Optimization for Engineers

– Summer Term 2022–

Dr. Johannes Hild

May 17, 2022

Contents

The module Optimization for Engineers

Module Contents

- Analysis for optimization of smooth, real-valued functions.
- Algorithms for optimization of smooth, real-valued functions.
- Methods for optimization of noisy functions.

Competences

For the lecture we aim at the following goals:

- Get an overview of optimization problem classes and solution strategies.
- Get a feeling for solvability and conditions of optimality.
- Get experience in crafting algorithms out of tools and subroutines.
- Be able to solve the benchmark problem.

The Benchmark Problem

The benchmark problem is defined as follows:

Problem 0.1 (Benchmark Problem)**:**

$$\begin{aligned} \textit{minimize} \quad & f(u,v,w) = \alpha(v+1)u^2 + \exp(\beta w + 1)v^2 + \gamma\sqrt{u+1}w^2 + \psi(u,v,w) \\ \textit{such that} \quad & x = (u,v,w)^\top \in \Omega_\square := [0,8] \times [-4,4] \times [-1,1] \\ \textit{and} \quad & h(u,v,w) = (u-4)^2 + v^2 + w^2 - 9 = 0. \end{aligned}$$

The parameters α, β and γ are found by solving an approximation problem, $\psi : \Omega_\square \to [-1.0e-3, 1.0e-3]$ is random noise on a small scale.

The lecture aims to provide all necessary competences to solve problems like the benchmark problem. In the programming homeworks we develop all necessary algorithms step by step.

If you want 5 ECTS...

- Pass the mandatory e-learning tests and then pass the final exam (online, scheduled in Juli, resit exam is scheduled in October).
- Follow the lecture (3 SWS), do the home exercises in the lecture notes, solve previous exams.

If you want additional 2.5 ECTS...

- Complete the mandatory programming homeworks.
- In addition you need to pass the programming homeworks multiple choice exam (online, scheduled in Juli).
- Follow the programming discussion (1 SWS).
- Programming Homework is not content of the final exam.

Literature

- Kelley, C. T.: Iterative Methods for Optimization. Frontiers in Applied Mathematics 18, SIAM Philadelphia 1999.
- Boyd, S. and Vandenberghe, L.: Convex Optimization. Cambridge University Press.
- Beck, A.: Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB. SIAM 2014.
- Luenberger, D. G. and Ye, Y.: Linear and Nonlinear Programming. Springer 2008.

1 Fundamental Definitions

In this chapter we introduce the mathematical formulation of optimization problems and classify types of functions and problems. We start with the general definition of a valid optimization problem in $\mathbb{R}^n$:

1.1 Defining Optimization Problems

Problem 1.1 (General Optimization Problem)**:**
Let $\Omega \subseteq \mathbb{R}^n$ be a nonempty set. Let the function $f : \Omega \to \mathbb{R}$ be bounded from below. The problem of finding $x_* \in \Omega$ such that

$$f(x_*) = \min_{x \in \Omega} f(x)$$

is denoted as

$$\begin{aligned} \textit{minimize} \quad & f(x) \\ \textit{such that} \quad & x \in \Omega \end{aligned} \tag{1.1}$$

We call

- f the **objective** (function),
- Ω the **feasible set** (or admissible set),
- $x \in \Omega$ a **feasible point**,
- $x_* \in \Omega$ a **minimizing point** (or minimizer) and
- $f(x_*)$ the **minimal value**.

1.2 Objective Functions

For objective functions $f : \Omega \to \mathbb{R}$ the following properties are of interest:

Definition 1.2 (Function Properties)**:**
Let $\emptyset \neq \Omega \subseteq \mathbb{R}^n$. We call the function $f : \Omega \to \mathbb{R}$

- ***bounded from below**, if there is a lower bound $f_L \in \mathbb{R}$ such that $f(x) \geq f_L$ for all $x \in \Omega$. The **infimum of f** is the biggest existing lower bound.*
- ***coercive** or radially unbounded, if $\Omega = \mathbb{R}^n$ and $\lim ||x|| \to \infty$ always implies $\lim f(x) \to \infty$.*

- ***convex on*** Ω*, if* Ω *is a convex set and for all* $x,\ y \in \Omega$ *and all* $\lambda \in [0,1]$ *holds:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y). \tag{1.2}$$

If especially

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y) \quad \textit{for all} \quad \lambda \in (0,1) \tag{1.3}$$

holds for $x \neq y$*, the function is called* ***strictly convex on*** Ω*. If in addition there is a* $\varepsilon > 0$ *such that for all* $x,\ y \in \Omega,\ x \neq y$ *and all* $\lambda \in (0,1)$ *holds*

$$f(\lambda x + (1-\lambda)y) + \varepsilon(1-\lambda)\lambda||x-y||^2 < \lambda f(x) + (1-\lambda) f(y), \tag{1.4}$$

we call f ***uniformly (or strongly) convex on*** Ω*.*

The **smoothness of objective functions** is also highly important for both theory and application. We distinguish the following smoothness properties:

- **Discontinuous** objectives and noisy objectives: Require algorithms for noisy functions.
- **Lipschitz-continuous** objectives: Some existence and convergence theorems hold and algorithms with approximate gradients can be applied.
- **Continuously differentiable** objectives: More existence and convergence theorems hold and algorithms with exact gradients can be applied.
- **Twice continuously differentiable** objectives: Many existence and convergence theorems hold and algorithms with exact Hessians can be applied.

Depending on the objective f we distinguish subtypes of optimization problems

- $f(x) = c^\top x$, $c \in \mathbb{R}^n$ (**linear** objective, well-posed only with bounded Ω),
- $f(x) = \frac{1}{2} x^\top A x - b^\top x$, $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ (**quadratic** objective),
- $f(x) = \frac{1}{2}\sum_{j=1}^m r_j(x)^2$, $r_j : \Omega \to \mathbb{R}$ (**least squares** objective),
- $f(x)$ is convex on Ω (**convex** objective),
- $f(x)$ is a nonlinear objective in any other way (**generally nonlinear** objective).

Example 1.3:
A) The generally nonlinear objective $f : \mathbb{R}^2 \to \mathbb{R}$ *with*

$$f(u,v) = u^2\sqrt{(1+v^2)}$$

is bounded from below with $\inf f = 0$ *and coercive, but not convex on* $\mathbb{R}^2$*. It is twice continuously differentiable.*

B) The twice continuously differentiable $f(x) = x^2$ *with* $\inf f = 0$ *is coercive and strictly convex on* $\mathbb{R}$*:*

$$
\begin{aligned}
f(\lambda x + (1-\lambda)y) &= (\lambda x + (1-\lambda)y)^2 \\
&= \lambda^2 x^2 + 2\lambda(1-\lambda)xy + (1-\lambda)^2 y^2 - (\lambda x^2 + (1-\lambda)y^2) + (\lambda x^2 + (1-\lambda)y^2) \\
&= -\lambda(1-\lambda)x^2 + \lambda(1-\lambda)2xy - \lambda(1-\lambda)y^2 + (\lambda x^2 + (1-\lambda)y^2) \\
&= -\lambda(1-\lambda)(x-y)^2 + (\lambda x^2 + (1-\lambda)y^2) < \lambda x^2 + (1-\lambda)y^2 = \lambda f(x) + (1-\lambda)f(y)
\end{aligned}
$$

C) The twice continuously differentiable function $f(x) = \exp(x)$ *with* $\inf f = 0$ *is not coercive, but strictly convex on* $\mathbb{R}$*. It is uniformly convex on every bounded interval* $[a,b] \subset \mathbb{R}$ *but not on* $\mathbb{R}$ *itself.*

1.3 Feasible Sets

For feasible sets $\Omega \subseteq \mathbb{R}^n$ the following properties are of interest:

- $\Omega \neq \emptyset$: Non-emptiness is obviously required for solutions to exist.
- Ω is unbounded or bounded.
- Ω is open and/or closed.
- Ω is **compact** $\Leftrightarrow$ Ω **is closed and bounded**: All sequences have converging subsequences.

Depending on Ω we distinguish between

- $\Omega = \mathbb{R}^n$ (**unconstrained** or unrestricted problem),
- $\Omega = \Omega_\circ = \{x \in \mathbb{R}^n : ||x - x_*|| < \varepsilon\}$ (**open ball environment**),
- $\Omega = \Omega_\Box = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$ (**box constraints**), where the lower bounds $\{a_i\}_{i=1}^n$ and upper bounds $\{b_i\}_{i=1}^n$ should satisfy

$$-\infty < a_i < b_i < \infty \quad \text{for all} \quad i = 1, \ldots, n. \tag{1.5}$$

- $\Omega = \{x \in \mathbb{R}^n : Mx - c = 0 \text{ with } M \in \mathbb{R}^{m \times n}, c \in \mathbb{R}^m\}$ (**affine linear equality constraints**),
- $\Omega = \{x \in \mathbb{R}^n : h(x) = 0 \text{ with } h : \mathbb{R}^n \to \mathbb{R}^m\}$ (**equality constraints**),
- $\Omega = \{x \in \mathbb{R}^n : g(x) \preceq 0 \text{ with } g : \mathbb{R}^n \to \mathbb{R}^m\}$ (**inequality constraints**),
- $\Omega \subseteq \mathbb{Z}^n$ (**integer constraints**).

Note: With the expression $a \preceq b$ we say that each component of the vector a must be smaller than or equal to the corresponding component of b.

Example 1.4:
We can easily verify:

- $\mathbb{R}^n$ *is unbounded and open (and also closed by definition).*
- *Box constraints are compact by definition.*
- *Affine linear equality constraints consisting of more than one element are closed but unbounded.*

Also we prefer Ω to be a convex set:

Definition 1.5 (Convex Sets)**:**
A ***set*** $\Omega \subseteq \mathbb{R}^n$ *is called* ***convex****, if for all* x*,* $y \in \Omega$ *and all* $\lambda \in [0, 1]$ *holds:*

$$\lambda x + (1 - \lambda)y \in \Omega. \tag{1.6}$$

Lemma 1.6 (Examples for Convex Sets)**:**
In certain situations we can verify the convexity of Ω*:*

- $\Omega = \mathbb{R}^n$ *is convex.*
- $\Omega_\Box = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$ *is convex.*
- $\Omega_\circ = \{x \in \mathbb{R}^n : ||x - x_*|| < \varepsilon\}$ *(open ball environment) is convex.*
- $\Omega_H = \{x \in \mathbb{R}^n : Mx - c = 0 \text{ with } M \in \mathbb{R}^{m \times n}, c \in \mathbb{R}^m\}$ *is convex.*
- $\Omega_G = \{x \in \mathbb{R}^n : g(x) \preceq 0 \text{ with convex } g : \mathbb{R}^n \to \mathbb{R}^m\}$ *is convex.*
- *If two sets* $\Omega_a \subseteq \mathbb{R}^n$ *and* $\Omega_b \subseteq \mathbb{R}^n$ *are convex, so is the intersection* $\Omega_c = \Omega_a \cap \Omega_b$*.*

Proof. For each case we have to show that for all x, $y \in \Omega$ and all $\lambda \in [0, 1]$ holds

$$\lambda x + (1 - \lambda)y \in \Omega. \tag{1.7}$$

This is trivial for any vector space, especially $\mathbb{R}^n$.
In the case of box constraints consider the vector of lower bounds $a \in \mathbb{R}^n$ and the vector of upper bounds $b \in \mathbb{R}^n$. If $a \preceq x \preceq b$ and $a \preceq y \preceq b$ is true, then $\lambda a \preceq \lambda x \preceq \lambda b$ and $(1 - \lambda)a \preceq (1 - \lambda)y \preceq (1 - \lambda)b$ is true. And then

$$a \preceq \lambda x + (1 - \lambda)y \preceq b. \tag{1.8}$$

The convexity for the open ball environment and similar sets follows from the triangle inequality of the norm:

$$
\begin{aligned}
||\lambda(x-x_*)+(1-\lambda)(y-x_*)|| &\leq \lambda||x-x_*||+(1-\lambda)||y-x_*|| \\
&\leq \max(||x-x_*||,||y-x_*||)<\varepsilon
\end{aligned}
$$

In the case of affine linear equality constraints the solution set of $Mx-c=0$ is convex because

$$M(\lambda x+(1-\lambda)y)-c=\lambda(Mx-c)+(1-\lambda)(My-c)=0. \tag{1.9}$$

The solution set of $g(x)\preceq 0$ is convex because

$$g(\lambda x+(1-\lambda)y)\overbrace{\preceq}^{g\text{ convex}}\lambda g(x)+(1-\lambda)g(y)\preceq 0. \tag{1.10}$$

At last we prove that the intersection $\Omega_c=\Omega_a\cap\Omega_b$ is again convex. Consider $x,y\in\Omega_c$, then:

$$\lambda x+(1-\lambda)y\in\Omega_a \text{ and at the same time } \lambda x+(1-\lambda)y\in\Omega_b \tag{1.11}$$

so we conclude $\lambda x+(1-\lambda)y\in\Omega_c$. □

Example 1.7:
The set $\Omega_g=\{x=(u,v)^\top\in\mathbb{R}^2: g_1(u,v)=u^2-1\leq 0, g_2(u,v)=v^2-4\leq 0\}\}$ *is defined by two quadratic inequality constraints.*
The set $\Omega_\square=[-1,1]\times[-2,2]$ *is a set of box constraints, which is compact and convex. In fact both sets describe the same set of points.*
The set $\Omega_h=\{x=(u,v)^\top\in\mathbb{R}^2: h_1(u,v)=u^2-1=0, h_2(u,v)=v^2-4=0\}\}$ *is defined by two quadratic equality constraints and describes the same set of points as the integer constraint set* $\Omega_z=\{\begin{pmatrix}-1\\-2\end{pmatrix},\begin{pmatrix}1\\-2\end{pmatrix},\begin{pmatrix}-1\\2\end{pmatrix},\begin{pmatrix}1\\2\end{pmatrix}\}$.

Home Exercise 1.1 (Problem Specification)**:**
Consider the problem

$$
\begin{aligned}
&\textit{minimize} \quad f(u,v)=u^2-2uv+v^2 \\
\textit{s.t.} \quad &(u,v)^\top\in\Omega:=\{(u,v)^\top\in\mathbb{R}^2: 0\leq u\leq 2, -1\geq v\geq -2\}
\end{aligned}
$$

- ***a)*** *Decide if the objective is linear, quadratic and/or least squares. State c, A, b and* r_j *if applicable.*
- ***b)*** *Redesign* Ω *in terms of box constraints:* $\Omega=\Omega_\square=[a_1,b_1]\times[a_2,b_2]$.
- ***c)*** *Redesign* Ω *in terms of four inequality constraints* $g_r(u,v)\leq 0$*, r=1,2,3,4.*

1.4 Types of Solutions

We also distinguish between different qualities of solvability (**wellposedness**). A problem can have the following solution properties:

- **Existence** of local or global solutions.
- Local and global **uniqueness** of existing solutions.
- **Continuous dependence** of solutions with respect to f and Ω.

To decide **existence** we define the solutions we are looking for:

Definition 1.8 (Local Minimizing Points and Global Minimizing Points)**:**
We say that $x_* \in \Omega$ is a local minimizing point (or LMP) of $f : \Omega \to \mathbb{R}$, if there is a ε-neighborhood $\mathcal{B}_\varepsilon(x_*) := \{x \in \mathbb{R}^n : ||x - x_*|| < \varepsilon\}$ such that

$$f(x_*) \leq f(x) \quad \text{for all} \quad x \in \mathcal{B}_\varepsilon(x_*) \cap \Omega. \tag{1.12}$$

If in addition

$$f(x_*) = \inf_{x \in \Omega} f(x) \tag{1.13}$$

we say that x_* is a global minimizing point (or GMP) of f on Ω.

Remark:
Many theorems in this script use the ball environment
$\mathcal{B}_\varepsilon(x_*) := \{x \in \Omega : ||x - x_*|| < \varepsilon\}$ for sake of simplicity. Typically the proofs work with more general environments, too. $\square$

If we are interested in **local uniqueness**, we require the definition of strict solutions:

Definition 1.9 (Strict Local Minimizing Points and Strict Global Minimizing Points)**:**
We say that $x_* \in \Omega$ is a strict (LMP) of $f : \mathbb{R}^n \to \mathbb{R}$, if there is a ε-neighborhood $\mathcal{B}_\varepsilon(x_*)$ such that

$$f(x_*) < f(x) \quad \text{for all} \quad x \in (\mathcal{B}_\varepsilon(x_*) \cap \Omega) \setminus \{x_*\}. \tag{1.14}$$

If in addition

$$f(x_*) = \inf_{x \in \Omega} f(x)$$

we say that x_* is a strict (GMP) of f on Ω.

Example 1.10:
In Figure 1 a function $f : [-5, 5] \mapsto \mathbb{R}$ *is depicted with the following minimal points:*

- *At* $x = -4$ *we have a strict (LMP).*
- *At* $x = -1$ *we have a strict (LMP), which is also a strict (GMP) on* $[-5, 5]$.
- *The closed interval* $[1, 2]$ *consists of (nonstrict) (LMPs), which are also (GMPs) on* $[-5, 5]$.
- *The open interval* $(3, 4)$ *consists of (nonstrict) (LMPs).*

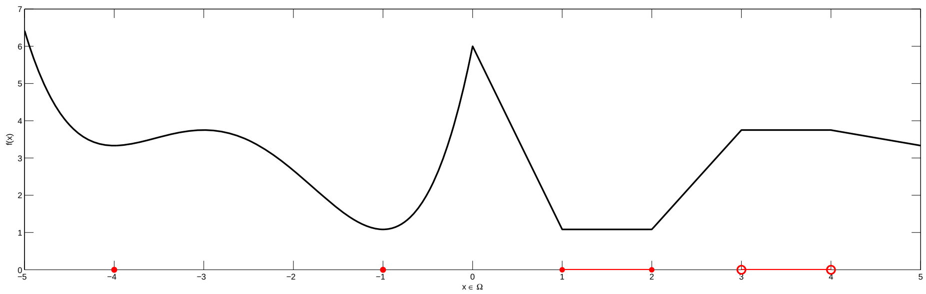

Figure 1: Function with piecewise smooth parts. (LMPs) are marked red.

We speak of **global uniqueness**, if there exists exactly one (LMP) which is also a (GMP) in Ω.

Remark:
The problem

$$\begin{aligned} \text{maximize} \quad & f(x), \\ \text{s.t.} \quad & x \in \Omega \end{aligned}$$

is equivalent to

$$\begin{aligned} \text{minimize} \quad & -f(x), \\ \text{s.t.} \quad & x \in \Omega \end{aligned}$$

and global or local solutions of minimizing $-f$ on Ω are corresponding solutions of maximizing f on Ω. $\square$

Home Exercise 1.2 (Convexity)**:**
Consider the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = 1 - \frac{1}{2}\exp(-(u^2+v^2)) \\ \textit{s.t.} \quad & (u,v)^\top \in \Omega := \{(u,v)^\top \in \mathbb{R}^2 : (u-1)^2 + v^2 \geq 4\} \end{aligned}$$

a) *Prove that Ω is not a convex set.*

b) *Prove that f is not a convex function on $\mathbb{R}^2$.*

c) *Estimate the infimum of f on Ω with the help of $(u-1)^2 + v^2 \geq 4$. State a (GMP) x_* such that $f(x_*) = \inf_\Omega f$*

Next we discuss **continuous dependency** in a very basic approach:

Definition 1.11 (Continuous Dependency)**:**
Let the objective f or the feasible set Ω depend on a real parameter $\alpha \in \mathbb{R}$. We say that a globally unique (GMP) $x_*(\alpha) \in \Omega$ is depending continuously on α, if the function $x_* : \mathbb{R} \to \Omega$ that maps α to its corresponding (GMP) is continuous.

Example 1.12:
An easy example for a not continuously depending solution shows up in the following problem:

$$\begin{aligned} \textit{minimize} \quad & f_\alpha(x) = -\alpha x, \alpha \in \mathbb{R} \setminus \{0\} \\ \textit{s.t.} \quad & x \in \Omega := [-1, 1] \end{aligned}$$

The unique (GMP) jumps depending on the sign of α: $x_* = \begin{cases} -1 & \textit{for} \quad \alpha < 0 \\ 1 & \textit{for} \quad \alpha > 0 \end{cases}$.
The mapping is not defined for $\alpha = 0$, because in this situation exists no unique (GMP). So the mapping $\alpha \mapsto x_*(\alpha)$ is not continuous and in consequence x_* does not depend continuously on α.

Continuous dependency is a good property in higher level optimization, where the goal is to tune your problem parameters in such a way that you can control the globally unique (GMP).

Home Exercise 1.3 (Solution types)**:**
Consider the problem

$$\begin{aligned} &\textit{minimize} \quad f(x) = (x-1)^2 \\ \textit{s.t.} \quad & x \in \Omega_\alpha := [-\alpha, \alpha] \textit{ with some parameter } \alpha > 0 \end{aligned}$$

a) *Find the (GMP) of f on $\mathbb{R}$.*

b) *Find the (GMP) of f on Ω_α for $\alpha < 1$ in dependence of α.*

c) *Decide if the (GMP) depends continuously on α for all $\alpha > 0$.*

2 Optimality Conditions

Optimality conditions help to decide if some $x_* \in \Omega$ is a (LMP). Because optimality conditions directly depend on Ω and the smoothness of the objective, we distinguish three cases:

- Ω is an open ball environment $\mathcal{B}_\varepsilon(x_*)$. This can be applied for $\Omega = \mathbb{R}^n$.
- Ω is a set of box constraints.
- Ω is defined by equality and inequality constraints.

2.1 Optimality Conditions for Open Environments

We start this section with a helpful lemma:

Lemma 2.1 (Applications of Taylor's Theorem)**:**
Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and $d \in \mathbb{R}^n$ a direction. Then there exists some $t \in (0,1)$ such that

$$f(x+d) = f(x) + \nabla f(x+td)^\top d. \tag{2.1}$$

If in addition f is twice continuously differentiable, then there exists some $t \in (0,1)$ such that

$$f(x+d) = f(x) + \nabla f(x)^\top d + \frac{1}{2} d^\top \nabla^2 f(x+td) d \tag{2.2}$$

and we also verify

$$\nabla f(x+d) = \nabla f(x) + \int_0^1 \nabla^2 f(x+td) d \, \mathrm{d}t. \tag{2.3}$$

Proof. Define $F(t) := f(x+td)$. Then first order Taylor expansion leads to

$$F(1) = F(0) + F'(t) = f(x) + \nabla f(x+td)^\top d \quad \text{for some} \quad t \in (0,1). \tag{2.4}$$

Second order Taylor expansion leads to equation (2.2). And realizing $\frac{\partial}{\partial t}(\nabla f(x+td)) = \nabla^2 f(x+td) d$ leads to equation (2.3). □

Using this lemma we can derive necessary conditions for x_* to be a (LMP).

Theorem 2.2 (First Order Necessary Condition for Open Environments)**:**
Let $x_* \in \mathbb{R}^n$ be a (LMP) of f on $\Omega_\circ = \mathcal{B}_\varepsilon(x_*)$ and let f be continuously differentiable in $\Omega_\circ$, then $\nabla f(x_*) = 0$.

Proof. Assume $\nabla f(x_*) \neq 0$. Define the direction $d := -\frac{\nabla f(x_*)}{||\nabla f(x_*)||}$, then $d^\top \nabla f(x_*) = -||\nabla f(x_*)|| < 0$. Because ∇f is continuous in $\Omega_\circ$ we know that $d^\top \nabla f(x) < 0$ holds for all x in the smaller environment $\mathcal{B}_\delta(x_*)$, $\delta \leq \varepsilon$. We can write this as

$$d^\top \nabla f(x_* + \lambda d) < 0 \quad \text{for all } \lambda \text{ with} \quad 0 < \lambda < \delta \tag{2.5}$$

We use Lemma 2.1: For each λ exists some $t \in (0,1)$ such that

$$f(x_* + \lambda d) = f(x_*) + \underbrace{\nabla f(x_* + t\lambda d)^\top \lambda d}_{<0}. \tag{2.6}$$

So this means

$$f(x_* + \lambda d) < f(x_*) \quad \text{for all } \lambda \text{ with} \quad 0 < \lambda < \delta \tag{2.7}$$

and therefore x_* is not a (LMP). $\square$

Points that satisfy $\nabla f(x_*) = 0$ are called **critical points**. Depending on the **definiteness** of the Hessian in $\mathcal{B}_\delta(x_*)$ critical points can be **valley points** (local minimizing points), **hill points** (local maximizing points) or **saddle points** (see Table 1). Theorem 2.2 is also called **stationarity condition for open environments**.

Definiteness of A	Definition	Eigenvalues of A	Critical Point
positive definite	$x^\top Ax > 0$ for all $x \in \mathbb{R}^n/\{0\}$	all positive	valley
positive semi-definite	$x^\top Ax \geq 0$ for all $x \in \mathbb{R}^n/\{0\}$	all positive or zero	unknown
negative semi-definite	$x^\top Ax \leq 0$ for all $x \in \mathbb{R}^n/\{0\}$	all negative or zero	unknown
negative definite	$x^\top Ax < 0$ for all $x \in \mathbb{R}^n/\{0\}$	all negative	hill
indefinite	none of above	mixed signs	saddle

Table 1: Definiteness of symmetric matrices $A \in \mathbb{R}^{n \times n}$

The following optimality conditions use the Hessian matrix:

Theorem 2.3 (Second Order Necessary Condition for Open Environments)**:**
Let $x_* \in \mathbb{R}^n$ be a (LMP) of f on $\Omega_\circ = \mathcal{B}_\varepsilon(x_*)$ and let f be twice continuously differentiable in $\Omega_\circ$, then $\nabla f(x_*) = 0$ and in addition $\nabla^2 f(x_*)$ is positive semi-definite.

Proof. For $\nabla f(x_*) = 0$ see above. Assume the Hessian $\nabla^2 f(x_*)$ is not positive semi-definite. Then there is a direction $d \in \mathbb{R}^n$ with $||d|| = 1$ such that $d^\top \nabla^2 f(x_*) d < 0$. As $\nabla^2 f$ is continuous in $\Omega_\circ$, in the smaller environment $\mathcal{B}_\delta(x_*)$ with $\delta \leq \varepsilon$ holds

$$d^\top \nabla^2 f(x_* + \lambda d) d < 0 \quad \text{for all } \lambda \text{ with} \quad 0 < \lambda < \delta \tag{2.8}$$

We apply again Lemma 2.1:

$$f(x_* + \lambda d) = f(x_*) + \underbrace{\nabla f(x_*)^\top}_{=0} \lambda d + \underbrace{\frac{1}{2}\lambda^2 d^\top \nabla^2 f(x_* + t\lambda d) d}_{<0} \tag{2.9}$$

for some $t \in (0,1)$. Again x_* is not a (LMP) because $f(x_* + \lambda d) < f(x_*)$. □

Theorem 2.4 (Second Order Sufficiency for Open Environments)**:**
For $x_* \in \mathbb{R}^n$ let f be twice continuously differentiable in a neighborhood $\Omega_\circ = \mathcal{B}_\varepsilon(x_*)$, let $\nabla f(x_*) = 0$ and let $\nabla^2 f(x_*)$ be positive definite. Then x_* is a strict (LMP) of f on $\Omega_\circ$.

Proof. As $\nabla^2 f$ is continuous in $\Omega_\circ$, in the smaller environment $\mathcal{B}_\delta(x_*)$, $\delta \leq \varepsilon$ the Hessian is still positive definite (semi-definiteness would not suffice). For any direction d with $0 < ||d|| < \delta$ holds:

$$f(x_* + d) = f(x_*) + \underbrace{\nabla f(x_*)^\top}_{=0} d + \underbrace{\frac{1}{2} d^\top \nabla^2 f(x_* + td) d}_{>0} \tag{2.10}$$

for some $t \in (0,1)$. This means $f(x_* + d) > f(x_*)$, i.e. the definition of (LMP). □

Example 2.5:
Let $\alpha \in \mathbb{R}$ be an unknown parameter. For the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = 20u^2 - 4uv + \alpha v^2 \\ \textit{s.t.} \quad & (u,v)^\top \in \mathbb{R}^2 \end{aligned}$$

use the optimality conditions to find (LMPs) in dependence of α.

We compute

$$\nabla f(u,v) = \begin{pmatrix} 40u - 4v \\ -4u + 2\alpha v \end{pmatrix} \qquad \textit{and} \qquad \nabla^2 f(u,v) = \begin{pmatrix} 40 & -4 \\ -4 & 2\alpha \end{pmatrix}$$

We search for the solution of $\nabla f(u_*, v_*) = (0,0)$ and get $(u_*, v_*) = (0,0)$ for every $\alpha \in \mathbb{R}$, and for $\alpha = \frac{1}{5}$ we get the line $(u_*, v_*) = \mu(1,10)$ with $\mu \in \mathbb{R}$. For these points the first order necessary condition is satisfied. Now we check definiteness of $\nabla^2 f(u_*, v_*)$:

*The Hessian of a smooth function is always **symmetric**, in addition we know:*

- *If $det(A) = 0$ then A is **indefinite** or **semi-definite** of any kind, eigenvalues λ_i must be checked.*
- *If $det(A) > 0$ and all leading principal minors $det(A_i) > 0$, then A is **positive definite (Sylvester's criterion)**.*
- *If $det(A) < 0$ then A is **indefinite or negative definite**. Check $-A$ for positive definiteness.*

In our case $det(\nabla^2 f(u_*, v_*)) = 80\alpha - 16$, and the first leading principal minor is $40 > 0$.
Case 1: For $\alpha > \frac{1}{5}$ the Hessian is s.p.d (symmetric positive definite) and in this case (u_*, v_*) is (LMP).
Case 2: For $\alpha = \frac{1}{5}$ the eigenvalues are $\lambda_1 = 0$ and $\lambda_2 = 40 + \frac{2}{5}$ and therefore the Hessian is s.p.s. (symmetric positive semi-definite). So any point $(u_*, v_*) = (\mu, 10\mu)$ could be a (LMP).
Case 3: For $\alpha < \frac{1}{5}$ the Hessian is not s.p.s (especially not s.p.d.), so (u_*, v_*) qualifies not as (LMP).

For Case 2 we have to compare $f(u_*, v_*)$ with $\inf f(u, v)$. The function values on the line $(u_*, v_*) = (\mu, 10\mu)$ are
$f(u_*, v_*) = 20\mu^2 - 40\mu^2 + \frac{100}{5}\mu^2 = 0$. With skill we realize:

$$f(u, v) = 20u^2 - 4uv + \frac{1}{5}v^2 = \left(\sqrt{20}u - \frac{1}{\sqrt{5}}v\right)^2 \geq 0$$

We conclude $f(u_*, v_*) = \inf f(u, v) = 0$ and therefore the complete line $(u_*, v_*) = (\mu, 10\mu)$ consists of nonstrict (GMPs).

Home Exercise 2.1 ((LMPs) and (GMPs))**:**
Consider the problem

$$\begin{aligned} &\textit{minimize} \quad f(u, v) = \max(u^6 + v^2, \alpha) \\ &\textit{s.t.} \quad (u, v)^\top \in \mathbb{R}^2 \quad \textit{with parameter} \quad \alpha \geq 0 \end{aligned}$$

- ***a)*** *Identify all points in $\mathbb{R}^2$, which satisfy the first and/or second order necessary optimality condition in dependence of α.*
- ***b)*** *Use the definition of (GMPs) to decide, which points in $\mathbb{R}^2$ are (GMPs). For which α exists a strict (GMP)?*
- ***c)*** *Now assume the restriction: $(u, v)^\top \in \Omega := \mathbb{Z}^2$. Prove that every point $(u, v)^\top \in \mathbb{Z}^2$ satisfies the definition of a strict (LMP) for f on $\mathbb{Z}^2$.*

Home Exercise 2.2 (Definiteness of Hessians):
Consider the function

$$f(u, v, w) = \exp(u + v) + \exp(u - v) + \sin(w)$$

a) *Compute the Hessian* $\nabla^2 f(u, v, w)$.

b) *Compute the eigenvalues of the Hessian* $\nabla^2 f(u, v, w)$.

c) *Decide at which points* $(u, v, w)^\top \in \mathbb{R}^3$ *the Hessian* $\nabla^2 f(u, v, w)$ *is positive definite.*

d) *State all (LMPs) of minimizing* $f(u, v, w)$ *s.t.* $(u, v, w)^\top \in \mathbb{R}^3$.

2.2 Stationarity for Convex Sets

The optimality conditions for open environments are not valid for closed sets. Especially not in cases, in which the candidate x_* is on the boundary of Ω. For example

$$\begin{aligned} \text{minimize} \quad & f(x) = \sqrt{x} \\ \text{s.t.} \quad & x \in [1, 2] \end{aligned}$$

is solved by $x_* = 1$, but $\nabla f(x = 1) = \frac{1}{2} \neq 0$ and $\nabla^2 f(x = 1) = -\frac{1}{4} < 0$.

To get proper optimality conditions we need a way to identify stationarity on the boundary of Ω and need a method to ignore information, that leads out of Ω. Convexity for Ω is required in this context. We start with the following lemma:

Lemma 2.6 (Basic 1D Stationarity):
Let $\phi : [0, b] \to \mathbb{R}$ *be continuously differentiable and* $b > 0$ *some upper bound. If* $\phi(t)$ *has a (LMP) at* $t = 0$, *then* $\phi'(0) \geq 0$.

Proof. For sufficiently small $h > 0$ we get $\phi(h) \geq \phi(0)$, because ϕ has a (LMP) at $t = 0$. Because ϕ is continuously differentiable, we can look at the right sided differential quotient:

$$\phi'(0) = \lim_{h \to 0} \frac{\phi(h) - \phi(0)}{h} \geq 0. \tag{2.11}$$

□

Then we introduce stationarity:

Definition 2.7 (Stationarity Condition)**:**
*Let Ω be convex and let $f : \Omega \to \mathbb{R}$ be continuously differentiable. A point $x_* \in \Omega$ is called **stationary**, if*

$$\nabla f(x_*)^\top (y - x_*) \geq 0 \quad \textit{for all} \quad y \in \Omega. \tag{2.12}$$

In consequence every (LMP) is stationary:

Theorem 2.8 (First Order Necessary Condition for Convex Sets)**:**
Let $x_* \in \Omega$ be a (LMP) of f on convex Ω and let f be continuously differentiable in $\mathcal{B}_\varepsilon(x_*)$, then x_* is stationary.

