# Continuous Optimization

Christoph Ortner
ortner@maths.ox.ac.uk
Mathematical Institute, 24-29 St Giles', Oxford, OX1 3LB

March 1, 2009

# Contents

# 1 Introduction

## 1.1 Outline

## 1.2 Examples

***Example 1.1 (Data Fitting).*** An experiment is described by the nonlinear relation $b = R(a, x)$, where $R : \mathbb{R}^{K_1} \times \mathbb{R}^N \to \mathbb{R}^{K_2}$, $a \in \mathbb{R}^{K_1}$ describes the condition under which the experiment is conducted, and $b \in \mathbb{R}^{K_2}$ is the outcome of the experiment. The vector $x$ contains a set of parameters which are unknown and which need to be determined. Upon repeating the experiment $M$ times, we obtain $M$ data pairs $(a_i, b_i)_{i=1}^M \subset \mathbb{R}^{K_1} \times \mathbb{R}^{K_2}$. To estimate the parameter vector $x$ we minimize

$$\sum_{i=1}^{M} |R(a_i, x) - b_i|^2$$

with respect to the unknown $x \in \mathbb{R}^N$. This is an example of a *nonlinear least squares problem* (as opposed to a linear least squares problem where $x \mapsto R(a, x)$ is linear). □

***Example 1.2 (Optimal Luggage Size).*** An airline imposes size restrictions on the luggage passengers may take on board: Luggage must be rectangular, must not exceed 150cm in any spatial direction, and the surface occupied when it is placed on any side must not exceed 2000cm$^2$.

Let $x_1, x_2, x_3$ denote the height, length, and width of a piece of luggage. To maximise its volume under the stated constraints, we need to solve the following optimization problem:

$$\begin{aligned} \min_{(x_1,x_2,x_3)\in\mathbb{R}^3} \quad & -x_1x_2x_3 \\ \text{s.t.} \quad & x_1^2 + x_2^2 + x_3^2 \le 150^2 \\ & x_i \ge 0, \qquad i \in \{1,2,3\} \\ & x_ix_j \le 2000, \qquad i \ne j \in \{1,2,3\}. \end{aligned}$$

□

***Example 1.3 (Finite Deformation Elasticity).*** (a) The energy of an elastic body with reference configuration $\Omega \subset \mathbb{R}^3$ and deformation $y \in \mathrm{C}^1(\Omega; \mathbb{R}^3)$ is described by

$$E(y) = \int_\Omega W(\nabla y(x))\,\mathrm{d}x$$

On a portion $\Gamma$ of the boundary $\partial\Omega$, a discplacement $y_0$ is applied. To find the steady state of the body, we are required to minimize $E(y)$ over all deformations $y$ which satisfy $y = y_0$ on $\Gamma$.

(b) Suppose the elastic body is lying on a rigid, flat surface spanning the plane $\{x_3 = 0\}$. Under the action of gravity, the total energy of the body becomes $I(y) = E(y) - g\int_\Omega y_3(x)\,\mathrm{d}x$. Thus, to find the stead state, we need to minimize $I(y)$ over all deformations $y$ satisfying $y_3(x) \ge 0$ for all $x \in \Omega$. □

**Further Examples:**

- Optimal control
- Optimal design (e.g., material design, shape optimization: "Find the shape of a bridge capable of sustaining given amount of traffic using the smallest amount of material", ...)
- Expenditure minimisation
- .....

### 1.3 The general optimization problem

Let $f : \mathbb{R}^N \to \mathbb{R}$ be the objective function. Furthermore, we assume throughout that the admissible (or feasible) set $\Omega$ is given by

$$\Omega = \big\{x \in \mathbb{R}^N : c_j(x) = 0, j \in \mathcal{E}, c_j(x) \geq 0, j \in \mathcal{I}\big\},$$

where $c : \mathbb{R}^N \to \mathbb{R}^{M_e+M_i}$, $\mathcal{E} = \{1, \ldots, M_e\}$, and $\mathcal{I} = \{M_e + 1, \ldots, M_e + M_i\}$. A point $x \in \Omega$ is called *feasible* or *admissible*.

A *global minimizer* of $f$ in $\Omega$ (or simply, a global minimizer) is a point $x_* \in \Omega$ such that

$$f(x_*) \leq f(x) \qquad \forall x \in \Omega. \tag{1}$$

A point $x_* \in \Omega$ in a *local minimizer* of $f$ is $\Omega$ (or simply, a local minimizer) if there exists $r > 0$ such that

$$f(x_*) \leq f(x) \qquad \forall x \in \Omega \cap B_r(x_*). \tag{2}$$

A point $x_* \in \Omega$ is a *strict local minimizer* of $f$ in $\Omega$ (or simply, a strict local minimizer) if there exists $r > 0$ such that

$$f(x_*) < f(x) \qquad \forall x \in (\Omega \cap B_r(x_*)) \setminus \{x_*\}. \tag{3}$$

We will typically seek local minimizers since, for non-convex optimisation problems, it is unrealistic to expect that one can find a global minimiser.

## 2 Preliminaries

### 2.1 Linear algebra

Elements of the vector space $\mathbb{R}^N$ are usually denoted $x, y, z$ with components $x = (x^j)_{j=1}^N$. The space is equipped with the Euclidean inner product (the 'dot-product')

$$x \cdot y = x^T y = \sum_{j=1}^N x^j y^j,$$

and with the Euclidean norm

$$|x| = \Big(\sum_{j=1}^N |x^j|^2\Big)^{1/2} = (x \cdot x)^{1/2}.$$

If we want to be particularly careful to distinguish this norm from other norms, we will write $|\cdot| = |\cdot|_2$.

Two of the most important inequalities for Euclidean spaces are the Cauchy Inequality

$$x \cdot y \leq \tfrac{1}{2}|x|^2 + \tfrac{1}{2}|y|^2, \tag{4}$$

and the Cauchy–Schwarz Inequality

$$x \cdot y \leq |x||y|, \tag{5}$$

with equality if, and only if, $x$ is a multiple of $y$.

***Problem 2.1.***

(a) Prove Cauchy's Inequality (4). *Hint: First prove it for $N = 1$ then generalize.*

(b) Use Cauchy's Inequality to derive the Cauchy-Schwarz Inequality (5). *Hint: First prove it for $|x| = |y| = 1$, then generalize.* □

***Problem 2.2.*** A map $|\cdot| : \mathbb{R}^N \to [0, +\infty)$ is called a norm if

$$\begin{aligned} &|x| = 0 \Leftrightarrow x = 0, \\ &|\lambda x| = |\lambda||x| \qquad \forall \lambda \in \mathbb{R}, x \in \mathbb{R}^N, \quad \text{and} \\ &|x + y| \leq |x| + |y| \qquad \forall x, y \in \mathbb{R}^N. \end{aligned}$$

It is trivial to see that the Euclidean 2-norm $|\cdot| = |\cdot|_2$ satisfies the first two conditions. Use the Cauchy–Schwarz Inequality to prove the third condition (also called the triangle inequality of the Minkowski Inequality). □

Open and closed balls in $\mathbb{R}^N$ are denoted

$$B_r(x) = \{x' \in \mathbb{R}^N : |x - x'| < r\} \quad \text{and} \quad \bar{B}_r(x) = \{x' \in \mathbb{R}^N : |x - x'| \leq r\}.$$

A linear mapping $L : \mathbb{R}^N \to \mathbb{R}^M$ can always be represented by a matrix-vector operation $L(x) = Ax$, where $A \in \mathbb{R}^{M \times N}$. We shall therefore never distinguish the two points of view. The operator-norm of a matrix $A \in \mathbb{R}^{M \times N}$ is given by

$$\|A\| = \max_{\substack{x \in \mathbb{R}^N \\ |x|=1}} |Ax|.$$

***Problem 2.3.*** Show that the operator norm $\|\cdot\|$ is a norm on the space $\mathbb{R}^{M \times N}$ of matrices. Further, given $A \in \mathbb{R}^{M \times N}$, prove that $\|A\|$ is the smallest constant $C \geq 0$ such that $|Ax| \leq C|x|$ holds for all $x \in \mathbb{R}^N$. □

A matrix $A \in \mathbb{R}^{N \times N}$ is called *invertible* if the map $x \mapsto Ax$ is 1-1 and onto, and is otherwise called *singular*. The set of invertible matrices is denoted

$$\mathrm{Iso}_N = \{A \in \mathbb{R}^{N \times N} : A \text{ is invertible}\}.$$

The following Lemma shows that the $\mathrm{Iso}_N$ is open.

**Lemma 2.1.** *Let $T, S \in \mathbb{R}^{N \times N}$ and $T$ is invertible. If $\|T - S\| \leq 1/\|T^{-1}\|$ then $S$ is invertible, $S^{-1} = \sum_{n=0}^{\infty} [T^{-1}(T - S)]^n T^{-1}$, and in particular, $\|S^{-1}\| \leq \|T^{-1}\|/(1 - \|T^{-1}\|\|T - S\|)$.*

*Proof.* Let $A \in \mathbb{R}^{N\times N}$ with $\|A\| < 1$, and set $R_k = \sum_{n=0}^{k} A^n$, then $R_k$ converges to some matrix $R$. Furthermore, since $R_k(I-A) = I - A^{k+1}$ it follows that $R(I-A) = I$, i.e., $R = (I-A)^{-1}$.

Upon noting that $\|T^{-1}(T-S)\| \le \|T^{-1}\|\|T-S\| < 1$, we can set $A = T^{-1}(T-S)$ to obtain that $T^{-1}S = I - T^{-1}(T-S)$ is invertible and that $(T^{-1}S)^{-1} = \sum_{n=0}^{\infty}[T^{-1}(T-S)]^n$, from which we immediately deduce the result. □

The *condition number* of an invertible matrix $A$ is denoted

$$\kappa(A) = \|A\|\|A^{-1}\|.$$

If $A$ is not invertible then $\kappa(A) = +\infty$.

***Problem 2.4.*** Suppose that $A$ is invertible, and that $Ax = b$ and $A\tilde{x} = \tilde{b}$ (here, $\tilde{b}$ is the erroneous data, and $\tilde{x}$ the erroneous solution). Prove that the relative error satisfies

$$\frac{|x-\tilde{x}|}{|x|+|\tilde{x}|} \le \kappa(A)\frac{|b-\tilde{b}|}{|b|+|\tilde{b}|}.$$ □

A matrix $A \in \mathbb{R}^{N\times N}$ is called *positive definite* (in short, $A > 0$) if $x^TAx > 0$ for all $x \neq 0$. It is called *positive semi-definite* (in short, $A \ge 0$) if $x^TAx \ge 0$ for all $x \in \mathbb{R}^N$. The set of positive (semi-)definite matrices are denoted

$$\mathbb{R}^{N\times N}_{>} = \{A \in \mathbb{R}^{N\times N} : A > 0\} \quad \text{and} \quad \mathbb{R}^{N\times N}_{\ge} = \{A \in \mathbb{R}^{N\times N} : A \ge 0\}.$$

Similarly, we define the terms negative (semi-)definite and the respective sets. If a matrix $A$ is neither positive nor negative semi-definite, we call it *indefinite*. If a matrix is symmetric and positive definite, we say it is *spd*.

**Proposition 2.2.** *If $A \in \mathbb{R}^{N\times N}$ is symmetric then there exist eigenvalues $\lambda_1 \le \cdots \le \lambda_N \in \mathbb{R}$ and eigenvectors $v_1, \ldots, v_N$ such that*

$$Av_n = \lambda_n v_n, \qquad n = 1, \ldots, N.$$

*The set $\{v_n; n = 1, \ldots N\}$ is an orthonormal basis of $\mathbb{R}^N$. We call $\sigma(A) := \{\lambda_1, \ldots, \lambda_N\}$ the* spectrum *of $A$.*

*Moreover, $A$ has the representation $A = QDQ^T$ where $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ and $Q = (v_1|\ldots|v_N)$. This representation is unique up to a perturbation of the eigenvalues (or of the columns of $Q$).*

*Proof.* see lecture courses on linear algebra. □

***Problem 2.5 (Eigenvalues).***

(a) Prove that a symmetric matrix $A \in \mathbb{R}^{N\times N}$ with spectrum $\sigma(A) = \{\lambda_1, \ldots, \lambda_N\}$ is invertible if, and only if, $0 \notin \sigma(A)$. *Hint: What are the eigenvalues and eigenvectors of $A^{-1}$?*

(b) Show that $\|A\| = \max_{n=1,\dots,N} |\lambda_n|$, and that $\|A^{-1}\| = \frac{1}{\min_{n=1,\dots,N} |\lambda_n|}$. In particular, $\kappa(A) = \max_n |\lambda_n| / \min_n |\lambda_n|$

(c) Suppose $A \in \mathbb{R}^{N\times N}$ is spd, prove that $A^{-1}$ is spd.

(d) Prove that, for any symmetric and positive semi-definite matrix $A$, there exists a unique spd matrix $\sqrt{A} = A^{1/2}$ such that $(A^{1/2})^2 = A$. □

Apart from the standard Euclidean norm $|\cdot|$ we will occasionally use slightly more general Euclidean norms of the form

$$|x|_B = (x^T B x)^{1/2},$$

where $B \in \mathbb{R}^{N\times N}$ is a symmetric and positive definite matrix. The operator norm of a matrix $A \in \mathbb{R}^{N\times N}$ with respect to the $B$-norm is defined by

$$\|A\|_B = \sup_{\substack{x\in\mathbb{R}^N \\ |x|_B=1}} |Ax|_B.$$

In the following exercises we will establish some basic properties of these $B$-norms.

***Problem 2.6.*** Let $A, B \in \mathbb{R}^{N\times N}$ where $A$ is symmetric, and $B$ is spd.

(a) Prove that $|\cdot|_B$ is a norm.

(b) Compute the operator norm of $A$ with respect to the $|\cdot|_B$-norm in terms of the standard operator norm of a related matrix. □

## 2.2 Multi-variable calculus

Let $U \subset \mathbb{R}^N$ be an open set and let $f : U \to \mathbb{R}^M$. We say that $f$ is continuous at $x \in U$ if $f(x_j) \to f(x)$ whenever $x_j \to x$. We say that $f$ is continuous in $U$ (or $f \in \mathrm{C}(U)$) if $f$ is continuous in each point $x \in U$ (and similarly for all definitions which follow).

**Definition 2.3 (Derivatives).** *$f$ is (Fréchet-) differentiable at $x$ if there exists a matrix $A \in \mathbb{R}^{M\times N}$ such that*

$$\lim_{h\to 0} \frac{|f(x+h) - f(x) - Ah|}{|h|} = 0. \tag{6}$$

*We call $Df(x) := A$ the* (Fréchet-)derivative *of $f$ at $x$ and $\nabla f(x) := Df(x)^T$ the* gradient *of $f$ at $x$. We say that $f$ is continuously differentiable in $U$ (or $f \in \mathrm{C}^1(U)$) if $Df : U \to \mathbb{R}^{M\times N}$ is continous.*

*If $f : U \to \mathbb{R}$, i.e. $M = 1$, we say $f$ is twice differentiable at $x$ if $f$ is differentiable at $x$ and if there exists $H \in \mathbb{R}^{N\times N}$ such that*

$$\lim_{h\to 0} \frac{|f(x+h) - f(x) - Df(x)h - \frac{1}{2}h^T H h|}{|h|^2} = 0,$$

*and we call $D^2 f(x) = \nabla^2 f(x) = H$ the* Hessian *of $f$ at $x$. We say that $f \in \mathrm{C}^2(U)$ if $D^2 f \in \mathrm{C}(U)$.* □

It is most convenient and intuitive to think of the derivative as a linear approximation to $f$ in a neighbourhood, i.e.,

$$f(x+h) = f(x) + Df(x)h + o(|h|),$$

and of the second derivative as a quadratic approximation,

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

Here, and throughout, the little-$o$-notation is used to denote a generic function $o : B_\varepsilon(0) \to \mathbb{R}^M$ (where $M$ should be obvious from the context) which satisfies $\lim_{|x|\searrow 0} o(|x|)/|x| = 0$. Similarly, the big-$O$-notation is used to denote a generic funciton $O : B_\varepsilon(0) \to \mathbb{R}^M$ which satisfies $\limsup_{|x|\searrow 0} O(x)/|x| < +\infty$.

In practise, derivatives are represented by partial derivatives. For example, let $f \in \mathrm{C}^1(\mathbb{R}^N)$, then $\nabla f(x) \cdot e_i = \partial f(x)/\partial x^{(i)}$, and hence,

$$\nabla f(x) = \Big(\frac{\partial f(x)}{\partial x^{(1)}}, \dots, \frac{\partial f(x)}{\partial x^{(N)}}\Big)^T.$$

***Problem 2.7.*** If $F \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R}^M)$, represent $DF(x)$ in terms of the partial derivatives $\partial F^{(i)}/\partial x^{(j)}$. Hence, given $f \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R})$, find a form for the Hessian $\nabla^2 f(x)$ in terms of second partial derivatives of $f$ and deduce that it is symmetric. □

***Problem 2.8.*** Let $U$ be an open convex set in $\mathbb{R}^N$. Prove the following Taylor formulas:

(a) If $f \in \mathrm{C}^1(U; \mathbb{R}^M)$ then

$$f(x+h) = f(x) + \nabla f(x) \cdot h + \int_0^1 (\nabla f(x+th) - \nabla f(x)) \cdot h \,\mathrm{d}t.$$

(b) If $f \in \mathrm{C}^2(U; \mathbb{R})$ then

$$\begin{aligned} f(x+h) = \; & f(x) + \nabla f(x) \cdot h + \tfrac{1}{2}h^T \nabla^2 f(x) h \\ & + h^T \Big[ \int_0^1 (1-t)(\nabla^2 f(x+th) - \nabla^2 f(x)) \,\mathrm{d}t \Big] h. \end{aligned}$$

Hint: define $\varphi(t) = f(x+th)$ and use (or prove) the one-dimensional analogues. □

**Definition 2.4 (Lipschitz-Continuity).** *We say that $f : U \to \mathbb{R}^M$, where $U$ is open or closed, is* Lipschitz continuous *in $U$ if there exists $L \geq 0$ such that*

$$|f(x) - f(x')| \leq L|x - x'| \qquad \forall x, x' \in U,$$

*and we let $\mathrm{Lip}_U(f)$ be the smallest $L$ for which this holds (the* Lipschitz constant *of $f$ in $U$). If $U = \mathbb{R}^N$ then we omit the subscript $U$.*

*We say that $f$ is* locally Lipschitz continuous *in an open set $U$ if $f$ is Lipschitz continuous in every* closed *subset of $U$.*

*If $f : U \to \mathbb{R}^{M\times N}$ then we replace the Euclidean norm with the operator norm in the definition of Lipschitz continuity.*

## 2.3 The Implicit Function Theorem

The Implicit Function Theorem is one of the most important tools in nonlinear analysis. We will require it for characterizing the geometry of admissible sets in constrained optimization. For a proof of the Implicit Function Theorem see any multi-variable textbook.

**Theorem 2.5.** *Let $F : \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N$ be continuously (Frechet-) differentiable in a neighbourhood of a point $(x_0, y_0) \in \mathbb{R}^N \times \mathbb{R}^M$ and assume that $F(x_0, y_0) = 0$ and that $D_x F(x_0, y_0)$ is invertible (here, $D_x$ denotes the derivative of $x \mapsto F(x, y)$ with $y$ fixed). Then there exist open sets $U \subset \mathbb{R}^N, V \subset \mathbb{R}^M$, and $g \in \mathrm{C}^1(V; U)$ such that*

$$\big\{(x, y) \in U \times V : F(x, y) = 0\big\} = \{(g(y), y) : y \in V\}.$$

## 2.4 Optimality conditions for unconstrained optimization

Condition (2) for $x_*$ to be a local minimizer is intuitive, but difficult to verify in practise. Instead we will use so-called necessary and sufficient optimality conditions.

**Proposition 2.6 (Necessary Optimality Conditions).** *Suppose that $f : \mathbb{R}^N \to \mathbb{R}$ and that $x_*$ is a local minimizer of $f$ in $\mathbb{R}^N$.*

*(a) If $f$ is differentiable at $x_*$ then $\nabla f(x) = 0$.*

*(b) If $f$ is twice differentiable at $x_*$ then $\nabla^2 f(x) \geq 0$.*

***Problem 2.9.*** Prove Proposition 2.6. □

**Definition 2.7 (Critical Points).** *A point $x_* \in \mathbb{R}^N$ satisfying $\nabla f(x_*) = 0$ is called a* first-order critical point *(or simply,* critical point*). If, in addition, $\nabla^2 f(x_*) \geq 0$ we call $x_*$ a* second-order critical point.

**Proposition 2.8 (Sufficient Optimality Conditions).** *Suppose that $f : \mathbb{R}^N \to \mathbb{R}$ is twice differentiable at $x_*$, that $\nabla f(x_*) = 0$ and that $\nabla f(x_*) > 0$, then $x_*$ is a strict local minimizer of $f$.*

***Problem 2.10.*** Prove Proposition 2.8. □

## 2.5 Convergence rates

In continuous optimization, we construct sequences $(x_n)_{n\in\mathbb{N}}$ converging to some limit $x_*$, typically the solution of a minimization problem. The speed of convergence has immediate impact on the complexity of the method. To measure the speed of convergence of sequences, we introduce the notion of *convergence rate*.

