MA40050

Numerical Optimisation & Large-Scale Systems

Pranav Singh
ps2106@bath.ac.uk
Department of Mathematical Sciences
University of Bath
Bath BA2 7AY

These notes are heavily based on the notes by Melina Freitag & Robert Scheichl.

April 30, 2020

Contents

1 Introduction 1

1.1 An example . . . 1
1.2 Other examples (not taught) . . . 3
1.3 Mathematical problem statement . . . 5

2 Preliminaries 5

2.1 Linear algebra primer . . . 5
2.2 Multi-variable calculus primer . . . 8
2.3 Optimality conditions for unconstrained optimisation . . . 11
2.4 Convergence rates . . . 12

3 Newton's Method 13

4 Line Search Methods 16

4.1 The basic steepest descent algorithm . . . 16
4.2 Variable-metric steepest descent . . . 20

5 Trust Region Methods 25

5.1 The Cauchy-point . . . 26
5.2 Accepting/rejecting updates & trust region radius management . . . 26
5.3 Global convergence of trust region methods . . . 27
5.4 The dogleg method . . . 28

6 Quasi-Newton Methods 31

6.1 The Dennis–Moré condition for superlinear convergence . . . 31
6.2 The secant condition and quasi-Newton updates . . . 32
6.3 The Sherman–Morrison–Woodbury formula . . . 33
6.4 The Wolfe conditions . . . 34
6.5 The BFGS method . . . 35

7 Optimality Conditions for Constrained Optimisation 37

7.1 A basic first-order optimality condition . . . 37
7.2 The linearised tangent cone . . . 38
7.3 The Karush–Kuhn–Tucker conditions . . . 40
7.4 Second-order optimality conditions . . . 41

8 The Method of Lagrange Multipliers 43

8.1 Equality constraints – sequential quadratic programming . . . 43
8.2 Inequality & equality constraints – the active set method . . . 45

9 Penalty and Augmented Lagrangian Methods 48

9.1 The ℓ^2-penalty method . . . 48
9.2 The augmented Lagrangian approach . . . 49

10 Large Scale Systems 53

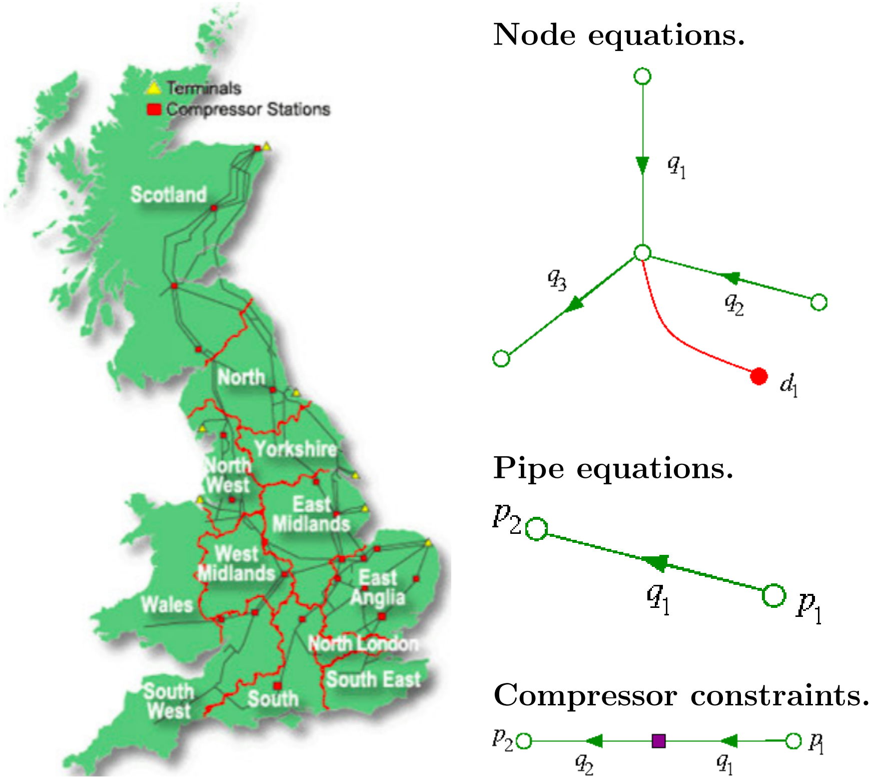

Figure 1: The Transco National Transmission System

1 Introduction

What is nonlinear programming?

Nonlinear (constrained) optimisation $\equiv$ nonlinear programming:

$$\min_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{c}_{\mathcal{E}}(\mathbf{x}) = 0 \quad \text{and} \quad \mathbf{c}_{\mathcal{I}}(\mathbf{x}) \geq 0$$

where $f : \mathbb{R}^N \longrightarrow \mathbb{R}$ is the *objective function*, $\mathbf{c}_{\mathcal{E}} : \mathbb{R}^N \longrightarrow \mathbb{R}^{M_e}$ $(M_e \leq N)$ are the *equality constraints* and $\mathbf{c}_{\mathcal{I}} : \mathbb{R}^N \longrightarrow \mathbb{R}^{M_i}$ are the *inequality constraints.*

1.1 An example

Optimisation of a high-pressure gas network (Fig. 1). A collaboration between British Gas (Transco), Oxford University and Rutherford Appleton Laboratory (RAL), courtesy of Nick Gould (RAL).

(a) **Node Equations.**

$$q_1 + q_2 - q_3 - d_1 = 0$$

where q_i are *flows* on Pipe i and d_j the *demands* at Node j (Fig. 1 (top right)).

This is an example of **general (linear) equality constraints**

$$A\mathbf{q} - \mathbf{d} = \mathbf{0}$$

with A linear, sparse and structured.

(b) **Pipe Equations.**

$$p_2^2 - p_1^2 + k_1 q_1^{2.8359} = 0$$

where p_i are *pressures*, q_i are *flows* and k_i are *constants.*

This is an example of **general (non-linear) equality constraints**

$$A(\mathbf{p}) + \mathrm{diag}(k_i g(q_i)) = \mathbf{0}$$

with A non-linear, sparse, structured.

(c) **Compressor Constraints.**

$$q_1 - q_2 + z_1 \cdot c_1(p_1, q_1, p_2, q_2) \geq 0$$

where p_i are *pressures*, q_i are *flows*, z_i are *0–1 variables* (= 1 if machine is on) and c_i are *nonlinear functions.*

This is an example of **general inequality constraints**

$$A\,\mathbf{q} + \mathrm{diag}(z_i)\, c(\mathbf{p}, \mathbf{q}) \geq \mathbf{0}$$

with c non-linear, A sparse, structured and including 0–1 variables.

(d) **Other Constraints.** Bounds on pressures and flows

$$\begin{aligned} p_{\min} &\leq p \leq p_{\max} \\ q_{\min} &\leq q \leq q_{\max} \end{aligned}$$

In general: Simple bounds on variables.

(e) **Choice of objective function $f(\mathbf{x})$.** Many possible objectives:

- maximize / minimize sum of pressures
- minimize compressor fuel cost
- minimize supply

+ combinations of these

Actual Data. *British Gas National Transmission System*

- 199 nodes
- 196 pipes
- 21 compressors

Leading to a steady state problem with ∼ **400 variables**.

For a 24-hour variable-demand problem with 10 minute discretization this would immediately go up to ∼ **58,000 variables !**

Challenge: Solve this in real time!

This problem is **typical** of **real-world, large-scale** applications and the **motivation for the course**:

- linear constraints
- nonlinear constraints
- simple bounds
- structure
- integer variables
- global minimum “required”
- discretization

Example 1.1 (Data Fitting). An experiment is described by the nonlinear relation $y = F(p, x)$, where $F : \mathbb{R}^P \times \mathbb{R}^{K_I} \to \mathbb{R}^{K_O}$, $x \in \mathbb{R}^{K_I}$ describes the condition under which the experiment is conducted (inputs), and $y \in \mathbb{R}^{K_O}$ is the outcome of the experiment (observations or outputs). The vector p 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 $(x_i, y_i)_{i=1}^M \subset \mathbb{R}^{K_I} \times \mathbb{R}^{K_O}$. To estimate the parameter vector p we minimize

$$\sum_{i=1}^{M} |F(p, x_i) - y_i|^2$$

with respect to the unknown $p \in \mathbb{R}^P$. This is an example of a *nonlinear least squares problem* (as opposed to a linear least squares problem where $p \mapsto F(p, x)$ is linear). To prevent over-fitting, we may add a regularizing or penalty term $R(p)$ to the objective function, where $R : \mathbb{R}^P \to \mathbb{R}_+$.

1.2 Other examples (not taught)

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 Γ 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 Γ.

(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 $y_3(x) \geq 0$ for all $x \in \Omega$.

Other Application Areas:

- Minimum energy problems
- Structural design problems
- Traffic equilibrium models
- Production scheduling problems
- Portfolio selection
- Parameter determination in financial markets
- Hydro-electric power scheduling
- Gas production models
- Efficient models of alternative energy sources
- Computer tomography (image reconstruction)
- 3D reconstruction (computer vision)
- Image denoising
- Optimal control & PDE constrained optimization
- Back propagation in neural networks

1.3 Mathematical problem statement

Let $f : \mathbb{R}^N \to \mathbb{R}$ be the (smooth) *objective function*. Furthermore, we assume throughout that the *admissible (or feasible) set* Ω 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 $c : \mathbb{R}^N \to \mathbb{R}^{M_e+M_i}$, $\mathcal{E} = \{1, \dots, M_e\}$ and $\mathcal{I} = \{M_e + 1, \dots, M_e + M_i\}$. A point $x \in \Omega$ is called *feasible* or *admissible.*

A *global minimizer* of f in Ω (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 $x_* \in \Omega$ is a *local minimizer* of f in Ω (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 Ω (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 minimizer.

2 Preliminaries

2.1 Linear algebra primer

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

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

If we want 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.

Exercise.

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

(b) Show that (4) implies (5). [*Hint: First prove it for $|x| = |y| = 1$, then generalize.*]

(c) Use (5) to prove the *Triangle Inequality*

$$|x + y| \leq |x| + |y| \ .$$

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

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

Exercise. Show that $\| \cdot \|$ is a norm on the space of $M \times N$ matrices $\mathbb{R}^{M \times N}$.

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*.

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

$$\|S^{-1}\| \leq \frac{\|T^{-1}\|}{1 - \|T^{-1}\| \|T - S\|} \ .$$

Proof. Let $A \in \mathbb{R}^{N \times N}$ with $\|A\| < 1$. Then, $\|A^n\| \leq \|A\|^n \xrightarrow{n \to \infty} 0$ and so

$$(I - A)(I + A + A^2 + \ldots + A^n) = I - A^{n+1} \ \xrightarrow{n \to \infty} \ I \ .$$

Thus,

$$(I - A)^{-1} = \sum_{n=0}^{\infty} A^n$$

and

$$\|(I - A)^{-1}\| \leq \sum_{n=0}^{\infty} \|A\|^n = \frac{1}{1 - \|A\|} \ .$$

Now, let $A = T^{-1}(T - S)$. Then, $\|A\| \leq \|T^{-1}\| \|T - S\| < 1$, by assumption. Hence,

$$S^{-1}T = (I - T^{-1}(T - S))^{-1} = \sum_{n=0}^{\infty} (T^{-1}(T - S))^n$$

and

$$\|S^{-1}\| \le \|S^{-1}T\|\|T^{-1}\| \le \frac{\|T^{-1}\|}{1 - \|T^{-1}\|\|T-S\|}\,.$$

□

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

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

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

Exercise. Let A be invertible and E such that $\|E\|\|A^{-1}\| \le \frac{1}{2}$. Furthermore, suppose that $Ax = b$ and $(A+E)\tilde{x} = b$. Show that then

$$\frac{|x - \tilde{x}|}{|x|} \le 2\kappa(A)\frac{\|E\|}{\|A\|}.$$

A matrix $A \in \mathbb{R}^{N\times N}$ is called *positive definite* (in short, $A > 0$) if

$$x^T A x > 0, \qquad \text{for all } \; x \ne 0.$$

It is called *positive semi-definite* (in short, $A \ge 0$) if

$$x^T A x \ge 0, \qquad \text{for all } \; x \in \mathbb{R}^N.$$

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 $\lambda_1 \le \dots \le \lambda_N \in \mathbb{R}$ and eigenvectors $v_1, \dots, v_N$ such that*

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

The set $\{v_1, \dots v_N\}$ is an orthonormal basis of $\mathbb{R}^N$. The set $\sigma(A) := \{\lambda_1, \dots, \lambda_N\}$ is called the spectrum *of A.*

Proof. For Part (a), see **Algebra 2A**. □

Exercise (see handout). Show that the following properties hold:

1. A has the spectral decomposition $A = QDQ^T$ where $D = \mathrm{diag}(\lambda_1, \dots, \lambda_N)$ and $Q = (v_1| \dots |v_N)$. The matrix Q is orthogonal, i.e., $Q^{-1} = Q^T$ and $|Qx| = |x|$, for all x. This representation is unique up to a permutation of the columns of D and Q.

2. A is invertible if, and only if, $0 \notin \sigma(A)$.

3. If A is invertible then $\sigma(A^{-1}) = \{1/\lambda_1, \dots, 1/\lambda_N\}$ and the eigenvectors are the same.

4. $\|A\| = \max_{n \le N} |\lambda_n|$ and $\|A^{-1}\| = 1/\min_{n \le N} |\lambda_n|$. Thus, $\kappa(A) = \dfrac{\max_{n \le N} |\lambda_n|}{\min_{n \le N} |\lambda_n|}$.

5. $h^T A h \ge \min_{n \le N} \lambda_n |h|^2$, for all $h \in \mathbb{R}^N$. In particular, A is spd if, and only if, $\lambda_n > 0$, for all $n = 1, \dots, N$.

6. A is spd if, and only if, A^{-1} is spd.

7. If A is positive semi-definite, then $A^{1/2} := Q\mathrm{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_N})Q^T$ is symmetric positive semidefinite and satisfies $(A^{1/2})^2 = A$. In fact, it is the unique symmetric positive semidefinite matrix that satisfies this. If A Is spd then $A^{1/2}$ is spd.

Apart from the standard Euclidean norm $|\cdot|$, we will also use 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 spd matrix.

The associated operator norm of a matrix $A \in \mathbb{R}^{N \times N}$ is

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

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

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

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

2.2 Multi-variable calculus primer

(For more details please refer to the 2nd Year Analysis modules.)

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 $x \in U$.

Definition 2.3 (Derivatives).

(a) 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.

(b) Let $M = 1$. We say $f : U \to \mathbb{R}$ is *twice differentiable* at x if f is differentiable at x and if there exists $H \in \mathbb{R}^{N\times N}$ such that

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

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 convenient and intuitive to think of the (first) derivative as a *linear approximation* to F in a neighbourhood of x, i.e.,[1]

$$F(x+h) = F(x) + DF(x)h + o(|h|).$$

The second derivative provides a *quadratic approximation*,

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

edited 11 Feb

If s is a unit vector ($|s| = 1$), $DF(x)s$ is the rate of change in F at x in the direction s,

$$DF(x)s = \frac{\mathrm{d}}{\mathrm{d}\alpha} F(x+\alpha s)\bigg|_{\alpha=0}.$$

In practice, derivatives are represented by partial derivatives, e.g., for $F \in \mathrm{C}^1(\mathbb{R}^N)$,

$$DF(x)e_i = \nabla F(x)\cdot e_i = \frac{\partial F}{\partial x_i}(x)$$

and hence

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_N}(x)\right)^T.$$

end edit

Exercise. Let $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$. Show that the Hessian $\nabla^2 f(x)$ is a symmetric matrix with entries

$$\left[\nabla^2 f(x)\right]_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}(x), \qquad i,j = 1,\ldots,N.$$

[1] 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 $\lim_{x\to 0} o(x)/x = 0$. Similarly, the big-O-notation is used to denote a generic function $O : B_\varepsilon(0) \to \mathbb{R}^M$ which satisfies $\limsup_{x\to 0} |O(x)/x| < +\infty$.

Definition 2.4 (Lipschitz-Continuity).

(a) Let $U \subset \mathbb{R}^N$ be open or closed. We say that $F : U \to \mathbb{R}^M$, is *Lipschitz continuous* in U if there exists $L \geq 0$ such that

$$|F(x) - F(x')| \leq L|x - x'| \qquad \text{for all } \ x, x' \in U.$$

The smallest such L is denoted $\mathrm{Lip}_U(F)$, the *Lipschitz constant* of F in U. We write $\mathrm{Lip}(F)$ if $U = \mathbb{R}^N$.

(b) Let $U \subset \mathbb{R}^N$ be open. We say that F is *locally Lipschitz continuous* in U if F is Lipschitz continuous in every *closed* subset of U.

(c) Lipschitz continuity for $F : U \to \mathbb{R}^{M \times N}$ is defined in the same way (simply replacing the Euclidean norm with the operator norm above).

We finish this section with two important theorems.