Proof. As Ω is convex: $x_* + t(y - x_*) \in \mathcal{B}_\varepsilon(x_*) \cap \Omega$ for all $y \in \mathcal{B}_\varepsilon(x_*) \cap \Omega$ and all $t \in [0, 1]$. We define

$$\begin{aligned} \text{minimize} \quad & \phi(t) := f(x_* + t(y - x_*)) \\ \text{s.t.} \quad & t \in [0, 1] \end{aligned}$$

The (not necessarily unique) (GMP) of this problem is $t_* = 0$. Because of Lemma 2.6, ϕ' must be positive or zero at the (GMP) and we get:

$$\phi'(t_*) \geq 0 \Rightarrow \nabla f(x_*)^\top (y - x_*) \geq 0$$

□

Checking stationarity using Definition 2.7 is tedious. A better mechanic for checking stationarity is the use of a projection mapping:

Definition 2.9 (Projection Mapping)**:**
*The projection mapping $P : \mathbb{R}^n \to \Omega$ denotes the **projection** of x into a convex set Ω such that $P(x)$ is closest to x:*

$$P(x) = y_* \quad \textit{with } y_* \textit{ minimizing} \quad ||y - x|| \quad \textit{s.t.} \quad y \in \Omega \tag{2.13}$$

Remark:
Obviously, if $x \in \Omega$, then $P(x) = x$. In addition the projection is unique because of the convexity of Ω: If there are y_1, y_2 with $||y_1 - x|| = ||y_2 - x||$, then $y_* = \frac{y_1 + y_2}{2}$ is in Ω and $||y_* - x||$ is the height of the isosceles triangle $\{||y_1 - x||, ||y_2 - x||, ||y_1 - y_2||\}$:

$$||y_* - x|| < ||y_1 - x|| = ||y_2 - x|| \quad \text{or} \quad y_1 = y_2 = y_* \tag{2.14}$$

□

Example 2.10 (Projection into Ball Constraints)**:**
We want to project the general point $x \in \mathbb{R}^n$ into the closed ball $\Omega_{\delta_k} := \{x \in \mathbb{R}^n : \quad ||x - x_k|| \leq \delta_k\}$. $P(x) = y_*$ is the (GMP) of

$$\textit{minimize} \quad f_x(y) = ||y - x|| \quad \textit{s.t.} \quad ||y - x_k|| \leq \delta_k \tag{2.15}$$

This is solved by

$$y_* = \begin{cases} x & \textit{if} \quad x \in \Omega_{\delta_k} \\ x_k + \frac{\delta_k}{||x - x_k||}(x - x_k) & \textit{else} \end{cases} \tag{2.16}$$

The second case holds because for all $z \in \Omega$ holds:

$$||x - x_k|| = ||x - z + z - x_k|| \leq ||x - z|| + ||z - x_k|| \leq ||x - z|| + \delta_k$$

but because x and x_k and $x_k + \frac{\delta_k}{||x - x_k||}(x - x_k)$ are all on the same connecting line, we also get:

$$||x - x_k|| = \delta_k + ||x - (x_k + \frac{\delta_k}{||x - x_k||}(x - x_k))||.$$

In combination we get:

$$||x - (x_k + \frac{\delta_k}{||x - x_k||}(x - x_k))|| \leq ||x - z||.$$

For $\Omega = \Omega_\square$ (**box constraints**) the i-th component of the projection is:

Definition 2.11 (Projection into Box Constraints)**:**

$$P(x)_i = \begin{cases} a_i & \textit{if} \quad x_i \leq a_i \\ x_i & \textit{if} \quad a_i < x_i < b_i \\ b_i & \textit{if} \quad x_i \geq b_i \end{cases} \tag{2.17}$$

Exercise 2.12:
For

$$a = \begin{pmatrix} 0 \\ 1 \\ 3 \\ 4 \end{pmatrix}, x = \begin{pmatrix} 3 \\ 0 \\ 4 \\ 6 \end{pmatrix}, b = \begin{pmatrix} 4 \\ 2 \\ 7 \\ 5 \end{pmatrix}$$

we get the projected point $P(x) = (3, 1, 4, 5)^\top$.

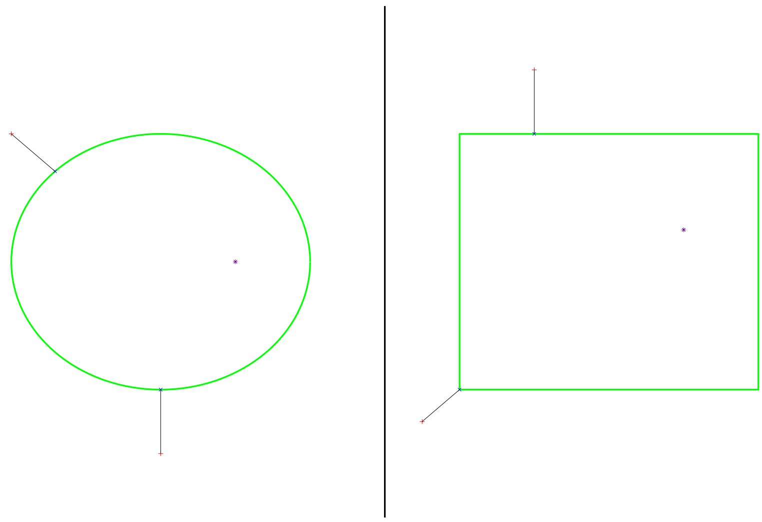

Figure 2: Projection of points into a disc and into box constraints

Now instead of checking Definition 2.7, the check of stationarity, which is required for a (LMP), can be done using the projection:

Theorem 2.13 (Stationarity Check with Projection)**:**
Let Ω be convex and let $f : \Omega \to \mathbb{R}$ be continuously differentiable. Then $x_* \in \Omega$ is stationary if and only if $x_* = P(x_* - t\nabla f(x_*))$ for all $t \geq 0$.

Proof. We define $x_{*+1}(t) := x_* - t\nabla f(x_*)$ and $P_{*+1}(t) := P(x_{*+1}(t))$ and look at the function

$$\phi(\tau) := \tfrac{1}{2}||\underbrace{(1-\tau)P_{*+1}(t) + \tau z}_{=:y} - x_{*+1}(t)||^2 \quad \text{with} \quad \tau \in [0,1] \tag{2.18}$$

and some $z \in \Omega$. This function is minimal at $\tau_* = 0$ because the projection is defined as:

$$||P_{*+1}(t) - x_{*+1}(t)|| \leq ||y - x_{*+1}(t)|| \quad \text{for all} \quad y \in \Omega \tag{2.19}$$

Using Lemma 2.6 we get:

$$0 \leq \phi'(0) = (P_{*+1}(t) - x_{*+1}(t))^\top (z - P_{*+1}(t)) \tag{2.20}$$
$$0 \leq (P_{*+1}(t) - x_*)^\top (z - P_{*+1}(t)) + t\nabla f(x_*)^\top (z - P_{*+1}(t)) \tag{2.21}$$

We now set $z = x_*$ and end up with

$$||P_{*+1}(t) - x_*||^2 \leq t\nabla f(x_*)^\top (x_* - P_{*+1}(t)) . \tag{2.22}$$

But if x_* is a stationary point, then

$$\nabla f(x_*)^\top (y - x_*) \geq 0 \quad \text{for all} \quad y \in \Omega, \tag{2.23}$$

especially for $y = P_{*+1}(t)$. In combination with (2.22) this means $\nabla f(x_*)^\top (x_* - P_{*+1}(t)) = 0$. We conclude $||P_{*+1}(t) - x_*||^2 = 0$, which again leads to $P(x_* - t\nabla f(x_*)) = x_*$ for all $t \geq 0$.

Assume now that x_* is not stationary (especially $\nabla f(x_*) \neq 0$) , then there are $y \in \Omega$ and $t > 0$ such that $\nabla f(x_*)^\top(y - x_*) = -t||\nabla f(x_*)||^2 < 0$ is true. Then y can be rewritten as $y = x_* - t\nabla f(x_*) = x_{*+1}(t) \neq x_*$. Because $y \in \Omega$ it follows $P_{*+1}(t) \neq x_*$. □

2.3 Second Order Optimality Conditions for Box Constraints

The stationarity conditions of the previous section work for general convex Ω, but the next conditions are not so easy to formulate for the general case. We therefore stick to box constraints. We know that box constraints are compact and convex. The projection is presented in Definition 2.11. The following mechanic allows us to trace, which boundaries of $\Omega_\square$ are touched by some point x:

Definition 2.14 (Active Index Sets for Box Constraints)**:**
Consider the set of ***box constraints***

$$\Omega_\square := [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$$

with lower and upper bounds satisfying $-\infty \prec a \prec b \prec \infty$. At a point $x \in \Omega_\square$ the ***active index set*** *$\mathcal{A}(x) \subseteq \{1, 2, \ldots, n\}$ is defined as:*

$$\mathcal{A}(x) := \{i \in \{1, \ldots, n\} |\ x_i = a_i \quad \textit{or} \quad x_i = b_i\} \tag{2.24}$$

Exercise 2.15:
For

$$a = \begin{pmatrix} 0 \\ \boldsymbol{1} \\ 3 \\ 4 \end{pmatrix}, x = \begin{pmatrix} 3 \\ \boldsymbol{1} \\ 4 \\ \boldsymbol{5} \end{pmatrix}, b = \begin{pmatrix} 4 \\ 2 \\ 7 \\ \boldsymbol{5} \end{pmatrix}$$

we get the set $\mathcal{A}(x) = \{2, 4\}$.

The **active index set of a point** x tells us, at which indexes a box constraint is **active**. We now want to delete all Hessian information for active box constraints, this is called **reduction**:

Definition 2.16 (Matrix Reduction)**:**
For the set $\Omega := \Omega_\Box$ *and a matrix* $B : \Omega \to \mathbb{R}^{n\times n}$, *the* ***reduced matrix*** B_Ω *is defined as*

$$\big(B_\Omega(x)\big)_{i,j} = \begin{cases} \delta_{i,j} & \text{if} \quad i \text{ or } j \in \mathcal{A}(x) \\ \big(B(x)\big)_{i,j} & \text{else} \end{cases} \tag{2.25}$$

where $\delta_{i,j}$ *is* ***Kronecker's delta****.*

Exercise 2.17:
For some optimization problem with box constraints let $\mathcal{A}(x_*) = \{2, 4\}$ *and the Hessian of* $f : \mathbb{R}^4 \to \mathbb{R}$ *is*

$$\nabla^2 f(x_*) = \begin{pmatrix} a & \boldsymbol{b} & c & \boldsymbol{d} \\ \boldsymbol{e} & \boldsymbol{f} & \boldsymbol{g} & \boldsymbol{h} \\ i & \boldsymbol{j} & k & \boldsymbol{l} \\ \boldsymbol{m} & \boldsymbol{n} & \boldsymbol{o} & \boldsymbol{p} \end{pmatrix}$$

then the reduced Hessian $\nabla^2_\Omega f$ *is:*

$$\nabla^2_\Omega f(x_*) = \begin{pmatrix} a & 0 & c & 0 \\ 0 & 1 & 0 & 0 \\ i & 0 & k & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Lemma 2.18 (Reduction of S.P.D. Matrices)**:**
If for the set $\Omega := \Omega_\Box$ *the matrix* $B : \Omega \to \mathbb{R}^{n\times n}$ *is s.p.d. for some* $x_* \in \Omega$, *so is the reduced matrix* B_Ω *at* x_*.

Proof. Assume that $|\mathcal{A}(x_*)| = M \leq n$ and without loss of generality the last M indices of x_* are active. Then the first $n - M$ leading principal minors of B_Ω are positive, because B is s.p.d (Sylvester's criterion). The remaining M leading principal minors are all equal to $\det((B_\Omega)_{n-M}) > 0$. $\Box$

Theorem 2.19 (Necessary Second Order Condition for Box Constraints)**:**
Let $\Omega_\Box$ *be box constraints and let* $f : \Omega_\Box \to \mathbb{R}$ *be twice continuously differentiable.*

Then if $x_* \in \Omega_\square$ is a (LMP) of f on $\Omega_\square$, then it is stationary and $\nabla^2_\Omega f(x_*)$ is a s.p.s. matrix.

Proof. Stationarity follows from Theorem 2.8. Assume now $|\mathcal{A}(x_*)| = M \leq n$. Without loss of generality the first M indices of x_* are active and we write $x_* = (\mu_1^*, \mu_2^*, \dots, \mu_M^*, \nu_1^*, \dots, \nu_{n-M}^*)$. Then the function

$$\phi(\nu) := f(\mu_*, \nu) \tag{2.26}$$

has an unconstrained (LMP) $\nu_* \in \mathbb{R}^{n-M}$ and $\nabla^2 \phi(\nu_*)$ is a s.p.s. matrix. This means

$$\nabla^2_\Omega f(x_*) = \begin{pmatrix} \mathbb{E} & 0 \\ 0 & \nabla^2 \phi(\nu_*) \end{pmatrix} \tag{2.27}$$

which is again a s.p.s. matrix (see Lemma 2.18). □

Now to get a proper sufficiency condition, we need to exclude all **degenerate cases**, in which a (LMP) is stationary and at the same time is a critical point of f on a subspace of $\mathbb{R}^n$.

Definition 2.20 (Nondegeneracy)**:**
*Let $\Omega_\square$ be box constraints and let $f : \Omega_\square \to \mathbb{R}$ be continuously differentiable. Then a point $x_* \in \Omega$ is a **nondegenerate** stationary point if it is **stationary** and*

$$\nabla f_i(x_*) \neq 0 \quad \textit{for all} \quad i \in \mathcal{A}(x_*) \tag{2.28}$$

*If in addition x_* is a (LMP), it is called a **nondegenerate (LMP)** or strict complementary (LMP).*

Theorem 2.21 (Second Order Sufficiency for Box Constraints)**:**
Let $\Omega_\square$ be box constraints and let $f : \Omega_\square \to \mathbb{R}$ be continuously differentiable. Let $x_* \in \Omega$ be a nondegenerate stationary point and let the reduced Hessian $\nabla^2_\Omega f(x_*)$ be s.p.d., then x_* is a nondegenerate (LMP).

Exercise 2.22:
For the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = \frac{1}{2} x^\top \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} x \\ \textit{s.t.} \quad & x = (u,v)^\top \in [1,2]^2 \end{aligned}$$

decide if one of the points $x_a = (2,2)^\top$ *or* $x_b = (2,1)^\top$ *is a (LMP).*

We compute

$$\nabla f(x) = \begin{pmatrix} -u \\ v \end{pmatrix} \qquad \textit{and} \qquad \nabla^2 f(x) = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

is indefinite.

We first check stationarity for x_a*:*

$$P(x_a - t\nabla f(x_a)) = P(\begin{pmatrix} 2+2t \\ 2-2t \end{pmatrix}) \neq \begin{pmatrix} 2 \\ 2 \end{pmatrix} \qquad \textit{for all} \quad t > 0, \tag{2.29}$$

so x_a *cannot be a (LMP).*

For x_b *we get:*

Stationarity:

$$P(x_b - t\nabla f(x_b)) = P(\begin{pmatrix} 2+2t \\ 1-t \end{pmatrix}) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} = x_b \quad \textit{for all} \quad t > 0 \quad \checkmark \tag{2.30}$$

Nondegeneracy:

$$A(x_b) = \{1,2\} \quad \textit{and} \quad (\nabla f(x_b))_1 = -2 \neq 0; \qquad (\nabla f(x_b))_2 = 1 \neq 0 \quad \checkmark \tag{2.31}$$

Reduced Hessian:

$$\nabla^2_\Omega f(x_b) = \mathbb{E} \quad \textit{is s.p.d.} \qquad \checkmark \tag{2.32}$$

In conclusion x_b *is a nondegenerate (LMP).*

Home Exercise 2.3 (Optimality Conditions for Box Constraints)**:**
Consider the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v,w) = v^2 - w(u-1)^2 \\ \textit{s.t.} \quad & x = (u,v,w)^\top \in \Omega_\square := [0,2]^3 \end{aligned}$$

Decide if the points $x_1 = (1,0,0)^\top$, $x_2 = (1,0,2)^\top$ *and* $x_3 = (0,0,2)^\top$ *are stationary or even nondegenerate stationary and determine the definiteness of the corresponding reduced Hessian. Decide if each of these points is, could be or is not a (LMP).*

Home Exercise 2.4 (Transformation of Ω to Box Constraints)**:**
Consider the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = u \\ \textit{s.t.} \quad & x = (u,v)^\top \in \Omega := \{(u,v)^\top \in \mathbb{R}^2 : \; u^2 + 4v^2 \le 4\} \end{aligned}$$

a) *Show that $x_* = (-2,0)^\top$ is stationary for this problem.*

b) *Rewrite this problem in terms of the transformation $(u,v)^\top = (r\cos(\phi), \frac{1}{2} r \sin(\phi))^\top$ with proper box constraints for r and ϕ.*

c) *Express $x_* = (-2,0)^\top$ in terms of $\tilde{x}_* := (r_*, \phi_*)^\top$ and show that $\tilde{x}_*$ is a nondegenerate (LMP) for the transformed problem.*

Home Exercise 2.5 (Construction of Projection)**:**
Consider the convex set

$$\begin{aligned} \Omega := \{(u,v) \in \mathbb{R}^2 : \; u \in [0,1] \quad \textit{and} \quad v \in [0, L(u)]\} \\ \textit{with line} \quad L : u \mapsto 2 - u \end{aligned}$$

a) *Assume the general point $x_0 = (u_0, v_0) \in \mathbb{R}^2$. Formulate the projection onto the line $P_L : \mathbb{R}^2 \to L$ by first finding the solution $u_* \in \mathbb{R}$ of the problem*

$$\textit{minimize} \quad g(u) := \frac{1}{2}\left((u-u_0)^2 + (2-u-v_0)^2\right) \qquad \textit{s.t.} \quad u \in \mathbb{R}.$$

in dependence of u_0 and v_0, then explicitly formulate $P_L : (u_0, v_0)^\top \mapsto (u_*, L(u_*))^\top$.

b) *State the projection of the following points into Ω: $x_1 = (0,1)^\top$, $x_2 = (1,2)^\top$ and $x_3 = (3,2)^\top$.*

c) *Show that $x^* = (0,0)^\top$ is stationary with respect to Ω and $f(u,v) = u + v$.*

2.4 Optimality Conditions for Equality and Inequality Constraints

In this section we analyze optimality conditions for Ω defined by equality constraints and inequality constraints, i.e. equations, which have to be satisfied for a point x_* to be feasible. Additional box constraints are possible, too. First we define the context:

Definition 2.23 (Optimization Problem with Equality and Inequality Constraints)**:**
For the problem

$$\begin{aligned} &\textit{minimize} \quad f(x) \\ \textit{s.t.} \quad &x \in \Omega := \left\{ \begin{matrix} & x \in \Omega_\square & \\ h_j(x) = 0 & \textit{for} & j = 1, \dots, m \\ g_r(x) \le 0 & \textit{for} & r = 1, \dots, s \end{matrix} \right\} \end{aligned}$$

*let the objective f, all **equality constraints** $h_j : \mathbb{R}^n \to \mathbb{R}$ and all **inequality constraints** $g_r : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable. For $x_* \in \Omega$ we also define the **index set of active inequality constraints**: $\mathcal{A}_g(x_*) := \{r = 1, \dots, s : g_r(x_*) = 0\}$.*

We can write these constraints also as vector valued functions, i.e. we collect all h_j in $h : \mathbb{R}^n \to \mathbb{R}^m$ with $h(x) = 0$ and all g_r in $g : \mathbb{R}^n \to \mathbb{R}^s$ with $g(x) \preceq 0$. Also, there is no need to define the index set of active equality constraints, because all of them have to be always active. If the box constraints, equality constraints and inequality constraints overlap each other too much, we get into trouble both in theory and in numerical methods. Therefore we require constraints to satisfy **constraint qualifications**. One of these qualifications is called (LICQ):

Definition 2.24 (Linear Independence Constraint Qualification (LICQ))**:**
Consider a problem from Definition 2.23. If at the (LMP) $x_* \in \Omega$ the set

$$C = \{e_i\}_{i \in \mathcal{A}_\square(x_*)} \cup \{\nabla h_j(x_*)\}_{j=1}^m \cup \{\nabla g_r(x_*)\}_{r \in \mathcal{A}_g(x_*)}$$

*is linearly independent, we say that the **Linear Independence Constraint Qualification (LICQ)** is satisfied.*

The index set $\mathcal{A}_\square(x_*)$ is defined in Definition 2.14. Other constraint qualifications are possible and can be found in the literature.

Exercise 2.25:
For the set $\Omega := \{(u, v)^\top \in \mathbb{R}^2, h(u, v) = u^2 + v^2 - 1 = 0, g(u, v) = u^2 + v - 1 \le 0\}$ we check the (LICQ) condition at all points $(u, v)^\top \in \Omega$: First we need to distinguish cases, in which $g(u, v)$ is active or not: If $u^2 + v - 1 < 0$, we get

$$C_a = \{\nabla h(u, v)\} = \{(2u, 2v)^\top\}$$

This does not contain a zero vector, because $(0, 0)^\top \notin \Omega$, and is linear independent for all points $(u, v)^\top \in \Omega$. If $u^2 + v - 1 = 0$, we get

$$C_b = \{\nabla h(u, v), \nabla g(u, v)\} = \{(2u, 2v)^\top, (2u, 1)^\top\} \overbrace{=}^{g=0} \{(2u, 2 - 2u^2))^\top, (2u, 1)^\top\}$$

This does not contain a zero vector, but for $(0,1)^\top \in \Omega$ the vectors are linear dependent. The LICQ does not hold for $(0,1)^\top \in \Omega$. We can treat this by separating Ω into two sets as follows: First we need to realize, that $\bar{\Omega} = \{(u,v)^\top \in \mathbb{R}^2, h(u,v) = u^2 + v^2 - 1 = 0, g(u,v) = v \leq 0\}$ contains the same points as $\Omega \setminus (0,1)^\top$ and is convex and satisfies (LICQ) at all its points. The isolated point $(0,1)^\top$ itself could be described as the (LICQ)-compatible set $\tilde{\Omega} = \{h_1(u,v) = u = 0, h_2(u,v) = v - 1 = 0\}$, but this has no practical value. In practical application one would simply evaluate the objective at $(0,1)^\top$, which is a (LMP), and compare this to the solution on $\bar{\Omega}$.

The upcoming (KKT) conditions are the first order necessary optimality conditions in the context of equality and inequality constraints. Box constraints are treated as inequality constraints for the sake of simplicity. The following approach uses the Lagrangian function, which is shortly introduced in this section: Assume that for a problem from Definition 2.23 we have found some point $x_* \in \Omega$, which is a (LMP) of f on Ω. The first order conditions for stationarity then demand, that we are not allowed to have some $y_* \in \mathcal{B}_\delta(x_*) \cap \Omega$ such that

$$\nabla f(x_*)^\top (y_* - x_*) < 0 \tag{2.33}$$
$$\nabla g_r(x_*)^\top (y_* - x_*) \leq 0 \quad \text{for all} \quad r \in \mathcal{A}_g(x_k) \tag{2.34}$$
$$\nabla h_j(x_*)^\top (y_* - x_*) = 0 \quad \text{for all} \quad j = 1, \ldots, m, \tag{2.35}$$

because otherwise we can expect to find a feasible point on the line segment $y_* - x_*$ leading to a smaller objective value. Be aware that this is only a hand-waving argument to understand the structure of the Lagrangian function. A valid mathematical proof requires the definition of tangential cones and can be found in the literature.

The following lemma leads to the structure of the Lagrangian function:

Lemma 2.26:
A y_* satisfying equations (2.33), (2.34) and (2.35) cannot be found, if there are $\mu_r \geq 0$ and $\lambda_j \in \mathbb{R}$ such that

$$\nabla f(x_*) + \sum_{r \in \mathcal{A}_g(x_*)} \mu_r \nabla g_r(x_*) + \sum_{j=1}^{m} \lambda_j \nabla h_j(x_*) = 0 \tag{2.36}$$

Proof. We define $d_* := y_* - x_*$ and see

$$\nabla f(x_*)^\top d_* \overset{!}{<} 0$$
$$- \sum_{r \in \mathcal{A}_g(x_*)} \underbrace{\mu_r \nabla g_r(x_*)^\top d_*}_{\leq 0} - \sum_{j=1}^{m} \underbrace{\lambda_j \nabla h_j(x_*)^\top d_*}_{=0} \overset{!}{<} 0 f$$

□

The coefficients $\mu_r \geq 0$ and $\lambda_j \in \mathbb{R}$ for the combination of the constraint gradients are called **Lagrangian multipliers**. If the (LICQ) from Definition 2.24 holds, the μ_r and λ_j are unique, if they exist.

We can easily verify:

Corollary 2.27 (Lagrangian Function)**:**
For the auxiliary function

$$L(x, \mu, \lambda) := f(x) + \sum_{r=1}^{s} \mu_r g_r(x) + \sum_{j=1}^{m} \lambda_j h_j(x) \tag{2.37}$$

holds that $\nabla_x L(x, \mu, \lambda) = 0$ is equivalent to equation (2.36), if $\mu_r = 0$ for $r \notin \mathcal{A}_g(x_k)$.

This helps us to define the necessary first order optimality condition for a problem with equality and inequality constraints.

Theorem 2.28 (Karush-Kuhn-Tucker (KKT))**:**
Consider a problem from Definition 2.23 satisfying (LICQ) or a similar constraint qualification. If x_* is a (LMP), then there exist unique vectors $\mu_* \in \mathbb{R}^s$ and $\lambda_* \in \mathbb{R}^m$ such that

$$\nabla_x L(x_*, \mu_*, \lambda_*) = 0, \tag{2.38}$$

$$h_j(x_*) = 0 \quad \textit{for all} \quad j = 1, \ldots, m. \tag{2.39}$$

$$g_r(x_*) \leq 0 \quad \textit{and} \quad \mu_r^* \geq 0 \quad \textit{for all} \quad r = 1, \ldots, s. \tag{2.40}$$

$$\mu_r^* = 0 \quad \textit{for all} \quad r \notin \mathcal{A}_g(x_*). \tag{2.41}$$

If in addition $\lambda_j^* \neq 0$ for all $j = 1, \ldots, m$ and $\mu_r^* \neq 0$ for all $r \in \mathcal{A}_g(x_*)$, the (LMP) and the multipliers satisfy the **strict complementarity condition**. We call a point satisfying the conditions from Theorem 2.28 a **(KKT)-point** or **critical point of the Lagrangian** or **stationary point** of f on Ω.

Exercise 2.29:
Find x_* solving the (KKT) conditions for the problem

$$\begin{aligned} &\textit{minimize} \quad f(u, v) = u + v \\ \textit{s.t.} \quad &(u, v)^\top \in \Omega := \left\{ (u, v)^\top \in \mathbb{R}^2 : \quad u^2 + v^2 \leq 1 \right\}. \end{aligned}$$

Also check the (LICQ) for x_* and decide, if x_* is a (LMP).

The feasible set Ω describes the set bounded by a circle with radius $r = 1$. We identify $g(x) = u^2 + v^2 - 1$ and the Lagrangian function is:

$$L(u, v, \mu) = u + v + \mu(u^2 + v^2 - 1) \tag{2.42}$$

We have to distinguish different cases of activity: If $u^2 + v^2 - 1 < 0$, then $\mu = 0$ and a (KKT) point has to satisfy $\nabla f(u, v) = 0$, but this is not possible.

So we conclude $u^2 + v^2 - 1 = 0$ and get the (KKT) conditions

$$1 + 2\mu u = 0 \Rightarrow \mu = \frac{-1}{2u}, \qquad (u \neq 0), \tag{2.43}$$

$$1 + 2\mu v = 0 \Rightarrow u = v \tag{2.44}$$

$$u^2 + v^2 - 1 = 0 \Rightarrow 2u^2 = 1 \Rightarrow u = v = \pm\frac{1}{\sqrt{2}}. \tag{2.45}$$

So $x^*_{1/2} = (\pm\frac{1}{\sqrt{2}}, \pm\frac{1}{\sqrt{2}})^\top$ and $\mu_* = \frac{-1}{2u}$ solve the (KKT) system. But $\mu_* \geq 0$ must hold, so $x_* = (\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})^\top$. The (LICQ) is satisfied, because $\nabla g(x_*) = 2x_* \neq 0$.

With the upcoming existence and uniqueness theorems it is easy to show that $x_* = (\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})^\top$ is the unique (GMP) of this problem.

Home Exercise 2.6 (Constraint Qualifications)**:**
Consider the set of feasible points

$$\Omega := \{(u, v)^\top \in \Omega_\square = [-\pi, \pi] \times [-1, 1] : \; g(u, v) = \sin(u) - v \leq 0, h(u, v) = v = 0\}$$

- ***a)*** *Decide if Ω is a convex set.*
- ***b)*** *Determine all points in Ω, for which the (LICQ) with respect to $\Omega_\square$, g and h is not satisfied.*
- ***c)*** *Consider the objective $f(u, v) = -u^2 + \exp(v)$. State the (GMPs) of f on Ω .*

Home Exercise 2.7 ((KKT) Conditions)**:**
Consider the problem

$$\begin{aligned} &\textit{minimize} \quad f(u, v, w) = \sinh(u) - u + 4w \\ &\textit{s.t.} \quad x = (u, v, w)^\top \in \Omega := \{(u, v, w)^\top \in \mathbb{R}^3 : \; -u \leq 0, v^2 + w^2 + 4w = 0\} \end{aligned}$$

- ***a)*** *Formulate the Lagrangian $L(x, \lambda, \mu)$ and the gradient $\nabla_x L(x, \lambda, \mu)$.*
- ***b)*** *Find all points x_*, λ_*, μ_* satisfying the (KKT) conditions.*
- ***c)*** *Check if the (LICQ) is satisfied at the points satisfying the (KKT) conditions.*

2.5 Existence and Uniqueness Theorems

In the previous sections we describe conditions of optimality that help us to decide, if a given point is a (LMP) or (GMP). The following theorems help us to understand, under which conditions a (LMP) or (GMP) exists at all.

Theorem 2.30 (Existence of (GMP))**:**
A) Let $f : \mathbb{R}^n \to \mathbb{R}$ with f continuous and coercive, then there exists at least one (GMP) x_* of f on every nonempty closed subset $\Omega \subseteq \mathbb{R}^n$.

B) Let $f : \Omega \to \mathbb{R}$ with f continuous and $\Omega \subset \mathbb{R}^n$ nonempty, closed and bounded (i.e. compact), then there exists at least one (GMP) x_* of f on Ω.

Proof. We start with B): The image of a continuous function on a nonempty compact set is a nonempty compact set itself. In our case the nonempty compact set Ω is mapped to a nonempty compact set in $\mathbb{R}$, which must have a minimal and a maximal value. There must be at least one $x_* \in \Omega$ that is mapped to the minimal value.

To prove A), we take one point $x_0 \in \Omega$ and evaluate $f_0 = f(x_0)$. Then we build the so called level set $\mathcal{N} := \{x \in \Omega : \quad f(x) \leq f_0\}$. If a (GMP) exists in Ω, it must be in the level set. On the other hand $\mathcal{N}$ is closed itself, because it is the intersection of closed Ω with the obviously closed $\{x \in \mathbb{R}^n : \quad f(x) \leq f_0\}$. Because f is coercive, the level set $\mathcal{N}$ is also bounded. We can therefore apply B) to get the result. □

To guarantee the uniqueness of solutions, we typically require some kind of convexity for the objective f. The following lemma connects convexity of f with the gradients and definiteness of the Hessian of f. These properties are easier to handle than the definition of convexity.

Lemma 2.31 (Sufficient Gradient and Hessian Checks for Convexity)**:**
Let $\Omega \subseteq \mathbb{R}^n$ be convex and $f : \Omega \to \mathbb{R}$ be continuously differentiable. Then the following holds:

1) *If and only if for all x, $y \in \Omega$ holds $f(y) - f(x) \geq \nabla f(x)^\top (y - x)$, then f is convex on Ω.*
2) *If and only if for all x, $y \in \Omega$, $x \neq y$ holds $f(y) - f(x) > \nabla f(x)^\top (y - x)$, then f is strictly convex on Ω.*
3) *If and only if there is $\varepsilon > 0$ such that for all x, $y \in \Omega$ with $x \neq y$ holds:*
   $f(y) - f(x) > \nabla f(x)^\top (y - x) + \varepsilon ||y - x||^2$, then f is uniformly convex on Ω.

If f is twice continuously differentiable, then the Hessian can be used to check for convexity:

4) If and only if $\nabla^2 f(x)$ is s.p.s. for all $x \in \Omega$, then f is convex on Ω.

5) If (but not only if) $\nabla^2 f(x)$ is s.p.d. for all $x \in \Omega$, then f is strictly convex on Ω.

6) If and only if there is independent $\varepsilon > 0$ such that for all $x \in \Omega$ and $d \in \mathbb{R}^n$ holds: $d^\top \nabla^2 f(x) d \geq \varepsilon ||d||^2$, then f is uniformly convex on Ω.

Theorem 2.32 (General (GMP) Condition for Convex Objectives)**:**
If $f : \Omega \to \mathbb{R}$ is convex on convex $\Omega \subseteq \mathbb{R}^n$ and x_* a (LMP), then x_* is also a (GMP) on Ω. If in addition f is continuously differentiable, then $\nabla f(x_*) = 0$ is sufficient for x_* to be a (GMP) on Ω.

Proof. A) Let x_* be a (LMP) but not a (GMP) on Ω. Then there is $y_* \in \Omega$ with $f(y_*) < f(x_*)$. We define the point $z := \lambda x_* + (1-\lambda) y_*$ and choose $\lambda \in (0,1)$ such that z satisfies $f(x_*) \leq f(z)$. Due to the convexity of f we have

$$f(z) \leq \lambda f(x_*) + (1-\lambda) f(y_*) < \lambda f(x_*) + (1-\lambda) f(x_*) = f(x_*) \tag{2.46}$$

This is a contradiction, so every (LMP) is a (GMP).

B) Now if f is differentiable, let x_*, $y_* \in \Omega$ with

$$\nabla f(x_*) = 0 \tag{2.47}$$

and assume $f(y_*) < f(x_*)$.

We use Lemma 2.31:

$$0 > f(y_*) - f(x_*) \geq \nabla f(x_*)^\top (y_* - x_*) = 0 \Rightarrow \perp \tag{2.48}$$

□

For box constraints and other simple bounds with an projection $P : \mathbb{R}^n \to \Omega$ we can sharpen the result:

Corollary 2.33 ((GMP) Condition for Simple Bounds)**:**
Let Ω be convex and let $f : \Omega \to \mathbb{R}$ be continuously differentiable and convex on Ω. Then satisfying a stationarity condition (Definition 2.7 or Theorem 2.13) is necessary and sufficient for all (GMPs) $x_* \in \Omega$.

Proof. A) Stationarity is necessary for (GMP): Combine Theorem 2.8 and Theorem 2.32.

B) Stationarity is sufficient for (GMP): Assume that $x_* \in \Omega$ satisfies stationarity:

$$\nabla f(x_*)^\top (y - x_*) \geq 0 \quad \text{for all} \quad y \in \Omega \tag{2.49}$$

and combine this with convexity of f on Ω:

$$f(y) - f(x_*) \geq \nabla f(x_*)^\top (y - x_*) \quad \text{for all} \quad y \in \Omega \tag{2.50}$$

leading to

$$f(y) \geq f(x_*) \quad \text{for all} \quad y \in \Omega \tag{2.51}$$

□

For problems with equality and inequality constraints we can do the same:

Corollary 2.34 ((GMP) Condition for Equality and Inequality Constraints)**:** *Consider a problem from Definition 2.23 satisfying (LICQ) or a similar constraint qualification. If all $g_r : \Omega \to \mathbb{R}$ are convex functions and $h : \Omega \to \mathbb{R}^m$ is affine linear, i.e. $Mx - c = 0$ with $M \in \mathbb{R}^{m \times n}, c \in \mathbb{R}^m$, then Ω is convex. If in addition the objective $f : \Omega \to \mathbb{R}$ is convex on Ω, then satisfying the (KKT) conditions is necessary and sufficient for all (GMPs) $x_* \in \Omega$.*

Proof. Because of Lemma 1.6 the set Ω is convex.

(KKT) is necessary for (GMP): Combine Theorem 2.28 and Theorem 2.32.

(KKT) is sufficient for (GMP): Assume that $x_* \in \Omega$ satisfies (KKT). We also realize because of the requirements and Lemma 2.31: $\nabla {g_r}^\top(x_*)(y - x_*) \leq g_r(y) - g_r(x_*)$ and $\nabla h^\top(x_*)(y - x_*) = M(y - x_* - c + c) = h(y) - h(x_*) = 0$.

So for any $y \in \Omega$ holds:

$$f(y) - f(x_*) \geq \nabla f(x_*)^\top (y - x_*) \tag{2.52}$$

$$= - \sum_{r \in \mathcal{A}_g(x_*)} \mu_r \nabla {g_r}^\top(x_*)(y - x_*) - \sum_{j=1}^{m} \lambda_j \nabla {h_j}^\top(x_*)(y - x_*) \tag{2.53}$$

$$= - \sum_{r \in \mathcal{A}_g(x_*)} \mu_r \nabla {g_r}^\top(x_*)(y - x_*) \geq - \sum_{r \in \mathcal{A}_g(x_*)} \mu_r (g_r(y) - g_r(x_*)) \tag{2.54}$$

$$= \sum_{r \in \mathcal{A}_g(x_*)} \mu_r (-g_r(y)) \geq 0. \tag{2.55}$$

□

Depending on the strictness of the convexity, we can decide if there is a set of (GMPs) or if the (GMP) is unique.