**Definition 2.9 (Convergence Rates).** *Let $(x_n) \subset \mathbb{R}^N$ and $x_* \in \mathbb{R}^N$.*

*(i) We say that $x_n \to x_*$ with q-order $\alpha > 1$ if there exists $K \geq 0$ such that $|x_{n+1} - x_*| \leq K|x_n - x_*|^\alpha$. If $\alpha = 2$, we say that $x_n \to x_*$ q-quadratically.*

*(ii) We say that $x_n \to x_*$ q-superlinearly if $|x_{n+1} - x_*|/|x_n - x_*| \to 0$ as $n \to \infty$.*

*(iii) We say that $x_n \to x_*$ q-linearly with q-factor $\sigma \in (0,1)$ if $|x_{n+1} - x_*| \leq \sigma |x_n - x_*|$.*

Sometimes, the notion of q-order is too strong, and it is more convenient to relax it. We say that $x_n \to x_*$ with r-order $\alpha > 1$ if there exists $(\xi_n)_{n\in\mathbb{N}}$ such that $|x_n - x_*| \leq \xi_n$ and $\xi_n \to 0$ with q-order $\alpha$. Similarly, we say that $x_n \to x_*$ r-linearly (or r-superlinearly) if there exists $(\xi_n)_{n\in\mathbb{N}}$ such that $\xi_n \to 0$ q-linearly (or q-superlinearly).

***Example 2.10.*** Let $\rho \in (0,1)$.

(a) The sequence $(\rho, \rho^2, \rho^3, \dots)$ converges to zero q-linearly with q-factor $\rho$.

(b) The sequence $(\rho, \rho, \rho^2, \rho^2, \rho^3, \rho^3, \dots)$ converges to zero r-linearly.

(c) The sequence $(\rho, \rho^2, \rho^4, \rho^8, \rho^{16}, \dots)$ converges to zero q-quadratically. □

***Example 2.11.*** Suppose that $x_n \to x_*$. We assume that $|x_0 - x_*| = 1/2$ and we terminate the sequence we have achieved $|x_n - x_*| \leq 10^{-10}$. If $x_n \to x_*$ q-linearly with q-factor $\sigma = 1/2$ (which is quite fast), then it would take 33 iterations to achieve the desired accuracy. If $x_n \to x_*$ q-quadratically with constant $C = 1$, then 6 iterations suffice. □

# 3 Newton's Method

When solving an optimization problem analytically, one usually derives the first-order criticality condition $\nabla f(x) = 0$, finds all its solutions, and then discards those which are not (local) minimizers. Our first optimization method will mimic this idea. Since we are now solving a nonlinear system, we will disregard the objective function and assume that we are given $F : \mathbb{R}^N \to \mathbb{R}^N$ and wish to solve $F(x) = 0$. (For an optimisation problem, $F(x) = \nabla f(x)$.)

Solving nonlinear systems directly is in general impossible. However, using the interpretation of the derivative as a linear approximation, we can replace the nonlinear system $F(x) = 0$ by a linear system. Suppose, for example, that we have an initial guess $x$ for a root $x_*$. If $F$ is continuously differentiable in a neighbourhood containing both $x$ and $x_*$, then

$$0 = F(x_*) = F(x) + DF(x)(x_* - x) + o(x_* - x).$$

In particular, if we solve

$$h = x - DF(x)^{-1}F(x),$$

then we may expect that $x + h$ is closer to $x_*$, and we can iterate the idea to obtain a sequence which hopefully converges to $x_*$.

**Algorithm 3.1 (Newton's Method).**
**Input:** $x_0 \in \mathbb{R}^N$
1: **for** $k = 0, 1, 2, \dots$ **do**
2: $\quad x_{k+1} = x_k - DF(x_k)^{-1}F(x_k)$
3: **end for**

Our two basic assumption in the motivation of this algorithm are (i) that $DF(x_k)$ is invertible for all $k$, and (ii) that we are starting sufficiently close to a root $x_*$. In the following lemma we make these assumptions precise. Since invertibility of $DF$ is so important in Newton's method we call points $x \in \mathbb{R}^N$ where $F$ is differentiable and where $DF(x)$ is invertible *regular points*, and *singular points* if $F$ is differentiable but $DF(x)$ is singular.

**Lemma 3.1.** *Let $x_* \in \mathbb{R}^N$ and $R > 0$ such that $F$ is continuously differentiable in a neighbourhood of $x_*$ with locally Lipschitz continuous derivative $DF$. If, furthermore, $DF(x_*)$ is invertible then there exists $r > 0$ such that $DF(x) \in \mathrm{Iso}_N$ and $\|DF(x)^{-1}\| \leq 2\|DF(x_*)^{-1}\|$ for all $x \in \bar{B}_r(x_*)$.*

*Proof.* Let $R > 0$ such that $F$ is differentiable in $\bar{B}_R(x_*)$ and such that $DF$ is Lipschitz continuous in $\bar{B}_R(x_*)$. Let $\sigma := \|DF(x_*)^{-1}\|$ and $L := \mathrm{Lip}_{\bar{B}_{R(x_*)}}(DF)$. According to Lemma 2.1, all matrices $S$ satisfying $\|DF(x_*) - S\| < 1/\sigma$ are invertible with $\|S^{-1}\| \leq \sigma/(1 - \sigma\|DF(x_*) - S\|)$.

For $x \in B_r(x_*)$, $r \leq R$, we have $\|DF(x_*) - DF(x)\| \leq Lr$. Thus, for $r = \min(R, 1/(2L\sigma))$, $x \in \bar{B}_r(x_*)$, we have $DF(x) \in \mathrm{Iso}_N$ and

$$\|DF(x)^{-1}\| \leq \frac{\sigma}{1 - \sigma Lr} \leq \frac{\sigma}{1 - 1/2} = 2\sigma. \qquad \square$$

From Lemma 3.1 it follows that, in the neighbourhood of a root $x_*$ with $DF(x_*) \in \mathrm{Iso}_N$, Newton's method is well-defined. We shall now verify that, if the initial guess is sufficiently close, then the iterates produced by Newton's Method do not leave this neighbourhood and in fact converge to the root $x_*$.

**Theorem 3.2.** *Suppose $U \subset \mathbb{R}^N$ is open, $F \in \mathrm{C}^1(U; \mathbb{R}^N)$ is continuously differentiable and $DF$ is locally Lipschitz continuous in $U$. Suppose, further, that $x_* \in U$, $F(x_*) = 0$, and that $DF(x_*)$ is invertible. Then there exists $R > 0$ such that, for $x_0 \in \bar{B}_R(x_*)$, Newton's Method with starting value $x_0$ is well-defined and converges q-quadratically to $x_*$.*

*Proof.* In view of Lemma 3.1 there exists $r > 0$ such that $DF$ is Lipschitz continuous in $\bar{B}_r(x_*)$ with constant $L$, and $DF(x)$ is invertible with $\|DF(x)^{-1}\| \leq 2\sigma := 2\|DF(x_*)^{-1}\|$ for $x \in \bar{B}_r(x_*)$.

Suppose that $x_k \in \bar{B}_r(x_*)$, then, by the definition of Newton's method, we have

$$DF(x_k)(x_{k+1} - x_*) = DF(x_k)(x_k - x_*) - F(x_k) = DF(x_k)(x_k - x_*) - (F(x_k) - F(x_*)).$$

Since $F$ is differentiable, we can expand

$$F(x_k) - F(x_*) = \int_0^1 \frac{\mathrm{d}}{\mathrm{d}t} F(x_* + t(x_k - x_*))\,\mathrm{d}t = \int_0^1 DF(x_* + t(x_k - x_*))\,\mathrm{d}t\,(x_k - x_*),$$

which gives

$$DF(x_k)(x_{k+1} - x_*) = \int_0^1 \big(DF(x_k) - DF(x_* + t(x_k - x_*))\big)\,\mathrm{d}t\,(x_k - x_*).$$

Multiplying by $DF(x_k)^{-1}$ and taking norms, we obtain

$$|x_{k+1} - x_*| \leq \tfrac{1}{2} L\sigma |x_k - x_*|^2. \tag{7}$$

In particular, if $r \leq 1/(2L\sigma)$ (compare this with the proof of Lemma 3.1) then $|x_{k+1} - x_*| \leq \frac{1}{2}|x_k - x_*|$. This proves that the sequence $(x_k)$ remains inside $\bar{B}_r(x_*)$ and that $x_k \to x_*$ as $k \to \infty$. q-quadratic convergence is established by estimate (7). □

***Problem 3.1 (Linear Convergence to Singular Points).*** Suppose that the sequence $(x_n)$ is generated by Newton's Method, and converges to a singular point $x_*$. Then we would *typically* expect that $x_n \to x_*$ q-linearly. To demonstrate this, let $F(x) = x^k$, $k \geq 2$, and prove that, independent of the starting point $x_0$, Newton's Method converges q-linearly to zero and compute the q-factor. □

***Problem 3.2 (Failure of Global Convergence).*** Give an example of a strictly monotone real function $F : \mathbb{R} \to \mathbb{R}$ with $F(0) = 0$, and of a starting point $x_0$ so that Newton's Method does not converge. □

It remains to describe suitable termination criteria for Newton's method. Since termination is generally more an art than a science, we shall only give examples and not go into too much detail.

1. The most common termination criterion is step length. Note that, in the neighbourhood of a (regular) root, $|x_{n+1} - x_*| = O(|x_n - x_*|^2)$, and hence $|x_{n+1} - x_n| = |x_n - x_*| + O(|x_n - x_*|^2)$. Therefore, we could terminate Newton's method as soon as $|x_{n+1} - x_n|$ falls below a certain tolerance.

   The obvious problem with this criterion is that it may prematurely terminate the iteration if $x_n$ is not close to a regular root, or if the root is singular.

2. Another common termination criterion is the norm of the residual $|F(x_n)|$. We could terminate Newton's method as soon as $|F(x_n)|$ falls below a prescribed tolerance.

   The obvious problem with this approach is again premature termination. For example, if $F(x) = e^x$, then the algorithm will terminate without recognising that it is not at all converging to a root (as there exists none).

3. In practise, one usually uses a combination of 1. and 2..

# 4 Line Search Methods

Although, in terms of its local convergence rate, Newton's method leaves nothing to wish for, it falls short of the ideal in at least two respects:

- it may converge to local maxima or saddle points.
- it may converge slowly or not at all if the starting guess is not good.

A possible solution for both of these issues is to formulate algorithms which ensure that the objective function *decays in each iteration*. There are two classes of algorithms which we shall cover: line search methods and trust region methods. In line search methods (present section) a descent direction is computed in each step, following by a one-dimensional search along this direction to compute a new iterate with lower energy. Trust region methods build quadratic models of the target functional and minimize these models in a small neighbourhood of the current iterate (the trust region) to obtain the next iterate.

### 4.1 The basic steepest descent algorithm

Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ and $x \in \mathbb{R}^N$. A direction $s \in \mathbb{R}^N$ is a *descent direction* for $f$ at $x$, if

$$\nabla f(x) \cdot s = \lim_{t \searrow 0} \frac{f(x+ts) - f(x)}{t} < 0.$$

The direction of steepest descent is obtained by minimizing the slope $\nabla f(x) \cdot s$ over all $s$ with fixed length, e.g., $|s| = 1$, i.e., we wish to find $\hat{s} \in \mathbb{R}^N$, $|\hat{s}| = 1$ such that

$$\nabla f(x) \cdot \hat{s} \leq \nabla f(x) \cdot s \qquad \forall s \in \mathbb{R}^N, |s| = 1. \tag{8}$$

**Proposition 4.1.** *The direction of steepest descent of $f$ at $x$ is $\hat{s} = -\nabla f(x)/|\nabla f(x)|$.*

*Proof.* Let $|s| = 1$ then, using the Cauchy–Schwarz Inequality (5),

$$\nabla f(x) \cdot \hat{s} = \ -|\nabla f(x)| = -|\nabla f(x)||s| \leq -\nabla f(x) \cdot s. \qquad \square$$

Proposition 4.1 motivates the choice $s = -\nabla f(x)/|\nabla f(x)|$, or simply, $s = -\nabla f(x)$ as search direction for the current iterate $x$. Clearly, if we take a small step in this direction, then the energy will decrease strictly. More precisely, if $f$ is differentiable at $x$ and if $s = -\nabla f(x)$, then

$$f(x + \alpha s) = f(x) - \alpha |\nabla f(x)|^2 + o(\alpha). \tag{9}$$

Hence, for $\alpha$ sufficiently small, we have $f(x + \alpha s) < f(x)$.

Monotonicity of the objective function is not sufficient in theory, hence we shall impose the slightly stronger *sufficient descent condition* (or Armijo condition)

$$f(x + \alpha s) \leq f(x) + \theta_{sd} \alpha \nabla f(x) \cdot s, \tag{10}$$

where $\theta_{sd} \in (0,1)$ is a user-defined parameter. In practise it is usually taken very small, for example, $\theta_{sd} = 10^{-3}$.

However, we are not allowed to take arbitrarily small steps as the iteration might stagnate in that case. The following *backtracking line search* method takes small steps only if required to satisfy (10). We formulate it slightly more generally then we need at the moment in order to be able to refer to it later on.

**Algorithm 4.1. LINESEARCH**
**Input:** $x, s \in \mathbb{R}^N$ such that $\nabla f(x) \cdot s < 0$, $\theta_{sd} \in (0,1)$;
**Output:** $\alpha > 0$ s.t. (10) is satisfied
1: $\alpha \leftarrow 1$;
2: **while** $f(x + \alpha s) > f(x) + \theta_{sd} \alpha \nabla f(x) \cdot s$ **do**
3: $\quad \alpha \leftarrow \alpha/2$;
4: **end while**
5: **return** $\alpha$;

Generalizing (9) to

$$f(x + \alpha s) = f(x) + \alpha \nabla f(x) \cdot s + o(\alpha),$$

it follows immediately that, for sufficiently small $\alpha$, the termination condition is satisfied, and thus Algorithm 4.1 terminates and returns the desired output.

There are many different linesearch algorithms which are, in many ways, more sophisticated than the Backtracking-Armijo Algorithm; see [4, Sections 3.1, 3.5] for further detail, in particular, the *Wolfe Conditions* in Sec. 3.1 and Algorithm 3.5 in Sec. 3.5.

**Algorithm 4.2 (A Basic Steepest Descent Method).**
**Input:** $x_0 \in \mathbb{R}^N$, $\theta_{\rm sd} \in (0,1)$
1: **for** $n = 0, 1, 2, \ldots$ **do**
2: $\quad \alpha_n \leftarrow \text{LINESEARCH}[x = x_n,\, s = -\nabla f(x_n),\, \theta_{sd}]$;
3: $\quad x_{n+1} = x_n - \alpha_n \nabla f(x_n)$;
4: **end for**

**Theorem 4.2 (Global Convergence of Steepest Descent).** *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ is bounded below and that $\nabla f$ is globally Lipschitz continuous. Then Algorithm 4.2 is* globally convergent*: For any $\theta_{sd} \in (0,1)$, and for any starting value $x_0 \in \mathbb{R}^N$,*

$$\sum_{n=0}^{\infty} |\nabla f(x_n)|^2 < +\infty. \tag{11}$$

*Proof.* Since $f(x_n)$ is monotonically decreasing and bounded below, there exists $f_* \in \mathbb{R}$ such that $f(x_n) \to f_*$. Let $L = \mathrm{Lip}(f)$.

From (10), which is guaranteed by the backtracking line-search, we obtain

$$f(x_{n+1}) \le f(x_n) - \theta_{sd}\alpha_n |\nabla f(x_n)|^2,$$

from which we immediately deduce

$$\theta_{\rm sd} \sum_{n=0}^{\infty} \alpha_n |\nabla f(x_n)|^2 \le f(x_0) - f_*.$$

It remains to prove that the $\alpha_n$ are bounded below. For any $\alpha \in (0,1]$, and $s_n = -\nabla f(x_n)$, we have

$$\begin{aligned} f(x_n + \alpha s_n) = &\ f(x_n) + \alpha \nabla f(x_n)\cdot s_n + \alpha \int_0^1 \big(\nabla f(x_n + t\alpha s_n) - \nabla f(x_n)\big)\cdot s_n \,\mathrm{d}t \\ \le &\ f(x_n) - \alpha |\nabla f(x_n)|^2 + \tfrac{1}{2} L \alpha^2 |\nabla f(x_n)|^2. \end{aligned} \tag{12}$$

In particular, if $\alpha \le 2(1-\theta)/L$ then the sufficient decrease condition (10) is satisfied and the linesearch terminates. Since $\alpha$ is reduced in steps of $\frac{1}{2}$, it follows that

$$\alpha_n \ge \min(1, (1-\theta_{sd})/L)$$

and (11) now follows immediately. □

***Remark 4.3.*** The requirements of global Lipschitz continuity and boundedness of $f$ from below can be relaxed. Most textbooks require only that $\nabla f$ is Lipschitz continuous in the sublevel set $\{x \in \mathbb{R}^N : f(x) \leq f(x_0)\}$, and that the sequence $(f(x_n))_{n\in\mathbb{N}}$ is bounded below.

Convergence of the iterates $(x_n)_{n\in\mathbb{N}}$ is established in Problem 4.1. □

***Problem 4.1.*** In Theorem 4.2 we have established, in essence, that $\nabla f(x_n) \to 0$, but we have not excluded divergence of the iterates $(x_n)_{n\in\mathbb{N}}$. However, in many applications we would not expect this to occur.

Establish (a)–(c), assuming, in addition to the hypothesis of Theorem 4.2, that $f$ is coercive, i.e.,

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

(a) The iterates $(x_n)_{n\in\mathbb{N}}$ are bounded.

(b) There exists a convergent subsequence $x_{n_k} \to x_*$ where $\nabla f(x_*) = 0$.

(c) Suppose now that this accumulation point $x_*$ is a *strict local minimizer*. In this case, the entire sequence converges, i.e., $x_n \to x_*$ as $n \to \infty$. □

Note that, in Theorem 4.2 we have not adressed the convergence rate of the steepest descent algorithm. We will investigate this in a simplified situation. A geometric picture of the poor performance of Steepest Descent, even for only slightly ill-conditioned problems, is shown in Figure 1.

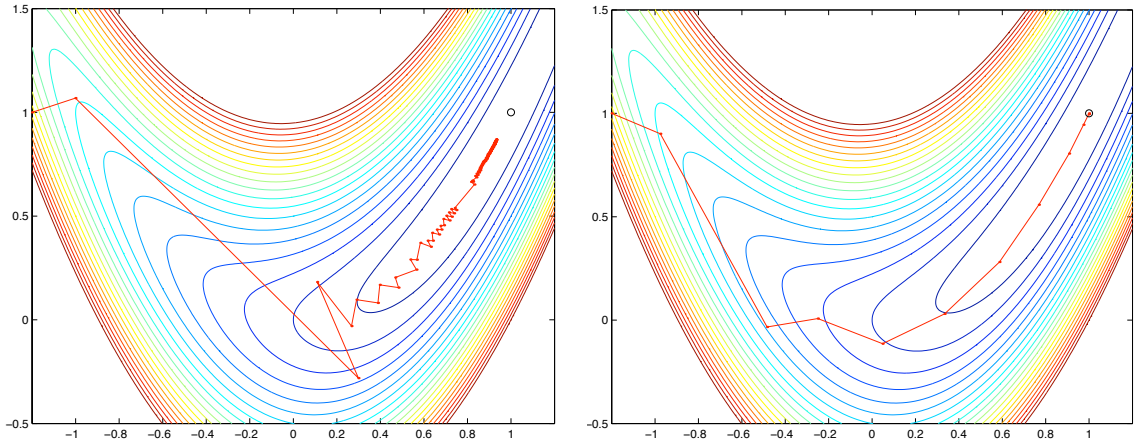


Figure 1: Iterates of a steepest descent method, and of Newton's method for the objective function $f(x, y) = 10(y - x^2)^2 + (x - 1)^2$. We observe that both methods behave similarly away from the solution, but that Newton's method is significantly more efficient in the final steps. Because it has no second derivative information, the steepest descent method oscillates back and forth between the sides of the 'energy valley'. (image taken from [3])

We make this observation precise in the following result.

**Proposition 4.4.** *Suppose that* $f(x) = \frac{1}{2}x^T Ax$ *where* $A$ *is spd and* $\|A\| \geq 1$ *(if* $\|A\| <$

1, *then we need to rescale the problem). Suppose, further, that the sequence* $(x_n)_{n\in\mathbb{N}}$ *is generated by Algorithm 4.2. Then* $f(x_n) \to 0$ *q-linearly with q-factor* $1-2\theta_{sd}(1-\theta_{sd})/\kappa(A)$.