Theorem 2.5 (Integral Mean Value Theorem). *Let $x_0 \in \mathbb{R}^N$ and $r > 0$. If $F : \mathbb{R}^N \to \mathbb{R}^M$ is continuously differentiable on $B_r(x_0)$ then*

$$F(x') = F(x) + \left[\int_0^1 DF\big(x + t(x' - x)\big)\,\mathrm{d}t\right](x' - x)\,, \quad \text{for all } \ x, x' \in B_r(x_0)\,.$$

Proof. Let $\phi(t) := F\big(x + t(x' - x)\big)$. Using the chain rule, we have

$$\begin{aligned}\phi_i'(t) = \frac{\mathrm{d}}{\mathrm{d}t}F_i\big(x + t(x' - x)\big) &= \sum_{j=1}^{N} \frac{\partial F_i}{\partial x_j}\big(x + t(x' - x)\big)(x_j' - x_j) \\ &= \nabla F_i\big(x + t(x' - x)\big)^T (x' - x)\,.\end{aligned}$$

So,

$$\phi'(t) = D_x F\big(x + t(x' - x)\big)(x' - x)$$

and

$$F(x') - F(x) = \phi(1) - \phi(0) = \int_0^1 \phi'(t)\,\mathrm{d}t = \left[\int_0^1 D_x F(x + t(x' - x))\,\mathrm{d}t\right](x' - x)\,.$$

□

On **Problem Sheet 1** you are asked to prove some Taylor formulae of similar type, as well as the following *Contraction Mapping Theorem.*

Definition 2.6.

(a) A Lipschitz continous function with $\mathrm{Lip}_U(F) < 1$ is called a *contraction* on U.

(b) Let $F : \mathbb{R}^N \to \mathbb{R}^N$. If $F(x) = x$ then $x \in \mathbb{R}^N$ is called a *fixpoint* of F.

Theorem 2.7 (Contraction Mapping Theorem). *Suppose*

(i) $G : \overline{B}_r(x_0) \to \overline{B}_r(x_0)$, *for some* $x_0 \in \mathbb{R}^N$ *and* $r > 0$, *and*

(ii) there exists $0 \le \alpha < 1$ *such that* $|G(x') - G(x)| \le \alpha |x' - x|$, *for all* $x, x' \in \overline{B}_r(x_0)$.

Then

(a) The fixpoint iteration $x^{k+1} = G(x^k)$ *converges to* $x_* \in \overline{B}_r(x_0)$, *for any* $x^0 \in \overline{B}_r(x_0)$.

(b) x_* *is the unique fixpoint of* G *in* $\overline{B}_r(x_0)$.

Proof. See **Problem Sheet 1**. □

Note. A sufficient condition for (i) is

$$|x_0 - G(x_0)| \le (1 - \alpha) r \, . \tag{7}$$

Exercise. Find the root of $F(x) = x - \exp(-x)$.

2.3 Optimality conditions for unconstrained optimisation

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

Proposition 2.8 (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* $\nabla^2 f(x_*) \ge 0$.

Proof. See **Problem Sheet 2**. □

Definition 2.9 (Critical Points). A point $x_* \in \mathbb{R}^N$ where $\nabla f(x_*) = 0$ is called a *first-order critical point* (or simply a *critical point*).

If, in addition, $\nabla^2 f(x_*) \ge 0$ we call x_* a *second-order critical point.*

Proposition 2.10 (Sufficient Optimality Conditions). *Suppose that* $f : \mathbb{R}^N \to \mathbb{R}$ *is twice differentiable at* x_*, *that* $\nabla f(x_*) = 0$ *and that* $\nabla^2 f(x_*) > 0$. *Then,* x_* *is a strict local minimizer of* f.

Proof. First note that $\nabla^2 f(x_*) > 0$ implies that there exists a $C_0 > 0$ such that

$$h^T \nabla^2 f(x_*) h \ge C_0 |h|^2, \qquad \text{for all} \;\; h \in \mathbb{R}^N.$$

(It follows from Item 5 on the eigenvalue handout that we can choose $C_0 = \min_{n=1}^N \lambda_n$.)

Now, let $h \neq 0$. Then

$$f(x_* + h) - f(x_*) = \underbrace{\nabla f(x_*)}_{=0} \cdot h + \tfrac{1}{2} h^T \nabla^2 f(x_*) h + o(|h|^2) \geq |h|^2 \left(\frac{C_0}{2} + \frac{o(|h|^2)}{|h|^2} \right)$$

Since $o(x)/x \to 0$ as $x \to 0$, there exists a $r > 0$ such that $\left|\frac{o(|h|^2)}{|h|^2}\right| < \frac{C_0}{4}$, for all $|h|^2 \leq r^2$. Hence, setting $x = x_* + h$, we get

$$f(x) - f(x_*) \geq \frac{C_0}{4} |x - x_*|^2 > 0, \quad \text{for all } \; x \in B_r(x_*) \backslash \{x_*\} .$$

□

edited 19 Feb

Note that $\nabla^2 f(x_*) > 0$ is not a necessary condition for x_* being a local minimizer. For instance, $x_* = 0$ is a local minimizer of $f(x) = x^4$ but $f''(x_*) = 0$. Higher order derivative tests are possible in theory, but impractical for large scale problems.

end edit

2.4 Convergence rates

All methods we consider in this course will be *iterative*, i.e., we construct sequences $(x_n)_{n \in \mathbb{N}}$ converging to some limit x_*, typically the solution of a minimisation problem. The speed of convergence has an immediate impact on the cost of the method. To measure the speed of convergence of sequences, we introduce the notion of *convergence rate.*

Definition 2.11 (Convergence Rates). Let $(x_n) \subset \mathbb{R}^N$ and $x_* \in \mathbb{R}^N$.

(i) We say that $x_n \to x_*$ with *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 $x_n \to x_*$ *quadratically.*

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

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

More precisely, this notion of convergence is called the *q-order.* The sequences are said to converge *q-quadratically*, *q-superlinearly* and *q-linearly* with *q-factor* σ. Weaker notions of convergence exist, but the q-order will be sufficient for our purposes here.

Example 2.12. Let $\rho \in (0,1)$.

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

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

Example 2.13. Suppose that $x_n \to x_*$, that $|x_0 - x_*| = 1/2$ and that we terminate when $|x_n - x_*| \leq 10^{-10}$.

- If $x_n \to x_*$ linearly with convergence factor $\sigma = 1/2$ (which is quite fast), then we would terminate after 33 iterations.
- If $x_n \to x_*$ quadratically with constant $K = 1$, then only 6 iterations suffice.

3 Newton's Method

The typical approach to solve an optimisation problem analytically is to derive the first-order criticality condition $\nabla f(x) = 0$, to find all its solutions, and then to discard those which are not (local) minimizers. Therefore, we first consider the following problem:

$$\text{Given } F : \mathbb{R}^N \to \mathbb{R}^N, \text{ find } x \in \mathbb{R}^N \text{ such that } F(x) = 0.$$

Solving this directly is *in general impossible.* However, using the interpretation of the derivative as a linear approximation, we can approximate the nonlinear system $F(x) = 0$ (locally) by a linear system.

Starting with an initial guess x_0 for a root x_*. If F is continuously differentiable in a neighbourhood U of x_0 that contains x_*, then

$$0 = F(x_*) = F(x_0) + DF(x_0)(x_* - x_0) + o(|x_* - x_0|). \tag{8}$$

Provided $|x_* - x_0|$ is sufficiently small, we can neglect $o(|x_* - x_0|)$ and solve

$$0 = F(x_0) + DF(x_0)(x_1 - x_0)$$

for x_1. Due to (8), we may expect x_1 to be closer to x_* than x_0. We can iterate the idea to obtain a sequence $(x_n)_{n=0}^{\infty}$ that (hopefully) converges to x_*.

Algorithm 3.1 (Newton's Method).

```
Input: $x_0 \in \mathbb{R}^N$
for $k = 0, 1, 2, \ldots$ do
    $x_{k+1} \leftarrow x_k - DF(x_k)^{-1} F(x_k)$
end for
```

The two basic assumptions in the motivation of this algorithm were (i) that $DF(x_k)$ is invertible for all k, and (ii) that x_0 is sufficiently close to a root x_*. We will now make these assumptions more precise. The following lemma is essentially a corollary of Lemma 2.1.

Lemma 3.1. *Let $x_* \in \mathbb{R}^N$ and $R > 0$ such that $DF(x)$ exists and is Lipschitz continuous in $\bar{B}_R(x_*)$. If, furthermore, $DF(x_*)$ is invertible then there exists $r > 0$ such that $DF(x)$ is invertible and*

$$\|DF(x)^{-1}\| \le 2\|DF(x_*)^{-1}\|, \qquad \textit{for all } \ x \in \bar{B}_r(x_*).$$

Proof. Let $L, R > 0$ be such that F is differentiable in $\bar{B}_R(x_*)$ and such that

$$\|DF(x) - DF(x_*)\| \le L|x - x_*|, \qquad \text{for all } x \in \bar{B}_R(x_*)\,.$$

Now, we let $S = DF(x)$, $T = DF(x_*)$ and $\sigma := \|DF(x_*)^{-1}\|$. The Lipschitz condition above implies that $\|T - S\| \le Lr$, for any $r \le R$. Thus, by choosing $r = \min\left(R, \frac{1}{2L\sigma}\right)$ we have

$\|T - S\| \leq \frac{1}{2\sigma} < 1/\|T^{-1}\|$ and we can apply Lemma 2.1. It follows that $S = DF(x)$ is invertible and

$$\|DF(x)^{-1}\| = \|S^{-1}\| \leq \frac{\sigma}{1 - \sigma\|T - S\|} \leq \frac{\sigma}{1 - \frac{1}{2}} = 2\|DF(x_*)^{-1}\|\,, \quad \text{for all} \;\; x \in \bar{B}_r(x_*).$$

□

From Lemma 3.1 it follows that, in the neighbourhood of a root x_*, Newton's method is well-defined. We shall now verify that for a sufficiently close initial guess, the iterates produced by Newton's Method do not leave this neighbourhood and in fact converge to the root x_*.

Theorem 3.2 (Newton Convergence). *Suppose $U \subset \mathbb{R}^N$ is open, $F \in \mathrm{C}^1(U; \mathbb{R}^N)$ 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 any $x_0 \in \bar{B}_R(x_*)$, Newton's Method is well-defined and converges quadratically to x_*.*

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

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

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

Applying Theorem 2.5 (IMVT), we can expand

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

Note that $x_* + t(x_k - x_*) \in \bar{B}_r(x_*)$, for all $t \in [0, 1]$, so that $DF(x_* + t(x_k - x_*))$ exists. Together, these two equations give

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

Hence, multiplying by $DF(x_k)^{-1}$, taking norms and applying Lemma 3.1, we obtain

$$\begin{aligned} |x_{k+1} - x_*| &\leq \|DF(x_k)^{-1}\| \left\| \int_0^1 \big(DF(x_k) - DF(x_* + t(x_k - x_*))\big)\,\mathrm{d}t \right\| |x_k - x_*| \\ &\leq 2\sigma \int_0^1 \big\|DF(x_k) - DF(x_* + t(x_k - x_*))\big\|\,\mathrm{d}t |x_k - x_*| \\ &\leq 2\sigma L \int_0^1 \big|(1 - t)(x_k - x_*)\big|\,\mathrm{d}t |x_k - x_*| = \sigma L |x_k - x_*|^2 \qquad (9) \end{aligned}$$

In particular, if $r \leq 1/(2L\sigma)$ (cf. proof of Lemma 3.1) then $|x_{k+1} - x_*| \leq \frac{1}{2}|x_k - x_*|$. It follows by induction that the sequence $(x_k)_{k \geq 0}$ remains inside $\bar{B}_r(x_*)$ provided $x_0 \in \bar{B}_r(x_*)$. Also,

$$|x_{k+1} - x_*| \leq \tfrac{1}{2}|x_k - x_*| \leq \big(\tfrac{1}{2}\big)^2 |x_{k-1} - x_*| \leq \ldots \leq \big(\tfrac{1}{2}\big)^{k+1} |x_0 - x_*|$$

and so $x_k \to x_*$ as $k \to \infty$. Due to (9) the convergence is quadratic. □

- There are many alternative convergence proofs with slightly different assumptions.
- If F is differentiable, but $DF(x_*)$ is singular, Newton's Method still converges *typically*, but *only linearly* (cf. Problem Sheet 2).
- Newton's Method is **not globally** convergent. See Problem Sheet 2 for an example.

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. **Step length.** This is the most common termination criterion. Note that, under the assumptions of Theorem 3.2, $|x_{n+1} - x_*| = O(|x_n - x_*|^2)$. Hence,

   $$|x_{n+1} - x_n| = |x_n - x_*| + O(|x_n - x_*|^2).$$

   and we could terminate the method as soon as $|x_{n+1} - x_n|$ falls below a certain tolerance.

   Problem: We may prematurely terminate the iteration even if x_n is not close to a regular root, or if the root is singular.

2. **Residual norm:** We could terminate Newton's method as soon as $|F(x_n)|$ falls below a prescribed tolerance.

   Problem: Again premature termination. For example, if $F(x) = e^x$, then the algorithm will terminate without recognising that there is in fact no root.

3. In practice, 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, globally it is far from an ideal method:

- it may converge slowly or not at all if the starting guess is not good;
- it may converge to local maxima or saddle points;
- it requires explicit knowledge of the derivative of F.

A possible solution for the first two issues is to formulate algorithms which ensure that the objective function *decays in each iteration*. There are two classes of algorithms we shall cover:

- **line search methods** – computing at each step a descent direction and carrying out a 1D search along this direction to find a new iterate with lower objective function;
- **trust region methods** – building at each step a quadratic model of the objective function and minimising this model in a neighbourhood of the current iterate.

4.1 The basic steepest descent algorithm

Definition 4.1. 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 $|s| = 1$: Find $\hat{s} \in \mathbb{R}^N$ with $|\hat{s}| = 1$ such that

$$\nabla f(x) \cdot \hat{s} \le \nabla f(x) \cdot s \qquad \text{for all } \; s \in \mathbb{R}^N, \quad |s| = 1. \tag{10}$$

Proposition 4.2.

$$\hat{s} = -\frac{\nabla f(x)}{|\nabla f(x)|}\,.$$

Proof. Let $s \in \mathbb{R}^N$ with $|s| = 1$. Then, using the Cauchy–Schwarz Inequality (5),

$$\nabla f(x) \cdot \hat{s} = -\,|\nabla f(x)| = -|\nabla f(x)||-s| \le \nabla f(x) \cdot s.$$

□

Exercise. When is the Newton direction $s = -\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)$ a descent direction? Is $\nabla^2 f(x) > 0$ a necessary/sufficient condition?

Proposition 4.2 motivates the choice $s = -\nabla f(x) \neq 0$ (if x is not yet a minimum). If we take a small step in this direction, then the objective function will decrease strictly. More precisely,

$$f(x + \alpha s) = f(x) - \alpha |\nabla f(x)|^2 + o(\alpha). \tag{11}$$

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

Unfortunately, monotonicity of $f(x_n)$ is not sufficient to obtain convergence (in general). Instead, we shall impose the slightly stronger *sufficient descent condition* (or *Armijo condition*)

$$f(x + \alpha s) \leq f(x) + \theta_{sd} \alpha \nabla f(x) \cdot s, \tag{12}$$

where $\theta_{sd} \in (0, 1)$ is a user-defined parameter (typically very small, e.g. $\theta_{sd} = 10^{-3}$).

However, for faster convergence we would like to take as big steps as possible. The following *backtracking line search* method takes small steps only if required to satisfy (12).

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. (12) is satisfied

```
$\alpha \leftarrow 1$;
while $f(x + \alpha s) > f(x) + \theta_{sd} \alpha \nabla f(x) \cdot s$ do
    $\alpha \leftarrow \alpha/2$;
end while
return $\alpha$;
```

Generalising (11) to

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

it follows immediately that the while loop in Algorithm 4.1 terminates for sufficiently small α (cf. proof of Theorem 4.3 below). There are many different and more sophisticated linesearch algorithms (see Section 6.4 for an example).

Algorithm 4.2 (Basic Steepest Descent).
Input: $x_0 \in \mathbb{R}^N$, $\theta_{\mathrm{sd}} \in (0, 1)$

```
for $n = 0, 1, 2, \dots$ do
    $s_n \leftarrow -\nabla f(x_n)$;
    $\alpha_n \leftarrow$ LINESEARCH$[x = x_n,\ s = s_n,\ \theta_{sd}]$;
    $x_{n+1} \leftarrow x_n - \alpha_n \nabla f(x_n)$;
end for
```

Theorem 4.3 (Global Convergence of Steepest Descent). *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ is bounded below and that ∇f is globally Lipschitz continuous.*

(a) For any $\theta_{sd} \in (0,1)$ and $x_0 \in \mathbb{R}^N$,

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

(b) If, in addition, f is coercive, that is, $\lim_{|x|\to\infty} f(x) = +\infty$, then there exists a convergent subsequence $x_{n_k} \to x_*$ with $\nabla f(x_*) = 0$.

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 be the Lipschitz constant of ∇f on $\mathbb{R}^N$. Due to the backtracking linesearch, (12) holds with α_n from Algorithm 4.1, i.e.

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

from which we immediately deduce

$$\theta_{\mathrm{sd}} \sum_{n=0}^{\infty} \alpha_n |\nabla f(x_n)|^2 \le \sum_{n=0}^{\infty} f(x_n) - f(x_{n+1}) = f(x_0) - f_*.$$

If we can show that the sequence $(\alpha_n)_{n\ge 0}$ is bounded from below by some $\underline{\alpha} > 0$, then

$$\sum_{n=0}^{\infty} |\nabla f(x_n)|^2 \le \frac{f(x_0) - f_*}{\underline{\alpha}\,\theta_{\mathrm{sd}}} < +\infty\,,$$

and (13) follows.

