar{\alpha} > 0$ such that x^* is also a solution of*

$$\min_{x \in \mathbb{R}^n} f(x) + \alpha \pi(x) \quad \forall \alpha \geq \bar{\alpha}.$$

The utilization of exact penalty functions obviously overcomes the difficulty of $\alpha \to \infty$ but comes as price:

Theorem 3.2.6
Let $x^* \in X$ be a solution of (PC) with $\nabla f(x^*) \neq 0$ and let π be exact. Then, π is not differentiable.

Proof.
Assume that π is differentiable. Since π is exact there exists an $\bar{\alpha} > 0$ such that x^* is a minimizer of $P(\cdot, \alpha)$ for all $\alpha > \bar{\alpha}$. Especially it holds for all $\alpha > \bar{\alpha}$ that

$$\nabla P(x^*, \alpha) = 0 \;\Leftrightarrow\; \nabla f(x^*) + \alpha \pi(x^*) = 0 \; .$$

For arbitrary $\alpha_1, \alpha_2 \geq \bar{\alpha}$ with $\alpha_1 \neq \alpha_2$ it holds

$$\begin{aligned} & \nabla f(x^*) + \alpha_1 \nabla \pi(x^*) = 0 = \nabla f(x^*) + \alpha_2 \nabla \pi(x^*) \\ \Leftrightarrow\; & (\alpha_1 - \alpha_2) \nabla \pi(x^*) = 0 \\ \Leftrightarrow\; & \nabla \pi(x^*) = 0 \; . \end{aligned}$$

Since $0 = \nabla P(x^*, \alpha) = \nabla f(x^*) + \alpha \pi(x^*) = \nabla f(x^*)$ we have a contradiction. □

The most popular exact penalty function is the *l1-penalty*

$$\pi_1(x) := \sum_{i=1}^{m} g_i(x)^+ + \sum_{j=1}^{p} |h_j(x)| = \|g(x)^+\|_1 + \|h(x)\|_1$$

Theorem 3.2.7
Let $x^* \in X$ be a strict (local) solution of (PC) which meets the LICQ. Then π_1 is exact in x^*.

Proof. Cf. Geiger and Kanzow (2013, Theorem 5.14). □

3.2.3 Barrier Methods

For penalty methods we have seen that $x^{(k)} \to x^*$ with $x^{(k)} \notin X$, i.e. the sequence approximates x^* from outside of the feasible set. This is improper, for example, if f is only defined on X.

Barrier-methods approximate the solution from the interior of X, i.e. $x^{(k)} \in X$ for all $k \in \mathbb{N}_0$. For the inequality constrained problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m \end{aligned}$$

a popular approach is to minimize the *logarithmic barrier function*

$$B(x, \alpha) := \begin{cases} f(x) - \alpha \underbrace{\sum_{i=1}^{m} \log(-g_i(x))}_{=:\pi_b(x)}, & \text{if } x \in X \\ \infty, & \text{if } x \notin X \end{cases}$$

for some $\alpha > 0$. To guarantee the existence of a minimizer of $B(\cdot, \alpha)$ we need a strictly feasible point x, i.e. $g_i(x) < 0$ for all $i = 1, ..., m$.

Definition 3.2.8
A point $x \in X$ with $g_i(x) < 0$ for all $i = 1, ..., m$ is said to fulfil the Slater condition. *We define the strict interior of X as the set*

$$X^\circ = \{x \in X : g_i(x) < 0, \ \forall i = 1, ..., m\}.$$

Theorem 3.2.9
Let $\emptyset \neq X \subseteq \mathbb{R}^n$ be compact and let $f, g_i \in C^1(X)$. If the Slater condition is fulfilled for some $\bar{x} \in X$, then $B(\cdot, \alpha)$ has a minimizer for all $\alpha > 0$.

Proof.
Since $\bar{x} \in X$ meets the Slater condition it holds $c := B(\bar{x}, \alpha) < \infty$. We define the set

$$X_c := \{x \in X : B(x, \alpha) \leq c\}$$

such that a minimizer of $B(\cdot, \alpha)$ has to be a solution of

$$\min_{x \in X_c} B(x, \alpha).$$

To prove the existence of a minimizer it suffices to show that X_c is compact. Since $X_c \subseteq X$ the set X_c is bounded. Let $(x^{(k)})_{k \in \mathbb{N}_0} \subseteq X_c$ denote a sequence with limit $\tilde{x} \in X$. Since $B(\cdot, \alpha)$ is continuous on X_c it holds

$$B(\tilde{x}, \alpha) = \lim_{k \to \infty} B(x^{(k)}, \alpha) \leq c,$$

such that $x \in X_c$, i.e. X_c is closed. □

In analogy to the penalty methods we obtain the log-barrier method:

Algorithm 3.2.10: log-barrier method

Input: $x^{(0)} \in X^0,\ c < 1,\ \alpha_0 > 0,\ k \leftarrow 0$

1 **for** $k = 0, 1, \ldots$ **do**

2 $\quad x^{(k+1)} \leftarrow \operatorname{argmin}_{x \in \mathbb{R}^n} B(x, \alpha_k)$; `// warm start with` $x^{(k)}$

3 $\quad \alpha_{k+1} \leftarrow c\alpha_k$

4 **end**

Lemma 3.2.11
Let $(x^{(k)})_{k \in \mathbb{N}_0}$ and $(\alpha_k)_{k \in \mathbb{N}_0}$ be generated from Algorithm 3.2.10. Then it holds

1. *$(\pi_b(x^{(k)}))_{k \in \mathbb{N}_0}$ is monotonically increasing.*
2. *$(f(x^{(k)}))_{k \in \mathbb{N}_0}$ is monotonically decreasing.*

Proof.

1. Since $B(x^{(k+1)}, \alpha_k) \leq B(x^{(k)}, \alpha_k)$ and $B(x^{(k)}, \alpha_{k-1}) \leq B(x^{(k+1)}, \alpha_{k-1})$ we have

$$\begin{aligned} & B(x^{(k+1)}, \alpha_k) - B(x^{(k+1)}, \alpha_{k-1}) \leq B(x^{(k)}, \alpha_k) - B(x^{(k)}, \alpha_{k-1}) \\ \Leftrightarrow\ & (\alpha_k - \alpha_{k-1})\pi_b(x^{(k+1)}) \leq (\alpha_k - \alpha_{k-1})\pi_b(x^{(k)}) \\ \overset{\alpha_k \leq \alpha_{k-1}}{\Leftrightarrow}\ & \pi_b(x^{(k+1)}) \geq \pi_b(x^{(k)}) \ . \end{aligned}$$

2. By the optimality of $x^{(k+1)}$ for $B(\cdot, \alpha_k)$ we have
   (a) $B(x^{(k)}, \alpha_{k-1}) \leq B(x^{(k+1)}, \alpha_{k-1})$,
   (b) $B(x^{(k+1)}, \alpha_k) \leq B(x^{(k)}, \alpha_k)$.