Theorem 2.35 (Existence and Uniqueness of (GMPs))**:**
Let $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ with Ω convex. Consider the optimization problem

$$\begin{aligned} \textit{minimize} \quad & f(x) \\ \textit{s.t.} \quad & x \in \Omega \end{aligned}$$

Then we have:

1) If f is convex on Ω, then the set of (GMPs) on Ω is convex (possibly empty).

2) If f is strictly convex on Ω, then the problem has at most one (GMP).

3) If f is uniformly convex on Ω and $\Omega \neq \emptyset$ and closed, then there exists exactly one (GMP).

Remark:
Theorem 2.35 can be combined with Lemma 2.31 to generate propositions like the following:
If f is s.p.d. on Ω, then the problem has at most one (GMP). □

Let us revisit Exercise 2.29:

Exercise 2.36:
Find x_* solving the (KKT) conditions for the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = u + v \\ \textit{s.t.} \quad & (u,v)^\top \in \Omega := \left\{ (u,v)^\top \in \mathbb{R}^2 : \quad u^2 + v^2 \leq 1 \right\}. \end{aligned}$$

Also check the (LICQ) for x_* and decide, if x_* is a (LMP).

We already found out that Ω is bounded by a circle with radius $r = 1$ and that $x^*_{1/2} = (\pm\frac{1}{\sqrt{2}}, \pm\frac{1}{\sqrt{2}})^\top$ and $\mu_* = \frac{-1}{2u}$ solve the (KKT) system.
We can now argue as follows:

- *Ω is compact, because bounded and closed. f is convex on $\mathbb{R}^n$, because $\nabla^2 f$ is s.p.s everywhere.*
- *Because of Theorem 2.30 a (GMP) exists. This (GMP) must satisfy (KKT).*
- *Because $x_* = (\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})^\top$ is the only valid solution, it must be the (GMP).*

Alternative argumentation:

- *Ω is a convex set, because $g(u,v) = u^2 + v^2 - 1 \leq 0$ is a strictly convex function on Ω because $\nabla^2 g$ is s.p.d everywhere.*
- *Because of Corollary 2.34, (KKT) is sufficient for (GMP).*
- *Because $x_* = (\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})^\top$ is the only valid solution, it must be the (GMP).*

3 Solving Optimality Conditions

In the upcoming sections we discuss solution techniques for finding (LMPs) by directly solving optimality conditions. This also leads to our first algorithm, the conjugate gradient solver.

3.1 Problem Simplification

If optimization problems show up in practical application, the very first step is to check, if simplifications are possible without changing the solution set.

Definition 3.1 (Solution Equivalence)**:**
Consider the problems

$$\textit{minimize} \quad f_a(x), \quad \textit{s.t.} \quad x \in \Omega_a$$

and

$$\textit{minimize} \quad f_b(x), \quad \textit{s.t.} \quad x \in \Omega_b.$$

*Both problems are **solution equivalent**:*

- *If all (LMPs) x_*^a of f_a on Ω_a and all (LMPs) x_*^b of f_b on Ω_b are members of $\Omega_a \cap \Omega_b$.*
- *If and only if x_* is a (LMP) of f_a on Ω_a, then it is a (LMP) of f_b on Ω_b.*
- *If and only if x_* is a (GMP) of f_a on Ω_a, then it is a (GMP) of f_b on Ω_b.*

Here are some examples for simplifications, that lead to solution equivalent problems:

- If we have to minimize $f(g(x))$ and $f : g(\Omega) \to \mathbb{R}$ is strictly monotonically increasing on $g(\Omega)$, then minimizing $g : \Omega \to \mathbb{R}$ is solution equivalent.
- If we have to minimize $f(g(x))$ and $f : g(\Omega) \to \mathbb{R}$ is strictly monotonically decreasing on $g(\Omega)$, then minimizing $-g : \Omega \to \mathbb{R}$ is solution equivalent.
- Minimizing some f on Ω is solution equivalent to minimizing on a smaller or larger Ω, as long as no existing (LMPs) get cut out or new (LMPs) are brought in.

We should therefore express the feasible set Ω as simple as possible, especially we use box constraints whenever possible. If Ω consists of a finite number of disjoint parts, it is smart to optimize the objective on each of these disjoint parts. This is called **branching**. For example, a problem with m inequality constraints can always be split up in 2^m subproblems (**branches**), where in each subproblem each inequality constraint is either active and treated as equality constraint $g_r(x) = 0$ with $\mu_r \geq 0$ or each inequality constraint is inactive with $\mu_r = 0$ and $g_r(x) < 0$. Each branch is solved separately and the resulting minimizers are compared to each other to find the best solution. Smart branching techniques allow the elimination of whole branches: For example if some $g_r(x) = 0$ can never be satisfied, all branches containing this condition are eliminated.

Sometimes a solution can be pinpointed by **relaxing** (i.e. enlarging) the feasible set, while staying solution equivalent at the same time, as the following exercise shows:

Exercise 3.2:
Solve

$$\begin{aligned} \textit{minimize} \quad & f(u,v,w) = \sqrt{u^2 - w} \\ \textit{s.t.} \quad & (u,v,w)^\top \in \Omega \\ := \{(u,v,w)^\top \in \mathbb{R}^3 : \quad & h_1(u,v,w) = u + v^2 + 2 = 0, h_2(u,v,w) = w + v^2 + 1 = 0\} \end{aligned}$$

We simplify this problem as follows: At first h_2 can be incorporated into f, which leads to elimination of w and we get

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = \sqrt{u^2 + v^2 + 1} \\ \textit{s.t.} \quad (u,v)^\top \in \Omega := \{(u,v)^\top \in \mathbb{R}^2 : \quad & h(u,v) = u + v^2 + 2 = 0\} \\ \textit{and set} \quad & w = -v^2 - 1. \end{aligned}$$

Now we use the strict monotony of $\sqrt{\cdot}$ to get

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = u^2 + v^2 \quad \textit{(convex!)} \\ \textit{s.t.} \quad (u,v)^\top \in \Omega := \{(u,v)^\top \in \mathbb{R}^2 : \quad & h(u,v) = u + v^2 + 2 = 0\} \\ \textit{and set} \quad & w = -v^2 - 1. \end{aligned}$$

We can now find a (KKT) point: The system

$$\begin{aligned} \nabla f(u,v) + \lambda \nabla h(u,v) = \begin{pmatrix} 2u + \lambda \\ 2v + \lambda 2v \end{pmatrix} = 0 \\ h(u,v) = u + v^2 + 2 = 0 \end{aligned}$$

is only solved by $(u_*, v_*, \lambda_*)^\top = (-2, 0, 4)^\top$. Sadly, $h(u,v)$ is not affine linear, so we cannot apply Theorem 2.34 directly and cannot be sure, that this is also a (LMP) or (GMP). But because $h(u,v)$ is convex (check Hessian) and $\lambda_* > 0$, the following relaxation trick works:

We consider the related problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = u^2 + v^2 \\ \textit{s.t.} \quad (u,v)^\top \in \Omega_R := \{(u,v)^\top \in \mathbb{R}^2 : \quad & g(u,v) = u + v^2 + 2 \le 0\} \\ \textit{and set} \quad & w = -v^2 - 1. \end{aligned}$$

We have to distinguish cases of activity: If $g(u,v) < 0$, then the following (KKT) system has no solution

$$\nabla f(u,v) = \begin{pmatrix} 2u \\ 2v \end{pmatrix} = 0$$
$$g(u,v) = u + v^2 + 2 < 0$$

But for $g(u,v) = 0$ we can solve

$$\nabla f(u,v) + \mu \nabla g(u,v) = \begin{pmatrix} 2u + \mu \\ 2v + \mu 2v \end{pmatrix} = 0$$
$$g(u,v) = u + v^2 + 2 = 0$$

again with $(u_*, v_*, \mu_*)^\top = (-2, 0, 4)^\top$. Because Ω_R is convex and the objective $f(u,v)$ is a convex function on Ω_R, we can apply Theorem 2.34 and know for sure: $(u_*, v_*)^\top = (-2, 0)^\top$ is a (GMP) of $f(u,v)$ on Ω_R. Because this (GMP) is also a member of the original non convex set defined by $h(u,v) = 0$, we conclude: $(u_*, v_*)^\top = (-2,0)^\top$ is a (GMP) of $f(u,v)$ on Ω.

Final result: $(u_*, v_*, w_*)^\top = (-2, 0, -1)^\top$ is the (GMP) of the original problem.

Another related technique is the conversion of inequality constraints to equality constraints and half open box constraints with the help of **slack variables**:

Lemma 3.3 (Slack Variables)**:**
Consider

$$\Omega := \{x \in \mathbb{R}^n : \quad g(x) \preceq 0 \quad \textit{for} \quad g : \mathbb{R}^n \to \mathbb{R}^m\}$$

and

$$\tilde{\Omega} := \{(\tilde{x}, \tilde{y}) \in \mathbb{R}^n \times [0,\infty)^m : \quad h(\tilde{x}, \tilde{y}) := g(\tilde{x}) + \tilde{y} = 0 \quad \textit{for} \quad g : \mathbb{R}^n \to \mathbb{R}^m\}$$

Then for all points $x \in \Omega$ exists a unique $\tilde{y} \in [0,\infty)^m$ such that $(x, \tilde{y}) \in \tilde{\Omega}$. Also for all $(\tilde{x}, \tilde{y}) \in \tilde{\Omega}$ follows $\tilde{x} \in \Omega$.

3.2 Exact Line Search Problem

A very common subproblem in the upcoming optimization algorithms is the **line search problem**

$$\begin{aligned} &\text{minimize} \quad f(x) \\ \text{s.t.} \quad & x \in L := \{x \in \mathbb{R}^n : \quad x = x_0 + td_0 \quad \text{with} \quad t \in (a,b)\} \end{aligned}$$

The vectors $x_0, d_0 \in \mathbb{R}^n$ are fixed and $(a,b) \subseteq \mathbb{R}$ is a real and open interval.

This problem is solution equivalent to a one dimensional problem in the following sense: A (LMP) or (GMP) t_* of

$$\begin{aligned} \text{minimize} \quad & \phi(t) := f(x_0 + td_0) \\ \text{s.t.} \quad & t \in (a,b) \end{aligned}$$

leads to a (LMP) or (GMP) x_* of f on the line L by setting $x_* = x_0 + t_* d_0$.

If ϕ is convex on (a,b), then using Theorem 2.32 leads to the following **exact line search condition**:

$$\nabla \phi(t_*) = \nabla f(x_0 + t_* d_0)^\top d_0 \stackrel{!}{=} 0 \tag{3.1}$$

If ϕ is not convex on (a,b), the sufficient second order optimality condition is

$$\nabla^2 \phi(t_*) = {d_0}^\top \nabla^2 f(x_0 + t_* d_0) d_0 \stackrel{!}{>} 0 \tag{3.2}$$

Home Exercise 3.1 (Exact Line Search)**:**
Consider the function

$$f(u,v) = (u^5 + 2u^4 + u^3)(v+1)$$

a) *Perform exact line search at $x_0 = (1,0)^\top$ in direction $d_0 = (-1,0)^\top$ to find three possible step sizes t_1, t_2 and t_3 solving the line search problem.*

b) *Check if the second order necessary optimality condition holds at t_1, t_2 and t_3.*

c) *Compare $f(x_0 + td_0)$ at the three step sizes to decide which one is the optimal step size.*

3.3 Unconstrained Quadratic Program

Another very common subproblem in the upcoming optimization algorithms is the **unconstrained quadratic program**

$$\begin{aligned} \text{minimize} \quad & f(x) = \frac{1}{2} x^\top A x - b^\top x \\ \text{s.t.} \quad & x \in \mathbb{R}^n \end{aligned}$$

with given $A \in \mathbb{R}^{n\times n}$ being s.p.d. and $b \in \mathbb{R}^n$. Looking at the optimality conditions (f is strictly convex) we realize that the (GMP) x_* exists uniquely and solves:

$$\nabla f(x_*) = Ax_* - b \stackrel{!}{=} 0 \Leftrightarrow Ax_* = b \tag{3.3}$$

This means that solving an unconstrained quadratic program is equivalent to solving a linear system of equations with s.p.d. system matrices. Of course we can use standard approaches like LU-decomposition or Cholesky decomposition (A is s.p.d.!), leading to:

$$A = LL^\top, \tag{3.4}$$

where L is a nonsingular lower triangular matrix. Solving $Ax_* = b$ then reduces to solving

$$Ly = b, \quad L^\top x_* = y \tag{3.5}$$

which is done efficiently with **forward and backward substitution**.

A related problem is the **quadratic program with affine linear constraints**:

$$\begin{aligned} \text{minimize} \quad & f(x) = \frac{1}{2}x^\top Ax - b^\top x \\ \text{s.t.} \quad & x \in \Omega := \{x \in \mathbb{R}^n : \quad Mx - c = 0\} \neq \emptyset \end{aligned}$$

for given given $A \in \mathbb{R}^{n\times n}$ being s.p.d. and $b \in \mathbb{R}^n$ and $M \in \mathbb{R}^{m\times n}$ and $c \in \mathbb{R}^m$. Theorem 2.34 tells us that the (GMP) is the solution of the (KKT) system:

$$\begin{aligned} Ax - b + \lambda^\top M &= 0 \\ Mx - c &= 0 \end{aligned}$$

which can be written as

$$\begin{pmatrix} A & M^\top \\ M & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$$

The matrix $\begin{pmatrix} A & M^\top \\ M & 0 \end{pmatrix}$ is symmetric by construction, but positive definiteness requires $x^\top Ax + 2x^\top M^\top \lambda > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}, \lambda \in \mathbb{R}^m$ by definition. But under mild conditions, a method like the upcoming conjugate gradient solver can be used to solve problems of this type directly.

3.4 Conjugate Gradient Solvers

A nice alternative to LU-decomposition or Cholesky decomposition for solving $Ax_* = b$ with s.p.d. matrices A is the conjugate gradient solver. The method works as follows:

- Starting at some $x_0 \in \mathbb{R}^n$ we want to construct a sequence $x_{j+1} = x_j + t_j d_j$.
- d_j is member of a set of A-conjugate directions, which have special properties.
- t_j minimizes $\frac{1}{2}x^\top Ax - b^\top x$ on the line $x_j + t_j d_j$. This is a line search subproblem.
- The sequence $x_{j+1} = x_j + t_j d_j$ will end after at most n steps with x_n being the exact solution of $Ax = b$.

To derive this method, we first need the definition of **A-conjugate directions**:

Definition 3.4 (A-Conjugate Directions)**:**
*Let A be s.p.d., then a vector system d_j with $j = 0, ..., n-1$, $d_j \neq 0$ is called A-**conjugate** (or A-orthogonal) if*

$$d_j{}^\top A d_{\tilde{j}} = 0 \quad \textit{for all} \quad j \neq \tilde{j} \tag{3.6}$$

Example 3.5:
For the case $A = LL^\top$, the system $\{d_j\}_{j=0}^{n-1}$ solving

$$L^\top d_j = p_j \tag{3.7}$$

with p_j satisfying

$$p_j \neq 0, \qquad p_j{}^\top p_{\tilde{j}} = 0 \quad \textit{for all} \quad j \neq \tilde{j} \tag{3.8}$$

is A-conjugate because

$$d_j{}^\top A d_{\tilde{j}} = (L^\top d_j)^\top (L^\top d_{\tilde{j}}) = p_j{}^\top p_{\tilde{j}} = 0 \tag{3.9}$$

Lemma 3.6 (Linear Independence of A-Conjugate Directions)**:**
Let d_j with $j = 0, ..., n-1$ be A-conjugate. Then the set $\{d_j\}_{j=0}^{n-1}$ is linearly independent and the inverse of A satisfies

$$A^{-1} = \sum_{j=0}^{n-1} \frac{1}{\rho_j} d_j d_j{}^\top \quad \textit{with} \quad \rho_j = d_j{}^\top A d_j > 0 \tag{3.10}$$

Proof. Assume $d_0, d_1, \ldots, d_{n-1}$ are not linearly independent. Then there are coefficients α_j with $j = 0, \ldots, n-1$ and not all zero, such that $\sum_{j=0}^{n-1} \alpha_k d_k = 0$. Let especially $\alpha_{\tilde{j}} \neq 0$, then

$$0 = \left(\sum_{j=0}^{n-1} \alpha_j d_j\right)^\top A d_{\tilde{j}} = \sum_{j=0}^{n-1} \alpha_j d_j{}^\top A d_{\tilde{j}} = \alpha_{\tilde{j}} \cdot d_{\tilde{j}}{}^\top A d_{\tilde{j}} \overset{s.p.d.}{\neq} 0 \Rightarrow \perp$$

Because our $\{d_j\}_{j=0}^{n-1}$ are linearly independent, there are coefficients α_j with $j = 0, \dots, n-1$ such that $x \in \mathbb{R}^n$ can be composed: $x = \sum_{j=0}^{n-1} \alpha_j d_j$. Look at:

$$\begin{aligned}(\sum_{j=0}^{n-1} \frac{1}{\rho_j} d_j {d_j}^\top) Ax &= \sum_{j=0}^{n-1} \frac{1}{\rho_j} d_j \left[{d_j}^\top A \sum_{\tilde{j}=0}^{n-1} \alpha_{\tilde{j}} d_{\tilde{j}} \right] \\ &= \sum_{j=0}^{n-1} \frac{\alpha_j}{\rho_j} d_j {d_j}^\top A d_j = \sum_{j=0}^{n-1} \alpha_j d_j = x.\end{aligned}$$

□

Home Exercise 3.2 (Conjugate Directions for Inverse Matrices)**:**
Consider the matrix

$$A = \frac{1}{4} \begin{pmatrix} 3 & 0 & -1 \\ 0 & 8 & 0 \\ -1 & 0 & 3 \end{pmatrix}.$$

- ***a)*** *Compute the eigenvalues* $\{\lambda_i\}_{k=1}^3$ *and a set of pairwise orthogonal eigenvectors* $\{v_i\}_{i=1}^3$ *of* A.
- ***b)*** *Prove in general: A set of pairwise orthogonal eigenvectors of a matrix* A *is always* A*-conjugate.*
- ***c)*** *Compute the inverse matrix of* A *with the formula:* $A^{-1} = \sum_{i=1}^3 \frac{1}{{v_i}^\top A v_i} v_i {v_i}^\top$.

Let us now compute the step sizes t_j:

Lemma 3.7 (Optimal Step Sizes for Quadratic Line Search)**:**
The line search problem

$$\begin{aligned} &\textit{minimize} \quad f(x) = \frac{1}{2} x^\top A x - b^\top x \\ \textit{s.t.} \quad & x \in L := \{x \in \mathbb{R}^n : \quad x = x_j + t d_j \quad \textit{with} \quad t \in \mathbb{R}\} \end{aligned}$$

is solved by $x_* = x_j + t_j d_j$ *with*

$$t_j = -\frac{(Ax_j - b)^\top d_j}{\rho_j} \quad \textit{with} \quad \rho_j = {d_j}^\top A d_j > 0 \tag{3.11}$$

Proof. Because f is convex on the convex line L, we only require

$$0 \overset{!}{=} \nabla f(x_j + t_j d_j)^\top d_j = (A(x_j + t_j d_j) - b)^\top d_j = (Ax_j - b)^\top d_j + t_j {d_j}^\top A d_j \tag{3.12}$$

□

3.5 Gram-Schmidt Orthogonalization

With the step sizes given, we only need to know how to obtain a set of A-conjugate descent directions. One possibility is the Gram-Schmidt orthogonalization procedure.

Lemma 3.8 (Gram-Schmidt Orthogonalization)**:**
If the set $\{p_j\}_{j=0}^{n-1}$ *is linearly independent, then the vectors* $\{d_j\}_{j=0}^{n-1}$ *constructed by*

$$d_0 = p_0 \tag{3.13}$$

$$d_{j+1} := p_{j+1} - \sum_{i=0}^{j} \frac{{p_{j+1}}^\top A d_i}{\rho_i} d_i \tag{3.14}$$

are A-conjugate and $\operatorname{span}\{p_0, p_1, \dots, p_j\} = \operatorname{span}\{d_0, d_1, \dots, d_j\}$ *for all* $j \leq n-1$.

Proof. Induction over $j = 0, \dots, n-1$.
Initiation: For $j = 0$ we have $d_0 = p_0$ and $d_1 = p_1 - \frac{{p_1}^\top A p_0}{\rho_0} p_0 \neq 0$. We can verify that ${d_1}^\top A d_0 = 0$ and $\operatorname{span}\{d_0, d_1\} = \operatorname{span}\{d_0, p_1 - \frac{{p_1}^\top A d_0}{\rho_0} d_0\} = \operatorname{span}\{d_0, p_1\} = \operatorname{span}\{p_0, p_1\}$.

Assume: The lemma is true for vector sets $d_0, d_1, \dots, d_{j+1}$.
Induction: Define

$$d_{j+2} := p_{j+2} - \sum_{i=0}^{j+1} \frac{{p_{j+2}}^\top A d_i}{\rho_i} d_i \neq 0 \tag{3.15}$$

and observe for all $0 \leq \tilde{j} \leq j+1$:

$${d_{j+2}}^\top A d_{\tilde{j}} = {p_{j+2}}^\top A d_{\tilde{j}} - \sum_{i=0}^{j+1} \frac{{p_{j+2}}^\top A d_i}{\rho_i} {d_i}^\top A d_{\tilde{j}} = {p_{j+2}}^\top A d_{\tilde{j}} - {p_{j+2}}^\top A d_{\tilde{j}} = 0. \tag{3.16}$$

So d_{j+2} is A-conjugate to all $d_{\tilde{j}}$ with $0 \leq \tilde{j} \leq j+1$.

Now let $x \in \operatorname{span}\{d_0, \dots, d_{j+2}\}$, then

$$\begin{aligned} x &\in \operatorname{span}\{d_0, \dots, d_{j+1}, p_{j+2} - \sum_{i=0}^{j+1} \frac{{p_{j+2}}^\top A d_i}{\rho_i} d_i\} \\ &= \operatorname{span}\{d_0, \dots, d_{j+1}, p_{j+2}\} = \operatorname{span}\{p_0, \dots, p_{j+1}, p_{j+2}\}. \end{aligned}$$

□

So if we have a linearly independent set $\{p_j\}_{j=0}^{n-1}$, we can construct the A-conjugate descent directions using Lemma 3.8. It is especially sufficient to have a set of A-conjugate descent directions $\{d_j\}_{j=0}^{\tilde{j}}$ and one $p_{\tilde{j}+1}$ with $p_{\tilde{j}+1} \perp \operatorname{span}\{d_0, ..., d_{\tilde{j}}\}$ to construct $d_{\tilde{j}+1}$.

Home Exercise 3.3 (Gram-Schmidt Orthogonalization):
Consider the quadratic problem

$$\text{minimize} \quad f(x) = \frac{1}{2}x^\top Ax - b^\top x \quad \text{with} \quad A = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 3 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix}$$
$$\text{s.t.} \quad x \in \mathbb{R}^3$$

a) *Use Gram-Schmidt to construct a A-conjugate set of vectors d_0, d_1 and d_2 out of the unit vector set $p_0 = (1,0,0)^\top$, $p_1 = (0,1,0)^\top$ and $p_2 = (0,0,1)^\top$.*

b) *Perform three conjugate direction steps starting at $x_0 = (1,4,0)^\top$ to get the solution x_3 of the quadratic problem.*

c) *Verify that $\nabla f(x_1) \perp p_0$, $\nabla f(x_2) \perp \operatorname{span}\{p_0, p_1\}$ and $\nabla f(x_3) = 0$.*

3.6 General Conjugate Direction Algorithm

With every set of A-conjugate directions $d_0, d_1, \ldots, d_{n-1}$ we can perform the following algorithm ($Ax_j - b$ is called r_j here, because it is used as residual):

Algorithm 3.9 (Conjugate Direction Method):
For minimizing $\frac{1}{2}x^\top Ax - b^\top x$ with s.p.d. matrix A and ***known conjugate directions****:*

1. *Input: $A \in \mathbb{R}^{n\times n}$; b, $x_0 \in \mathbb{R}^n$; $d_0, d_1, \ldots, d_{n-1} \in \mathbb{R}^n$.*
2. *Set $r_0 \leftarrow Ax_0 - b$.*
3. *For $j = 0, \ldots, n-1$ do*
   a) *Set $\tilde{d}_j \leftarrow Ad_j$.*
   b) *Set $\rho_j \leftarrow {d_j}^\top \tilde{d}_j$.*
   c) *Set $t_j \leftarrow -\frac{{r_j}^\top d_j}{\rho_j}$.*
   d) *Set $x_{j+1} \leftarrow x_j + t_j d_j$.*
   e) *Set $r_{j+1} \leftarrow r_j + t_j \tilde{d}_j$ (or alternatively $r_{j+1} \leftarrow Ax_{j+1} - b$).*
4. *Output: $x_* \leftarrow x_n$ and r_n will be zero.*

Exercise 3.10:
Solve the following unconstrained quadratic program

$$\begin{array}{rl} \textit{minimize} & f(u,v) = (u-3)^2 + 2v^2 \\ \textit{s.t.} & x = (u,v)^\top \in \mathbb{R}^2 \end{array}$$

with the conjugate direction method using the A-conjugate directions $d_0 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ *and* $d_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ *at* $x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

We remember $A = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$ *and* $b = \begin{pmatrix} 6 \\ 0 \end{pmatrix}$. *We get* $\nabla f_0 = Ax_0 - b = (-6,0)^\top = r_0$. *Next we compute* $\tilde{d}_0 = (2,-4)^\top$, $\rho_0 = {d_0}^\top \tilde{d}_0 = 6$, $t_0 = -\frac{\nabla {f_0}^\top d_0}{\rho_0} = 1$ *and* $x_1 = x_0 + d_0 = (1,-1)^\top$. *Then we update* $r_1 = \nabla f_1 = \nabla f_0 + t_0 A d_0 = -4(1,1)^\top$. *Next we compute* $\tilde{d}_1 = 4(1,1)^\top$, $\rho_1 = 12$, $t_1 = 1$ *and* $x_2 = x_1 + d_1 = (3,0)^\top$. *Also* $r_2 = 0$.

It is surprising that the conjugate direction method terminates after a finite number of steps. This feature of conjugate directions can be proved easily:

Theorem 3.11 (Properties of Conjugate Direction Method)**:**
Let $\{d_0, ..., d_{n-1}\}$ *be A-conjugate. Let* x_j, t_j *be computed according to:*

1. $x_0 \in \mathbb{R}^n$ *given.*
2. *For* $j = 0, ..., n-1$*:*

$$t_j = -\frac{(Ax_j - b)^\top d_j}{\rho_j} \quad \textit{and} \quad x_{j+1} = x_j + t_j d_j. \tag{3.17}$$

Then

1. $\nabla f(x_j) = (Ax_j - b) \perp \operatorname{span}\{d_0, ..., d_{j-1}\} =: V_j$.
2. x_j *is a (GMP) of* f *on* $x_0 + V_j$.
3. x_n *solves* $Ax = b \Leftrightarrow x_n$ *is the (GMP) of the problem*

$$\begin{array}{rl} \textit{minimize} & f(x) = \frac{1}{2} x^\top A x - b^\top x \\ \textit{s.t.} & x \in \mathbb{R}^n \end{array}$$

Proof. For $i \leq j-1$ holds $x_j = x_i + \sum\limits_{k=i}^{j-1} t_k d_k$, so

$$\nabla f(x_j) = Ax_j - b = Ax_i - b + \sum_{k=i}^{j-1} t_k A d_k = \nabla f(x_i) + \sum_{k=i}^{j-1} t_k A d_k. \tag{3.18}$$

So for each $i \leq j-1$:

$$d_i{}^\top \nabla f(x_j) = d_i{}^\top \nabla f(x_i) + \sum_{k=i}^{j-1} t_k d_i{}^\top A d_k = d_i{}^\top \nabla f(x_i) + t_i \rho_i = 0. \tag{3.19}$$

and in consequence for all $\tilde{d} \in V_j = \operatorname{span}\{d_0, ..., d_{j-1}\}$ holds

$$\tilde{d}^\top \nabla f(x_j) = \sum_{k=0}^{j-1} \alpha_k d_k{}^\top \nabla f(x_j) = 0 \tag{3.20}$$

We formulate the stationarity condition from Definition 2.7 for this situation:

$$\nabla f(x_j)^\top (y - x_j) \overset{!}{\geq} 0 \quad \text{for all} \quad y \in \Omega := x_0 + V_j, \tag{3.21}$$

This condition is necessary according to Theorem 2.8, but also sufficient in the context of convex objectives on convex sets (see Theorem 2.33). Because $x_j = x_0 + \sum\limits_{k=0}^{j-1} t_k d_k$, we realize that $y - x_j \in \operatorname{span}\{V_j - \sum\limits_{k=0}^{j-1} t_k d_k\} = V_j$. Combination with equation (3.20) leads to

$$\nabla f(x_j)^\top (y - x_j) = 0 \quad \text{for all} \quad y \in \Omega := x_0 + V_j, \tag{3.22}$$

So x_j is a (GMP) on $x_0 + V_j$. And especially for $j = n$ the (GMP) is $x_n \in x_0 + V_n = \mathbb{R}^n$. □

3.7 Conjugate Gradient Algorithm

In general it is quite tedious to generate a set of A-conjugate directions. But with the upcoming conjugate gradient solver we establish a very sophisticated way to build the set of A-conjugate directions in real time: We just use Lemma 3.8 for the choice $p_j = -\nabla f(x_j)$. By executing this idea we get $d_0 = -\nabla f(x_0) =: -\nabla f_0$ and

$$d_{j+1} = -\nabla f_{j+1} + \sum_{k=0}^{j} \frac{\nabla f_{j+1}{}^\top A d_k}{\rho_k} d_k \tag{3.23}$$

An important consequence of $\nabla f_{j+1} \perp \operatorname{span}\{d_0, ..., d_j\}$ is:

$$\nabla f_{j+1}{}^\top d_{j+1} = -||\nabla f_{j+1}||^2 + \sum_{k=0}^{j} \frac{\nabla f_{j+1}{}^\top A d_k}{\rho_k} \underbrace{\nabla f_{j+1}{}^\top d_k}_{=0} = -||\nabla f_{j+1}||^2. \tag{3.24}$$

This changes the step size computation to $t_j = \frac{||\nabla f_j||^2}{\rho_j}$, but is also used to simplify the update of d_j. But first we realize $\nabla f_{j+1} = A(x_j + t_j d_j) - b = \nabla f_j + t_j A d_j$ and in consequence we get $A d_j = \frac{\nabla f_{j+1} - \nabla f_j}{t_j}$ and

$$d_{j+1} = -\nabla f_{j+1} + \sum_{k=0}^{j} \frac{\nabla f_{j+1}{}^\top (\nabla f_{k+1} - \nabla f_k)}{\rho_k t_k} d_k \tag{3.25}$$

We combine $\nabla f_k \in \operatorname{span}\{d_0, ..., d_j\}$ for $k \leq j$ and $\nabla f_{j+1} \perp \operatorname{span}\{d_0, ..., d_j\}$ to conclude: $\nabla f_{j+1} \perp \nabla f_k$ for $k \leq j$.

This has heavy consequences:

$$d_{j+1} = -\nabla f_{j+1} + \frac{\nabla f_{j+1}{}^\top \nabla f_{j+1}}{-\nabla f_j{}^\top d_j} d_j = -\nabla f_{j+1} + \frac{||\nabla f_{j+1}||^2}{||\nabla f_j||^2} d_j \tag{3.26}$$

Algorithm 3.12 (Conjugate Gradient Solver)**:**
For solving $Ax = b$ with s.p.d. matrix A

1. *Input:* $A \in \mathbb{R}^{n \times n}$; b; $\delta > 0$.
2. *Set* $x_j \leftarrow b$ *(or otherwise given),* $r_j \leftarrow A x_j - b$ *and* $d_j \leftarrow -r_j$.
3. *While* $||r_j|| > \delta$ *do*
   a) *Set* $\tilde{d}_j \leftarrow A d_j$.
   b) *Set* $\rho_j \leftarrow d_j{}^\top \tilde{d}_j$.
   c) *Set* $t_j \leftarrow \frac{||r_j||^2}{\rho_j}$.
   d) *Set* $x_j \leftarrow x_j + t_j d_j$.
   e) *Set* $r_{old} \leftarrow r_j$.
   f) *Set* $r_j \leftarrow r_{old} + t_j \tilde{d}_j$.
   g) *Set* $\beta_j \leftarrow \frac{||r_j||^2}{||r_{old}||^2}$
   h) *Set* $d_j \leftarrow -r_j + \beta_j d_j$.
4. *Output:* $x_* \leftarrow x_j$.

Remark:
We showed in Theorem 3.11 that the algorithm terminates after n steps, returning

the (GMP). The termination condition $||r_j|| = ||Ax_j - b|| < \delta$ is useful if x_j can be accepted with some small residual and the dimension n is very large. It then can happen that x_j for $j << n$ is sufficiently close to x_*. For large scale problems with sparse matrices, the conjugate gradient method beats Cholesky decomposition in efficiency, because the decomposition can lead to nonsparse but still large scale matrices L. Also, $x_0 = b$ is used as default starting value to directly solve cases, where A is the unit matrix. The termination condition can be satisfied after very few steps, if A has a good condition number $\kappa(A) := \frac{\lambda_{max}}{\lambda_{min}}$. This can be enforced with **preconditioning**, i.e. transforming the problem into solving the solution equivalent problem $S^\top ASx = S^\top b$, whereas $S^\top S = B$ is s.p.d. and both close to A^{-1} and cheap to compute. An example using incomplete Cholesky decomposition is found in the appendix (Algorithm 11.1). The requirement A being s.p.d. is not necessary for the algorithm to solve a system $Ax = b$, the algorithm also works if $\rho_j \neq 0$ for all $j = 1, \dots, n$. □

Exercise 3.13:
Solve

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = (u-3)^2 + 2v^2 \\ \textit{s.t.} \quad & x = (u,v)^\top \in \mathbb{R}^2 \end{aligned}$$

using conjugate gradient solver with starting value $x_0 = \begin{pmatrix}1\\1\end{pmatrix}$ *and* $\varepsilon = \frac{1}{1000}$.

We see

$$(u-3)^2 + 2v^2 = u^2 - 6u + 9 + 2v^2 = \frac{1}{2}\begin{pmatrix}u & v\end{pmatrix}\begin{pmatrix}2 & 0\\0 & 4\end{pmatrix}\begin{pmatrix}u\\v\end{pmatrix} - (6,0)\begin{pmatrix}u\\v\end{pmatrix} + 9$$

so $A = \begin{pmatrix}2 & 0\\0 & 4\end{pmatrix}$ *and* $b = \begin{pmatrix}6\\0\end{pmatrix}$.

We get $\nabla f_0 = Ax_0 - b = (-4,4)^\top = r_0 = -d_0$ *and* $||r_0|| = 4\sqrt{2}$.

Next we compute $\tilde{d}_0 = (8,-16)^\top$, $\rho_0 = d_0{}^\top \tilde{d}_0 = 96$, $t_0 = \frac{||r_0||^2}{\rho_0} = \frac{1}{3}$ *and* $x_1 = x_0 + \frac{1}{3}d_0 = \frac{1}{3}(7,-1)^\top$.

Then we update $r_1 = \nabla f_1 = \nabla f_0 + t_0\tilde{d}_0 = \frac{-4}{3}(1,1)^\top$ *and check the residual:* $||r_1|| = \frac{4}{3}\sqrt{2}$. *We require another iteration starting with* $\beta_0 = \frac{||r_1||^2}{||r_0||^2} = \frac{\frac{32}{9}}{32} = \frac{1}{9}$ *and* $d_1 = -\nabla f_1 + \beta_0 d_0 = \frac{8}{9}(2,1)^\top$.

Next we compute $\tilde{d}_1 = \frac{32}{9}(1,1)^\top$, $\rho_1 = \frac{8}{9}\frac{32}{9}(2,1)(1,1)^\top = \frac{256}{27}$, $t_1 = \frac{32}{9}\frac{27}{256} = \frac{3}{8}$ *and* $x_2 = x_1 + \frac{3}{8}d_1 = (3,0)^\top$. x_2 *is (GMP) because* $r_2 = 0$.