Let $s_n = -\nabla f(x_n)$. Then, for any $\alpha \in (0,1]$,

$$\begin{aligned} f(x_n + \alpha s_n) &= f(x_n) + \alpha \nabla f(x_n) \cdot s_n + \alpha \int_0^1 \big(\nabla f(x_n + t\alpha s_n) - \nabla f(x_n)\big) \cdot s_n \,\mathrm{d}t \\ &\le f(x_n) - \alpha |\nabla f(x_n)|^2 + \alpha \int_0^1 L |t\alpha s_n||s_n| \,\mathrm{d}t \\ &= f(x_n) - \alpha \left(1 - \frac{\alpha L}{2}\right) |\nabla f(x_n)|^2 \end{aligned} \tag{14}$$

(using Prob. Sheet 1, Q. 2(a), the Cauchy-Schwarz inequality and Lipschitz continuity of ∇f). In particular, if

$$1 - \frac{\alpha L}{2} \ge \theta_{\mathrm{sd}} \quad \Leftrightarrow \quad \alpha \le \frac{2(1-\theta_{\mathrm{sd}})}{L}$$

then the sufficient decrease condition (12) is satisfied and the linesearch terminates. Since, α is reduced by a factor of 2 in each step, it follows that

$$\alpha_n \ge \underline{\alpha} := \min\left(1, \frac{1-\theta_{sd}}{L}\right) > 0, \quad \text{for all } \ n \ge 0.$$

To prove Part (b), we show first that $(x_n)_{n\geq 0}$ is bounded. Assume the converse, that is, there exists a subsequence $(x_{n_k})_{k\geq 0}$ such that $|x_{n_k}| \to \infty$ as $k \to \infty$. Then, $f(x_{n_k}) \to \infty$, as $k \to \infty$. But $f(x_n) < f(x_0) < \infty$, for all $n \in \mathbb{N}$, leading to a contradiction.

Since $(x_n)_{n\geq 0}$ is bounded, there exists a compact set $K \subset \mathbb{R}^N$ such that $(x_n)_{n\geq 0} \subset K$. The existence of a convergent subsequence $x_{n_k} \to x_*$ with $x_* \in K$ follows from the Bolzano-Weierstrass Theorem. Moreover, it follows from Part (a) that

$$\nabla f(x_*) = \lim_{k\to\infty} \nabla f(x_{n_k}) = 0,$$

which completes the proof. □

Remark 4.4.

(a) It is posible to relax the assumptions that f is bounded from below and that ∇f is global Lipschitz continuous.

(b) If this accumulation point x_* in Theorem 4.3 is a *strict local minimizer*, then $x_n \to x_*$ as $n \to \infty$. We skip the proof of this result.

Note that Theorem 4.3 does not adress the rates of convergence. A geometric picture of the poor performance of Steepest Descent, even for only slightly ill-conditioned problems, is shown in Figure 2. We will make this observation more precise in a simplified situation.

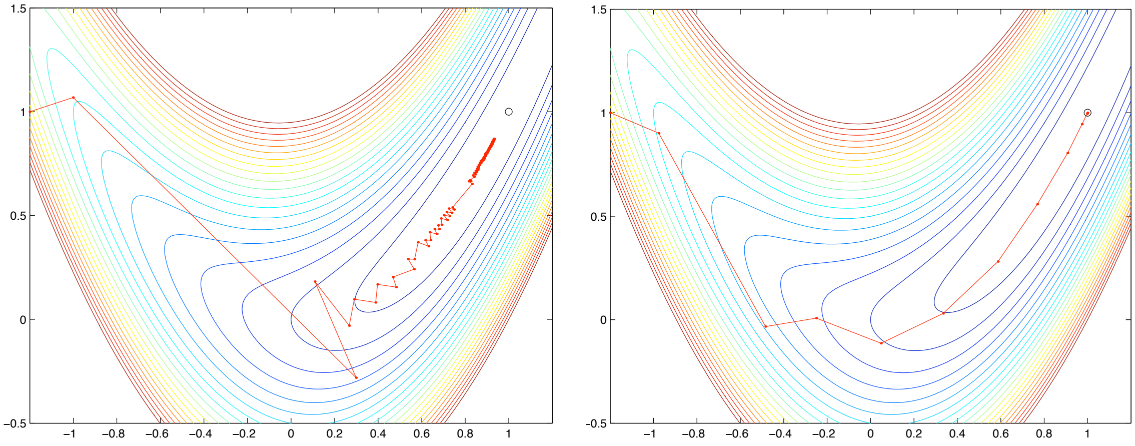

Figure 2: Steepest Descent (left) and Newton (right) iterates for the objective function $f(x, y) = 10(y - x^2)^2 + (x - 1)^2$ (the Rosenbrock function). Both methods behave similarly away from the solution, but Newton's method is significantly more efficient in the final steps. Without any second derivative information, the steepest descent method oscillates back and forth between the sides of the "energy valley".

Proposition 4.5. *Suppose that $f(x) = \frac{1}{2}x^T Ax$, where A is spd and $\|A\| \geq 1$. Let $(x_n)_{n\in\mathbb{N}}$ be the sequence generated by Algorithm 4.2. Then, $f(x_n) \to 0$ q-linearly with q-factor*

$$\sigma_{sd} := 1 - \frac{2\theta_{sd}(1 - \theta_{sd})}{\kappa(A)}$$

and $|x_n|_A \to 0$ q-linearly with q-factor $\sigma_{sd}^{1/2}$ (where $|y|_A = \sqrt{y^T Ay}$ is the energy norm *of y).*

Proof. Recall from the proof of Theorem 4.3 that

$$\alpha_n \geq \min\left(1, \frac{1-\theta_{sd}}{L}\right).$$

Here, the Lipschitz constant for $\nabla f(x) = Ax$ is $L = \|A\|$. Due to the assumption $\|A\| \geq 1$, we have $\alpha_n \geq (1-\theta_{sd})/\|A\|$. Together with the Armijo condition (12) this implies

$$f(x_{n+1}) \leq f(x_n) - \theta_{sd}\alpha_n |\nabla f(x_n)|^2 \leq f(x_n) - \frac{\theta_{sd}(1-\theta_{sd})}{\|A\|}|Ax_n|^2. \tag{15}$$

Now,

$$|Ay|^2 = (A^{1/2}y)^T A(A^{1/2}y) \geq \lambda_{\min}(A)|A^{1/2}y|^2 = \frac{y^T Ay}{\|A^{-1}\|} = \frac{2f(y)}{\|A^{-1}\|}$$

which together with (15) shows that

$$\tfrac{1}{2}|x_n|_A^2 = \tfrac{1}{2}x_n^T Ax_n = f(x_n) \to 0$$

q-linearly with q-factor $\sigma_{sd} = 1 - 2\theta_{sd}(1-\theta_{sd})/\kappa(A)$, and consequently $|x_n|_A \to 0$ q-linearly with q-factor $\sigma_{sd}^{1/2}$. □

Remark 4.6.

(a) By rescaling the problem, the result is also true for $\|A\| < 1$.

(b) Since $|x_n| = |A^{-1/2}A^{1/2}x_n| \leq \|A^{-1/2}\||x_n|_A$, we see that x_n also converges in the standard Euclidean norm $|\cdot|_2$. This type of convergence is called r-linear convergence.

(b) The result can be generalised easily to other $f \in \mathrm{C}^2(\mathbb{R}^N)$ with $\sup_{x\in\mathbb{R}^N} \|\nabla^2 f(x)\| < +\infty$. The convergence factor then depends on $\kappa(\nabla^2 f(x_*))$.

4.2 Variable-metric steepest descent

In the preceding section, we have seen that basic Steepest Descent is provably *globally convergent*, but the local performance can be very poor if $\kappa(\nabla^2 f(x_*)) \gg 1$. In this section, we aim to considerably improve this. First, let us generalise Theorem 4.3.

Theorem 4.7 (Global convergence of general descent methods). *Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ be bounded below and ∇f be globally Lipschitz continuous. Now, let s_n in Algorithm 4.2 be an arbitrary descent direction at x_n.*

(a) If there exists a constant $\delta > 0$ such that $|s_n| \geq \delta|\nabla f(x_n)|$, for all $n \geq 0$. Then, for any $\theta_{sd} \in (0,1)$ and $x_0 \in \mathbb{R}^N$, this generalised steepest descent method satisfies

$$\sum_{n=0}^{\infty} \cos(\theta_n)^2\, |\nabla f(x_n)|^2 < +\infty, \tag{16}$$

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

(b) If θ_n is bounded away from $\pi/2$, uniformly in n, and $\lim_{|x|\to\infty} f(x) = +\infty$, then there exists a convergent subsequence $x_{n_k} \to x_*$ with $\nabla f(x_*) = 0$.

Proof. Skipped, but similar to the proof of Theorem 4.3. □

Remark 4.8. The condition $|s_n| \geq \delta|\nabla f(x_n)|$ in Theorem 4.7 can be removed, e.g. by changing Line 1 in Algorithm 4.1 to

```
1: $\alpha \leftarrow \max(1, |\nabla f(x_n)|/|s_n|)$;
```

The generalisation of Algorithm 4.2 in Theorem 4.7 is only useful if we can provide a simple method to compute a better descent direction s_n.

The problem with $s_n = -\nabla f(x_n)$ was not the steepest descent idea itself, but the short steplengths, caused by the large local Lipschitz constant L for ∇f (cf. proof of Proposition 4.5). Recall the Newton direction

$$s_n = -\left[\nabla^2 f(x_n)\right]^{-1} \nabla f(x_n).$$

Hence, the problem in the steepest descent method was just a bad choice of norm that we measured distances in. (The size of L depends strongly on the norm, here $|\cdot|$.)

Let $B \in \mathbb{R}^{N\times N}$ be spd. Instead of minimising $\nabla f(x)\cdot s$ over all $s \in \mathbb{R}^N$ with $|s| = 1$, we now minimise over all $s \in \mathbb{R}^N$ with $|s|_B = 1$. (Recall $|x|_B := \left(x^T Bx\right)^{1/2}$, for any $B > 0$.)

Proposition 4.9. *The direction of steepest descent of f at x, with respect to the B-norm, is*

$$\hat{s} = -\frac{B^{-1}\nabla f(x)}{|B^{-1}\nabla f(x)|_B}.$$

It satisfies

$$\nabla f(x)\cdot\hat{s} \leq \nabla f(x)\cdot s, \quad \textit{for all } \; s \in \mathbb{R}^N, \quad |s|_B = 1.$$

Proof. Let $s \in \mathbb{R}^N$, $|s|_B = 1$. Then, using the Cauchy–Schwarz Inequality for the B-inner product,

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

□

For maximal flexibility in our choice of descent direction, we allow the *norm* $|\cdot|_B$ to change at each step of the descent method.

Algorithm 4.3 (Generalized Steepest Descent Method).

```
Input: $x_0$, $\theta_{sd}$
for $n = 0, 1, 2, \ldots$ do
    Choose an spd matrix $B_n \in \mathbb{R}^{N \times N}$;
    $s_n \leftarrow -B_n^{-1} \nabla f(x_n)$;
    $\alpha_n \leftarrow$ LINESEARCH$[x = x_n, s = s_n, \theta_{sd}]$;
    $x_{n+1} = x_n + \alpha_n s_n$
end for
```

Remark 4.10. The more common motivation for Algorithm 4.3 in the optimization literature is to assume that B_n is an approximation of $\nabla^2 f(x_n)$. Then, 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 $B_n > 0$, then m_n has a unique minimizer

$$x_* = x_n - B_n^{-1} \nabla f(x_n)\,.$$

So Algorithm 4.3 simply *damps* the steplengths to achieve global convergence.

Before discussing how to choose B_n, let us discuss the global and local convergence properties of Algorithm 4.3. First, it is possible to prove a similar global convergence result for Algorithm 4.3 as for Algorithm 4.2 in Theorem 4.3. In particular, if $\kappa(B_n) \le \overline{\kappa} < +\infty$, uniformly in $n \in \mathbb{N}$, and if the assumptions on f in Theorem 4.3 hold true, then

$$\sum_{n=0}^{\infty} |\nabla f(x_n)|^2_{B_n^{-1}} < +\infty\,,$$

for any $\theta_{sd} \in (0, 1)$ and $x_0 \in \mathbb{R}^N$. (We skip this proof. See [4, Thm. 4.12].) As before, if f is coercive then there exists a convergent subsequence $x_{n_k} \to x_*$ with $\nabla f(x_*) = 0$.

More interesting is how the local convergence speed changes. As before, we prove this only for the special case of a quadratic objective function f.

Proposition 4.11. *Let $A, B \in \mathbb{R}^{N \times N}$ be spd and such that $\lambda_{\max}\big(B^{-1/2} A B^{-1/2}\big) \ge 1$. Now, suppose that $f(x) = \frac{1}{2} x^T A x$ and that $(x_n)_{n \in \mathbb{N}}$ is generated by Algorithm 4.3 with $B_n = B$, for all $n \ge 0$. Then, $f(x_n) \to 0$ q-linearly with q-factor*

$$\sigma_{gsd} := 1 - \frac{2\theta_{sd}(1 - \theta_{sd})}{\kappa(B^{-1/2} A B^{-1/2})}$$

and $|x_n|_A \to 0$ q-linearly with q-factor $\sigma_{gsd}^{1/2}$.

Proof. First recall from Section 2.1 that, for any spd matrix $C \in \mathbb{R}^{N \times N}$, we have

$$\lambda_{\min}(C)|y|^2 \le y^T C y \le \lambda_{\max}(C)|y|^2, \quad \text{for any } \ y \in \mathbb{R}^N,$$

and $\kappa(C) = \lambda_{\max}(C)/\lambda_{\min}(C)$.

Now, let $s_n = -B^{-1}\nabla f(x_n)$. Then, since $B^{-1/2}AB^{-1/2}$ is spd, using the second inequality above, we get

$$\begin{aligned} s_n^T A s_n = &\left(B^{-1/2}\nabla f(x_n)\right)^T B^{-1/2}AB^{-1/2}\left(B^{-1/2}\nabla f(x_n)\right) \\ &\le \lambda_{\max}\,|B^{-1/2}\nabla f(x_n)| = -\lambda_{\max}\,\nabla f(x_n)\cdot s_n\,, \end{aligned} \tag{17}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $B^{-1/2}AB^{-1/2}$.

Using Prob. Sheet 1, Q. 2(a)) and the bound in (17), we have, for any $\alpha \in (0,1]$,

$$\begin{aligned} f(x_n + \alpha s_n) &= f(x_n) + \alpha\nabla f(x_n)\cdot s_n + \int_0^1 \big(\nabla f(x_n + t\alpha s_n) - \nabla f(x_n)\big)\cdot(\alpha s_n)\,\mathrm{d}t \\ &= f(x_n) + \alpha\nabla f(x_n)\cdot s_n + \alpha\int_0^1 (x_n + t\alpha s_n - x_n)^T A s_n\,\mathrm{d}t \\ &\le f(x_n) + \alpha\left(1 - \frac{\alpha\lambda_{\max}}{2}\right)\nabla f(x_n)\cdot s_n \end{aligned} \tag{18}$$

Thus, the sufficient decrease condition (12) is satisfied and the linesearch terminates, if α satisfies

$$\left(1 - \frac{\alpha\lambda_{\max}}{2}\right) \ge \theta_{sd} \quad \Leftrightarrow \quad \alpha \le \frac{2(1-\theta_{sd})}{\lambda_{\max}}$$

Since α is reduced by a factor 2 in each step of backtracking line search and since we assumed $\lambda_{\max} \ge 1$, it follows that

$$\alpha_n \ge \frac{1-\theta_{sd}}{\lambda_{\max}}\,, \quad \text{for all} \;\; n \ge 0. \tag{19}$$

In the same way as in (17), we can also show that

$$-\nabla f(x_n)\cdot s_n = x_n^T AB^{-1}Ax_n = (A^{1/2}x_n)^T A^{1/2}B^{-1}A^{1/2}(A^{1/2}x_n) \ge \lambda_{\min}\,\underbrace{|A^{1/2}x_n|^2}_{=\,2f(x_n)}\,,$$

where $\lambda_{\min}$ is the minimum eigenvalue of $A^{1/2}B^{-1}A^{1/2}$. Substituting this bound into (18) and using the bound on α_n in (19), we get

$$f(x_{n+1}) \le \left(1 - 2\theta_{sd}(1-\theta_{sd})\frac{\lambda_{\min}}{\lambda_{\max}}\right) f(x_n)\,.$$

The result follows since the matrices $A^{1/2}B^{-1}A^{1/2}$ and $B^{-1/2}AB^{-1/2}$ have the same eigenvalues. Indeed, for any eigenpair (λ, z) of $A^{1/2}B^{-1}A^{1/2}$, we have

$$\left(B^{-1/2}AB^{-1/2}\right)\left(B^{-1/2}A^{1/2}\right)z = \left(B^{-1/2}A^{1/2}\right)\left(A^{1/2}B^{-1}A^{1/2}\right)z = \lambda\left(B^{-1/2}A^{1/2}\right)z$$

and thus, $(\lambda, B^{-1/2}A^{1/2}z)$ is an eigenpair of $B^{-1/2}AB^{-1/2}$. □

Proposition 4.11 indicates that if $\kappa(B^{-1/2}AB^{-1/2}) \ll \kappa(A)$ then the q-factor can be significantly lowered by this *"preconditioning"* process, leading to a much faster convergence of Algorithm 4.3. In particular, if $B = \nabla^2 f(x) = A$ (Newton) then $\kappa(B^{-1/2}AB^{-1/2}) = 1$. The result can again be generalised to other $f \in C^2(\mathbb{R}^N)$ with $\sup_{x\in\mathbb{R}^N}\kappa(\nabla^2 f(x)) < +\infty$.

Wish list for the choice of B_n:

(B.1) B_n must be spd to guarantee that $|\cdot|_{B_n}$ is a norm and that $s_n = -B_n^{-1}\nabla f(x_n)$ is a descent direction, i.e.