*Proof.* Recall from the proof of Theorem 4.2 that $\alpha_n \geq \min(1, (1-\theta_{sd})/L)$ where $L$ is a local or global Lipschitz constant for $\nabla f$. Since, in the present case, $\nabla f(x) = Ax$, it follows that $L = \|A\|$. In particular, since $\|A\| \geq 1$, we have $\alpha_n \geq (1-\theta_{sd})/\|A\|$.

From the Armijo condition (10) we obtain

$$f(x_{n+1}) \leq f(x_n) - \theta_{sd}\alpha_n|\nabla f(x_n)|^2 \leq f(x_n) - \frac{\theta_{sd}(1-\theta_{sd})}{\|A\|}|Ax_n|^2.$$

Finally, using $y^TAy \geq \|A^{-1}\|^{-1}|y|^2$, we deduce

$$f(x_{n+1}) \leq f(x_n) - \frac{\theta_{sd}(1-\theta_{sd})}{\|A\|\|A^{-1}\|}x_n^TAx_n = \Big(1 - \frac{2\theta_{sd}(1-\theta_{sd})}{\kappa(A)}\Big)f(x_n). \qquad \square$$

***Remark 4.5.*** If we define the *energy norm* $|x|_A = \sqrt{x^TAx}$ then we see that $|x_n|_A^2 \to 0$ q-linearly with q-factor $c$, say, and as a consequence, $|x_n|_A \to 0$ with q-factor $\sqrt{c}$.

Moreover, $|x_n| = |A^{-1/2}A^{1/2}x| \leq \|A^{-1/2}\||x|_A$ which shows that $x_n \to 0$ r-linearly in the standard norm $|\cdot|_2$. $\square$

***Problem 4.2.*** In Proposition 4.4, we have shown only that $f(x_n)$ converges *at least* q-linearly. Here, we show that this result is sharp. Let $f(x) = \frac{1}{2}x^TAx$ again, where

$$A = \begin{pmatrix} \gamma & 0 \\ 0 & 1 \end{pmatrix}.$$

Consider applying the steepest descent method with *exact line-searches* to this objective function.

(a) Prove that $f$ is strictly convex and has a unique critical point $x_* = 0$ which is the global minimizer.

(b) Prove that the function $\phi(\alpha) = f(x_n - \alpha\nabla f(x_n))$ has a unique minimizer $\alpha_n$ which is the step-length for the exact line-search. Compute $\alpha_n$ explicitly.

(c) Show that, if $x_0 = (1, \gamma)^T$, then

$$x_n = \left(\frac{\gamma-1}{\gamma+1}\right)^n \begin{pmatrix} (-1)^n \\ \gamma \end{pmatrix}.$$

(d) Deduce that $x_n \to x_*$ q-linearly with q-factor $c := (\kappa(A)-1)/(\kappa(A)+1) = 1-2/(1+\kappa(A))$, and that $f(x_n) \to 0$ q-linearly with q-factor $c^2$. $\square$

The results of Proposition 4.4 and of Problem 4.2 can be generalized. The proof of the following theorem is essentially a perturbation of the proof of Proposition 4.4 and not particularly interesting.

**Theorem 4.6.** *Suppose* $f \in \mathrm{C}^2(\mathbb{R}^N)$ *with bounded Hessians,* $\sup_{x\in\mathbb{R}^N}\|\nabla^2 f(x)\| < +\infty$, *and that the iterates* $(x_n)_{n\in\mathbb{N}}$ *generated by the steepest descent method converge to a point* $x_*$ *where* $\nabla f(x_*) = 0$ *and* $\nabla^2 f(x_*) > 0$ *(in particular* $x_*$ *is a strict local minimizer).*

*Then* $f(x_n) \to f(x_*)$ *q-linearly with q-factor* $1 - c/\kappa(\nabla^2 f(x_*))$ *and* $x_n \to x_*$ *r-linearly: namely, for* $n$ *sufficiently large, we have* $|x_n - x_*| \leq C\sqrt{f(x_n) - f(x_*)}$, *where* $C = 2(\min\sigma(\nabla^2 f(x_*)))^{-1/2}$.

## 4.2 General descent methods

In the preceding section, we have developed a first algorithm for which we are able to prove *global convergence*. We have seen, however, that its local performance can be very poor. In the present and the following section, we will begin the investigation of generalizations of the Steepest Descent Method, keeping (and improving) the global convergence properties while considerably improving the local convergence speed.

We can formulate a general line-search algorithm as follows:

**Algorithm 4.3 (General Descent Method).**
**Input:** $x_0 \in \mathbb{R}^N$, $\theta_{sd} \in (0,1)$;
1: **for** $n = 0, 1, 2, \ldots$ **do**
2: Choose a descent direction $s_n$;
3: $\alpha_n \leftarrow \text{LINESEARCH}[x = x_n,\, s = s_n,\, \theta_{sd}]$;
4: $x_{n+1} = x_n + \alpha_n s_n$;
5: **end for**

The following theorem establishes global convergence of the general descent method, provided that the angle between the search direction and the steepest descent direction are uniformly bounded away from $\pi/2$.

**Theorem 4.7 (Global Convergence of General Descent Methods).** *Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ be bounded below, and assume $\nabla f$ is globally Lipschitz continuous. Moreover, assume that there exists a constant $\delta > 0$ such that $|s_n| \geq \delta |\nabla f(x_n)|$ for all $n$. Then, for any $\theta_{sd} \in (0,1)$ and $x_0 \in \mathbb{R}^N$,*

$$\sum_{n=0}^{\infty} \cos(\theta_n)^2 |\nabla f(x_n)|^2 < +\infty, \tag{13}$$

*where* $\cos(\theta_n) = -\frac{\nabla f(x_n)}{|\nabla f(x_n)|} \cdot \frac{s_n}{|s_n|}$.

***Problem 4.3.*** Prove Theorem 4.7. Hint: follow the proof of Theorem 4.2 closely and distinguish the two cases (i) $\alpha_n < 1$ and (ii) $\alpha_n \geq 1$. □

***Remark 4.8.*** The condition $|s_n| \geq \delta |\nabla f(x_n)|$ in Theorem 4.7 can be removed if we generalize the LINESEARCH algorithm to allow a larger initial $\alpha$, for example, $\alpha_{init} = |\nabla f(x_n)|/|s_n|$. □

## 4.3 Variable-Metric Steepest Descent

The general setting of Algorithm 4.3 is only useful if we can provide a simple method to compute the descent direction $s_n$. The simplest choice is the steepest descent direction $s_n = -\nabla f(x_n)$ which we have analyzed in Section 4.1, finding that it performs poorly for many problems.

The fault, however, does not lie with the steepest-descent idea itself. To motivate the following discussion, let us recall from the proof of Proposition 4.4 that short step-lengths are mainly caused by a large local Lipschitz constant for $\nabla f$ (in other words, $f(x_n) +$

$\alpha\nabla f(x_n)\cdot s_n$ is not a good approximation for $f(x_n+\alpha s_n)$ unless $\alpha$ is very small). However, the local Lipschitz constant,

$$\mathrm{Lip}_U(\nabla f) = \sup_{x,x'\in U} \frac{|\nabla f(x) - \nabla f(x')|}{|x-x'|}$$

can depend strongly on the norm, in this case $|\cdot|$, with respect to which we measure it!

We will generalize the steepest-descent idea by simply using a different choice of norm, namely,

$$|x|_B := \left(x^T B x\right)^{1/2},$$

where $B \in \mathbb{R}^{N\times N}$ is symmetric and positive definite. The $B$-norm $|\cdot|_B$ is still a Euclidean norm as it is associated with the inner product $(x,x') \mapsto x^T B x'$.

At the beginning of Section 4.1 we minimized the *slope* $\nabla f(x)\cdot s$ over all search directions $s \in \mathbb{R}^N$ with unit norm $|s| = |s|_2 = 1$ to obtain the steepest descent direction. Now, we minimize it over all directions with unit $B$-norm $|s|_B = 1$.

**Proposition 4.9.** *The direction of steepest descent, with respect to the $B$-norm, of $f$ at $x$ is $\hat{s} = -B^{-1}\nabla f(x)/|B^{-1}\nabla f(x)|_B$, i.e.,*

$$\nabla f(x)\cdot\hat{s} \le \nabla f(x)\cdot s \qquad \forall s\in\mathbb{R}^N, |s|_B = 1.$$

*Proof.* Let $|s|_B = 1$, then, using the Cauchy–Schwartz Inequality for the $B$-inner product,

$$\begin{aligned}\nabla f(x)\cdot\hat{s} = & -(B^{-1}\nabla f(x))^T B \frac{B^{-1}\nabla f(x)}{|B^{-1}\nabla f(x)|_B} = -|B^{-1}\nabla f(x)|_B \\ = & -|B^{-1}\nabla f(x)|_B |s|_B \le (B^{-1}\nabla f(x))^T B s = \nabla f(x)\cdot s. \end{aligned}$$

□

***Remark 4.10.*** Another point of view we could take is to note that

$$\nabla f(x)\cdot s = (B^{-1}\nabla f(x))^T B s \qquad \forall s\in\mathbb{R}^N,$$

i.e., the vector $B^{-1}\nabla f(x)$ *represents* the gradient $\nabla f(x)$ in the $B$-inner product. Consequently, we call

$$\nabla_B f(x) := B^{-1}\nabla f(x)$$

the $B$-gradient of $f$ at $x$ (or gradient with respect to the metric induced by $B$). Thus, the steepest descent direction with respect to the $B$-norm is $-\nabla_B f(x)$. □

Let us now briefly investigate the Lipschitz constant of $\nabla_B f$ with respect to the metric $B$. Note that, for $R$ "small", we have

$$L_I := \sup_{x'\in\bar{B}_R(x)} \frac{|\nabla f(x) - \nabla f(x')|}{|x-x'|} \approx \sup_{x'\in\bar{B}_R(x)} \frac{|\nabla^2 f(x)(x-x')|}{|x-x'|} = \|\nabla^2 f(x)\|,$$

while, for positive definite $B \in \mathbb{R}^{N\times N}$, we have

$$\begin{aligned} L_B := & \sup_{|x'-x|_B\le R} \frac{|\nabla_B f(x) - \nabla_B f(x')|_B}{|x-x'|_B} \approx \sup_{|x'-x|_B\le R} \frac{|B^{-1}\nabla^2 f(x)(x-x')|_B}{|x-x'|_B} \\ = & \sup_{|B^{1/2}(x'-x)|\le R} \frac{|B^{-1/2}\nabla^2 f(x) B^{-1/2} B^{1/2}(x-x')|}{|B^{1/2}(x-x')|} = \|B^{-1/2}\nabla^2 f(x) B^{-1/2}\|. \end{aligned}$$

In particular, if $\nabla^2 f(x)$ is positive definite and if $B = \nabla^2 f(x)$ then $L_B = 1$. Note, that this choice corresponds precisely to Newton's method, which, in this context, is optimal!

In order to allow maximal flexibility in our choice of descent directions we allow the *metric* $B$ to change at each step of the descent method.

**Algorithm 4.4 (Generalized Steepest Descent Method).**
**Input:** $x_0$, $\theta_{sd}$
1: **for** $n = 0, 1, 2, \dots$ **do**
2: Choose an spd matrix $B_n \in \mathbb{R}^{N\times N}$;
3: $s_n \leftarrow -\nabla_{B_n} f(x_n) = -B_n^{-1}\nabla f(x_n)$;
4: $\alpha_n \leftarrow \text{LINESEARCH}[x = x_n, s = s_n, \theta_{sd}]$;
5: $x_{n+1} = x_n + \alpha_n s_n$
6: **end for**

***Remark 4.11.*** Another motivation for Algorithm 4.4 (and the more common one in the optimization literature) is to assume that $B_n$ is an approximation to $\nabla^2 f(x_n)$. In this case, the quadratic model

$$m_n(x) = f(x_n) + \nabla f(x_n)\cdot(x-x_n) + \tfrac{1}{2}(x-x_n)^T B_n (x-x_n)$$

can be expected to be a better approximation to $f$ than the linear model. If $B_n$ is positive definite then $m_n$ has a unique minimizer

$$x = x_n - B_n^{-1}\nabla f(x_n).$$

The variable-metric steepest descent algorithm then simply *damps* the steplengths in order to achieve global convergence. □

Algorithm 4.4 still leaves us with the somewhat nebulous task of deciding how to choose the metrics $B_n$ at each step. We shall defer a more detailed discussion until we have established to global and local convergence results and only note that the matrices $B_n$ should be spd (and hence $B_n^{-1}$ is also spd) so that the $|\cdot|_{B_n}$ are really norms. This guarantees that the directions $s_n = -\nabla_{B_n} f(x_n)$ are indeed descent directions: if $\nabla f(x_n) \neq 0$, then

$$\nabla f(x_n)\cdot s_n = -\nabla f(x_n)^T B_n^{-1}\nabla f(x_n) < 0.$$

For our global convergence result we shall require the much stronger condition that the matrices $B_n$ have uniformly bounded condition number.

**Theorem 4.12 (Global Convergence of Generalized Steepest Descent).** *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ is bounded below and that $\nabla f$ is globally Lipschitz continuous. Assume, further, that the matrices $B_n$ chosen in Algorithm 4.4 satisfy $\kappa(B_n) \le \bar{\kappa} < +\infty$.*

*Then, for any line-search parameter $\theta_{sd} \in (0,1)$, and any $x_0 \in \mathbb{R}^N$ Algorithm 4.4 is* globally convergent*:*

$$\sum_{n=0}^{\infty} |\nabla_{B_n} f(x_n)|_{B_n}^2 = \sum_{n=0}^{\infty} |B_n^{-1/2} \nabla f(x_n)|^2 < +\infty. \tag{14}$$

*In particular,* (11) *holds.*

*Proof.* Let $f_* := \inf_{n\in\mathbb{N}} f(x_n) > -\infty$, $U_n = \{f(x) \le f(x_n)\}$, $d := \operatorname{dist}(U_1, \partial U_0)$, $L = \operatorname{Lip}_{U_0}(\nabla f)$.

From the sufficient-decrease condition guaranteed by the backtracking linesearch, we obtain

$$\theta_{sd} \sum_{n=0}^{\infty} \alpha_n |\nabla_{B_n} f(x_n)|_{B_n}^2 \le f(x_0) - f_*.$$

Next, we note that $|s_n|_{B_n} \ge \sqrt{b_0}|s_n|$, and hence, if $\alpha \le \sqrt{b_0} d/|s_n|_{B_n} =: r_n$ then, as in (12), we obtain

$$f(x_n + \alpha s_n) \le f(x_n) - \alpha |\nabla_{B_n} f(x_n)|_{B_n}^2 + \tfrac{1}{2} L_{B_n} \alpha^2 |\nabla_{B_n} f(x_n)|_{B_n}^2,$$

where

$$L_{B_n} = \sup_{|x - x_n|_{B_n} \le \min(1, r_n)} \frac{|\nabla_{B_n} f(x) - \nabla_{B_n} f(x_n)|_{B_n}}{|x - x_n|_{B_n}} \le L/b_0.$$

In particular, for

$$\alpha_n \ge \min\big(1, \sqrt{b_0} d/2|s_n|_{B_n}, (1-\theta_{sd})/L_n\big),$$

the backtracking algorithm terminates. We can now proceed as in the proof of Theorem 4.2. If the sequence $|s_n|_{B_n}$ is bounded above, then $\inf_n \alpha_n > 0$ and hence the result follows. If the sequence $|s_n|_{B_n}$ were unbounded, then we would deduce $f_* = -\infty$. □

From the proof of Theorem 4.12 we see that the local Lipschitz constant $L_B$ defined above is the correct constant to measure step lengths. Thus, if we can choose the metrics $B_n$ in such a way that the algorithm can take bigger steps, then we will typically require far fewer iterations of the steepest descent method.

Next, we investigate the local convergence speed of the generalized steepest descent algorithm. As before, we prove a result only for the simple situation where $f$ is a positive quadratic.

**Proposition 4.13.** *Suppose that $f(x) = \frac{1}{2} x^T A x$ where $A$ is symmetric and positive definite, and $\|B^{-1}A\| \ge 1$ (otherwise, we rescale the problem). Suppose, further, that $(x_n)_{n\in\mathbb{N}}$ is generated by Algorithm 4.4 with $B_n = B$. Then $f(x_n) \to 0$ q-linearly with q-factor $1 - 2\theta_{sd}(1-\theta_{sd})/\kappa(B^{-1}A)$.*

*Proof.* We have $\nabla_B f(x) = B^{-1}Ax$, and hence $L_B = \|B^{-1/2} A B^{-1/2}\|$. Since $\nabla f$ is globally Lipschitz continuous, we recall from the proof of Theorem 4.2 that $\alpha_n \ge \min(1, (1 - \theta_{sd})/L_B)$. In particular, since we assumed that $\|B^{-1}A\| \ge 1$, we have $\alpha_n \ge (1-\theta_{sd})/L_B$.

From the Armijo condition we obtain

$$f(x_{n+1}) \le f(x_n) - \theta_{sd}\alpha_n |\nabla_B f(x_n)|_B^2 \le f(x_n) - \frac{\theta_{sd}(1-\theta_{sd})}{L_B} |B^{-1/2} A x_n|^2.$$

Setting $y = A^{1/2}x_n$, we have

$$|B^{-1/2}Ax_n|^2 = |(B^{-1/2}A^{1/2})y| = y^T A^{1/2}B^{-1}A^{1/2}y \geq \mu_0|y|^2,$$

where $\mu_0$ is the smallest eigenvalue of $A^{1/2}B^{-1}A^{1/2}$, and we can deduce

$$f(x_{n+1}) \leq \Big(1 - \frac{2\theta_{sd}(1-\theta_{sd})}{L_B/\mu_0}\Big) f(x_n).$$

To conclude the proof, we need to show that $L_B/\mu_0 \leq \kappa(B^{-1}A)$. First, we notice that

$$A^{1/2}B^{-1}A^{1/2}z = \mu z \quad \Leftrightarrow \quad B^{-1}Az' = \mu z',$$

where $z' = A^{-1/2}z$, and hence, $\mu_0$ is also the smallest eigenvalue of $B^{-1}A$, or equivalently, $1/\mu_0$ is the largest eigenvalue of $A^{-1}B = (B^{-1}A)^{-1}$. This implies $1/\mu_0 \leq \|(B^{-1}A)^{-1}\|$. Furthermore, $L_B$ is the largest eigenvalue of $B^{-1/2}AB^{-1/2}$, and by a similar argument, it is also the largest eigenvalue of $B^{-1}A$, and hence $L_B \leq \|B^{-1}A\|$. Hence, we conclude that, indeed, $L_B/\mu_0 \leq \kappa(B^{-1}A)$. □

Proposition 4.13 indicates that, for general twice differentiable $f$, we can expect a reduction in the objective by roughly $1-c/\kappa(B_n^{-1}\nabla^2 f(x_n))$ at each step. Although $c$ depends in theory on $\theta_{sd}(1-\theta_{sd})$, this is usually overly pessimistic and can safely be ignored. Thus, even if we do not achieve the quadratice convergence of Newton's method, if $\kappa(B_n^{-1}\nabla^2 f(x_n))$ remains bounded by a moderate constant, then the $q$-factor can be lowered significantly in this *preconditioning* process, as we shall henceforth label it, leading to a much fast linear convergence rate.

Based on this observation, on our discussions above, and on the formulation of the algorithm, we can assemble the following wish list for the matrices $B_n$:

(B.1) The matrices $B_n$ must be symmetric and positive definite, so that $|\cdot|_{B_n}$ is a norm. This guarantees that the directions $s_n = -B_n^{-1}\nabla f(x_n)$ are descent directions.

(B.2) Ideally $B_n \approx \nabla^2 f(x_n)$ in order to mimic Newton's method, or, more generally, $\kappa(B_n^{-1}\nabla^2 f(x_n))$ should be moderate in order to allow large step lengths in the pre-asymptotic range, and to obtain good local convergence properties.

(B.3) The matrix-vector multiplications $B_n x$ and $B_n^{-1}x$ should be efficient to carry out.

Several strategies exist how to pick $B_n$ at each step of the steepest descent method. The following list discusses some of the more common ideas:

1. *Newton's Method:* Unfortunately, the choice $B_n = \nabla^2 f(x_n)$ is only possible in rare circumstances. But whenever possible, it should be made. The line-search aspect then leads to a globally convergent Newton method with quadratic local convergence properties.

2. *User-defined Metric:* For many problems, an appropriate metric $B$, in which the objective function $f$ has good scaling properties, i.e., $\kappa(B^{-1}\nabla^2 f(x))$ is moderate for all $x$, can be found. Analytical insight can lead to some of the best preconditioners for the steepest descent method.

3. *Damped Newton / Levenberg–Marquardt Method:* Sometimes, the choice $B_n = \nabla^2 f(x_n) + \mu_n E$ is made where $E$ is symmetric, positive definite, and $\mu_n \geq 0$ (the Levenberg–Marquardt parameter) can be adjusted so that $B_n$ is positive definite as well. The matrix $E$ should again be chosen by the user and will ideally be chosen so that $\kappa(E^{-1}\nabla^2 f(x))$ is moderate for all $x$. If $\mu_n = 0$ is eventually chosen then this method reduces to Newton's method and therefore exhibits locally quadratic convergence. In general, one can choose $\mu_n \searrow 0$, in which case one obtains superlinear convergence.