Home Exercise 3.4 (Conjugate Gradient Algorithm):
Consider the quadratic problem

$$\begin{aligned} \textit{minimize} \quad & f(\alpha,\beta,\gamma,\delta) = \frac{1}{2}\alpha^2 + \frac{3}{2}\beta^2 + \gamma^2 + \delta^2 - \alpha - \beta - \delta \\ \textit{s.t.} \quad & x = (\alpha,\beta,\gamma,\delta)^\top \in \mathbb{R}^4 \end{aligned}$$

a) *Find A and b such that $f(x) = \frac{1}{2}x^\top Ax - b^\top x$.*

b) *Perform maximal four conjugate gradient steps starting at $x_0 = b$ to get the solution x_* of the quadratic program.*

c) *Verify that A is s.p.d. and that $Ax_* = b$.*

4 Descent Algorithms

In the previous section we discussed methods to find solutions for the line search problem and the quadratic program by directly solving the corresponding optimality conditions. In the upcoming section we will introduce descent algorithms, which will generate a **descend sequence** of points $x_k \in \Omega$ with the **descent property** $f(x_{k+1}) < f(x_k)$. The sequence is generated by identifying **descent directions** at x_k and descending along these directions as deep as possible to get x_{k+1}. The identification process typically demands that we solve a series of **local line search problems** and **local quadratic programs**. If the sequence converges to a point x_* satisfying the optimality conditions (in theory) or satisfies a **termination criterion** based on optimality conditions (in practice), we have found a (LMP).

4.1 Basic Assumptions

We first make some restrictions, for which kind of problems descent algorithms can be applied:

Assumption 4.1 (Continuously Differentiable Objective on Simple Set)**:**
For the problem

$$\begin{aligned} \textit{minimize} \quad & f(x) \\ \textit{s.t.} \quad & x \in \Omega \end{aligned}$$

the objective f is bounded from below and at least once continuously differentiable, the gradient ∇f is Lipschitz continuous:

$$\textit{There is} \quad L > 0 \quad \textit{with} \quad ||\nabla f(x) - \nabla f(y)|| \leq L||x - y|| \quad \textit{for all} \quad x, y \in \mathbb{R}^n. \tag{4.1}$$

The feasible set $\Omega \neq \emptyset$ is either unconstrained with $\Omega = \mathbb{R}^n$ or a set of box constraints $\Omega_\Box = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_n, b_n]$.

The basic descent algorithm is then formulated as follows:

Algorithm 4.2 (Basic Descent Algorithm)**:**
Let Assumption 4.1 be true.

1. *Input:* f, $x_0 \in \Omega$; *choose* $\varepsilon > 0$.
2. *Set* $x_k \leftarrow x_0$.
3. *While* x_k *does not satisfy a* ***termination check*** *do*
   a) *Calculate a* ***descent direction*** d_k *of* f *at* x_k.
   b) *Calculate a* ***step size*** $t_k > 0$ *such that*
   $$f(x_k + t_k d_k) < f(x_k) \tag{4.2}$$
   and $x_k + t_k d_k \in \Omega$.
   c) *Set* $x_k \leftarrow x_k + t_k d_k$.
4. *Output:* $x_* \leftarrow x_k$.

4.2 Termination Checks

Descent algorithms typically terminate, if x_k is close to being stationary or satisfying a necessary first order optimality condition:

Definition 4.3 (Termination Checks)**:**
Consider the objective $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$. *For* $\Omega = \mathbb{R}^n$ *a point* $x \in \Omega$ ***satisfies the termination check*** *with tolerance* $\varepsilon > 0$, *if:*

$$||\nabla f(x)|| \leq \varepsilon \tag{4.3}$$

For $\Omega = \Omega_\Box$ *a point* $x \in \Omega$ ***satisfies the termination check*** *with tolerance* $\varepsilon > 0$, *if:*

$$||x_k - P(x_k - \nabla f(x_k))|| \leq \varepsilon \tag{4.4}$$

with the projection $P : \mathbb{R}^n \to \Omega_\Box$ *from Definition 2.11.*

4.3 Descent Directions

Definition 4.4:
For $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ *a vector* $d \in \mathbb{R}^n$ *is called a* ***descent direction*** *at* $x \in \mathbb{R}^n$, *if there is* $\varepsilon > 0$ *such that:*

$$f(x + td) < f(x) \quad \text{for all} \quad t \in (0, \varepsilon]. \tag{4.5}$$

If in addition $x + td \in \Omega$ for all $t \in (0, \varepsilon]$, we say that the descent direction ***does not lead out of*** *Ω.*

For smooth functions, descent directions can be identified with the gradient:

Lemma 4.5 (Descent Direction Check)**:**
If $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable in $\mathcal{B}_\varepsilon(x)$ and

$$\nabla f(x)^\top d < 0 \quad \textit{for some} \quad d \in \mathbb{R}^n \tag{4.6}$$

then d is a descent direction at x.

Proof.

$$0 > \nabla f(x)^\top d = \lim_{t \to 0} \frac{f(x + td) - f(x)}{t}. \tag{4.7}$$

So there is $\varepsilon > 0$ such that for all $t \in (0, \varepsilon]$ holds:

$$\frac{1}{t}\big(f(x + td) - f(x)\big) < 0. \tag{4.8}$$

□

Remark:
We know nothing about $d \neq 0$ if $\nabla f(x)^\top d = 0$, which marks a critical point. But we can show that

$$\nabla f(x)^\top d > 0 \quad \text{for some} \quad d \in \mathbb{R}^n \tag{4.9}$$

implies that d is not a descent direction at x. □

Theorem 4.6 (Examples for Descent Directions)**:**
1) Let $B(x) \in \mathbb{R}^{n \times n}$ be any s.p.d. matrix and $\nabla f(x) \neq 0$, then

$$B(x)d(x) := -\nabla f(x) \tag{4.10}$$

is a descent direction.
2) The steepest descent direction is $d(x) = -\nabla f(x)$ (for $B(x) \equiv \mathbb{E}$) and $d_* := -\frac{\nabla f(x)}{||\nabla f(x)||}$ solves the local linear model:

$$\begin{aligned} &\textit{minimize} \quad \nabla f(x)^\top d \\ &\textit{s.t.} \quad d \in \mathbb{R}^n, ||d|| \leq 1 \end{aligned}$$

Proof. 1) If B is s.p.d., so is B^{-1}. Let $d(x) = -B^{-1}(x)\nabla f(x)$, then

$$\nabla f(x)^\top d = -\nabla f(x)^\top B^{-1}(x)\nabla f(x) < 0.$$

2) For $d_* = -\frac{\nabla f(x)}{||\nabla f(x)||}$ holds $\nabla f(x)^\top d_* = -||\nabla f(x)||$, which is the infimum of $\nabla f(x)^\top d$ on $||d|| \leq 1$ and therefore the (GMP) because

$$\nabla f(x)^\top d = \cos(\angle[\nabla f(x), d]) \cdot ||\nabla f(x)|| \cdot \underbrace{||d||}_{\leq 1} \geq (-1) \cdot ||\nabla f(x)||$$

□

Home Exercise 4.1 (Descent Directions)**:**
Consider the function

$$f(u, v) = 1 - u^2 - v^2$$

a) *State a descent direction for f at the general point $x_0 = (r_0 \cos(\phi_0), r_0 \sin(\phi_0))^\top$ depending on $r_0 > 0$, $\phi_0 \in [0, 2\pi]$.*

b) *Is there a descent direction at the point $x_* = (0, 0)^\top$?*

c) *Prove in general: If $d_k \in \mathbb{R}^n$ is a descent direction at some $x_k \in \mathbb{R}^n$ for continuously differentiable $f : \mathbb{R}^n \to \mathbb{R}$, then αd_k with $\alpha > 0$ is also a descent direction.*

4.4 Step Size for Unconstrained Problems

We already discussed the exact line search problem section 3.2 and its optimality conditions. But because exact line search can be tedious and only $f(x_k + t_k d_k) < f(x_k)$ has to be satisfied, numerical approximations of t_k that lead to a descent are preferred. One simple method to descent on a line is the golden section line search method:

Algorithm 4.7 (Golden Section Line Search)**:**
For minimizing $\phi(t) = f(x_k + td_k)$ in the interval $[t_a, t_d]$.

1. *Input: f, x_k, d_k; choose $\varepsilon > 0$.*
2. *Set $\gamma \leftarrow \frac{\sqrt{5}-1}{2}$.*
3. *Calculate $t_b \leftarrow t_d - \gamma(t_d - t_a)$ and $t_c \leftarrow t_a + \gamma(t_d - t_a)$.*
4. *While $|t_d - t_a| > \varepsilon$ do*
   a) *If $\phi(t_b) < \phi(t_c)$ set $t_d \leftarrow t_c$ and $t_c \leftarrow t_b$ and $t_b \leftarrow t_d - \gamma(t_d - t_a)$.*

b) Else set $t_a \leftarrow t_b$ *and* $t_b \leftarrow t_c$ *and* $t_c \leftarrow t_a + \gamma(t_d - t_a)$.

5. *Output:* $t_s \leftarrow \frac{t_a+t_d}{2}$.

The result t_s of golden section line search approximates a random (LMP) $t_* \in [t_a, t_d]$ of the exact line search problem with precision $|t_s - t_*| < \frac{\varepsilon}{2}$. It is not clear how to choose $[t_a, t_d]$. Another issue is the fixed number of calculation steps: Golden section line search executes all steps, even if the starting point is already close to the solution.

Alternatively, we can use gradient information to construct an advanced line search method. We start this discussion with the definition of **local linear and local quadratic models**, which will show up on different occasions later.

Definition 4.8 (Linear and Quadratic Models)**:**
Let $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ *be continuously differentiable and* $x_k \in \Omega$*. Then the local linear model* $L_f(x; x_k) : \Omega \to \mathbb{R}$ *is*

$$L_f(x; x_k) = f(x_k) + \nabla f(x_k)^\top (x - x_k) \tag{4.11}$$

If in addition f *is twice continuously differentiable, the local quadratic model* $Q_f(x; x_k) : \Omega \to \mathbb{R}$ *is*

$$Q_f(x; x_k) = f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k) \tag{4.12}$$

If we consider a linear model $L_f(x; x_k)$ for the objective f at iteration step k and identify a t_k such that $x_{k+1} = x_k + t_k d_k$, then the reduction in function value of the linear model $L_f(x; x_k)$ would be

$$L_f(x_k; x_k) - L_f(x_{k+1}; x_k) = f(x_k) - f(x_k) - \nabla f(x_k)^\top (x_{k+1} - x_k) = -\nabla f(x_k)^\top (t_k d_k),$$

but the actual reduction of the function f is $f(x_k) - f(x_{k+1})$.

We say that a step size t_k leads to a sufficient decrease for given $\sigma \in (0, \frac{1}{2})$ if

$$\frac{f(x_k) - f(x_{k+1})}{L_f(x_k; x_k) - L_f(x_{k+1}; x_k)} \geq \sigma \tag{4.13}$$

$$\Leftrightarrow f(x_k) - f(x_k + t_k d_k) \geq -\sigma \nabla f(x_k)^\top t_k d_k \tag{4.14}$$

$$\Leftrightarrow f(x_k + t_k d_k) \leq f(x_k) + \sigma t_k \nabla f(x_k)^\top d_k \tag{4.15}$$

On the other hand we want to make sure that the step size is sufficiently large. With respect to given $\eta \in (\sigma, 1)$ we demand that the steepness at $x_k + t_k d_k$ is larger than the current steepness:

$$\nabla f(x_k + t_k d_k)^\top d_k \geq \rho \nabla f(x_k)^\top d_k \tag{4.16}$$

This consideration leads to

Definition 4.9 (Wolfe-Powell Step Size)**:**
For the line search problem

$$\begin{aligned} \textit{minimize} \quad & \phi(t) := f(x_k + td_k) \\ \textit{s.t.} \quad & t \in (0, \infty) \end{aligned}$$

with $\nabla f(x_k)^\top d_k < 0$ *(descent direction) a step size* t_* *satisfies the* ***sufficient decrease condition*** *(or Armijo rule or first Wolfe-Powell condition) with respect to* $\sigma \in (0, \frac{1}{2})$ *if*

$$f(x_k + t_* d_k) \leq f(x_k) + \sigma t_* \nabla f(x_k)^\top d_k \tag{4.17}$$

The step size satisfies the ***sufficient steepness condition*** *(or second Wolfe-Powell condition) with respect to* $\rho \in (\sigma, 1)$*, if*

$$\nabla f(x_k + t_* d_k)^\top d_k \geq \rho \nabla f(x_k)^\top d_k \tag{4.18}$$

It can be shown that under Assumption 4.1 with $\Omega = \mathbb{R}^n$ a Wolfe-Powell step size $t_* \in (0, \infty)$ can always be found. We introduce now an algorithm, that starts with $t_0 = 1$ and returns a step size $t_* \in (0, \infty)$ satisfying the Wolfe-Powell conditions:

Algorithm 4.10 (Wolfe-Powell Line Search)**:**
For reducing $\phi(t) = f(x_k + td_k)$ *in the interval* $(0, \infty)$*.*

1. *Input:* f*,* x_k*,* d_k*; choose* $\sigma \in (0, \frac{1}{2}), \rho \in (\sigma, 1)$*.*
2. *If* $\nabla f(x_k)^\top d_k \geq 0$*, return error (descent direction check fails).*
3. *Define* $W1(t) = f(x_k + td_k) \leq f(x_k) + t\sigma \nabla f(x_k)^\top d_k$ *(bool-valued function).*
4. *Define* $W2(t) = \nabla f(x_k + td_k)^\top d_k \geq \rho \nabla f(x_k)^\top d_k$ *(bool-valued function).*
5. *Set* $t \leftarrow 1$*.*
6. *If* $W1(t) == FALSE$ *do* ***backtracking****:*
   a) *Set* $t \leftarrow \frac{t}{2}$*.*
   b) *While* $W1(t) == FALSE$ *do*
      i. *Set* $t \leftarrow \frac{t}{2}$*.*
   c) *Set* $t_- \leftarrow t$ *and* $t_+ \leftarrow 2t$*.*
7. *Elseif* $W2(t) == TRUE$ *return* $t_* \leftarrow t$*.*

8. *Else do **fronttracking***
   a) *Set* $t \leftarrow 2t$.
   b) *While* $W1(t) == TRUE$ *do*
      i. *Set* $t \leftarrow 2t$.
   c) *Set* $t_- \leftarrow \frac{t}{2}$ *and* $t_+ \leftarrow t$.
9. *Set* $t \leftarrow t_-$.
10. *While* $W2(t) == FALSE$ *do **refining***
    a) *Set* $t \leftarrow \frac{t_- + t_+}{2}$.
    b) *If* $W1(t) == TRUE$ *set* $t_- \leftarrow t$, *else set* $t_+ \leftarrow t$.
11. *Output:* $t_* \leftarrow t_-$.

Lemma 4.11 (Termination of Wolfe-Powell Line Search)**:**
Let Assumption 4.1 with $\Omega = \mathbb{R}^n$ be true. If $\nabla f(x_k)^\top d_k < 0$, then the Wolfe-Powell line search (Algorithm 4.10) terminates after a finite number of steps with a step size t_k that satisfies both Wolfe-Powell conditions.

Proof. In the first half of the algorithm we generate a t_- that satisfies $W1$ and a t_+ that violates $W1$ by either backtracking or fronttracking. Fronttracking must lead to a situation in which t_+ does not satisfy $W1$. Otherwise this would be a contradiction to f continuous and bounded from below.

To show the termination of backtracking for $t_- = 2^{-j}$, we assume the contrary: For all $j \in \mathbb{N}$ we have

$$f(x_k + 2^{-j} d_k) > f(x_k) + \sigma 2^{-j} \nabla f(x_k)^\top d_k$$

which implies

$$\lim_{j\to\infty} \frac{f(x_k + 2^{-j} d_k) - f(x_k)}{2^{-j}} \geq \sigma \nabla f(x_k)^\top d_k,$$

$$\text{or in the limit} \quad \underbrace{\nabla f(x_k)^\top d_k}_{<0} \geq \sigma(\nabla f(x_k)^\top d_k), \quad \text{leading to} \quad 1 \leq \sigma \qquad \Rightarrow \perp$$

In the refining loop we construct a interval of decreasing size $[t_-, t_+]$, whereas t_- satisfies $W1$ and t_+ violates $W1$ all the time. We write this as:

$$\begin{aligned} f(x_k + t_- d_k) - f(x_k) - \sigma t_- \nabla f(x_k)^\top d_k &\leq 0 \quad \text{and} \\ f(x_k + t_+ d_k) - f(x_k) - \sigma t_+ \nabla f(x_k)^\top d_k &> 0 \end{aligned}$$

This not only converges to a point t_* satisfying $f(x_k+t_*d_k)-f(x_k)-t_*\sigma\nabla f(x_k)^\top d_k = 0$, but also

$$\begin{aligned}\frac{d}{dt}(f(x_k+t_*d_k)-f(x_k)-\sigma t_*\nabla f(x_k)^\top d_k) \geq 0\\ \nabla f(x_k+t_*d_k)^\top d_k - \sigma\nabla f(x_k)^\top d_k \geq 0\\ \nabla f(x_k+t_*d_k)^\top d_k \geq \sigma\nabla f(x_k)^\top d_k > \rho\nabla f(x_k)^\top d_k\end{aligned}$$

$\square$

Exercise 4.12:
Perform Wolfe-Powell line search (Algorithm 4.10) for

$$\begin{aligned}\textit{minimize} \quad & f(u,v) = \frac{1}{2}(u-3)^2+v^2\\ s.t. \quad & (u,v)^\top \in \mathbb{R}^2\end{aligned}$$

with starting point $x_0=(1,1)^\top$, $d_0=-\nabla f(x_0)$ *and* $\sigma=\frac{3}{8}$ *and* $\rho=\frac{5}{8}$.

We know:

$$\begin{aligned}\nabla f(u,v) &= \begin{pmatrix} u-3\\ 2v\end{pmatrix}\\ d_0=-\nabla f(x_0) &= \begin{pmatrix} 2\\ -2\end{pmatrix}\\ x_0+td_0 &= (1+2t, 1-2t)^\top\\ f(x_0+td_0) &= 6t^2-8t+3\\ \nabla f(x_0+td_0) &= (-2+2t, 2-4t)^\top\end{aligned}$$

We formulate:

$$\begin{aligned}W1(t) &= f(x_0+td_0)-f(x_0)+t\sigma\nabla f(x_0)^\top\nabla f(x_0) = 6t^2-8(1-\sigma)t \overset{!}{\leq} 0\\ W2(t) &= -\nabla f(x_0+td_0)^\top\nabla f(x_0)+\rho\nabla f(x_0)^\top\nabla f(x_0) = 12t-8(1-\rho) \overset{!}{\geq} 0\end{aligned}$$

We test $t=1$ *and fail* $W1(1)=1$. *We execute backtracking and succeed with* $W1(\frac{1}{2}) = -1$. $W2(\frac{1}{2})=3$ *is also successful, so no refinement is required.*

Home Exercise 4.2 (Line Search Algorithms)**:**
Consider the function

$$f(x) = \sqrt{|x|}$$

a) *At* $x_0 = -1$ *show that* $d_0 = \frac{3}{2}$ *is a descent direction.*

b) *Compute all* $t_* > 0$ *that satisfy the sufficient decrease condition* $f(x_0 + t_* d_0) \leq f(x_0) + \sigma t_* \nabla f(x_0)^\top d_0$ *for the choice* $\sigma = \frac{1}{4}$.

c) *Compute all* $t_* > 0$ *that satisfy the sufficient steepness condition* $\nabla f(x_k + t_* d_k)^\top d_k \geq \rho \nabla f(x_k)^\top d_k$ *for the choice* $\rho = \frac{1}{2}$.

d) *Which requirements are missing for this objective to apply the Wolfe-Powell Termination theorem?*

e) *Compute iterates* t_a, t_d *resulting from golden section line search with initial data* $x_0, d_0, [t_a, t_d] = [0, 1]$ *until* $|t_d - t_a| < \frac{1}{4}$.

4.5 Descent Algorithm for Unconstrained Problems

With descent directions and line search methods we have gathered all ingredients for a descent optimization algorithm.

Algorithm 4.13 (Descent Algorithm for Unconstrained Problems)**:**
Let Assumption 4.1 be true with $\Omega = \mathbb{R}^n$.

1. *Input:* $f \in \mathcal{C}^1$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.
2. *Set* $x_k \leftarrow x_0$, *choose a mechanism to generate s.p.d. matrices* B_k.
3. *While* $||\nabla f(x_k)|| > \varepsilon$ *do*
   a) *Solve* $B_k d_k = -\nabla f(x_k)$ *for* d_k.
   b) *Find* t_k *such that* $f(x_k + t_k d_k) \leq f(x_k) + \sigma t_k \nabla f(x_k)^\top d_k$
      and
      $\nabla f(x_k + t_k d_k)^\top d_k \geq \rho \nabla f(x_k)^\top d_k$
   c) *Set* $x_k \leftarrow x_k + t_k d_k$, *update* B_k.
4. *Output:* $x_* \leftarrow x_k$.

With the choice $B_K = \mathbb{E} \Leftrightarrow d_k = -\nabla f(x_k)$, this algorithm is called **steepest descent algorithm**. Other choices for B_k are discussed later.

Exercise 4.14:
Solve

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = (u-3)^2 + 2v^2 \\ \textit{s.t.} \quad & (u,v)^\top \in \mathbb{R}^2 \end{aligned}$$

using Algorithm 4.13 with steepest descent, exact line search and starting point $x_0 = (1,1)^\top$ *until the termination criterion* $||\nabla f(x_k)|| \le \varepsilon := \sqrt{2}$ *holds.*

Check the gradient:

$$\nabla f(x_0) = \begin{pmatrix} 2u_0 - 6 \\ 4v_0 \end{pmatrix} = \begin{pmatrix} -4 \\ 4 \end{pmatrix}$$

and $||\nabla f(x_0)|| = 4\sqrt{2}$. *The first descent direction is*

$$d_0 = -\nabla f(x_0) = \begin{pmatrix} 4 \\ -4 \end{pmatrix}$$

and t_0 *satisfies*

$$0 \overset{!}{=} \big(2(u_0 + t_0(d_0)_1) - 6, \quad 4(v_0 + t_0(d_0)_2)\big) \begin{pmatrix} 4 \\ -4 \end{pmatrix} =$$

$$\big(-4 + 8t_0, \quad 4 - 16t_0\big) \begin{pmatrix} 4 \\ -4 \end{pmatrix} = -16 + 32t_0 - 16 + 64t_0 = 96t_0 - 32$$

so $t_0 = \frac{1}{3}$ *and* $x_1 = (1,1)^\top + \frac{1}{3}(4,-4)^\top = (\frac{7}{3}, -\frac{1}{3})^\top$.
Check the gradient:

$$\nabla f(x_1) = \begin{pmatrix} 2\frac{7}{3} - 6 \\ 4\frac{-1}{3} \end{pmatrix} = \begin{pmatrix} \frac{-4}{3} \\ \frac{-4}{3} \end{pmatrix}$$

and $||\nabla f(x_1)|| = \frac{4}{3}\sqrt{2}$. *The next step leads to*

$$d_1 = -\nabla f(x_1) = \begin{pmatrix} \frac{4}{3} \\ \frac{4}{3} \end{pmatrix}$$

and t_1 *satisfies*

$$0 \overset{!}{=} \big(2(u_1 + t_1(d_1)_1) - 6, \quad 4(v_1 + t_1(d_1)_2)\big) \begin{pmatrix} \frac{4}{3} \\ \frac{4}{3} \end{pmatrix} =$$

$$\big(\tfrac{-4}{3} + \tfrac{8}{3}t_1, \quad \tfrac{-4}{3} + \tfrac{16}{3}t_1\big) \begin{pmatrix} \frac{4}{3} \\ \frac{4}{3} \end{pmatrix} = \frac{-16}{9} + \frac{32}{9}t_1 + \frac{-16}{9} + \frac{64}{9}t_1 = \frac{-32+96t_1}{9}$$

so $t_1 = \frac{1}{3}$ *again and* $x_2 = (\frac{7}{3}, -\frac{1}{3})^\top + \frac{1}{3}(\frac{4}{3}, \frac{4}{3})^\top = (\frac{25}{9}, \frac{1}{9})^\top$.
Check the gradient:

$$\nabla f(x_2) = \begin{pmatrix} 2\frac{25}{9} - 6 \\ 4\frac{1}{9} \end{pmatrix} = \begin{pmatrix} \frac{-4}{9} \\ \frac{4}{9} \end{pmatrix}$$

and $||\nabla f(x_2)|| = \frac{4}{9}\sqrt{2} < \varepsilon$.

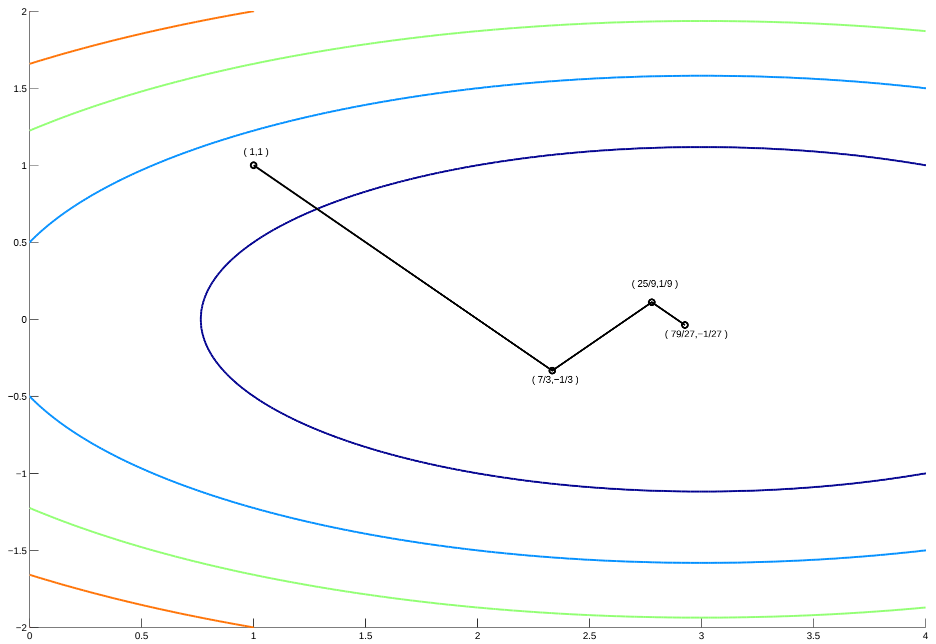

Figure 3: Path of the sequence x_k in Exercise 4.14

We now want to show that Algorithm 4.13 creates a sequence x_k for which every cumulation point is a (LMP):

Theorem 4.15 (Convergence of Descent Algorithms)**:**
Let Assumption 4.1 be true. Consider for $k \in \mathbb{N}_0$ the sequence

$$x_0 \in \mathbb{R}^n \qquad \text{and} \qquad x_{k+1} = x_k + t_k d_k \tag{4.19}$$

with $t_k > 0$ satisfying the sufficient decrease condition for $\sigma \in (0, \frac{1}{2})$:

$$f(x_k + t_k d_k) \leq f(x_k) + \sigma t_k \nabla f(x_k)^\top d_k \tag{4.20}$$

Furthermore let $d_k = -B_k^{-1} \nabla f(x_k)$ with $B_k \in \mathbb{R}^{n \times n}$ and let $\varepsilon, M > 0$ exist such that for all $k \in \mathbb{N}_0$ and $y \in \mathbb{R}^n$ holds: B_k is s.p.d. and $\varepsilon ||y||^2 < y^\top B_k^{-1} y < M ||y||^2$. Then

$$\lim_{k \to \infty} \nabla f(x_k) = 0 \tag{4.21}$$

and every cumulation point $x_* \in \mathbb{R}^n$ satisfies $\nabla f(x_*) = 0$.

Proof. Due to the sufficient decrease condition the sequence $\{f(x_k)\}_{k \in \mathbb{N}_0}$ is non-increasing and, because f is bounded from below, converges to $f_* \in \mathbb{R}$. Especially

$$\lim_{k\to\infty} f(x_k) - f(x_{k+1}) = 0 \tag{4.22}$$

but also

$$f(x_{k+1}) - f(x_k) \leq \sigma t_k \nabla f(x_k)^\top d_k = -\sigma t_k \nabla f(x_k)^\top B_k^{-1} \nabla f(x_k) \leq -\sigma\varepsilon t_k ||\nabla f(x_k)||^2$$
$$||\nabla f(x_k)||^2 \leq \frac{f(x_k) - f(x_{k+1})}{\sigma\varepsilon t_k}$$

It can be shown[1] that t_k is always greater than a positive value depending on σ and M. So in the limit holds:

$$\lim_{k\to\infty} ||\nabla f(x_k)||^2 \leq 0 \tag{4.23}$$

□

The theorem does not guarantee the existence of cumulation points, try for example to minimize $\exp(x)$, so additional properties like uniform convexity, coercivity or constraints are necessary. On the other hand, the second Wolfe-Powell condition (sufficient steepness) is not required for convergence.

Home Exercise 4.3 (Steepest Descent with Wolfe Powell Line Search)**:**
Consider the quadratic problem

$$\textit{minimize} \quad f(x) = \frac{1}{2} x^\top A x \quad \textit{with} \quad A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$
$$\textit{s.t.} \quad x = \in \mathbb{R}^2$$

a) *Perform two steepest descent steps with Wolfe Powell line search (see Algorithms 4.13 and 4.10) starting at $x_0 = (1, 0)^\top$ and $\sigma = \frac{1}{4}, \rho = \frac{1}{2}$ to get x_1 and x_2.*

b) *Check if x_2 is a (GMP) of this problem.*

c) *Use the second order sufficient optimality conditions to find the solution x_* of this problem.*

4.6 Projected Descent for Box Constraints

Let us now turn our attention back to box constraints. We want to update Algorithm 4.13 in such a way, that it also works for optimization problems with $\Omega = \Omega_\Box$. The

[1] compare Boyd & Vandenberghe: *Convex Optimization*, section *analysis for backtracking line search*

update $x_{k+1} = x_k + t_k d_k$ must obviously be changed to

$$x_{k+1} = P(x_k + t_k d_k) = P(x_{k+1}(t_k)), \tag{4.24}$$

whereas $P : \mathbb{R}^n \to \Omega_\square$ is the known projection from Definition 2.11, d_k is some descent direction and t_k is computed using line search methods reducing $\phi(t) = f(P(x_{k+1}(t)))$.

We can now think of combining the objective and the projection to

$$\begin{aligned} \text{minimize} \quad & f(P(x)) \\ \text{s.t.} \quad & x \in \mathbb{R}^n \end{aligned}$$

and use the theory for unconstrained problems. But the main flaw here is, that $f(P(x))$ is **not continuously differentiable** at every x with $\mathcal{A}(x) \neq \{\}$ and all the proofs fail.

We therefore have to use a slightly worse step size mechanic:

Definition 4.16 (Projected Backtracking)**:**
For the line search problem

$$\begin{aligned} \textit{minimize} \quad & \phi(t) := f(x_k + t d_k) \\ \textit{s.t.} \quad & t \in (0, 1] \end{aligned}$$

with $\nabla f(x_k)^\top d_k < 0$ *(descent direction) a step size* t_* *satisfies the* ***sufficient decrease condition for box constraints*** *with respect to* $\sigma \in (0, \frac{1}{2})$ *if*

$$f(P(x_k + t_* d_k)) \leq f(x_k) - \frac{\sigma}{t_*} ||x_k - P(x_k - t_* \nabla f(x_k))||^2 \tag{4.25}$$

It can be shown that under Assumption 4.1 with $\Omega = \Omega_\square$ a projection step size $t_* \in (0, 1]$ can always be found. The corresponding algorithm is:

Algorithm 4.17 (Projected Backtracking Line Search)**:**
For reducing $\phi(t) = f(x_k + t d_k)$ *in the interval* $(0, 1]$.

1. *Input:* f, x_k, d_k*; choose* $\sigma \in (0, \frac{1}{2})$.
2. *If* $\nabla f(x_k)^\top d_k \geq 0$, *return error (descent direction check fails).*
3. *Define* $W1(t) = f(P(x_k + t d_k)) \leq f(x_k) - \frac{\sigma}{t} ||x_k - P(x_k - t \nabla f(x_k))||^2$ *(bool-valued function).*
4. *Set* $t \leftarrow 1$.
5. *While* $W1(t) == FALSE$ *do* ***backtracking****:*

   a) *Set* $t \leftarrow \frac{t}{2}$.

6. *Output:* $t_* \leftarrow t$.

This algorithm only performs backtracking, but can handle points with active box constraints. The resulting algorithm is:

Algorithm 4.18 (Projected Descent Algorithm for Box Constraints)**:**
Let Assumption 4.1 be true with $\Omega = \Omega_\Box$.

1. *Input:* $f \in \mathcal{C}^1$; $P : \mathbb{R}^n \to \Omega$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.
2. *Set* $x_k \leftarrow P(x_0)$, *choose a mechanism to generate s.p.d. matrices* B_k.
3. *While* $||x_k - P(x_k - \nabla f(x_k))|| > \varepsilon$ *do*
   a) *Reduce* B_k *to* $B_{k,\Omega}$ *depending on the active set* $\mathcal{A}(x_k)$.
   b) *Solve* $B_{k,\Omega} d_k = -\nabla f(x_k)$ *for* d_k.
   c) *Find* t_k *such that*
   $$f(P(x_k + t_k d_k)) \leq f(x_k) - \frac{\sigma}{t_k} ||x_k - P(x_k + t_k d_k)||^2$$
   d) *Set* $x_k \leftarrow P(x_k + t_k d_k)$, *update* B_k.
4. *Output:* $x_* \leftarrow x_k$.

For the choice $B_k = \mathbb{E}$ we get the **projected steepest descent method**.

Theorem 4.19 (Finite Termination with Correct Active Set)**:**
Let ∇f *be Lipschitz continuous with Lipschitz constant* L. *Assume there is* $M, \bar{\kappa}$ *such that the matrices* B_k *are s.p.d. with* $||B_k|| < M$ *and the condition numbers* $\kappa(B_k) \leq \bar{\kappa}$. *Then* $||x_k - P(x_k - \nabla f(x_k))|| \to 0$ *for* $k \to \infty$ *and any limit point of* $\{x_k\}$ *is stationary. If* x_* *is nondegenerate, then* $\mathcal{A}(x_k) = \mathcal{A}(x_*)$ *after finitely many steps.*

Exercise 4.20:
For the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = \frac{1}{2} || \begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} -1 \\ 0.5 \end{pmatrix} ||^2 \\ \textit{s.t.} \quad & x = (u,v)^\top \in [0,1] \times [0,1] \end{aligned}$$

execute Algorithm 4.18 to compute x_* using the choice $B_k = \mathbb{E}$ with starting point $x_0 = (0,0)^\top$ and ε sufficiently small. Instead of using projected backtracking line search, always set $t_k = 1$.

We compute

$$\nabla f(x) = \begin{pmatrix} u+1 \\ v-0.5 \end{pmatrix}$$

and

$$||x_0 - P(x_0 - \nabla f(x_0))|| = ||P(-\nabla f(x_0))|| = ||P(\begin{pmatrix} -1 \\ 0.5 \end{pmatrix})|| = || \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} || = 0.5 > \varepsilon$$

Because $t_0 = 1$, $x_1 = (0, 0.5)^\top$. Next we look at

$$||x_1 - P(x_1 - \nabla f(x_1))|| = || \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} - P(\begin{pmatrix} 0 \\ 0.5 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix})|| = ||0|| < \varepsilon$$

So $x_1 = x_*$.