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

(B.2) Ideally $B_n \approx \nabla^2 f(x_n)$ to mimic Newton's method, or more generally, to achieve moderate $\kappa(B_n^{-1/2}\nabla^2 f(x_n) B_n^{-1/2})$.

(B.3) The matrix-vector products $B_n x$ and $B_n^{-1}x$ should be cheap.

Common choices:

1. *Newton's Method:* The choice $B_n = \nabla^2 f(x_n)$ is only possible in rare cases, e.g. globally convex f. But whenever possible, it should be used. The line-search aspect then leads to global convergence (*locally quadratic*).

2. *User-defined Metric B_n,* e.g. using analytical insight, such that $\kappa(B_n^{-1/2}\nabla^2 f(x_n) B_n^{-1/2})$ is moderate for all n.

3. *Damped Newton (the Levenberg–Marquardt Method):* Choose $B_n = \nabla^2 f(x_n) + \mu_n E$ with E spd and $\mu_n \geq 0$ (the Levenberg–Marquardt parameter) adjusted so that B_n is spd. The matrix E should again be chosen by the user such that $\kappa(E^{-1/2}\nabla^2 f(x) E^{-1/2})$ is moderate for all x (but it can also be $E = I$ for simplicity).

   If eventually $\mu_n = 0$, for $n \geq n_0$, then it reduces to Newton's method and therefore exhibits locally quadratic convergence. Typically, one chooses $\mu_n \searrow 0$ leading to superlinear convergence.

4. *Quasi-Newton Methods:* In practice it is often difficult, expensive or impossible to compute and invert the Hessian $\nabla^2 f(x_n)$ at every step. Instead, one can use quantities already computed in the optimisation process to approximate $\nabla^2 f(x_n)$ (see Section 6).

5 Trust Region Methods

Idea:

1. In each iteration, replace the objective function $f(x)$ by a quadratic *model* $m_n(x)$.
2. Choose a neighbourhood R_n of x_n where m_n is *trusted* to approximate f well.
3. Find x_{n+1} by (approximately) minimising m_n over the trust region R_n:

$$x_{n+1} \approx \operatorname*{argmin}_{x \in R_n} m_n(x) \tag{20}$$

Note that the *trust region subproblem* (20) is a *contrained* optimisation problem. Normally (see below), we do the opposite, i.e. replace a constrained problem by a sequence of unconstrained ones. Thus, (20) can only be solved efficiently if R_n is very simple.

To recover the local convergence speed of Newton's method, we 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 approximate the Hessian $\nabla^2 f(x_n)$).

IMPORTANT. Since we are minimizing m_n over a bounded region R_n, **we do not require anymore that H_n is positive definite (or even invertible) !**

The trust region R_n is typically a closed ball in some norm. Here, for simplicity always

$$R_n = \{x \in \mathbb{R}^N : |x - x_n| \leq \Delta_n\} \quad \text{with} \quad \Delta_n > 0,$$

the *trust region radius*. It reflects the quality of the model m_n and is adjusted at each step.

Algorithm 5.1 (Prototype Trust Region Method).

```
Input: $x_0$, $\Delta_0$
for $n = 0, 1, 2, \ldots$ do
    Compute $\nabla f(x_n)$ and $H_n$;
    Approximately solve (20) to obtain candidate $x'_{n+1}$;
    Decide whether to accept ($x_{n+1} = x'_{n+1}$) or reject ($x_{n+1} = x_n$) the candidate;
    Adjust trust region radius to get new radius $\Delta_{n+1}$;
end for
```

Let us now adress each step of the trust region method separately.

5.1 The Cauchy-point

The most crucial point is how to solve the trust region subproblem. This is still the subject of ongoing research.

The main objective for the trust region algorithm was to ensure global convergence. Thus, a simple idea for the (approximate) solution of the trust region subproblem is to ensure that the method is **at least** as efficient as basic steepest descent. The *Cauchy point* is obtained when basic steepest descent is applied to m_n at x_n and restricted to R_n. Since m_n is quadratic, we can use exact line search and restrict the search to $\alpha \in \big(0, \frac{\Delta_n}{|\nabla f(x_n)|}\big]$, i.e. the intersection of the half-line $\{x_n + \alpha s_n : \alpha > 0\}$ with R_n.

Let $s_n = -\nabla f(x_n)$, then

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

Since $\phi'(\alpha) = -|\nabla f(x_n)|^2 + \alpha \nabla f(x_n)^T H_n \nabla f(x_n)$, the minimizer of $\phi(\alpha)$ over $\alpha \in \big(0, \frac{\Delta_n}{|\nabla f(x_n)|}\big]$ is given by

$$\alpha_n^c = \begin{cases} \dfrac{\Delta_n}{|\nabla f(x_n)|}, & \text{if } \nabla f(x_n)^T H_n \nabla f(x_n) \leq \dfrac{|\nabla f(x_n)|^3}{\Delta_n}, \\[2ex] \dfrac{|\nabla f(x_n)|^2}{\nabla f(x_n)^T H_n \nabla f(x_n)}, & \text{otherwise.} \end{cases} \tag{21}$$

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

$$x_n^c = x_n - \alpha_n^c \nabla f(x_n). \tag{22}$$

All methods discused below will require at least as much reduction of the model as the Cauchy point. We will see that this guarantees global convergence. For indefinite Hessians, $\nabla f(x_n)^T H_n \nabla f(x_n)$ may be ≤ 0, in which case $\phi'(\alpha) < 0$ and Cauchy point is at the boundary. edited 22 Apr

5.2 Accepting/rejecting updates & trust region radius management

To decide whether a candidate computed in the trust region subproblem is indeed a good iterate, we compare the decrease in m_n with the actual decrease in f:

$$\rho_n = \frac{f(x_n) - f(x'_{n+1})}{m_n(x_n) - m_n(x'_{n+1})}. \tag{23}$$

The ratio ρ_n between *actual reduction* and *predicted reduction* also tells us whether m_n is "trustworthy" in R_n and therefore can be used to adjust the trust region radius Δ_n if necessary.

Let $\rho_{ac} \in (0, 1/4)$ and $\Delta_{max} > 0$ be two user-defined parameters. We use the following heuristics to decide whether to accept the candidate x'_{n+1} and to adjust the radius Δ_{n+1}:

Accepting/rejecting x'_{n+1}		Radius management	
$\rho_n \geq \rho_{ac}$	$x_{n+1} = x'_{n+1}$	$\rho_n < \frac{1}{4}$	$\Delta_{n+1} = \frac{1}{4}\Delta_n$
$\rho_n < \rho_{ac}$	$x_{n+1} = x_n$	$\rho_n > \frac{3}{4}$ and $\lvert x'_{n+1} - x_n\rvert = \Delta_n$	$\Delta_{n+1} = \min(2\Delta_n, \Delta_{\max})$
		Otherwise	$\Delta_{n+1} = \Delta_n$

In particular, x'_{n+1} is always rejected and Δ_{n+1} always decreased if $\rho_n \leq 0$ (or equivalently if $f(x'_{n+1}) \geq f(x_n)$).

Together with the idea of the Cauchy Point this leads to the basic trust region algorithm:

Algorithm 5.2 (Trust Region Method).

```
Input: $x_0$, $\Delta_0$, $\Delta_{\max}$, $\rho_{ac}$;
for $n = 0, 1, 2, \dots$ do
    Compute $\nabla f(x_n)$ and $H_n$;
    Compute approximate minimizer $x'_{n+1}$ of (20) with $m_n(x'_{n+1}) \leq m_n(x^c_n)$;
    Evaluate $\rho_n$ in (23);
    if $\rho_n \geq \rho_{ac}$ then
        $x_{n+1} \leftarrow x'_{n+1}$;
    else
        $x_{n+1} \leftarrow x_n$;
    end if
    if $\rho_n < 1/4$ then
        $\Delta_{n+1} \leftarrow \frac{1}{4}\Delta_n$
    else if $\rho_n > \frac{3}{4}$ and $|x'_{n+1} - x_n| = \Delta_n$ then
        $\Delta_{n+1} \leftarrow \min(2\Delta_n, \Delta_{max})$;
    else
        $\Delta_{n+1} \leftarrow \Delta_n$;
    end if
end for
```

5.3 Global convergence of trust region methods

Before proving convergence, let us establish the fact that x'_{n+1} is accepted for sufficiently small trust region radius Δ_n. As stated above, we assume that the solution x'_{n+1} of the trust region subproblem leads to at least as much reduction in the quadratic model than the Cauchy point, i.e.

$$m_n(x'_{n+1}) \leq m_n(x^c_n), \quad \text{for all} \;\; n \geq 0. \tag{24}$$

Lemma 5.1. *Let $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$ and ∇f be Lipschitz continuous with Lipschitz constant L. Let x_n be the nth iterate in Algorithm 5.2. Suppose that $\nabla f(x_n) \neq 0$ and that $\|H_n\| < +\infty$. If*

$$\Delta_n \leq \frac{3}{4} \frac{|\nabla f(x_n)|}{L + \|H_n\|},$$

then

(i) $\alpha^c_n = \Delta_n \,/\, |\nabla f(x_n)|$ and $m_n(x_n) - m_n(x^c_n) \geq \frac{1}{2}\Delta_n |\nabla f(x_n)|$;

(ii) if, in addition, (24) holds then $\rho_n \geq 1/4$ and x'_{n+1} in Algorithm 5.2 is accepted.

Proof. See Problem Sheet 5. □

This leads to the following global convergence result.

Theorem 5.2. *Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ be bounded from below and let ∇f be Lipschitz continuous. Consider the sequence of iterates $(x_n)_{n\geq 0}$ that is produced by Algorithm 5.2. If* (24) *holds and* $\max_{n\in\mathbb{N}} \|H_n\| =: \beta < +\infty$ *then*

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

Proof. By assumption, there exists $M \in \mathbb{R}$ such that $f(x) \geq M$, for all $x \in \mathbb{R}^N$. Suppose, for contradiction, that $|\nabla f(x_n)| \geq \varepsilon > 0$, for all $n \geq 0$. Due to Lemma 5.1, if

$$\Delta_n \leq \frac{3}{4}\frac{\epsilon}{L+\beta} \leq \frac{3}{4}\frac{|\nabla f(x_n)|}{L+\|H_n\|}$$

then $\alpha_n^c = \Delta_n \,/\, |\nabla f(x_n)|$, $\rho_n \geq \frac{1}{4}$, x'_{n+1} is accepted and $\Delta_{n+1} \geq \Delta_n$. Since $\Delta_{n+1} \geq \frac{1}{4}\Delta_n$, for all $n \in \mathbb{N}$ in Algorithm 5.2, this implies

$$\Delta_n \geq \min\left(\Delta_0, \frac{3}{16}\frac{\epsilon}{L+\beta}\right) =: \Delta_{min} > 0\,, \quad \text{for all} \;\; n \geq 0.$$

Let $n \geq 0$ be an index where x'_{n+1} is accepted. Then $\rho_n \geq \rho_{ac}$, i.e.

$$f(x_n) - f(x_{n+1}) \geq \rho_{ac}(m_n(x_n) - m_n(x'_{n+1})) \geq \rho_{ac}(m_n(x_n) - m_n(x_n^c)). \tag{25}$$

Since $\Delta_{\min} \,/\, |\nabla f(x_n)| \leq \alpha_n^c$ and since the function $\alpha \mapsto m_n(x_n - \alpha\nabla f(x_n))$ is strictly decreasing on $[0, \alpha_n^c]$, it follows from Lemma 5.1(i) with $\Delta_n = \Delta_{\min}$ that

$$m_n(x_n) - m_n(x_n^c) \geq m_n(x_n) - m_n\left(x_n - \frac{\Delta_{min}}{|\nabla f_n|}\nabla f_n\right) \geq \tfrac{1}{2}\Delta_{min}|\nabla f_n|$$

Substituting this into (25), we conclude that, whenever a guess x'_{n+1} is accepted,

$$f(x_n) - f(x_{n+1}) \geq \tfrac{1}{2}\rho_{ac}\Delta_{min}|\nabla f(x_n)|.$$

In particular, if $(n_k)_{k\geq 0}$ is the subsequence of all those indices n where x'_{n+1} is accepted then

$$\sum_{j=0}^{\infty} |\nabla f(x_{n_j})| \leq 2\frac{f(x_{n_0}) - M}{\rho_{ac}\,\Delta_{min}} < +\infty,$$

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 x'_{n+1} of the trust region subproblem satisfies $m_n(x'_{n+1}) \leq m_n(x_n^c)$. For example, taking $x'_{n+1} = x_n^c$ 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. We need to find a more sophisticated way of computing x'_{n+1}. The real advantage of the trust region framework is the ease with which this can be achieved.

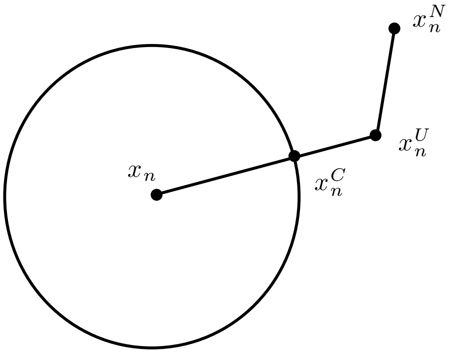

Figure 3: The dogleg path.

Ideally, we would like to use the Newton point whenever possible and revert to the Cauchy point otherwise. The *dogleg method* is a practical version of this strategy that aims to use only freely available information. For simplicity, we assume that $H_n = \nabla^2 f(x_n)$ and write $\nabla f_n := \nabla f(x_n)$, for all $n \geq 0$, for the remainder of this section.

Idea:

1. If $\nabla f_n^T H_n \nabla f_n \leq 0$, we take $x'_{n+1} = x_n^c = x_n - \dfrac{\Delta_n}{|\nabla f_n|}\nabla f_n$.

2. If $\nabla f_n^T H_n \nabla f_n > 0$, we attempt to compute the Newton point

$$x_n^N := x_n - H_n^{-1}\nabla f_n\,.$$

   If the solve with H_n fails, we again take $x'_{n+1} = x_n^c$.
   (This may only mean that H_n is ill-conditioned rather than singular, but in that case the Newton direction will likely be a poor choice anyway.)

3. If $\nabla f_n^T H_n \nabla f_n > 0$ and H_n is invertible, then the Newton point x_n^N and the *unidirectional minimizer* (along the steepest descent direction)

$$x_n^U := x_n - \frac{|\nabla f_n|^2}{\nabla f_n^T H_n \nabla f_n}\nabla f_n$$

   are well-defined. We take those as nodes in the piecewise linear *dogleg path* (see Fig. 3)

$$\Gamma_n = \{x_n + t(x_n^U - x_n) : 0 \leq t \leq 1\} \;\cup\; \{x_n^U + t(x_n^N - x_n^U) : 0 \leq t \leq 1\}$$

   and choose x'_{n+1} to be the minimizer of m_n in R_n along this path.

The following lemma is very useful to compute the minimiser along the dogleg path.

Lemma 5.3. *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. If*

$$(x_n^N - x_n^U) \cdot (x_n^U - x_n) > 0, \tag{26}$$

then the distance from x_n is strictly increasing along Γ_n and the trust region model m_n is strictly decreasing. If (26) is violated and $x_n^N \neq x_n^U$, then H_n is **not** *positive definite.*

Proof. Skipped.

[*Sketch:* It is an easy exercise to show that (26) implies $|x_n^U - x_n| < |x_n^N - x_n|$. The fact that the distance is then strictly increasing is a simple geometric exercise. Also, m_n is strictly decreasing along the first "leg" of the dogleg by definition. To show this for the second "leg", consider $\phi(t) := m_n(x_n^U + t(x_n^N - x_n^U))$ and prove that $\phi'(t) < 0$, for all $t \in [0,1]$. The final result can be established via some linear algebra.] □

We now have a clear strategy how to approximately solve the trust region subproblem:

$$x'_{n+1} = \begin{cases} \underset{x \in \Gamma_n \cap R_n}{\text{argmin}}\ m_n(x), & \text{if } \nabla f_n^T H_n \nabla f_n > 0 \text{ and } H_n \text{ is invertible and (26) holds,} \\ x_n^c, & \text{otherwise.} \end{cases} \tag{27}$$

On Problem Sheet 5, you are asked to formulate the details of this strategy in a pseudo-code.

With this choice of trust region subproblem solution, the dogleg method automatically switches from the globally convergent steepest descent method to a quadratically convergent Newton method, whenever it is convenient. We skip this result.

[*Sketch:* Since Lemma 5.3 implies $m_n(x'_{n+1}) \leq m_n(x_n^c)$, for all $n \geq 0$, the global convergence follows immediately from Theorem 5.2. To prove locally quadratic convergence, we show that, for n sufficiently big, we always have $\rho_n \geq \frac{1}{4}$ and $\Delta_n \geq \frac{3}{4}|x_n - x_*|$, so that $x'_{n+1} = x_n^N$ and the iteration reduces to Newton's method.]

There are many alternative methods to find good approximate minimizers of the trust region subproblem (20). See Nocedal & Wright [3] for some other approaches, such as *Steihaug's method*, which is based on the Conjugate Gradient method.

6 Quasi-Newton Methods

Let us return to the question of finding a good choice of metric for the variable-metric steepest descent method from Section 4.2, and discuss a powerful, general, and popular principle for constructing the matrices $B_n \approx \nabla^2 f(x_n)$, especially for large-scale systems. These will also provide good approximate Hessians H_n for the trust region method.