4. *Quasi-Newton Methods:* Practicioners do not, in general, want to worry about preconditioning the optimization method but would prefer that their software could automatically choose the matrices $B_n$. Furthermore, Hessian's are often expensive to compute and even more expensive to invert. Quasi-Newton methods assemble approximations $B_n$ to the Hessians (or ecen their inverses) by recycling quantities already used in the optimization process, thus achieving superlinear convergence. These methods will be covered in Section 6.

# 5 Trust Region Methods

Line search methods, based on the one-dimensional approximate minimization along lines

$$\min_{\alpha \geq 0} f(x_n + \alpha s_n)$$

are one of two fundamental approaches to globally convergent unconstrained optimization methods. *Trust Region Methods* constitute a second fundamental class which is based on the following principles:

1. In iteration $n$ we locally replace the objective function $f(x)$ by a quadratic *model* $m_n(x)$ which is easier to minimize.

2. We choose a neighbourhood $R_n$ of the current iterate $x_n$ where $m_n$ is *trusted* to approximate $f$ well.

3. The next iterate $x_{n+1}$ is found by approximately minimising the model function $m_n$ over the trust region $R_n$:
$$x_{n+1} \approx \operatorname*{argmin}_{x \in R_n} m_n(x) \tag{15}$$

Note that the *trust region subproblem* (15) is *contrained* optimisation problem. This seems to be the exact opposite of the usual practise of replacing constrained problems by a sequence of unconstrained problems. Thus, the trust region subproblem can only solved efficiently if $R_n$ and $m_n$ are simple enough.

In order to recover the local convergence speed of (Quasi-)Newton methods, we will use a quadratic model,

$$m_n(x) = f(x_n) + \nabla f(x_n) \cdot (x - x_n) + \tfrac{1}{2}(x - x_n)^T H_n (x - x_n),$$

where $H_n \in \mathbb{R}^{N \times N}$ is symmetric, and should be an approximation to the Hessian $\nabla^2 f(x_n)$. Note that we are now minimizing $m_n$ over a bounded region only, and hence we do not require anymore that $H_n$ is positive definite!

The trust region $R_n$ is typically a closed ball in some norm. For simplicity, we shall only consider the standard Euclidean 2-norm,

$$R_n = \{x \in \mathbb{R}^N : |x - x_n| \leq \Delta_n\},$$

where $\Delta_n > 0$ is called the *trust region radius* and is adjusted at each step. The trust region radius should be updated at each step and reflects the local variation of the hessian, i.e., the quality the model $m_n$.

A prototype for a trust region method can be formulated as follows:

**Algorithm 5.1 (Prototype Trust Region Method).**

**Input:** $x_0$, $\Delta_0$, $B$

1: **for** $n = 0, 1, 2, \ldots$ **do**
2: Compute $\nabla f(x_n)$ and $H_n$.
3: Approximately solve the trust region subproblem (15) to obtain a candidate $y_{n+1}$
4: Decide whether to accept $y_{n+1}$ ($x_{n+1} = y_{n+1}$) or to reject it ($x_{n+1} = x_n$).
5: Adjust the trust region radius to get a new radius $\Delta_{n+1}$.
6: **end for**

We will now, in turn, adress each step of the trust region method separately.

***Remark 5.1.*** For problems where the standard 2-norm does is not a suitable metric, it may be preferrable to choose norm balls with respect to a $B$-norm as trust regions, i.e.,

$$R_n = \big\{x \in \mathbb{R}^N : |x - x_n|_B \leq \Delta_n\big\}.$$

Although we could again adjust $B$ at each step, this may be unnecessary, since we can obtain additional curvature information from the (approximate) hessians $H_n$. □

## 5.1 The Cauchy-Point

The solution of the trust region subproblem is the most crucial aspect in the implementation of the trust region method, and the subject of ongoing research. Before we concentrate on this difficult task, we will begin to investigate under which conditions a trust region method can be guaranteed to converge. A simple idea is to make it *at least* as efficient as a steepest descent method.

The Cauchy point is obtained when a steepest descent line-search is applied to $m_n$ at $x_n$ and is restricted to $R_n$. Since $m_n$ is quadratic, we do an exact linesearch. Let $s_n = -\nabla f(x_n)$, then

$$m_n(x_n + \alpha s_n) = f(x_n) - \alpha|\nabla f(x_n)|^2 + \tfrac{1}{2}\alpha^2 \nabla f(x_n)^T H_n \nabla f(x_n).$$

Thus, the step length which minimizes the model in $R_n$ is given by

$$\alpha_n^c = \begin{cases} \frac{\Delta_n}{|\nabla f(x_n)|}, & \text{if } \nabla f(x_n)^T H_n \nabla f(x_n) \leq 0, \\ \min\left(\frac{\Delta_n}{|\nabla f(x_n)|}, \frac{|\nabla f(x_n)|^2}{\nabla f(x_n)^T H_n \nabla f(x_n)}\right), & \text{if } \nabla f(x_n)^T H_n \nabla f(x_n) > 0. \end{cases} \tag{16}$$

The *Cauchy Point* of the trust region subproblem is defined as

$$x_n^c = x_n - \alpha_n^c \nabla f(x_n). \tag{17}$$

When we discuss different methods for the solution of the trust region subproblem later, we will always require that they achieve at least as much reduction of the model as the Cauchy point does. In some sense we can immagine that by enforcing this, we guarantee that the trust region method is at least as good as the steepest descent method and will therefore be globally convergent.

### 5.2 Accepting and rejecting updates; trust region radius management

After we have computed a candidate $y_{n+1}$ for the next iterate, we need to decide whether it is in fact a good iterate. To this end, we will compare the decrease in the model function with the actual decrease in function value, i.e., we define

$$\rho_n = \frac{f(x_n) - f(y_{n+1})}{m_n(x_n) - m_n(y_{n+1})}. \tag{18}$$

The ratio $\rho_n$ between *actual reduction* and *predicted reduction* also tells us whether $m_n$ is a good local approximation to $f$ in $R_n$ and therefore can be used to adjust the trust region radius if necessary. We will define two user-defined parameters $\eta_{ac} \in (0, 1/4)$ and $\Delta_{max} > 0$ and use the following heuristics to decide whether to accept the candidate $y_{n+1}$ and to adjust the trust region radius.

- If $\rho_n \geq \rho_{ac}$, then $y_{n+1}$ is accepted, i.e., $x_{n+1} = y_{n+1}$. Otherwise, $x_{n+1} = x_n$.
- If $\rho_n < \frac{1}{4}$, we set $\Delta_{n+1} = \frac{1}{4}|y_{n+1} - x_n|$.
- If $\rho_n > \frac{3}{4}$, and if $|x_{n+1} - x_n| = \Delta_n$ then we set $\Delta_{n+1} = 2\Delta_n$.
- In all other cases we leave the radius unchanged, $\Delta_{n+1} = \Delta_n$.

This, together with the idea of the Cauchy Point, leads to the following basic trust region algorithm:

**Algorithm 5.2 (Trust Region Method).**

**Input:** $x_0$, $\Delta_0$, $\eta_{ac}$;

**for** $n = 0, 1, 2, \ldots$ **do**
    Evaluate $H_n$ and $\nabla f(x_n)$;
    Compute an approximate minimizer $y_{n+1}$ of (15);
    Evaluate $\rho_n$ using (18);
    **if** $\rho_n \geq \eta_{ac}$ **then**
        $x_{n+1} \leftarrow y_{n+1}$;
    **else**
        $x_{n+1} \leftarrow x_n$;
    **end if**
    **if** $\rho_n < 1/4$ **then**
        $\Delta_{n+1} \leftarrow \frac{1}{4}|y_{n+1} - x_n|$
    **else if** $\rho_n > \frac{3}{4}$ and $|y_{n+1} - x_n| = \Delta_n$ **then**
        $\Delta_{n+1} \leftarrow \min(2\Delta_n, \Delta_{max})$;
    **else**
        $\Delta_{n+1} \leftarrow \Delta_n$;

16: **end if**
17: **end for**

## 5.3 Globally convergence of trust region methods

In this section, we state and prove the global convergence result for the trust region method formulated in Algorithm 5.2. Since we are mainly concerned with convergence, we use some fairly rough estimates to establish convergence in quite general situations.

It is clear that we will need to establish the fact that $y_{n+1}$ is accepted for a sufficiently small trust region radius. Hence, we will carefully study this question before moving on to the actual convergence result. We will re-use parts of the proof of the following Lemma in the final convergence proof, so it should be studied carefully.

**Lemma 5.2.** *Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ and let $\nabla f$ be Lipschitz cts. with Lipschitz constant $L$.*

*(i) If $\nabla f(x_n) \neq 0$, and if $\Delta_n \leq |\nabla f(x_n)|/\|H_n\|$, then $\alpha_n^c = \Delta_n/|\nabla f(x_n)|$ and*

$$m_n(x_n) - m_n(x_n^c) \geq \tfrac{1}{2}\Delta_n |\nabla f(x_n)|.$$

*(ii) Under the same conditions,*

$$\rho_n \geq 1 - \frac{\Delta_n(L + \|H_n\|)}{|\nabla f(x_n)|}.$$

*(iii) In particular, if $\Delta_n \leq \frac{3}{4}|\nabla f(x_n)|/(L + \|H_n\|)$, then $\rho_n \geq 1/4$.*

***Problem 5.1.*** Prove Lemma 5.2. □

Lemma 5.2 immediately implies the approximate solutions of the trustregion subproblems will be accepted as soon as $\Delta_n$ has been reduced enough, and hence $y_{n+1}$ is accepted infinitely often. We can use this result to prove global convergence of the trust region method.

**Theorem 5.3.** *Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ is bounded below, and assume that $\nabla f$ is Lipschitz continuous and that $\max_{n\in\mathbb{N}} \|H_n\| =: \beta < +\infty$. Suppose, further, that all solutions $y_{n+1}$ of the trust region subproblem satisfy $m_n(y_{n+1}) \leq m_n(x_n^c)$, $n \in \mathbb{N}$. Then*

$$\liminf_{n\to\infty} |\nabla f(x_n)| = 0.$$

*Proof.* Suppose, for contradiction, that $|\nabla f_n| \geq \epsilon > 0$ for all $n$. Lemma 5.2 (iii) implies that, if $\Delta_n \leq \frac{3}{4}\epsilon/(L+\beta)$ then $y_{n+1}$ is accepted and $\Delta_{n+1} \geq \Delta_n$. Hence, we can conclude that

$$\Delta_n \geq \min\left(\Delta_0, \tfrac{3}{16}\epsilon/(L+\beta)\right) =: \Delta_{min}.$$

Suppose that $n$ is an index where $y_{n+1}$ was accepted, then $\rho_n \geq \eta_{ac}$, which reads

$$f(x_n) - f(x_{n+1}) \geq \eta_{ac}(m_n(x_n) - m_n(x_{n+1})) \geq \eta_{ac}(m_n(x_n) - m_n(x_n^c)).$$

Furthermore, we know that $\alpha \mapsto m_n(x_n - \alpha \nabla f(x_n))$ is strictly decreasing for $0 \leq \alpha \leq \alpha_n^C$. Hence, we can estimate $m_n(x_n^C)$ above by

$$m_n(x_n^C) \leq m_n(x_n - \tfrac{\Delta_{min}}{|\nabla f_n|} \nabla f_n).$$

For this possibly smaller step, Proposition 5.2(i) gives

$$m_n(x_n) - m_n(\tilde{x}_n^C) \geq \tfrac{1}{2} \Delta_{min} |\nabla f_n|$$

Hence, we conclude that, whenever a guess $y_{n+1}$ is accepted,

$$f(x_n) - f(x_{n+1}) \geq \tfrac{1}{2} \eta_{ac} \Delta_{min} |\nabla f(x_n)| = c |\nabla f_n|,$$

where $c = c(\eta_{ac}, \epsilon, L, \beta, \Delta_0) > 0$. In particular, if $(n_j)_{j \in \mathbb{N}}$ is the subsequence of all those $n$ where $y_{n+1}$ is accepted then

$$c \sum_{j=0}^{\infty} |\nabla f(x_{n_j})| \leq f(x_{n_0}) - \inf f < +\infty,$$

which gives the desired contradiction.

In particular, there exists at least a subsequence of gradients which tend to zero, and thus the result is established. □

## 5.4 The dogleg method

We have proven global convergence of *any* trust region algorithm for which the solution $y_{n+1}$ of the trust region subproblem satisfies $m_n(y_{n+1}) \leq m_n(x_n^C)$. For example, taking $y_{n+1} = x_n^C$ will give a convergent method, however, this would simply result in the steepest descent method which we know to perform badly for ill-conditioned problems. Thus, we need to find a more sophisticated way of computing $y_{n+1}$. The real advanteage of the trust region framework is the ease with which this can be achieved.

Let us assume, for the remainder of this section that $H_n = \nabla^2 f(x_n)$ for all $n$. In this case, we could use the following strategy:

> If $\nabla^2 f(x_n) > 0$ and if $x_n^N := x_n - \nabla^2 f(x_n) \nabla f(x_n) \in R_n$
> then take $y_{n+1} = x_n^N$, and otherwise take $y_{n+1} = x_n^C$.

The *dogleg method* is a more practical version of this principle. Note, for example, that testing whether $\nabla^2 f(x_n)$ is positive definite is fairly expensive and should be avoided. The only information we have freely available is whether $\nabla f_n^T H_n \nabla f_n > 0$.

In the following, we will carefully motivate the dogleg idea in several steps.

1. If $\nabla f_n^T H_n \nabla f_n \leq 0$, or if $H_n$ is not invertible, then we discard the Newton idea and take $y_{n+1} = x_n^C$. Of course we cannot test directly whether $H_n$ is invertible, however, if we simply attempt to solve the linear system defining the (quasi-)Newton step, we will find out whether the system is ill-conditioned.

2. If $\nabla f_n^T H_n \nabla f_n > 0$ and $H_n$ is invertible, then the Newton point $x_n^N$ (defined whenever $H_n$ is invertible) and the unidirectional minimizer $x_n^U$ (defined whenever $\nabla f_n^T H_n \nabla f_n > 0$), are well-defined:

$$x_n^N := x_n - \nabla^2 f(x_n)^{-1} \nabla f(x_n), \qquad \text{and} \qquad x_n^U := x_n - \frac{|\nabla f_n|^2}{\nabla f_n^T H_n \nabla f_n} \nabla f_n.$$

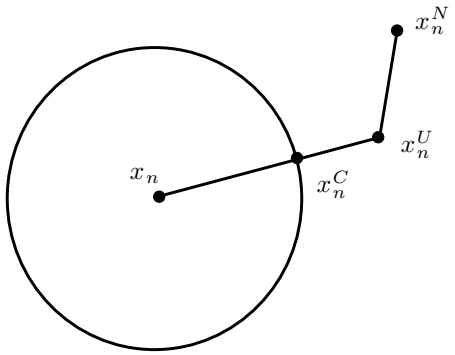


Figure 2: The dogleg path.

We take those as nodes in the piecewise linear *dogleg path* (cf. Figure 2)

$$\begin{aligned}\mathcal{C}_n = &\ \text{conv}\{x_n, x_n^U\} \cup \{x_n^U, x_n^N\} \\ = &\ \{tx_n + (1 - tx_n^U) : 0 \le t \le 1\} \cup \{tx_n^U + (1-t)x_n^N\}.\end{aligned}$$

3. Our goal will be to minimize $m_n$ over $\mathcal{C}_n \cap R_n$. One can prove that, if $H_n > 0$ then the distance from $x_n$ is strictly increasing along the path while the trust region model $m_n$ is strictly decreasing. It turns out, however, that this is true under a more general condition. It is geometrically evident (though we will prove it in Lemma 5.4) that, if

$$(x_n^N - x_n^U) \cdot (x_n^U - x_n) > 0, \tag{19}$$

then the distance is really increasing along the dogleg path (cf. Figure 2). We can furthermore prove that, under this condition, $m_n$ is also strictly decreasing.

**Lemma 5.4.** *Suppose that $\nabla f_n^T H_n \nabla f_n > 0$ and that $H_n$ is invertible so that $x_n^U$ and $x_n^N$ are both well-defined.*

*(i) If $H_n > 0$ then, either $x_n^U = x_n^N$, or (19) holds.*

*(ii) If (19) holds then, $|x_n^U - x_n| < |x_n^N - x_n|$ and for each $\delta \in [0, |x_n^N - x_n|]$ there exists a unique $y_n^D(\delta)$ such that $|y_n^D(\delta) - x_n| = \delta$.*

*(iii) If (19) holds then $\delta \mapsto m_n(y_n^D(\delta))$ is strictly decreasing.*

***Problem 5.2.*** Prove Lemma 5.4. □

4. We now have a clear strategy how to find an approximate solution to the trust region subproblem:

$$y_{n+1} = \begin{cases} x_n^C, & \text{if } \nabla f_n^T H_n \nabla f_n \le 0 \text{ or } H_n \text{ is singular or (19)fails,} \\ \text{argmin}\{m_n(y) : y \in \mathcal{C}_n \cap R_n\}, & \text{otherwise.} \end{cases} \tag{20}$$

***Problem 5.3.*** Describe an algorithm in pseudo-code which computes the dogleg solution of the trust region subproblem. □

We would expect that, with this choice of trust region subproblem solution, the method should automatically switch from the globally convergent steepest descent method to a quadratically convergent Newton method, whenever it is convenient. The following Theorem shows that this is indeed the case:

**Theorem 5.5.** *Suppose that $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$ is bounded below, that $\nabla^2 f$ is Lipschitz, that $\sup_{x\in\mathbb{R}^N} \|\nabla^2 f(x)\| =: \beta < 0$, and that $H_n = \nabla^2 f(x_n)$ for all $n$.*

*Suppose, further, that $(x_n)_{n\in\mathbb{N}}$ is generated by the trust region algorithm 5.2 with dogleg solution for the subproblem, and that $x_n \to x_*$ where $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*) > 0$. Then $x_n \to x_*$ q-quadratically.*

*Proof.* Since $\nabla^2 f_* > 0$ we can assume that, for $n \geq n_0$,

$$\|H_n\| \leq 2\|\nabla^2 f_*\| = c_1, \quad \|H_n^{-1}\| \leq 2\|\nabla^2 f_*^{-1}\| = c_2, \quad h^T H_n h \geq c_0|h|^2.$$

In particular, the dogleg $\mathcal{C}_n$ is well-defined and $y_{n+1}$ is given by case two of (20).

From the theory of Newton's method we know that, for $n \geq n_0$,

$$|x_n^N - x_*| \leq C|x_n - x_*|^2 \leq \tfrac{1}{2}|x_n - x_*| \qquad \forall\, n \geq n_1 \geq n_0.$$

From this, we deduce that, for $n \geq n_1$,

$$c_2|\nabla f_n| \geq |H_n^{-1}\nabla f_n| = |x_n^N - x_n| \geq |x_n - x_*| - \tfrac{1}{2}|x_n - x_*| = \tfrac{1}{2}|x_n - x_*|.$$

Conversely, it is easy to show that $|\nabla f_n| \leq c_1|x_n - x_*|$. Finally, we recall that

$$|m_n(y) - f(y)| \leq \tfrac{1}{6}L|y - x_n|^3 \qquad \forall y \in R_n.$$

We now distinguish two cases.

*Case (i).* If $\Delta_n \geq \frac{3}{2}|x_n - x_*|$ then, since $|x_n^N - x_n| \leq \frac{3}{2}|x_n - x_*|$, it follows that $y_{n+1} = x_n^N$ for which we can write the *predicted decrease* as

$$m_n(x_n) - m_n(x_n^N) = \tfrac{1}{2}\nabla f_n^T H_n^{-1}\nabla f_n \geq \tfrac{1}{2}c_0|\nabla f_n|^2,$$

and therefore deduce

$$\rho_n \geq 1 - \frac{1}{3Lc_0}\frac{|x_n^N - x_n|^3}{|\nabla f_n|^2}.$$

Using the fact that $|x_n^N - x_n| \leq \frac{3}{2}|x_n - x_*| \leq 3c_2|\nabla f_n|$, we obtain

$$\rho_n \geq 1 - C|x_n - x_*|.$$

In particular, there exists $n_2 \geq n_1$ such that $\rho_n \geq \frac{1}{4}$ for all $n \geq n_2$ for which *Case (i)* occurs.