Multiplication of (a) with $\alpha_k/\alpha_{k-1} = c \in]0,1[$ yields

$$\frac{\alpha_k}{\alpha_{k-1}} f(x^{(k)}) - \alpha_k \pi_b(x^{(k)}) \leq \frac{\alpha_k}{\alpha_{k-1}} f(x^{(k+1)}) - \alpha_k \pi_b(x^{(k+1)}),$$

and addition with (b) yields

$$\begin{aligned}
&\frac{\alpha_k}{\alpha_{k-1}} f(x^{(k)}) + f(x^{(k+1)}) - \alpha_k \pi_b(x^{(k)}) - \alpha_k \pi_b(x^{(k+1)}) \\
&\qquad \leq \frac{\alpha_k}{\alpha_{k-1}} f(x^{(k+1)}) + f(x^{(k)}) - \alpha_k \pi_b(x^{(k+1)}) - \alpha_k \pi_b(x^{(k)}) \\
\Leftrightarrow\ &\frac{\alpha_k}{\alpha_{k-1}} f(x^{(k)}) + f(x^{(k+1)}) \leq \frac{\alpha_k}{\alpha_{k-1}} f(x^{(k+1)}) + f(x^{(k)}) \\
\Leftrightarrow\ &\left(\frac{\alpha_k}{\alpha_{k-1}} - 1\right) f(x^{(k)}) \leq \left(\frac{\alpha_k}{\alpha_{k-1}} - 1\right) f(x^{(k+1)}) \\
\Leftrightarrow\ &f(x^{(k)}) \geq f(x^{(k+1)}) \ .
\end{aligned}$$

□

Theorem 3.2.12
Let $X^0 \neq \emptyset$, $f, g_i \in C(X)$ and let $(x^{(k)})_{k\in\mathbb{N}_0}$ be generated by Algorithm 3.2.10. Then, every accumulation point of the sequence is a global solution of (PC).

Proof.
Let x^* be an accumulation point of the sequence $(x^{(k)})_{k\in\mathbb{N}_0} \subseteq X^\circ$ and without loss of generality we assume $x^{(k)} \to x^*$, such that $x^* \in \bar{X^\circ} \subseteq X$. Assume that x^* is not a global solution of (PC), such that there exists an $\bar{x} \in X$ with $f(\bar{x}) < f(x^*)$. Since f is continuous, there exists an $\hat{x} \in X$ with $f(\hat{x}) < f(x^*)$. It holds

$$f(x^{(k+1)}) - \alpha_k \pi_b(x^*) \overset{\pi_b \nearrow}{\leq} f(x^{(k+1)}) - \alpha_k \pi_b(x^{(k+1)}) \leq f(\hat{x}) - \alpha_k \pi_b(\hat{x})$$

which yields

$$f(x^*) = \lim_{k\to\infty} f(x^{(k+1)}) = \lim_{k\to\infty} f(x^*) - \underbrace{\alpha_k}_{\to 0} \underbrace{(\pi_b(\hat{x}) - \pi_b(x^*))}_{<\infty} = f(\hat{x})$$

which is a contraction to $f(x^*) > f(\hat{x})$. □

3.2.4 Augmented Lagrangian Method

We consider the equality constrained problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0 \end{aligned} \tag{PEQ}$$

If x^* is solution of (PEQ) it is also a solution of

$$\begin{aligned} \min \quad & f(x) + \frac{\alpha}{2}\|h(x)\|^2 \\ \text{s.t.} \quad & h(x) = 0 \end{aligned} \tag{PEQ2}$$

with arbitrary $\alpha > 0$.

Definition 3.2.13
The Lagrangian of (PEQ2)

$$L_A(x, \mu; \alpha) = f(x) + \frac{\alpha}{2}\|h(x)\|^2 + \mu^\top h(x)$$

is called augmented Lagrangian *of* (PEQ).

Lemma 3.2.14
Let $Q \in \mathbb{R}^{n\times n}$ be symmetric and positive semidefinite and let $P \in \mathbb{R}^{n\times n}$ be symmetric and positive definite on $\operatorname{kern}(Q) := \{x \in \mathbb{R}^n : Qx = 0\}$. *Then there exists an $\bar\alpha > 0$ such that $P + \alpha Q$ is symmetric and positive definite for all $\alpha \geq \bar\alpha$.*

Theorem 3.2.15
Let (x^*, μ^*) be a KKT-point of (PEQ) *that meets the SSOC (cf. Theorem 3.1.12). Then there exists an $\bar\alpha > 0$ such that for all $\alpha \geq \bar\alpha$ the point x^* is a strict minimizer of $L_A(\cdot, \mu^*; \alpha)$.*

Proof.
It holds

$$L_A(x, \mu; \alpha) = L(x, \mu) + \frac{\alpha}{2}\|h(x)\|^2$$

such that

$$\begin{aligned} \nabla_x L_A(x, \mu; \alpha) &= \nabla_x L(x, \mu) + \alpha \sum_{j=1}^{p} h_j(x)\nabla h_j(x) \\ \nabla^2_{xx} L_A(x, \mu; \alpha) &= \nabla^2_{xx} L(x, \mu) + \alpha J_h(x)^\top J_h(x), \end{aligned}$$

where $J_h(x)$ denotes the Jacobi matrix of h in x. Since (x^*, μ^*) is a KKT-point of (PEQ) it holds

$$L_A(x^*, \mu^*; \alpha) = \underbrace{\nabla_x L(x^*, \mu^*)}_{=0} + \alpha \sum_{j=1}^{p} \underbrace{h_j(x^*)}_{=0 \ (x^* \in X)} \nabla h_j(x^*) = 0.$$

With $B_* := J_h(x^*)$ we have

$$\nabla^2_{xx} L_A(x^*, \mu^*; \alpha) = \nabla^2_{xx} L(x^*, \mu^*) + \alpha B_*^\top B_* \ .$$

Due to the SSOC it holds

$$d^\top \nabla^2_{xx} L(x^*, \mu^*) d > 0 \quad \forall d \neq 0 : \ \nabla h_j(x^*)^\top d = 0, \ j = 1, ..., p,$$

i.e.

$$\nabla^2_{xx} L(x^*, \mu^*) > 0$$

on $\ker(B_*) = \ker(B_*^\top B_*)$. By Lemma 3.2.14 [with $Q = B_*^\top B_*$ and $P = \nabla^2_{xx} L(x^*, \mu^*)$] there exists an $\bar{\alpha} > 0$ such that $\nabla^2_{xx}(x^*, \mu^*; \alpha) > 0$ for all $\alpha > \bar{\alpha}$, such that, by Theorem 2.1.5, x^* is a strict local minimizer of $L_A(\cdot, \mu^*; \alpha)$ for all $\alpha > \bar{\alpha}$. □

The above theorem implies that a solution of (PEC) can be found by solving the unconstrained problem

$$\min_{x \in \mathbb{R}^n} L(\cdot, \mu^*; \alpha)$$

for some fixed and „large enough“ α. The advantages of this approach are that L_A is continuously differentiable and we do not need $\alpha \to \infty$. The disadvantage is that μ^* is unknown. However, μ^* can be estimated/updated. For given $\mu^{(k)}$ and fixed $\alpha > \bar{\alpha}$ let

$$x^{(k+1)} \leftarrow \operatorname*{argmin}_{x \in \mathbb{R}^n} L_A(x, \mu^{(k)}; \alpha)$$

such that

$$0 = \nabla_x L_A(x^{(k+1)}, \mu^{(k)}; \alpha) = \nabla f(x^{(k+1)}) + \sum_{j=1}^{p} (\mu_j^{(k)} + \alpha h_j(x^{(k)})) \nabla h_j(x^{(k+1)}) \ .$$