6.1 The Dennis–Moré condition for superlinear convergence

Let $(B_n)_{n\geq 0}$ be a family of invertible matrices. Let us start by considering the iteration

$$x_{n+1} = x_n - B_n^{-1}\nabla f(x_n), \quad \text{for all} \;\; n \geq 0. \tag{28}$$

By revisiting the proof of quadratic convergence of Newton's method we establish a condition on B_n that is necessary and sufficient to ensure superlinear convergence of the iteration above. We write for short $\nabla f_n = \nabla f(x_n)$ and $\nabla^2 f_n = \nabla^2 f(x_n)$ in this section.

Theorem 6.1. *Let $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$ and suppose that $x_* \in \mathbb{R}^N$ such that $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*) > 0$. Let $(x_n)_{n\in\mathbb{N}}$ be as defined in (28), and assume that $x_n \to x_*$ as $n \to \infty$. Then the convergence is superlinear if, and only if, the* Dennis–Moré condition

$$\lim_{n\to\infty} \frac{|(B_n - \nabla^2 f_n)s_n|}{|s_n|} = 0, \quad \textit{with} \;\; s_n = x_{n+1} - x_n\,, \tag{29}$$

is satisfied.

Proof. We prove only the sufficiency of (29) and assume for simplicity that $\nabla^2 f(x)$ is Lipschitz continuous in $B_R(x_*)$ with Lipschitz constant $L > 0$, for some $R > 0$.

Recall from Theorem 3.2 that, for n sufficiently large (and thus $|x_n - x_*|$ sufficiently small), the Newton step $x_n^N = x_n - (\nabla^2 f_n)^{-1}\nabla f_n$ is well-defined,

$$\|(\nabla^2 f_n)^{-1}\| \leq 2\sigma \quad \text{and} \quad |x_n^N - x_*| \leq L\sigma|x_n - x_*|^2.$$

where $\sigma := \|\nabla^2 f(x_*)^{-1}\|$. Let $s_n = x_{n+1} - x_n = -B_n^{-1}\nabla f_n$, then

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

and so there exists a $n_1 \in \mathbb{N}$ such that, for all $n \geq n_1$,

$$|x_{n+1} - x_*| \leq |x_n^N - x_*| + |x_n^N - x_{n+1}| \leq L\sigma|x_n - x_*|^2 + 2\sigma|(B_n - \nabla^2 f_n)s_n|\,. \tag{30}$$

Hence,

$$|s_n| \leq |x_n - x_*| + |x_{n+1} - x_*| \leq |x_n - x_*|\Big(1 + L\sigma|x_n - x_*|\Big) + 2\sigma|(B_n - \nabla^2 f_n)s_n|\,.$$

Since $x_n \to x_*$ and due to (29), there exists $n_2 \geq n_1$ such that, for all $n \geq n_2$,

$$|s_n| \leq 2|x_n - x_*| + \tfrac{1}{2}|s_n| \quad \Longleftrightarrow \quad |s_n| \leq 4|x_n - x_*|\,.$$

Combining this again with (30), then finally leads to

$$\frac{|x_{n+1} - x_*|}{|x_n - x_*|} \leq L\sigma|x_n - x_*| + 2\sigma\frac{|(B_n - \nabla^2 f_n)s_n|}{|x_n - x_*|} \leq L\sigma|x_n - x_*| + 8\sigma\frac{|(B_n - \nabla^2 f_n)s_n|}{|s_n|} \;\to\; 0,$$

as $n \to \infty$, due to (29) again, i.e. $x_n \to x_*$ q-superlinearly. □

Note. Importantly, we do **not** require $\|B_n - \nabla^2 f_n\| \to 0$ for superlinear convergence, only that the action of B_n in the direction $x_{n+1} - x_n$ decays sufficiently rapidly.

6.2 The secant condition and quasi-Newton updates

edited 22 Apr

In practice we use line search (cf. Section 4.2), i.e.

$$x_{n+1} - x_n = \alpha_n s_n =: d_n \,,$$

but, provided $\alpha_n \neq 0$, (29) is equivalent to

end edit

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

From now on we also denote

$$y_n := \nabla f_{n+1} - \nabla f_n \,.$$

Let us now establish a simple condition on B_n to satisfy (29).

Assuming again Lipschitz continuity of $\nabla^2 f(x)$ with constant L, it follows from Theorem 2.5 (IMVT) that, for n sufficiently big,

$$|y_n - \nabla^2 f_n d_n| = |\nabla f_{n+1} - \nabla f_n - \nabla^2 f_n d_n| = \left| \int_0^1 \left(\nabla^2 f(x_n + t d_n) - \nabla^2 f_n \right) d_n \, \mathrm{d}t \right|$$
$$\leq \int_0^1 \|\nabla^2 f(x_n + t d_n) - \nabla^2 f_n\| \, \mathrm{d}t \, |d_n| \leq \frac{L}{2} |d_n|^2.$$

Hence, ideally we would like to choose B_n such that $B_n d_n = y_n$, and then (29) holds.

However, this is difficult to enforce, since then B_n would depend on x_{n+1}. Instead, we require that the next *update* B_{n+1} satisfies

$$B_{n+1} d_n = y_n. \tag{31}$$

This condition is called the *secant condition.* If, in addition, $\|B_{n+1} - B_n\| \to 0$, then it can be shown fairly easily that the Dennis–Moré condition (29) holds and the resulting *quasi-Newton (QN) method* converges superlinearly.

We will now construct B_{n+1} by finding a simple update formula for B_n such that B_{n+1} is symmetric, (31) is satisfied and $B_{n+1} - B_n$ is minimised in a suitable norm. Since (31) constitutes a single constraint for this minimisation problem, we could search among all symmetric rank-1 perturbations $B_n \pm vv^T$ of B_n. It can be shown that there is in fact only one symmetric rank-1 matrix that satisfies (31), the **SR1 update:**

$$B_{n+1} = B_n + \frac{(y_n - B_n d_n)(y_n - B_n d_n)^T}{(y_n - B_n d_n)^T d_n}. \tag{32}$$

The SR1 update has two major shortcomings: (i) it is undefined (or numerically unstable) when $(y_n - B_n d_n)^T d_n = 0$ (or small) and (ii) $B_n > 0$ does not necessarily imply $B_{n+1} > 0$ which is problematic in variable-metric line search. It is still very useful within a trust region framework (not discussed here, see [4, Sec. 6.6] and [3, Sec. 8.2]).

As we will see below (Lemma 6.5), we can enforce $B_n > 0 \Rightarrow B_{n+1} > 0$, by adding the *curvature condition*

$$y_n^T d_n > 0, \quad \text{for all} \quad n \geq 0. \tag{33}$$

This can be ensured by a more advanced line search method (see Section 6.4).

Minimising $B_{n+1} - B_n$ among all symmetric matrices that satisfy (31) and (33) then leads to rank-2 update formulae. Different QN methods arise through different choices of the norm. The two most popular ones are (see [3, Sec. 8.1] for more details):

Davidon–Fletcher–Powell (DFP) update:

$$B_{n+1} = (I - \rho_n y_n d_n^T) B_n (I - \rho_n d_n y_n^T) + \rho_n y_n y_n^T, \quad \text{where} \quad \rho_n = \frac{1}{y_n^T d_n}. \tag{34}$$

(minimising $\|B - B_n\|_W$, where $\|\cdot\|_W$ denotes a suitably weighted Frobenius norm).

Broyden–Fletcher–Goldfarb–Shanno (BFGS) update:

$$B_{n+1} = B_n - \frac{(B_n d_n)(B_n d_n)^T}{d_n^T B_n d_n} + \frac{y_n y_n^T}{y_n^T d_n}. \tag{35}$$

(minimising $\|B^{-1} - B_n^{-1}\|_{W'}$ in a suitably weighted, but different, Frobenius norm).

6.3 The Sherman–Morrison–Woodbury formula

In this section, we will present a simple 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*

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

Proof. Left as an **Exercise**. □

Example. Let $B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, $U = \begin{bmatrix} -1 \\ 3 \end{bmatrix}$, $V = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$. Invert $B + UV^T = \begin{bmatrix} 3 & -1 \\ -6 & 5 \end{bmatrix}$.

$$1 + V^T B^{-1} U = 1 + [-2, 1] \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \end{bmatrix} = 1 + [-2, 1] \begin{bmatrix} -1 \\ 3/2 \end{bmatrix} = \frac{9}{2}.$$

Hence,

$$\begin{aligned}(B + UV^T)^{-1} &= \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} - \frac{2}{9} \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} -1 \\ 3 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} \begin{bmatrix} -2 \\ 1 \end{bmatrix} \right)^T \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix} - \frac{2}{9} \begin{bmatrix} -1 \\ 3/2 \end{bmatrix} \begin{bmatrix} -2 \\ 1/2 \end{bmatrix}^T = \begin{bmatrix} 5/9 & 1/9 \\ 2/3 & 1/3 \end{bmatrix}.\end{aligned}$$

Using the SMW formula, the inverses of our quasi-Newton updates can be written using similarly simple updating formulae.

Example. Suppose B_n is invertible and $y_n^T d_n > 0$, and let B_{n+1} be the BFGS update (35). Then its inverse is given by

$$B_{n+1}^{-1} = (I - \rho_n d_n y_n^T) B_n^{-1} (I - \rho_n y_n d_n^T) + \rho_n d_n d_n^T. \tag{36}$$

Note. It is fairly tedious to prove this (even given the formula).

By exchanging B_n and B_n^{-1} as well as y_n and d_n, we see immediately that the inverse of the BFGS update takes the form of the DFP update and vice-versa (immediately providing us also with a formula for the inverse of (34)).

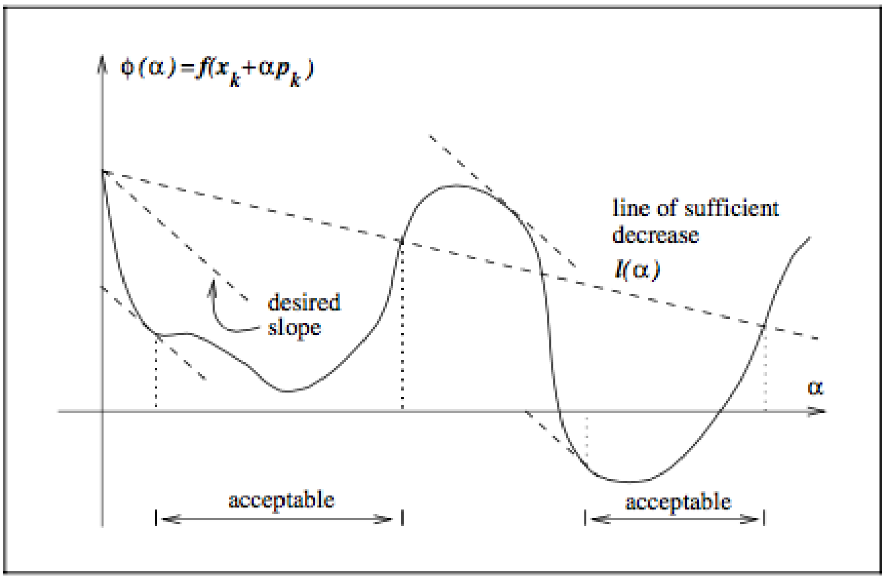

Figure 4: Geometric interpretation of the Wolfe conditions.

6.4 The Wolfe conditions

For the DFP and the BFGS updates (34) and (35) to be well-defined we require that $y_n^T d_n \neq 0$. In fact, if $y_n^T s_n > 0$ then $B_n > 0 \Rightarrow B_{n+1} > 0$.

To ensure this, in addition to the sufficient decrease (or Armijo) condition (12), i.e. $f(x_n + \alpha_n s_n) \leq f(x_n) + \theta_{sd}\alpha_n \nabla f_n^T s_n$, we also require that the *curvature condition*

$$\nabla f(x_n + \alpha_n s_n) \cdot s_n \geq \theta_c \nabla f_n^T s_n \tag{37}$$

is satisfied, for some $\theta_c \in (\theta_{sd}, 1)$. Together, (12) and (37) are called the *Wolfe conditions.* (Intuitively, (37) prevents the linesearch from stopping when significant further progress can be made, cf. Fig. 4 for a geometric interpretation.) A typical value for θ_c is 0.9.

Lemma 6.3. *Let s_n be a descent direction at x_n, and let $x_{n+1} = x_n + \alpha_n s_n$ satisfy* (37)*. Then*

$$y_n^T d_n > 0.$$

Proof. Note that (37) is equivalent to

$$y_n^T d_n = (\nabla f_{n+1} - \nabla f_n)^T (x_{n+1} - x_n) \geq (\theta_c - 1)\alpha_n \nabla f_n^T s_n\,.$$

Since s_n is a descent direction and $\theta_c < 1$, the right-hand side is positive. □

Now consider the following practical linesearch algorithm that guarantees the Wolfe conditions.

Algorithm 6.1 (WLINESEARCH).

```
Input: $x, s$ s.t. $\nabla f(x)^T s < 0, 0 < \theta_{sd} < \theta_c < 1$
Output: $\alpha > 0$ s.t. (12) and (37) are satisfied
$\alpha \leftarrow 1$, $\underline{\alpha} \leftarrow 0$, $\overline{\alpha} \leftarrow 0$;
while either (12) or (37) fails do
    if (12) fails then
        $\overline{\alpha} \leftarrow \alpha$; $\alpha \leftarrow \frac{1}{2}(\underline{\alpha} + \overline{\alpha})$;    % Reduce $\alpha$
    else if (37) fails then
        $\underline{\alpha} \leftarrow \alpha$;                          % Increase $\alpha$
        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 $\alpha$;
```

Proposition 6.4. *Suppose $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$ is bounded below. Then Algorithm 6.1 terminates in a finite number of iterations and returns a steplength α that satisfies* (12) *and* (37)*.*

Proof. Left as an **Exercise**. (See [4, Problem 6.4] for a proof strategy.) □

Note. If $\theta_{sd} < 1/2$, then, for n sufficiently large, the steplength $\alpha_n = 1$ satisfies the Wolfe conditions (12) and (37) and we recover the basic QN iteration in (28).

6.5 The BFGS method

Lemma 6.5. *Suppose that $B_n = B_n^T > 0$ and $y_n^T d_n > 0$. Then B_{n+1} obtained via the BFGS update* (35) *satisfies $B_{n+1} = B_{n+1}^T > 0$.*

Proof. Left as an **Exercise**. □

Algorithm 6.2 (A Simple BFGS Algorithm).

```
Input: $x_0 \in \mathbb{R}^N$, $B_0^{-1} \in \mathbb{R}^{N \times N}$ spd, $0 < \theta_{sd} < \theta_c < 1$;
for $n = 0, 1, 2, \dots$ do
    $s_n \leftarrow -B_n^{-1} \nabla f_n$;
    $\alpha_n \leftarrow$ WLINESEARCH[$x = x_n$,$s = s_n$,$\theta_{sd}, \theta_c$];
    $x_{n+1} \leftarrow x_n + \alpha_n s_n$;
    $d_n \leftarrow x_{n+1} - x_n$; $y_n \leftarrow \nabla f_{n+1} - \nabla f_n$;
    Update $B_{n+1}^{-1}$ using (36);
end for
```

Remark 6.6.

(a) Lemma 6.5 shows that this BFGS method is *well-defined.*

(b) Algorithm 6.2 reduces the cost of Algorithm 4.3 from $O(N^3)$ operations (LU factorisation of B_n) to $O(N^2)$ operations (low-rank matrix update plus matrix multiplication), which is particularly useful for large-scale systems ($N \gg 1$). In fact, if $B_0 = I$ (or sparse) and $n \ll N$, by storing d_k, y_k for all $k < n$, we do not even have to explicitly form B_n and can multiply with B_n in $O(Nn)$ operations. (**Exercise.**)

(c) As stated already, for sufficiently large n, we have $\alpha_n = 1$ and B_n satisfies the Dennis-Moré condition (29). Thus, it follows from Theorem 6.1 that Algorithm 6.2 converges locally superlinearly.

(d) The global convergence theory is somewhat incomplete. See [3, Sec. 8.4] for a proof under fairly strong assumptions.

7 Optimality Conditions for Constrained Optimisation

Having established a broad and satisfactory theory for unconstrained optimisation, we now turn to the harder problem of constrained optimisation. We will start by deriving the counterparts of the optimality conditions in Section 2.3.

7.1 A basic first-order optimality condition

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

Let $\mathcal{E} = \{1, \dots, M_e\}$, $\mathcal{I} = \{M_e + 1, \dots, M\}$ and $M_i = M - M_e$, and recall from Section 1.3 the general definition of the constrained optimisation problem

$$\min_{\Omega} f(x) \tag{38}$$

and of the *admissible set*

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

where $c_1, \dots, c_{M_e}$ are the equality constraints and $c_{M_e+1}, \dots, c_M$ are the inequality constraints. For the remainder, we assume that $c = (c_i)_{j=1}^M \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R}^M)$.

Definition 7.1 (Tangent Cone). See Figure 5 for an illustration.

(a) Let $\gamma \in \mathrm{C}^2((-\varepsilon, \varepsilon), \mathbb{R}^N)$, for some $\varepsilon > 0$. The path γ is called *admissible* if $\gamma|_{(0,\varepsilon)} \subset \Omega$.

(b) Let $x \in \Omega$. The *tangent cone* $T_\Omega(x)$ of Ω at x is the set of all vectors $d \in \mathbb{R}^N$ for which there exists an admissible path[2] γ with $\gamma(0) = x$ and $\gamma'(0) = d$.