Home Exercise 4.4 (Projected Steepest Descent with Projected Backtracking Line Search)**:**
Consider the quadratic problem

$$\begin{aligned} &\text{minimize} \quad f(x) = \frac{1}{2} x^\top A x \quad \text{with} \quad A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \\ &\text{s.t.} \quad x =\in \Omega_\square := [1,2] \times [-1,1] \end{aligned}$$

- **a)** *Perform two projected steepest descent steps with projected backtracking line search (see Algorithms 4.18 and 4.17) starting at $x_0 = (1,0)^\top$ and $\sigma = \frac{1}{4}$ to get x_1 and x_2.*
- **b)** *Check the second order sufficient optimality conditions to decide if x_2 is a non-degenerate (LMP) of this problem.*
- **c)** *How can the problem be relaxed to a solution equivalent problem to show that x_2 is a nondegenerate (LMP)?*

5 Descent Direction Choice and Convergence

With the choice $d_k = -\nabla f(x_k)$ in Algorithm 4.13 or in Algorithm 4.18 we end up with the (projected) steepest descent method. Steepest descent methods converge to (LMPs), but they are still a bad choice, because they converge slowly.

The slow convergence rate can be observed: A typical steepest descent path is oscillating in a zig-zag behavior. This **zig-zagging effect** always occurs and is based on the fact that two consecutive steepest descent directions under exact line search are always orthogonal:

$$\nabla\phi(t_k) = \nabla f(x_k + t_k d_k)^\top d_k = -{d_{k+1}}^\top d_k \overset{!}{=} 0 \tag{5.1}$$

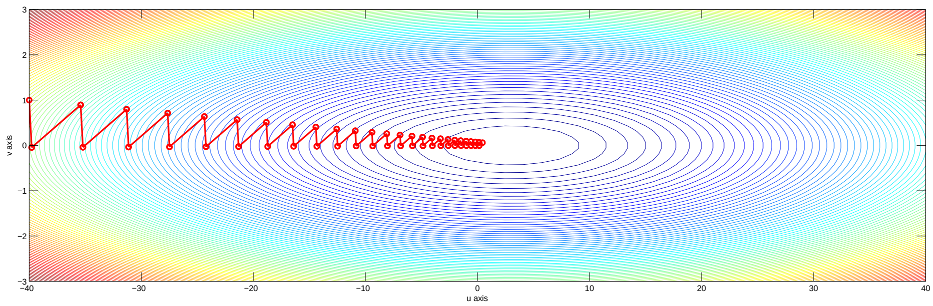

Figure 4: Zig-Zagging path of a sequence x_k generated by steepest descent (scaled)

5.1 Q-Convergence Rates

Before we introduce choices for d_k that lead to faster algorithms, we want to quantify the speed of convergence and therefore the performance of a descent algorithm. We therefore define quotient convergence rates:

Definition 5.1 (Quotient Convergence)**:**
Let $\{x_k\}_{k\in\mathbb{N}} \subset \mathbb{R}^n$ be a sequence converging to $x_* \in \mathbb{R}^n$, i.e. for all $\varepsilon > 0$ exists $K \in \mathbb{N}$ such that the distance $||x_k - x_*|| < \varepsilon$ for all $k > K$. We use the abbreviation $\Delta_k := ||x_k - x_*||$ and say that

1. *x_k converges Q-linearly to x_*, if there is $\sigma \in (0,1)$ such that*

$$\frac{\Delta_{k+1}}{\Delta_k} \leq \sigma \qquad \text{for all } k \text{ sufficiently large,} \tag{5.2}$$

2. *x_k converges Q-superlinearly to x_*, if*

$$\lim_{k\to\infty} \frac{\Delta_{k+1}}{\Delta_k} = 0, \tag{5.3}$$

3. x_k *converges Q-quadratically to* x_**, if there is* $\mu > 0$ *such that*

$$\frac{\Delta_{k+1}}{\Delta_k^2} \leq \mu \quad \textit{for all } k \textit{ sufficiently large.} \tag{5.4}$$

Example 5.2:
Determine the Q-convergence rate of the sequence

$$x_k = 1 + k^{-(k^p)} \quad \textit{for} \quad k \in \mathbb{N} \quad \textit{for the parameter} \quad p \in \{0, 1\}.$$

We see $\lim_{k\to\infty} x_k = 1 =: x_*$ *and* $\Delta_k = k^{-(k^p)}$*. The Q-convergence rate is determined by first looking at the quotient:*

$$\frac{\Delta_{k+1}}{\Delta_k} = \frac{(k+1)^{-((k+1)^p)}}{k^{-(k^p)}} = \frac{k^{(k^p)}}{(k+1)^{((k+1)^p)}} = \begin{cases} \frac{k}{(k+1)} & \textit{for} \quad p = 0 \\ \frac{k^k}{(k+1)^{(k+1)}} = \frac{1}{k+1}\left(\frac{k}{k+1}\right)^k & \textit{for} \quad p = 1 \end{cases}$$

For $p = 0$ *we see that the convergence rate is slower than Q-linear. Higher order convergence rates are out of question.*

For $p = 1$ *we realize*

$$\lim_{k\to\infty} \underbrace{\frac{1}{k+1}}_{\to 0} \underbrace{\left(\frac{k}{k+1}\right)^k}_{\to e^{-1}} = 0$$

so we have a Q-superlinear (implying Q-linear) convergence rate. In order to decide, if Q-quadratic convergence rate applies, we look at

$$\frac{\Delta_{k+1}}{\Delta_k^2} = \frac{k^{2k}}{(k+1)^{(k+1)}} = \frac{k^k}{k+1}\left(\frac{k}{k+1}\right)^k.$$

The term $\frac{k^k}{k+1}$ *is unbounded for* $k \to \infty$*, so the convergence rate is lower than Q-quadratic.*

Home Exercise 5.1 (Q-Convergence)**:**
Decide if the following sequences converge Q-linearly, Q-superlinearly or Q-quadratically to zero:

a) $a_k = \exp(-k)$.

b) $b_k = \sqrt{k^{-1}}$.

c) $c_k = \frac{1}{k!}$.

d) *For the golden section line search iterates the length of the search interval behaves as follows:* $\Delta t_0 = 1$ *and* $\Delta t_{k+1} = \frac{\sqrt{5}-1}{2}\Delta t_k$. *Show that this sequence converges Q-linearly to zero and estimate* σ.

We want to show the following result for steepest descent methods:

Corollary 5.3 (Q-Linear Convergence of Steepest Descent)**:**
Consider minimizing a quadratic program $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ *on* $\mathbb{R}^n$. *Then the exact line search step size for the quadratic program with steepest descent is*

$$t_k = \frac{\nabla f_k{}^\top \nabla f_k}{\nabla f_k{}^\top A \nabla f_k}, \tag{5.5}$$

but the sequence

$$x_{k+1} = x_k - t_k \nabla f(x_k) \tag{5.6}$$

converges to $x_* = A^{-1}b$ *only Q-linearly with* $\sigma \in (0,1)$*:*

$$||x_{k+1} - x_*||_A \leq \sigma ||x_k - x_*||_A \tag{5.7}$$

The norm $||x||_A := \sqrt{x^\top Ax}$ *is induced by the s.p.d. matrix* A.

Proof. For the exact line search step size see Lemma 3.7 with $d_k = -\nabla f(x_k) = -(Ax_k - b)$. To check the Q-convergence rate of the sequence, we have analyze the distance $||x_{k+1} - x_*||_A$ for $k \to \infty$:

$$\begin{gathered} ||x_{k+1} - x_*||_A^2 = (x_{k+1} - x_*)^\top A(x_{k+1} - x_*) = \\ (x_k - t_k\nabla f_k - x_*)^\top A(x_k - t_k \nabla f_k - x_*) \\ = (x_k - x_*)^\top A(x_k - x_*) - t_k \nabla f_k{}^\top A(x_k - x_*) - t_k(x_k - x_*)^\top A \nabla f_k + t_k^2 \nabla f_k{}^\top A \nabla f_k \end{gathered}$$

Notice that if x_* is a (LMP), then $\nabla f(x_*) = 0 \Leftrightarrow Ax_* = b$. So:

$$\nabla f_k = Ax_k - b = Ax_k - Ax_* = A(x_k - x_*) \quad \text{and} \quad A^{-1}\nabla f_k = x_k - x_* \tag{5.8}$$

$$\begin{aligned} ||x_{k+1} - x_*||_A^2 &= ||x_k - x_*||_A^2 - t_k \nabla f_k{}^\top \nabla f_k - t_k \nabla f_k{}^\top \nabla f_k + t_k^2 \nabla f_k{}^\top A \nabla f_k && (5.9) \\ &= ||x_k - x_*||_A^2 - t_k \nabla f_k{}^\top \nabla f_k - t_k \nabla f_k{}^\top \nabla f_k + t_k \nabla f_k{}^\top \nabla f_k && (5.10) \\ &= \left(1 - t_k \frac{\nabla f_k{}^\top \nabla f_k}{||x_k - x_*||_A^2}\right) ||x_k - x_*||_A^2 && (5.11) \end{aligned}$$

We use

$$||x_k - x_*||_A^2 = (x_k - x_*)^\top A(x_k - x_*) = \nabla f_k{}^\top (x_k - x_*) = \nabla f_k{}^\top A^{-1} \nabla f_k \tag{5.12}$$

and get

$$||x_{k+1} - x_*||_A^2 = \left(1 - \frac{||\nabla f_k||^2}{||\nabla f_k||_A^2} \frac{||\nabla f_k||^2}{||\nabla f_k||_{A^{-1}}^2}\right) ||x_k - x_*||_A^2 \tag{5.13}$$

We can show for s.p.d. matrices A with eigenvalues $0 < \lambda_{min} \leq \lambda_i \leq \lambda_{max}$:

$$||\nabla f_k||_A^2 \leq \lambda_{max} \cdot ||\nabla f_k||^2 \tag{5.14}$$

$$||\nabla f_k||_A^2 ||\nabla f_k||_{A^{-1}}^2 \leq \lambda_{max} \cdot ||\nabla f_k||^2 \frac{1}{\lambda_{min}} \cdot ||\nabla f_k||^2 \tag{5.15}$$

$$\frac{\lambda_{min}}{\lambda_{max}} \leq \frac{||\nabla f_k||^2}{||\nabla f_k||_A^2} \frac{||\nabla f_k||^2}{||\nabla f_k||_{A^{-1}}^2} \tag{5.16}$$

$$1 - \frac{\lambda_{min}}{\lambda_{max}} \geq 1 - \frac{||\nabla f_k||^2}{||\nabla f_k||_A^2} \frac{||\nabla f_k||^2}{||\nabla f_k||_{A^{-1}}^2} \tag{5.17}$$

This leads to

$$||x_{k+1} - x_*||_A^2 \leq \underbrace{\left(\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max}}\right)}_{:=\sigma^2} ||x_k - x_*||_A^2 \tag{5.18}$$

or

$$||x_{k+1} - x_*||_A \leq \sigma ||x_k - x_*||_A \tag{5.19}$$

Using the Kantorovich inequality a better estimate is possible, leading to

$$||x_{k+1} - x_*||_A \leq \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} ||x_k - x_*||_A \tag{5.20}$$

□

5.2 Newton's Method

The slow Q-linear convergence rate of steepest descent motivates more sophisticated choices for d_k. A very good choice with Q-quadratic convergence rate is the **exact Newton descent** $d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$. But the requirements to execute this are stricter. First we assume:

Assumption 5.4 (Twice Continuously Differentiable Objective)**:**
For the nonlinear objective $f : \mathbb{R}^n \to \mathbb{R}$ *holds:*

1. *$f \in \mathcal{C}^2$ and the Hessian is Lipschitz continuous: There is $L > 0$ such that $||\nabla^2 f(x) - \nabla^2 f(y)|| \leq L||x - y||$ for all $x, y \in \mathbb{R}^n$.*
2. *$\nabla^2 f(x)$ is s.p.d. for $x \in \mathcal{B}_\delta(x_*)$, where x_* solves $\nabla f(x_*) = 0$ and $\delta > 0$.*

We can interpret the steepest descent step $d_k = -\nabla f(x_k)$ as approach to minimize the local linear model of f at x_k (compare Definition 4.8)

$$L_f(x; x_k) = f(x_k) + \nabla f(x_k)^\top (x - x_k) \tag{5.21}$$

For an exact Newton step we want to minimize the local quadratic model $Q_f(x; x_k)$ of f at x_k:

$$Q_f(x; x_k) = f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k) \tag{5.22}$$

Because the quadratic model is a quadratic program on $\mathbb{R}^n$, we know that the (GMP) x_*^Q solves

$$\nabla^2 f(x_k)(x_*^Q - x_k) = -\nabla f(x_k) \tag{5.23}$$

We can then choose $d_k = x_*^Q - x_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$. Sadly, Newton steps only work reliably if the Hessian $\nabla^2 f(x_k)$ is s.p.d., otherwise we do not necessarily get a descent direction.

Assumption 5.4 covers s.p.d. of $\nabla^2 f(x)$ only in the environment of the solution x_*. The Q-quadratic convergence is therefore only guaranteed locally in the environment $\mathcal{B}_\delta(x_*)$:

Theorem 5.5 (Exact Newton Step)**:**
*Let Assumption 5.4 be true. Then for sufficiently small $\delta > 0$ there is $K > 0$ such that for $x_k \in B_\delta(x_*)$ and x_{k+1} is generated with a **exact Newton step***

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \tag{5.24}$$

*we get **locally Q-quadratic convergence**:*

$$||x_{k+1} - x_*|| \leq K||x_k - x_*||^2. \tag{5.25}$$

Proof. From Lemma 2.1 with $d = x_k - x_*$ we know

$$\nabla f(x_k) = \underbrace{\nabla f(x_*)}_{=0} + \int_0^1 \nabla^2 f\left(x_* + t(x_k - x_*)\right)(x_k - x_*)\,\mathrm{d}t$$

Now for $\delta > 0$ sufficiently small we have

$$
\begin{aligned}
x_{k+1} - x_* &= x_k - x_* - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \\
&= \nabla^2 f(x_k)^{-1} \big(\nabla^2 f(x_k) \cdot (x_k - x_*) - \nabla f(x_k)\big) \\
&= \nabla^2 f(x_k)^{-1} \int_0^1 \big(\nabla^2 f(x_k) \cdot (x_k - x_*) - \nabla^2 f\left(x_* + t(x_k - x_*)\right) \cdot (x_k - x_*)\big)\ \mathrm{d}t
\end{aligned}
$$

By the Lipschitz continuity of $\nabla^2 f(\cdot)$ and

$$
\int_0^1 \nabla^2 f(x_* + t(x_k - x_*))\ \mathrm{d}t = \int_0^1 \nabla^2 f(x_k + \tau(x_* - x_k))\ \mathrm{d}\tau \qquad \text{(substitute } \tau = 1 - t\text{)}
$$

we obtain with $M_\delta := \max_{x \in B_\delta(x_*)} ||\nabla^2 f(x)^{-1}||$

$$
||x_{k+1} - x_*|| \le ||\nabla^2 f(x_k)^{-1}|| \cdot \int_0^1 ||\nabla^2 f(x_k) - \nabla^2 f\left(x_k + \tau(x_* - x_k)\right)||\ \mathrm{d}\tau ||x_k - x_*|| \tag{5.26}
$$

$$
\le ||\nabla^2 f(x_k)^{-1}|| \cdot \int_0^1 L||\tau(x_* - x_k)||\ \mathrm{d}\tau ||x_k - x_*|| \tag{5.27}
$$

$$
\le ||\nabla^2 f(x_k)^{-1}|| \cdot \frac{L}{2} ||(x_* - x_k)||^2 \le \underbrace{\frac{L}{2} M_\delta}_{=:K} ||x_k - x_*||^2 \tag{5.28}
$$

□

The corresponding algorithm is

Algorithm 5.6 (Newton Descent)**:**
For solving locally convex nonlinear programs using the Hessian

1. *Input:* $f \in \mathcal{C}^2$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.
2. *Set* $x_k \leftarrow x_0$.
3. *While* $||\nabla f(x_k)|| > \varepsilon$ *do:*
   a) *Solve* $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$ *for* d_k *(exact Newton direction).*
   b) *Set* $x_k \leftarrow x_k + d_k$.
4. *Output:* $x_* \leftarrow x_k$.

Remark:

1. The conjugate gradient solver (see Algorithm 3.12) can be used to solve $\nabla^2 f(x_k)d_k = -\nabla f(x_k)$. Useage of preconditioners is recommended (see Algorithm 11.1).
2. The computation of the Hessian $\nabla^2 f(x)$ is often expensive. If $\nabla^2 f(x)$ cannot be computed offline analytically, then $\nabla^2 f(x)$ has to be approximated. In Algorithm 11.5 (unconstrained) and Algorithm 11.6 (box constraints) this is solved by using central differences, but the result is not the full matrix $\nabla^2 f$, but only $\nabla^2 f \cdot d$. We address this later in Algorithm 6.3.
3. Insufficient approximation of the Hessian, like e.g. $\nabla^2 f(x_k) \approx \nabla^2 f(x_0)$ for all k, denies the quadratic convergence but still leads to linear convergence if the gradient is not perturbed.
4. In practical application, if $\nabla^2 f(x_k)d_k = -\nabla f(x_k)$ does not lead to a descent direction (non s.p.d. hessian), the algorithm is switched to a globally reliable technique like steepest descent for the current step.
   A check like $-\nabla f(x_k)^\top d_k \geq \varepsilon ||d_k||^2$ is used to verify that d_k is a valid descent direction.

$\square$

Exercise 5.7:
For the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = \frac{1}{3}u^3 - uv + v^2 \\ \textit{s.t.} \quad & x = (u,v)^\top \in \mathbb{R}^2 \end{aligned}$$

compute x_2 *using exact Newton's method with* $x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ *and again with* $x_3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

The gradient is

$$\nabla f(u,v) = \begin{pmatrix} u^2 - v \\ -u + 2v \end{pmatrix} \tag{5.29}$$

and the Hessian is

$$\nabla^2 f(u,v) = \begin{pmatrix} 2u & -1 \\ -1 & 2 \end{pmatrix} \tag{5.30}$$

Now $\nabla f_0 = (1, -1)^\top$ *and the system*

$$\nabla^2 f_0 d_0 = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} d_0 = -\begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{5.31}$$

is solved by $d_0 = \frac{1}{3}(-1,1)^\top$*, so* $x_1 = x_0 + d_0 = \frac{1}{3}(2,1)^\top$*.*

Next $\nabla f_1 = \frac{1}{9}(1,0)^\top$ *and the system*

$$\nabla^2 f_1 d_1 = \frac{1}{3}\begin{pmatrix} 4 & -3 \\ -3 & 6 \end{pmatrix} d_1 = -\frac{1}{9}\begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{5.32}$$

is solved by $d_1 = \frac{1}{15}(-2,-1)^\top$*, so* $x_2 = x_1 + d_1 = \frac{1}{15}(8,4)^\top$*. We assume that the algorithm converges to the (LMP)* $x_* = \frac{1}{4}(2,1)^\top$*.*

For starting point $x_3 = (0,0)^\top$ *we cannot find a descent direction (saddle point).*

Home Exercise 5.2 (Newton's Method)**:**
Consider the problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = u^5 + 2v^2 - 8uv \\ \textit{s.t.} \quad & x = (u,v)^\top \in \mathbb{R}^2 \end{aligned}$$

a) *Use the leading principal minors to show: If* $u > \sqrt[3]{\frac{4}{5}}$ *and* $v \in \mathbb{R}$ *then the Hessian* $\nabla^2 f(u,v)$ *is s.p.d.*

b) *Starting at* $x_0 = (1,0)^\top$ *perform one Newton step to get* x_1*.*

c) *Check the second order optimality conditions at the point* $x_* = (0,0)^\top$*.*

5.3 Trust Region Methods

Up to now we discussed that the steepest descent method results from minimizing a local linear model, but is slow and requires line search. And that the exact Newton descent method results from minimizing a local quadratic model, but requires an s.p.d. environment to find a valley point. A mixture of both ideas are **trust region methods**: We consider a originally unconstrained problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{s.t.} \quad & x \in \mathbb{R}^n \end{aligned}$$

and choose the descent direction at some x_k in the following way: We define a trust region radius δ_k leading to the closed ball set $\Omega_{\delta_k} = \{x \in \mathbb{R}^n : ||x - x_k|| \le \delta_k\}$. We now use again a local quadratic model to approximate f on Ω_{δ_k} and solve the **local quadratic program subject to the trust region**:

$$\begin{aligned} \text{minimize} \quad & Q_f(x; x_k) = f(x_k) + \nabla f(x_k)^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k) \\ \text{s.t.} \quad & x \in \Omega_{\delta_k} \end{aligned}$$

to get the solution x_{δ_k} or some close approximation using a projection method. We measure the quality of the descent using a sufficient decrease condition:

$$\frac{f(x_k) - f(x_{\delta_k})}{Q_f(x_k; x_k) - Q_f(x_{\delta_k}; x_k)} =: \sigma_{\delta_k} \tag{5.33}$$

In this approach the Hessian $\nabla^2 f(x_k)$ is not required to be s.p.d. at all. Instead of the closed ball $\Omega_{\delta_k} = \{x \in \mathbb{R}^n : \ ||x - x_k|| \leq \delta_k\}$, where the projection is discussed in Example 2.10, we can in theory also use box constraints. The descent quality σ_{δ_k} is used in the following algorithm:

Algorithm 5.8 (Trust Region Method)**:**
For minimizing $f(x)$, s.t. $x \in \mathbb{R}^n$.

1. *Input: $f \in \mathcal{C}^2$; $x_0 \in \mathbb{R}^n$; $\delta_0 > 0$, $\varepsilon > 0$.*
2. *Set $x_k \leftarrow x_0$, set $\delta_k \leftarrow \delta_0$.*
3. *While $||\nabla f(x_k)|| > \varepsilon$ do*
   a) *Initialize quadratic model $Q_f(x; x_k)$ at x_k, set $\sigma_{\delta_k} \leftarrow 0$.*
   b) *While $\sigma_{\delta_k} \leq 0$ do*
      i. *Find x_{δ_k} as exact or approximate (GMP) of $Q_f(x; x_k)$ on Ω_{δ_k}.*
      ii. *Compute σ_{δ_k} using (5.33).*
      iii. *If $\sigma_{\delta_k} < \frac{1}{4}$, set $\delta_k \leftarrow \frac{1}{4}\delta_k$.*
      iv. *Else if $\sigma_{\delta_k} > \frac{3}{4}$ and $||x_{\delta_k} - x_k|| = \delta_k$, set $\delta_k \leftarrow 2\delta_k$.*
   c) *Set $x_k \leftarrow x_{\delta_k}$.*
4. *Output: $x_* \leftarrow x_k$.*

The convergence rate of this algorithm relies heavily on the choice of finding x_{δ_k} as approximate or exact (GMP) of $Q_f(x; x_k)$ on Ω_{δ_k}:

- Choosing the exact (GMP) is called **exact Trust Region step**.
- If x_* is in the trust region and $\nabla^2 f$ is s.p.d., exact Trust Region step equals an exact Newton step in terms of effort and convergence rate.
- Choosing the approximate (GMP) to be the result of projected line search in direction $-\nabla f(x_k)$ leads to the so called **Cauchy point**, but inherits the Q-linear convergence rate from steepest descent. It should only be accepted, if the Cauchy point is on the boundary of the trust region.
- If $\nabla^2 f$ is symmetric indefinite, the solution x_* of $\nabla^2 f(x_k)(x_* - x_k) = -\nabla f(x_k)$ (saddle point) can still lead to a decrease and can be accepted as approximate

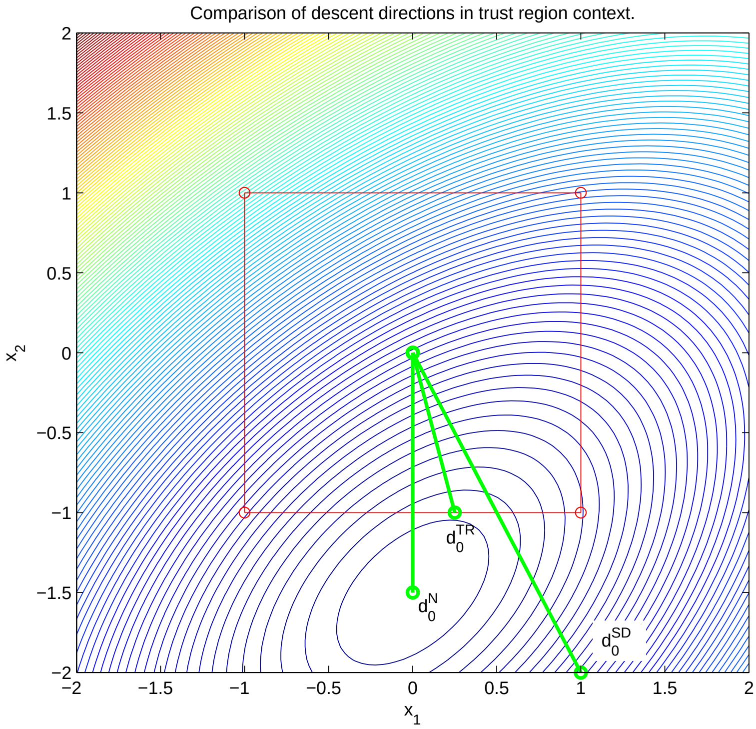

Figure 5: Comparison of descent directions in Example 5.9

(GMP). This point is called **Newton point**.

- **Dogleg methods** compute both the **Cauchy point** and the **Newton point** and search for the best candidate minimizing the model on the path x_k to Cauchy point to Newton point without leaving Ω_{δ_k} (see literature).

Example 5.9:
Look at some function f with the quadratic model

$$Q_f(x; x_0) = f_0 + \nabla f_0{}^\top (x - x_0) + \frac{1}{2}(x - x_0)^\top \nabla^2 f_0 (x - x_0) \tag{5.34}$$

$$= \begin{pmatrix} -3 & 6 \end{pmatrix} x + \frac{1}{2} x^\top \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix} x \tag{5.35}$$

at $x_0 = (0,0)^\top$. We can compute the following descent directions for f at x_0:

- *The steepest descent direction is $d_0^{SD} := -\nabla f_0 = (3, -6)^\top$.*

- *The Newton step direction is $d_0^N := -\nabla^2 f_0^{-1} \nabla f_0 = (0, -\frac{3}{2})^\top$.*

- *If the trust region radius* δ_0 *is large enough to contain the minimizer of the quadratic model, the trust region direction* d_0^{TR} *equals* d_0^N*.*

- *If* δ_0 *is smaller (see figure),* d_0^{TR} *leads to the (GMP) of the trust region subproblem at the boundary of the trust region.*

Home Exercise 5.3 (Trust Region with Constraints)**:**
Consider the problem

$$\begin{aligned} \textit{minimize} \quad & f(x) = -\frac{1}{24}x^3 + \frac{3}{4}x^2 - 4x + 6 \\ \textit{s.t.} \quad & x \in \Omega_\square := [0, 9] \end{aligned}$$

a) *Formulate the quadratic model* $Q_f(x; x_k)$ *of* f *at the general point* $x_k \in \mathbb{R}$*.*

b) *Perform an exact trust region step at* $x_0 = 8$ *with* $\Omega_0 = [6, 10]$ *to get* x_1 *and compute* σ_1*. The (GMP) of the model is not unique, use the smallest* x*-value.*

c) *Perform an exact trust region step at* $x_1 = 6$ *with* $\Omega_1 = [4, 8]$ *to get* x_2 *and compute* σ_2*.*

d) *Perform an exact trust region step at* $x_2 = 4$ *with* $\Omega_2 = [2, 6]$ *to get* x_3*.*

6 Newton-type Methods

Newton-type methods are a class of algorithms using reduced Hessian information for computing a descent direction. These methods have superlinear or quadratic convergence rates, but require smoothness and local convexity of the objective.

6.1 Inexact Newton Methods

For the exact Newton steps we obviously require

$$\nabla^2 f(x_k)d_k + \nabla f(x_k) = 0, \tag{6.1}$$

to find a descent direction. We relax this requirement as follows:

Definition 6.1 (Inexact Newton Step)**:**
*A descent direction d_k of f at x_k satisfies the **inexact Newton condition**, if for some $0 < \theta_k < 1$ holds:*

$$||\nabla^2 f(x_k)d_k + \nabla f(x_k)|| \leq \theta_k ||\nabla f(x_k)|| \tag{6.2}$$

Theorem 6.2 (Inexact Newton Convergence Rate)**:**
Let Assumption 5.4 be true. Then for sufficiently small $\delta > 0$ there is $K > 0$ such that for $x_k \in B_\delta(x_*)$ and $x_{k+1} = x_k + d_k$ with

$$||\nabla^2 f(x_k)d_k + \nabla f(x_k)|| \leq \theta_k ||\nabla f(x_k)|| \tag{6.3}$$

we get:

$$||x_{k+1} - x_*|| \leq K(||x_k - x_*|| + \theta_k)||x_k - x_*||. \tag{6.4}$$

It depends now on θ_k which convergence dominates in inexact Newton methods (compare Definition 5.1):

1. If θ_k is small enough to guarantee $K\theta_k < 1$ for sufficiently large k we get Q-linear convergence rate.
2. If $\lim_{k\to\infty} \theta_k \to 0$ the convergence rate is Q-superlinear.
3. If there is some $\mu > 0$ such that $\theta_k \leq \mu||\nabla f(x_k)||$ for sufficiently large k the convergence rate is Q-quadratic.

With this result in mind we design an algorithm that finds descent directions satisfying inequality (6.3) whenever possible with tolerance $\theta_k := \min(\frac{1}{2}, \sqrt{||\nabla f_k||})$. We store this in $\eta_k := \min(\frac{1}{2}, \sqrt{||\nabla f_k||}) \cdot ||\nabla f_k||$. At the same time we want to reduce the effort to compute the full Hessian and instead use directional Hessian approximations that return the vectors $d_H \approx \nabla^2 f(x_k) d_k$. A special CG-approach to solve $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$ is implemented, that works with knowing only d_H. We also add a line search method like Wolfe-Powell to take care of nonconvex areas of the objective, whenever the positive curvature check $\rho_k > \varepsilon ||d_k||^2$ fails:

Algorithm 6.3 (Inexact Newton-CG Descent)**:**
For solving nonlinear problems with exact gradient information:

1. *Input:* $f \in \mathcal{C}^1$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.
2. *Set* $x_k \leftarrow x_0$, $\eta_k \leftarrow \min(\frac{1}{2}, \sqrt{||\nabla f_k||}) \cdot ||\nabla f(x_k)||$.
3. *While* $||\nabla f(x_k)|| > \varepsilon$ *do*
   a) *Set* $d_k \leftarrow -\nabla f(x_k)$.
   b) *Approximate* $d_H \leftarrow \nabla^2 f(x_k) d_k$ *with Alg. 11.5.*
   c) *Set* $\rho_k \leftarrow {d_k}^\top d_H$.
   d) *If* $\rho_k > \varepsilon ||d_k||^2$ *do the first CG-step:*
      i. *Set* $r_j \leftarrow \nabla f(x_k)$ *and* $p_j \leftarrow -r_k$ *and* $x_j \leftarrow x_k$.
      ii. *Set* $d_A \leftarrow d_H$ *and set* $\rho_j \leftarrow \rho_k$.
      iii. *Set* $t_j \leftarrow \frac{||r_j||^2}{\rho_j}$ *and set* $x_j \leftarrow x_j + t_j d_j$.
      iv. *Set* $r_{old} \leftarrow r_j$ *and set* $r_j \leftarrow r_{old} + t_j d_A$.
      v. *Set* $\beta_j \leftarrow \frac{||r_j||^2}{||r_{old}||^2}$ *and set* $d_j \leftarrow -r_j + \beta_j d_j$.
      vi. *While* $||r_j|| > \eta_k$ *do additional CG steps:*
         A. *Approximate* $d_A \leftarrow \nabla^2 f(x_k) d_j$ *and set* $\rho_j \leftarrow {d_j}^\top d_A$.
         B. *Set* $t_j \leftarrow \frac{||r_j||^2}{\rho_j}$ *and set* $x_j \leftarrow x_j + t_j d_j$.
         C. *Set* $r_{old} \leftarrow r_j$ *and set* $r_j \leftarrow r_{old} + t_j d_A$.
         D. *Set* $\beta_j \leftarrow \frac{||r_j||^2}{||r_{old}||^2}$ *and set* $d_j \leftarrow -r_j + \beta_j d_j$.
      vii. *Set* $d_k \leftarrow x_j - x_k$.
   e) *Calculate a step size* $t_k > 0$ *for* f *at* x_k *in direction* d_k *with Wolfe-Powell line search.*
   f) *Set* $x_k \leftarrow x_k + t_k d_k$.
4. *Output:* $x_* \leftarrow x_k$.

Remark:
$||r_k||$ approximates $||r_n|| = ||\nabla f(x_k) + \sum_{j=0}^{n-1} t_j \nabla^2 f(x_k) d_j|| \approx ||\nabla f(x_k) + \nabla^2 f(x_k) d_k||$. This is because the underlying CG-approach tries to solve $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$. The projected version of this algorithm is found in the appendix (Algorithm 11.7). A step size different from $t_k = 1$ is in theory only needed, if the positive curvature check fails.

Inexact Newton-CG Descent is related to nonlinear conjugate methods. These methods build descent directions of the kind $d_k = \sum_{j=0}^{n-1} t_j d_j$, whereas the d_j directions are updated with $d_j \leftarrow -r_j + \beta_j d_j$. Different alternatives for β_j can be used and lead to the methods of Fletcher-Reeves, Polak-Ribiere and Hestenes-Stiefel (see literature). □

6.2 Quasi-Newton Methods

The idea to generate Newton-like steps with vanishing $||\nabla^2 f(x_k) d_k + \nabla f(x_k)||$ also leads to *Quasi-Newton Methods*. In these methods we establish a sequence of s.p.d. matrices $H_k \in \mathbb{R}^{n \times n}$ in the main loop such that the Dennis-Moré condition holds:

Theorem 6.4 (Dennis-Moré Condition)**:**
Let Assumption 5.4 be true. For $x_0 \in \mathbb{R}^n$ the sequence $x_{k+1} = x_k + d_k$ with $d_k = -H_k^{-1} \nabla f(x_k)$ satisfies the requirements for Theorem 6.2, if the matrices $H_k \in \mathbb{R}^{n \times n}$ are s.p.d. and satisfy the ***Dennis-Moré condition****:*

$$\lim_{k \to \infty} \frac{||(H_k - \nabla^2 f(x_k))(x_{k+1} - x_k)||}{||x_{k+1} - x_k||} = 0 \tag{6.5}$$

Sufficient for the Dennis-Moré condition are

$$\lim_{k \to \infty} \frac{||(H_{k+1} - H_k)(x_{k+1} - x_k)||}{||x_{k+1} - x_k||} = 0 \tag{6.6}$$

in combination with

$$\lim_{k \to \infty} \frac{||(H_{k+1})(x_{k+1} - x_k) - (\nabla f(x_{k+1}) - \nabla f(x_k))||}{||x_{k+1} - x_k||} = 0 \tag{6.7}$$

Descent algorithms that generate H_k matrices according to Theorem 6.4 converge Q-superlinearly (see literature). Obviously equation (6.7) holds, if the **secant equation** or **quasi-Newton condition** is satisfied:

$$H_{k+1} \underbrace{(x_{k+1} - x_k)}_{:= \Delta x_k} = \underbrace{\nabla f(x_{k+1}) - \nabla f(x_k)}_{:= \Delta g_k} \tag{6.8}$$

Also if we make sure that $\lim_{k\to\infty} ||(H_{k+1} - H_k)|| = 0$ holds, we satisfy equation (6.6). We can formulate these requirements as an optimization problem!