Since (x^*, μ^*) is a KKT-point of (PEQ) we also know that

$$0 = \nabla_x L(x^*, \mu^*) = \nabla f(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) \ .$$

This leads to the so called Hestens-Powell update formula

$$\mu^{(k+1)} \leftarrow \mu^{(k)} + \alpha h(x^{(k+1)}) \ .$$

Algorithm 3.2.16: Augmented Lagrangian method

Input: $x^{(0)} \in \mathbb{R}^n,\ \mu^{(0)} \in \mathbb{R}^p,\ \alpha_0 > 0,\ c \in]0, 1[,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|h(x^{(k)})\| > \varepsilon$ **do**
  $x^{(k+1)} \leftarrow \operatorname*{argmin}_{x \in \mathbb{R}^n} L_A(x, \mu^{(k)}, \alpha_k)$; `// warm start with` $x^{(k)}$
  $\mu^{(k+1)} \leftarrow \mu^{(k)} + \alpha_k h(x^{(k+1)})$
  if $\|h(x^{(k+1)})\| \geq c\|h(x^{(k)})\|$ **then**
    $\alpha_{k+1} \leftarrow 10\alpha_k$
  end
  $k \leftarrow k + 1$
end

For a general constrained problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0, \\ & h(x) = 0 \end{aligned}$$

we use slack variables s_i and can solve

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) + s_i^2 = 0, \qquad i = 1, ..., m \\ & h(x) = 0 \end{aligned}$$

using the augmented Lagrangian approach as well. However, much more efficient method exists for those general problems (cf. SQP-method in Section 3.4).

3.3 Quadratic Programming

An optimization problem with quadratic objective function and linear constraints, i.e.

$$\begin{aligned} \min \quad & \frac{1}{2}x^\top Qx + c^\top x \\ \text{s.t.} \quad & a_i^\top x \leq \alpha_i \qquad i = 1, ..., m \\ & b_j^\top x = \beta_j \qquad j = 1, ..., p, \end{aligned} \tag{QP}$$

with symmetric matrix $Q \in \mathbb{R}^{n\times n}$, $c \in \mathbb{R}^n$, $a_i, b_j \in \mathbb{R}^n$ and $\alpha_i, \beta_j \in \mathbb{R}$ is called *quadratic program* (QP). These types of problems frequently occur as autonomous problems as well as as subproblems as in the SQP-method (cf. Section 3.4).

3.3.1 Equality Constrained Quadratic Programming

We first consider the case of only equality constraints, i.e.

$$\begin{aligned} \min \quad & \frac{1}{2}x^\top Qx + c^\top x \\ \text{s.t.} \quad & b_j^\top x - \beta_j = 0 \qquad j = 1, ..., p\ . \end{aligned} \tag{EQP}$$

Let x^* denote a solution of (EQP). By Exercise 31 the ACQ is met for (EQP) such that by Theorem 3.1.7 there exists a Lagrangian multiplier $\mu^* \in \mathbb{R}^p$ such that (x^*, μ^*) meets the KKT-conditions

1. $0 = \nabla_x L(x^*, \mu^*) = \nabla f(x^*) + \sum_{j=1}^p \mu_j^* h_j(x^*) = Qx^* + c + \sum_{j=1}^p \mu_j^* b_j$,
2. $h_j(x^*) = 0 \Leftrightarrow b_j^\top x^* = \beta_j\ \forall j = 1, ..., p$.

This is equivalently written as

$$\begin{aligned} Qx^* + B^\top \mu^* &= -c \\ Bx^* &= \beta\ , \end{aligned}$$

where

$$B := \begin{bmatrix} -b_1^\top - \\ \vdots \\ -b_p^\top - \end{bmatrix} \in \mathbb{R}^{p \times n}, \quad \beta := \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \in \mathbb{R}^p,$$

which directly provides the following results.

Theorem 3.3.1
A pair (x^*, μ^*) is a KKT-point of (EQP) if and only if it is a solution of the linear system

$$\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} x \\ \mu \end{pmatrix} \overset{!}{=} \begin{pmatrix} -c \\ \beta \end{pmatrix}.$$

Thus, solving (EQP) can be achieved by solving a linear system. For the following examinations a reformulation will become helpful.

Theorem 3.3.2
Let $x^{(k)} \in X$ be feasible for (EQP). Then $(x^*, \mu^*) \in \mathbb{R}^n \times \mathbb{R}^p$ is a KKT-point of (EQP) if and only if $x^* = x^{(k)} + \Delta x^*$, where $(\Delta x^*, \mu^*)$ is a solution of

$$\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \mu \end{pmatrix} \overset{!}{=} \begin{pmatrix} -\nabla f(x^{(k)}) \\ 0 \end{pmatrix} = \begin{pmatrix} -Qx^{(k)} - c \\ 0 \end{pmatrix}.$$

Proof.
Let (x^*, μ^*) be a KKT-point of (EQP) and let $x^* = x^{(k)} + \Delta x^*$. By Theorem 3.3.1 it holds

$$\begin{aligned}
&\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} x^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} -c \\ \beta \end{pmatrix} \\
\Leftrightarrow\ &\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} x^{(k)} + \Delta x^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} -c \\ \beta \end{pmatrix} \\
\Leftrightarrow\ &\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} x^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} -c \\ \beta \end{pmatrix} - \begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} x^{(k)} \\ 0 \end{pmatrix} \\
\Leftrightarrow\ &\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} \Delta x^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} -c - Qx^{(k)} \\ \beta - \underbrace{Bx^{(k)}}_{=\beta,\ \text{since } x^{(k)} \in X} \end{pmatrix} = \begin{pmatrix} -c - Qx^{(k)} \\ 0 \end{pmatrix}.
\end{aligned}$$

□

The question arises under which conditions a solution of the KKT-system exists.

Lemma 3.3.3
If $\operatorname{rank}(B) = p \leq n$ (i.e. the LICQ) and if $Q > 0$ on $\ker(B) \setminus \{0\}$ (i.e. the SSOC), then the KKT-matrix

$$\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix}$$

is nonsingular.

Proof.
Since a matrix is nonsingular if and only if its kernel is equal to $\{0\}$ we assume

$$\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} x \\ \mu \end{pmatrix} = 0$$

such that

$$\begin{aligned} &(1)\ Qx + B^\top \mu = 0 \qquad \text{and} \qquad (2)\ Bx = 0 \\ \Rightarrow\ & 0 = x^\top (Qx + B^\top \mu) = x^\top Qx + x^\top B^\top \mu = x^\top Qx + (Bx)^\top \mu \overset{(2)}{=} x^\top Qx \\ \Rightarrow\ & x = 0 \text{ since } Q > 0 \text{ on } \operatorname{kern}(B) \\ \overset{(1)}{\Rightarrow}\ & B^\top \mu = 0 \\ \Rightarrow\ & \mu = 0 \text{ since } B \text{ is of full column rank.} \end{aligned}$$

□

The KKT-system can be solved brute force with classical decomposition or iterative methods such as the CG-method. For this special problem, however, more sophisticated methods exist.