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

Proposition 7.2 (First-order Optimality Condition). *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$, that $c \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R}^M)$ and that x_* is a local minimiser of f in Ω. Then,*

$$\nabla f(x_*)^T d \geq 0, \qquad \textit{for all} \;\; d \in T_\Omega(x_*). \tag{39}$$

Proof. Since x_* is a local minimiser of f in Ω, there exists $r > 0$ such that

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

Now, let $d \in T_\Omega(x_*)$ and let γ be an admissible path at x_* with $\gamma(0) = x_*$ and $\gamma'(0) = d$. Then,

$$\exists r \in (0, \varepsilon] : \quad f(\gamma(t)) \geq f(\gamma(0)) = f(x_*), \quad \text{for all } \; t \in (0, r).$$

Since $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$ and $\gamma \in \mathrm{C}^2((-\varepsilon, \varepsilon), \mathbb{R}^N)$, this implies that

$$0 \leq \lim_{t \searrow 0} \frac{f(\gamma(t)) - f(\gamma(0))}{t} = \frac{\mathrm{d}f(\gamma(t))}{\mathrm{d}t}\bigg|_{t=0} = \nabla f(\gamma(0))^T \gamma'(0) = \nabla f(x_*)^T d.$$

□

[2] The condition that γ is twice continuously differentiable may be restrictive in some instances, but it is sufficient for our puposes, as we will discover below. It is not necessary (cf. [3, Ch. 12] where a proof based on admissible sequences is given).

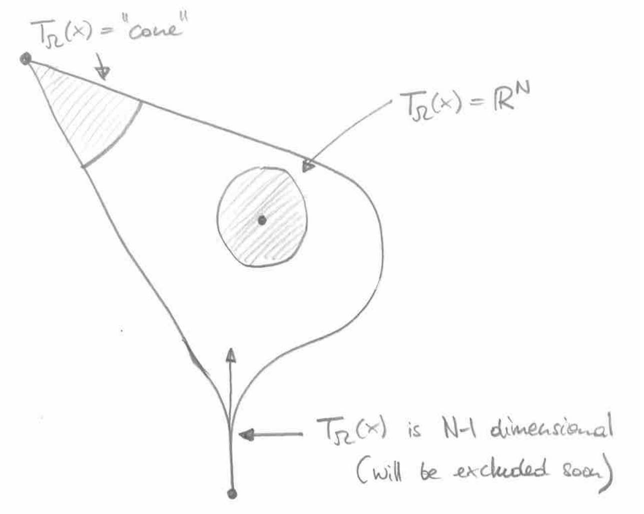
$T_\Omega(x)$ = "cone"
$T_\Omega(x) = \mathbb{R}^N$
$T_\Omega(x)$ is N-1 dimensional
(will be excluded soon)

Figure 5: Geometric illustration of the tangent cone in three typical cases.

Second-order optimality conditions are a bit more tricky (see Thm. 7.8 below).

7.2 The linearised tangent cone

Unfortunately the optimality condition in (39) is not very useful in practice. Let us start by giving a more practical characterisation of the tangent cone (excluding some pathologies along the way).

Let us first define the set of active constraints, the *active set*, for any $x \in \Omega$:

$$\mathcal{A}(x) := \big\{ j \in \mathcal{E} \cup \mathcal{I} : c_j(x) = 0 \big\}.$$

Obviously, $\mathcal{E} \subset \mathcal{A}(x)$, for all $x \in \Omega$. The *inactive set* is defined as

$$\mathcal{A}'(x) := \mathcal{I} \setminus \mathcal{A}(x) = \big\{ j \in \mathcal{I} : c_j(x) > 0 \big\}.$$

Since c is continuous, it follows that $\mathcal{A}'(x) \subset \mathcal{A}'(y)$, for all y in a neighbourhood of x, i.e. we can simply ignore inactive constraints $c_j(x) > 0$, $j \in \mathcal{A}'(x)$.

To eliminate certain pathologies (like cusps in Ω) we require a crucial technical condition which we will employ throughout rest of the course.

Definition 7.3. Suppose $c \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R}^M)$. 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.

The following Lemma provides a characterisation of the tangent cone in terms of the set of linearised admissible directions.

Lemma 7.4. *Let $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^M)$ and $x \in \Omega$. Then*

$$T_\Omega(x) \subset \mathcal{F}(x),$$

the linearised tangent cone

$$\mathcal{F}(x) := \big\{ d \in \mathbb{R}^N : \nabla c_j(x)^T d = 0 \;\; \forall j \in \mathcal{E} \;\; \textit{and} \;\; \nabla c_j(x)^T d \geq 0 \;\; \forall j \in \mathcal{I} \cap \mathcal{A}(x) \big\}.$$

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

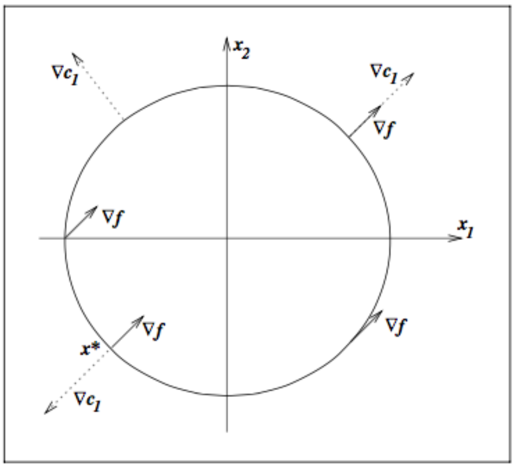

Figure 6: Illustration of Example 7.6.

Proof. We will only prove $T_\Omega(x) \subset \mathcal{F}(x)$. For a proof of the reverse inclusion in the case that LICQ holds, see [4, Lem. 7.5]. It uses the *Implicit Function Theorem* [4, Thm. 2.5] to prove the existence of an admissible path, which is why we require that $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^M)$.

Let $d \in T_\Omega(x)$ and let γ be an admissible path with $\gamma(0) = x$ and $\gamma'(0) = d$. By definition, we have $\gamma(t) \in \Omega$, for all $t \in (0,\varepsilon)$, and thus $c_j(\gamma(t)) = 0$ and $c_j(\gamma(t)) \geq 0$, for $j \in \mathcal{E}$ and for $j \in \mathcal{I}$, respectively. Also, $c_j(\gamma(0)) = c_j(x) = 0$, for all $j \in \mathcal{A}(x)$. Hence,

$$\nabla c_j(x)^T d = \nabla c_j(\gamma(0))^T \gamma'(0) = \left.\frac{\mathrm{d}c_j(\gamma(t))}{\mathrm{d}t}\right|_{t=0} = \lim_{t \searrow 0} \frac{c_j(\gamma(t)) - c_j(\gamma(0))}{t} \begin{cases} = 0, & j \in \mathcal{E}, \\ \geq 0, & j \in \mathcal{I} \cap \mathcal{A}(x). \end{cases}$$

□

In the special case where $\mathcal{I} = \emptyset$, we can easily deduce the following corollary of Proposition 7.2.

Corollary 7.5. *Suppose that $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$ and $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^M)$ with $M = M_e$ and $M_i = 0$ (i.e. Ω only contains equality constraints). Let x_* be a local minimiser of f in Ω and assume the LICQ holds at x_*. Then,*

$$\nabla f(x_*) \in \mathit{span}\big\{\nabla c_j(x_*) : j \in \mathcal{E}\big\}. \tag{40}$$

Proof. We have $T_\Omega(x_*) = \big\{d \in \mathbb{R}^N : \nabla c_j(x_*)^T d = 0 \quad \forall j \in \mathcal{E}\big\}$, due to Lemma 7.4. This implies that if $d \in T_\Omega(x_*)$ then also $-d \in T_\Omega(x_*)$, and (39) becomes

$$\nabla f(x_*)^T d = 0, \quad \text{for all } \; d \in T_\Omega(x_*).$$

Due to the LICQ, this is equivalent to $\nabla f(x_*) \in \mathrm{span}\big\{\nabla c_j : j \in \mathcal{E}\big\}$. □

Example 7.6. Consider the two-dimensional, linear objective function $f(x) = x_1 + x_2$ with a single equality constraint $c_1(x) = x_1^2 + x_2^2 - 2 = 0$.

We can see by inspection that the admissible set for this problem is the circle of radius $\sqrt{2}$ centred at the origin (cf. Fig. 6). The unique minimiser x_* is clearly $(-1,-1)^T$. From any other point x on the circle, it is easy to find a way to move that stays admissible (i.e. remains on the circle) while decreasing f.

We also see from Fig. 6 that clearly $\nabla f(x_*)$ is parallel to $\nabla c_1(x_*)$, i.e. there exists a scalar λ_* such that $\nabla f(x_*) = \lambda_* \nabla c_1(x_*)$, as predicted in Corollary 7.5.

7.3 The Karush–Kuhn–Tucker conditions

From now on, we assume throughout that x_* is a local minimiser of f in Ω and that the LICQ holds at x_*. The simple characterisation of the first-order optimality condition in Corollary 7.5 can be generalised to the case of inequality constraints and leads to the famous *Karush–Kuhn–Tucker (KKT) conditions* and to the *Lagrange multiplier method.*

For example, considering Example 7.6 with inequality constraint $-c_1 \geq 0$ (i.e. Ω is the interior of the disk with radius $\sqrt{2}$) we obtain the same minimiser and again $\nabla f(x_*) = \lambda_* \nabla c_1(x_*)$. But, for inequality constraints, the sign of λ_* matters; clearly, for $c_1 \geq 0$ (i.e. Ω is the exterior of the disk with radius $\sqrt{2}$), $x_* = (-1,-1)^T$ is not a minimiser, even though we still have $\nabla f(x_*) = \lambda_* \nabla c_1(x_*)$.

Theorem 7.7. *Let $f \in \mathrm{C}^1(\mathbb{R}^N;\mathbb{R})$, $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^M)$ and let x_* be a local minimiser of f in Ω at which the LICQ holds. Then, there exists $\lambda_* \in \mathbb{R}^M$ such that*

$$\nabla f(x_*) = \sum_{j=1}^{M} \lambda_{*,j} \nabla c_j(x_*), \tag{41a}$$

$$c_j(x_*) = 0 \qquad j \in \mathcal{E} \tag{41b}$$

$$c_j(x_*) \geq 0 \qquad j \in \mathcal{I} \tag{41c}$$

$$\lambda_{*,j} \geq 0 \qquad \forall j \in \mathcal{I} \tag{41d}$$

$$\lambda_{*,j} c_j(x_*) = 0 \qquad \forall j \in \mathcal{E} \cup \mathcal{I}. \tag{41e}$$

Geometric Interpretation of Theorem 7.7.

- Conditions (41b) and (41c) are obvious.
- The sets $\mathcal{M}_j := \{x \in \Omega : c_j(x) = 0\}$ are hypersurfaces with normals $\nabla c_j(x)$. These normals must point *into* the admissible set Ω for active inequality constraints, $j \in \mathcal{I} \cap \mathcal{A}(x_*)$ (for equality constraints, $j \in \mathcal{E}$, the normal may point inwards or outwards). Hence (41a) and (41d) are simply translating condition (39) that $-\nabla f(x_*)$ has no component pointing into the admissible set. edited 22 Apr end edit
- Finally, (41e) states that, if $c_j(x_*) > 0$ for some $j \in \mathcal{I}$ (i.e. $j \in \mathcal{A}'(x_*)$) then $\lambda_{*,j} = 0$, i.e., this constraint is simply irrelevant for the problem (at least in a neighbourhood of x_*).

Proof of Theorem 7.7. As stated above, it is clear that (41b) and (41c) are satisfied. Setting $\lambda_{*,j} = 0$, for all $j \in \mathcal{A}'(x_*)$, we can also ensure that (41e) holds.

Now, we can assume, without loss of generality, that $\mathcal{A}(x_*) = \mathcal{E} \cup \mathcal{I}$. Then, the existence of a $\lambda_* \in \mathbb{R}^M$ such that (41a) holds follows from Corollary 7.5.

It only remains to show that $\lambda_{*,j} \geq 0$, for $j \in \mathcal{I}$. Since the LICQ holds at x_*, there exists a $d \in T_\Omega(x_*)$ such that

$$\nabla c_j(x_*)^T d > 0 \quad \text{and} \quad \nabla c_k(x_*)^T d = 0, \quad \text{for } k \neq j.$$

This can, for example, be achieved by an orthogonalisation procedure. Now,

$$0 \leq \nabla f(x_*)^T d = \lambda_{*,j} \nabla c_j(x_*)^T d.$$

Since $\nabla c_j(x_*)^T d > 0$, this implies $\lambda_{*,j} \geq 0$. □

When no inequality constraints are present, the KKT conditions can be formulated in a very compact way using the *Lagrangian* associated with the constrained optimisation problem (38), i.e. the functional $\mathcal{L} \in \mathrm{C}^1(\mathbb{R}^{N+M}; \mathbb{R})$ defined by

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{j=1}^{M} \lambda_j c_j(x) = f(x) - \lambda^T c(x). \tag{42}$$

In that case, the KKT conditions (41) reduce to solving the nonlinear system

$$\nabla_{x,\lambda} \mathcal{L}(x, \lambda) = \begin{pmatrix} \nabla f(x) - \nabla c(x)\lambda \\ -c(x) \end{pmatrix} = 0. \tag{43}$$

We will return to this in Section 8.

However, the Lagrangian plays an important role in any constrained optimisation problem, in particular in the definition of second-order optimality conditions.

7.4 Second-order optimality conditions

Any pair (x_*, λ_*) which satisfies the KKT conditions (41) is called a *KKT point*, the equivalent of a first-order critical point in the unconstrained case.

Let us now discuss second-order necessary and sufficient optimality conditions to check whether a KKT point is a local minimiser for (38) or not. For this purpose, we first define, for any KKT point (x_*, λ_*), the *critical cone*

$$\mathcal{C}(x_*, \lambda_*) = \big\{ d \in T_\Omega(x_*) : \lambda_{*,j} \nabla c_j(x_*)^T d = 0, \quad \text{for all } j \in \mathcal{E} \cup \mathcal{I} \big\}. \tag{44}$$

Let us discuss $\mathcal{C}(x_*, \lambda_*)$. The conditions on $\lambda_{*,j} \nabla c_j(x_*)^T d$ in (44), for $j \in \mathcal{E} \cup \mathcal{A}'(x_*)$, are in fact irrelevant and do not actually restrict $T_\Omega(x_*)$. If $\mathcal{I} = \emptyset$ (i.e. Ω contains only equality constraints), then $\mathcal{C}(x_*, \lambda_*) = T_\Omega(x_*)$.

Now, if $d \in T_\Omega(x_*) \backslash \mathcal{C}(x_*, \lambda_*)$, then there exists a $j \in \mathcal{I} \cap \mathcal{A}(x_*)$ such that $\lambda_{*,j} \nabla c_j(x_*)^T d > 0$. Since $d \in \mathcal{F}(x_*)$, it follows from (41a) and (41d) that for any admissible path γ with $\gamma(0) = x_*$ and $\gamma'(0) = d$ we have

$$\left. \frac{\mathrm{d} f(\gamma(t))}{\mathrm{d}t} \right|_{t=0} = \nabla f(x_*)^T d = \sum_{j=1}^{M} \lambda_{*,j} \nabla c_j(x_*)^T d > 0$$

and so f is strictly increasing along γ.

If, on the other hand, $d \in \mathcal{C}(x_*, \lambda_*)$ and $\lambda_{*,j} \nabla c_j(x_*)^T d = 0$, for all $j \in \mathcal{I} \cap \mathcal{A}(x_*)$, then

$$\nabla f(x_*)^T d = \sum_{j=1}^{M} \lambda_{*,j} \nabla c_j(x_*)^T d = 0$$

and we need second derivative information to decide whether x_* is a local minimiser of f in Ω.

Theorem 7.8 (Second-order optimality conditions). *Suppose that $f \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R})$ and $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^M)$, and let $x_* \in \Omega$ where the LICQ holds.*

(a) If x_* is a local minimiser of f in Ω then the KKT conditions (41) are satisfied at x_* and

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

where $\nabla_x^2 \mathcal{L}(x, \lambda) = \nabla_x^2 f(x) - \sum_{j=1}^M \lambda_j \nabla_x^2 c_j(x)$.

(b) If the KKT conditions (41) are satisfied at x_* and

$$d^T \nabla_x^2 \mathcal{L}(x_*, \lambda_*) d > 0 \qquad \textit{for all} \;\; d \in \mathcal{C}(x_*, \lambda_*) \setminus \{0\}, \tag{45}$$

then x_* is a strict local minimiser of f in Ω.

Proof. Skipped. See [4, Thm. 7.8] for a proof of Part (b).

□

We will see these conditions in action in a couple of examples in the next section.

8 The Method of Lagrange Multipliers

For purely equality constrained problems, the reformulation (43) of the KKT conditions (41) together with the second-order optimality condition (45) in Theorem (7.8) provides a clear solution recipe, called the *Method of Lagrange Multipliers* or *Sequential Quadratic Programming (SQP)*. We will discuss this in the next subsection.

In the general 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. This is called the *Active Set Method* and will be described on an example in Section 8.2.

8.1 Equality constraints – sequential quadratic programming

We start by considering the special case of a quadratic objective function $f(x) := \frac{1}{2}x^T Hx - g^T x$ and a set of linear (equality) constraints $c(x) := b - Ax$, where $H \in \mathbb{R}^{N\times N}$, $A \in \mathbb{R}^{M\times N}$ with $M \le N$, $b \in \mathbb{R}^M$ and $g \in \mathbb{R}^N$. The resulting constrained optimisation problem

$$\min_{Ax=b} \tfrac{1}{2}x^T Hx - x^T g \tag{46}$$

is called a *quadratic program.*

If $\mathrm{rank}(A) = M$ and H is positive definite on $\ker(A)$, then it can be shown that (46) has a unique solution and the KKT conditions for (46) can be written (in block matrix form) as

$$\begin{pmatrix} H & -A^T \\ -A & 0 \end{pmatrix}\begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} g \\ -b \end{pmatrix}, \tag{47}$$

(cf. **Problem Sheet 6**).