*Case (ii).* If $\Delta_n \leq \frac{3}{2}|x_n - x_*|$ then we also have $\Delta_n \leq 3c_2|\nabla f_n|$. We now estimate

$$m_n(x_n) - m_n(y_{n+1}) \geq m_n(x_n) - m_n(x_n^C) \geq m_n(x_n) - m_n(\tilde{x}_n^C),$$

where $\tilde{x}_n^C = x_n - \tilde{\Delta}_n/|\nabla f_n|\nabla f_n$ lies on the segment between $x_n$ and $x_n^C$. We pick $\tilde{\Delta}_n = \min(\Delta_n, |\nabla f_n|/c_2)$, so that Lemma 5.2 (i) and (ii) apply with $\Delta_n = \tilde{\Delta}_n$ (and in particular, $|x_n^C - x_n| \geq |\tilde{x}_n^C - x_n|$), yielding

$$\rho_n \geq 1 - \frac{\frac{1}{6}L\Delta_n^3}{\frac{1}{2}\tilde{\Delta}_n|\nabla f_n|} \geq 1 - C\frac{\Delta_n|\nabla f_n|}{\min(\Delta_n, |\nabla f_n|/c_2)},$$

where $C = 3c_2^2 L$. From this, it clearly implies that there exists $n_3 \geq n_2$ such that $\rho_n \geq \frac{1}{4}$ for all $n \geq n_3$ for which *Case (ii)* occurs.

In conclusion, we have shown that, for $n \geq n_3$, $y_{n+1}$ is always accepted, and moreover that $\Delta_n \geq \Delta_{n_3}$ for all $n \geq n_3$. This shows that for $n$ sufficiently large, we are always in *Case (i)* and therefore $x_{n+1} = x_n^C$. Hence, the dogleg method reduces to Newton's method, and therefore its convergence is locally q-quadratic. □

# 6 Quasi-Newton Methods

Let us recall the variable-metric steepest descent method from Section 4.3. The goal of the present section is to present a powerful, general, and popular principle for constructing the matrices $B_n$. We will then also apply this principle for computing approximate Hessians $H_n$ for the trust region method.

## 6.1 The Dennis–Moré Condition for Superlinear Convergence

We begin by investigating the proof of quadratic convergence of Newton's method in order to establish conditions on the metrics $B_n$ under which we might recover similar convergence rates. We assume throughout this section that $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$ and that $\nabla^2 f$ is Lipschitz continuous.

Suppose that, for all $n$, the iteration

$$x_{n+1} = x_n - B_n^{-1}\nabla f(x_n)$$

is well-defined and $x_n \to x_*$ where $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*) > 0$. In this case, for $n$ sufficiently large, the Newton points $x_n^N = x_n - \nabla^2 f_n^{-1}\nabla f_n$ are also well-defined and we know from the theory in Section 3 that

$$|x_n^N - x_*| \leq C|x_n - x_*|^2 \qquad \forall n \geq n_1.$$

Let $s_n := -B_n^{-1}\nabla f_n = x_{n+1} - x_n$, then

$$x_n^N - x_{n+1} = (x_n^N - x_n) - s_n = -\nabla^2 f_n^{-1}\nabla f_n - s_n = \nabla^2 f_n^{-1}(B_n - \nabla^2 f_n)s_n.$$

Since $\nabla^2 f_n^{-1}$ is uniformly bounded, for $x_n$ near $x_*$, we can use this to estimate

$$|x_{n+1} - x_*| \leq |x_n^N - x_*| + |x_n^N - x_{n+1}| \leq C\big(|x_n - x_*|^2 + |(B_n - \nabla^2 f_n)s_n|\big). \tag{21}$$

From this we can see that, if the speed of convergence of $|(B_n - \nabla^2 f_n)s_n|$ to zero will crucially affect the convergence rate of $(x_n)_{n\in\mathbb{N}}$. This is made precise in the following theorem.

**Theorem 6.1.** *Let $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$, $x_* \in \mathbb{R}^N$ s.t. $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*) > 0$. Let $(B_n)_{n\in\mathbb{N}}$ be a family of invertible matrices and let $(x_n)_{n\in\mathbb{N}}$ be generated by the iteration $x_{n+1} = x_n - B_n^{-1}\nabla f(x_n)$. Furthermore, assume that $x_n \to x_*$ as $n \to \infty$.*

*Under these conditions, the convergence is superlinear if, and only if, the* Dennis–Moré condition

$$\lim_{n\to\infty} \frac{|(B_n - \nabla^2 f_n)s_n|}{|s_n|} = 0, \tag{22}$$

*where $s_n = x_{n+1} - x_n$, is satisfied.*

*Proof.* We only prove sufficiency of (22), necessity of (22) is left as an exercise.

First, we note that, if $\nabla^2 f$ is only continuous but not Lipschitz then, instead of $C|x_n - x_*|^2$ we would have obtained $\epsilon_n|x_n - x_*|$ where $\epsilon_n \to 0$ in (21). Hence, (21) and (22) together give

$$|x_{n+1} - x_*| = o(|x_n - x_*|) + o(|s_n|).$$

From this, we derive

$$|s_n| \leq |x_{n+1} - x_*| + |x_n - x_*| = o(|x_n - x_*|) + o(|s_n|) + |x_n - x_*|.$$

Since $x_n \to x_*$, $|s_n| \to 0$ and hence, it follows that $|s_n| \leq 2|x_n - x_*|$ for sufficiently large $n$. Hence, we can now conclude that

$$|x_{n+1} - x_*| = o(|x_n - x_*|),$$

which is precisely the required superlinear convergence. □

***Problem 6.1.*** Complete the proof of Theorem (22), by showing that (22) is also necessary for superlinear convergence. □

**Notation:** For the remainder of the section on quasi-Newton methods, we assume that $(x_n)_{n\in\mathbb{N}}$ is generated by a linesearch method with search directions $s_n = -B_n^{-1}\nabla f_n$ and steplengths $\alpha_n$. Furthermore, we denote $y_n = \nabla f_{n+1} - \nabla f_n$ and $d_n = x_{n+1} - x_n = \alpha_n s_n$. Note that, if $\alpha_n = 1$ as we assumed in the present section then $s_n = d_n$. Note also that, if $\alpha_n \neq 1$, then (22) is equivalent to

$$\lim_{n\to\infty} \frac{|(\nabla^2 f_n - B_n)d_n|}{|d_n|} = 0.$$

## 6.2 The secant condition and quasi-Newton updates

Note, that the Dennis–Moré condition does not require that $\|B_n - \nabla^2 f_n\| \to 0$ to achieve superlinear convergence, but only that their action in the direction $d_n = x_{n+1} - x_n$ decays sufficiently rapidly. Let us next note that

$$\begin{aligned}\nabla^2 f_n d_n = &\int_0^1 \nabla^2 f(x_n + td_n)d_n \,\mathrm{d}t + \int_0^1 \big(\nabla^2 f_n - \nabla^2 f(x_n + td_n)\big)d_n \,\mathrm{d}t \\ = &\big(\nabla f(x_{n+1}) - \nabla f(x_n)\big) + O(|d_n|^2).\end{aligned}$$

Hence, the ideal situation $B_n d_n \nabla^2 f_n d_n$ can be written as

$$B_n d_n = y_n + O(|d_n|^2), \qquad \text{where } y_n = \quad \nabla f_{n+1} - \nabla f_n,$$

and hence $B_n d_n = y_n$ might be a desirable condition. Of course this is difficult to enforce directly as $B_n$ would then depend on $x_{n+1}$. Instead, we require that the next *update* $B_{n+1}$ satisfies if,

$$B_{n+1} d_n = y_n. \tag{23}$$

This condition is called the *secant condition* since it generalizes the one-dimensional secant method. For example, if the sequence of metrics $(B_n)_{n\in\mathbb{N}}$ satisfy (23) as well as $\|B_{n+1} - B_n\| \to 0$, then

$$B_n d_n = y_n + (B_n - B_{n+1}) d_n,$$

from which the Dennis–Moré condition can be quickly established, and hence, the resulting quasi-Newton method will be superlinearly convergent.

This observation motivates the Davidon–Fletcher–Powell (DFP) update. The DFP update $B_{n+1}$ is obtained by minimizing $\|B - B_n\|_W$, where $\|\cdot\|_W$ denotes a suitably weighted Frobenius norm, subject to $B^T = B$ and $B d_n = y_n$ which gives (see [4, Sec. 6.1] for further detail)

$$B_{n+1} = (I - \rho_n y_n d_n^T) B_n (I - \rho_n d_n y_n^T) + \rho_n y_n y_n^T, \quad \text{where} \quad \rho_n = \frac{1}{y_n^T d_n}. \tag{24}$$

It is easy to see that, if $y_n^T d_n > 0$ (we will see later that this can be enforced by an improved linesearch method) and $B_{n+1}$ is spd then $B_n$ is spd. Thus, using only information that we would need to compute for the steepest descent method anyhow, we can construct a quasi-Newton update which satisfies the secant condition and, intuitively, should therefore significantly improve the convergence rate. This comes at a cost of having to invert a matrix at step, however, we will see in the next section that matrices of the type (24) can be inverted quite efficiently.

Another popular update is the BFGS update (named after Broyden, Fletcher, Goldfarb, and Shanno who discovered it independently). Based on the fact that all the above observations could have been made in terms of the inverses $B_n^{-1}$ and $\nabla^2 f_n^{-1}$, it aims to find an update which minimizes $\|B_n^{-1} - B_{n+1}^{-1}\|_W$ again in a suitably weighted matrix norm [4, Sec. 6.1]. This leads to

$$B_{n+1} = B_n - \frac{(B_n d_n)(B_n d_n)^T}{d_n^T B_n d_n} + \frac{y_n y_n^T}{y_n^T d_n}. \tag{25}$$

This is the most popular quasi-Newton method, and we will revisit it again in Section 6.5. For the time being, let us only note that the DFP and DFGS updates are clearly only well-defined if $y_n^T d_n \neq 0$. We will see in Section 6.4 that this can be enforced by a more advanced line-search procedure.

### 6.3 The Sherman–Morrison–Woodbury Formula

In this section, we will present a formula that will make it easy to invert the matrices generated by quasi-Newton updates.

**Lemma 6.2 (Sherman–Morrison–Woodbury Formula).** *Let $B \in \mathbb{R}^{N\times N}$ be invertible, $U, V \in \mathbb{R}^{N\times M}$, then $B + UV^T$ is invertible if, and only if, $I + V^T B^{-1} U$ is invertible, and*

$$(B + UV^T)^{-1} = B^{-1} - B^{-1} U (I + V^T B^{-1} U)^{-1} V^T B^{-1}.$$

***Problem 6.2.*** Prove Lemma 6.2. □

We can use the SMW formula to write the inverses of quasi-Newton updates using similarly simple updating formulas.

***Problem 6.3.*** Show (assuming all terms are well-defined) that if $B_{n+1}$ is the DFP update (24) then its inverse is given by

$$B_{n+1}^{-1} = B_n^{-1} - \frac{(B_n^{-1}y_n)(B_n^{-1}y_n)^T}{y_n^T B_n^{-1} y_n} + \frac{d_n d_n^T}{y_n^T d_n}. \qquad \square$$

*Hint: invert $B_{n+1}^{-1}$ to arrive at $B_{n+1}$.*

## 6.4 The Wolfe conditions

For the DFP (24) and BFGS (25) to be well-defined and to preserve positive definiteness we require that $y_n^T d_n \neq 0$ (or equivalently $y_n^T s_n > 0$). Later on, we will even require that $y_n^T d_n > 0$. It turns out that this requirement can be achieved by a more sophisticated linesearch algorithm.

In addition to the sufficient decrease (or Armijo) condition (10) we require that the *curvature condition*

$$\nabla f(x_n + \alpha_n s_n) \cdot s_n \geq \theta_c \nabla f(x_n) \cdot s_n \tag{26}$$

is satisfied, for some $\theta_c \in (\theta_{sd}, 1)$. Together, the sufficient decrease and curvature conditions are called the *Wolfe conditions*. Intuitively, (26) prevents us from stopping the linesearch when significant further progress can be made. A typical value for $\theta_c$ is 0.9 for Newton or quasi-Newton methods.

The following lemma explains the term *curvature condition* and also demonstrates that it is sufficient for the DFP and BFGS updates to be well-defined. Below, we will see that it also ensure that these updates preserve positivity of the matrices $B_n$.

**Lemma 6.3.** *Suppose $s_n$ is a descent direction at $x_n$, and $x_{n+1}$ satisfies the curvature condition, then*

$$y_n \cdot d_n = (\nabla f_{n+1} - \nabla f_n) \cdot (x_{n+1} - x_n) > 0.$$

*Proof.* The condition (26) can be rewritten as

$$(\nabla f_{n+1} - \nabla f_n) \cdot (x_{n+1} - x_n) = (\theta_c - 1)\nabla f(x_n) \cdot (x_{n+1} - x_n).$$

Since $s_n$ is descent direction, it follows that the right-hand side is positive. □

It remains to construct a linesearch algorithm which guarantees the Wolfe conditions. To simplify the notation, let $\phi(\alpha) = f(x + \alpha s)$ in the following algorithm. The Wolfe conditions then become

$$\phi(\alpha) \leq \phi(0) + \theta_{\mathrm{sd}} \alpha \phi'(0) \quad \text{and} \quad \phi'(\alpha) \geq \theta_{\mathrm{c}} \phi'(0). \tag{27}$$

**Algorithm 6.1.** WLINESEARCH

**Input:** $x, s$ s.t. $\nabla f(x) \cdot s < 0$, $0 < \theta_{sd} < \theta_c < 1$.

$\alpha \leftarrow 1$, $\underline{\alpha} \leftarrow 0$, $\overline{\alpha} \leftarrow 0$;
**while** (27) fails **do**
  **if** $\phi(\alpha) \geq \phi(0) + \theta_{\rm sd}\alpha\phi'(0)$ **then**
    $\overline{\alpha} \leftarrow \alpha$; $\alpha \leftarrow \frac{1}{2}(\underline{\alpha} + \overline{\alpha})$;
  **else if** $\phi'(\alpha) < \theta_{\rm c}\phi'(0)$ **then**
    $\underline{\alpha} \leftarrow \alpha$;
    **if** $\overline{\alpha} = 0$ **then**
      $\alpha \leftarrow 2\underline{\alpha}$;
    **else**
      $\alpha \leftarrow \frac{1}{2}(\underline{\alpha} + \overline{\alpha})$;
    **end if**
  **end if**
**end while**
**return** $\alpha$;

Note that we are in fact seeking to impose a strict version of the Armijo condition. This makes is slightly easier to prove termination of the algorithm.

**Proposition 6.4.** *Suppose $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$ is bounded below. Then Algorithm 6.1 terminates after a finite number of iterations and returns a steplength $\alpha$ which satisfies both Wolfe conditions.*

***Problem 6.4.*** Prove Proposition 6.1: Assume, for contradiction, that the algorithm does not terminate.

(i) Show that $\underline{\alpha}$ always satisfies $\phi(\underline{\alpha}) \leq \phi(0) + \theta_{sd}\underline{\alpha}\phi'(0)$ with strict inequality if $\underline{\alpha} > 0$.

(ii) Show that $\overline{\alpha}$ cannot remain zero for all iterations.

(iii) Show that, if $\overline{\alpha} > 0$, then $\phi(\overline{\alpha}) \geq \phi(0) + \overline{\alpha}\theta_{sd}\phi'(0)$. Hence, deduce that there exists $\tilde{\alpha} \in [\underline{\alpha}, \overline{\alpha}]$ such that $\phi'(\tilde{\alpha}) > \theta_c\phi'(0)$.

(iv) Use the fact that, once $\overline{\alpha} > 0$, the interval $[\underline{\alpha}, \overline{\alpha}]$ is halved at each iteration, to arrive at a contradiction. □

***Problem 6.5.*** Under the conditions of Theorem 6.1 prove that, if $\theta_{sd} < 1/2$, then, for $n$ sufficiently large, the steplength $\alpha_n = 1$ satisfies the Wolfe conditions (10) and (26). □

## 6.5 The BFGS method

The most popular quasi-Newton update for linesearch methods is the BFGS method (25). In particular, the BFGS update is a *rank-2 update* which we can invert using the SMW formula. This as well as preservation of positivity and symmetry of the BFGS update are established in the following lemma.

**Lemma 6.5.** *Let $B_{n+1}$ be the BFGS update of $B_n$ and suppose that $B_n$ is spd and that $d_n^T y_n > 0$, then $B_{n+1}$ is spd and $B_{n+1}^{-1}$ is given by*

$$B_{n+1}^{-1} = (I - \frac{d_n y_n^T}{d_n^T y_n}) B_n^{-1} (I - \frac{y_n d_n^T}{y_n^T d_n}) + \frac{d_n d_n^T}{d_n^T y_n}. \tag{28}$$

***Problem 6.6.*** Prove Lemma 6.5. *Hint: Prove* (28) *first, using the fact that* $y_n^T d_n > 0$. *Then show that* $B_n^{-1}$ *spd implies* $B_{n+1}^{-1}$ *spd.* □

The preceding lemma shows (i) that BFGS updates preserve positivity and are therefore particularly useful for linesearch methods. Furthermore, it provides a formula for updating $B_n^{-1}$ rather than $B_n$. Thus, we can replace the expensive ($O(N^3)$ flops) matrix inversion (or LU or Cholesky factorization) by a simple matrix multiplication ($O(N^2)$ flops). In fact, to apply $B_n^{-1}$ we would only need to store the vectors $d_k, y_k$ for $k < n$, so the update (28) does not even have to be computed explicitly.

**Algorithm 6.2 (A Simple BFGS Algorithm).**
**Input:** $x_0 \in \mathbb{R}^N$, $B_0^{-1} \in \mathbb{R}^{N\times N}$ spd, $0 < \theta_{sd} < \theta_c < 1$;
1: **for** $n = 0, 1, 2, \ldots$ **do**
2: $s_n \leftarrow -B_n^{-1}\nabla f_n$;
3: $\alpha_n \leftarrow$ WLINESEARCH$[x = x_n, s = s_n, \theta_{sd}, \theta_c]$;
4: $x_{n+1} \leftarrow x_n + \alpha_n s_n$;
5: $d_n \leftarrow x_{n+1} - x_n$; $y_n \leftarrow \nabla f_{n+1} - \nabla f_n$;
6: Update $B_{n+1}^{-1}$ using (28);
7: **end for**

The facts established in Lemma 6.5 show that this BFGS method is *well-defined*, i.e., each iteration can be executed succesfully.

Unfortunately, the global convergence theory for the BFGS method is somewhat incomplete. This stems from the difficulty of proving uniform boundedness of $B_n$ from above and below. The local convergence theory, however, is very satisfactory. The following results, taken from [4, Sec. 6.4] establishes local superlinear convergence and global convergence provided the objective function is *uniformly convex*.

**Theorem 6.6 (Superlinear Convergence of BFGS).** *Suppose* $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$ *and that there exist constants* $0 < m \le M < \infty$ *such that*

$$m|h|^2 \le h^T\nabla^2 f(x)h \le M|h|^2 \qquad \forall x, h \in \mathbb{R}^N. \tag{29}$$

*Then, for any* $x_0 \in \mathbb{R}^N$ *and spd* $B_0$, *the iterates* $(x_n)_{n\in\mathbb{N}}$ *generated by the BFGS algorithm 6.2 converge superlinearly to the unique minizer of* $f$ *in* $\mathbb{R}^N$.

*If* $f$ *does not satisfy* (29), *but if* $x_* \in \mathbb{R}^N$ *with* $\nabla f(x_*) = 0$ *and* $\nabla^2 f(x_*) > 0$ *then the conclusion remains true provided* $x_0$ *is sufficiently close to* $x_*$.

## 6.6 The SR1 update

Since the following idea is typically used in connection with trust region rather than line search methods, we change our notation from $B_n$ to $H_n$ (a hessian instead of a metric).

A much simpler idea than the DFP and BFGS updates is to require that $H_{n+1}$ is a symmetric rank-1 (SR1) update of $H_n$, i.e., $H_{n+1} = H_n + \sigma vv^T$, where $\sigma \in \{-1, 1\}$ and $v \in \mathbb{R}^N$. (Note that, by comparison, BFGS is a rank-2 update.) It turns out that this leads

to the unique formula (see Problem 6.7)

$$H_{n+1} = H_n + \frac{(y_n - H_n d_n)(y_n - H_n d_n)^T}{(y_n - H_n d_n)^T d_n}. \tag{30}$$

The SR1 update has several shortcomings: it is undefined when $(y_n - H_n d_n)^T d_n = 0$ and there are issues with numerical instability if $(y_n - H_n d_n)^T d_n$ is very small. Furthermore, $H_{n+1}$ need not be positive even if $H_n$ is positive. It is nevertheless a popular update for trust region methods where positivity of the Hessian approximation may not even be desired. There, it has the clear advantage over positive definite updates that it can approximate indefinite Hessians.

***Problem 6.7 (The SR1 Update).***

(a) Show that the following are necessary conditions for the SR1 update $H_{n+1} = H_n + \sigma v v^T$ where $\sigma \in \{1, -1\}, v \in \mathbb{R}^N$, to satisfy the secant equation (assuming all terms are well-defined):

$$\sigma = \operatorname{sign}[d_n^T(y_n - H_n d_n)] \quad \text{and} \quad v = \delta(y_n - H_n d_n) \quad \text{where} \quad \delta = \pm |d_n^T(y_n - H_n d_n)|^{-1/2}.$$

Hence deduce that (30) is the only symmetric rank-1 update which satisfies the secant equation (23).