1. Range-space method:
   If Q is symmetric positive definite, we obtain from the first KKT-equation that
   $$x = -Q^{-1}(B^\top \mu + c)$$
   and plugging this into the second KKT-equation yields
   $$\begin{aligned} & -BQ^{-1}B^\top \mu - BQ^{-1}c = \beta \\ \Leftrightarrow\ & \mu = -(BQ^{-1}B)(BQ^{-1}c + \beta)\ . \end{aligned}$$
   Thus, we get the solution via solving $(B^\top Q^{-1}B)\mu \overset{!}{=} -(BQ^{-1}c + \beta)$ and computing $x^* = Q^{-1}(B^\top \mu^* + c)$. This approach is especially useful if Q^{-1} is explicitly known, as for example in the SQP-method with quasi-Newton update.

2. Null-space method:
   Find an arbitrary basis $z_1, ..., z_{n-p}$ of $\operatorname{kern}(B)$ and set $Z := [z_1, ..., z_{n-p}] \in \mathbb{R}^{n\times(n-p)}$. Assume $\operatorname{rank}(B) = p$ and $Z^\top QZ > 0$. Let $\hat{x}$ be a particular solution of the second KKT-equation, i.e. $B\hat{x} = \beta$ such that the KKT-system reads
   $$\begin{bmatrix} Q & B^\top \\ B & 0 \end{bmatrix} \begin{pmatrix} \bar{x} \\ \mu \end{pmatrix} = \begin{pmatrix} -c - Q\hat{x} \\ 0 \end{pmatrix},$$
   where $x := \hat{x} + \bar{x}$ (cf. Theorem 3.3.2). The second equation is equivalent to finding a $z \in \mathbb{R}^{n-p}$ such that $\bar{x} = Zz$ and plugging this into first equation yields
   $$\begin{aligned} & QZz + B^\top \mu \overset{!}{=} -c - Q\hat{x} \\ \Leftrightarrow\ & \underbrace{Z^\top QZ}_{>0} z + \underbrace{Z^\top B^\top}_{=0} \mu = -Z^\top (Q\hat{x} + c). \end{aligned}$$
   Hence
   $$\begin{aligned} z^* &= -(Z^\top QZ)^{-1} Z(Q\hat{x} + c) \\ x^* &= \hat{x} + Zz^*\ . \end{aligned}$$

3.3.2 Active Set Method

We now turn back to the general (QP)

$$\begin{aligned} \min \quad & \frac{1}{2}x^\top Qx + c^\top x \\ \text{s.t.} \quad & a_i^\top x \leq \alpha_i \qquad i = 1, ..., m \\ & b_j^\top x = \beta_j \qquad j = 1, ..., p\ . \end{aligned} \tag{QP}$$

The basic idea of the active set method is to solve (QP) by repetitively solving an (EQP). This is motivated by the fact that a solution x^* of (QP) is also a solution of the equality constrained QP

$$\begin{aligned} \min \quad & \frac{1}{2}x^\top Qx + c^\top x \\ s.t. \quad & a_i^\top x = \alpha_i \qquad \forall i \in A_X(x^*) \\ & b_j^\top x = \beta_j \qquad j = 1, ..., p, \end{aligned} \tag{EQP-active}$$

where $A_X(x^*)$ is the active set in x^*.

Lemma 3.3.4
Let x^* be a solution of (QP), then x^* is also a solution of (EQP-active). Conversely, if x^* is a feasible point of (QP) and a KKT-point of (EQP-active) and if the corresponding Lagrangian multiplier λ^* satisfies

$$\lambda_i^* \geq 0 \qquad \forall i \in A_X(x^*),$$

then x^* is also a KKT-point of (QP).

Proof.
The first statement is obvious. For the second statement let x^* be feasible for (QP) and a KKT-point of (EQP-active) and let $\lambda_i^*,\ i \in A_X(x^*)$, such that

$$\begin{aligned} \text{KKT} \quad & \begin{cases} Qx^* + c + \sum_{i \in A_X(x^*)} \lambda_i^* a_i + \sum_{j=1}^p \mu_j^* b_j = 0 \\ \lambda_i^*(a_i^\top x^* - \alpha_i) = 0 \qquad \forall i \in A_X(x^*) \end{cases} \\ \text{Assumption} \quad & \left\{ \lambda_i^* \geq 0 \qquad \forall i \in A_X(x^*) \right. \end{aligned}$$

Defining $\lambda_i^* = 0$ for all $i \notin A_X(x^*)$ we obtain

$$\text{KKT of (QP)}: \begin{cases} Qx^* + c + \sum_{i=1}^m \lambda_i^* c_i + \sum_{j=1}^p \mu_j^* b_j = 0 \\ a_i^\top x^* \leq \alpha_i \qquad \forall i = 1, ..., m \\ b_j^\top x^* = \beta_j \qquad \forall j = 1, ..., p \\ \lambda_i^* \geq 0 \qquad \forall i = 1, ..., m \\ \lambda_i^*(a_i^\top x^* - \alpha_i) = 0 \qquad \forall i = 1, ..., m \end{cases}$$

such that (x^*, λ^*, μ^*) is a KKT-point of (QP). □

The above theorem yields that, if $A_X(x^*)$ is known a priori, the (QP) reduces to finding a feasible point of (QP) that solves (EQP-active). However, $A_X(x^*)$ is generally unknown. The active set strategy is to find the true active set by successive exchange steps.

Let therefore $\mathcal{A}_k$ be an approximation of $A_X(x^*)$ (*working set*) and define

$$A_k := \begin{bmatrix} - \ a_1^\top - \\ \vdots \\ - \ a_{\#\mathcal{A}_k}^\top - \end{bmatrix} \in \mathbb{R}^{\#\mathcal{A}_k \times n} \quad \text{and} \quad B := \begin{bmatrix} - \ b_1^\top - \\ \vdots \\ - \ b_p^\top - \end{bmatrix} \in \mathbb{R}^{p \times n}.$$

The active set method is defined as follows:

Algorithm 3.3.5: Active set method

Input: $x^{(0)} \in X$, $\mathcal{A}_0 = A_X(x^{(0)}$

for $k = 0, 1, \ldots$ **do**

  $\begin{pmatrix} \Delta x^{(k)} \\ \lambda^{(k)} \\ \mu^{(k)} \end{pmatrix} = \begin{bmatrix} Q & A_k^\top & B^\top \\ A_k & 0 & 0 \\ B & 0 & 0 \end{bmatrix}^{-1} \begin{pmatrix} -Qx^{(k)} - c \\ 0 \\ 0 \end{pmatrix}$

  if $\Delta x^{(k)} = 0$ **then**

    if $\lambda^{(k)} \geq 0$ **then**

      STOP: $x^{(k)}$ is solution of (QP)

    end

    else

      $x^{(k+1)} \leftarrow x^{(k)}$

      $\lambda_q^{(k)} \leftarrow \operatorname{argmin}\{\lambda_i^{(k)}, \ i \in \mathcal{A}_k\} < 0$

      $\mathcal{A}_{k+1} \leftarrow \mathcal{A}_k \setminus \{q\}$

      $k \leftarrow k + 1$

    end

  end

  else

    if $x^{(k)} + \Delta x^{(k)} \in X$ **then**

      $x^{(k+1)} \leftarrow x^{(k)} + \Delta x^{(k)}$

      $\mathcal{A}_{k+1} \leftarrow \mathcal{A}_k$

      $k \leftarrow k + 1$

    end

    else

      $t_k \leftarrow \dfrac{\alpha_r - a_r^\top x^{(k)}}{a_r^\top \Delta x^{(k)}} = \min\limits_{\substack{i \notin \mathcal{A}_k \\ a_r^\top \Delta x^{(k)} > 0}} \dfrac{\alpha_i - a_i^\top x^{(k)}}{a_i^\top \Delta x^{(k)}}$