In the general case, we can solve the nonlinear system $\nabla_{x,\lambda}\mathcal{L} = 0$ in (43) via Newton's method or any of the other methods described in Sections 3-6.

Suppose that $f \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R})$, $c \in \mathrm{C}^2(\mathbb{R}^N;\mathbb{R}^M)$ (again understood as equality constraints) and that $(x_0, \lambda_0) \in \mathbb{R}^{N+M}$ is a starting guess that is sufficiently close to a KKT point (x_*, λ_*) for this problem. Then we can apply Newton's method to compute a sequence $(x_n, \lambda_n)_{n\ge 0}$ that converges to (x_*, λ_*) by solving in each step a quadratic program. This method is presented in Algorithm 8.1.

Algorithm 8.1 (A simple Sequential Quadratic Programming (SQP) iteration).

```
Input: $(x_0, \lambda_0) \in \mathbb{R}^{N+M}$
for $n = 0, 1, 2, \ldots$ do
    Solve $\begin{pmatrix} \nabla_x^2\mathcal{L}(x_n,\lambda_n) & -\nabla c(x_n) \\ -\nabla c(x_n)^T & 0 \end{pmatrix} s_n = -\nabla_{x,\lambda}\mathcal{L}(x_n,\lambda_n)$;
    $\begin{pmatrix} x_{n+1} \\ \lambda_{n+1} \end{pmatrix} \leftarrow \begin{pmatrix} x_n \\ \lambda_n \end{pmatrix} + s_n$;
end for
```

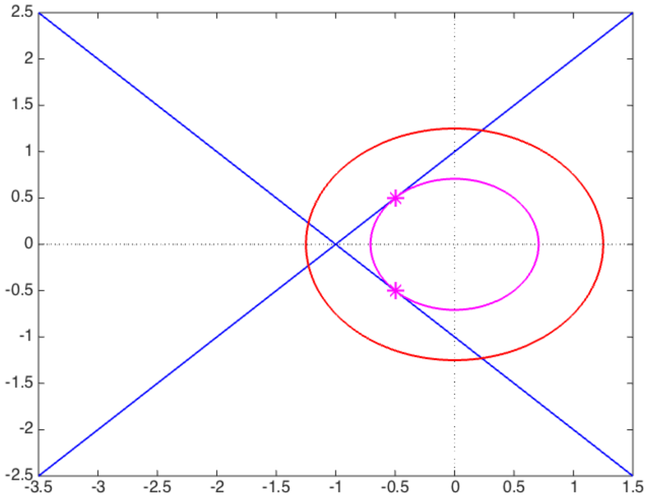

Figure 7: Illustration of Example 8.1. The constraint is a (degenerate) hyperbola. The objective function is a circle. The minima lie at $(-1/2, \pm 1/2)^T$ where the two curves are tangent.

Under appropriate conditions on f and c and provided the initial guess (x_0, λ_0) is sufficiently close to (x_*, λ_*) this SQP iteration is locally q-quadratically convergent (cf. **Problem Sheet 6**).

Example 8.1. Consider the equality constrained optimisation problem with objective function $f(x) := x_1^2 + x_2^2$ and constraint $c(x) := x_2^2 - (x_1+1)^2 = 0$ in $\mathbb{R}^2$ (cf. Fig. 7). Then

$$\mathcal{L}(x,\lambda) = x_1^2 + x_2^2 - \lambda\Big(x_2^2 - (x_1+1)^2\Big)$$

$$r := \nabla_{x,\lambda}\mathcal{L}(x,\lambda) = \begin{pmatrix} 2x_1 + 2\lambda(x_1+1) \\ 2x_2 - 2\lambda x_2 \\ (x_1+1)^2 - x_2^2 \end{pmatrix}$$

$$\mathcal{H} := \begin{pmatrix} \nabla_x^2\mathcal{L}(x,\lambda) & -\nabla c(x) \\ -\nabla c(x)^T & 0 \end{pmatrix} = \begin{pmatrix} 2(1+\lambda) & 0 & 2(x_1+1) \\ 0 & 2(1-\lambda) & -2x_2 \\ 2(x_1+1) & -2x_2 & 0 \end{pmatrix}$$

Applying Algorithm 8.1 with $x_0 = (0,1)$, $\lambda_0 = 0$, we get $r_0 = (0,2,0)^T$ and the first Newton system is

$$\begin{pmatrix} 2 & 0 & 2 \\ 0 & 2 & -2 \\ 2 & -2 & 0 \end{pmatrix} \begin{pmatrix} s_{0,1} \\ s_{0,2} \\ s_{0,3} \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix}.$$

From the first and third equation, we see immediately that $s_{0,1} = s_{0,2} = -s_{0,3}$. Thus, it follows from the second equation that $s_0 = (-1/2, -1/2, 1/2)^T$ and so $x_1 = (-1/2, 1/2)$, $\lambda_1 = 1/2$ and $r_0 = (-1/2, 1/2, 0)^T$. The next Newton system is

$$\begin{pmatrix} 3 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} s_{1,1} \\ s_{1,2} \\ s_{1,3} \end{pmatrix} = \begin{pmatrix} 1/2 \\ -1/2 \\ 0 \end{pmatrix}.$$

From the third equation we see that $s_{1,1} = s_{1,2}$. But adding the first and the second equation, we deduce that then $s_{1,1} = s_{1,2} = 0$. It follows from the first equation that $s_{1,3} = 1/2$. Therefore, $x_2 = (-1/2, 1/2)$, $\lambda_2 = 1$ which satisfies $\nabla_{x,\lambda}\mathcal{L}(x_2, \lambda_2) = 0$ and is thus a KKT point. The SQP method has converged in two iterations.

Let us check that the second order optimality condition holds at $(x_*, \lambda_*) = (x_2, \lambda_2)$. Since we have only equality constraints and $\nabla c(x_*) = (1, -1)^T$,

$$\mathcal{C}(x_*, \lambda_*) = \mathcal{F}(x_*) = \{d \in \mathbb{R}^2 : \nabla c(x_*)^T d = 0\} = \{(d_1, d_1)^T : d_1 \in \mathbb{R}\}.$$

Hence,

$$d^T \nabla_x^2 \mathcal{L}(x_*, \lambda_*) d = \begin{pmatrix} d_1 \\ d_1 \end{pmatrix}^T \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} d_1 \\ d_1 \end{pmatrix} = 4d_1^2 > 0 \quad \forall d \in \mathcal{C}(x_*, \lambda_*)\backslash\{0\}.$$

Again we saw how well Newton's method works (when it works!). Unfortunately it will fail for starting guesses that are too far away from a minimum.

For example, picking any starting guess with $x_{0,2} = 0$ in Example 8.1, we can see that the second row of the Newton system becomes $2(1 - \lambda_0)s_{0,2} = 0$ and so $x_{n,2} = 0$, for all $n \geq 1$, and the method does not converge to a KKT point. Thus, in Section 9, we will again address the question of how to construct a globally convergent scheme.

8.2 Inequality & equality constraints – the active set method

Unfortunately, when $\mathcal{I} \neq \emptyset$, i.e. for inequality constrained problems, no such simple method exists. One way to circumvent this problem, especially if the number M_i of inequality constraints is small, is to consider each case of $\mathcal{A}(x_*)$ separately, the *active set method*, which reduces the inequality constrained problem to a set of equality constrained ones. We illustrate this method on an example.

Example 8.2. Consider the objective function $f(x) = x_1^3 + x_2$ with equality constraint $c_1(x) := x_1^2 + 2x_2^2 - 1 = 0$ and inequality constraint $c_2(x) := x_1 \geq 0$ in $\mathbb{R}^2$ (cf. Fig. 8). We have

$$\nabla f(x) = \begin{pmatrix} 3x_1^2 \\ 1 \end{pmatrix}, \quad \nabla c_1(x) = \begin{pmatrix} 2x_1 \\ 4x_2 \end{pmatrix} \quad \text{and} \quad \nabla c_2(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

and the KKT conditions in (41) become:

$$\begin{array}{rclcrclc} 3x_1^2 - 2x_1\lambda_1 - \lambda_2 &=& 0 & \text{(i)} & x_1 &\geq& 0 & \text{(iv)} \\ 1 - 4x_2\lambda_1 &=& 0 & \text{(ii)} & \lambda_2 &\geq& 0 & \text{(v)} \\ x_1^2 + 2x_2^2 - 1 &=& 0 & \text{(iii)} & \lambda_2 x_1 &=& 0 & \text{(vi)} \end{array} \qquad (48)$$

The active set method now looks at the cases $\mathcal{A}(x_*) = \{1\}$ and $\mathcal{A}(x_*) = \{1, 2\}$ separately.

Case $\mathcal{A}(x_*) = \{1\}$: Then $x_1 > 0$, and so (48-vi) implies $\lambda_2 = 0$. It follows from (48-ii) that $x_2 = (4\lambda_1)^{-1}$ and dividing (48-i) by x_1, it follows that $\lambda_1 = \frac{3}{2}x_1$. Substituting these two equations into (48-iii) we can deduce that

$$x_1^4 - x_1^2 + \frac{1}{18} = 0.$$

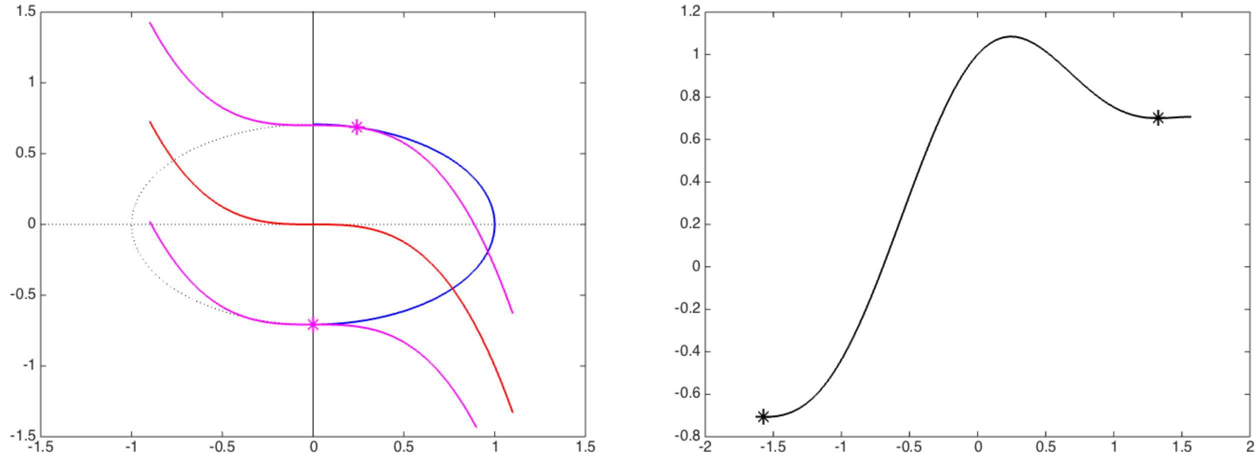

Figure 8: Illustration of Example 8.2. The equality constraint is an ellipse. The inequality constraint restricts Ω to the right half plane. The objective function is a cubic which is tangent at $x = (0.243, 0.686)^T$. This is a local minimum since $f(x)$ grows to the left and right of that point. The global minimum is at $x = (0, -\sqrt{2}/2)^T$, where the inequality constraint becomes active. The figure on the right visualises f on Ω as a function of angle from $-\pi/2$ to $\pi/2$. The two minima a clearly visible.

This can be solved by substituting $y = x_1^2$ to give $y = (3 \pm \sqrt{7})/6 \approx 0.059$ and 0.941. Due to (48-iv), we are only interested in the positive roots of y, leaving two possible solutions

$$x^{(1)} = \begin{pmatrix} 0.243 \\ 0.686 \end{pmatrix} \quad \text{and} \quad x^{(2)} = \begin{pmatrix} 0.970 \\ 0.172 \end{pmatrix}$$

with $\lambda_1 > 0$ and objective function values $f(x^{(1)}) = 0.7003$ and $f(x^{(2)}) = 1.0845$, respectively.

Case $\mathcal{A}(x_*) = \{1, 2\}$: Then $x_1 = 0$ and (48-iii) implies $x_2 = \pm\sqrt{2}/2$. Due to (48-i), we have $\lambda_2 = 0$. Finally, it follows from (48-ii) that $\lambda_1 = (4x_2)^{-1} = \pm\sqrt{2}/4$ and so both points satisfy all the conditions in (48) – note that constraint c_1 is an equality constraint and so λ_1 is allowed to be negative.

Since $x_2 \neq 0$, ∇c_1 and ∇c_2 are linearly independent, so that the LICQ holds in both cases and we obtain two further KKT points

$$x^{(3)} = \begin{pmatrix} 0 \\ \sqrt{2}/2 \end{pmatrix} \quad \text{and} \quad x^{(4)} = \begin{pmatrix} 0 \\ -\sqrt{2}/2 \end{pmatrix}$$

with $f(x^{(3)}) = \sqrt{2}/2 \approx 0.707$ and $f(x^{(4)}) = -\sqrt{2}/2 \approx -0.707$, respectively.

Since these four points are the only ones satisfying the KKT conditions, the point with the smallest objective value, namely $x^{(4)}$, has to be the global minimum of the problem.

To decide whether $x^{(1)}, \ldots, x^{(3)}$ are local minimisers we need to check the second order optimality condition (45). First of all, we have

$$\nabla_x^2 \mathcal{L}(x, \lambda) = \begin{pmatrix} 6x_1 - 2\lambda_1 & 0 \\ 0 & -4\lambda_1 \end{pmatrix}$$

Now, we need to distinguish the two cases again.

Case $\mathcal{A}(x_*) = \{1\}$: Recall that here $\lambda_2 = 0$, $x_1 > 0$, $\lambda_1 = \frac{3}{2}x_1$ and $x_2 = (6x_1)^{-1}$. Therefore, the critical cone is

$$\mathcal{C}(x, \lambda) = \left\{ d \in \mathbb{R}^2 : \begin{pmatrix} 2x_1 \\ 4x_2 \end{pmatrix}^T d = 0 \right\} = \left\{ d \in \mathbb{R}^2 : d_2 = -\frac{x_1}{2x_2} d_1 \right\}$$

and thus, for any $d \in \mathcal{C}(x, \lambda)$,

$$d^T \nabla_x^2 \mathcal{L}(x, \lambda) d = (6x_1 - 2\lambda_1) d_1^2 - 4\lambda_1 \left(\frac{x_1}{2x_2} \right)^2 d_1^2 = 3x_1 d_1^2 - 6x_1(3x_1)^2 d_1^2 = 3x_1(1 - 18x_1^4) d_1^2.$$

If the RHS is positive at a KKT point, then the second order condition (45) holds and that point is a local minimiser. If it is negative the point is not a minimiser. (If it is zero, the second-order condition is also inconclusive.) Since $18(x_1^{(1)})^4 = 0.0628$ and $18(x_1^{(2)})^4 = 15.937$, it follows that $x^{(1)}$ is a local minimiser, while $x^{(2)}$ is **not**.

Case $\mathcal{A}(x_*) = \{1, 2\}$: Again $\lambda_2 = 0$ and $x_2 \neq 0$. Therefore, the critical cone is

$$\begin{aligned} \mathcal{C}(x, \lambda) &= \left\{ d \in \mathbb{R}^2 : \begin{pmatrix} 2x_1 \\ 4x_2 \end{pmatrix}^T d = 0 \;\; \& \;\; \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T d \geq 0 \right\} \\ &= \left\{ d \in \mathbb{R}^2 : d_2 = -\frac{x_1}{2x_2} d_1 \;\; \& \;\; d_1 \geq 0 \right\}. \end{aligned}$$

In this case, $x_1 = 0$, $x_2 \neq 0$ and $\lambda_1 = \pm\sqrt{2}/4$. Hence, for any $d \in \mathcal{C}(x, \lambda)$,

$$d^T \nabla_x^2 \mathcal{L}(x, \lambda) d = (6x_1 - 2\lambda_1) d_1^2 - 4\lambda_1 \left(\frac{x_1}{2x_2} \right)^2 d_1^2 = -2\lambda_1 d_1^2$$

Since $\lambda_1^{(3)} = \sqrt{2}/4 > 0$, $x^{(3)}$ is **no** local minimiser, and since $\lambda_1^{(4)} = -\sqrt{2}/4 < 0$, we can confirm that the global minimiser $x^{(4)}$ does indeed satisfy the second order condition (45).

9 Penalty and Augmented Lagrangian Methods

For simplicity, let $\mathcal{I} = \emptyset$, i.e., $M_i = 0$ and $M = M_e$. In this case, the KKT conditions simply become

$$\nabla_{x,\lambda}\mathcal{L}(x_*, \lambda_*) = 0. \tag{49}$$

As mentioned above, Newton's method, applied to (49) "typically" converges q-quadratically (cf. **Problem Sheet 6**). The question is again how to construct a globally convergent scheme. Unfortunately here, $\nabla^2_{x,\lambda}\mathcal{L}(x_*, \lambda_*)$ is intrinsically indefinite, and so steepest descent does not work directly. Instead, we will aim at replacing the constrained problem (38) by a sequence of unconstrained problems where violation of the constraint $c(x) = 0$ is *penalised.*

9.1 The ℓ^2-penalty method

Let us define the *merit function*

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

and minimise Φ for increasing values of μ, so that in the limit, as $\mu \to \infty$, it gives rise to a solution of the constrained minimisation problem.