Definition 6.5 (Quasi-Newton Optimization Problem)**:**
For given $H_k \in \mathbb{R}^{n\times n}$ and $\Delta x_k, \Delta g_k \in \mathbb{R}^n$ the quasi-Newton optimization problem is

$$\begin{aligned} &\text{minimize} \quad f(H) := ||H - H_k|| \\ \text{s.t.} \quad &H \in \Omega := \{H \in \mathbb{R}^{n\times n} : \quad H\Delta x_k = \Delta g_k\} \end{aligned}$$

The solution of this optimization problem is not unique, so additional constraints are required. Several choices for additional constraints can be established to guarantee for example the s.p.d. property. The following lemma states the BFGS update formula:

Lemma 6.6 (Quasi-Newton Update Formulas)**:**
Let H_k be s.p.d. and $x_{k+1} = x_k + d_k$ and $H_k d_k = -\nabla f(x_k)$. Then for $\Delta x_k := x_{k+1} - x_k$ and $\Delta g_k := \nabla f(x_{k+1}) - \nabla f(x_k)$ the ***update formula of Broyden-Fletcher-Goldfarb-Shanno (BFGS)*** *is:*

$$H_{k+1} := H_k + \frac{\Delta g_k \Delta g_k{}^\top}{\Delta g_k{}^\top \Delta x_k} - \frac{H_k \Delta x_k (H_k \Delta x_k)^\top}{\Delta x_k{}^\top H_k \Delta x_k} \tag{6.9}$$

If in addition $\Delta g_k{}^\top \Delta x_k > 0$ holds, this formula satisfies the secant equation

$$H_{k+1}\Delta x_k = \Delta g_k \tag{6.10}$$

and H_{k+1} is s.p.d.

If B_k is the inverse of H_k and $r_k := \Delta x_k - B_k \Delta g_k$, then the inverse of H_{k+1} can be computed using the ***inverse (BFGS) update formula****:*

$$B_{k+1} := B_k + \frac{r_k \Delta x_k{}^\top + \Delta x_k r_k{}^\top}{\Delta g_k{}^\top \Delta x_k} - \frac{r_k{}^\top \Delta g_k}{(\Delta g_k{}^\top \Delta x_k)^2} \Delta x_k \Delta x_k{}^\top \tag{6.11}$$

Proof. The update formula is sufficient for the secant equation because

$$H_{k+1}\Delta x_k = H_k \Delta x_k + \Delta g_k - H_k \Delta x_k = \Delta g_k \tag{6.12}$$

H_{k+1} is s.p.d. by induction: First of all H_{k+1} is symmetric, because H_k is symmetric and dyadic products are symmetric. Then look at $d \neq 0$:

$$d^\top H_{k+1} d = d^\top H_k d + \frac{d^\top \Delta g_k \Delta g_k{}^\top d}{\Delta g_k{}^\top \Delta x_k} - \frac{d^\top H_k \Delta x_k \Delta x_k{}^\top H_k d}{\Delta x_k{}^\top H_k \Delta x_k} \tag{6.13}$$

$$= d^\top H_k d + \frac{|d^\top \Delta g_k|^2}{\Delta g_k{}^\top \Delta x_k} - \frac{|d^\top H_k \Delta x_k|^2}{\Delta x_k{}^\top H_k \Delta x_k} \overset{!}{>} 0 \tag{6.14}$$

Because of the Cauchy-Schwarz inequality

$$|a^\top b|^2 \begin{cases} = |a^\top a| \cdot |b^\top b| & \text{if } a \text{ and } b \text{ are linearly dependent,} \\ < |a^\top a| \cdot |b^\top b| & \text{else,} \end{cases} \tag{6.15}$$

we get:

$$|d^\top H_k \Delta x_k|^2 = |d^\top L L^\top \Delta x_k|^2 \begin{cases} = |d^\top H_k d| \cdot |\Delta x_k{}^\top H_k \Delta x_k| & \text{if} \quad d = \alpha \Delta x_k, \\ < |d^\top H_k d| \cdot |\Delta x_k{}^\top H_k \Delta x_k| & \text{else.} \end{cases} \tag{6.16}$$

And therefore

$$d^\top H_k d - \frac{|d^\top H_k \Delta x_k|^2}{\Delta x_k{}^\top H_k \Delta x_k} \begin{cases} = 0 & \text{if} \quad d = \alpha \Delta x_k, \\ > 0 & \text{else.} \end{cases} \tag{6.17}$$

If $d = \alpha \Delta x_k \neq 0$, we get

$$d^\top \Delta g_k = \alpha \underbrace{\Delta x_k{}^\top \Delta g_k}_{>0} \neq 0 \tag{6.18}$$

leading to $|d^\top \Delta g_k|^2 > 0$.

The inverse formula can be verified by checking that $B_{k+1} H_{k+1} = H_{k+1} B_{k+1} = \mathbb{E}$ under the condition $B_k H_k = H_k B_k = \mathbb{E}$. □

A similar approach is done by Davidon, Fletcher and Powell: Look up *(DFP) update formula* in the literature.

We now want to use the inverse (BFGS) update formula in the context of a descent algorithm. But we first have to discuss the necessary condition $\Delta g_k{}^\top \Delta x_k > 0$. In fact, if we introduce a line search method and redefine $x_{k+1} = x_k + t_k d_k$ and in consequence $\Delta x = t_k d_k$, we get

$$\Delta g_k{}^\top \Delta x_k = \nabla f(x_{k+1})^\top t_k d_k - \nabla f(x_k)^\top t_k d_k \overset{!}{>} 0 \tag{6.19}$$

This holds for exact line search in a descent direction, because $\nabla f(x_{k+1})^\top d_k = 0$ (exact line search) and $-\nabla f(x_k)^\top d_k > 0$ (descent direction). The sufficient steepness condition from Wolfe-Powell line search, see condition (4.18), is also sufficient to guarantee $\Delta g_k{}^\top \Delta x_k > 0$. Linear convergence is covered by Theorem 4.15. Q-superlinear convergence requires $t_k = 1$, but this will hold close to the (LMP), if at the same time B_k is close enough to the true inverse Hessian.

If Wolfe-Powell line search returns $t_k \neq 1$ we can end up in a situation, where the proof of Lemma 6.6 is not valid and the resulting matrix B_k loses s.p.d. property. We take care of this in the upcoming algorithm by doing a descent direction check and resetting B_k if necessary. For the projection version see Alg. 11.8.

Algorithm 6.7 (Quasi-Newton with inverse BFGS-Update (Broyden, Fletcher, Goldfarb, Shanno))**:**
For solving nonlinear programs using a matrix B_k converging to the inverse Hessian

1. *Input:* $f \in \mathcal{C}^1$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.
2. *Set* $x_k \leftarrow x_0$, $B_k \leftarrow \mathbb{E}$.
3. *While* $||\nabla f(x_k)|| > \varepsilon$ *do*
   a) *Set* $d_k = -B_k \nabla f(x_k)$.
   b) *If* d_k *is not a descent direction, set* $d_k = -\nabla f(x_k)$ *and* $B_k \leftarrow \mathbb{E}$.
   c) *Find* t_k *with Wolfe-Powell line search.*
   d) *Set* $\Delta g_k \leftarrow \nabla f(x_k + t_k d_k) - \nabla f(x_k)$ *and* $\Delta x_k \leftarrow t_k d_k$.
   e) *Set* $x_k \leftarrow x_k + t_k d_k$.
   f) *Update* B_k *according to the inverse (BFGS) update formula from Lemma 6.6.*
4. *Output:* $x_* \leftarrow x_k$.

Home Exercise 6.1 (Quasi-Newton Methods)**:**
Consider the (trivial) quadratic problem

$$\begin{aligned} \textit{minimize} \quad & f(u,v) = \frac{1}{2}(u^2 + v^2) \\ \textit{s.t.} \quad & x = (u,v)^\top \in \mathbb{R}^2 \end{aligned}$$

a) *Verify that the matrix* $H_0 = \begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix}$ *is s.p.d. and compute the inverse matrix* $B_0 = H_0^{-1}$.

b) *At* $x_0 = (1,0)^\top$ *perform a quasi-Newton step with respect to* f *using the iteration scheme* $x_1 = x_0 + t_0 d_0$, *with* t_0 *resulting from exact line search and* d_0 *satisfies* $d_0 = -B_0 \nabla f(x_0)$.

c) *Update* H_0 *to* H_1 *using the (BFGS) update formula.*

6.3 Gauss-Newton Steps for Nonlinear Least Squares

A special Newton-type method satisfying the requirements for Theorem 6.2, the **Gauss-Newton method**, can be tailored for **least squares problems**:

Definition 6.8 (Least Squares Problem)**:**

$$\begin{aligned} \textit{minimize} \quad & f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j(x)^2 = \frac{1}{2}R(x)^\top R(x) \\ \textit{s.t.} \quad & x \in \mathbb{R}^n \end{aligned}$$

with ***error vector*** $R : \mathbb{R}^n \to \mathbb{R}^m$ *and the* j*-th component being* $r_j : \mathbb{R}^n \to \mathbb{R}$.

These objective functions commonly occur in fitting problems or if multiple objectives have to be satisfied. Because minimizing $||y - x||$ is solution equivalent to minimizing $\frac{1}{2}||y - x||^2$, least squares methods also show up in finding projection mappings (see Definition 2.9).

Example 6.9:
Consider the problem

$$\textit{minimize} \quad f^1(x), f^2(x), \ldots, f^m(x). \quad \textit{s.t.} \quad x \in \mathbb{R}^n$$

Each f^j *can be minimized separately by* x_*^j*, but typically* $x_*^j \neq x_*^{\tilde{j}}$ *for* $j \neq \tilde{j}$*. So a reasonable compromise could be*

$$\begin{aligned} \textit{minimize} \quad & f(x) = \frac{1}{2}\sum_{j=1}^{m} \left(f^j(x) - f_*^j\right)^2 =: \frac{1}{2}\sum_{j=1}^{m} r_j(x)^2 \\ \textit{s.t.} \quad & x \in \mathbb{R}^n \end{aligned}$$

If m is sufficiently smaller than n, the problem is underdetermined, $f(x_*) = 0$ can be attained and the problem is reduced to solving $R(x_*) = 0$. But typically the number of error components m is much larger than the number of unknown coefficients n (overdetermined problem).

In order to derive methods that solve least squares problems, we need to check the optimality conditions:
The i-th component of the gradient is

$$\frac{\partial}{\partial x_i} f(x) = \sum_{j=1}^{m} r_j(x) \frac{\partial}{\partial x_i} r_j(x) \tag{6.20}$$

We define the **Jacobian matrix** $J : \mathbb{R}^n \to \mathbb{R}^{m\times n}$ with the matrix entries

$$(J(x))_{j,i} = \frac{\partial r_j(x)}{\partial x_i} \quad \text{for} \quad 1 \le j \le m, \quad 1 \le i \le n. \tag{6.21}$$

and realize

$$\frac{\partial}{\partial x_i} f(x) = \sum_{j=1}^{m} (J(x))_{j,i} r_j(x) = (J(x)^\top R(x))_i \tag{6.22}$$

The **first order necessary optimality condition for the unconstrained case** is therefore

$$\nabla f(x_*) = J(x_*)^\top R(x_*) = 0 \tag{6.23}$$

Next we consider the i,k-th component of the Hessian $\nabla^2 f(x)$:

$$\frac{\partial^2}{\partial x_i \partial x_k} f(x) = \sum_{j=1}^{m} \frac{\partial}{\partial x_i} \left(r_j(x) \frac{\partial r_j(x)}{\partial x_k} \right) \tag{6.24}$$

$$= \sum_{j=1}^{m} \frac{\partial r_j(x)}{\partial x_i} \frac{\partial r_j(x)}{\partial x_k} + \sum_{j=1}^{m} r_j(x) \frac{\partial^2 r_j(x)}{\partial x_i \partial x_k} \tag{6.25}$$

So

$$\nabla^2 f(x) = J(x)^\top J(x) + \sum_{j=1}^{m} r_j(x) \nabla^2 r_j(x) \tag{6.26}$$

In the upcoming Gauss-Newton step we discard the second term and approximate $\nabla^2 f(x)$ as $J(x)^\top J(x)$. For small $r_j(x)$, this leads to an inexact Newton method:

$$x_{k+1} = x_k - \left(J(x_k)^\top J(x_k)\right)^{-1} \nabla f(x_k) \tag{6.27}$$

$$= x_k - \left(J(x_k)^\top J(x_k)\right)^{-1} J(x_k)^\top R(x_k) \tag{6.28}$$

In order to be able to invert $J(x)^\top J(x)$, we need to have at least as much error components as optimization variables, otherwise
$\det(J(x)^\top J(x))$ is zero. The **Gauss-Newton step** can then be seen as solving

$$J(x_k)^\top J(x_k) \underbrace{(x_{k+1} - x_k)}_{:=d_k} = -J(x_k)^\top R(x_k) \tag{6.29}$$

or

$$J(x_k)^\top \left(R(x_k) + J(x_k)(x_{k+1} - x_k)\right) = 0 \tag{6.30}$$

This means that x_{k+1} is the (LMP) of the following problem

$$\text{minimize} \quad \psi(x) := \frac{1}{2}||R(x_k) + J(x_k)(x - x_k)||^2 \quad \text{s.t.} \quad x \in \mathbb{R}^n, \tag{6.31}$$

because the first order optimality condition is:

$$\nabla\psi(x) = J(x_k)^\top \big(R(x_k) + J(x_k)(x - x_k)\big) \overset{!}{=} 0 \tag{6.32}$$

In consequence, applying this modified Newton step is equivalent to solving a linear least squares problem in each step.

Assumption 6.10 (Wellposed Least Squares Problem)**:**
We assume $x_* \in \mathbb{R}^n$ is a (LMP) of $\frac{1}{2}R(x)^\top R(x)$, the vector $R(x) \in \mathbb{R}^m$ is Lipschitz-continuously differentiable near x_* with Jacobian $J(x) \in \mathbb{R}^{m\times n}$ and $J(x_*)^\top J(x_*) \in \mathbb{R}^{n\times n}$ has full rank n ($\to m \geq n$).

For short we write $J_k = J(x_k)$ and $J_* = J(x_*)$ as well as $R_k = R(x_k)$ and $R_* = R(x_*)$.

Theorem 6.11 (Gauss-Newton Step)**:**
Let Assumption 6.10 be true. Then for sufficiently small $\delta > 0$ there is $K > 0$ such that for $x_k \in B_\delta(x_*)$ and

$$x_{k+1} = x_k - \left({J_k}^\top J_k\right)^{-1} {J_k}^\top R_k, \tag{6.33}$$

we get the following convergence estimate:

$$||x_{k+1} - x_*|| \leq K\big(||x_k - x_*|| + ||R_*||\big)||x_k - x_*||. \tag{6.34}$$

Proof. Let $\delta > 0$ be small enough such that for all $x \in B_\delta(x_*)$ holds: $J(x)$ is Lipschitz with Lipschitz-constant L and $J(x)^\top J(x)$ is regular. We define $e_k := x_k - x_*$ and realize

$$\begin{aligned} e_{k+1} &= e_k - \left({J_k}^\top J_k\right)^{-1} {J_k}^\top R_k \\ &= \left({J_k}^\top J_k\right)^{-1} {J_k}^\top \left(J_k e_k - R_k\right) \end{aligned}$$

Using Taylor we know

$$R_* = R(x_k - e_k) = R_k - \int_0^1 J(x_k - te_k)e_k \,\mathrm{d}t,$$

which helps us to show

$$\begin{aligned} J_k e_k - R_k &= -R_* + R_* - R_k + J_k e_k \\ &= -R_* - \int_0^1 J(x_k - t e_k) e_k \, \mathrm{d}t + J_k e_k \\ &= -R_* + \int_0^1 \big(J_k - J(x_k - t e_k)\big) e_k \, \mathrm{d}t \end{aligned}$$

Now multiplying this with ${J_k}^\top$ and using ${J_*}^\top R_* = 0$, we conclude

$$\begin{aligned} &{J_k}^\top (J_k e_k - R_k) \\ &= (J_* - J_k)^\top R_* + {J_k}^\top \left(\int_0^1 \big(J_k - J(x_k - t e_k)\big) e_k \, \mathrm{d}t \right) \end{aligned}$$

or in terms of norms

$$\begin{aligned} &||{J_k}^\top (J_k e_k - R_k)|| \\ &\leq ||(J_* - J_k)^\top R_*|| + ||{J_k}^\top|| \int_0^1 ||J_k - J(x_k - t e_k)|| \, \mathrm{d}t ||e_k|| \\ &\leq L||e_k|| \cdot ||R_*|| + ||J_k|| \int_0^1 L||x_k - x_k - t e_k|| \, \mathrm{d}t ||e_k|| \\ &\leq L||e_k|| \cdot ||R_*|| + \frac{L}{2} ||J_k|| \cdot ||e_k||^2. \end{aligned}$$

Coming back to the estimation of e_{k+1} gives:

$$\begin{aligned} ||e_{k+1}|| &\leq ||\left({J_k}^\top J_k\right)^{-1}|| \cdot \left(L||e_k|| \cdot ||R_*|| + \frac{L}{2} ||J_k|| \cdot ||e_k||^2 \right) \\ &\leq K(||R_*|| + ||e_k||)||e_k|| \end{aligned}$$

whereas K is chosen as follows:

$$K = L \cdot \max_{x_k \in B_\delta(x_*)} \left[||\left({J_k}^\top J_k\right)^{-1}||, \tfrac{1}{2} ||\left({J_k}^\top J_k\right)^{-1}|| \cdot ||J_k|| \right]$$

□

Remark:
Theorem 6.11 states that Gauss-Newton steps converge fast for small $||R_*||$. But if $||R_*||$ is large, we have to expect linear convergence or worse for big L in $B_\delta(x_*)$. □

Exercise 6.12:
Perform one Gauss-Newton step for

$$f(u,v) = \frac{1}{2} R(u,v)^\top R(u,v) = \frac{1}{2}(u-3)^2 + \frac{1}{2}(uv-3)^2 + \frac{1}{2}(uv^2-3)^2$$

with $(u,v)^\top \in \mathbb{R}^2$ *and starting point* $x_0 = (1,1)^\top$. *We recognize*

$$R(u,v) = \begin{pmatrix} u-3 \\ uv-3 \\ uv^2-3 \end{pmatrix}, \qquad J(u,v) = \begin{pmatrix} 1 & 0 \\ v & u \\ v^2 & 2uv \end{pmatrix},$$

$$J(u,v)^\top R(u,v) = \begin{pmatrix} u-3+uv^2-3v+uv^4-3v^2 \\ u^2v-3u+2u^2v^3-6uv \end{pmatrix},$$

$$J(u,v)^\top J(u,v) = \begin{pmatrix} 1+v^2+v^4 & uv+2uv^3 \\ uv+2uv^3 & u^2+4u^2v^2 \end{pmatrix}$$

To perform the Gauss-Newton step we do not invert ${J_k}^\top J_k$ *directly but instead solve* ${J_0}^\top J_0 d_0 = -{J_0}^\top R_0$ *for* $x_0 = (1,1)^\top$*:*

$$\begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix} d_0 = -\begin{pmatrix} -6 \\ -6 \end{pmatrix}$$

leading to $d_0 = (2,0)^\top$ *and* $x_1 = x_0 + d_0 = (3,1)^\top$ *which is the (GMP) because* $R(3,1) = 0$.

Home Exercise 6.2 (Modeling Nonlinear Least Squares)**:**
Consider the curve

$$\gamma(\phi; a,b,c) = \begin{pmatrix} a\cos(\phi) \\ \sin(b\phi) \\ c\phi \end{pmatrix}$$

depending on $\phi \in [0, 2\pi)$ *with unknown parameters* $a,b,c > 0$. *It is known that the curve passes the points*

$$\gamma_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad at \quad \phi_0 = 0, \gamma_1 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \quad at \quad \phi_1 = 1, \gamma_2 = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} \quad at \quad \phi_2 = 2.$$

a) *Formulate an error vector* $R : \mathbb{R}^3 \to \mathbb{R}^7$ *as a mapping of* (a,b,c) *such that*

$$\frac{1}{2} R^\top R = f(a,b,c) = \frac{1}{2} \sum_{i=0}^{2} ||\gamma(\phi_i; a,b,c) - \gamma_i||^2.$$

b) *Compute the Jacobian* $J(a,b,c)$ *of* $R(a,b,c)$. *Compute* $\nabla f(a,b,c) = J^\top R$ *and* $A(a,b,c) := J^\top J$.

c) *Show that at* $(a_*, b_*, c_*) = (0, \frac{\pi}{2}, 1)$ *we have* $\nabla f(a_*, b_*, c_*) = 0$ *and* $A(a_*, b_*, c_*)$ *is s.p.d.*

6.4 Levenberg-Marquardt Method

If Assumption 6.10 fails and the rank of J_k is smaller than n, the matrix ${J_k}^\top J_k$ can have a zero eigenvalue (and is not s.p.d. and not invertible). In any case, ${J_k}^\top J_k$ is still s.p.s., because $x^\top J^\top Jx = (Jx)^\top Jx \geq 0$. We can approximate a s.p.s. by a s.p.d. matrix by shifting zero eigenvalues into the positive range. This is done by adding a diagonal matrix $\alpha\mathbb{E}$:

$$x^\top(J^\top J + \alpha\mathbb{E})x = (Jx)^\top Jx + \alpha x^\top x > 0 \quad \text{for all} \quad x \neq 0 \tag{6.35}$$

The modified Gauss-Newton step is

$$x_{k+1} = x_k - \left({J_k}^\top J_k + \alpha_k\mathbb{E}\right)^{-1} {J_k}^\top R_k \tag{6.36}$$

with $\alpha_k > 0$ sufficiently large. This approach is called **Levenberg-Marquardt step**.

The descent direction implied by the Levenberg-Marquardt step is modified by the choice of α_k. If α_k is very large compared to ${J_k}^\top J_k$, the Levenberg-Marquardt step is a steepest descent step with a small step size $\frac{1}{\alpha_k}$ and slow convergence rate:

$$x_{k+1} \approx x_k - \frac{1}{\alpha_k}{J_k}^\top R_k = x_k - \frac{1}{\alpha_k}\nabla f(x_k) \tag{6.37}$$

Theorem 6.13 (Levenberg-Marquardt Step)**:**
Let $\alpha > 0$, then ${J_k}^\top J_k + \alpha\mathbb{E}$ is s.p.d. with smallest eigenvalue being $\lambda_{min} + \alpha$ and $d_k(\alpha)$ is the unique solution of

$$\left({J_k}^\top J_k + \alpha\mathbb{E}\right) d_k(\alpha) = -{J_k}^\top R_k. \tag{6.38}$$

Then

$$||d_k(\alpha)|| \leq ||{J_k}^\top R_k|| \frac{1}{(\lambda_{min} + \alpha)} \tag{6.39}$$

and if in addition $\alpha > 0$ is sufficiently large we get $f(x_k + d_k(\alpha)) < f(x_k)$.

Proof. Because ${J_k}^\top J_k$ is symmetric, there exists an orthonormal matrix V consisting of eigenvectors such that

$${J_k}^\top J_k = VDV^\top \tag{6.40}$$

with D being the diagonal matrix of the eigenvalues $\lambda_i \geq 0$.

The same transformation V can be used for the shifted matrix

$${J_k}^\top J_k + \alpha\mathbb{E} = VDV^\top + \alpha VV^\top = V(D + \alpha\mathbb{E})V^\top, \tag{6.41}$$

leading to $\lambda_i + \alpha > 0$.

This means for each $\alpha > 0$ holds

$$d_k(\alpha) = -\left(J_k{}^\top J_k + \alpha\mathbb{E}\right)^{-1} J_k{}^\top R_k =$$
$$-\left(V(D+\alpha\mathbb{E})V^\top\right)^{-1} J_k{}^\top R_k = -V\left(D+\alpha\mathbb{E}\right)^{-1} V^\top J_k{}^\top R_k$$

It is easy to show now that

$$\begin{aligned} ||d_k(\alpha)||^2 &= d_k(\alpha)^\top d_k(\alpha) \\ &= \left(-V\left(D+\alpha\mathbb{E}\right)^{-1} V^\top J_k{}^\top R_k\right)^\top \left(-V\left(D+\alpha\mathbb{E}\right)^{-1} V^\top J_k{}^\top R_k\right) \\ &= R_k{}^\top J_k V\left(D+\alpha\mathbb{E}\right)^{-2} V^\top J_k{}^\top R_k \leq ||\nabla f_k||^2 \frac{1}{(\lambda_{min}+\alpha)^2} \end{aligned}$$

Now if α is chosen large enough, $||d_k(\alpha)||$ is getting small enough to reach a point in the environment where $f(x_k + d_k(\alpha)) < f(x_k)$ holds, because $d_k(\alpha)$ is a descent direction:

$$\nabla f_k{}^\top d_k(\alpha) = -\left(J_k{}^\top R_k\right)^\top \underbrace{\left(J_k{}^\top J_k + \alpha\mathbb{E}\right)^{-1}}_{\text{s.p.d.}} J_k{}^\top R_k < 0 \qquad \checkmark \tag{6.42}$$

□

The upcoming Levenberg-Marquardt algorithm is designed to find optimal parameters $p \in \mathbb{R}^n$ for a function $f(x,p)$, for which measure results f_j at m measure points x_j are available. The j-th component of the error vector is then defined as $R_j(p) = f(x_j, p) - f_j$.

Algorithm 6.14 (Levenberg-Marquardt Descent)**:**
For minimizing $f(p) = \frac{1}{2}R(p)^\top R(p)$

1. *Input:* $R \in \mathcal{C}^1$; $p_0 \in \mathbb{R}^n$; $\varepsilon > 0$; $\alpha_0 > 0$; $\beta > 1$.
2. *Set* $p_k \leftarrow p_0$ *and* $\alpha_k \leftarrow \alpha_0$.
3. *While* $||J(p_k)^\top R(p_k)|| > \varepsilon$ *do*
   a) *Solve* $\left(J(p_k)^\top J(p_k) + \alpha_k\mathbb{E}\right) d_k = -J(p_k)^\top R(p_k)$ *for* d_k *with conjugate gradient solver.*
   b) *If* $\frac{1}{2}R(p_k+d_k)^\top R(p_k+d_k) < \frac{1}{2}R(p_k)^\top R(p_k)$ *do*
      i. *Accept* $p_{k+1} \leftarrow p_k + d_k$ *and reset* $\alpha_k \leftarrow \alpha_0$.
   c) *Else increase* $\alpha_k \leftarrow \beta\alpha_k$.
4. *Output:* $p_* \leftarrow p_k$.

Home Exercise 6.3 (Levenberg-Marquardt Algorithm)**:**
Consider the process function $p(t; u, v) = u \cdot v^t$ *with unknown parameters* $u, v \in \mathbb{R} \times \mathbb{R}^+$ *and the following measure data:* $t_0 = 0$, $t_1 = 1$, $t_2 = 2$ *and* $p_0 = p_1 = p_2 = 3$.

a) *Formulate an error vector* $R : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^3$ *as a mapping of* (u, v) *such that*

$$\frac{1}{2} R^\top R = f(u, v) = \frac{1}{2} \sum_{i=0}^{2} ||p(t_i; u, v) - p_i||^2$$

and compute the Jacobian $J(u, v)$ *of* $R(u, v)$.

b) *Perform one Levenberg-Marquardt step at* $x_0 = (0, 1)^\top$ *using a general* $\alpha_0 > 0$ *to get* x_1 *in dependence of* α_0.

c) *Use only* $R(u, v)$ *to justify that* $\lim_{\alpha_0 \to 0} x_1$ *is a (GMP) of* f *on* $\mathbb{R} \times \mathbb{R}^+$.

7 Algorithms for Finding (KKT) Points

In the previous sections we discussed descent algorithms for unconstrained problems (Algorithm 4.13) and for problems with box constraints (Algorithm 4.18). We introduced Newton-type methods for unconstrained problems only, but these can be translated into the context of box constraints using projections.

The situation is different for problems with equality and inequality constraints: All (LMPs) x_* of such a problem (see Definition 2.23) satisfy the (KKT) conditions and especially $\nabla_x L(x_*, \mu_*, \lambda_*) = 0$. But the triple (x_*, μ_*, λ_*) is in general a saddle point of the Lagrangian function and therefore cannot be found with descent algorithms. This issue is inherent in the structure of the Lagrangian function defined in equation (2.37).

7.1 Barrier Methods and Penalty Methods

Barrier methods and penalty methods are a type of solution strategy, in which the objective $f(x)$ is augmented with a suitable function of the constraints $g_r(x)$ or $h_j(x)$, such that minimizing the combined function leads approximately to (KKT) points.

For minimizing an objective $f(x)$ under inequality constraints $g_r(x) \leq 0$ the following **barrier problem** can be formulated:

$$\begin{aligned} \text{minimize} \quad & B(x) := f(x) - \beta \sum_{r=1}^{s} \ln(-g_r(x)) \\ \text{s.t.} \quad & x \in \Omega_B = \{x \in \mathbb{R}^n : g_r(x) < 0\} \end{aligned}$$

with $\beta > 0$ being a barrier parameter. A (LMP) $x_*(\beta)$ of $B(x)$ with $g_r(x_*(\beta)) < 0$ satisfies

$$\nabla B(x) = \nabla f(x) - \sum_{r=1}^{s} \frac{\beta}{g_r(x)} \nabla g_r(x) = 0 \tag{7.1}$$

and the Hessian is

$$\nabla^2 B(x) = \nabla^2 f(x) + \sum_{r=1}^{s} \frac{\beta}{g_r(x)} \left(\frac{1}{g_r(x)} \nabla g_r(x) \nabla g_r(x)^\top - \nabla^2 g_r(x) \right) \tag{7.2}$$

If β is chosen close enough to zero but still positive and $\nabla g_r(x) \nabla g_r(x)^\top$ is s.p.d., $B(x)$ gets locally convex close to the boundary, enforcing a (LMP) $x_*(\beta)$ in the open set Ω_B. Now assume that for $\beta \to 0$ we get some $x_*(\beta)$ with $g_r(x_*(\beta)) < 0$. Then $B(x_*) \to f(x_*)$ and $\frac{-\beta}{g_r(x_*)}$ approximates the Lagrangian multiplier $\mu_{*,r}$: This means $x_*(\beta)$ is close to the true (LMP) of the original function $f(x)$.

For minimizing an objective $f(x)$ under equality constraints $h_j(x) = 0$ a classical **penalty approach** is solving the following problem:

$$\text{minimize} \quad C(x) := f(x) + \frac{1}{2}\gamma \sum_{j=1}^{m} (h_j(x))^2 \tag{7.3}$$

$$\text{s.t.} \quad x \in \mathbb{R}^n \tag{7.4}$$

with $\gamma > 0$ being a penalty parameter. A (LMP) $x_*(\gamma)$ of $C(x)$ on $\mathbb{R}^n$ satisfies

$$\nabla C(x) = \nabla f(x) + \sum_{j=1}^{m} \gamma h_j(x) \nabla h_j(x) = 0 \tag{7.5}$$

and the Hessian is

$$\nabla^2 C(x) = \nabla^2 f(x) + \sum_{j=1}^{m} \gamma \left(\nabla h_j(x) \nabla h_j(x)^\top + h_j(x) \nabla^2 h_j(x) \right) \tag{7.6}$$

If γ is chosen large enough and $\nabla h_j(x) \nabla h_j(x)^\top$ is s.p.d, then $C(x)$ gets locally convex close to the set $\Omega_C = \{x \in \mathbb{R}^n : h_j(x) = 0\}$, enforcing a (LMP) with $h_j(x_*) \to 0$. The Lagrangian multiplier $\lambda_{*,j}$ is then approximated by the term $\gamma h_j(x_*)$.

In barrier algorithms it is necessary to compute the (LMPs) of $B(x)$ for a decreasing sequence of barrier parameters to find a (LMP) of the original problem. Likewise, penalty algorithms require to compute the (LMPs) of $C(x)$ for a sequence of strongly increasing penalty parameters. Other choices of augmentations can be found in the literature.

Exercise 7.1:
Look at the problem

$$\textit{minimize} \quad f(x) = 2x + 2$$
$$\textit{s.t.} \quad x^2 - 1 \leq 0.$$

The (LMP) is $x_* = -1$ with $\mu_* = 1$.

For the constraint $x^2 - 1 \leq 0$ we can use a barrier approach and verify:

$$B(x) = 2x + 2 - \beta \ln(1 - x^2) \quad \textit{is minimized by} \quad x_*(\beta) = \frac{\beta - \sqrt{\beta^2 + 4}}{2}$$

$$\textit{and} \quad \mu_*(\beta) = \frac{\beta}{g_r(x_*(\beta))} = \frac{\beta}{1 - (x_*(\beta))^2} = \frac{2}{-\beta + \sqrt{\beta^2 + 4}}$$

both $(x_*(\beta), \mu_*(\beta))$ converge to (x_*, μ_*) for $\beta \to 0$.

If instead the equality constraint $x^2 - 1 = 0$ is used, a similar result can be found for the penalty approach.

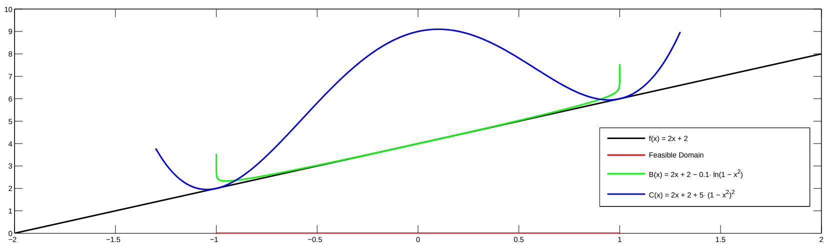

Figure 6: The function $f(x) = 2x + 2$ (black) is augmented with a barrier approach (green) or a penalty approach (blue), compare Exercise 7.1.

Home Exercise 7.1 (Barrier Approach)**:**
Consider the problem

$$\begin{aligned} & \textit{minimize} \quad f(u,v) = 2 - v \\ \textit{s.t.} \quad & x = (u,v)^\top \in \Omega := \{(u,v)^\top \in \mathbb{R}^2 :\ 1 - v \geq u, 1 + u \geq v\} \end{aligned}$$

a) *Formulate a barrier function* $B(u, v; \beta_1, \beta_2)$ *for this problem.*

b) *Find* $(u_\beta, v_\beta)^\top$ *solving* $\nabla B(u_\beta, v_\beta) = 0$, $1 - v_\beta > u_\beta$ *and* $1 + u_\beta > v_\beta$ *in dependence of* $\beta_1, \beta_2 > 0$.

c) *Compute the (GMP)* $(u_*, v_*)^\top := \lim_{\beta_1,\beta_2 \to 0}(u_\beta, v_\beta)^\top$ *and check which inequality constraints from* Ω *are active at* $(u_*, v_*)^\top$*. Estimate the multipliers* μ_1 *and* μ_2 *using the limit* $\lim_{\beta_r \to 0} \frac{-\beta_r}{g_r(u_\beta, v_\beta)}$ *for* $r = 1, 2$.

7.2 Augmented Lagrangian Method

Barrier and penalty type methods are a strategy to handle the inequality or equality constraints by shifting them into the objective and generate local convexity. In the upcoming augmented Lagrangian method the idea is now, to shift the equality constraints into the Lagrangian and generate a sequence of (x_k, λ_k) converging to a (KKT) point.

We use the slack mechanics from Lemma 3.3 to reformulate inequality constraints as a combination of equality constraints and box constraints, allowing us to solve the augmented Lagrangian subproblem with projected descent methods:

Corollary 7.2 (Solution Equivalent Reformulation)**:**
The general problem with equality and inequality constraints (see Definition 2.23) can

be reformulated as

$$\begin{aligned} \textit{minimize} \quad & f(x) \\ \textit{s.t.} \quad [x,y] \in \Omega := & \left\{ \begin{matrix} x \in \Omega_\Box, y \in \mathbb{R}^s \\ y_r \geq 0 \quad \textit{for} \quad r = 1, \ldots, s \\ h_j(x) = 0 \quad \textit{for} \quad j = 1, \ldots, m \\ g_r(x) + y_r = 0 \quad \textit{for} \quad r = 1, \ldots, s \end{matrix} \right\} \end{aligned}$$

The $r = 1, \ldots, s$ variables y_r are the slack variables.