      $x^{(k+1)} \leftarrow x^{(k)} + t_k \Delta x^{(k)}$

      $\mathcal{A}_{k+1} \leftarrow \mathcal{A}_k \cup \{r\}$ $k \leftarrow k + 1$

    end

  end

end

We will show that if $x^{(0)} \in X$, then $x^{(k)} \in X$ such that the active set method produces only feasible points. Therefore, if $\Delta x^{(k)} = 0$ and $\lambda^{(k)} \geq 0$, Theorem 3.3.2 yields that $x^{(k)}$ is a solution of (QP). Otherwise if $\Delta x^{(k)} = 0$ and $\lambda_q^{(k)} < 0$ for some $q \in \mathcal{A}_k$, $x^{(k)}$ cannot be a solution of (QP) but we also cannot decrease the objective function on the current working set $\mathcal{A}_k$. Therefore we remove the index q from $\mathcal{A}_k$.

If $\Delta x^{(k)} \neq 0$ and $x^{(k)} + \Delta x^{(k)} \in X$ we update $x^{(k+1)} \leftarrow x^{(k)} + \Delta x^{(k)}$ and do not alter the working set. Finally, if $x^{(k)} + \Delta x^{(k)} \notin X$, the new iterate would violate a constraint $\{1, ..., m\} \setminus \mathcal{A}_k$. To ensure that the new iterate stays feasible for (QP) we use a damping factor $t_k > 0$ that guarantees

$$a_i^\top (x^{(k)} + t_k \Delta x^{(k)}) \leq \alpha_i \quad \forall i \in \{1, ..., m\} \setminus \mathcal{A}_k \ .$$

Since $a_i^\top x^{(k)} \leq \alpha_i$ this holds always true if $a_i^\top \Delta x^{(k)} \leq 0$. Otherwise we choose

$$t_k = \frac{\alpha_r - a_r^\top x^{(k)}}{a_r^\top \Delta x^{(k)}} = \min_{i \notin \mathcal{A}_k} \frac{\alpha_i - a_i^\top x^{(k)}}{a_i^\top \Delta x^{(k)}} \ ,$$

such that $x^{(k+1)} \leftarrow x^{(k)} + t_k \Delta x^{(k)} \in X$ and add the index r to the working set, since it is now active. Note that such an index r with $a_r^\top \Delta x^{(k)} > 0$ always exists, since otherwise $x^{(k)} + \Delta x^{(k)}$ would already be feasible.

3.4 SQP-Method

We consider

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0, \\ & h(x) = 0. \end{aligned} \tag{PC}$$

Sequential quadratic programming (SQP) methods are the most successful methods for the numerical solution of (PC). The basic idea is to approximate the (PC) by a (QP) in each iteration. They can be seen as an extension of (Quasi-) Newton methods to constrained optimization.

3.4.1 Lagrange-Newton Method

We first consider the equality constrained problem

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

A point (x^*, μ^*) is a KKT-point of (PEC) if and only if

$$\begin{aligned} \nabla_x L(x^*, \mu^*) &= \nabla f(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) = 0 \\ \nabla_\mu L(x^*, \mu^*) &= h(x^*) = 0, \end{aligned}$$

i.e. if

$$\nabla L(x^*, \mu^*) = 0.$$

This is a nonlinear system which can be solved by means of the Newton-method. Let $(x^{(k)}, \mu^{(k)})$ be an approximation of (x^*, μ^*). Then the Newton systems reads

$$\begin{aligned}
&\nabla^2 L(x^{(k)}, \mu^{(k)}) \begin{pmatrix} \Delta x^{(k)} \\ \Delta \mu^{(k)} \end{pmatrix} \overset{!}{=} -\nabla L(x^{(k)}, \mu^{(k)}) \\
\Leftrightarrow\ &\begin{bmatrix} \nabla^2_{xx} L(x^{(k)}, \mu^{(k)}) & \nabla^2_{x\mu} L(x^{(k)}, \mu^{(k)}) \\ \nabla^2_{\mu x} L(x^{(k)}, \mu^{(k)}) & \nabla^2_{\mu\mu} L(x^{(k)}, \mu^{(k)}) \end{bmatrix} \begin{pmatrix} \Delta x^{(k)} \\ \Delta \mu^{(k)} \end{pmatrix} \overset{!}{=} - \begin{pmatrix} \nabla_x L(x^{(k)}, \mu^{(k)}) \\ \nabla_\mu L(x^{(k)}, \mu^{(k)}) \end{pmatrix} \\
\Leftrightarrow\ &\begin{bmatrix} \nabla^2_{xx} L(x^{(k)}, \mu^{(k)}) & J_h(x^{(k)})^\top \\ J_h(x^{(k)}) & 0 \end{bmatrix} \begin{pmatrix} \Delta x^{(k)} \\ \Delta \mu^{(k)} \end{pmatrix} \overset{!}{=} - \begin{pmatrix} \nabla_x L(x^{(k)}, \mu^{(k)}) \\ h(x^{(k)}) \end{pmatrix} \qquad \text{(LN-System)}
\end{aligned}$$

This approach leads to the Lagrange-Newton method:

Algorithm 3.4.1: Lagrange-Newton method

Input: $x^{(0)} \in \mathbb{R}^n,\ \mu^{(0)} \in \mathbb{R}^p,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|\nabla L(x^{(k)}, \mu^{(k)})\| > \varepsilon$ **do**
- compute $(\Delta x^{(k)}, \Delta \mu^{(k)})$ as solution of (LN-System)
- $x^{(k+1)} \leftarrow x^{(k)} + \Delta x^{(k)}$
- $\mu^{(k+1)} \leftarrow \mu^{(k)} + \Delta \mu^{(k)}$
- $k \leftarrow k + 1$

end

As for the Newton-method we obtain local superlinear/quadratic convergence to a KKT-point of (PEC) provided the matrix $\nabla^2 L(x^*, \mu^*)$ is nonsingular.

Theorem 3.4.2
Let (x^*, μ^*) be a KKT-point of (PEC) and assume

1. *$\nabla h_1(x^*), ..., \nabla h_p(x^*)$ are linear independent, i.e. the LICQ,*
2. *$d^\top \nabla^2_{xx} L(x^*, \mu^*) d > 0 \quad \forall d \neq 0 : \nabla h_j(x^*)^\top d = 0\ \forall j = 1, ..., p$, i.e. the SSOC.*

Then

$$\nabla^2 L(x^*, \mu^*) = \begin{bmatrix} \nabla^2_{xx} L(x^*, \mu^*) & J_h(x^*)^\top \\ J_h(x^*) & 0 \end{bmatrix}$$

is nonsingular.