(b) Show that, if $y_n = H_n d_n$ then $H_{n+1} = H_n$ is the only symmetric rank-1 update satisfying the secant condition, and, if $d_n^T(y_n - H_n d_n) = 0$ but $y_n \neq H_n d_n$ then there exists not SR1 update satisfying the secant equation.

(c) Explain why, as $d_n^T(y_n - H_n d_n) \to 0$, you would except numerical instability. Combined with part (b), motivate the following safe-guarded SR1 update: fix $\eta > 0$, then update $H_{n+1} = H_n$ if $|d_n^T(y_n - H_n d_n)| < \eta$ and let $H_{n+1}$ be given by (30) otherwise.

(d) Use the SMW formula to show that, if $H_n$ is invertible and $d_n^T(y_n - H_n d_n) \neq 0$, then $H_{n+1}$ is invertible and the following update for $H_{n+1}^{-1}$ holds

$$H_{n+1}^{-1} = H_n^{-1} + \frac{(d_n - H_n^{-1} y_n)(d_n - H_n^{-1} y_n)^T}{(d_n - H_n^{-1} y_n)^T y_n}.$$

(e) Formulate a dogleg algorithm (for a trust region method) which uses approximate Hessians obtained by a safe-guarded SR1 update. □

# 7 Optimality Conditions for Constrained Optimization

Having established a broad and satisfactory theory for unconstrained optimization, we now turn to methods for constrained optimization. Beforehand, we will spend some effort to study the structure of the admissible sets, of the constraints, and of the solutions. Most importantly, we will derive the counterparts of the optimality conditions in Section 2.4. We will also begin a discussion of the method of Lagrange multipliers and Newton's method for equality-constrained optimization problems.

## 7.1 A basic optimality condition

In this section, we will derive and formulate 'basic' optimality conditions which will then motivate the study of the tangent space and the method of Lagrange multipliers.

Recall the general optimisation problem

$$\min_{\Omega} f(x), \tag{31}$$

where $\Omega$ is the admissible set,

$$\Omega = \{x \in \mathbb{R}^N : c_j(x) = 0, j \in \mathcal{I}_e, c_j(x) \geq 0, j \in \mathcal{I}_i\},$$

and where $c \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R}^{M_e+M_i})$.

Suppose now that $x_*$ is a local minimizer of $f$ in $\Omega$ and that there exists an *admissible path* $\gamma \in \mathrm{C}^2([0,\varepsilon); \Omega)$ (by this, we mean that $\gamma$ is continuously differentiable in $(0,\varepsilon)$ with bounded derivative $\dot{\gamma}$) with $\gamma(0) = x_*$, then

$$\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t))|_{t=0} = \nabla f(x_*) \cdot \dot{\gamma}(0) \geq 0.$$

Loosely speaking, this condition states that there are no descent directions for $f$ leading from $x_*$ *into* the admissible set $\Omega$. The condition that $\gamma$ is twice differentiable may be restrictive in some instances (e.g., Nocedal and Wright [4] even consideradmissible sequences) but sufficient for most puposes as we will discover in the next section.

Clearly the question arises to characterize the set of all vectors $d \in \mathbb{R}^N$ for which we can construct such a curve $\gamma$ with $\dot{\gamma}(0) = d$. We will take up this task in the next section. In the meantime, we can nevertheless define the *tangent cone* to $\Omega$ at $x_*$ in an abstract way.

**Definition 7.1 (Tangent Cone).** *Let $x \in \Omega$, then the* tangent cone *of $\Omega$ at $x$ is the set $T_\Omega(x)$ containing all vectors $d \in \mathbb{R}^N$ for which there exists an admissible surve $\gamma$ such that $\gamma(0) = x$ and $\dot{\gamma}(0) = d$.*

This quite general definition will admit certain pathologies which we will be careful to exclude in the following sections. For the time being, however, it provides a first straightforward generalization of the first-order optimality conditions in unconstrained optimisation.

**Proposition 7.2.** *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$, that $c \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R}^{M_e+M_i})$, and that $x_*$ is a local minimizer of $f$ in $\Omega$, then,*

$$\nabla f(x_*) \cdot d \geq 0 \qquad \forall d \in T_\Omega(x_*). \tag{32}$$

***Remark 7.3.*** The second-order optimality conditions cannot be immediately generalized. To see this note that, for $f(x) = -\frac{1}{2}|x|^2$ and $c(x) = |x|^2 - 1$ (understood as an equality constraint), we have that $f$ is constant in $\Omega$ and hence every $x \in \Omega$ is a local minimizer. However, it is clear that

$$h^T \nabla^2 f(x) h = -|h|^2 < 0 \qquad \forall h \in T_\Omega(x) \setminus \{0\}$$

Moreover, if we chose $f(x) = \frac{1}{2}|x|^2$ and $c$ as before, then once again, $f$ is constant in $\Omega$ and so the fact that

$$h^T \nabla^2 f(x) h = |h|^2 > 0 \qquad \forall h \in T_\Omega(x) \setminus \{0\}$$

cannot be used to decude that these points are strict local minimizers. □

## 7.2 The tangent cone

Although the criticality condition derived in the previous section is fairly general in that it is the immediate generalization of the first-order optimality condition for unconstrained optimisation to the constrained case, it gives us no analytical tools to study critical points. Hence, it is crucial that we give a precise characterisation of the tangent cone. It turns out that in many important situations, the tangent cone $T_\Omega(x)$ can be characterized in terms of $Dc(x)$. This will lead immediately to the KKT conditions and to the method of Lagrange multipliers.

For $x \in \Omega$, let us first define the set of active constraints (or simply the *active set*),

$$\mathcal{A}(x) = \big\{ j \in \{1, \dots, M_e + M_i\} : c_j(x) = 0 \big\}.$$

For example, that $\mathcal{E} \subset \mathcal{A}$. The *inactive set* is defined as

$$\mathcal{A}'(x) = \{1, \dots, M_e + M_i\} \setminus \mathcal{A}(x) = \big\{ j \in \{M_e + 1, \dots, M_e + M_i\} : c_j(x) > 0 \big\}.$$

Since $c$ is continuous, it follows that $\mathcal{A}'(x) \subset \mathcal{A}'(y)$ for all $y$ in a neighbourhood of $x$. This means that, in essence, the inactive constraints $c_j(x) > 0$, $j \in \mathcal{A}'(x)$ can simply be ignored.

A crucial technical condition which we will employ throughout the theory of constrained optimisation is the following. We can think about it (though strictly speaking this is false) that, locally, no active constraint can be expressed in terms of the other active constraints.

**Definition 7.4.** *Suppose $c \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R}^{M_e+M_i})$. We say that the* linear independence constraint qualification *(LICQ) holds at a point $x$ if the set $\{\nabla c_j(x) : j \in \mathcal{A}(x)\}$ is linearly independent.*

*If we define $c_{\mathcal{A}}(x) = (c_i(x))_{i \in \mathcal{A}(x)}$, then the LICQ is equivalent to the requirement that $Dc_{\mathcal{A}}(x)$ has full rank.*

The following Lemma provides a characterisation of the tangent cone in terms of the *set of linearized admissible directions* $\mathcal{F}(x)$.

**Lemma 7.5.** *Let $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^{M_e+M_i})$, and $x \in \Omega$, then*

$$\begin{aligned} T_\Omega(x) \subset \big\{ d \in \mathbb{R}^N : \ & \nabla c_j(x) \cdot d = 0 \quad \forall j \in \mathcal{E}, \text{ and} \\ & \nabla c_j(x) \cdot d \geq 0 \quad \forall j \in \mathcal{I} \cap \mathcal{A}(x) \big\} =: \mathcal{F}(x). \end{aligned}$$

*If, the LICQ hold at $x$, then $T_\Omega(x) = \mathcal{F}(x)$.*

*Proof.* We assume, without loss of generality, that $\mathcal{A} = \mathcal{E} \cup \mathcal{I}$.

The inclusion $T_\Omega(x) \subset \mathcal{F}(x)$ is quickly established. Namely, if $d \in T_\Omega(x)$ then for an admissible curve $\gamma \in \mathrm{C}^2([0,\varepsilon);\Omega)$ with $\gamma(0) = x$ and $\dot\gamma(0) = d$, we have $c_j(\gamma(t)) = 0$ for $j \in \mathcal{E}$ and $c_j(\gamma(t)) \geq 0$ for $j \in \mathcal{I}$, and hence, using also the fact $c_j(x) = 0$ for all $j \in \mathcal{A}$,

$$\lim_{t\to 0} \frac{c_j(\gamma(t)) - c_j(x)}{t} = \nabla c_j \cdot d \begin{cases} = 0, & j \in \mathcal{E}, \\ \geq 0, & j \in \mathcal{I} \cap \mathcal{A}. \end{cases}$$

To prove the reverse inclusion we will construct an admissible curve using the Implicit Function Theorem. Since $Dc(x) = Dc_{\mathcal{A}}(x)$ has full rank we assume, without loss of generality, that the first $M := M_e + M_i$ columns of $Dc$ are linearly independent (again, upon re-ordering the coordinates).

Let $N = \ker(Dc(x))$, let $d \in \mathcal{F} \setminus \{0\}$, and define $R : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$,

$$R(z,t) = \begin{pmatrix} c(z) - tDc(x)d \\ \Pi_N(z - x - td) \end{pmatrix},$$

where $\Pi_N \in \mathbb{R}^{N-M\times N}$ is a matrix containing an orthonormal basis of $N$ in its rows. Then $R(x,0) = 0$ and

$$D_z R(x,0) = \begin{pmatrix} Dc(x) \\ \Pi_N \end{pmatrix}$$

is invertible. To see this, simply note that $Dc(x)$ has full rank, and that $Dc(x)\Pi_N = 0$.

Hence, the Inverse Function Theorem provides us with a function $\gamma \in \mathrm{C}^1(-\varepsilon,\varepsilon);\mathbb{R}^N)$ so that $\gamma(0) = x$ and $R(\gamma(t),t) = 0$ for all $t \in (-\varepsilon,\varepsilon)$. Moreover,

$$0 = D_z R(x,0)\dot\gamma(0) + D_t R(x,0) = \begin{pmatrix} Dc(x) \\ \Pi_N \end{pmatrix} \dot\gamma(0) - \begin{pmatrix} Dc(x)d \\ \Pi_N d \end{pmatrix},$$

from which we immediately see that $\dot\gamma(0) = d$. To show that $\gamma(t)$ is admissible, we note that

$$0 = R(\gamma(t),t) = \begin{pmatrix} c(\gamma(t)) - tDc(x)d \\ \Pi_N(\gamma(t) - x - td) \end{pmatrix}.$$

Hence, for all $i \in \mathcal{E} \cup \mathcal{I}$, we have

$$c_j(\gamma(t)) = tDc(x)d \begin{cases} = 0, & j \in \mathcal{E} \\ \geq 0, & j \in \mathcal{I}, t \geq 0. \end{cases}$$

Since $R$ is twice-differentiable in $z$ and $t$ is follows that $\gamma$ is also twice differentiable in a neighbourhood of 0. Hence, $\gamma$ is admissible and $d \in T_\Omega(x)$. □

**Corollary 7.6.** *In the proof of Lemma 7.5 we have shown that, if $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^{M_e+M_i})$ and if $x$ satisfies the LICQ, then, for any $d \in T_\Omega(x)$, there exists a path $\gamma \in \mathrm{C}^2((-\varepsilon,\varepsilon);\Omega)$ such that $c_j(\gamma(t)) = t\nabla c_j(x) \cdot d$ for all $j \in \mathcal{A}(x)$.*

### 7.3 The Karush–Kuhn–Tucker conditions

Having characterised the set of admissible directions (the tangent cone $T_\Omega(x)$) in terms of a set of linear conditions ( the linearized cone $\mathcal{F}(x)$), it will now be fairly straightforward to derive 'useful' optimality conditions for constrained optimisation problems. We assume throughout that $x_*$ is a local minimizer of $f$ in $\Omega$ and that the LICQ hold at $x_*$.

Let us assume for the moment, that we have only equality conditions, i.e., $M_i = 0$. Then Proposition 7.2 shows that $\nabla f(x_*) \in T_\Omega(x_*)^\perp$. This can be generalized to the case of inequality constraints where the follows set of conditions are called the *Karush–Kuhn–Tucker (KKT) conditions.*

**Theorem 7.7.** *Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$, $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^{M_e+M_i})$ and let $x_*$ be a local minimizer of $f$ in $\Omega$ at which the LICQ hold.*

*Then there exists $\lambda_* \in \mathbb{R}^{M_e+M_i}$ such that*

$$\nabla f(x_*) = \sum_{j=1}^{M_e+M_i} \lambda_*^{(j)} \nabla c_i(x_*), \tag{33a}$$

$$c_j(x_*) = 0 \qquad j \in \mathcal{E} \tag{33b}$$

$$c_j(x_*) \geq 0 \qquad j \in \mathcal{I} \tag{33c}$$

$$\lambda_*^{(j)} \geq 0 \qquad \forall j \in \mathcal{I} \tag{33d}$$

$$\lambda_*^{(j)} c_j(x_*) = 0 \qquad \forall j \in \mathcal{E} \cup \mathcal{I}. \tag{33e}$$

Before we turn to the proof of Theorem 7.7, we should comment on the geometric meaning of the KKT conditions (apart from (33b) and (33c) which are obvious). The main observation is that, the sets $\mathcal{M}_j = \{c_j(x) = 0\}$ are hypersurfaces and that the gradient $\nabla c_j(x)$ denotes a normal vector to $\mathcal{M}_j$. In the case of $j \in \mathcal{I}$, $\nabla c_j(x)$ points *outside* the admissible set $\Omega$ (in the case $j \in \mathcal{E}$, this is trivial since both sides are on the *outside*). Hence (33a) and (33d) are simply translating condition (32) that $\nabla f(x_*)$ has no component pointing inside the admissible set. Finally, (33e) states that, if $c_j(x_*) > 0$ for some $j \in \mathcal{I}$ then $\lambda_*^{(j)}$ must be zero, i.e., this constraint is simply irrelevant to the problem (at least in a neighbourhood of the solution).

*Proof of Theorem 7.7.* After setting $\lambda_*^{(j)} = 0$ for all $j \in \mathcal{A}'(x_*)$, we can assume, without loss of generality, that $\mathcal{A}(x_*) = \mathcal{E} \cup \mathcal{I}$.

We will show that $\nabla f(x_*) \in \ker Dc(x_*)^\perp$. To this end, let $d \in \ker Dc(x_*)$, i.e., $\nabla c_j(x_*) \cdot d = 0$ for all $j$. In that case, Corollary 7.6 provides $\gamma(t) \in \mathrm{C}^2((-\varepsilon,\varepsilon);\mathbb{R}^N)$ such that $\gamma(0) = x_*$, $\dot\gamma(0) = d$, and

$$c_j(\gamma(t)) = t \nabla c_j(x_*) \cdot d = 0.$$

Since $t = 0$ is a minimizer of $f(\gamma(t))$ it follows that, $\nabla f_* \cdot d = 0$.

Next, a simple comparison of dimensions shows that $\ker Dc(x_*)^\perp = \operatorname{span}\{\nabla c_j(x_*)\}$. In particular, there exists a vector $\lambda \in \mathbb{R}^{M_e+M_i}$ of coefficients for $\nabla f(x_*)$ in that basis.

It only remains to show that $\lambda^{(j)} \geq 0$ for $j \in \mathcal{I}$. Let $d \in \mathbb{R}^N$ so that $\nabla c_j(x_*) \cdot d > 0$ but $\nabla c_k(x_*) \cdot d = 0$ for $k \neq j$. This can, for example be achieved by an orthogonalization

procedure. Then $d \in T_\Omega(x_*)$ and hence

$$0 \leq \nabla f(x_*) \cdot d = \lambda^{(j)} \nabla c_j(x_*) \cdot d.$$

Since $\nabla c_j(x_*) \cdot d > 0$, it follows that $\lambda^{(j)} \geq 0$. □

## 7.4 The method of Lagrange multipliers

The *Lagrangian* associated with the bound-constrained optimisation problem (31) is the functional $\mathcal{L} \in \mathrm{C}^1(\mathbb{R}^N \times \mathbb{R}^{M_e+M_i}; \mathbb{R})$,

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{j=1}^{M_e+M_i} \lambda^{(j)} c_j(x) = f(x) - \lambda^T c(x). \tag{34}$$

The KKT conditions provide a recipe for computing solutions to constrained optimisation problems. This is particularly easy when no inequality constraints are present. In that case, we only need to find the critical points of $\mathcal{L}$, i.e., we need to solve the nonlinear system

$$\nabla_{x,\lambda} \mathcal{L}(x, \lambda) = \begin{pmatrix} \nabla f(x) - \nabla c(x) \lambda \\ -c(x) \end{pmatrix} = 0, \tag{35}$$

and then identify the local minimizers. Note that the number of degrees of freedom and the number of equations in (35) are the same, and hence we may expect that the system is well-posed. This made precise in Problem 7.3 below.

In the case where inequality constraints are present no such simple method exists. In essence, one has to distinguish several cases taking each Lagrange multiplier for an inequality constraint to be either zero or positive. However, we can at least generalize the second-order sufficient optimality conditions to check whether a point is a local minimizer. To this end, we first define the *critical cone* $\mathcal{C}(x_*, \lambda_*)$ for any pair $(x_*, \lambda_*)$ which satisfies the KKT conditions,

$$\mathcal{C}(x_*, \lambda_*) = \left\{ d \in T_\Omega(x) : \lambda_*^{(j)} \nabla c_j(x_*) \cdot d = 0 \quad \forall j \in \mathcal{E}(x_*) \cup \mathcal{I}(x_*) \right\}. \tag{36}$$

To motivate this definition, see the following problem. Note also that $\mathcal{C}(x_*, \lambda_*) = T_\Omega(x_*)$ in the case of equality constraints.

***Problem 7.1.*** Suppose that the pair $(x_*, \lambda_*)$ satisfies the KKT conditions. Let $d \in T_\Omega(x_*) \setminus \mathcal{C}(x_*, \lambda_*)$, i.e., there exists $j \in \mathcal{E} \cup \mathcal{I}$ such that $\lambda_*^{(j)} \nabla c_j(x_*) \cdot d \neq 0$. Show that for any admissible path $\gamma$ with $\gamma(0) = x_*$ and $\dot\gamma(0) = d$, $\frac{\mathrm{d}}{\mathrm{d}t} f(\gamma(t))|_{t=0} > 0$. □

**Theorem 7.8 (Second order optimality conditions).** *Suppose that $f \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R})$, $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^{M_e+M_i})$ and that $x_* \in \mathbb{R}^N$ satisfies the LICQ.*

*(a) If $x_*$ is a local minimizer of $f$ in $\Omega$ then, in addition to the KKT system (33), it satisfies*

$$d^T \nabla_x^2 \mathcal{L}(x_*, \lambda_*) d \geq 0 \qquad \forall d \in \mathcal{C}(x_*, \lambda_*).$$

*(b) If $x_*$ satisfies the KKT system (33) as well as*

$$d^T \nabla_x^2 \mathcal{L}(x_*, \lambda_*) d > 0 \qquad \forall d \in \mathcal{C}(x_*, \lambda_*) \setminus \{0\}, \tag{37}$$

*then it is a strict local minimizer of $f$ in $\Omega$.*

*Proof.* Part (a) is left as an exercise.

To prove part (b), the sufficiency condition, let $x_k \in \Omega$, $x_k \to x_*$ and assume, for contradiction, that $f(x_k) \leq f(x_*)$ for all $k$. The sequence $d_k = (x_k - x_*)/|x_k - x_*|$ lies on the unit sphere and hence we may assume without loss of generality that $d_k \to d$ as $k \to \infty$ (upon extracting a convergent subsequence and dropping the subscripts). It is fairly straightforward to check that $d \in \mathcal{F}(x_*)$. Setting $t_k = |x_k - x_*|$ and expanding $f$ near $x_*$, we have

$$d(x_*) \geq f(x_k) = f_* + t_k \nabla f_* \cdot d_k + O(t_k^2)$$

and hence $\nabla f_* \cdot d \leq 0$. On the other hand, since the pair $(x_*, \lambda_*)$ satisfies the KKT conditions,

$$0 \geq \nabla f_* \cdot d = \sum_{j=1}^{M_e+M_i} \lambda_*^{(j)} \nabla c_j(x_*) \cdot d.$$

By definition of $\mathcal{F}(x_*)$, we see that each part of the sum is non-negative, which can only be true if $\lambda_*^{(j)} \nabla c_j(x_*) \cdot d = 0$ for all $j$, i.e., if $d \in \mathcal{C}(x_*, \lambda_*)$. Assumption (37) therefore implies that $d^T \nabla_x^2 \mathcal{L}_* d > 0$.