Algorithm 9.1 (ℓ^2-penalty method).
Input: $x_0^S \in \mathbb{R}^N, \mu_0 > 0, \tau_0 > 0$

```
for $n = 0, 1, 2, \dots$ do
    Use an unconstrained optimisation method with starting guess $x_n^S$, compute $x_n$ such that $|\nabla_x \Phi(\mu_n; x_n)| \le \tau_n$;
    Choose $\mu_{n+1}, \tau_{n+1}$; $x_{n+1}^S \leftarrow x_n$;
end for
```

Typically, the choice of the updated values μ_{n+1} and τ_{n+1} depend on information about "how difficult" the optimisation problem in Step 2 had been. will enter the updating procedure for μ_n and τ_n. For the moment, we will only assume that $\mu_n \to \infty$ and $\tau_n \to 0$.

Since

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

if we choose

$$\lambda_n = \mu_n c(x_n), \tag{50}$$

we get

$$\nabla_x \Phi(\mu_n; x_n) = \nabla_x \mathcal{L}(x_n, \lambda_n) \quad \text{and} \quad c(x_n) = \frac{\lambda_n}{\mu_n} \to 0,$$

as $\mu_n \to \infty$ (provided λ_n is bounded).

Theorem 9.1. *Let $f \in \mathrm{C}^1(\mathbb{R}^N; \mathbb{R})$ and $c \in \mathrm{C}^2(\mathbb{R}^N; \mathbb{R}^M)$, and suppose that the sequence $\{x_n\}_{n \ge 0}$ generated by Algorithm 9.1 (with $\tau_n \to 0$ and $\mu_n \to \infty$) converges to a point x_* at* *which the LICQ holds. Furthermore, let λ_n be defined by (50). Then, $\lambda_* := \lim_{n\to\infty}\lambda_n$ exists and (x_*, λ_*) is a KKT point, i.e., $\nabla_{x,\lambda}\mathcal{L}(x_*, \lambda_*) = 0$.*

Proof. Skipped. For a proof, see [4, Thm. 8.1]. □

Let us discuss (informally) the local convergence properties of Algorithm 9.1 under the assumptions of Theorem 9.1 and assuming in addition that the strong second-order optimality condition (45) holds.

Taylor expansion around (x_*, λ_*), where $\nabla_{x,\lambda}\mathcal{L}(x_*, \lambda_*) = 0$, gives

$$\nabla_{x,\lambda}\mathcal{L}(x_n, \lambda_n) = \nabla^2_{x,\lambda}\mathcal{L}_* \begin{pmatrix} x_n - x_* \\ \lambda_n - \lambda_* \end{pmatrix} + o\Big(|x_n - x_*| + |\lambda_n - \lambda_*|\Big)$$

Since $\nabla^2_{x,\lambda}\mathcal{L}_*$ is invertible, for n sufficiently big, there exists a constant C independent of n such that

$$\left|\begin{pmatrix} x_n - x_* \\ \lambda_n - \lambda_* \end{pmatrix}\right| \le C \left|\begin{pmatrix} \nabla_x\Phi(\mu_n; x_n) \\ c(x_n) \end{pmatrix}\right| \le C \left|\begin{pmatrix} \tau_n \\ \lambda_n\mu_n^{-1} \end{pmatrix}\right| \sim \mu_n^{-1}.$$

This is a slow convergence rate which requires very large values of the penalty parameter μ_n for satisfactory accuracies. This, in turn, leads to extremely ill-conditioned unconstrained optimisation problems in Step 2 that may be difficult and unreliable to solve.

9.2 The augmented Lagrangian approach

A better approach is based on the augmented Lagrangian

$$\mathcal{L}_A(\mu; x, \lambda) := \mathcal{L}(x, \lambda) + \frac{\mu}{2}\sum_{j=1}^{M} |c_j(x)|^2.$$

We essentially replaces Φ by $\mathcal{L}_A$ in Algorithm 9.1, leaving also λ fixed at each iteration and updating it subsequently. It is based on the observation that

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

if (x_*, λ_*) is a KKT point, i.e., KKT points are always critical points of $\mathcal{L}_A$. More importantly, the additional penalisation turns x_* into a *local minimiser* (rather than a saddle point) of $\mathcal{L}_A(\mu; \cdot, \lambda_*)$, provided μ is sufficiently large.

Proposition 9.2. *Suppose that (x_*, λ_*) is a KKT point where the LICQ and the strong second-order optimality condition (45) hold. Then, there exists $\bar\mu > 0$ such that, for all $\mu \ge \bar\mu$, $\nabla^2_x\mathcal{L}_A(\mu; x_*, \lambda_*)$ is positive definite and x_* is a strict local minimiser of $\mathcal{L}_A(\mu; \cdot, \lambda_*)$ in $\mathbb{R}^N$.*

Proof.
It suffices to show that, for μ sufficiently big, $\nabla^2_x\mathcal{L}_A(\mu; x_*, \lambda_*)$ is positive definite.

First, note that

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

and let $\lambda_{\min} < 0$ and $\lambda_{\max} > 0$, respectively, denote the minimum and maximum eigenvalue of $\nabla_x^2 \mathcal{L}_* := \nabla_x^2 \mathcal{L}(x_*, \lambda_*)$.

Now, $h \in \mathcal{C}(x_*, \lambda_*)$ is equivalent to $\nabla c(x_*)^T h = 0$ and so (45) is equivalent to

$$h^T \nabla_x^2 \mathcal{L}_A(x_*, \lambda_*) h \geq c_0 |h|^2, \quad \text{for all } \; h \in \ker \nabla c(x_*)^T \tag{51}$$

and for some constant $c_0 > 0$, independent of h.

Also, due to LICQ, $\nabla c(x_*)$ is full rank and so

$$h^T \nabla c(x_*) \nabla c(x_*)^T h \geq c_1 |h|^2, \quad \text{for all } \; h \in \left(\ker \nabla c(x_*)^T\right)^\perp,$$

the orthogonal complement of $\ker\nabla c(x_*)^T$, for some constant $c_1 > 0$. Hence, if $\lambda_{\min} < 0$ is the minimum eigenvalue of $\nabla_x^2 \mathcal{L}(x_*, \lambda_*)$, then

$$h^T \nabla_x^2 \mathcal{L}_A(x_*, \lambda_*) h \geq (\lambda_{\min} + \mu c_1)|h|^2, \quad \text{for all } \; h \in \left(\ker \nabla c(x_*)^T\right)^\perp. \tag{52}$$

Let $h \in \mathbb{R}^N$. There exist unique $h_0 \in \ker\nabla c(x_*)^T$ and $h_1 \in \left(\ker\nabla c(x_*)^T\right)^\perp$, such that $h = h_0 + h_1$. Thus, using the Cauchy-Schwarz inequality (4) with $u = \varepsilon^{1/2} h_0$ and $v = \varepsilon^{-1/2} \nabla_x^2 \mathcal{L}_* h_1$, we get

$$|2 h_0^T \nabla_x^2 \mathcal{L}_* h_1| = 2|u^T v| \leq |u|^2 + |v|^2 \leq \varepsilon |h_0|^2 + \varepsilon^{-1} \lambda_{\max}^2 |h_1|^2. \tag{53}$$

Finally, combining this bound with the bounds in (51) and (52), we get

$$\begin{aligned} h^T \nabla_x^2 \mathcal{L}_A(\mu; x_*, \lambda_*) h &= h_0^T \nabla_x^2 \mathcal{L}_* h_0 \; + \; 2 h_0^T \nabla_x^2 \mathcal{L}_* h_1 \; + \; h_1^T \nabla_x^2 \mathcal{L}_* h_1 + \mu |\nabla c(x_*)^T h_1|^2 \\ &\geq (c_0 - \varepsilon)|h_0|^2 \; + \; (\mu c_1 + \lambda_{\min} - \varepsilon^{-1} \lambda_{\max}^2)|h_1|^2. \end{aligned}$$

Hence, setting $\epsilon := \frac{1}{2} c_0$, we deduce

$$h^T \nabla_x^2 \mathcal{L}_A(\mu; x_*, \lambda_*) h > 0, \quad \text{for all } \; h \neq 0 \; \text{ and } \; \mu > \overline{\mu} := \frac{2\lambda_{\max}^2}{c_0 c_1} - \frac{\lambda_{\min}}{c_1}.$$

□

It is clear from Proposition 9.2 that a good update for λ at each step of the augmented Lagrangian method will be crucial. Note that

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

Hence, a natural choice is

$$\lambda_{n+1} = \lambda_n - \mu_n c(x_n), \tag{54}$$

which implies

$$|c(x_n)| = \frac{|\lambda_n - \lambda_{n+1}|}{\mu_n}, \tag{55}$$

leading to a much faster convergence rate than in the ℓ^2-penalty method, especially if $\mu_n \to \infty$ (but that is not necessary here).

Algorithm 9.2 (Basic Augmented Lagrangian Algorithm).

```
Input: $x_0^S \in \mathbb{R}^N$, $\lambda_0 \in \mathbb{R}^M$, $\mu_0 > 0$, $\tau_0 > 0$;
1: for $n = 0, 1, 2, \dots$ do
2:     Using an unconstrained optimisation method with starting guess $x_n^S$, compute $x_n$
       such that $|\nabla_x \mathcal{L}_A(\mu_n; x_n, \lambda_n)| \le \tau_n$.
3:     $\lambda_{n+1} \leftarrow \lambda_n - \mu_n c(x_n)$;
4:     Choose $\mu_{n+1}$ and $\tau_{n+1}$, and set $x_{n+1}^S \leftarrow x_n$;
5: end for
```

Theorem 9.3. *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 9.2 (with $\tau_n \to 0$ and $\mu_n \to \infty$) converges to a point x_* at which the LICQ holds. Then, $\lambda_n \to \lambda_*$ and (x_*, λ_*) is a KKT point.*

Let us discuss again the local convergence properties of Algorithm 9.2. The result in Proposition 9.2 is of little practical value, since it assumes that λ_* is known. The following result gives conditions on λ_n under which $\nabla_x^2 \mathcal{L}_A(\mu_n; x_n, \lambda_n)$ is positive definite and $x_n \to x_*$ (locally) superlinearly.

edited
29 Apr

Theorem 9.4. *Suppose that the assumptions of Proposition 9.2 are satisfied at x_* and λ_*, and let $\overline{\mu}$ be as chosen in that theorem. If the sequences $\{x_n\}_{n\ge0}$ and $\{\lambda_n\}_{n\ge0}$ generated by Alg. 9.2 (with $\tau_n \to 0$) converge to x_* and λ_*, then there exists $n_0 \in \mathbb{N}$, such that, for all $n \ge n_0$ and $\mu_n \ge \overline{\mu}$, the matrix $\nabla_x^2 \mathcal{L}_A(\mu_n; x_n, \lambda_n)$ is positive definite and the unconstrained minimisation problem in Step 2 of Algorithm 9.2 has a unique solution x_n. Furthermore, there exists a positive constant C independent of n, such that*

$$|\lambda_{n+1} - \lambda_*| \le C \frac{|\lambda_n - \lambda_*|}{\mu_n}, \qquad \textit{for all } \; n \ge n_0. \tag{56}$$

and

$$|x_n - x_*| \le C \frac{|\lambda_n - \lambda_*|}{\mu_n}, \qquad \textit{for all } \; n \ge n_0. \tag{57}$$

end
edit

We can draw the following conclusions from this theorem:

- If $\mu_n \to \infty$, then it follows from (56) and (57) that $\lambda_n \to \lambda_*$ and $x_n \to x_*$ q-superlinearly.
- However, it is not necessary for $\mu_n \to \infty$. It suffices that $\mu_n \ge \max(\overline{\mu}, 2C^{-1})$, for all $n \ge n_0$. Then, $\lambda_n \to \lambda_*$ and $x_n \to x_*$ q-linearly.

A very good and popular implementation of the augmented Lagrangian method is the LANCELOT code by Conn, Gould and Toint (`http://www.swmath.org/software/500`).

edited 29 Apr

Remark 9.5.

(a) A merit function $\Phi(\mu; x)$ is called *exact* if there is a positive scalar $\overline{\mu}$ such that for any $\mu > \overline{\mu}$, all local solutions of the constrained problem are (unconstrained) local minimisers of $\Phi(\mu; x)$.

The merit function,

$$\Phi(\mu; x) = f(x) + \frac{\mu}{2}|c(x)|,$$

is exact and is called the *exact* ℓ^2 function. Note that the penalty term is $|c(x)|$, not $|c(x)|^2$ as considered in Section 9.1. Another example is ℓ^1 merit function,

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

Both of these are non-differentiable. An example of a smooth and exact merit function is Fletcher's augmented Lagrangian [3, Section. 15.4].

(b) Penalty and augmented Lagrangian methods (as well as their theory) can be extended to non-smooth penalty functions and to inequality constraints. For inequality constraints, the ℓ^2-penalty function would become

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

where $z^- = \min(0, z)$.

(c) An alternative approach for purely inequality constrained problems is the *(logarithmic) barrier method* (a type of *interior point method*). The barrier method makes a choice of merit function where only strictly admissible points have finite merit function value,

$$P(\mu; x) = \begin{cases} f(x) - \mu\sum_{j=1}^{M}\log c_j(x), & c_j(x) > 0, \quad \text{for all} \quad j \in \mathcal{I}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{58}$$

One of the challenges for this approach is to find an admissible starting point for the unconstrained optimisation problem at the heart of the algorithm. This is typically done using *primal-dual ideas*. If you are interested, see [4, Chapter 9] or [3, Section 17.2] for details.

end edit

10 Large Scale Systems

This chapter will not be examinable. Here are some ideas for further reading.

Conjugate Gradient Methods.

In truely large-scale problems, the Quasi-Newton methods described in Section 6 will also become too costly to apply. In general, they require storing an $N \times N$ dense approximation of the Hessian.

An alternative for computing good search directions is based on the *conjugate gradient (CG) method* [3, Chapter 5]. For a quadratic objective function

$$f(x) = \tfrac{1}{2}x^T Ax - b^T x,$$

we can construct a good search direction s_n at the nth iteration by making it *conjugate* – i.e., orthogonal in the A-inner product – to all previous search directions s_j, $j = 0, \dots, n-1$. In practice, this can be achieved very efficiently, using only the previous search direction s_{n-1}.

It can be shown that in exact arithmetic the CG method finds the minimum of any quadratic objective function in at most N iterations. More importantly, in general the convergence is significantly faster. It depends on the condition number of A and on how "clustered" the eigenvalues of A are. (See **MA30051** for details.)

The conjugate gradient idea can be extended to general nonlinear objective functions: *Fletcher-Reeves Method* or *Polak-Ribière Method* [3, Section 5.2]. To ensure global convergence (in the general nonlinear case), the CG method can again be combined with line search or trust region methods. One very popular method for large-scale systems is *Steihaug's Method* [3, Section 4.1]. It combines the CG Method with a dogleg-like trust region approach and a Quasi Newton approximation of the Hessian.

Inexact and Modified Newton Methods.

Another popular approach for large-scale optimisation problems is the use of *inexact Newton methods* [3, Chapter 6]. The Newton system is solved approximately, using an iterative method for linear systems, such as the CG method or another problem-specific approach. The quadratic convergence of Newton's method can be maintained, if the tolerance for the inexact solution of the Newton system decreases proportionally to $|\nabla f(x_n)|$ (cf. **Problem Sheet 5**).

If direct solvers are used in conjunction with line search, it is in general necessary to modify the Hessian to ensure it is positive definite. A popular approach for modifying a Hessian matrix that is not positive definite is based on a *modified Cholesky factorisation.* In that approach, the diagonal elements encountered during the factorisation process are increased if necessary [3, Section 6.3]. However, several other modified Newton approaches exist.

Nonlinear Least Squares.

An important special class of large-scale nonlinear optimisation problems are least squares problems, where the objective function takes the special form

$$f(x) = \tfrac{1}{2}\sum_{j=1}^{M} R_j^2(x),$$

and each R_j is a smooth function from $\mathbb{R}^N$ to $\mathbb{R}$ [3, Chapter 10]. This is also a way to combine multiple objectives.

As discussed on **Problem Sheet 3**, the gradient and the Hessian of f take a special form in that case, leading to simplications and particular approximations such as the *Gauss-Newton Hessian* or the popular *Levenberg-Marquardt Method* [3, Section 10.2]. There are strong links to statistics and inference methods.

Large-Scale Constrained Optimisation

An excellent code for large-scale constrained optimisation that contains the above mentioned LANCELOT implementation of the augmented Lagrangian method but also interior point methods and active set methods is the GALAHAD package by Gould, Orban and Toint (available at `http://www.swmath.org/software/1408`).

Acknowledgements

These lecture notes are based very closely on the lecture notes of C. Ortner [4]. These are themselves mainly based on the monograph by Nocedal & Wright [3]. Dennis & Schnabel [1] is a very good reference for the material in Chapter 3. A good introductory paper on constrained optimisation is Gould & Leyffer [2].

References

[1] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.

[2] N.I.M. Gould and S. Leyffer, An introduction to algorithms for nonlinear optimization, in *Frontiers in Numerical Analysis* (J Blowey, A Craig, T Shardlow, eds), Springer, 2003 (available at `http://www.numerical.rl.ac.uk/people/nimg/oumsc/lectures/paper.pdf`).

[3] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, New York, 1999.

[4] C. Ortner, *Continuous Optimization*, Lecture Notes, University of Oxford, 2009 (available at `https://homepages.warwick.ac.uk/staff/C.Ortner/teaching/files/opt_ln.pdf`).