Proof.
Let

$$d := \begin{pmatrix} d^{(1)} \\ d^{(2)} \end{pmatrix} \in \mathbb{R}^n \times \mathbb{R}^p$$

such that

$$
\begin{aligned}
&\begin{bmatrix} \nabla^2_{xx} L(x^*, \mu^*) & J_h(x^*)^\top \\ J_h(x^*) & 0 \end{bmatrix} \begin{pmatrix} d^{(1)} \\ d^{(2)} \end{pmatrix} = 0 \\
\Leftrightarrow\ & I) \quad \nabla^2_{xx} L(x^*, \mu^*) d^{(1)} + \sum_{j=1}^{p} g_j^{(2)} \nabla h_j(x^*) = 0 \\
& II) \quad \nabla h_j(x^*)^\top d^{(1)} = 0 \quad \forall j = 1, ..., p \\
\Rightarrow\ & (d^{(1)})^\top L(x^*, \mu^*) d^{(1)} + \sum_{j=1}^{p} d^{(2)} (\underbrace{(d^{(1)})^\top \nabla h_j(x^*)}_{=0,\ II)}) = 0 \\
\Rightarrow\ & d^{(1)})^\top L(x^*, \mu^*) d^{(1)} = 0 \\
\overset{2.}{\Rightarrow}\ & d^{(1)} = 0 \\
\overset{I)}{\Rightarrow}\ & \sum_{j=1}^{p} d_j^{(2)} \nabla h_j(x^*) = 0 \\
\overset{LICQ}{\Rightarrow}\ & d_j^{(2)} = 0.
\end{aligned}
$$

□

For a general (PC) the KKT-conditions read

$$
\begin{aligned}
\nabla_x L(x, \lambda, \mu) &= 0 \\
h(x) &= 0 \\
g(x) &\leq 0 \\
\lambda &\geq 0 \\
\lambda_i g_i(x) &= 0 \quad \forall i = 1, ..., m
\end{aligned}
$$

which is equivalently written as

$$
\begin{aligned}
\nabla_x L(x, \lambda, \mu) &= 0 \\
h(x) &= 0 \\
\min\{-g_i(x), \lambda_i\} &= 0 \quad \forall i = 1, ..., m
\end{aligned}
$$

and we could proceed as in the Lagrange-Newton method. However, since the min-function is not differentiable everywhere, this approach is not straightforward. We can overcome this problem, for example, by a semi-smooth Newton method.

3.4.2 Local SQP-Method

The Lagrange-Newton method require the solution of

$$
\nabla^2 L(x^*, \mu^*) \begin{pmatrix} \Delta x \\ \Delta \mu \end{pmatrix} \overset{!}{=} -\nabla L(x^{(k)}, \mu^{(k)})
$$

in each iteration, which is equivalently written as

$$\begin{aligned} B_k \Delta x + J_h(x^{(k)})^\top \Delta\mu &\overset{!}{=} -\nabla L(x^{(k)}, \mu^{(k)}) \\ \nabla h_j(x^*)\Delta x &\overset{!}{=} -h_j(x^{(k)}) \quad \forall j = 1, ..., p \end{aligned}$$

where $B_k := \nabla^2_{xx} L(x^{(k)}, \mu^{(k)})$ or later $B_k \approx \nabla^2_{xx} L(x^{(k)}, \mu^{(k)})$. Defining $\mu^+ := \mu^{(k)} + \Delta\mu$ this reads

$$\begin{aligned} B_k \Delta x + J_h(x^{(k)})^\top \mu^+ &= -\nabla f(x^{(k)}) \\ \nabla h_j(x^{(k)})^\top \Delta x &= -h_j(x^{(k)}). \end{aligned}$$

This exactly represents the KKT-conditions of the quadratic program

$$\begin{aligned} \min_{\Delta x \in \mathbb{R}^n} \quad & \frac{1}{2} \Delta x^\top B_k \Delta x + \Delta f(x^{(k)})^\top \Delta x \\ \text{s.t.} \quad & h_j(x^{(k)}) + \nabla h_j(x^{(k)})^\top \Delta x = 0 \qquad \forall j = 1, \dots, p. \end{aligned}$$

This directly motivates to consider

$$\begin{aligned} \min_{\Delta x \in \mathbb{R}^n} \quad & \frac{1}{2} \Delta x^\top B_k \Delta x + \Delta f(x^{(k)})^\top \Delta x \\ \text{s.t.} \quad & h_j(x^{(k)}) + \nabla h_j(x^{(k)})^\top \Delta x = 0 \qquad \forall j = 1, \dots, p \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^\top \Delta x \leq 0 \qquad \forall i = 1, \dots, m \end{aligned} \tag{QP_k}$$

to obtain the update

$$x^{(k+1)} \leftarrow x^{(k)} + \Delta x^{(k)}$$

in order to solve (PC), where $\Delta x^{(k)}$ is a solution of (QP_k). Note that the objective function of (QP_k) is quadratic approximation of f and that the constraints are linear approximations to the constraints of (PC). To determine a solution of (PC), this approach requires the sequential solution of (QP_k), i.e. a (QP). Hence, it is called SQP-method.

Algorithm 3.4.3: Local SQP-method

Input: $x^{(0)} \in \mathbb{R}^n,\ \lambda^{(0)} \in \mathbb{R}^m,\ \mu^{(0)} \in \mathbb{R}^p,\ \varepsilon \geq 0,\ k \leftarrow 0$

while $\|\nabla_x L(x^{(k)}, \lambda^{(k)}, \mu^{(k)})\| > \varepsilon$ **do**

$$\begin{aligned} \Delta x^{(k)} \leftarrow \operatorname*{argmin}_{\Delta x \in \mathbb{R}^n} \ & \frac{1}{2} \Delta x^\top B_k \Delta x + \Delta f(x^{(k)})^\top \Delta x \\ \text{s.t.} \quad & h_j(x^{(k)}) + \nabla h_j(x^{(k)})^\top \Delta x = 0 \qquad \forall j = 1, ..., p \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^\top \Delta x \leq 0 \qquad \forall i = 1, ..., m \end{aligned}$$

$x^{(k+1)} \leftarrow x^{(k)} + \Delta x^{(k)}$

$k \leftarrow k + 1$

end

Since the (local) SQP-method is based on the LN-method (which is based on Newtons-method) we can expect local superlinear/quadratic convergence to a KKT-point of (PC) under appropriate assumptions.

Theorem 3.4.4
Let (x^*, λ^*, μ^*) be a KKT-point of (PC) and assume

1. $g_i(x^*) + \lambda_i^* \neq 0 \quad \forall i = 1, ..., m$ *(strict complementary)*
2. x^* *meets the LICQ*
3. $\nabla^2_{xx} L(x^*, \lambda^*, \mu^*) > 0$ *on* $C_X(x^*, \lambda^*)$ *(SSOC)*

Then there exists $\varepsilon > 0$ such that for all $(x^{(0)}, \lambda^{(0)}, \mu^{(0)}) \in B_\varepsilon(x^*, \lambda^*, \mu^*)$ the iterates of **??** *3.4.3 fulfil:*

1. $(x^{(k)}, \lambda^{(k)}, \mu^{(k)})_{k \in \mathbb{N}_0}$ *is well-defined*
2. $(x^{(k)}, \lambda^{(k)}, \mu^{(k)}) \to (x^*, \lambda^*, \mu^*)$ *superlinearly*
3. $(x^{(k)}, \lambda^{(k)}, \mu^{(k)}) \to (x^*, \lambda^*, \mu^*)$ *quadratically, if $\nabla^2 f, \nabla g_i$ $(i = 1, ..., m)$, $\nabla^2 h_j$ $(j = 1, ..., p)$ are locally Lipschitz.*

Proof. Cf. Geiger and Kanzow (2013, Theorem 5.31.). □

3.4.3 Global SQP-Method

In analogy to the quasi-Newton method we want to achieve a globally convergent SQP-method by introducing an Armijo-based step length. We will show that, under appropriate assumptions, the solution $\Delta x^{(k)}$ of (QP_k) is a descent direction of the exact $l1$-penalty function $P_1(\cdot, \alpha)$. Therefore, we could find a step length by applying the backtracking line search to $P_1(\cdot, \alpha)$. However, the problem is that $P_1(\cdot, \alpha)$ is not differentiable everywhere.