This means, we can concentrate on problems with equality constraints and box constraints, for which we define:

Definition 7.3 (Augmented Lagrangian Subproblem)**:**
For

$$\begin{aligned} \textit{minimize} \quad & f(x) \\ \textit{s.t.} \quad x \in \Omega := \{x \in \Omega_\Box \quad & \textit{with} \quad h_j(x) = 0 \quad \textit{for} \quad j = 1, \ldots, m\} \end{aligned}$$

the corresponding ***augmented Lagrangian subproblem*** *is:*

$$\begin{aligned} \textit{minimize} \quad & A(x) := f(x) + \sum_{j=1}^{m} \alpha_j h_j(x) + \frac{1}{2}\gamma \sum_{j=1}^{m} h_j^2(x) \\ \textit{s.t.} \quad & x \in \Omega_\Box \end{aligned}$$

with $\alpha \in \mathbb{R}^m$ being a guess for the Lagrangian multipliers and $\gamma > 0$ being the penalty parameter.

A (LMP) x_* of this problem found by a projected descent method satisfies the stationarity property:

$$\nabla A(x_*)^\top (y - x_*) =$$
$$\left(\nabla f(x_*) + \sum_{j=1}^{m} (\alpha_j + \gamma h_j(x_*)) \nabla h_j(x_*) \right)^\top (y - x_*) \geq 0 \quad \text{for all} \quad y \in \Omega_\Box$$

If we get the situation, that for some large enough γ the (LMP) x_* satisfies the equality constraints $h_j(x_*) = 0$, then (x_*, α_*) is the (KKT) point of the augmented Lagrangian subproblem from Definition 7.3. But this requires that $\alpha_* \in \mathbb{R}^m$ is guessed correctly as Lagrangian multiplier vector.

This can in fact be achieved by an algorithm in which γ_k is iteratively increased to get stationary points x_k for a sequence of $\alpha_k \in \mathbb{R}^m$, which are updated with $\alpha_k \leftarrow \alpha_k + \gamma_k h(x_k)$. This is done by the following algorithm:

Algorithm 7.4 (Augmented Lagrangian Descent)**:**
For solving nonlinear problems with equality constraints and box constraints, $A_k(x)$ is the augmented Lagrangian with parameters α_k, γ_k.

1. *Input: $f, h \in \mathcal{C}^1$; $P : \mathbb{R}^n \to \Omega_\Box$; $x_0 \in \mathbb{R}^n$; $\alpha_0, \varepsilon, \delta > 0$.*
2. *Set $x_k \leftarrow P(x_0)$, $\alpha_k \leftarrow \alpha_0$, $\gamma_k \leftarrow 10$, $\varepsilon_k \leftarrow 1/\gamma_k$ and $\delta_k \leftarrow 1/\gamma_k^{0.1}$.*
3. *Build the augmented Lagrangian objective $A_k(x)$ out of f, h depending on current α_k, γ_k.*
4. *While $||x_k - P(x_k - \nabla A_k(x_k))|| > \varepsilon$ or $||h(x_k)|| > \delta$ do*
   a) *Use a projection method to minimize $A_k(x)$ subject to $x \in \Omega_\Box$ to tolerance ε_k. Set x_k to the minimizer.*
   b) *If $||h(x_k)|| \leq \delta_k$, update multiplier $\alpha_k \leftarrow \alpha_k + \gamma_k h(x_k)$, tighten tolerances $\varepsilon_k \leftarrow \max(\varepsilon_k/\gamma_k, \varepsilon)$, $\delta_k \leftarrow \max(\delta_k/\gamma_k^{0.9}, \delta)$.*
   c) *Else increase the penalty parameter $\gamma_k \leftarrow \max(10, \sqrt{\gamma_k})\gamma_k$, reconfigure tolerances $\varepsilon_k \leftarrow 1/\gamma_k$, $\delta_k \leftarrow 1/\gamma_k^{0.1}$.*
   d) *Update the augmented Lagrangian objective $A_k(x)$ depending on current α_k, γ_k.*
5. *Output: $x_* \leftarrow x_k$ and $\lambda_* \leftarrow \alpha_k$.*

Home Exercise 7.2 (Augmented Lagrangian Algorithm)**:**
For $x = (u, v)^\top$ consider the optimization problem

$$\begin{aligned} \text{minimize} \quad & f(x) = u^2 + v^2 \\ \text{s.t.} \quad & x \in \Omega := \{x \in \mathbb{R}^2 : \ h(x) = u + v + 1 = 0\} \end{aligned}$$

a) *Write down the augmented Lagrangian function $A(x; \alpha, \gamma)$ for this problem and the gradient $\nabla_x A(x; \alpha, \gamma)$.*

b) *For $\alpha_0 = 0$ and $\gamma = 2$ find $x_1 \in \mathbb{R}^2$ solving $\nabla_x A(x; \alpha_0, \gamma) = 0$. Compute $\alpha_1 = \alpha_0 + \gamma h(x_1)$.*

c) *Find $x_2 \in \mathbb{R}^2$ solving $\nabla_x A(x; \alpha_1, \gamma) = 0$ and compute α_2.*

8 Derivative-Free Methods

In practical application we often do not have an analytical representation of the objective function we have to minimize and can only measure $f(x_k)$, whereas x_k is a set of measure points. An objective like this is called a **black box**. Sometimes the evaluation of the objective is expensive and a high number of function evaluations to reconstruct the objective using a least squares approach is not an option. Think about the following example:

Example 8.1:
A chemical reaction has to be optimized using catalysis. The objective is a black box and maps a set of catalysts and the starting temperature collected in $x \in \mathbb{R}^n$ onto the reaction time $T(x) > 0$, which can take up to a week. Obviously the time to evaluate this objective is $T(x)$ itself.

Assumptions like continuous differentiability or Lipschitz-continuity of the gradient cannot be made, but we can still apply some math in this context with cheap local gradient approximations.

Definition 8.2 (Noisy Functions)**:**
Consider an objective function f, bounded from below, that represents a smooth function f_s but is subject to noise denoted by ψ with $||\psi||$ small:

$$f(x) = f_s(x) + \psi(x) \tag{8.1}$$

*In addition we demand: If x_* is a (LMP) of f, then $\psi(x_*) = 0$. We call such a function a **noisy function**.*

8.1 The Simplex Gradient

We would like to discuss gradient and Hessian approximation in the context of noisy functions. We want to decompose our function domain into a set of disjoint domain parts, called simplexes.

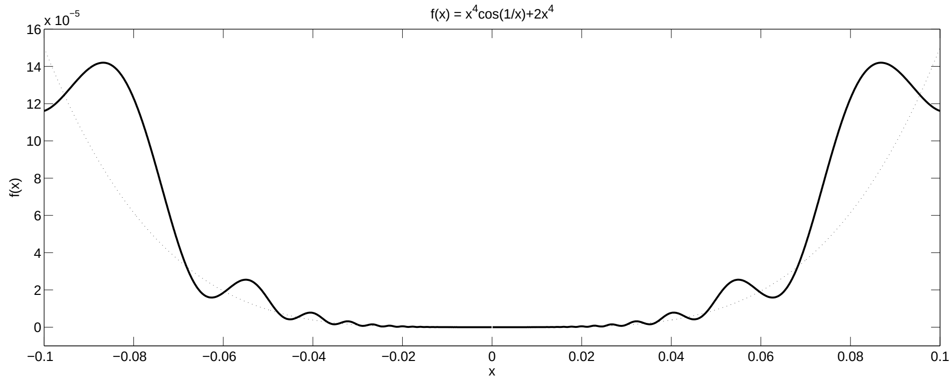

Figure 7: Example for a noisy function

Definition 8.3 (Convex Hull)**:**
The convex hull of a point set $\{x_j\}_{j=0}^m$ *is the convex combination of its points:*

$$\left\{ x \in \mathbb{R}^n |\; x = \sum_{j=0}^{m} \alpha_j x_j \quad \textit{with} \quad \alpha_j \geq 0 \quad \textit{and} \quad \sum_{j=0}^{m} \alpha_j = 1 \right\}.$$

Definition 8.4 (Regular Simplex and Reflected Simplex)**:**
A ***simplex*** $\mathcal{S} \subset \mathbb{R}^n$ *is the convex hull of* $n+1$ *points* $\{x_j\}_{j=0}^n$ *Where* x_j *is the j-th vertex of* $\mathcal{S}$*. For a simplex* $\mathcal{S}$ *we define the matrix of its direction vectors*

$$\mathcal{V}(\mathcal{S}) = \mathcal{V} = (x_1 - x_0 | x_2 - x_0 | \dots | x_n - x_0) = (\mathcal{V}_1 | \dots | \mathcal{V}_n) \in \mathbb{R}^{n \times n}$$

We call a vertex $\mathcal{S}$ *regular if its matrix* $\mathcal{V}$ *is regular and its* ***condition number*** $\kappa(\mathcal{V})$*.*

Furthermore we have the ***diameter***

$$\operatorname{diam}(\mathcal{S}) := \max_{0 \leq i,j \leq n} ||x_i - x_j||. \tag{8.2}$$

and introduce the ***oriented lengths*** $\sigma_+(\mathcal{S})$ *and* $\sigma_-(\mathcal{S})$ *as follows*

$$\begin{aligned} \sigma_+(\mathcal{S}) &= \max_{1 \leq j \leq n} ||x_0 - x_j|| \quad \textit{(longest edge from } x_0\textit{)} \\ \sigma_-(\mathcal{S}) &= \min_{1 \leq j \leq n} ||x_0 - x_j|| \quad \textit{(shortest edge from } x_0\textit{)} \end{aligned}$$

and conclude with triangle inequality

$$\sigma_+(\mathcal{S}) \leq \operatorname{diam}(\mathcal{S}) \leq 2\sigma_+(\mathcal{S}) \tag{8.3}$$

For each regular simplex $\mathcal{S} \subset \mathbb{R}^n$ *exists the regular* ***reflected simplex*** $\mathcal{R} \subset \mathbb{R}^n$ *consisting of the points* x_0 *and* $x_j^R = x_0 - (x_j - x_0)$ *for all* $j = 1, \dots, n$*. The matrix of its direction vectors is*

$$\mathcal{V}(\mathcal{R}) = -\mathcal{V} = (-(x_1 - x_0) | -(x_2 - x_0) | \dots | -(x_n - x_0)) \in \mathbb{R}^{n \times n}$$

We now define the forward and the centered simplex gradient.

Definition 8.5 (Simplex Gradients)**:**
*Let $\mathcal{S}$ be a regular simplex with vertexes $\{x_j\}_{j=0}^n$. The **forward simplex gradient** of f on $\mathcal{S}$ is defined as*

$$D(f : \mathcal{S}) = \mathcal{V}^{-\top}\delta(f : \mathcal{S}) \tag{8.4}$$

with

$$\delta(f : \mathcal{S}) := (f(x_1) - f(x_0), f(x_2) - f(x_0), \ldots, f(x_n) - f(x_0))^\top \tag{8.5}$$

is the function difference of f on $\mathcal{S}$.

*The **centered simplex gradient** of f on $\mathcal{S}$ and its reflected simplex $\mathcal{R}$ and is defined as*

$$D_C(f : \mathcal{S}) = \frac{D(f : \mathcal{S}) + D(f : \mathcal{R})}{2} = \frac{\mathcal{V}^{-\top}(\delta(f : \mathcal{S}) - \delta(f : \mathcal{R}))}{2} \tag{8.6}$$

Exercise 8.6:
For the function $f(u, v) = (u - 1)^2 + (v - 2)^2$ compute the forward simplex gradient on the simplex defined by

$$x_0 = \begin{pmatrix} h \\ 0 \end{pmatrix}, \quad x_1 = \begin{pmatrix} 0 \\ h \end{pmatrix}, \quad x_2 = \begin{pmatrix} h \\ h \end{pmatrix}$$

with $h \neq 0$. Compute the difference $||\nabla f(x_0) - D(f : \mathcal{S})||$.

The simplex matrix is

$$\mathcal{V} = \begin{pmatrix} -h & 0 \\ h & h \end{pmatrix} \qquad \textit{and} \qquad \mathcal{V}^{-\top} = \frac{1}{h}\begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}$$

The function differences are

$$\delta(f : \mathcal{S}) = \begin{pmatrix} f(x_1) - f(x_0) \\ f(x_2) - f(x_0) \end{pmatrix} = \begin{pmatrix} (h-2)^2 - (h-1)^2 - 3 \\ (h-2)^2 - 4 \end{pmatrix}$$

And the forward simplex gradient is

$$\begin{aligned} D(f : \mathcal{S}) = \mathcal{V}^{-\top}\delta(f : \mathcal{S}) &= \frac{1}{h}\begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} (h-2)^2 - (h-1)^2 - 3 \\ (h-2)^2 - 4 \end{pmatrix} \\ &= \frac{1}{h}\begin{pmatrix} (h-1)^2 - 1 \\ (h-2)^2 - 4 \end{pmatrix} = \begin{pmatrix} h-2 \\ h-4 \end{pmatrix} \end{aligned}$$

We easily see $\nabla f(u, v) = (2u - 2, 2v - 4)^\top$ and compute:

$$||\nabla f(x_0) - D(f : \mathcal{S})|| = ||\begin{pmatrix} 2h-2 \\ -4 \end{pmatrix} - \begin{pmatrix} h-2 \\ h-4 \end{pmatrix}|| = ||\begin{pmatrix} h \\ -h \end{pmatrix}|| = \sqrt{2}h$$

Simplex gradients are usually evaluated only to approximate the gradient at the point x_0. The other vertexes $x_1, \dots, x_n$ of a simplex $\mathcal{S}$ are typically the first vertexes of a neighbored simplices $\bar{\mathcal{S}}$, for which again a simplex gradient can be evaluated. We can show:

Lemma 8.7 (Approximation Quality of Simplex Gradients)**:**
Let $\mathcal{S}$ be a regular simplex, let ∇f be Lipschitz continuous in an open neighborhood containing $\mathcal{S}$. Then there is $K \geq 0$ depending on the Lipschitz constant only, such that for the forward simplex gradient holds:

$$||\nabla f(x_0) - D(f : \mathcal{S})|| \leq K\kappa(\mathcal{V})\sigma_+(\mathcal{S}) \tag{8.7}$$

If in addition $\nabla^2 f$ is Lipschitz continuous in an open neighborhood containing $\mathcal{S}$ and $\mathcal{R}$, then there is $K_C \geq 0$ depending on the Lipschitz constant only, such that for the centered simplex gradient holds:

$$||\nabla f(x_0) - D_C(f : \mathcal{S})|| \leq K\kappa(\mathcal{V})\sigma_+(\mathcal{S})^2 \tag{8.8}$$

This means that centered simplex gradients on regular simplices with sufficient small edge lengths are preferred. For the simplex in Exercise 8.6 we get $\kappa(\mathcal{V}) = 1$ and $\sigma_+(\mathcal{S}) = \sqrt{2}h$, so $K \geq 1$ is the corresponding bound.

Because descent algorithms typically only work for exact gradients, the approximation with the simplex gradient can lead to failure in line search algorithms, if the simplex for the approximation is too large. Therefore an outer loop is placed around the basic descent algorithm and whenever the descent algorithm fails, the size of the simplex is halved using the center of the longest edge, $\kappa(\mathcal{V})$ is improved and its last iterate x_k is used as new starting point x_0. The approximation of the Hessian for noisy functions with a simplex approach is not recommended, but quasi-Newton methods or Gauss-Newton iterations can still be executed using the simplex gradient.

Home Exercise 8.1 (Central Simplex Gradient)**:**
Consider the function $f(u, v) = (u - 1)^2 + (v - 2)^2$ and the simplex $\mathcal{S}$ depending on the scale $h \neq 0$ and defined by the points

$$x_0 = \begin{pmatrix} h \\ 0 \end{pmatrix}, \quad x_1 = \begin{pmatrix} 2h \\ 0 \end{pmatrix}, \quad x_2 = \begin{pmatrix} h \\ h \end{pmatrix}$$

- ***a)*** *State the points defining the reflected simplex $\mathcal{R}$ in dependence of h.*
- ***b)*** *Compute the centered simplex gradient $D_C(f : \mathcal{S})$.*
- ***c)*** *Compute the difference $||\nabla f(x_0) - D_C(f : \mathcal{S})||$.*

8.2 Implicit Filtering

In the following approach we want to combine a scaled unit central simplex gradient with quasi-Newton methods.

Definition 8.8 (Scaled Unit Central Simplex Gradient)**:**
*For a **scale** $h > 0$ let $\mathcal{S}_h$ be the simplex at x_0 with the remaining $j = 1, \ldots, n$ vertexes being $x_j = x_0 + he_j$, so $V = h\mathbb{E}$. The reflected vertexes are $x_j^R = x_0 - he_j$ for all $j = 1, \ldots, n$.*

The scaled unit central simplex gradient is:

$$D_C(f : \mathcal{S}_h) = \frac{\delta(f : \mathcal{S}_h) - \delta(f : \mathcal{R}_h)}{2h} =: \nabla_h f(x_0) \tag{8.9}$$

*We say that some $x_* \in \mathbb{R}^n$ leads to a **stencil failure**, if*

$$f(x_*) \leq f(x) \quad \textit{for all} \quad x \in \{x = x_* \pm he_j, j = 1, \ldots, n\} \tag{8.10}$$

The usability of this gradient approximation depends highly on the choice of the scale h:

- If h is sufficiently larger than the wavelength of the noise $\psi(x)$, then $\nabla_h f(x_k)$ is a poor approximation of $\nabla f_s(x_k)$.
- If h is smaller than the wavelength of the noise $\psi(x)$, then $\nabla_h f(x_k)$ is a good approximation of the noise $\nabla\psi(x_k)$. But this is not helpful at all!

Because we do not know the wavelength of the noise $\psi(x)$, we need to define a set of decreasing scales $H := \{h_0, h_1, \ldots, h_m : h_j > h_{j+1} \quad \text{for} \quad j = 0, \ldots, m\}$. Then we execute a descent method for each of these m scales h_j separately until termination.

Definition 8.9 (Termination Criteria for Implicit Filtering)**:**
For a given scale h a descent method in the context of implicit filtering terminates by satisfying at least one of these conditions:

- *If $||\nabla_h f(x_k)|| \leq \varepsilon h$.*
- *If x_k leads to a stencil failure.*
- *If a maximum number of iterations is reached in the main loop (recommended: $200\times$ problem dimension).*
- *If a maximum number of iterations is reached in the line search loop (recommended: 10).*

Algorithm 8.10 (Implicit Filtering with inverse BFGS-Update)**:**
For minimizing noisy functions at scale h_j using a matrix B_k approximating inverse Hessian-information

1. *Input:* $f : \mathbb{R}^n \to \mathbb{R}$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$; $h > 0$; $\sigma \in (0, \frac{1}{2})$.
2. *Set* $x_k \leftarrow x_0$, $B_k \leftarrow \mathbb{E}$.
3. *While the termination criteria from Definition 8.9 are not satisfied do*
   a) *Set* $d_k = -\beta_k B_k \nabla_h f(x_k)$ *with* $\beta_k = \min(1, \frac{10h}{||B_k \nabla_h f(x_k)||})$.
   b) *Find* t_k *with backtracking such that* $f(x_k + t_k d_k) \le f(x_k) + \sigma t_k \nabla_h f(x_k)^\top d_k$.
   c) *Set* $\Delta g_k \leftarrow \nabla_h f(x_k + t_k d_k) - \nabla_h f(x_k)$ *and* $\Delta x_k \leftarrow t_k d_k$.
   d) *Set* $x_k \leftarrow x_k + t_k d_k$.
   e) *Update* B_k *according to the inverse (BFGS) update formula from Lemma 6.6.*
4. *Output:* $x_* \leftarrow x_k$.

The usage of β_k makes sure that the length of the descent direction is bounded. After we have executed Algorithm 8.10 for all m scales h_j for one common starting point, we get up to m different solutions x_{h_j}. We can choose the one with the lowest objective value, call it $x^*_{h_j}$, set $x_0 \leftarrow x^*_{h_j}$ and restart the whole procedure for all scales. We end the restarting procedure, if x_0 is a **minimum at all scales**, i.e. if $f(x_{h_j}) = f(x_0)$ for all scales $j : 1, \dots, m$.

Home Exercise 8.2 (Implicit Filtering)**:**
Consider the black box function $f : \mathbb{R} \to \mathbb{R}$ *with the following properties:*

- *If* f *is evaluated at a integer value* $z \in \mathbb{Z}$, *then the return value is* $f(z) = z^4 + z^3 - 11z^2 - 9z$.
- *If* f *is evaluated at a noninteger value* $x \notin \mathbb{Z}$, *then the return value is* $f(x) = x^4 + x^3 - 11x^2 - 9x + \psi(x)$ *with unknown but small perturbation* ψ.
- *If Algorithm 8.10 is applied with some starting point* x_0 *and some scale* h, *the return value* x^*_h *is*

$$x^*_h = \begin{cases} x_0 - h & \text{if} \quad f(x_0 - h) < f(x_0) \quad \text{and} \quad f(x_0 - h) \le f(x_0 + h) \\ x_0 + h & \text{if} \quad f(x_0 + h) < f(x_0) \quad \text{and} \quad f(x_0 + h) < f(x_0 - h) \\ x_0 & \text{else} \end{cases}$$

a) *For all scales* $h \in \{1, 2, 3\}$ *state all return values* x^*_h *for the starting point* $x_0 = 0$.

b) *Show that* $x_* = 2$ *is a minimum at all scales* $h \in \{1, 2, 3\}$.

9 Appendix

9.1 The Benchmark Problem

The following benchmark problem is designed to show the application of optimization in real-world situations. Many techniques in this lecture are used in different steps to solve the benchmark problem. The different steps can be executed with the optimization routines from the programming homeworks.

The benchmark problem is defined as follows:

$$\begin{aligned} \text{minimize} \quad & f(u,v,w) = \alpha(v+1)u^2 + \exp(\beta w + 1)v^2 + \gamma\sqrt{u+1}w^2 + \psi(u,v,w) \\ \text{such that} \quad & x = (u,v,w)^\top \in \Omega_\square := [0,8] \times [-4,4] \times [-1,1] \\ \text{and} \quad & h(u,v,w) = (u-4)^2 + v^2 + w^2 - 9 = 0. \end{aligned}$$

The parameters α, β and γ are unknown.

The first task is to estimate α, β and γ with the Levenberg-Marquardt method by using the following measure data:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_i	0	8	0	8	0	8	0	8	4	0	8	4	4	4	4
v_i	-4	-4	4	4	-4	-4	4	4	0	0	0	-4	4	0	0
w_i	-1	-1	-1	-1	1	1	1	1	0	0	0	0	0	-1	1
f_i	22	-522	22	1014	337	-207	337	1329	48	0	192	-101	283	84	84

This requires to model an error vector $R(\alpha,\beta,\gamma)$ and its Jacobian $J(\alpha,\beta,\gamma)$ for the benchmark problem as well as a handle for the box constraints and the equality constraint. The method converges for trivial starting points to the parameter set $\alpha_* \approx 3$, $\beta_* \approx 2$ and $\gamma_* \approx 16$.

Our goal is now to minimize the objective $f(u,v,w)$ for the computed parameter set with the augmented Lagrangian method in combination with implicit filtering. This requires

- Outer loop with augmented Lagrangian.
- Implicit Filtering with projected inverse BFGS-Update inside of augmented Lagrangian.
- Additional safeguards to make sure that augmented Lagrangian does not get stuck.

10 Home Exercise Solutions

Solution of Home Exercise 1.1

a) f is quadratic with $A = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
and least squares with $r_j = \sqrt{2}(u - v)$.

b) $\Omega_\Box = [0, 2] \times [-2, -1]$.

c) $\Omega = \{g_1 = -u \leq 0, g_2 = u - 2 \leq 0, g_3 = -v - 2 \leq 0, g_4 = v + 1 \leq 0\}$. $\Box$

Solution of Home Exercise 1.2

a) $x_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ and $x_2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$ are in Ω, but $x_3 = \frac{1}{2}x_1 + \frac{1}{2}x_2 = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}$ is not:

$$(\frac{-3}{2})^2 + 1^2 = \frac{13}{4} < 4 \Rightarrow \quad \text{not convex}$$

b) Let $x_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $x_5 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$ and $\lambda = \frac{1}{2}$:

$$\lambda f(x_4) + (1 - \lambda) f(x_5) = \frac{1}{2} - \frac{1}{4e} + \frac{1}{2} - \frac{1}{4e^9} = \frac{4e^9 - e^8 - 1}{4e^9}$$

but

$$f(\lambda x_4 + (1 - \lambda)x_5) = f(\begin{pmatrix} 2 \\ 0 \end{pmatrix}) = 1 - \frac{1}{2e^4} = \frac{4e^9 - 2e^5}{4e^9} > \frac{4e^9 - e^8 - 1}{4e^9}$$

so f is not convex on $\mathbb{R}^2$.

c) We can estimate the infimum by looking for the smallest possible $u^2 + v^2$, because this maximizes $\exp(-(u^2 + v^2))$. Now $(u - 1)^2 + v^2 \geq 4$ is equal to $u^2 + v^2 \geq 3 + 2u$. So $u^2 + v^2$ is bounded from below by $3 + 2u$ but also by zero. u itself is

- bounded from above by $-\sqrt{4 - v^2} + 1$, which is closest to zero for $v = 0$ and implies $u = -1$ and $u^2 + v^2 = 1$.
- bounded from below by $\sqrt{4 - v^2} + 1$, which is closest to zero for $v = \pm 2$ and implies $u = 1$ and $u^2 + v^2 = 5$.

The infimum is therefore attained at the (GMP) $x_* = (-1, 0)^\top$. $\Box$

Solution of Home Exercise 1.3
a) The (GMP) on $\mathbb{R}$ is obviously $x_* = 1$, because the infimum of zero is reached for $x_* = 1$.

b) In the interval $[-\alpha, \alpha]$ for $\alpha \in (0,1)$ the objective is strictly decreasing, the lowest value is attained at α. $x_*(\alpha) = \alpha$.

c) The function $(0,1) \to \Omega_\alpha$ mapping α to the (GMP) $x_*(\alpha)$ is:

$$\alpha \mapsto \begin{cases} \alpha & \text{if } \alpha \in (0,1) \\ 1 & \text{if } \alpha \in [1,\infty) \end{cases}$$

This mapping is continuous, so the (GMP) depends continuously on α. $\square$

Solution of Home Exercise 2.1
a)

$$\nabla f(u,v) = \begin{cases} \begin{pmatrix} 6u^5 \\ 2v \end{pmatrix}, & \text{if} \quad u^6 + v^2 > \alpha \quad \text{or} \quad \alpha = 0 \\ \quad \text{not continuous} \quad , & \text{if} \quad u^6 + v^2 = \alpha > 0 \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, & \text{if} \quad u^6 + v^2 < \alpha \end{cases}$$

$$\nabla^2 f(u,v) = \begin{cases} \begin{pmatrix} 30u^4 & 0 \\ 0 & 2 \end{pmatrix}, & \text{if} \quad u^6 + v^2 > \alpha \quad \text{or} \quad \alpha = 0 \\ \quad \text{not continuous} \quad , & \text{if} \quad u^6 + v^2 = \alpha > 0 \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, & \text{if} \quad u^6 + v^2 < \alpha \end{cases}$$

The set of points satisfying the first and second order necessary optimality conditions are $\mathcal{A} := \{(u,v)^\top \in \mathbb{R}^2 : u^6 + v^2 < \alpha \quad \text{or} \quad (u,v) = (0,0)\}$.

b) $\inf_{(u,v)^\top \in \mathbb{R}^2} f(u,v) = \alpha$. We see $f(u,v) = \alpha$ for all $(u,v) \in \mathcal{A}$. So all points in $\mathcal{A}$ are (GMPs). A strict (GMP) only exists for $\alpha = 0$ and is $(u_*, v_*)^\top = (0,0)^\top$.

c) Let $z_* \in \mathbb{Z}^2$, let $B_1(z_*) := \{x \in \mathbb{Z}^2 : ||x - z_*|| < 1\}$. We see, that the environment $B_1(z_*) \setminus z_* = \emptyset$, so $f(z_*) < f(x)$ for all $x \in B_1(z_*)$ holds. $\square$

Solution of Home Exercise 2.2
a)

$$\nabla f(u,v,w) = \begin{pmatrix} \exp(u+v) + \exp(u-v) \\ \exp(u+v) - \exp(u-v) \\ \cos(w) \end{pmatrix}$$

$$\nabla^2 f(u,v,w) = \begin{pmatrix} \exp(u+v)+\exp(u-v) & \exp(u+v)-\exp(u-v) & 0 \\ \exp(u+v)-\exp(u-v) & \exp(u+v)+\exp(u-v) & 0 \\ 0 & 0 & -\sin(w) \end{pmatrix}$$

b)

$$\det(\nabla^2 f(u,v,w) - \lambda) =$$
$$(-\sin(w)-\lambda)\det\begin{pmatrix} \exp(u+v)+\exp(u-v)-\lambda & \exp(u+v)-\exp(u-v) \\ \exp(u+v)-\exp(u-v) & \exp(u+v)+\exp(u-v)-\lambda \end{pmatrix} =$$
$$-(\sin(w)+\lambda)\left((\exp(u+v)+\exp(u-v)-\lambda)^2 - (\exp(u+v)-\exp(u-v))^2\right) = 0$$

is solved by the eigenvalues $\lambda_1 = 2\exp(u+v)$, $\lambda_2 = 2\exp(u-v)$, $\lambda_3 = -\sin(w)$.

c) $\nabla^2 f(u,v,w)$ is s.p.d., if all eigenvalues are positive. λ_1 and λ_2 are positive for all $u, v \in \mathbb{R}$. $\lambda_3 = -\sin(w)$ is positive for $(2k-1)\pi < w < 2k\pi$ with $k \in \mathbb{Z}$.

d) There are no (LMPs), the first component of the gradient is never zero. □

Solution of Home Exercise 2.3

$$\nabla f(u,v,w) = \begin{pmatrix} -2w(u-1) \\ 2v \\ -(u-1)^2 \end{pmatrix}, \qquad \nabla^2 f(u,v,w) = \begin{pmatrix} -2w & 0 & -2(u-1) \\ 0 & 2 & 0 \\ -2(u-1) & 0 & 0 \end{pmatrix}$$

For $x_1 = (1,0,0)^\top$:
$P(x_1 - t\nabla f(x_1)) = P(x_1) = x_1$ for all $t > 0$, so x_1 is stationary. The active set is $\mathcal{A}_1 = \{2,3\}$. But $\nabla f(x_1) = 0$, which does not match $\mathcal{A}_1$, so x_1 is only degenerate stationary. The reduced Hessian is generated using the Hessian:

$$\nabla^2 f(x_1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \nabla^2_{\Omega_\square} f(x_1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is s.p.s. We conclude: x_1 could be a (LMP).

For $x_2 = (1,0,2)^\top$:
$P(x_2 - t\nabla f(x_2)) = x_2$ for all $t > 0$, so x_2 is stationary. The active set is $\mathcal{A}_2 = \{2,3\}$ and $\nabla f(x_2) = 0$, so x_2 is only degenerate stationary. Reduced Hessian:

$$\nabla^2 f(x_2) = \begin{pmatrix} -4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \nabla^2_{\Omega_\square} f(x_2) = \begin{pmatrix} -4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This is not s.p.s. We conclude: x_2 is no (LMP).

For $x_3 = (0,0,2)^\top$:
$P(x_3 - t\nabla f(x_3)) = P((-4t, 0, 2+t)^\top) = x_3$ for all $t > 0$, so x_3 is stationary. The active set is $\mathcal{A}_3 = \{1,2,3\}$, but $\nabla f(x_3) = (4,0,-1)^\top$ means that x_3 is degenerate stationary. The reduced Hessian is $\nabla^2_{\Omega_\square} f(x_3) = \mathbb{E}$ and therefore s.p.d. Because of the degeneracy we can only conclude that x_3 could be a (LMP). $\square$

Solution of Home Exercise 2.4

a) For $y = (u_y, v_y) \in \Omega$ we realize $\nabla f(x_*)^\top (y - x_*) = (u_y + 2) \geq 0$ because $u_y \in [-2,2]$ holds.

b)

$$\begin{aligned} \text{minimize} \quad & \tilde{f}(r,\phi) = r\cos(\phi) \\ \text{s.t.} \quad & \tilde{x} = (r,\phi)^\top \in \tilde{\Omega} := [0,2] \times [0, 2\pi] \end{aligned}$$

c) $(-2,0)^\top = (r_* \cos(\phi_*), \frac{1}{2} r_* \sin(\phi_*))^\top$ is solved by $r_* = 2$ and $\phi_* = \pi$. We check:
Stationarity:

$$P(\tilde{x}_* - t\nabla \tilde{f}(\tilde{x}_*)) = P(\begin{pmatrix} 2+t \\ \pi \end{pmatrix}) = \begin{pmatrix} 2 \\ \pi \end{pmatrix} = \tilde{x}_* \quad \text{for all} \quad t > 0 \quad \checkmark$$

Nondegeneracy:
Active set: $\mathcal{A} = \{1\}$

$$\nabla \tilde{f}(r_*, \phi_*) = \begin{pmatrix} \cos(\phi_*) \\ -r_* \sin(\phi_*) \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad \checkmark$$

Reduced Hessian:

$$\nabla^2 \tilde{f}(r_*, \phi_*) = \begin{pmatrix} 0 & -\sin(\phi_*) \\ -\sin(\phi_*) & -r_* \cos(\phi_*) \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}, \qquad \nabla^2_{\tilde{\Omega}} \tilde{f}(r_*, \phi_*) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \quad \checkmark$$

$\square$

Solution of Home Exercise 2.5

a) Because L is convex, the projection problem is uniquely solved by u_* satisfying $\nabla g(u_*) = 0$. The gradient is $\nabla g(u) = \big(2u - u_0 + v_0 - 2\big)$. So $u_* = 1 + \frac{u_0 - v_0}{2}$. We end up with $P_L : (u_0, v_0)^\top \mapsto (1 + \frac{u_0 - v_0}{2}, 1 - \frac{u_0 - v_0}{2})^\top$.

b) $P(x_1) = x_1$ because $x_1 \in \Omega$.

The projection of x_2 has to be on L, because this is the closest boundary. The closest point on L next to x_2 is $P_L(x_2) = \frac{1}{2}(1,3)^\top$. This is also a point in Ω, so $P(x_2) = P_L(x_2)$.

At first glance x_3 is close to both L and the upper bound $u \equiv 1$. But $P_L(x_3) = \frac{1}{2}(3,1)^\top$ is not in Ω. The projection on the upper bound, $P_\Box(x_3) = (1,2)^\top$, is also not in Ω. Instead $P(x_3) = (1,1)^\top \in \Omega$, which is the common point of L and $u \equiv 1$.

c)

$$P(x^* - t\nabla f(x^*)) = P(\begin{pmatrix} -t \\ -t \end{pmatrix}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = x^* \quad \text{for all} \quad t > 0.$$

□

Solution of Home Exercise 2.6

a) Ω can be simplified to $\Omega := \{(u,v)^\top \in [-\pi, 0] \times \{0\} \quad \text{or} \quad (u,v)^\top = (\pi,0)^\top\}$. This is not convex because $(0,0)^\top$ and $(\pi,0)^\top$ are in Ω but $\frac{1}{2}(\pi,0)^\top$ is not.

b) h is always active. We have to check different cases for other active constraints: If $(u,v)^\top = (\pi,0)^\top$, then the box constraint in u is active and g is active. The (LICQ) vector set is $C_1 := \{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} \cos(\pi) \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$. This set C_1 is linearly dependent, the (LICQ) is not satisfied.

If $(u,v)^\top = (-\pi,0)^\top$, then the box constraint in u is active and g is active. The (LICQ) vector set $C_2 := \{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} \cos(-\pi) \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$ is linearly dependent, the (LICQ) is not satisfied.

If $(u,v)^\top = (0,0)^\top$, then g is active. The (LICQ) vector set $C_3 := \{\begin{pmatrix} \cos(0) \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$ is linearly independent, the (LICQ) is satisfied.