Combining all foregoing observations and the KKT conditions, we obtain

$$\begin{aligned} f(x_*) \geq\ & f(x_k) \overset{\text{KKT}}{\geq} f(x_k) - \sum_{j=1}^{M_e+M_i} \lambda_*^{(j)} c_j(x_k) = \mathcal{L}(x_k, \lambda_*) \\ =\ & \mathcal{L}(x_*, \lambda_*) + t_k \nabla_x \mathcal{L}_* \cdot d_k + \tfrac{1}{2} t_k^2 d_k^T \nabla_x^2 \mathcal{L}_* d_k + o(t_k^2) \\ =\ & f(x_*) + \tfrac{1}{2} t_k^2 d_k \nabla_x^2 \mathcal{L}_* d_k + o(t_k^2), \end{aligned}$$

or equivalently, $d_k^T \nabla_x^2 \mathcal{L}_* d_k \leq o(1)$ which, upon taking the limit as $k \to \infty$, gives a contradiction. □

***Problem 7.2.*** Prove Theorem 7.8 (a). □

***Problem 7.3.***

(a) Let $A \in \mathbb{R}^{M \times N}$ have full rank, and let $H \in \mathbb{R}^{N \times N}$ be positive definite in $\ker(A)$. Further, let $b \in \mathbb{R}^M$ and $g \in \mathbb{R}^N$. Show that the *quadratic program*

$$\min_{Ax=b} \tfrac{1}{2} x^T H x - x^T g \tag{38}$$

has at least one solution. *Hint: show that $f$ is coercive in $\Omega$.*

(b) Show that the KKT conditions for (38) can be written (in block matrix form) as

$$\begin{pmatrix} H & -A^T \\ -A & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} g \\ b \end{pmatrix}. \tag{39}$$

Show that the system matrix in (39) (the *KKT matrix*) is invertible and deduce that (38) has exactly one solution.

(c) Suppose that $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$, $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^{M_e})$ (understood as an equality constraint) and that $\nabla^2 f$ and $\nabla^2 c_j$ $(j = 1,\dots,M_e)$ are Lipschitz continuous. Moreover, suppose that the pair $(x_*,\lambda_*)$ satisfies the LICQ, the KKT conditions, and the strong second-order optimality condition

$$d^T\nabla^2\mathcal{L}(x_*,\lambda_*)d \geq c_0|d|^2 \qquad \forall d \in T_\Omega(x_*) = \mathcal{C}(x_*,\lambda_*).$$

where $c_0 > 0$. Prove that Newton's method for the nonlinear system $\nabla_{x,\lambda}\mathcal{L} = 0$ is q-quadratically convergent, provided the initial guess $(x_0,\lambda_0)$ is sufficiently close to $(x_*,\lambda_*)$. □

To conclude, we discuss an example how the method of Lagrange multipliers can be used to solve inequality constrained problems.

***Example 7.9 (Example 12.8 [4]).*** We wish to minimize

$$-\varepsilon(x-4)^2 + y^2 \quad \text{subject to} \quad x^2 + y^2 \geq 1,$$

where $\varepsilon < 1/3$. Let $f(x,y) = -\varepsilon(x-4)^2 + y^2$ and $c(x,y) = x^2 + y^2 - 1$.

This is a highly non-convex optimisation problem with several critical points. Moreover, $f(x,0) \to -\infty$ as $x \to \pm\infty$, hence there is no global minimizer.

To compute local solutions, we first derive the KKT system:

$$\begin{aligned} -2\varepsilon(x-4) &= 2\lambda x \\ 2y &= 2\lambda y \\ c(x,y) &\geq 0 \\ \lambda &\geq 0 \\ \lambda c(x,y) &= 0. \end{aligned}$$

We distinguish two cases. 1. If $c(x,y) > 0$ then $\lambda = 0$ and hence $x = 4, y = 0$. This is an admissible state, however, the Hessian

$$\nabla^2\mathcal{L}(x,y) = \begin{pmatrix} -2\varepsilon - 2\lambda & 0 \\ 0 & 2(1-\lambda) \end{pmatrix}$$

is a saddle point at $(x,y,\lambda) = (4,0,0)$.

2. If $c(x,y) = 0$ then $y^2 = 1 - x^2$ and either $y = 0$ or $y \neq 0$ and $\lambda = 1$.

2.1 If $y \neq 0$ then $\lambda = 1$ then

$$\nabla^2\mathcal{L}(x,y,\lambda) = \begin{pmatrix} -2.2 & 0 \\ 0 & 0 \end{pmatrix},$$

and since $y \neq 0$ in this case (this is easily established), the only tangent direction if $(y,-x)$ in which the hessian is negative. Hence, this point is not a local minimizer.

2.2 If $y = 0$ then the condition $c(x,y) = 0$ gives $x = 1$ and the $\partial_x\mathcal{L} = 0$ gives $\lambda = 3\varepsilon$ and we obtain

$$\nabla^2\mathcal{L}(x,y,\lambda) = \begin{pmatrix} -8\varepsilon & 0 \\ 0 & 2(1-3\varepsilon) \end{pmatrix},$$

Further, $\nabla c(x,y) = (2x, 2y) = (2,0)$. The only direction which is orthogonal is $(0,-1)$ along which $\nabla^2\mathcal{L}(1,0,3\varepsilon)$ is positive. Hence, $(x,y,\lambda) = (1,0,3\varepsilon)$ is the only local minimizer of $f$ in $\Omega$. □

***Problem 7.4.*** We show that the KKT Theorem implies the famous *Farkas' Lemma*:
Let $C \in \mathbb{R}^{M\times N}$, and $g \in \mathbb{R}^N$, then exactly one of the following statements is true;

(i) $\exists\, \lambda \in \mathbb{R}^M : C^T\lambda = g$ and $\lambda \geq 0$.

(ii) $\exists\, d \in \mathbb{R}^N : Cd \geq 0$ and $g \cdot d < 0$.

(Here, $z \in \mathbb{R}^K, z \geq 0$ means $z^{(j)} \geq 0$ for all $j$)

(a) Prove the above form of Farkas' Lemma, for the case that $M$ has *full rank*, by first setting up a suitable optimisation problem $\min_{c(x)\geq 0} f(x)$ and then distinguishing whether $x = 0$ is a local minimizer or not.

(b) What does the Farkas Lemma mean geometrically? *Hint: geometrically, what is the admissible set $\Omega$?*

(c) Now prove the Farkas Lemma again (still for the case that $C$ has full rank), by mimicking the proof of the KKT conditions from Theorem 7.7 for the particular optimisation problem you set up above.

(d) Where does the proof break down when $M$ does not have full rank? □

# 8 Penalty and Augmented Lagrangian Methods

For simplicity, we will only consider equality constraints in the present section, i.e., $M_i = 0$ and $M = M_e$. In this case the KKT conditions simply become

$$\nabla_{x,\lambda}\mathcal{L}(x_*, \lambda_*) = 0. \tag{40}$$

We have seen in Problem 7.3 that Newton's method, applied to the system (40) 'typically' converges q-quadratically. The question remains how to construct a globally convergent scheme. The idea which is usually pursued is to replace the constrained problem (31) by a sequence of unconstrained problems where violation of the constraint $c(x) = 0$ is *penalized*.

## 8.1 The $\ell^2$-penalty method

A first idea might be to define the 'merit function'

$$\Phi(\mu; x) = f(x) + \tfrac{1}{2}\mu \sum_{j=1}^{M} |c_j(x)|^2$$

and to minimize $\Phi$ for increasing values of $\mu$ which should eventually, in the limit $\mu \to \infty$ give rise to a solution of the constrained minimization problem.

**Algorithm 8.1.**

**Input:** $x_0^S \in \mathbb{R}^N, \mu_0 > 0, \tau_0 > 0$

1: **for** $n = 0, 1, 2, \ldots$ **do**

2: Use an unconstrained optimization method, using $x_n^S$ as starting guess, to compute $x_n$ such that $|\nabla_x \Phi(\mu_n; x_n)| \leq \tau_n$;

3: Choose $\mu_{n+1}, \tau_{n+1}$; $x_{n+1}^S \leftarrow x_n$;

4: **end for**

Typically, information about 'how difficult' the minimization problem in step 2 was will enter the updating procedure for $\mu_n$ and $\tau_n$. We will not discuss this in much detail until later but note that we will always assume that $\mu_n \to \infty$ while $\tau_n \to 0$.

To analyze potential limits of this iteration, we compute the gradient of $\Phi$,

$$\nabla_x \Phi(\mu_n; x_n) = \nabla f(x_n) - \sum_{j=1}^{M} \mu_n c_j(x_n) \nabla c_j(x_n).$$

We can see therefore, that if we define

$$\lambda_n = \mu_n c(x_n), \tag{41}$$

then $\nabla_x \Phi(\mu_n; x_n) = \nabla_x \mathcal{L}(x_n, \lambda_n)$.

**Theorem 8.1.** *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$, $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^M)$, and that $x_n$ generated by Algorithm 8.1 (with $\tau_n \to 0$ and $\mu_n \to \infty$) converges to a point $x_*$ at which the LICQ hold. Then,*

*(i)* $\lim_{n\to\infty} \lambda_n =: \lambda_*$ *exists, where $\lambda_n$ is defined by (41), and*

*(ii)* $(x_*, \lambda_*)$ *is a KKT point, i.e., $\nabla_{x,\lambda} f(x_*, \lambda_*) = 0$.*

*Proof.* We begin by proving that $x_* \in \Omega$. This follows, for example, from the fact that

$$\sum_{j=1}^{M} c_j(x_n) \nabla c_j(x_n) = \mu_n^{-1} \big( \nabla_x \Phi(\mu_n; x_n) - \nabla f(x_n) \big).$$

The term inside the brackets on the right-hand side is bounded and $\mu_n \to \infty$ and hence the right-hand side tends to zero. Since $c$ and $\nabla c$ are continuous we therefore obtain

$$\sum_{j=1}^{M} c_j(x_*) \nabla c_j(x_*) = \lim_{n\to\infty} \sum_{j=1}^{M} c_j(x_n) \nabla c_j(x_n) = 0.$$

Since the LICQ holds, this is only possible if $c_j(x_*) = 0$ for all $j$. Hence, $x_* \in \Omega$.

Next, we extend $\{\nabla c_j(x_*)\}$ to a basis of $\mathbb{R}^N$,

$$\{\nabla c_1(x_*), \ldots, \nabla c_M(x_*), \psi_{M+1}, \ldots, \psi_N\},$$

and define $\Psi_* \in \mathbb{R}^{N\times N}$ to be the matrix that has these basis vectors as its rows. It then follows that $\Psi_*$ is invertible. Similarly, we define $\Psi_n$ to be the matrix where the first $M$ rows $\nabla c_j(x_*)^T$ are replaced by $\nabla c_j(x_n)^T$, i.e., we have,

$$\Psi_* = \begin{pmatrix} Dc(c_*) \\ \Psi^T \end{pmatrix} \quad \text{and} \quad \Psi_n = \begin{pmatrix} Dc(x_n) \\ \Psi^T \end{pmatrix},$$

where $\Psi = (\psi_{M+1}, \dots \psi_N) \in \mathbb{R}^{N-M \times N}$. Since $\|\Psi_* - \Psi_n\| \to 0$ it follows that $\Psi_n$ is invertible and that $\|\Psi_n^{-1}\|$ is bounded for sufficiently large $n$ (cf. Lemma 2.1.

We now define 'kind of' dual basis vectors $\varphi_{*,j}$, $\varphi_{n,j}$ by

$$\Psi_n \varphi_{n,j} = e_j \quad \text{and} \quad \Psi_* \varphi_{*,j} = e_j, \quad j = 1, \dots, M.$$

It now follows easily that the vectors $|\varphi_{n,j}|$ are uniformly bounded and that $\varphi_{n,j} \to \varphi_{*,j}$. Moreover, by definition,

$$\varphi_{n/*,i} \cdot \nabla c_j(x_{n/*}) = \delta_{i,j}.$$

Hence, we obtain (using $\tau_n \to 0$)

$$\begin{aligned} \varphi_{*,i} \cdot \nabla f(x_*) = & \lim_{n\to\infty} \varphi_{n,i} \cdot \nabla f(x_n) \\ = & \lim_{n\to\infty} \Big[ \sum_{j=1}^{M} \lambda_n^{(j)} \varphi_{n,i} \cdot \nabla c_j(x_n) \Big] \\ = & \lim_{n\to\infty} \lambda_n^{(i)}. \end{aligned}$$

In particular, we deduce that $\lambda_n \to \lambda_*$ where $\lambda_*^{(j)} = \varphi_{*,j} \cdot \nabla f(x_*)$ and that

$$\begin{aligned} \nabla f(x_*) = & \lim_{n\to\infty} \nabla f(x_n) \\ = & \lim_{n\to\infty} \lambda_n^T Dc(x_n) \\ = & \lambda_*^T Dc(x_*), \end{aligned}$$

which shows that $\nabla_x \mathcal{L}(x_*, \lambda_*) = 0$ and, since $x_* \in \Omega$, that $(x_*, \lambda_*)$ is a KKT point. □

To get a sense of the local convergence properties of the $\ell^2$-penalty method, let us assume that $(x_n, \lambda_n) \to (x_*, \lambda_*)$ where the limit satisfies $\nabla_{x,\lambda} \mathcal{L}(x_*, \lambda_*) = 0$, the LICQ, as well as the strong second-order optimality condition (37).

In this case, one can fairly easily see that

$$\nabla^2_{x,\lambda} \mathcal{L}_* \left[ \begin{pmatrix} x_n \\ \lambda_n \end{pmatrix} - \begin{pmatrix} x_* \\ \lambda_* \end{pmatrix} \right] = \nabla_{x,\lambda} \mathcal{L}(x_n, \lambda_n) + \text{h.o.t.},$$

where h.o.t. stands for *higher order terms*, i.e., an additional term that is of order $o(|x_n - x_*| + |\lambda_n - \lambda_*|)$. Assuming, for simplicity, that $\nabla_x \Phi(\mu_n; x_n) = 0$, then $\nabla_x \mathcal{L}(x_n, \lambda_n)| = 0$ (by definition), and since $\nabla^2_{x,\lambda} \mathcal{L}_*$ is invertible, we deduce

$$|x_n - x_*| + |\lambda_n - \lambda_*| \approx |c(x_n)|,$$

for $n$ sufficiently large, where $\approx$ stands for an upper and lower bound in terms of a fixed constant. However, to bound $|c(x_n)|$ we can only use the fact that $\Phi(\mu_n; x_n)$ remains bounded from which it follows immediately that $|c(x_n)| = O(\mu_n^{-1/2})$, and so we obtain

$$|x_n - x_*| + |\lambda_n - \lambda_*| \leq C \mu_n^{-1/2}. \tag{42}$$

for $n$ sufficiently large. This is a slow convergence rate which requires very large penalty parameters to achieve satisfactory results. This, in turn, leads to extremely ill-conditioned unconstrained optimisation problems that may be difficult and unreliable to solve. (However, see Problem 8.2 below.)

## 8.2 The augmented Lagrangian approach

The augmented Lagrangian is defined as

$$\mathcal{L}_A(\mu; x, \lambda) = \mathcal{L}(x) + \tfrac{1}{2}\mu|c(x)|^2.$$

In this approach, one essentially replaces $\Phi$ by $\mathcal{L}_A$ in Algorithm 8.1, leaving also $\lambda$ fixed at each step and updating it consecutively. This method is based on the observation that, if $(x_*, \lambda_*)$ is a KKT point then

$$\nabla_x \mathcal{L}_A(\mu; x_*, \lambda_*) = \nabla_x \mathcal{L}(x_*, \lambda_*) + \mu \nabla c(x_*) c(x_*) = 0,$$

i.e., KKT points are always critical points of $\mathcal{L}_A$. Moreover, the additional penalization (or regularization) turns $x_*$ into a local minimizer (rather than a saddle point) provided provided $\mu$ is sufficiently large.

**Proposition 8.2.** *Suppose that $(x_*, \lambda_*)$ is a KKT point at which the LICQ hold, as well as the strong second-order optimality condition (37). Then there exists $\bar{\mu} > 0$ such that, for all $\mu \geq \bar{\mu}$, $x_*$ is a strict local minimizer of $\mathcal{L}_A(\mu; \cdot, \lambda_*)$ in $\mathbb{R}^N$.*

*Proof.* If we can show that $\nabla_x^2 \mathcal{L}_A(\mu; x_*, \lambda_*)$ is positive definite, provided $\mu$ is sufficiently large, then the result follows. This is established in the following lemma. □

**Lemma 8.3.** *Under the conditions of Proposition 8.2, there exist constants $c_0, c_1 > 0$ and $c_2 \in \mathbb{R}$ such that*

$$h^T \nabla_x^2 \mathcal{L}_A(x_*, \lambda_*) h \geq c_0 |h_0|^2 + c_1(\mu - c_2)|h_1|^2,$$

*where $h_0 \in \ker Dc(x_*)$ and $h_1 \in \ker Dc(x_*)^\perp$.*

*Proof.* Note that

$$\begin{aligned} \nabla_x^2 \mathcal{L}_A(\mu; x_*, \lambda_*) = &\ \nabla_x^2 \mathcal{L}(x_*, \lambda_*) + \mu \sum_{j=1}^M c_j(x_*) \nabla^2 c_j(x_*) + \mu \sum_{j=1}^M \nabla c_j(x_*) \nabla c_j(x_*)^T \\ = &\ \nabla_x^2 \mathcal{L}(x_*, \lambda_*) + \mu Dc(x_*)^T Dc(x_*). \end{aligned}$$

Let $h = h_0 + h_1$ be the composition described above, then, using the fact that $Dc(x_*)h = Dc(x_*)h_1$,

$$\begin{aligned} h^T \nabla_x^2 \mathcal{L}_A(\mu; x_*, \lambda_*) h = &\ h^T \nabla_x^2 \mathcal{L}(x_*, \lambda_*) h + \mu h^T Dc(x_*)^T Dc(x_*) h \\ = &\ h_0^T \nabla_x^2 \mathcal{L}_* h_0 + 2 h_0^T \nabla_x^2 \mathcal{L}_* h_1 + h_1^T \nabla_x^2 \mathcal{L}_* h_1 + \mu |Dc(x_*) h_1|^2 \\ \geq &\ \tilde{c}_0 |h_0|^2 - 2|h_0||h_1| \|\nabla_x^2 \mathcal{L}_*\| - |h_1|^2 \|\nabla_x^2 \mathcal{L}_*\| + \mu |Dc(x_*) h_1|^2, \end{aligned}$$

where $\tilde{c}_0$ is the coercivity constant of $\nabla_x^2 \mathcal{L}(x_*, \lambda_*)$ in $\ker Dc(x_*)$.

Since $Dc(x_*)$ has full rank, it follows that $Dc(x_*)$ is invertible on the subspace $\ker Dc(x_*)^\perp$ and hence

$$|Dc(x_*) h_1| \geq \tilde{c}_1 |h_1|.$$

Furthermore, using the Cauchy Inequality (4) we have

$$2\|\nabla_x^2\mathcal{L}_*\||h_0||h_1| = (\epsilon^{-1/2}2^{1/2}\|\nabla_x^2\mathcal{L}_*\||h_1|)(\epsilon^{1/2}2^{1/2}|h_0|) \leq \epsilon|h_0|^2 + \tilde{c}_2|h_1|^2,$$

where $\tilde{c}_2 = \epsilon^{-1}\|\nabla_x^2\mathcal{L}_*\|^2$. Hence, setting $\epsilon = \frac{1}{2}\tilde{c}_0 =: c_0$, we deduce

$$h^T\nabla_x^2\mathcal{L}_A(\mu; x_*, \lambda_*)h \geq c_0|h_0|^2 - (\|\nabla_x^2\mathcal{L}_*\| + \tilde{c}_2)|h_1|^2 + \tilde{c}_2\mu|h_1|^2.$$

From this, the stated result follows immediately. □

It is clear from Proposition 8.2 that a good update for $\lambda$ at each step of the augmented Lagrangian method will be crucial. Upon observing that

$$\nabla_x\mathcal{L}_A(\mu_n; x_n, \lambda_n) = \nabla f(x) - \sum_{j=1}^{M}\big[\lambda_n^{(j)} - \mu_n c_j(x_n)\big]\nabla c_j(x_n),$$

the update

$$\lambda_{n+1} = \lambda_n - \mu_n c(x_n)$$

becomes very natural.

Note, in particular, that

$$|c(x_n)| = \frac{|\lambda_n - \lambda_{n+1}|}{\mu_n}, \tag{43}$$

which, if $\lambda_n \to \lambda_*$, is much smaller than $1/\mu_n$ and hence we obtain a much faster convergence rate for the constraint than in the $\ell^2$-penalty method. In particular, we do not necessarily require that $\mu_n \to \infty$.