We recall the definition of directional derivates of a function $\varphi : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathbb{R}^n$ in direction $d \in \mathbb{R}^n$ as

$$\varphi'(x; d) := \lim_{t \searrow 0} \frac{1}{t} [\varphi(x + td) - \varphi(x)] \ .$$

Lemma 3.4.5
For $x, d \in \mathbb{R}^n$ and fixed $\alpha > 0$ it holds

$$\begin{aligned} P_1'(x, \alpha; d) = \nabla f(x)^\top d + \alpha \sum_{i:\ g_i(x) > 0} \nabla g_i(x)^\top d + \alpha \sum_{i:\ g_i(x) \leq 0} \max\{0, \nabla g_i(x)^\top d\} \\ + \alpha \sum_{j:\ h_j(x) > 0} \nabla h_j(x)^\top d - \alpha \sum_{j:\ h_j(x) < 0} \nabla h_j(x)^\top d + \alpha \sum_{j:\ h_j(x) = 0} |\nabla h_j(x)^\top d| \ . \end{aligned}$$

We now turn back to the SQP-method and consider the quadratic subproblem (QP_k) for given $x^{(k)}$ and symmetric positive definite matrix B_k. If $\Delta x^{(k)}$ is a solution of (QP_k), there exist Lagrangian multipliers $\lambda^{(k+1)}$, $\mu^{(k+1)}$ such that KKT-conditions hold, i.e.

1. $\nabla f(x^{(k)}) + B_k \Delta x^{(k)} + \sum \lambda_i^{(k+1)} \nabla g_i(x^{(k)}) + \sum_{j=1}^{p} \mu_j^{(k+1)} \nabla h_j(x^{(k)}) = 0$
2. $h_j(x^{(k)}) + \nabla h_j(x^{(k)})^\top \Delta x^{(k)} \leq 0 \qquad \forall j = 1, ..., p$
3. $\lambda_i^{(k+1)} \geq 0 \qquad \forall i = 1, ..., m$
4. $g_i(x^{(k)}) + \nabla g_i(x^{(k)}) \Delta x^{(k)} \leq 0 \qquad \forall i = 1, ..., m.$

This shows:

Lemma 3.4.6
If $\Delta x^{(k)} = 0$ is a solution of (QP_k), then $(x^{(k)}, \lambda^{(k+1)}, \mu^{(k+1)})$ is already a KKT-point of the original problem (PC).

Theorem 3.4.7
Let $\Delta x^{(k)} \neq 0$ be a solution of (QP_k) with B_k symmetric positive definite and let $\lambda^{(k+1)}$, $\mu^{(k+1)}$ denote the related Lagrangian multipliers. For

$$\alpha \geq \max\{\lambda_1^{(k+1)}, ..., \lambda_m^{(k+1)}, |\mu_1^{(k+1)}|, ..., |\mu_p^{(k+1)}|\}$$

it holds

$$P_1'(x^{(k)}, \alpha; \Delta x^{(k)}) \leq -(\Delta x^{(k)})^\top B_k \Delta x^{(k)} < 0,$$

i.e. $\Delta x^{(k)}$ is a descent direction of $P_1(\cdot, \alpha)$ in $x^{(k)}$.

Proof.
Due to the KKT-conditions it holds

$$\begin{aligned} \lambda_i^{(k+1)}(g_i(x^{(k)}) + \nabla g_i(x^{(k)})^\top \Delta x^{(k)} &= 0 \\ \nabla h_j(x^{(k)})^\top \Delta x^{(k)} &= -h_j(x^{(k)}) \\ \nabla g_i(x^{(k)})^\top \Delta x^{(k)} &= -g_i(x^{(k)}). \end{aligned}$$

With Lemma 3.4.5 we have

$$
\begin{aligned}
P_1'(x^{(k)}, \alpha; \Delta x^{(k)}) = \; & \nabla f(x^{(k)})^\top \Delta x^{(k)} + \alpha \sum_{i:\ g_i(x^{(k)})>0} \nabla g_i(x^{(k)})^\top \Delta x^{(k)} \\
& + \underbrace{\alpha \sum_{i:\ g_i(x^{(k)})\le 0} \max\{0, \underbrace{\nabla g_i(x^{(k)})^\top \Delta x^{(k)}}_{\le -g_i(x^{(k)})=0}\}}_{=0} \\
& + \alpha \sum_{j:\ h_j(x^{(k)})>0} \nabla \underbrace{h_j(x^{(k)})^\top \Delta x^{(k)}}_{=-h_j(x^{(k)})} - \alpha \sum_{j:\ h_j(x^{(k)})<0} \nabla \underbrace{h_j(x^{(k)})^\top \Delta x^{(k)}}_{=-h_j(x^{(k)})} \\
& + \underbrace{\alpha \sum_{j:\ h_j(x^{(k)})=0} \underbrace{|\nabla h_j(x^{(k)})^\top \Delta x^{(k)}|}_{=|-h_j(x^{(k)})|=0}}_{=0} \\
= \; & \nabla f(x^{(k)})^\top \Delta x^{(k)} + \alpha \sum_{i:\ g_i(x)>0} \underbrace{\nabla g_i(x)^\top \Delta x^{(k)}}_{\le -g_i(x^{(k)})} \\
& - \alpha \sum_{j:\ h_j(x^{(k)})>0} h_j(x^{(k)}) + \alpha \sum_{j:\ h_j(x^{(k)})<0} h_j(x^{(k)}) \\
\le \; & \nabla f(x^{(k)})^\top \Delta x^{(k)} - \sum_{j:\ g_i(x^{(k)})>0} \alpha g_i(x^{(k)}) - \sum_{j:\ h_j(x^{(k)})>0} \alpha h_j(x^{(k)}) \\
& + \sum_{j:\ h_j(x^{(k)})<0} \alpha h_j(x^{(k)}) + \sum_{i=1}^{m} \lambda_i^{(k+1)} (g_i(x^{(k)}) + \nabla g_i(x^{(k)})^\top \Delta x^{(k)}) \ . \qquad (*)
\end{aligned}
$$

Further, by the KKT-conditions it holds

$$
\begin{aligned}
\nabla f(x^{(k)})^\top \Delta x^{(k)} &= -(\Delta x^{(k)})^\top B_k \Delta x^{(k)} - \sum_{i=1}^{m} \lambda_i^{(k+1)} \nabla g_i(x^{(k)})^\top \Delta x^{(k)} - \sum_{j=1}^{p} \mu_j^{(k+1)} \underbrace{\nabla h_j(x^{(k)})^\top \Delta x^{(k)}}_{=-h_j(x^{(k)})} \\
&= -(\Delta x^{(k)})^\top B_k \Delta x^{(k)} - \sum_{i=1}^{m} \lambda_i^{(k+1)} \nabla g_i(x^{(k)})^\top \Delta x^{(k)} + \sum_{j=1}^{p} \mu_j^{(k+1)} h_j(x^{(k)}) \ .
\end{aligned}
$$