If $(u,v)^\top \in (-\pi,0) \times \{0\}$, then there is no active constraint other than h. The (LICQ) vector set $C_4 := \{\begin{pmatrix} 0 \\ 1 \end{pmatrix}\}$ is linearly independent, the (LICQ) is satisfied.

c) To find the (GMPs) of f on Ω we need to find the infimum of f on $[-\pi,0] \times \{0\} \cup \{(\pi,0)^\top\}$. The infimum is obviously $1-\pi^2$ and is achieved by the points $x^*_{1/2} = (\pm\pi, 0)$. □

Solution of Home Exercise 2.7

a) $L(u,v,w,\lambda,\mu) = \sinh(u) - u + 4w - \mu u + \lambda(v^2 + w^2 + 4w)$

and $\nabla_x L(u,v,w,\lambda,\mu) = \begin{pmatrix} \cosh(u) - 1 - \mu \\ 2\lambda v \\ 4 + \lambda(2w+4) \end{pmatrix}$.

b) We have to distinguish, if $-u \leq 0$ is active or not. If $u > 0$ then $\mu = 0$, leading to $\cosh(u) - 1 = 0$, which is not solvable for $u > 0$. So $u_* = 0$ is the only solution with (noncomplementary) multiplier $\mu_* = 0$. For v_*, w_*, λ_* we need to solve

$$\begin{pmatrix} 2\lambda v \\ 4 + \lambda(2w + 4) \\ v^2 + w^2 + 4w \end{pmatrix} = 0$$

The solutions are $v_* = 0$ and $w_1^* = 0$ or $w_2^* = -4$ with $\lambda_1^* = -1$ or $\lambda_2^* = 1$.

c) We check $x_1^* = (0,0,0)^\top$: $C := \{ \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} \}$ is linearly independent, the (LICQ) is satisfied. And for $x_2^* = (0,0,-4)^\top$ we get the set $C := \{ \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -4 \end{pmatrix} \}$, so the (LICQ) is again satisfied. □

Solution of Home Exercise 3.1

a)

$$\nabla\phi(t) = \nabla f(x_0 + td_0)^\top d_0 = 5(1-t)^4 + 8(1-t)^3 + 3(1-t)^2 = (1-t)^2(2-t)(\frac{16}{10} - t) \overset{!}{=} 0$$

is solved by $t_1 = 1$, $t_2 = 2$ and $t_3 = \frac{16}{10}$.

b)

$$\nabla^2\phi(t) = {d_0}^\top \nabla^2 f(x_0 + td_0)d_0 = 20(1-t)^3 + 24(1-t)^2 + 6(1-t) \overset{!}{\geq} 0.$$

We get $\nabla^2\phi(t_1) = 0$, $\nabla^2\phi(t_2) = -2 < 0$ and $\nabla^2\phi(t_3) = \frac{18}{25} > 0$. Second order condition is satisfied for t_1 and t_3.

c) We get $f(x_0 + t_1 d_0) = f(0,0) = 0$, $f(x_0 + t_2 d_0) = f(-1,0) = 0$, $f(x_0 + t_3 d_0) = f(-0.6, 0) = -\frac{108}{3125} < 0$. t_3 is the optimal step size. □

Solution of Home Exercise 3.2

a) The eigenvalues are $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = \frac{1}{2}$. A set of eigenvectors is for example $v_1 = (-1,0,1)^\top$, $v_2 = (0,1,0)^\top$, $v_3 = (1,0,1)^\top$.

b)

$$ {v_k}^\top A v_{\tilde{k}} = {v_k}^\top \lambda_{\tilde{k}} v_{\tilde{k}} = 0 \quad \text{for all} \quad k \neq \tilde{k}$$

c)

$$A^{-1} = \sum_{i=1}^{3} \frac{1}{{v_i}^\top A v_i} v_i {v_i}^\top = \frac{1}{2}\begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 3 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 3 \end{pmatrix}$$

□

Solution of Home Exercise 3.3

a)

$$d_0 = p_0 = (1,0,0)^\top \quad \text{and} \quad d_1 = p_1 - \frac{{p_1}^\top A d_0}{{d_0}^\top A d_0} d_0 = (0,1,0)^\top.$$

$$d_2 = p_2 - \frac{{p_2}^\top A d_0}{{d_0}^\top A d_0} d_0 - \frac{{p_2}^\top A d_1}{{d_1}^\top A d_1} d_1 = (0,0,1)^\top - (\frac{1}{3},0,0)^\top - (0,0,0)^\top = \frac{1}{3}(-1,0,3)^\top.$$

b)

$$x_0 = \begin{pmatrix} 1 \\ 4 \\ 0 \end{pmatrix} \quad \text{and} \quad d_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad t_0 = \frac{1}{3} \quad \text{so}$$

$$x_1 = \frac{1}{3}\begin{pmatrix} 4 \\ 12 \\ 0 \end{pmatrix} \quad \text{and} \quad d_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad t_1 = 0 \quad \text{so}$$

$$x_2 = \frac{1}{3}\begin{pmatrix} 4 \\ 12 \\ 0 \end{pmatrix} \quad \text{and} \quad d_2 = \frac{1}{3}\begin{pmatrix} -1 \\ 0 \\ 3 \end{pmatrix} \quad \text{and} \quad t_2 = 1 \quad \text{so} \quad x_3 = \begin{pmatrix} 1 \\ 4 \\ 1 \end{pmatrix}$$

c)$\nabla f(x_1) = \nabla f(x_2) = Ax_2 - b = \frac{1}{3}(0,0,-8)$ is orthogonal to both p_0 and p_1. $\nabla f(x_3) = Ax_2 - b = 0$ holds.

□

Solution of Home Exercise 3.4

a)

$$A = \nabla^2 f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad \text{and} \quad b = Ax - \nabla f = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

b)

$$x_0 = b = \begin{pmatrix}1\\1\\0\\1\end{pmatrix} \quad \text{and} \quad r_0 = \begin{pmatrix}0\\2\\0\\1\end{pmatrix} \quad \text{and} \quad d_0 = -r_0 = \begin{pmatrix}0\\-2\\0\\-1\end{pmatrix} \quad \text{and} \quad t_0 = \frac{5}{14} \quad \text{so}$$

$$x_1 = \frac{1}{14}\begin{pmatrix}14\\4\\0\\9\end{pmatrix} \quad \text{and} \quad r_1 = \frac{1}{14}\begin{pmatrix}0\\-2\\0\\4\end{pmatrix} \quad \text{and} \quad d_1 = \frac{1}{14\cdot 7}\begin{pmatrix}0\\10\\0\\-30\end{pmatrix} \quad \text{and} \quad \tau_1 = \frac{7}{15}$$

$$\text{so} \quad x_2 = \frac{1}{6}\begin{pmatrix}6\\2\\0\\3\end{pmatrix} \quad \text{and} \quad r_2 = \begin{pmatrix}0\\0\\0\\0\end{pmatrix}.$$

The algorithm terminates with $x_* = x_2$.

c) The eigenvalues of A are $\lambda_1 = 1$, $\lambda_2 = 3$, $\lambda_3 = 2$, $\lambda_4 = 2$. All eigenvalues are positive, so A is s.p.d. $Ax_2 = (1,1,0,1)^\top = b$ holds. □

Solution of Home Exercise 4.1

a) We choose for example $d_0 = -\nabla f(x_0) = 2(r_0\cos(\phi_0), r_0\sin(\phi_0))^\top$.

b) At $x_* = (0,0)^\top$ the objective f reaches its supremum, the point x_* is a global maximal point. So every direction $d_* \neq 0$ is a descent direction.

c) At x_k let d_k satisfy $f(x_k + t_k d_k) < f(x_k)$ for all $t_k \in (0, \varepsilon_k]$. Then also holds $f(x_k + t_k\alpha d_k) < f(x_k)$ for all $t_k \in (0, \varepsilon]$ with $\varepsilon := \frac{\varepsilon_k}{\alpha}$. So αd_k is a descent direction. □

Solution of Home Exercise 4.2

a) For $x < 0$ the gradient takes the form $\nabla f(x) = \frac{-1}{2\sqrt{-x}}$. We see $\nabla f(-1)^\top \frac{3}{2} = -\frac{3}{4} < 0$, so this is a descent direction.

b) We start with $f(x_0 + t_* d_0) \leq f(x_0) + \frac{1}{4} t_* \nabla f(x_0)^\top d_0$ leading to $\sqrt{|t_* \frac{3}{2} - 1|} \leq 1 - \frac{3}{16} t_*$. We have to split this in cases:

Case 1, $t \in (0, \frac{2}{3}]$: $\sqrt{1 - t_* \frac{3}{2}} \leq 1 - \frac{3}{16} t_*$ is solved for $t_* \geq 0$. Case 2, $t \in (\frac{2}{3}, \frac{16}{3}]$: $\sqrt{t_* \frac{3}{2} - 1} \leq 1 - \frac{3}{16} t_*$ is solved for $t_* \leq \frac{16}{3}(5 - \sqrt{23})$. Case 3, $t \in (\frac{16}{3}, \infty)$: No solution exists. All in all for $t \in (0, \frac{16}{3}(5 - \sqrt{23})]$ the sufficient decrease condition holds.

c) We start with $\nabla f(x_0 + t_* d_0)^\top d_0 \geq \frac{1}{2} \nabla f(x_0)^\top d_0$ leading to $-2 \leq \sqrt{|t_* \frac{3}{2} - 1|}$ for

$t > \frac{2}{3}$. This is solved in the positive range for $t \in (\frac{2}{3}, \infty)$.

d) f is not continuously differentiable, ∇f is not Lipschitz-continuous. Check not possible for $d_0 = 1$.

e) We start with $(t_a, t_b, t_c, t_d) \leftarrow (0, 0.38, 0.61, 1)$, $(\phi_b, \phi_c) \leftarrow (0.65, 0.27)$, $|t_d - t_a| \leftarrow 1$ (all values are rounded).
We iterate:
$(t_a, t_b, t_c, t_d) \leftarrow (0.38, 0.61, 0.76, 1)$, $(\phi_b, \phi_c) \leftarrow (0.27, 0.38)$, $|t_d - t_a| \leftarrow 0.61$.
$(t_a, t_b, t_c, t_d) \leftarrow (0.38, 0.53, 0.61, 0.76)$, $(\phi_b, \phi_c) \leftarrow (0.45, 0.27)$, $|t_d - t_a| \leftarrow 0.38$.
$(t_a, t_b, t_c, t_d) \leftarrow (0.53, 0.61, 0.67, 0.76)$, $(\phi_b, \phi_c) \leftarrow (0.27, 0.10)$, $|t_d - t_a| \leftarrow 0.24$.
We terminate with $t_* = 0.65$.

□

Solution of Home Exercise 4.3

a)

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad d_0 = -Ax_0 = \begin{pmatrix} -2 \\ -1 \end{pmatrix} \quad \text{and} \quad t_0 = \frac{1}{2} \quad \text{so}$$
$$x_1 = \frac{1}{2}\begin{pmatrix} 0 \\ -1 \end{pmatrix} \quad \text{and} \quad d_1 = -Ax_1 = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad t_1 = \frac{1}{2} \quad \text{so}$$
$$x_2 = \frac{1}{4}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

b) No, $\nabla f(x_2) \neq 0$.

c) We need to satisfy $\nabla f(x_*) = Ax_* \overset{!}{=} 0$. The Hessian $\nabla^2 f = A$ is s.p.d. because $\det(A_{1,1}) = 2 > 0$ and $\det(A) = 1 > 0$. Especially A is regular, so $Ax_* = 0$ is solved only by $x_* = (0, 0)^\top$. At $x_* = (0, 0)^\top$ we satisfy $\nabla f(x_*) = 0$ and $\nabla^2 f$ is s.p.d. for all $x \in \mathbb{R}^2$, leading to a (GMP) on $\mathbb{R}^2$. □

Solution of Home Exercise 4.4

a)

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad d_0 = -Ax_0 = \begin{pmatrix} -2 \\ -1 \end{pmatrix} \quad \text{and} \quad t_0 = 1 \quad \text{so}$$
$$x_1 = P(\begin{pmatrix} -1 \\ -1 \end{pmatrix}) = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \text{and} \quad d_1 = -Ax_1 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad \text{and} \quad t_1 = 1 \quad \text{so}$$
$$x_2 = x_1$$

b) $x_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ is active in both components, but $\nabla f(x_2)$ has a zero in the second component. The point is degenerate and the sufficient condition cannot be applied.

c) We enlarge the box constraints in the degenerate component to $\tilde{\Omega}_\square := [1,2]\times[-2,1]$. The computations from above will lead to the same x_2 satisfying stationarity, but the second component is no longer active and nondegeneracy is satisfied. A is already s.p.d., so the reduced matrix $A_{\tilde{\Omega}_\square}$ is also s.p.d. The (GMP) of f on $\tilde{\Omega}_\square$ is also a member of the smaller set $\Omega_\square$.

□

Solution of Home Exercise 5.1

a) Look at

$$\frac{|a_{k+1}|}{|a_k|} = \exp(-k-1+k) = \frac{1}{e} < 1,$$

so the convergence rate is Q-linear, but not Q-superlinear.

b) Look at

$$\frac{|b_{k+1}|}{|b_k|} = \frac{\sqrt{(k+1)^{-1}}}{\sqrt{k^{-1}}} = \sqrt{1-\frac{1}{k+1}} \overset{k\to\infty}{\to} 1,$$

so the convergence rate is slower than Q-linear.

c) Look at

$$\frac{|c_{k+1}|}{|c_k|} = \frac{k!}{(k+1)!} = \frac{1}{k+1} \overset{k\to\infty}{\to} 0,$$

so the convergence rate is at least Q-superlinear.
And

$$\frac{|c_{k+1}|}{|c_k|^2} = \frac{k!k!}{(k+1)!} = \frac{k!}{k+1}$$

is unbounded, so the convergence rate is slower than Q-quadratic.

d)

$$\frac{|\Delta t_{k+1}|}{|\Delta t_k|} = \frac{\sqrt{5}-1}{2} < 1,$$

so the convergence rate of golden section line search is Q-linear. □

Solution of Home Exercise 5.2

a) We compute

$$\nabla f(u,v) = \begin{pmatrix} 5u^4 - 8v \\ -8u + 4v \end{pmatrix}, \qquad \nabla^2 f(u,v) = \begin{pmatrix} 20u^3 & -8 \\ -8 & 4 \end{pmatrix}$$

We check the leading principal minors of the Hessian: $\det(20u^3)$ is bigger than zero for $u > 0$ and $\det(\nabla^2 f(u,v)) = 80u^3 - 64 > 0$ for $u > \sqrt[3]{\frac{4}{5}}$, which is sufficient for the Hessian being s.p.d.

b)

$$\nabla f(x_0) = \begin{pmatrix} 5 \\ -8 \end{pmatrix}, \qquad \nabla^2 f(x_0) = \begin{pmatrix} 20 & -8 \\ -8 & 4 \end{pmatrix}$$

Now d_0 has to satisfy $\nabla^2 f(x_0) d_0 = -\nabla f(x_0)$, or $\begin{pmatrix} 20 & -8 \\ -8 & 4 \end{pmatrix} d_0 = \begin{pmatrix} -5 \\ 8 \end{pmatrix}$. The solution is $d_0 = \frac{1}{4}\begin{pmatrix} 11 \\ 30 \end{pmatrix}$ and $x_1 = x_0 + d_0 = \frac{1}{4}\begin{pmatrix} 15 \\ 30 \end{pmatrix}$.

c)

$$\nabla f(x_*) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \nabla^2 f(u,v) = \begin{pmatrix} 0 & -8 \\ -8 & 4 \end{pmatrix}$$

Because $\det(\nabla^2 f(x_*)) = -64 < 0$, one of the two eigenvalues is negative. x_* does not satisfy the second order optimality conditions. □

Solution of Home Exercise 5.3

a)

$$\begin{aligned} Q_f(x; x_k) &= f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}\nabla^2 f(x_k)(x - x_k)^2 \\ &= (-\frac{1}{24}x_k^3 + \frac{3}{4}x_k^2 - 4x_k + 6) + (-\frac{1}{8}x_k^2 + \frac{3}{2}x_k - 4)(x - x_k) + (-\frac{1}{8}x_k^2 + \frac{3}{4})(x - x_k)^2 \\ &= x^2(-\frac{1}{8}x_k + \frac{3}{4}) + x(\frac{1}{8}x_k^2 - 4) + (-\frac{1}{24}x_k^3 + 6) \end{aligned}$$

b) At $x_0 = 8$ the model is $Q_f(x; x_0) = -\frac{1}{4}x^2 + 4x - \frac{46}{3}$. Because this is a concave parabola, the (GMP) on Ω_0 is at the boundary. We check $Q_f(6; x_0) = -\frac{1}{3}$ and $Q_f(10; x_0) = -\frac{1}{3}$, so we choose the smaller $x_1 = 6$ for the new iterate. We get

$$\sigma_1 = \frac{f(x_0) - f(x_1)}{Q_f(x_0; x_0) - Q_f(x_1; x_0)} = \frac{2/3 - 0}{2/3 + 1/3} = \frac{2}{3}$$

c) At $x_1 = 6$ the model is $Q_f(x; x_1) = \frac{1}{2}x - 3$. This linear function is obviously minimal at the boundary $x = 4$. So $x_2 = 4$ is the next iterate. We get

$$\sigma_2 = \frac{f(x_1) - f(x_2)}{Q_f(x_1; x_1) - Q_f(x_2; x_1)} = \frac{0 + 2/3}{0 + 1} = \frac{2}{3}$$

d) At $x_2 = 4$ the model is $Q_f(x; x_2) = \frac{1}{4}x^2 - 2x + \frac{10}{3}$. This parabola is convex, so its (GMP) is at the vertex $x = 4 \in \Omega_2$, so $x_3 = 4$, which is in fact a (LMP) of f. □

Solution of Home Exercise 6.1

a) The leading principal minors of H_0 are $\det(5) > 0$ and $\det(H_0) = 1 > 0$, so H_0 is positive definite and obviously symmetric. The inverse is $B_0 = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}$.

b) We easily verify $\nabla f(u,v) = (u,v)^\top$ and $\nabla^2 f(u,v) = \mathbb{E}$. We get $d_0 = -B_0 \nabla f(x_0) = -\begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \end{pmatrix}$. Because the problem is quadratic, exact line search means $t_0 = -\frac{\nabla f(x_0)^\top d_0}{d_0{}^\top \mathbb{E} d_0} = \frac{1}{5}$. This means $x_1 = \frac{1}{5}(4,-2)^\top$.

c) We see $\Delta g_0 = \nabla f(x_1) - \nabla f(x_0) = x_1 - x_0 = t_0 d_0 = -\frac{1}{5}(1,2)^\top = \Delta x_0$. The BFGS update now demands:

$$H_1 = H_0 + \frac{\Delta g_0 \Delta g_0{}^\top}{\Delta g_0{}^\top \Delta x_0} - \frac{H_0 \Delta x_0 \Delta x_0{}^\top H_0}{\Delta x_0{}^\top H_0 \Delta x_0}$$
$$= \begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix} + \frac{1}{5}\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} - \begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix}\begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix} = \frac{1}{5}\begin{pmatrix} 21 & -8 \\ -8 & 9 \end{pmatrix}.$$

$\square$

Solution of Home Exercise 6.2

a)

$$f(a,b,c) = \frac{1}{2}\sum_{\phi=0}^{2} ||\gamma(\phi; a,b,c) - \gamma_\phi||^2$$
$$= \frac{1}{2}\left(|| \begin{pmatrix} a \\ 0 \\ 0 \end{pmatrix} ||^2 + || \begin{pmatrix} a\cos(1) \\ \sin(b) - 1 \\ c - 1 \end{pmatrix} ||^2 + || \begin{pmatrix} a\cos(2) \\ \sin(2b) \\ 2c - 2 \end{pmatrix} ||^2 \right).$$

leads to

$$R(a,b,c) = \begin{pmatrix} a \\ a\cos(1) \\ \sin(b) - 1 \\ c - 1 \\ a\cos(2) \\ \sin(2b) \\ 2c - 2 \end{pmatrix}$$

b)

$$J(a,b,c) = \begin{pmatrix} 1 & 0 & 0 \\ \cos(1) & 0 & 0 \\ 0 & \cos(b) & 0 \\ 0 & 0 & 1 \\ \cos(2) & 0 & 0 \\ 0 & 2\cos(2b) & 0 \\ 0 & 0 & 2 \end{pmatrix}, \qquad J^\top R = \begin{pmatrix} a(1+\cos(1)^2+\cos(2)^2) \\ \cos(b)(\sin(b)-1)+2\cos(2b)\sin(2b) \\ 5c-5 \end{pmatrix}$$

$$A(a,b,c) = J^\top J = \begin{pmatrix} 1+\cos(1)^2+\cos(2)^2 & 0 & 0 \\ 0 & \cos(b)^2+4\cos(2b)^2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

c)

$$\nabla f(0,\frac{\pi}{2},1) = J^\top R = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 0$$

$$A(0,\frac{\pi}{2},1) = J^\top J = \begin{pmatrix} 1+\cos(1)^2+\cos(2)^2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 5 \end{pmatrix} \quad \text{is s.p.d.}$$

□

Solution of Home Exercise 6.3

a)

$$R(u,v) = \begin{pmatrix} u-3 \\ uv-3 \\ uv^2-3 \end{pmatrix}, \qquad J(u,v) = \begin{pmatrix} 1 & 0 \\ v & u \\ v^2 & 2uv \end{pmatrix}$$

b) For Levenberg-Marquardt we need $J(x_0)^\top J(x_0) + \alpha_0 \mathbb{E} = \begin{pmatrix} 3+\alpha_0 & 0 \\ 0 & \alpha_0 \end{pmatrix}$. So d_0 solves $\begin{pmatrix} 3+\alpha_0 & 0 \\ 0 & \alpha_0 \end{pmatrix} d_0 = -J(x_0)^\top R(x_0) = \begin{pmatrix} 9 \\ 0 \end{pmatrix}$. We see $d_0 = (\frac{9}{3+\alpha_0}, 0)^\top$, leading to $x_1 = x_0 + d_0 = (\frac{9}{3+\alpha_0}, 1)^\top$.

c)

$$\lim_{\alpha_0 \to 0} x_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad \text{and} \quad R(3,1) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

If the error vector is zero, the (GMP) is reached, because $f(x_1) = \frac{1}{2}R^\top R = 0 = \inf f(x)$, so the definition of the (GMP) is satisfied. $\square$

Solution of Home Exercise 7.1

a) $B(u, v; \beta_1, \beta_2) = 2 - v - \beta_1 \ln(1 - u - v) - \beta_2 \ln(1 + u - v)$.

b) $\nabla B(u_\beta, v_\beta) = \begin{pmatrix} \beta_1 \frac{1}{1-u_\beta-v_\beta} - \beta_2 \frac{1}{1+u_\beta-v_\beta} \\ -1 + \beta_1 \frac{1}{1-u_\beta-v_\beta} + \beta_2 \frac{1}{1+u_\beta-v_\beta} \end{pmatrix} = 0$. Adding both lines leads to $-1 + \beta_1 \frac{2}{1-u_\beta-v_\beta} = 0$ and is solved by $u_\beta = 1 - v_\beta - 2\beta_1$. Subtracting both lines leads to $1 - \beta_2 \frac{2}{1+u_\beta-v_\beta} = 1 - \beta_2 \frac{1}{1-v_\beta-\beta_1} = 0$ and is solved by $v_\beta = 1 - \beta_1 - \beta_2$, which again leads to $u_\beta = \beta_2 - \beta_1$. The inequality conditions translate to $\beta_1 > -\beta_1$ and $\beta_2 > -\beta_2$ and are obviously satisfied.

c) $(u_*, v_*)^\top = \lim_{\beta_1,\beta_2\to 0}(u_\beta, v_\beta)^\top = (0, 1)^\top$. Both constraints are active at $(0, 1)^\top$. $\mu_1 = \lim_{\beta_1\to 0} \frac{\beta_1}{2\beta_1} = \frac{1}{2}$ and $\mu_2 = \lim_{\beta_2\to 0} \frac{\beta_2}{2\beta_2} = \frac{1}{2}$.

$\square$

Solution of Home Exercise 7.2

a) $A(x; \alpha, \gamma) = u^2+v^2+\alpha(u+v+1)+\frac{1}{2}\gamma(u+v+1)^2$ and $\nabla A(x) = \begin{pmatrix} 2u + \alpha + \gamma(u + v + 1) \\ 2v + \alpha + \gamma(u + v + 1) \end{pmatrix}$

b) $x_1 = \frac{-1}{3}(1, 1)^\top$ solves $\nabla A(x; 0, 2) = \begin{pmatrix} 2u + 2(u + v + 1) \\ 2v + 2(u + v + 1) \end{pmatrix} = 0$. $h(x_1) = \frac{1}{3}$ and $\alpha_1 = \frac{2}{3}$.

c) $x_2 = \frac{-1}{9}(4, 4)^\top$ solves $\nabla A(x; \frac{2}{3}, 2) = \begin{pmatrix} 2u + \frac{2}{3} + 2(u + v + 1) \\ 2v + \frac{2}{3} + 2(u + v + 1) \end{pmatrix} = 0$. $h(x_2) = \frac{1}{9}$ and $\alpha_2 = \frac{8}{9}$.

$\square$

Solution of Home Exercise 8.1

a)

$$x_0 = \begin{pmatrix} h \\ 0 \end{pmatrix}, \quad x_1^R = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x_2^R = \begin{pmatrix} h \\ -h \end{pmatrix}$$

b) $f(x_0) = h^2 - 2h + 5$ and $f(x_1) = 4h^2 - 4h + 5$ and $f(x_2) = 2h^2 - 6h + 5$ and $f(x_1^R) = 5$ and $f(x_2^R) = 2h^2 + 2h + 5$.

$$\delta(f : \mathcal{S}) = \begin{pmatrix} 3h^2 - 2h \\ h^2 - 4h \end{pmatrix}, \qquad \delta(f : \mathcal{R}) = \begin{pmatrix} -h^2 + 2h \\ h^2 + 4h \end{pmatrix}$$

$$D_C(f:\mathcal{S}) = \begin{pmatrix} 2h-2 \\ -4 \end{pmatrix} \quad \text{and} \quad \nabla f_0 = \begin{pmatrix} 2h-2 \\ -4 \end{pmatrix} \quad \text{so} \quad ||\nabla f(x_0) - D_C(f:\mathcal{S})|| = 0$$

□

Solution of Home Exercise 8.2

a) For $x_0 = 0$ we get $x_1^* = 1$ with $f(x_1^*) = -18$ and $x_2^* = 2$ with $f(x_2^*) = -38$ and $x_3^* = -3$ with $f(x_3^*) = -18$.

b) For $x_* = 2$ we get $x_1^* = x_2^* = x_3^* = 2$ with $f(x_*) = -38$.

□

11 Additional Algorithms

Algorithm 11.1 (Conjugate Gradient Solver with Preconditioner)**:**
For solving $Ax = b$ with s.p.d. matrix A, requires incompleteCholesky() and LLTSolver() defined below.

1. *Input: $A \in \mathbb{R}^{n \times n}$; b; $\delta > 0$.*
2. *Set $L \leftarrow$ incompleteCholesky(A).*
3. *Set $x_j \leftarrow$ LLTSolver(L, b) (or otherwise given), $r_j \leftarrow Ax_j - b$ and $d_j \leftarrow -$LLTSolver(L, r_j).*
4. *While $||r_j|| > \delta$ do*
   a) *Set $\tilde{d}_j \leftarrow Ad_j$.*
   b) *Set $\rho_j \leftarrow {d_j}^\top \tilde{d}_j$.*
   c) *Set $t_j \leftarrow \frac{{r_j}^\top LLTSolver(L,r_j)}{\rho_j}$.*
   d) *Set $x_j \leftarrow x_j + t_j d_j$.*
   e) *Set $r_{old} \leftarrow r_j$.*
   f) *Set $r_j \leftarrow r_{old} + t_j \tilde{d}_j$.*
   g) *Set $\beta_j \leftarrow \frac{{r_j}^\top LLTSolver(L,r_j)}{{r_{old}}^\top LLTSolver(L,r_{old})}$*
   h) *Set $d_j \leftarrow -$LLTSolver$(L, r_j) + \beta_j d_j$.*
5. *Output: $x_* \leftarrow x_j$.*

Algorithm 11.2 (Incomplete Cholesky Decomposition)**:**
For cheap approximate decomposition of $A = LL^\top$ with s.p.d. matrix A. λ shifts the eigenvalues of the result into positive range, if set bigger than zero. δ allows to ignore elements, that are close to zero, and preserves sparsity. For $\lambda = 0$ and $\delta < 0$ this would be the complete Cholesky decomposition.

1. *Input: $A \in \mathbb{R}^{n \times n}$; $\lambda \geq 0$; $\delta \geq 0$.*
2. *For $k = 1 \dots n$ do*
   a) *Set $A_{k,k} \leftarrow \sqrt{\max(A_{k,k}, \lambda)}$.*
   b) *For $i = k + 1 \dots n$ do*
      i. *If $|A_{i,k}| > \delta$ set $A_{i,k} \leftarrow \frac{A_{i,k}}{A_{k,k}}$.*
      ii. *Else set $A_{i,k} \leftarrow 0$*
   c) *For $j = k + 1 \dots n$ do*
      i. *For $i = j \dots n$ do: If $|A_{i,j}| > \delta$ set $A_{i,j} \leftarrow A_{i,j} - A_{i,k}A_{j,k}$.*

3. *For $i = 1 \dots n$ do*
   a) *For $j = i+1 \dots n$ do: Set $A_{i,j} \leftarrow 0$.*
4. *Output: $L \leftarrow A$.*

Algorithm 11.3 (LLT-Solver)**:**
For computing $y = (LL^\top)^{-1} r$ with given lower triangle matrix L using forward and backward substitution.

1. *Input: $L \in \mathbb{R}^{n \times n}$; $r \in \mathbb{R}^n$.*
2. *For $i = 1 \dots n$ do*
   a) *Set $s_i \leftarrow r_i$*
   b) *For $j = 1 \dots i-1$ do*
      i. *Set $s_i \leftarrow s_i - L_{i,j} s_j$*
   c) *Set $s_i \leftarrow \frac{s_i}{L_{i,i}}$*
3. *For $i = n \dots 1$ do*
   a) *Set $y_i \leftarrow s_i$*
   b) *For $j = n \dots i+1$ do*
      i. *Set $y_i \leftarrow y_i - L_{j,i} y_j$*
   c) *Set $y_i \leftarrow \frac{y_i}{L_{i,i}}$*
4. *Output: y.*

Definition 11.4 (ε-Active Index Set)**:**
For numerical schemes using projection and matrix reduction, we define the more robust ε-active index set

$$\mathcal{A}^\varepsilon(x) := \{i \in \{1, \dots, n\} | \; x_i \leq a_i + \varepsilon \quad \text{or} \quad x_i \geq b_i - \varepsilon\} \tag{11.1}$$

for small $\varepsilon > 0$.

Algorithm 11.5 (Directional Hessian Approximation)**:**
Approximates $d_H \approx \nabla^2 f(x) d$ with central differences.

1. *Input: $f \in \mathcal{C}^1$; $x, d \in \mathbb{R}^n$; $\delta > 0$.*
2. *Set $d_H \leftarrow \frac{||d||}{2\delta}(\nabla f(x + \frac{\delta}{||d||} d) - \nabla f(x - \frac{\delta}{||d||} d))$.*
3. *Output: d_H.*

Algorithm 11.6 (Projected Approximation of Hessian times Direction)**:**
Approximates $\nabla^2_\Omega f(x)d$ for a given box constraint projection.

1. *Input:* $f \in \mathcal{C}^1$; $P : \mathbb{R}^n \to \Omega_\Box$; $x, d \in \mathbb{R}^n$; $\delta > 0$.
2. *Set* $x_p \leftarrow P(x)$ *and compute the active indexes* $A(x_p)$.
3. *Generate* d_r *with* $(d_r)_i \leftarrow (d)_i$ *for nonactive and* $(d_r)_i \leftarrow 0$ *for active indexes.*
4. *Set* $d_H \leftarrow \frac{||d_r||}{2\delta}(\nabla f(x + \frac{\delta}{||d_r||}d_r) - \nabla f(x - \frac{\delta}{||d_r||}d_r))$.
5. *Replace all active indexes* $(d_H)_i \leftarrow (d)_i$.
6. *Output:* d_H.

Algorithm 11.7 (Projected Inexact Newton-CG Descent)**:**
For solving nonlinear problems with exact gradient information and box constraints:

1. *Input:* $f \in \mathcal{C}^1$; $P : \mathbb{R}^n \to \Omega_\Box$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.
2. *Set* $x_k \leftarrow P(x_0)$, $\eta_k \leftarrow \min(\frac{1}{2}, \sqrt{||\nabla f_k||}) \cdot ||\nabla f(x_k)||$.
3. *While* $||x_k - P(x_k - \nabla f(x_k))|| > \varepsilon$ *do*
   a) *Set* $d_k \leftarrow -\nabla f(x_k)$.
   b) *Approximate* $d_H \leftarrow \nabla^2_\Omega f(x_k)d_k$ *with Alg. 11.6.*
   c) *Set* $\rho_k \leftarrow {d_k}^\top d_H$.
   d) *If* $\rho_k > \varepsilon||d_k||^2$ *do the first CG-step:*
      i. *Set* $r_j \leftarrow \nabla f(x_k)$ *and* $p_j \leftarrow -r_k$ *and* $x_j \leftarrow x_k$.
      ii. *Set* $d_A \leftarrow d_H$ *and set* $\rho_j \leftarrow \rho_k$.
      iii. *Set* $t_j \leftarrow \frac{||r_j||^2}{\rho_j}$ *and set* $x_j \leftarrow x_j + t_j d_j$.
      iv. *Set* $r_{old} \leftarrow r_j$ *and set* $r_j \leftarrow r_{old} + t_j d_A$.
      v. *Set* $\beta_j \leftarrow \frac{||r_j||^2}{||r_{old}||^2}$ *and set* $d_j \leftarrow -r_j + \beta_j d_j$.
      vi. *While* $||r_j|| > \eta_k$ *do additional CG steps:*
         A. *Approximate* $d_A \leftarrow \nabla^2_\Omega f(x_k)d_j$ *and set* $\rho_j \leftarrow {d_j}^\top d_A$.
         B. *Set* $t_j \leftarrow \frac{||r_j||^2}{\rho_j}$ *and set* $x_j \leftarrow x_j + t_j d_j$.
         C. *Set* $r_{old} \leftarrow r_j$ *and set* $r_j \leftarrow r_{old} + t_j d_A$.
         D. *Set* $\beta_j \leftarrow \frac{||r_j||^2}{||r_{old}||^2}$ *and set* $d_j \leftarrow -r_j + \beta_j d_j$.
      vii. *Set* $d_k \leftarrow x_j - x_k$.
   e) *Calculate a step size* $t_k > 0$ *for* f *at* x_k *in direction* d_k *with projected backtracking line search.*

   f) Set $x_k \leftarrow P(x_k + t_k d_k)$.

4. *Output:* $x_* \leftarrow x_k$.

Algorithm 11.8 (Projected Quasi-Newton with inverse BFGS-Update)**:**
For solving nonlinear programs using a matrix B_k *converging to the inverse Hessian*

1. *Input:* $f \in \mathcal{C}^1$; $P : \mathbb{R}^n \to \Omega_\square$; $x_0 \in \mathbb{R}^n$; $\varepsilon > 0$.
2. *Set* $x_k \leftarrow P(x_0)$, $B_k \leftarrow \mathbb{E}$.
3. *While* $||x_k - P(x_k - \nabla f(x_k))|| > \varepsilon$ *do*
   a) *Set* $d_k = -B_k \nabla f(x_k)$.
   b) *If* d_k *is not a descent direction, set* $d_k = -\nabla f(x_k)$ *and* $B_k \leftarrow \mathbb{E}$.
   c) *Calculate a step size* $t_k > 0$ *for* f *at* x_k *in direction* d_k *with projected backtracking line search.*
   d) *Set* $\Delta g_k \leftarrow \nabla f(x_k + t_k d_k) - \nabla f(x_k)$ *and* $\Delta x_k \leftarrow t_k d_k$.
   e) *Set* $x_k \leftarrow P(x_k + t_k d_k)$.
   f) *Update* B_k *according to the inverse (BFGS) update formula from Lemma 6.6.*
4. *Output:* $x_* \leftarrow x_k$.

Thank You For Your Attention!
Optimization for Engineers
– Summer Term 2022–