**Algorithm 8.2 (Basic Augmented Lagrangian Algorithm).**
**Input:** $x_0^S, \lambda_0 \in \mathbb{R}^N$, $\mu_0 > 0$, $\tau_0 > 0$
1: **for** $n = 0, 1, 2, \ldots$ **do**
2: Using $x_n^S$ as a starting point, compute $x_n$ such that $|\nabla_x\mathcal{L}_A(\mu_n; x_n, \lambda_n)| \leq \tau_n$.
3: $\lambda_{n+1} \leftarrow \lambda_n - \mu_n c(x_n)$;
4: Choose $\mu_{n+1}$, $\tau_{n+1}$; $x_{n+1}^S \leftarrow x_n$;
5: **end for**

**Theorem 8.4.** *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$, $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^M)$, and that $x_n$ generated by Algorithm 8.2 (with $\tau_n \to 0$ and $\mu_n \to \infty$) converges to a point $x_*$ at which the LICQ hold. Then, $\lambda_n \to \lambda_*$ where $(x_*, \lambda_*)$ is a KKT point.*

***Problem 8.1.*** Prove Theorem 8.4 (follow the proof of Theorem 8.1).

*Hint: closely follow the global convergence proof for the $\ell^2$-penalty method, but prove convergence of the multipliers first, and deduce afterwards that $x_* \in \Omega$.* □

Again, we discuss semi-heuristically how the augmented Lagrangian approach behaves in the neighbourhood of a KKT point $(x_*, \lambda_*)$ which satisfies the LICQ as well as the strong second-order optimality condition (37). If $\mu$ is sufficiently large then $\nabla_x\mathcal{L}_A(\mu; x_*, \lambda_*) =$

0 and $\nabla_x^2 \mathcal{L}_A(\mu; x_*, \lambda_*)$ is positive definite and we can estimate the error for $x_n$ and $\lambda_n$ separately. Assuming again that $\nabla_x \mathcal{L}_A(\mu_n; x_n, \lambda_n) = 0$, we obtain

$$\begin{aligned} \nabla_x^2 \mathcal{L}_A(\mu_n; x_*, \lambda_*)(x_n - x_*) &= \nabla_x \mathcal{L}_A(\mu_n; x_n, \lambda_*) + \text{h.o.t.} \\ &= \nabla_x \mathcal{L}_A(\mu_n; x_n, \lambda_*) - \nabla_x \mathcal{L}_A(\mu_n; x_n, \lambda_n) + \text{h.o.t.} \\ &= \nabla c(x_*)(\lambda_n - \lambda_*) + \text{h.o.t.}. \end{aligned} \tag{44}$$

Let $h := x_n - x_* = h_0 + h_1$ where $h_0 \in \ker Dc(x_*)$ and $h_1 \perp h_0$. Multiplying this equality with $h^T$ and using Lemma 8.3, we obtain

$$\begin{aligned} c_0(|h_0|^2 + (\mu_n - c_1)|h_1|^2) &\leq h^T \nabla_x^2 \mathcal{L}_A(\mu_n; x_*, \lambda_*) h \\ &= h^T \nabla c(x_*)(\lambda_n - \lambda_*) + \text{h.o.t.} \\ &= h_1^T \nabla c(x_*)(\lambda_n - \lambda_*) \\ &\leq C|h_1||\lambda_n - \lambda_*|. \end{aligned}$$

In particular, we obtain that, for $\mu_n$ sufficiently large,

$$|h_1| \leq C\mu_n^{-1}|\lambda_n - \lambda_*|.$$

Moreover, upong multiplying (44) with $h_0^T$, we can see fairly easily that $|h_0| \leq C|h_1|$ and hence we deduce that, for $n$ sufficiently large (so that the h.o.t.s become hidden),

$$|x_n - x_*| \leq C\mu_n^{-1}|\lambda_n - \lambda_*|. \tag{45}$$

In a next step, we show that a similar estimate holds for $|\lambda_{n+1} - \lambda_*|$. This follows from the fact that, by definition, $\nabla_x \mathcal{L}(x_n, \lambda_{n+1}) = 0$ and, similarly as in the calculations leading up to (42), and using (43), we obtain

$$|\lambda_{n+1} - \lambda_*| \approx |c(x_n)| = \mu_n^{-1}|\lambda_n - \lambda_{n+1}|.$$

Hence, for $\mu_n$ sufficiently large, it follows that

$$|\lambda_{n+1} - \lambda_*| \leq C\mu_n^{-1}|\lambda_n - \lambda_*|.$$

This analysis can be made rigorous and yields the following result.

**Theorem 8.5.** *Suppose that $(x_*, \lambda_*)$ is a KKT point at which the LICQ and the strong second order criticality (37) hold. Suppose, further, that $(x_n, \lambda_n) \to (x_*, \lambda_*)$ and $n \to \infty$, where $(x_n, \lambda_n)$ is generated using Algorithm 8.2.*

*Then there exists $\hat{\mu} > 0$ such that, if $\mu_n \geq \hat{\mu}$ for all $n \geq n_0$, then*

$$|x_n - x_*| + |\lambda_{n+1} - \lambda_*| \leq C\mu_n^{-1}|\lambda_n - \lambda_*| \qquad \forall n > n_0.$$

*Moreover, for $n \geq n_1 \geq n_0$, there exist locally unique points $\bar{x}_n$ near $x_*$ which satisfy $\nabla_x \mathcal{L}_A(\mu_n; \bar{x}_n, \lambda_n) = 0$ (i.e., approximate solutions $x_n$ to the subproblem in step 2 do exist).*

***Problem 8.2.*** Let $f \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R})$ and $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^M)$.

(a) Let $\Phi$ be the merit function of the $\ell^2$-penalty method. Show that

$$\nabla_x^2 \Phi(\mu; x) = \nabla_x^2 \mathcal{L}(x, \lambda) + \mu Dc(x)^T Dc(x),$$

where $\lambda = -\mu c(x)$. Explain why the system for computing the Newton direction $\nabla_x^2 \Phi(\mu; x) s^N = -\nabla_x \Phi(\mu; x)$ is ill-posed when $\mu$ is large.

(b) Show that the Newton direction $s^N$ can also be computed using the enlarged system

$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(x, \lambda) & -Dc(x)^T \\ -Dc(x) & \mu^{-1} I \end{pmatrix} \begin{pmatrix} s^N \\ \xi \end{pmatrix} = \begin{pmatrix} -\nabla_x \Phi(\mu; x) \\ 0 \end{pmatrix}.$$

Show that the condition number for this system is bounded, independent of $\mu$.

(c) Derive a similar system to stably compute the Newton direction for the augmented Lagrangian method. □

## 8.3 A non-smooth merit function

Assume that $f \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R})$ and $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^M)$. We consider a penalty method where the $\ell^2$-penalty function (or merit function) is replaced by

$$\Phi_1(\mu; x) = f(x) + \mu |c(x)| = f(x) + \mu \Big( \sum_{j=1}^{M} |c_j(x)|^2 \Big)^{1/2}.$$

***Problem 8.3.***

(a) Show that $\Phi_1(\mu; \cdot)$ is twice differentiable at every point $x \in \mathbb{R}^N \setminus \Omega$, but that (if $M < N$) it is not differentiable at any point $x \in \Omega$. For $x \in \mathbb{R}^N \setminus \Omega$ show that the gradient is given by

$$\nabla_x \Phi_1(\mu; x) = \nabla f(x) + \mu \nabla c(x) \frac{c(x)}{|c(x)|}.$$

(b) Suppose now that $x_*$ is an isolated local minimizer of $f$ in $\Omega$, with associated Lagrange multiplier $\lambda_*$, and $\nabla f(x_*) \neq 0$. Show that there exists $\bar{\mu} > 0$ such that, for all $\mu \geq \bar{\mu}$, $x_*$ is also an isolated local minimizer of $\Phi_1(\mu; \cdot)$.

(c) Try to find a an estimate for $\bar{\mu}$ in terms of $\lambda_*$ only. □

***Remark 8.6.*** To conclude the section on penalty and augmented Lagrangian methods, we remark that these methods (and their theory) can be extended to inequality constraints. For example, the $\ell^2$-penalty function would become

$$\Phi(\mu; x) = f(x) + \tfrac{1}{2} \mu \Big( \sum_{j \in \mathcal{E}} |c_j(x)|^2 + \sum_{j \in \mathcal{I}} |c_j(x)^-|^2 \Big),$$

where $z^- = \min(0, z)$. □

# 9 Barrier Methods

In this section we consider a purely inequality constrained optimisation problem

$$\min_{x\in\Omega} f(x) \quad \text{where} \quad \Omega = \{x \in \mathbb{R}^N : c_j(x) \geq 0, j = 1, \ldots, M\} \tag{46}$$

(i.e., $M = M_i$ and $M_e = 0$ in our previous notation).

In Section 8 we have seen examples of turning a constrained optimisation problem into a series of unconstrained problems by enforcing asymptotic feasibility into a 'merit function'. The *barrier method* makes a choice of merit function where only stritly admissible points have finite merit function value,

$$P(\mu; x) = \begin{cases} f(x) - \mu \sum_{j=1}^{M} \log c_j(x), & c_j(x) > 0 \quad \forall j \in \mathcal{I}, \\ +\infty, & \text{otherwise}, \end{cases} \tag{47}$$

where log denotes the natural logarithm. Here, we will let $\mu \to 0$. We say that $x$ is *strictly admissible* if $c_j(x) > 0$ for all $j$. A basic barrier algorithm can be outlined as follows.

**Algorithm 9.1 (Basic Barrier Algorithm).**
**Input:** $x_0^S$ strictly feasible, $\mu_0$, $\tau_0$
1: **for** $n = 0, 1, 2, \ldots$ **do**
2: Compute $x_n$ such that $|\nabla_x P(\mu_n; x_n)| \leq \tau_n$
3: Choose $\mu_{n+1}$, $\tau_{n+1}$, $x_{n+1}^S$;
4: **end for**

The choice of an initial starting point $x_0^S$ can be very difficult. This is addressed in Problem +++ below. A typical choice for $\mu_{n+1}$ is $\mu_{n+1} = \mu_n/10$.

## 9.1 Convergence of the barrier method

The termination criterion for the 'inner' minimisation problem in Algorithm 9.1 guarantees that

$$\nabla_x P(\mu_n, x_n) = \nabla f(x_n) - \sum_{j\in\mathcal{I}} \frac{\mu_n}{c_j(x_n)} \nabla c_j(x_n) = O(\tau_n),$$

As in the penalty case this gives us, once again, Lagrange multiplier estimates, by setting $\lambda_n^{(j)} = -\mu_n/c_j(x_n)$, $j \in \mathcal{I}$.

**Theorem 9.1.** *Let $x_n \to x_*$ where $(x_n)_{n\in\mathbb{N}}$ is generated by Algorithm 9.1. If the point $x_*$ satisfies the LICQ then $x_* \in \Omega$, $\lambda_n \to \lambda_*$ exists, and $(x_*, \lambda_*)$ is a KKT point of (46).*

***Problem 9.1.*** Prove Theorem 9.1. The proof is similar as for Theorems 8.1 and 8.4. □

## 9.2 The problem of starting point selection for the inner loop

A seemingly natural choice for the starting point is $x_{n+1}^S = x_n$. We will see in the following that this is rarely a good option.

The Newton step for $\Phi(\mu_{n+1}; \cdot)$ at the point $x_n$ is given by

$$\nabla_x^2 P(\mu_{n+1}; x_n) s^N = -\nabla_x P(\mu_{n+1}; x_n).$$

To see the structure behind this step, we first rewrite the hessian of $P$,

$$\begin{aligned}\nabla_x^2 P(\mu_{n+1}; x_n) &= \nabla^2 f(x_n) - \frac{\mu_{n+1}}{\mu_n}\sum_{j=1}^{M}\lambda_n^{(j)}\nabla^2 c_j(x_n) + \frac{\mu_{n+1}}{\mu_n^2}\sum_{j=1}^{M}(\lambda_n^{(j)})^2\nabla c_j(x_n)\nabla c_j(x_n)^T \\ &= \nabla_x^2\mathcal{L}(x_n,\lambda_n) + \frac{\mu_n-\mu_{n+1}}{\mu_n}\sum_{j=1}^{M}\lambda_n^{(j)}\nabla c_j^2(x_n) + \frac{\mu_{n+1}}{\mu_n^2}A_nA_n^T,\end{aligned}$$

where $A_n = [\cdots \lambda_n^{(j)}\nabla c_j(x_n)\cdots]$, and the gradient,

$$\begin{aligned}\nabla_x P(\mu_{n+1}; x_n) &= \nabla f(x_n) - \sum_{j=1}^{M}\frac{\mu_{n+1}}{c_j(x_n)}\nabla c_j(x_n) \\ &= \nabla\mathcal{L}(x_n,\lambda_n) + \frac{\mu_n-\mu_{n+1}}{\mu_n}\sum_{j=1}^{M}\lambda_n^{(j)}\nabla c_j(x_n).\end{aligned}$$

If $x_n \to x_*$ and $\lambda_n \to \lambda_*$ where $(x_*, \lambda_*)$ is a KKT point, then it is clear from this calculation that $\kappa(\nabla_x^2 P(\mu_n; x_n)) \sim \mu_n^{-1}$ as $n \to \infty$. This is a minor problem, however, which can be easily corrected using similar ideas as in Problem 8.2.

A much more severe problem is that a Newton step will almost always take us outside the admissible set $\Omega$. The following calculation is very heuristic but represents the typical behaviour of barrier methods fairly accuractely. We begin by noting that, for small $\mu_n, \mu_{n+1}$, and for $x_n \approx x_*, \lambda_n \approx \lambda_*$, we have (since $(x_*, \lambda_*)$ is a KKT point)

$$\nabla_x^2 P(\mu_{n+1}; x_n) \approx \frac{\mu_{n+1}}{\mu_n^2}A_nA_n^T \quad \text{and} \quad -\nabla_x P(\mu_{n+1}; x_n) \approx \frac{\mu_n-\mu_{n+1}}{\mu_n}\sum_{j=1}^{M}\lambda_n^{(j)}\nabla c_j(x_n).$$

In particular, it follows that

$$\frac{\mu_{n+1}}{\mu_n^2}\sum_{j=1}^{M}(\lambda_n^{(j)})^2(\nabla c_j(x_n)\cdot s^N)\nabla c_j(x_n) \approx \frac{\mu_n-\mu_{n+1}}{\mu_n}\sum_{j=1}^{M}\lambda_n^{(j)}\nabla c_j(x_n).$$

Since the LICQ holds at $x_*$ it also holds at $x_n$ and hence, comparing coefficients, we obtain

$$\frac{\mu_{n+1}}{\mu_n}\lambda_n^{(j)}\nabla c_j(x_n)\cdot s^N \approx \mu_n - \mu_{n+1},$$

or equivalently,

$$\nabla c_j(x_n)\cdot s^N \approx \Big(1 - \frac{\mu_n}{\mu_{n+1}}\Big)c_j(x_n),$$

which immediately gives

$$c_j(x_n + s^N) \approx c_j(x_n) + \nabla c_j(x_n)\cdot s^N \approx \Big(2 - \frac{\mu_n}{\mu_{n+1}}\Big)c_j(x_n).$$

In particular, all but the most modest decrease in $\mu$ leads to Newton updates which lie outside the admissible set $\Omega$. This shows that the starting points $x_{n+1}^S = x_n$ are not typically useful in practise.

### 9.3 The primal-dual barrier method

The problem of the starting point $x_{n+1}^S = x_n$ can be overcome by exploiting the fact that $\lambda_n$ gives a reliable Lagrange multiplier estimate in the limit as $n \to \infty$. Recall the KKT system for purely inequality-constrained systems,

$$\begin{aligned} \nabla_x \mathcal{L}(x, \lambda) &= 0 \\ \lambda^{(j)}, c_j(x) &\geq 0, \qquad j = 1, \ldots, M, \\ \lambda^{(j)} c_j(x) &= 0, \qquad j = 1, \ldots, M. \end{aligned}$$

The idea is to ignore the second condition and to perturb the third equation, leading to the nonlinear system

$$\begin{aligned} \nabla_x \mathcal{L}(x, \lambda) &= 0 \\ \lambda^{(j)} c_j(x) &= \mu, \qquad j = 1, \ldots, M, \end{aligned}$$

where $\mu > 0$. Note that a Newton step for this system will, essentially, be a Newton step for the inequality constrained system rather than the barrier system. Such an initial correction step will give an excellent starting point for the next iteration of the barrier method.

Recall that we have defined $\lambda_n$ so that $\lambda_n^{(j)} c_j(x_n) = \mu_n$. Hence, starting from $(x_n, \lambda_n)$ and taking one Newton step $(s^N, \xi^N)$ for this system, with $\mu = \mu_{n+1}$ gives

$$\begin{pmatrix} \nabla_x^2 \mathcal{L}(x_n, \lambda_n) & -Dc(x_n)^T \\ \Lambda_n Dc(x_n) & C_n \end{pmatrix} \begin{pmatrix} s^N \\ \xi^N \end{pmatrix} = \begin{pmatrix} -\nabla_x \mathcal{L}(x_n, \lambda_n) \\ (\mu_{n+1} - \mu_n) e \end{pmatrix}, \tag{48}$$

where $e = (1, \ldots, 1)^T$, $\Lambda_n = \operatorname{diag}(\lambda_n)$, and $C_n = \operatorname{diag}(c(x_n))$. Because we are correting both the primal variable $x$ and the dual variable $\lambda$, this is called the *primal-dual* system. We shall investigate the conditioning of this system below.

We can show that this correction step leads to a suitable starting point. Assume again, for simplicity, that all constraints are active in the limit and that the LICQ holds. Multiplying the second line of (48) with $\nabla c_n \Lambda_n$ gives

$$\nabla c_n \Lambda_n^2 \nabla c_n^T s^N + \mu_n \nabla c_n \xi^N = (\mu_{n+1} - \mu_n) \nabla c_n \Lambda_n e.$$

Using the first line of (48) we have

$$\nabla c(x_n) \xi^N = \nabla_x \mathcal{L}(x_n, \lambda_n) + \nabla_x^2 \mathcal{L}(x_n, \lambda_n) s^N \approx \nabla_x^2 \mathcal{L}(x_n, \lambda_n) s^N,$$

from which we obtain

$$\big(\mu_n \nabla_x^2 \mathcal{L}(x_n, \lambda_n) + \nabla c(x_n) \Lambda_n^2 \nabla c(x_n)^T\big) s^N \approx (\mu_{n+1} - \mu_n) \nabla c_n \Lambda_n e.$$

Similarly as in section 9.2 we can conclude, in view of the LICQ, that

$$\lambda_n^{(j)} \nabla c_j(x_n) \cdot s^N \approx \mu_{n+1} - \mu_n,$$

which gives

$$\begin{aligned} c_j(x_n + s^N) &\approx c_j(x_n) + \nabla c_j(x_n) \cdot s^N \approx c_j(x_n) + \frac{\mu_{n+1} - \mu_n}{\lambda_n^{(j)}} \\ &= c_j(x_n) \Big(1 + \frac{\mu_{n+1} - \mu_n}{\mu_n}\Big) = \frac{\mu_{n+1}}{\mu_n} c_j(x_n), \end{aligned}$$

which is precisely the scaling we had hoped to achieve.

Using the Primal-Dual Newton correction to compute a starting point at each step gives the following *Primal-Dual barrier algorithm.*

**Algorithm 9.2 (A Simple Primal-Dual Barrier Method).**
**Input:** $x_0^S$ s.t. $c(x_0^S) > 0$, $\mu_0 > 0$, $\tau_0$;
1: **for** $n = 0, 1, 2, \ldots$ **do**
2: Using $x_n^S$, compute $x_n$ such that $|\nabla_x P(\mu_n; x_n)| \leq \tau_n$;
3: Choose $\mu_{n+1}, \tau_{n+1}$;
4: Set $\lambda_n^{(j)} = -\mu_n / c_j(x_n)$;
5: Compute PD-correction step $(s^N, \xi^N)$ using (48), and set $x_{n+1}^S = x_n + s^N$;
6: **end for**

***Remark 9.2.*** In practise, the PD-correction step should be replaced by one or more *damped* Newton iterations. Another way of looking at this is that, the unconstrained minimization in step 2 of the algorithm should be performed using the primal dual Newton ideas throughout, taking $x_n = x_{n+1}^S$ as a starting point after all. □

# 10 Remarks on Large Scale Optimisation

# Acknowledgements

Here, I list the literature which I have followed in preparing these notes.

Section 5 on trust region methods follows largely the lecture notes of R. Hauser [1, Lectures 6 and 7] and partially also [2, Section 3.3.6]. The Dennis–Moré condition follows [4, Sec. 3.3], Algorithm 6.1 is taken from [1, Lecture 2]. The presentation of Section 7 is a mixture of [4, Ch. 12] and [1, Lectures 8–11]. However, I completely skipped the *Farkas Lemma* in the proof of the KKT conditions. To explain why this is fairly easy to do, I have added a problem, where one ought to prove the Farkas' Lemma for the case where the matrix defining the cone has full rank. Section 8 on penalty and augmented Lagrangian methods is inspired by [4, Ch. 17], though I have restricted the presentation to equality constraints. Section 9 follows [3, Ch. 4] and [1, Lecture 15].

# References

[1] R.A. Hauser. Continuous optimisation. Lecture Notes.

[2] C. T. Kelley. *Iterative methods for optimization*, volume 18 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[3] N.I.M.Gould. An introduction to algorithms for nonlinear optimization. RAL-TR-2002-03 (revised).

[4] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.