Plugging this into (*) yields

$$
\begin{aligned}
P_1'(x^{(k)}, \alpha; \Delta x^{(k)}) \leq & -(\Delta x^{(k)})^\top B_k \Delta x^{(k)} - \sum_{i=1}^{m} \lambda_i^{(k+1)} \nabla g_i(x^{(k)})^\top \Delta x^{(k)} + \sum_{j=1}^{p} \mu_j^{(k+1)} h_j(x^{(k)}) \\
& - \sum_{i:\ g_i(x^{(k)})>0} \alpha g_i(x^{(k)}) - \sum_{j:\ h_j(x^{(k)})>0} \alpha h_j(x^{(k)}) + \sum_{j:\ h_j(x^{(k)})<0} \alpha h_j(x^{(k)}) \\
& + \sum_{i=1}^{m} \lambda_i^{(k+1)} g_i(x^{(k)}) + \sum_{i=1}^{m} \lambda_i^{(k+1)} \nabla g_i(x^{(k)})^\top \Delta x^{(k)} \\
= & -(\Delta x^{(k)})^\top B_k \Delta x^{(k)} + \sum_{i:\ g_i(x^{(k)})>0} \lambda_i^{(k+1)} g_i(x^{(k)}) - \sum_{i:\ g_i(x^{(k)})>0} \alpha g_i(x^{(k)}) \\
& + \underbrace{\sum_{i:\ g_i(x^{(k)})\leq 0} \lambda_i^{(k+1)} g_i(x^{(k)})}_{\leq 0} - \sum_{j:\ h_j(x^{(k)})>0} \alpha h_j(x^{(k)}) \\
& + \sum_{j:\ h_j(x^{(k)})>0} \mu_j^{(k+1)} h_j(x^{(k)}) + \sum_{j:\ h_j(x^{(k)})<0} \alpha h_j(x^{(k)}) \\
& + \sum_{j:\ h_j(x^{(k)})<0} \mu_j^{(k+1)} h_j(x^{(k)}) \\
\leq & -(\Delta x^{(k)})^\top B_k \Delta x^{(k)} + \sum_{i:\ g_i(x^{(k)})>0} \underbrace{(\lambda_i^{(k+1)} - \alpha)}_{\leq 0} g_i(x^{(k)}) \\
& + \sum_{j:\ h_j(x^{(k)})>0} \underbrace{\underbrace{(\mu_j^{(k+1)} - \alpha)}_{\leq 0} h_j(x^{(k)})}_{\leq 0} + \sum_{j:\ h_j(x^{(k)})<0} \underbrace{\underbrace{(\alpha - \mu_j^{(k+1)})}_{\leq 0} h_j(x^{(k)})}_{\leq 0} \\
\overset{\alpha > \max\{\lambda_i, |\mu_j|\}}{\leq} & -(\Delta x^{(k)})^\top B_k \Delta x^{(k)}
\end{aligned}
$$

□

By means of this theorem and an adequate step size rule we obtain a global convergent SQP-method:

Algorithm 3.4.8: Global SQP-method

Input: $(x^{(0)}, \lambda^{(0)}, \mu^{(0)}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$, $B_o \in \mathbb{R}^{n \times n}$ symmetric positive definite, $\alpha > 0$, $k \leftarrow 0$, $\varepsilon \geq 0$

while $\|\Delta x^{(k)}\| > \varepsilon$ **do**
- $\Delta x^{(k)} \leftarrow$ argmin (QP_k)
- (with $\lambda^{(k+1)}, \mu^{(k+1)}$ related Lagrangian multipliers)
- t_k via backtracking line search for $P_1(\cdot, \alpha)$ with descent direction $\Delta x^{(k)}$
- $x^{(k+1)} \leftarrow x^{(k)} + t_k \Delta x^{(k)}$
- $B_{k+1} \leftarrow$ update of B_k that stays symmetric positive definite (e.g. BFGS)
- $k \leftarrow k + 1$

end

Remark 3.4.9

1. *In Algorithm 3.4.8 we choose α „sufficiently large“. A more practical approach is to update*
$$\alpha_{k+1} \leftarrow \max\{\alpha_k, \max\{\alpha_1^{(k+1)}, ..., \alpha_m^{(k+1)}, |\mu_1^{(k+1)}|, ..., |\mu_p^{(k+1)}|\} + c\}$$
for some $c > 0$.

2. *As we already know, the choice $B_k = \nabla^2_{xx} L(x^{(k)}, \lambda^{(k)}, \mu^{(k)})$ can cause problems since it might not be positive definite and requires the computation of $\nabla^2 f$, $\nabla^2 g_i$, and $\nabla^2 h_j$. We can use the already introduced BFGS-updates. However, Powell (1978) proposed a damped BFGS-update:*
Let
$$\begin{aligned} s^{(k)} &= x^{(k+1)} - x^{(k)}, \\ y^{(k)} &= \nabla_X L(x^{(k+1)}, \lambda^{(k)}, \mu^{(k)}) - \nabla_X L(x^{(k)}, \lambda^{(k)}, \mu^{(k)}) \end{aligned}$$
and
$$\eta^{(k)} := \theta_k y^{(k)} + (1 - \theta_k) B_k s^{(k)},$$
where
$$\theta_k := \begin{cases} 1, & if (s^{(k)})^\top y^{(k)} \geq 0.2 (s^{(k)})^\top B_k s^{(k)} \\ 0.8 \dfrac{(s^{(k)})^\top B_k s^{(k)}}{(s^{(k)})^\top y^{(k)}}, & else \end{cases}.$$
The damped BFGS-update reads
$$B_{k+1} \leftarrow B_k + \frac{(\eta^{(k)})^\top \eta^{(k)}}{(s^{(k)})^\top \eta^{(k)}} - \frac{B_k s^{(k)} (s^{(k)})^\top B_k^\top}{(s^{(k)})^\top B_k s^{(k)}}.$$
That is, $y^{(k)}$ in BFGS-update is replaced by $\eta^{(k)}$ (which equals $y^{(k)}$ if $\theta_k = 1$). Hence, B_{k+1} is an interpolation between B_k and B_{k+1}^{BFGS}.

3. *As for the local SQP method we expect superlinear/quadratic convergence for the global version. Therefore we need $t_k = 1$ for all k „sufficiently large“. For some problems, however, it might happen that $t_k < 1$ for all k (so called Maratos effect). In this case one can modify the linear and quadratic approximations of f, g, and h in a certain way to ensure $t_k = 1$ for k „large enough“. Those variants are called modified SQP-methods (cf. Geiger and Kanzow, 2013, Chapter 5.5.6-5.5.8).*

Bibliography

Alt, W. (2013). *Nichtlineare Optimierung: Eine Einführung in Theorie, Verfahren und Anwendungen*. Springer.

Geiger, C. and Kanzow, C. (2013). *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer.

Sun, W. and Yuan, Y.-X. (2006). *Optimization theory and methods: nonlinear programming*. Springer.